Vlsi (4361102) - Problem Solver

Antigravity AI

June 4, 2026

Vlsi

First Edition: 2026

Author: Antigravity AI
Website: milav.in
YouTube: @milav.dabgar

To my students,
who constantly inspire me to find new analogies,
break down complex theories, and make learning
an engineered science.

Contents

Preface

About this Book

Welcome to **Computer Networks**! This textbook is meticulously designed to serve as a comprehensive problem solver and study companion. It provides a robust and intuitive foundation in the core principles of this subject, bridging the gap between theoretical models and practical applications. A unique feature of this book is its focus on decoding complex concepts using clear, step-by-step methodologies and real-world analogies.

Target Audience

This book is intended for students and professionals looking to master the concepts of Computer Networks. Whether you are revising for exams, seeking to solidify your fundamental understanding, or looking for practical examples to clarify abstract concepts, this book has been structured to support your learning journey. It serves as an accessible guide designed to remove the fear of complex topics, replacing it with an appreciation for the underlying mechanics.

Scope and Coverage

The coverage is divided logically to follow standard academic curriculums. Each chapter focuses on key topics, breaking them down into digestible sections:

- **Theoretical Insights** – Core concepts explained intuitively.
- **Method Blocks** – Standardized, step-by-step procedures for solving specific types of problems.
- **Worked Examples** – Fully solved problems that apply the methods in practice.

We hope this resource proves invaluable in your studies and helps you achieve excellence in **Computer Networks**. Happy learning!

Acknowledgments

Writing a comprehensive book that connects the fundamental forces of Chemistry with practical engineering applications requires more than just academic knowledge—it requires a community of support. I would like to express my sincere gratitude to everyone who contributed to the realization of this project.

Specially, I extend my heartfelt gratitude to the **Department of Technical Education (DTE), Government of Gujarat**, and **Government Polytechnic, Palanpur**, for providing an environment that fosters innovation, academic rigor, and continuous professional development.

I am deeply thankful to my family for their unwavering patience and support during the countless hours spent researching, formatting, and refining these chapters.

Finally, my deepest appreciation goes to my students. Your curiosity, challenging questions, and continuous feedback are the true driving force behind the unique analogies and streamlined structure of this book. This textbook was engineered for you.

Antigravity AI

About the Author

Milav Jayeshkumar Dabgar is an engineering educator and R&D professional with over 9 years of experience in electronics hardware development, embedded systems, AI/ML, and full-stack software engineering. He currently serves as a **Lecturer (Class - II) & Innovation Leader** at Government Polytechnic, Palanpur, under the Education Department of the Government of Gujarat.

Milav holds a Master of Engineering (ME) in Communication Systems from L.D. College of Engineering and a Bachelor of Engineering (BE) in Electronics & Communication. He is also currently pursuing a **BS in Data Science and Applications from IIT Madras**, where he has completed both the **Diploma in Programming** and the **Diploma in Data Science** with distinction.

Most recently, he completed the **AICTE QIP PG Certificate Program in Deep Learning from SVNIT Surat** with a **perfect 10/10 CGPA**, topping his batch.

A dedicated mentor, Milav has guided numerous student teams in securing over **INR 45 lakhs** in innovation funding, including INR 25 lakhs from Shark Tank India and INR 20 lakhs through iHub Gujarat, resulting in two student patents. He is recognized as an **NPTEL Evangelist** and **NPTEL Discipline Star** in Computer Science, reflecting his commitment to continuous learning and academic excellence.

His technical expertise spans from embedded systems (8051, STM32, Arduino) to modern web technologies (Next.js, Python, PostgreSQL) and Data Science (TensorFlow, PyTorch). Through this textbook, he aims to simplify complex Engineering Chemistry concepts by integrating relatable, real-world analogies drawn from modern tech and engineering landscapes.

Milav is also an active content creator, running a popular **YouTube channel** (@milav.dabgar) where he shares technical tutorials and academic resources. He maintains a personal blog and resource hub at milav.in and can be reached for professional collaborations on LinkedIn.

CHAPTER 1
MOS Transistor

1.1 Introduction

This chapter covers the computational aspects of the MOS (Metal-Oxide-Semiconductor) transistor, focusing on the quantitative analysis of its operation, specifically drain current characteristics, threshold voltage variations, device parameters such as transconductance, and intrinsic capacitances.

1.2 Drain Current in MOS Transistors

Methodology:

Drain Current Calculation The drain current I_D of an NMOS transistor depends on its region of operation, which is determined by the terminal voltages: Gate-to-Source voltage (V_{GS}) and Drain-to-Source voltage (V_{DS}). The constant $k'_n = \mu_n C_{ox}$ is the process transconductance parameter, and W/L is the aspect ratio.

1. **Cut-off Region:** When $V_{GS} < V_{TN}$ (where V_{TN} is the threshold voltage):

$$I_D = 0$$

2. **Linear (Triode) Region:** When $V_{GS} \geq V_{TN}$ and $V_{DS} < V_{GS} - V_{TN}$:

$$I_D = k'_n \frac{W}{L} \left[(V_{GS} - V_{TN}) V_{DS} - \frac{V_{DS}^2}{2} \right]$$

3. **Saturation Region:** When $V_{GS} \geq V_{TN}$ and $V_{DS} \geq V_{GS} - V_{TN}$:

$$I_D = \frac{1}{2} k'_n \frac{W}{L} (V_{GS} - V_{TN})^2 (1 + \lambda V_{DS})$$

where λ is the channel-length modulation parameter. (Assume $\lambda = 0$ unless specified).

For a PMOS transistor, use $|V_{GS}|$, $|V_{DS}|$, $|V_{TP}|$, and $k'_p = \mu_p C_{ox}$.

Worked Example:

Determine NMOS Drain Current An NMOS transistor has the following parameters: process transconductance parameter $k'_n = 120\ \mu\text{A/V}^2$, threshold voltage $V_{TN} = 0.8$ V, and aspect ratio $W/L = 10$. Calculate the drain current I_D for the following terminal voltages (assuming $\lambda = 0$):

1. $V_{GS} = 0.5$ V and $V_{DS} = 2.0$ V
2. $V_{GS} = 2.0$ V and $V_{DS} = 0.5$ V
3. $V_{GS} = 2.0$ V and $V_{DS} = 2.0$ V

Solution:

1. **Check region for $V_{GS} = 0.5$ V:**
   Since $V_{GS} = 0.5$ V $< V_{TN} = 0.8$ V, the transistor is in the cut-off region.

$$I_D = 0 \text{ A}$$

2. **Check region for $V_{GS} = 2.0$ V and $V_{DS} = 0.5$ V:**
   Overdrive voltage $V_{OV} = V_{GS} - V_{TN} = 2.0 - 0.8 = 1.2$ V.
   Since $V_{DS} = 0.5$ V < 1.2 V, the transistor is in the linear (triode) region.

$$\begin{aligned} I_D &= k_n' \frac{W}{L} \left[(V_{GS} - V_{TN}) V_{DS} - \frac{V_{DS}^2}{2} \right] \\ &= 120 \times 10^{-6} \times 10 \times \left[(1.2)(0.5) - \frac{0.5^2}{2} \right] \\ &= 1.2 \times 10^{-3} \times [0.6 - 0.125] \\ &= 1.2 \times 10^{-3} \times 0.475 \\ &= 0.57 \text{ mA} \end{aligned}$$

3. **Check region for $V_{GS} = 2.0$ V and $V_{DS} = 2.0$ V:**
   Overdrive voltage $V_{OV} = V_{GS} - V_{TN} = 2.0 - 0.8 = 1.2$ V.
   Since $V_{DS} = 2.0$ V ≥ 1.2 V, the transistor is in the saturation region.

$$\begin{aligned} I_D &= \frac{1}{2} k_n' \frac{W}{L} (V_{GS} - V_{TN})^2 \\ &= 0.5 \times 120 \times 10^{-6} \times 10 \times (1.2)^2 \\ &= 0.6 \times 10^{-3} \times 1.44 \\ &= 0.864 \text{ mA} \end{aligned}$$

Worked Example:

Aspect Ratio Calculation A PMOS transistor in a circuit must carry a saturation drain current of 1.5 mA with a gate-to-source voltage of -2.5 V. Given $V_{TP} = -0.7$ V and $k_p' = 50\ \mu\text{A/V}^2$. Neglect channel-length modulation. Find the required aspect ratio (W/L) for this device.
Solution:

1. **Identify given parameters:**
   $I_D = 1.5$ mA $= 1500\ \mu$A, $V_{GS} = -2.5$ V, $V_{TP} = -0.7$ V, $k_p' = 50\ \mu\text{A/V}^2$.

2. **Check saturation condition:**
   For PMOS saturation, $|V_{DS}| \geq |V_{GS}| - |V_{TP}|$. It is stated the device is in saturation.

3. **Apply the PMOS saturation current formula:**

$$I_D = \frac{1}{2} k_p' \frac{W}{L} (|V_{GS}| - |V_{TP}|)^2$$

4. **Substitute the values and solve for W/L:**

$$\begin{aligned}
1500 \times 10^{-6} &= \frac{1}{2} \times 50 \times 10^{-6} \times \frac{W}{L} \times (|-2.5| - |-0.7|)^2 \\
1500 &= 25 \times \frac{W}{L} \times (2.5 - 0.7)^2 \\
1500 &= 25 \times \frac{W}{L} \times (1.8)^2 \\
1500 &= 25 \times 3.24 \times \frac{W}{L} \\
1500 &= 81 \times \frac{W}{L} \\
\frac{W}{L} &= \frac{1500}{81} \approx 18.52
\end{aligned}$$

The required aspect ratio is approximately 18.52.

1.3 Threshold Voltage and Body Effect

Methodology:

Threshold Voltage Calculation The threshold voltage V_T is not strictly constant; it is affected by the voltage difference between the source and the body (substrate) terminals, V_{SB}. This phenomenon is known as the body effect.
The threshold voltage with body effect is given by:

$$V_T = V_{T0} + \gamma \left(\sqrt{|2\Phi_F| + V_{SB}} - \sqrt{|2\Phi_F|} \right)$$

Where:

1. V_{T0} is the zero-bias threshold voltage (when $V_{SB} = 0$).
2. γ is the body-effect parameter (in $\text{V}^{1/2}$).
3. $2\Phi_F$ is the surface potential parameter (typically around 0.6 V to 0.8 V).
4. V_{SB} is the Source-to-Body voltage.

Worked Example:

Calculate Threshold Voltage with Body Effect An NMOS transistor has a zero-bias threshold voltage $V_{T0} = 0.7$ V, body-effect parameter $\gamma = 0.4$ $\text{V}^{1/2}$, and $|2\Phi_F| = 0.6$ V. The source terminal is at 1.5 V and the body terminal is grounded (0 V). Calculate the new threshold voltage V_T.
Solution:

1. **Identify parameters:**
   $V_{T0} = 0.7$ V, $\gamma = 0.4$ $\text{V}^{1/2}$, $|2\Phi_F| = 0.6$ V.
2. **Calculate V_{SB}:**
   $V_{SB} = V_S - V_B = 1.5 \text{ V} - 0 \text{ V} = 1.5 \text{ V}$.

3. **Apply the body effect equation:**

$$\begin{aligned} V_T &= V_{T0} + \gamma \left(\sqrt{|2\Phi_F| + V_{SB}} - \sqrt{|2\Phi_F|} \right) \\ &= 0.7 + 0.4 \left(\sqrt{0.6 + 1.5} - \sqrt{0.6} \right) \\ &= 0.7 + 0.4 \left(\sqrt{2.1} - \sqrt{0.6} \right) \\ &= 0.7 + 0.4 \left(1.449 - 0.775 \right) \\ &= 0.7 + 0.4(0.674) \\ &= 0.7 + 0.270 \\ &= 0.97 \text{ V} \end{aligned}$$

The threshold voltage increases to 0.97 V due to the body effect.

1.4 Transconductance and Output Resistance

Methodology:

Small-Signal Parameters For analyzing circuits, the small-signal parameters are crucial. The main parameters in the saturation region are the transconductance (g_m) and the output resistance (r_o).

1. **Transconductance (g_m):** Represents how much the drain current changes for a small change in gate-source voltage.

$$g_m = \frac{\partial I_D}{\partial V_{GS}} = k'_n \frac{W}{L} (V_{GS} - V_{TN})$$

Alternatively, g_m can be expressed as:

$$g_m = \sqrt{2k'_n \frac{W}{L} I_D} = \frac{2I_D}{V_{GS} - V_{TN}}$$

2. **Output Resistance (r_o):** Represents the channel-length modulation effect.

$$r_o = \left(\frac{\partial I_D}{\partial V_{DS}} \right)^{-1} = \frac{1}{\lambda I_D}$$

where λ is the channel-length modulation parameter and I_D is the drain current without considering λ.

Worked Example:

Calculate Transconductance and Output Resistance Consider an NMOS transistor biased in the saturation region with $I_D = 2$ mA. The transistor has $k'_n = 100\ \mu\text{A/V}^2$, $W/L = 20$, and a channel-length modulation parameter $\lambda = 0.05\ \text{V}^{-1}$. Determine the transconductance g_m and output resistance r_o.
Solution:

1. **Calculate Transconductance (g_m):**

We can use the formula dependent on I_D:

$$\begin{aligned} g_m &= \sqrt{2k'_n \frac{W}{L} I_D} \\ &= \sqrt{2 \times (100 \times 10^{-6}) \times 20 \times (2 \times 10^{-3})} \\ &= \sqrt{4000 \times 10^{-6} \times 2 \times 10^{-3}} \\ &= \sqrt{8 \times 10^{-6}} \\ &= 2.828 \times 10^{-3} \text{ A/V} = 2.83 \text{ mA/V} \end{aligned}$$

2. **Calculate Output Resistance (r_o):**

$$\begin{aligned} r_o &= \frac{1}{\lambda I_D} \\ &= \frac{1}{0.05 \times 2 \times 10^{-3}} \\ &= \frac{1}{0.1 \times 10^{-3}} \\ &= 10000 \ \Omega = 10 \ \text{k}\Omega \end{aligned}$$

The transconductance is 2.83 mA/V and the output resistance is 10 kΩ.

1.5 MOS Capacitances

Methodology:

Gate Capacitance Calculation The total gate capacitance C_g of a MOS transistor is dynamically distributed depending on its region of operation. The fundamental parameter is the oxide capacitance per unit area, C_{ox}.

$$C_{ox} = \frac{\epsilon_{ox}}{t_{ox}}$$

where $\epsilon_{ox} \approx 3.9\epsilon_0$ is the permittivity of silicon dioxide ($\epsilon_0 = 8.85 \times 10^{-14}$ F/cm or 8.85×10^{-12} F/m), and t_{ox} is the oxide thickness.
The total gate oxide capacitance is $C_g = W \cdot L \cdot C_{ox}$. The distribution is as follows:

1. **Cut-off Region:** The channel is not formed. The capacitance is mainly between the gate and the bulk (substrate).
$$C_{gb} = C_{ox}WL, \quad C_{gs} = 0, \quad C_{gd} = 0$$
2. **Linear Region:** The channel spans from source to drain. The capacitance is divided equally between source and drain.
$$C_{gs} = \frac{1}{2}C_{ox}WL, \quad C_{gd} = \frac{1}{2}C_{ox}WL, \quad C_{gb} = 0$$
3. **Saturation Region:** The channel is pinched off near the drain. The capacitance shifts mostly towards the source.
$$C_{gs} = \frac{2}{3}C_{ox}WL, \quad C_{gd} = 0, \quad C_{gb} = 0$$

Note: Overlap capacitances (C_{gso}, C_{gdo}) are often added to these intrinsic values in practical problems.

Worked Example:

Determine Oxide Capacitance and Gate Capacitance An NMOS transistor has a gate width $W = 5\ \mu$m, gate length $L = 0.5\ \mu$m, and an oxide thickness $t_{ox} = 10$ nm. Determine the oxide capacitance per unit area C_{ox} and the gate-to-source capacitance C_{gs} when the device operates in the saturation region. (Given $\epsilon_{ox} = 3.45 \times 10^{-11}$ F/m)

Solution:

1. **Identify parameters:**
   $W = 5 \times 10^{-6}$ m, $L = 0.5 \times 10^{-6}$ m, $t_{ox} = 10 \times 10^{-9}$ m, $\epsilon_{ox} = 3.45 \times 10^{-11}$ F/m.

2. **Calculate Oxide Capacitance per unit area (C_{ox}):**

$$\begin{aligned} C_{ox} &= \frac{\epsilon_{ox}}{t_{ox}} \\ &= \frac{3.45 \times 10^{-11}}{10 \times 10^{-9}} \\ &= 3.45 \times 10^{-3}\ \text{F/m}^2 \end{aligned}$$

3. **Calculate Total Gate Oxide Capacitance (C_g):**

$$\begin{aligned} C_g &= C_{ox} \cdot W \cdot L \\ &= 3.45 \times 10^{-3} \times (5 \times 10^{-6}) \times (0.5 \times 10^{-6}) \\ &= 3.45 \times 10^{-3} \times 2.5 \times 10^{-12} \\ &= 8.625 \times 10^{-15}\ \text{F} = 8.625\ \text{fF} \end{aligned}$$

4. **Calculate C_{gs} in the Saturation Region:**
   In saturation, $C_{gs} = \frac{2}{3} C_g$.

$$\begin{aligned} C_{gs} &= \frac{2}{3} \times 8.625\ \text{fF} \\ &= 5.75\ \text{fF} \end{aligned}$$

The oxide capacitance per unit area is 3.45 mF/m^2 and the gate-to-source capacitance is 5.75 fF.

CHAPTER 2
MOS Inverters

The MOS inverter is the fundamental building block of digital VLSI circuits. Analyzing its characteristics involves determining its voltage transfer characteristic (VTC) parameters such as critical voltages and noise margins, as well as its propagation delay and power dissipation.

2.1 Inverter Switching Threshold Voltage

Methodology:

Calculating Switching Threshold Voltage The switching threshold voltage (often denoted as V_M or V_{th}) is the point where the input voltage equals the output voltage ($V_{in} = V_{out}$). At this point, both the NMOS and PMOS transistors operate in the saturation region.

1. **Identify the operating region:** Set $V_{in} = V_{out} = V_M$. Since $V_{DS} > V_{GS} - V_T$ for both transistors, both are in saturation.
2. **Equate drain currents:** The current through the PMOS equals the current through the NMOS: $I_{DS,n} = |I_{DS,p}|$.
3. **Apply saturation current formulas:**

$$\frac{\mu_n C_{ox}}{2}\left(\frac{W}{L}\right)_n (V_M - V_{tn})^2 = \frac{\mu_p C_{ox}}{2}\left(\frac{W}{L}\right)_p (V_{DD} - V_M - |V_{tp}|)^2$$

   Let $\beta_n = \mu_n C_{ox}(W/L)_n$ and $\beta_p = \mu_p C_{ox}(W/L)_p$.

4. **Solve for V_M:** Taking the square root of both sides gives:

$$\sqrt{\frac{\beta_n}{\beta_p}}(V_M - V_{tn}) = V_{DD} - V_M - |V_{tp}|$$

   Rearranging for V_M:

$$V_M = \frac{V_{tn} + \sqrt{\frac{\beta_p}{\beta_n}}(V_{DD} - |V_{tp}|)}{1 + \sqrt{\frac{\beta_p}{\beta_n}}}$$

Worked Example:

Calculating the Switching Threshold of a CMOS Inverter Consider a CMOS inverter with the following parameters: $V_{DD} = 5$ V, $V_{tn} = 1$ V, $V_{tp} = -1$ V, $\mu_n C_{ox} = 50\,\mu\text{A/V}^2$, $\mu_p C_{ox} = 20\,\mu\text{A/V}^2$. The aspect ratios are $(W/L)_n = 2$ and $(W/L)_p = 5$. Calculate the switching threshold voltage V_M.

Step 1: Calculate the transconductance parameters β_n and β_p.

$$\beta_n = \mu_n C_{ox}\left(\frac{W}{L}\right)_n = 50 \times 10^{-6} \times 2 = 100\,\mu\text{A/V}^2$$

$$\beta_p = \mu_p C_{ox} \left(\frac{W}{L}\right)_p = 20 \times 10^{-6} \times 5 = 100\,\mu\text{A/V}^2$$

Step 2: Determine the ratio β_p/β_n.

$$\frac{\beta_p}{\beta_n} = \frac{100}{100} = 1 \implies \sqrt{\frac{\beta_p}{\beta_n}} = 1$$

Step 3: Apply the switching threshold formula.

$$V_M = \frac{V_{tn} + \sqrt{\frac{\beta_p}{\beta_n}}(V_{DD} - |V_{tp}|)}{1 + \sqrt{\frac{\beta_p}{\beta_n}}}$$

Substitute the given values:

$$V_M = \frac{1 + 1 \times (5 - |-1|)}{1 + 1} = \frac{1 + (5 - 1)}{2} = \frac{1 + 4}{2} = \frac{5}{2} = 2.5 \text{ V}$$

Conclusion: The switching threshold voltage V_M is 2.5 V, which is exactly $V_{DD}/2$. This represents a perfectly symmetric inverter.

2.2 Noise Margins of CMOS Inverters

Methodology:

Calculating Critical Voltages and Noise Margins Noise margins represent the amount of noise a logic circuit can withstand without flipping its state. For a CMOS inverter, they are defined by the critical voltages V_{IL} (maximum logic '0' input), V_{IH} (minimum logic '1' input), V_{OL} (logic '0' output), and V_{OH} (logic '1' output).

1. **Identify Output Voltages:** For an ideal CMOS inverter with no DC load, $V_{OH} = V_{DD}$ and $V_{OL} = 0$ V.
2. **Determine V_{IL}:** This is the point where $\frac{dV_{out}}{dV_{in}} = -1$.
   - NMOS is in saturation: $I_n = \frac{\beta_n}{2}(V_{in} - V_{tn})^2$
   - PMOS is in the linear region.
   - Equate $I_n = I_p$ and differentiate with respect to V_{in} setting $\frac{dV_{out}}{dV_{in}} = -1$.
   - For a perfectly symmetric inverter ($\beta_n = \beta_p$ and $V_{tn} = |V_{tp}| = V_t$), V_{IL} mathematically simplifies to:
   $$V_{IL} = \frac{3V_{DD} + 2V_t}{8}$$
3. **Determine V_{IH}:** This is the other point where $\frac{dV_{out}}{dV_{in}} = -1$.
   - NMOS is in the linear region and PMOS is in saturation.
   - Equate currents and differentiate. For a perfectly symmetric inverter, V_{IH} simplifies to:
   $$V_{IH} = \frac{5V_{DD} - 2V_t}{8}$$
4. **Calculate Noise Margins:**
   $$NM_L = V_{IL} - V_{OL} = V_{IL}$$
   $$NM_H = V_{OH} - V_{IH} = V_{DD} - V_{IH}$$

Worked Example:

Calculating Noise Margins of a Symmetric CMOS Inverter A symmetric CMOS inverter operates at $V_{DD} = 5$ V with $V_{tn} = |V_{tp}| = 1$ V. The transconductance parameters are matched such that $\beta_n = \beta_p$. Calculate the critical voltages and the noise margins.

Step 1: Identify V_{OH} and V_{OL}. Since it is an ideal CMOS inverter without a resistive load, $V_{OH} = V_{DD} = 5$ V and $V_{OL} = 0$ V.

Step 2: Calculate V_{IL}. Since the inverter is symmetric, use the exact formula for V_{IL}:

$$V_{IL} = \frac{3V_{DD} + 2V_t}{8}$$

Substitute $V_{DD} = 5$ V and $V_t = 1$ V:

$$V_{IL} = \frac{3(5) + 2(1)}{8} = \frac{15 + 2}{8} = \frac{17}{8} = 2.125 \text{ V}$$

Step 3: Calculate V_{IH}. Using the symmetric formula for V_{IH}:

$$V_{IH} = \frac{5V_{DD} - 2V_t}{8}$$

Substitute the values:

$$V_{IH} = \frac{5(5) - 2(1)}{8} = \frac{25 - 2}{8} = \frac{23}{8} = 2.875 \text{ V}$$

Step 4: Compute Noise Margins. Low Noise Margin (NM_L):

$$NM_L = V_{IL} - V_{OL} = 2.125 - 0 = 2.125 \text{ V}$$

High Noise Margin (NM_H):

$$NM_H = V_{OH} - V_{IH} = 5 - 2.875 = 2.125 \text{ V}$$

Conclusion: The critical input voltages are $V_{IL} = 2.125$ V and $V_{IH} = 2.875$ V. The noise margins are symmetrically matched with $NM_L = NM_H = 2.125$ V.

2.3 Propagation Delay of CMOS Inverters

Methodology:

Calculating Propagation Delay Using RC Model Propagation delay determines the switching speed of a digital circuit. The RC delay model approximates the transistors as linear resistors when they are turned on charging or discharging a load capacitance C_{load}.

1. **Identify Load Capacitance (C_{load}):** Find the total capacitance connected to the inverter's output node.
2. **Identify Equivalent Resistances:** Determine the equivalent ON-resistance for the NMOS ($R_{eq,n}$) and PMOS ($R_{eq,p}$).
3. **Calculate Propagation Delays:**
   - **High-to-Low Delay (t_{PHL}):** The time taken for the output to fall to 50% of V_{DD}. The NMOS discharges the load.
   $$t_{PHL} = 0.69 \cdot R_{eq,n} \cdot C_{load}$$
   - **Low-to-High Delay (t_{PLH}):** The time taken for the output to rise to 50% of V_{DD}. The PMOS charges the load.
   $$t_{PLH} = 0.69 \cdot R_{eq,p} \cdot C_{load}$$

4. **Calculate Average Propagation Delay (t_p):**

$$t_p = \frac{t_{PHL} + t_{PLH}}{2}$$

Worked Example:

Computing the Propagation Delay of a CMOS Inverter A CMOS inverter is driving a load capacitance $C_{load} = 50$ fF. The equivalent ON-resistances are given as $R_{eq,n} = 10$ kΩ and $R_{eq,p} = 25$ kΩ. Calculate the high-to-low delay the low-to-high delay and the average propagation delay.

Step 1: Calculate High-to-Low Delay (t_{PHL}). This is the delay when the NMOS is discharging the output.

$$t_{PHL} = 0.69 \times R_{eq,n} \times C_{load}$$

Substitute the given values:

$$t_{PHL} = 0.69 \times (10 \times 10^3\,\Omega) \times (50 \times 10^{-15}\,\text{F})$$

$$t_{PHL} = 0.69 \times 500 \times 10^{-12}\ \text{s} = 345 \times 10^{-12}\ \text{s} = 345\ \text{ps}$$

Step 2: Calculate Low-to-High Delay (t_{PLH}). This is the delay when the PMOS is charging the output.

$$t_{PLH} = 0.69 \times R_{eq,p} \times C_{load}$$

Substitute the given values:

$$t_{PLH} = 0.69 \times (25 \times 10^3\,\Omega) \times (50 \times 10^{-15}\,\text{F})$$

$$t_{PLH} = 0.69 \times 1250 \times 10^{-12}\ \text{s} = 862.5 \times 10^{-12}\ \text{s} = 862.5\ \text{ps}$$

Step 3: Calculate Average Propagation Delay (t_p).

$$t_p = \frac{t_{PHL} + t_{PLH}}{2}$$

$$t_p = \frac{345\ \text{ps} + 862.5\ \text{ps}}{2} = \frac{1207.5\ \text{ps}}{2} = 603.75\ \text{ps}$$

Conclusion: The inverter has an average propagation delay of 603.75 ps. The higher equivalent resistance of the PMOS leads to a slower rising transition.

2.4 Power Dissipation in CMOS Inverters

Methodology:

Calculating CMOS Power Dissipation Components The total power dissipation in a CMOS inverter consists of dynamic power static leakage power and short-circuit power.

1. **Calculate Dynamic Power (P_{dyn}):** Dynamic power is consumed to charge and discharge the load capacitance during switching operations.

$$P_{dyn} = \alpha \cdot C_{load} \cdot V_{DD}^2 \cdot f$$

Where:

- α = Switching activity factor (probability of output transition per clock cycle).
- C_{load} = Total load capacitance.
- V_{DD} = Supply voltage.
- f = Operating frequency.

2. **Calculate Static Power (P_{stat}):** Static power is due to subthreshold leakage and reverse-biased diode leakage when the circuit is in a steady state.

$$P_{stat} = I_{leak} \cdot V_{DD}$$

Where I_{leak} is the total leakage current of the OFF transistor.

3. **Calculate Total Power (P_{total}):** Typically short-circuit power (P_{sc}) is approximated as negligible in standard diploma-level calculations unless specified. Thus:

$$P_{total} = P_{dyn} + P_{stat}$$

Worked Example:

Calculating Total Power Dissipation A CMOS inverter operates at a supply voltage of $V_{DD} = 3.3$ V and a switching frequency of $f = 100$ MHz. It drives a load capacitance of $C_{load} = 2$ pF. The activity factor is $\alpha = 0.5$. The combined leakage current of the circuit is $I_{leak} = 5$ nA. Calculate the dynamic power static power and total power dissipation.

Step 1: Calculate Dynamic Power (P_{dyn}).

$$P_{dyn} = \alpha C_{load} V_{DD}^2 f$$

Substitute the given values:

$$P_{dyn} = 0.5 \times (2 \times 10^{-12} \text{ F}) \times (3.3 \text{ V})^2 \times (100 \times 10^6 \text{ Hz})$$

$$P_{dyn} = 10^{-12} \times 10.89 \times 10^8 = 10.89 \times 10^{-4} \text{ W} = 1.089 \text{ mW}$$

Step 2: Calculate Static Power (P_{stat}).

$$P_{stat} = I_{leak} V_{DD}$$

Substitute the values:

$$P_{stat} = (5 \times 10^{-9} \text{ A}) \times 3.3 \text{ V} = 16.5 \times 10^{-9} \text{ W} = 16.5 \text{ nW}$$

Step 3: Calculate Total Power (P_{total}).

$$P_{total} = P_{dyn} + P_{stat}$$

$$P_{total} = 1.089 \text{ mW} + 16.5 \text{ nW}$$

Converting static power to mW (16.5 nW = 0.0000165 mW):

$$P_{total} \approx 1.0890165 \text{ mW} \approx 1.089 \text{ mW}$$

Conclusion: The total power dissipation is overwhelmingly dominated by dynamic power resulting in approximately 1.089 mW. The static power is extremely small and negligible in this active scenario.

2.5 Depletion Load NMOS Inverter Characteristics

Methodology:

Calculating Output Voltages of a Depletion Load NMOS Inverter An NMOS inverter with a depletion-mode NMOS transistor as the pull-up load uses $V_{GS,load} = 0$. The threshold voltage of the depletion load is negative ($V_{t,dep} < 0$). This methodology is used to determine the logic low output level V_{OL}.

1. **Identify High Output Voltage (V_{OH}):** When the input is logic '0', the driver NMOS is OFF. The

depletion load acts as a pull-up and pulls the output completely to V_{DD}.

$$V_{OH} = V_{DD}$$

2. **Setup the Current Equation for Low Output Voltage (V_{OL}):** When the input is logic '1' ($V_{in} = V_{DD}$), the driver NMOS operates in the linear region and the depletion load is in saturation.
   - Driver current (Linear): $I_{driver} = \frac{\beta_{driver}}{2}[2(V_{DD} - V_{t,driver})V_{OL} - V_{OL}^2]$
   - Load current (Saturation): $I_{load} = \frac{\beta_{load}}{2}(0 - V_{t,dep})^2 = \frac{\beta_{load}}{2}|V_{t,dep}|^2$
3. **Equate Currents and Solve:** Set $I_{driver} = I_{load}$:

$$\frac{\beta_{driver}}{2}[2(V_{DD} - V_{t,driver})V_{OL} - V_{OL}^2] = \frac{\beta_{load}}{2}|V_{t,dep}|^2$$

   Defining the aspect ratio $k = \beta_{driver}/\beta_{load}$:

$$k[2(V_{DD} - V_{t,driver})V_{OL} - V_{OL}^2] = |V_{t,dep}|^2$$

4. **Apply Approximation:** For a well-designed inverter, V_{OL} is very small, making $V_{OL}^2 \approx 0$. The simplified formula is:

$$V_{OL} \approx \frac{|V_{t,dep}|^2}{2k(V_{DD} - V_{t,driver})}$$

Worked Example:

Calculating Low Output Voltage of Depletion Load Inverter Consider a depletion load NMOS inverter with $V_{DD} = 5$ V, $V_{t,driver} = 1$ V and $V_{t,dep} = -2$ V. The driver-to-load transconductance ratio is $k = \beta_{driver}/\beta_{load} = 4$. Calculate the exact low output voltage V_{OL} and verify with the approximation formula.
Step 1: Set up the current equation. The equation for V_{OL} is:

$$k[2(V_{DD} - V_{t,driver})V_{OL} - V_{OL}^2] = |V_{t,dep}|^2$$

Step 2: Substitute the known values.

$$4 \times [2(5-1)V_{OL} - V_{OL}^2] = |-2|^2$$

$$4 \times [8V_{OL} - V_{OL}^2] = 4$$

Divide both sides by 4:

$$8V_{OL} - V_{OL}^2 = 1$$

$$V_{OL}^2 - 8V_{OL} + 1 = 0$$

Step 3: Solve the quadratic equation for V_{OL}. Using the quadratic formula $V_{OL} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$:

$$V_{OL} = \frac{8 \pm \sqrt{64 - 4(1)(1)}}{2} = \frac{8 \pm \sqrt{60}}{2} = \frac{8 \pm 7.746}{2}$$

Since V_{OL} must be the smaller valid root (close to 0 V):

$$V_{OL} = \frac{8 - 7.746}{2} = \frac{0.254}{2} = 0.127 \text{ V}$$

Step 4: Check using the approximation formula.

$$V_{OL} \approx \frac{|V_{t,dep}|^2}{2k(V_{DD} - V_{t,driver})} = \frac{4}{2(4)(4)} = \frac{4}{32} = 0.125 \text{ V}$$

Conclusion: The exact low output voltage is 0.127 V. The simplified approximation gives 0.125 V which is highly accurate and practical for rapid analysis.

CHAPTER 3
MOS Circuits

3.1 CMOS Logic Circuits

CMOS (Complementary Metal-Oxide-Semiconductor) logic uses both NMOS and PMOS transistors in a complementary manner. The standard structure consists of a Pull-Up Network (PUN) made of PMOS transistors connected to the supply voltage (V_{DD}) and a Pull-Down Network (PDN) made of NMOS transistors connected to ground.

Methodology:

Designing CMOS Complex Logic Circuits To implement a given Boolean function $Y = f(A, B, C \dots)$ using standard CMOS logic:

1. **Invert the function**: Express the function in inverted form $Y = \overline{f_{PDN}(A, B, C \dots)}$ because a standard CMOS gate acts as an inverter. If the required function is not inherently inverted (like AND/OR), you must design for $\overline{Y}$ and add an inverter at the output.
2. **Design Pull-Down Network**: Use the uncomplemented expression inside the inversion to build the PDN with NMOS transistors.
   - For variables ANDed together (e.g. $A \cdot B$), connect their NMOS transistors in **series**.
   - For variables ORed together (e.g. $A + B$), connect their NMOS transistors in **parallel**.
3. **Design Pull-Up Network**: Construct the PUN with PMOS transistors using the dual of the PDN.
   - For variables ANDed together, connect PMOS transistors in **parallel**.
   - For variables ORed together, connect PMOS transistors in **series**.
4. **Connect the Networks**: Connect the PUN between V_{DD} and the output node Y. Connect the PDN between the output node Y and ground.

Worked Example:

Implementing a Complex CMOS Logic Gate Implement the Boolean function $Y = \overline{A \cdot (B + C) + D}$ using CMOS logic.
Step 1: Analyze the function The function is already in the inverted form. The logic for the PDN is given by the expression under the bar: $F_{PDN} = A \cdot (B + C) + D$.
Step 2: Design the Pull-Down Network

- B and C are ORed, so NMOS transistors M_{NB} and M_{NC} are in parallel.
- A is ANDed with $(B + C)$, so NMOS transistor M_{NA} is in series with the parallel combination of M_{NB} and M_{NC}.
- D is ORed with the entire expression $A \cdot (B + C)$, so NMOS transistor M_{ND} is in parallel with the entire branch containing M_{NA}, M_{NB} and M_{NC}.

Step 3: Design the Pull-Up Network The PUN is the dual of the PDN.

- B and C are in parallel in the PDN, so PMOS transistors M_{PB} and M_{PC} must be in series in the PUN.
- A is in series with the $(B + C)$ block in the PDN, so PMOS transistor M_{PA} must be in parallel with the series combination of M_{PB} and M_{PC}.
- D is in parallel with the entire block in the PDN, so PMOS transistor M_{PD} must be in series with the entire PUN block constructed so far.

Step 4: Connect the Networks

- Connect the source of M_{PD} to V_{DD}.
- The drain of M_{PD} connects to the parallel block of M_{PA} and the series pair $M_{PB} - M_{PC}$.
- The bottom of this PUN block connects to the output node Y.
- For the PDN, connect the top of the parallel branches (M_{ND} and M_{NA}) to the output node Y.
- Connect the bottom of the PDN branches to Ground.

3.2 Transistor Sizing in CMOS Logic

Methodology:

Transistor Sizing for Symmetrical CMOS Gates To ensure that a complex CMOS gate has a worst-case delay equal to that of a standard reference inverter (where NMOS size $= W$ and PMOS size $= 2W$):

1. **Analyze the PDN for worst-case pull-down**: Find the longest path from output to ground. If the longest path has N NMOS transistors in series, size each NMOS transistor in that path to $N \cdot W$. Transistors in parallel paths only conduct one at a time in the worst case, so their sizes depend on the series depth of their respective paths.
2. **Analyze the PUN for worst-case pull-up**: Find the longest path from V_{DD} to output. If the longest path has P PMOS transistors in series, size each PMOS transistor in that path to $P \cdot 2W$.
3. **Assign sizes**: Apply the calculated widths to all transistors.

Worked Example:

Sizing a 3-Input NOR Gate Determine the transistor sizes for a 3-input NOR gate ($Y = \overline{A + B + C}$) to achieve the same worst-case drive strength as a reference inverter with $W_n = W$ and $W_p = 2W$.
Step 1: Determine the logic structure

- **PDN**: A, B and C are ORed. The PDN consists of 3 NMOS transistors in parallel.
- **PUN**: By duality, the PUN consists of 3 PMOS transistors in series.

Step 2: Size the PDN (NMOS) The worst-case pull-down scenario occurs when only one of the inputs is HIGH. In this case, there is only 1 NMOS transistor discharging the output. To match the resistance of a single NMOS in the reference inverter, we require:

$$W_{N,A} = W_{N,B} = W_{N,C} = 1 \cdot W = W$$

Step 3: Size the PUN (PMOS) The worst-case pull-up scenario (the only pull-up scenario for a NOR gate) requires all three PMOS transistors to be turned ON. They are connected in series. To match the resistance of a single PMOS ($2W$) in the reference inverter, each of the 3 series PMOS transistors must be

three times as wide:

$$W_{P,A} = W_{P,B} = W_{P,C} = 3 \cdot 2W = 6W$$

Conclusion: Each NMOS transistor is sized at W and each PMOS transistor is sized at $6W$.

3.3 Transmission Gate Logic

A CMOS Transmission Gate (TG) acts as a bidirectional switch consisting of an NMOS and PMOS transistor connected in parallel. It passes both logic 0 and logic 1 without threshold voltage drops.

Methodology:

Implementing Logic with Transmission Gates To design a logic function using transmission gates:

1. **Select a control variable**: Choose one (or more) input variables to act as the select signal for the transmission gates. Let this be C.
2. **Construct a modified truth table**: Express the output Y as a function of the remaining variables for each state of the control variable C.
3. **Assign input signals**:
   - If $C = 0 \implies Y = X_0$, connect the input X_0 to a TG controlled by $\overline{C}$.
   - If $C = 1 \implies Y = X_1$, connect the input X_1 to a TG controlled by C.
4. **Combine outputs**: Tie the outputs of all TGs together to form the final output Y.

Worked Example:

Implementing a 2-to-1 Multiplexer Design a 2-to-1 multiplexer using transmission gates. The multiplexer has data inputs I_0 and I_1, a select input S and output Y.
Step 1: Select the control variable The select input S naturally acts as the control variable.
Step 2: Determine output for each state

- When $S = 0$, $Y = I_0$.
- When $S = 1$, $Y = I_1$.

Step 3: Assign inputs to TGs

- Use TG1 for the $S = 0$ path. Connect I_0 to the input of TG1. The NMOS gate of TG1 is connected to $\overline{S}$ and the PMOS gate is connected to S.
- Use TG2 for the $S = 1$ path. Connect I_1 to the input of TG2. The NMOS gate of TG2 is connected to S and the PMOS gate is connected to $\overline{S}$.

Step 4: Combine outputs Tie the output terminals of TG1 and TG2 together to form node Y.

3.4 Sequential MOS Circuits

Methodology:

Designing a CMOS D-Latch A positive-level sensitive D-latch passes the input D to the output Q when the clock (CLK) is HIGH and holds the last value when CLK is LOW.

1. **Input Transmission Gate**: Connect input D to a transmission gate (TG1) that is turned ON when $CLK = 1$.

2. **Forward Path**: Connect the output of TG1 to an inverter (INV1). The output of INV1 is $\overline{Q}$. Connect $\overline{Q}$ to another inverter (INV2) to produce the true output Q.
3. **Feedback Loop**: Connect the output of INV2 back to the input of INV1 through a second transmission gate (TG2).
4. **Feedback Control**: TG2 must be turned ON when $CLK = 0$. This ensures that when the input is disconnected, the cross-coupled inverters maintain the state.

Worked Example:

Master-Slave Edge-Triggered D Flip-Flop Design a positive edge-triggered D flip-flop using D-latches and explain its operation.
Step 1: Structure Cascade two D-latches. The first is the **Master** latch and the second is the **Slave** latch. Connect the output of the Master to the input of the Slave.
Step 2: Clocking

- Drive the Master latch with an inverted clock ($\overline{CLK}$). It becomes active (transparent) when $CLK = 0$.
- Drive the Slave latch with the true clock (CLK). It becomes active (transparent) when $CLK = 1$.

Step 3: Operation Analysis ($CLK = 0$)

- When $CLK = 0$, the Master latch is transparent. It samples the input D and passes it to its intermediate output Q_M.
- The Slave latch is opaque (holding state). The final output Q remains at its previous value.

Step 4: Operation Analysis (Rising Edge $CLK 0 \to 1$)

- At the rising edge, the Master latch becomes opaque. It locks in the value of D present just before the clock edge.
- The Slave latch becomes transparent. It takes the locked value Q_M from the Master and passes it to the final output Q.
- Any subsequent changes to D while $CLK = 1$ are ignored because the Master latch is closed.

CHAPTER 4

Fabrication of MOSFET

The fabrication of MOSFETs involves a sequence of chemical and physical processes to create integrated circuits on a semiconductor wafer. While much of fabrication is process-oriented, several key computational techniques are essential for determining material properties, layout dimensions, and electrical characteristics.

4.1 Oxidation and Oxide Capacitance

Methodology:

Oxide Capacitance Calculation The gate oxide layer acts as a dielectric between the gate electrode and the semiconductor substrate. Its capacitance per unit area is a critical parameter for MOSFET operation.

1. Identify the thickness of the oxide layer (t_{ox}) in centimeters (cm) or meters (m).
2. Determine the permittivity of silicon dioxide (ϵ_{ox}). Typically, $\epsilon_{ox} = \epsilon_r \epsilon_0$, where $\epsilon_r \approx 3.9$ and $\epsilon_0 = 8.854 \times 10^{-14}$ F/cm.
3. Calculate the oxide capacitance per unit area (C_{ox}) using the formula:

$$C_{ox} = \frac{\epsilon_{ox}}{t_{ox}}$$

4. To find the total gate capacitance (C_g), multiply C_{ox} by the gate area ($A = W \times L$):

$$C_g = C_{ox} \times W \times L$$

Worked Example:

Calculating Gate Oxide Capacitance A MOSFET has a gate oxide thickness of 10 nm. The channel length is 1 μm and the channel width is 5 μm. Calculate the oxide capacitance per unit area and the total gate capacitance. Assume $\epsilon_{ox} = 3.45 \times 10^{-13}$ F/cm.
Solution:

1. **Given Data:** $t_{ox} = 10$ nm $= 10 \times 10^{-7}$ cm
   $L = 1\mu\text{m} = 1 \times 10^{-4}$ cm
   $W = 5\mu\text{m} = 5 \times 10^{-4}$ cm
   $\epsilon_{ox} = 3.45 \times 10^{-13}$ F/cm
2. **Calculate Oxide Capacitance per Unit Area:**

$$C_{ox} = \frac{\epsilon_{ox}}{t_{ox}} = \frac{3.45 \times 10^{-13}}{10 \times 10^{-7}} = 3.45 \times 10^{-7} \text{ F/cm}^2$$

3. **Calculate Total Gate Area:**

$$A = W \times L = (5 \times 10^{-4}) \times (1 \times 10^{-4}) = 5 \times 10^{-8} \text{ cm}^2$$

4. **Calculate Total Gate Capacitance:**

$$C_g = C_{ox} \times A = (3.45 \times 10^{-7}) \times (5 \times 10^{-8}) = 1.725 \times 10^{-14} \text{ F} = 17.25 \text{ fF}$$

4.2 Sheet Resistance in Interconnects

Methodology:

Sheet Resistance and Line Resistance During fabrication, layers of polysilicon, metal, and diffused regions are created. The resistance of these layers is calculated using the concept of sheet resistance.

1. Determine the sheet resistance (R_s) of the material layer, usually provided in ohms per square ($\Omega/\square$). It is defined as $R_s = \frac{\rho}{t}$, where ρ is resistivity and t is thickness.
2. Identify the length (L) and width (W) of the conductive track.
3. Calculate the number of squares (n) in the track:

$$n = \frac{L}{W}$$

4. Calculate the total resistance (R) of the line:

$$R = R_s \times n = R_s \times \frac{L}{W}$$

Worked Example:

Polysilicon Line Resistance A polysilicon interconnect line has a sheet resistance of 20 $\Omega/\square$. The line needs to connect two components separated by a distance of 100 μm. If the minimum width of the polysilicon line is 2 μm, determine the total resistance of the interconnect.
Solution:

1. **Given Data:** $R_s = 20\ \Omega/\square$
   $L = 100\ \mu\text{m}$
   $W = 2\ \mu\text{m}$
2. **Calculate the Number of Squares:**

$$n = \frac{L}{W} = \frac{100}{2} = 50 \text{ squares}$$

3. **Calculate Total Resistance:**

$$R = R_s \times n = 20 \times 50 = 1000\ \Omega = 1\ \text{k}\Omega$$

4.3 Lambda Based Design Rules

Methodology:

Area Estimation using Lambda Rules Mead-Conway λ-based design rules express all layout dimensions in terms of a single scalable parameter, λ, which is typically half the minimum feature size of the fabrication process.

1. Identify the value of λ for the specific process technology.
2. Apply standard λ rules for dimensions (e.g. Poly width = 2λ, Active area width = 3λ, Contact hole

$= 2\lambda \times 2\lambda$).

3. Calculate dimensions in terms of λ. For example, an active region with one contact and a gate might have a length of L_{active} = spacing + contact + spacing + gate $= 2\lambda + 2\lambda + 2\lambda + 2\lambda = 8\lambda$.
4. Convert the λ-based area ($A = W_\lambda \times L_\lambda$) into physical units (like μm^2) by multiplying by λ^2.

Worked Example:

Estimating Minimum Transistor Area Consider a fabrication process where the minimum feature size is 0.5 μm. Using simplified λ-based design rules, estimate the active area required for a minimum-sized nMOS transistor. The transistor active area must accommodate the gate and two contact regions (source and drain). Assume the active region width is 3λ, contact size is $2\lambda \times 2\lambda$, poly-to-contact spacing is 2λ, active overlap of contact is 1λ, and gate length is 2λ.
Solution:

1. **Determine Lambda:** $\lambda = \frac{\text{Minimum Feature Size}}{2} = \frac{0.5\ \mu m}{2} = 0.25\ \mu m$.
2. **Determine Active Region Width (W):** Minimum active width $W = 3\lambda$.
3. **Determine Active Region Length (L):** The length includes the source contact region, the gate, and the drain contact region. Source/Drain region length = Active overlap (1λ) + Contact (2λ) + Poly-to-contact spacing (2λ) = 5λ. Total Length L = Source length + Gate length + Drain length = $5\lambda + 2\lambda + 5\lambda = 12\lambda$.
4. **Calculate Area in terms of Lambda:** Area $A = W \times L = 3\lambda \times 12\lambda = 36\lambda^2$.
5. **Calculate Physical Area:** $A = 36 \times (0.25\ \mu m)^2 = 36 \times 0.0625\ \mu m^2 = 2.25\ \mu m^2$.

4.4 Threshold Voltage Shift due to Ion Implantation

Methodology:

Threshold Voltage Adjustment Ion implantation is used during fabrication to adjust the threshold voltage (V_T) of a MOSFET by modifying the substrate doping concentration near the surface.

1. Determine the initial threshold voltage (V_{T0}).
2. Identify the implant dose (D_I), which is the number of implanted ions per unit area (cm^{-2}).
3. Note the elementary charge $q = 1.602 \times 10^{-19}$ C.
4. Use the gate oxide capacitance per unit area (C_{ox} in F/cm^2).
5. Calculate the change in threshold voltage (ΔV_T):

$$\Delta V_T = \frac{q \cdot D_I}{C_{ox}}$$

(Note: Add ΔV_T for p-type implants in n-channel devices, and subtract for n-type implants).

6. Calculate the final adjusted threshold voltage:

$$V_{T(final)} = V_{T0} \pm \Delta V_T$$

Worked Example:

Calculating Implant Dose for Threshold Voltage Adjustment An n-channel MOSFET has an initial threshold voltage of 0.4 V. It is required to adjust the threshold voltage to 0.8 V using a Boron (p-type) implant. The gate oxide capacitance per unit area is 3.45×10^{-7} F/cm^2. Determine the required implant dose.
Solution:

1. **Given Data:** $V_{T0} = 0.4$ V
   $V_{T(final)} = 0.8$ V
   $C_{ox} = 3.45 \times 10^{-7}$ F/cm^2
   $q = 1.602 \times 10^{-19}$ C

2. **Calculate Required Threshold Voltage Shift:**

$$\Delta V_T = V_{T(final)} - V_{T0} = 0.8 - 0.4 = 0.4 \text{ V}$$

3. **Calculate Implant Dose (D_I):** Rearranging the formula $\Delta V_T = \frac{q \cdot D_I}{C_{ox}}$ gives:

$$D_I = \frac{\Delta V_T \cdot C_{ox}}{q}$$

$$D_I = \frac{0.4 \times (3.45 \times 10^{-7})}{1.602 \times 10^{-19}}$$

$$D_I = \frac{1.38 \times 10^{-7}}{1.602 \times 10^{-19}} \approx 8.61 \times 10^{11} \text{ ions/cm}^2$$

   The required boron implant dose is approximately 8.61×10^{11} cm^{-2}.

4.5 Yield Calculation in Wafer Fabrication

Methodology:

Die Yield Estimation The number of functional dies on a silicon wafer depends on the defect density during the fabrication process. The Poisson yield model is commonly used for estimation.

1. Determine the wafer area ($A_{wafer} = \pi \times r^2$).
2. Identify the area of a single die (A_{die}).
3. Calculate the total number of gross dies on the wafer (ignoring edge losses for simplicity):

$$N = \frac{A_{wafer}}{A_{die}}$$

4. Identify the defect density (D_0) in defects per unit area.
5. Calculate the yield (Y) using the Poisson model:

$$Y = e^{-A_{die} \times D_0}$$

6. Calculate the number of good (functional) dies:

$$N_{good} = N \times Y$$

Worked Example:

Estimating Good Dies on a Wafer A fabrication facility produces chips on a 200 mm diameter silicon wafer. Each die has an area of 1 cm^2. The fabrication process has a defect density of 0.5 defects/cm^2. Estimate the total number of dies per wafer and the number of functional dies.

Solution:

1. **Given Data:** Wafer diameter = 200 mm $\implies$ radius $r = 100$ mm $= 10$ cm
   $A_{die} = 1$ cm^2
   $D_0 = 0.5$ defects/cm^2

2. **Calculate Wafer Area:**

$$A_{wafer} = \pi \times r^2 = \pi \times (10)^2 = 100\pi \approx 314.16 \text{ cm}^2$$

3. **Calculate Gross Dies (N):**

$$N = \frac{A_{wafer}}{A_{die}} = \frac{314.16}{1} = 314 \text{ dies}$$

   (Note: This is an upper limit; rigorous calculations account for edge dies.)

4. **Calculate Yield (Y):**

$$Y = e^{-A_{die} \times D_0} = e^{-1 \times 0.5} = e^{-0.5} \approx 0.6065 \text{ (or } 60.65\%)$$

5. **Calculate Number of Good Dies:**

$$N_{good} = N \times Y = 314 \times 0.6065 \approx 190 \text{ dies}$$

CHAPTER 5
Introduction to VERILOG

5.1 Gate-Level Modeling

Methodology:

Gate-Level Modeling Procedure Gate-level modeling (Structural modeling) describes a digital circuit in terms of logic gates and their interconnections.

1. Define the module with its name and port list (inputs and outputs).
2. Declare the `input` and `output` ports.
3. Declare internal connections using the `wire` data type if necessary.
4. Instantiate primitive logic gates (`and`, `or`, `not`, `nand`, `nor`, `xor`, `xnor`). The general syntax is: `gatename instance_name (output_port, input1, input2, ...);`
5. Close the module using `endmodule`.

Worked Example:

Gate-Level Model of a Half Adder Write a Verilog module for a Half Adder using gate-level modeling.
Solution:

1. **Identify Inputs and Outputs:** A Half Adder has two inputs (A and B) and two outputs (Sum and $Carry$).
2. **Logic Equations:**
   - $Sum = A \oplus B$ (XOR gate)
   - $Carry = A \cdot B$ (AND gate)
3. **Write the Verilog Code:**

```
module half_adder_gate(
    input A,
    input B,
    output Sum,
    output Carry
);
    // Instantiate XOR gate for Sum
    xor u1 (Sum, A, B);

    // Instantiate AND gate for Carry
    and u2 (Carry, A, B);

endmodule
```

5.2 Dataflow Modeling

Methodology:

Dataflow Modeling Procedure Dataflow modeling uses continuous assignment statements to describe the flow of data from inputs to outputs based on Boolean equations.

1. Define the module and declare `input` and `output` ports.
2. Declare any intermediate signals as `wire`.
3. Use the `assign` statement with logical operators (`&` for AND, `|` for OR, `^` for XOR, `~` for NOT) or conditional operators (`?  :`) to drive outputs.
4. Format: `assign output = boolean_expression;`
5. Close with `endmodule`.

Worked Example:

Dataflow Model of a Full Adder Write a Verilog module for a Full Adder using dataflow modeling.
Solution:

1. **Identify Inputs and Outputs:** Inputs are A, B and Cin. Outputs are Sum and $Cout$.
2. **Logic Equations:**
   - $Sum = A \oplus B \oplus Cin$
   - $Cout = (A \cdot B) + (B \cdot Cin) + (A \cdot Cin)$
3. **Write the Verilog Code:**

```
module full_adder_dataflow(
    input A,
    input B,
    input Cin,
    output Sum,
    output Cout
);
    // Continuous assignments for Sum and Cout
    assign Sum = A ^ B ^ Cin;
    assign Cout = (A & B) | (B & Cin) | (A & Cin);

endmodule
```

Worked Example:

Dataflow Model of a 4-to-1 Multiplexer Write a Verilog code for a 4:1 Multiplexer using the conditional operator in dataflow modeling.
Solution:

1. **Identify Ports:** Four data inputs ($D0$ to $D3$), a 2-bit select line (S) and one output (Y).
2. **Logic Formulation:** Use the ternary conditional operator `(condition ? true_value : false_value)`.
3. **Write the Verilog Code:**

```
module mux4to1_dataflow(
```

```
    input [3:0] D,
    input [1:0] S,
    output Y
);
    // Nested conditional operators
    assign Y = (S == 2'b00) ? D[0] :
               (S == 2'b01) ? D[1] :
               (S == 2'b10) ? D[2] : D[3];

endmodule
```

5.3 Behavioral Modeling

Methodology:

Behavioral Modeling Procedure Behavioral modeling describes the functionality of a circuit using procedural blocks (`always` or `initial`) similar to high-level programming languages.

1. Define the module and declare ports.
2. Outputs that are assigned within an `always` block MUST be declared as `reg` data type.
3. Create an `always` block with a sensitivity list. For combinational logic, use `always @(*)`. For sequential logic, use edge-triggered events like `always @(posedge clk)`.
4. Use procedural statements (`if-else` or `case`) inside the block.
5. Assign values using blocking (`=`) for combinational logic or non-blocking (`<=`) for sequential logic.
6. Close with `endmodule`.

Worked Example:

Behavioral Model of an 8-to-1 Multiplexer Write a Verilog code for an 8:1 Multiplexer using a case statement in behavioral modeling.
Solution:

1. **Identify Ports:** 8-bit input D, 3-bit select S and output Y.
2. **Data Types:** Y must be a `reg` because it is assigned in an `always` block.
3. **Write the Verilog Code:**

```
module mux8to1_behavioral(
    input [7:0] D,
    input [2:0] S,
    output reg Y
);
    always @(*) begin
        case (S)
            3'b000: Y = D[0];
            3'b001: Y = D[1];
            3'b010: Y = D[2];
            3'b011: Y = D[3];
            3'b100: Y = D[4];
            3'b101: Y = D[5];
            3'b110: Y = D[6];
```

```
            3'b111: Y = D[7];
            default: Y = 1'b0;
        endcase
    end
endmodule
```

Worked Example:

Behavioral Model of a D Flip-Flop Write a Verilog module for a positive edge-triggered D Flip-Flop with an asynchronous active-low reset.
Solution:

1. **Identify Ports:** Inputs are Data (D), Clock (clk) and Reset (rst_n). Output is Q.
2. **Sensitivity List:** The block must trigger on the positive edge of clk or the negative edge of rst_n.
3. **Assignment:** Use non-blocking assignments (`<=`) for sequential logic.
4. **Write the Verilog Code:**

```
module d_flip_flop(
    input D,
    input clk,
    input rst_n,
    output reg Q
);
    // Trigger on positive clock edge or negative reset edge
    always @(posedge clk or negedge rst_n) begin
        if (!rst_n) begin
            Q <= 1'b0; // Asynchronous reset
        end else begin
            Q <= D;    // Normal operation
        end
    end
endmodule
```

5.4 Testbench Creation

Methodology:

Testbench Writing Procedure A testbench is a Verilog module used to simulate and verify the functionality of a design block (Device Under Test or DUT).

1. Create a module with no ports.
2. Declare signals to connect to the DUT: use `reg` for inputs to the DUT (to drive them in `initial` blocks) and `wire` for outputs from the DUT.
3. Instantiate the DUT and map the testbench signals to the DUT ports.
4. Use an `initial` block to apply stimulus (test vectors) to the inputs, using delays (e.g. `#10`) to advance simulation time.
5. Optionally, use `$monitor` or `$display` to print output values during the simulation.

Worked Example:

Testbench for a Half Adder Write a Verilog testbench to verify the functionality of the Half Adder designed previously.

Solution:

1. **Signal Declaration:** Define `reg` for A and B, and `wire` for Sum and $Carry$.
2. **DUT Instantiation:** Connect these signals to the `half_adder_gate` module.
3. **Stimulus Generation:** Apply all four possible combinations of A and B (00, 01, 10, 11) with delays.
4. **Write the Verilog Code:**

```
module tb_half_adder;
    // 1. Declare signals
    reg A, B;
    wire Sum, Carry;

    // 2. Instantiate DUT
    half_adder_gate dut (
        .A(A),
        .B(B),
        .Sum(Sum),
        .Carry(Carry)
    );

    // 3. Apply Stimulus
    initial begin
        // Print header and monitor signal changes
        $monitor("Time=%0d A=%b B=%b Sum=%b Carry=%b",
                 $time, A, B, Sum, Carry);

        // Test vectors
        A = 0; B = 0; #10;
        A = 0; B = 1; #10;
        A = 1; B = 0; #10;
        A = 1; B = 1; #10;

        $finish; // End simulation
    end
endmodule
```

Formulae

Appendix: Key Formulae

This appendix provides a quick reference to the key formulae and mathematical relationships discussed in this book.

Unit 1: MOS Transistor Theory

MOSFET Capacitances

- **Gate Oxide Capacitance per Unit Area:**

$$C_{ox} = \frac{\epsilon_{ox}}{t_{ox}}$$

- **Cutoff Region:**

$$C_{gb} = C_{ox}WL, \quad C_{gs} = 0, \quad C_{gd} = 0$$

- **Linear (Triode) Region:**

$$C_{gs} = \frac{1}{2}C_{ox}WL, \quad C_{gd} = \frac{1}{2}C_{ox}WL, \quad C_{gb} = 0$$

- **Saturation Region:**

$$C_{gs} = \frac{2}{3}C_{ox}WL, \quad C_{gd} = 0, \quad C_{gb} = 0$$

Drain Current (I_D)

- **Linear Region (NMOS):**

$$I_D = k'_n \frac{W}{L}\left[(V_{GS} - V_{TN})V_{DS} - \frac{V_{DS}^2}{2}\right]$$

- **Saturation Region (NMOS):**

$$I_D = \frac{1}{2}k'_n \frac{W}{L}(V_{GS} - V_{TN})^2(1 + \lambda V_{DS})$$

- **Saturation Region (PMOS):**

$$I_D = \frac{1}{2}k'_p \frac{W}{L}(|V_{GS}| - |V_{TP}|)^2$$

Transconductance and Output Resistance

- **Transconductance (g_m):**

$$g_m = \frac{\partial I_D}{\partial V_{GS}} = k'_n \frac{W}{L}(V_{GS} - V_{TN}) = \sqrt{2k'_n \frac{W}{L} I_D} = \frac{2I_D}{V_{GS} - V_{TN}}$$

- **Output Resistance (r_o):**

$$r_o = \left(\frac{\partial I_D}{\partial V_{DS}}\right)^{-1} = \frac{1}{\lambda I_D}$$

Threshold Voltage

- **Threshold Voltage with Body Effect:**

$$V_T = V_{T0} + \gamma \left(\sqrt{|2\Phi_F| + V_{SB}} - \sqrt{|2\Phi_F|} \right)$$

Unit 2: CMOS Inverter Characteristics

Noise Margins

- **High and Low Noise Margins:**

$$NM_H = V_{OH} - V_{IH}, \quad NM_L = V_{IL} - V_{OL}$$

- **Switching Threshold (V_M / V_{inv}):**

$$V_M = \frac{V_{tn} + \sqrt{\frac{\beta_p}{\beta_n}}(V_{DD} - |V_{tp}|)}{1 + \sqrt{\frac{\beta_p}{\beta_n}}}$$

- **Input Voltages (Symmetric Inverter Approximations):**

$$V_{IL} = \frac{3V_{DD} + 2V_t}{8}, \quad V_{IH} = \frac{5V_{DD} - 2V_t}{8}$$

Depletion Load Inverter

- **Low Output Voltage (V_{OL}):**

$$\frac{\beta_{driver}}{2}[2(V_{DD} - V_{t,driver})V_{OL} - V_{OL}^2] = \frac{\beta_{load}}{2}|V_{t,dep}|^2$$

$$V_{OL} \approx \frac{|V_{t,dep}|^2}{2k(V_{DD} - V_{t,driver})} \quad \text{(approximate)}$$

Propagation Delay and Power Dissipation

- **Propagation Delay (t_p):**

$$t_{PHL} = 0.69 \cdot R_{eq,n} \cdot C_{load}, \quad t_{PLH} = 0.69 \cdot R_{eq,p} \cdot C_{load}$$

$$t_p = \frac{t_{PHL} + t_{PLH}}{2}$$

- **Power Dissipation:**

$$P_{dyn} = \alpha \cdot C_{load} \cdot V_{DD}^2 \cdot f, \quad P_{stat} = I_{leak} \cdot V_{DD}$$

$$P_{total} = P_{dyn} + P_{stat}$$

Unit 3: CMOS Logic Design

Transconductance Parameters

- **Process Transconductance (β):**

$$\beta_n = \mu_n C_{ox} \left(\frac{W}{L}\right)_n, \quad \beta_p = \mu_p C_{ox} \left(\frac{W}{L}\right)_p$$

- **Symmetric Inverter Condition:**

$$\frac{W_p}{W_n} = \frac{\mu_n}{\mu_p}$$

- **Beta Ratio (β_r):**

$$\beta_r = \frac{\beta_p}{\beta_n}$$

CMOS Logic Networks

- **Pull-Down Network Function:**

$$Y = \overline{f_{PDN}(A, B, C \dots)}$$

- **Aspect Ratio (Z):**

$$Z_{pd} = \frac{L_{pd}}{W_{pd}}, \quad Z_{pu} = \frac{L_{pu}}{W_{pu}}$$

Unit 4: Fabrication, Yield, and Interconnects

Resistance and Interconnects

- **Sheet Resistance:**

$$R = R_s \times n = R_s \times \frac{L}{W}$$

where $n = L/W$ is the number of squares.

Yield and Manufacturing

- **Number of Dies per Wafer:**

$$N = \frac{A_{wafer}}{A_{die}}$$

- **Yield Equation:**

$$Y = e^{-A_{die} \times D_0}$$

- **Number of Good Dies:**

$$N_{good} = N \times Y$$

Threshold Voltage Adjustment

- **Ion Implantation Shift:**

$$\Delta V_T = \frac{q \cdot D_I}{C_{ox}}$$

$$V_{T(final)} = V_{T0} \pm \Delta V_T$$

Unit 5: Arithmetic Circuits

Half Adder

- **Sum and Carry:**

$$Sum = A \oplus B, \quad Carry = A \cdot B$$

Full Adder

- **Sum and Carry Out:**

$$Sum = A \oplus B \oplus C_{in}$$

$$C_{out} = (A \cdot B) + (B \cdot C_{in}) + (A \cdot C_{in})$$