WINNER OF THE

*Frederick Emmons Terman Award in Electrical & Computer Engineering*

Exhibit #

Baker 07

01/10/2024 - JL

exhibitsticker.com

THIRD EDITION

# CMOS



## Circuit Design, Layout, and Simulation

R. JACOB BAKER

IEEE Series on Microelectronic Systems

WILEY

IEEE

IEEE PRESS

IPR2023-00697

Theta EX2019

This page intentionally left blank

# Multipliers

| Name | Symbol | Value |
|---|---|---|
| terra | T | $10^{12}$ |
| giga | G | $10^{9}$ |
| mega | M (MEG in SPICE) | $10^{6}$ |
| kilo | k | $10^{3}$ |
| milli | m | $10^{-3}$ |
| micro | μ (or u) | $10^{-6}$ |
| nano | n | $10^{-9}$ |
| pico | p | $10^{-12}$ |
| femto | f | $10^{-15}$ |
| atto | a (not used in SPICE) | $10^{-18}$ |

# Physical Constants

| Name | Symbol | Value/Units |
|---|---|---|
| Vacuum dielectric constant | $\varepsilon_0$ | 8.85 aF/μm |
| Silicon dielectric constant | $\varepsilon_{si}$ | $11.7\varepsilon_0$ |
| $SiO_2$ dielectric constant | $\varepsilon_{ox}$ | $3.97\varepsilon_0$ |
| $SiN_3$ dielectric constant | $\varepsilon_{Ni}$ | $16\varepsilon_0$ |
| Boltzmann's constant | k | $1.38 \times 10^{-23}$ J/K |
| Electronic charge | q | $1.6 \times 10^{-19}$ C |
| Temperature | T | Kelvin |
| Thermal voltage | $V_T$ | $kT/q$ = 26 mV @ 300K |

CMOS

**IEEE Press**
445 Hoes Lane
Piscataway, NJ 08854

**IEEE Press Editorial Board**
Lajos Hanzo, *Editor in Chief*

| | | |
|---|---|---|
| R. Abari | M. El-Hawary | S. Nahavandi |
| J. Anderson | B. M. Hammerli | W. Reeve |
| F. Canavero | M. Lanzerotti | T. Samad |
| T. G. Croda | O. Malik | G. Zobrist |

Kenneth Moore, *Director of IEEE Book and Information Services (BIS)*

IEEE Solid-State Circuits Society, *Sponsor*

**IPR2023-00697**
**Theta EX2019**

# CMOS

## Circuit Design, Layout, and Simulation

Third Edition

**R. Jacob Baker**

IEEE Press Series on Microelectronic Systems
Stuart K. Tewksbury and Joe E. Brewer, *Series Editors*



**IEEE PRESS**



**A JOHN WILEY & SONS, INC., PUBLICATION**

**IPR2023-00697**
**Theta EX2019**

Copyright © 2010 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permission.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic format. For information about Wiley products, visit our web site at www.wiley.com.

***Library of Congress Cataloging-in-Publication Data:***

Baker, R. Jacob, 1964-
CMOS : circuit design, layout, and simulation / Jake Baker. — 3rd ed.
p. cm.
Summary: "The third edition of CMOS: Circuit Design, Layout, and Simulation continues to cover the practical design of both analog and digital integrated circuits, offering a vital, contemporary view of a wide range of analog/digital circuit blocks, the BSIM model, data converter architectures, and much more. The 3rd edition completes the revised 2nd edition by adding one more chapter (chapter 30) at the end, which describes on implementing the data converter topologies discussed in Chapter 29. This additional, practical information should make the book even more useful as an academic text and companion for the working design engineer. Images, data presented throughout the book were updated, and more practical examples, problems are presented in this new edition to enhance the practicality of the book"— Provided by publisher.
Summary: "The third edition of CMOS: Circuit Design, Layout, and Simulation continues to cover the practical design of both analog and digital integrated circuits, offering a vital, contemporary view of a wide range of analog/digital circuit blocks, the BSIM
model, data converter architectures, and much more"— Provided by publisher.
ISBN 978-0-470-88132-3 (hardback)
1. Metal oxide semiconductors, Complementary—Design and construction. 2. Integrated circuits—Design and construction. 3. Metal oxide semiconductor field-effect transistors. I. Title.
TK7871.99.M44B35 2010
621.39'732—dc22 2010016630

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

**IPR2023-00697**
**Theta EX2019**

*To my wife Julie*

# Brief Contents


| | |
|---|---|
| Chapter 1 Introduction to CMOS Design | 1 |
| Chapter 2 The Well | 31 |
| Chapter 3 The Metal Layers | 59 |
| Chapter 4 The Active and Poly Layers | 83 |
| Chapter 5 Resistors, Capacitors, MOSFETs | 105 |
| Chapter 6 MOSFET Operation | 131 |
| Chapter 7 CMOS Fabrication *by Jeff Jessing* | 161 |
| Chapter 8 Electrical Noise: An Overview | 213 |
| Chapter 9 Models for Analog Design | 269 |
| Chapter 10 Models for Digital Design | 311 |
| Chapter 11 The Inverter | 331 |
| Chapter 12 Static Logic Gates | 353 |
| Chapter 13 Clocked Circuits | 375 |
| Chapter 14 Dynamic Logic Gates | 397 |
| Chapter 15 VLSI Layout Examples | 411 |
| Chapter 16 Memory Circuits | 433 |
| Chapter 17 Sensing Using $\Delta\Sigma$ Modulation | 483 |
| Chapter 18 Special Purpose CMOS Circuits | 523 |
| Chapter 19 Digital Phase-Locked Loops | 551 |
| Chapter 20 Current Mirrors | 613 |
| Chapter 21 Amplifiers | 657 |
| Chapter 22 Differential Amplifiers | 711 |
| Chapter 23 Voltage References | 745 |
| Chapter 24 Operational Amplifiers I | 773 |
| Chapter 25 Dynamic Analog Circuits | 829 |
| Chapter 26 Operational Amplifiers II | 863 |
| Chapter 27 Nonlinear Analog Circuits | 909 |
| Chapter 28 Data Converter Fundamentals *by Harry Li* | 931 |
| Chapter 29 Data Converter Architectures *by Harry Li* | 965 |
| Chapter 30 Implementing Data Converters | 1023 |
| Chapter 31 Feedback Amplifiers *with Harry Li* | 1099 |




IPR2023-00697
Theta EX2019

# Contents

**Preface** **xxxi**

**Chapter 1 Introduction to CMOS Design** **1**

1.1 The CMOS IC Design Process ..... 1

1.1.1 Fabrication 3

Layout and Cross-Sectional Views 4

1.2 CMOS Background ..... 6

The CMOS Acronym 6

CMOS Inverter 7

The First CMOS Circuits 7

Analog Design in CMOS 8

1.3 An Introduction to SPICE ..... 8

Generating a Netlist File 8

Operating Point 9

Transfer Function Analysis 10

The Voltage-Controlled Voltage Source 11

An Ideal Op-Amp 12

The Subcircuit 13

DC Analysis 13

Plotting IV Curves 14

Dual Loop DC Analysis 15

Transient Analysis 15

The SIN Source 16

An RC Circuit Example 17

Another RC Circuit Example 18

AC Analysis 19

Decades and Octaves 20

Decibels 20



IPR2023-00697
Theta EX2019

Pulse Statement 21
Finite Pulse Rise time 21
Step Response 22
Delay and Rise time in RC Circuits 22
Piece-Wise Linear (PWL) Source 23
Simulating Switches 24
Initial Conditions on a Capacitor 24
Initial Conditions in an Inductor 25
Q of an LC Tank 25
Frequency Response of an Ideal Integrator 26
Unity-Gain Frequency 26
Time-Domain Behavior of the Integrator 27
Convergence 28
Some Common Mistakes and Helpful Techniques 29

**Chapter 2 The Well 31**

The Substrate (The Unprocessed Wafer) 31
A Parasitic Diode 31
Using the N-well as a Resistor 32
2.1 Patterning ..... 32
2.1.1 Patterning the N-well 35
2.2 Laying Out the N-well ..... 36
2.2.1 Design Rules for the N-well 36
2.3 Resistance Calculation ..... 37
Layout of Corners 38
2.3.1 The N-well Resistor 38
2.4 The N-well/Substrate Diode ..... 39
2.4.1 A Brief Introduction to PN Junction Physics 39
Carrier Concentrations 40
Fermi Energy Level 42
2.4.2 Depletion Layer Capacitance 43
2.4.3 Storage or Diffusion Capacitance 45
2.4.4 SPICE Modeling 47
2.5 The RC Delay through the N-well ..... 49
RC Circuit Review 50
Distributed RC Delay 50
Distributed RC Rise Time 52
2.6 Twin Well Processes ..... 52

IPR2023-00697
Theta EX2019

Design Rules for the Well 53
SEM Views of Wells 55
**Chapter 3 The Metal Layers 59**
3.1 The Bonding Pad ..... 59
3.1.1 Laying Out the Pad I 60
Capacitance of Metal-to-Substrate 60
Passivation 62
An Important Note 62
3.2 Design and Layout Using the Metal Layers ..... 63
3.2.1 Metal1 and Via1 63
An Example Layout 63
3.2.2 Parasitics Associated with the Metal Layers 64
Intrinsic Propagation Delay 65
3.2.3 Current-Carrying Limitations 68
3.2.4 Design Rules for the Metal Layers 69
Layout of Two Shapes or a Single Shape 69
A Layout Trick for the Metal Layers 69
3.2.5 Contact Resistance 70
3.3 Crosstalk and Ground Bounce ..... 71
3.3.1 Crosstalk 71
3.3.2 Ground Bounce 72
DC Problems 72
AC Problems 72
A Final Comment 74
3.4 Layout Examples ..... 75
3.4.1 Laying Out the Pad II 75
3.4.2 Laying Out Metal Test Structures 78
SEM View of Metal 79
**Chapter 4 The Active and Poly Layers 83**
4.1 Layout Using the Active and Poly Layers ..... 83
The Active Layer 83
The P- and N-Select Layers 84
The Poly Layer 86
Self-Aligned Gate 86
The Poly Wire 88
Silicide Block 89
4.1.1 Process Flow 89

**IPR2023-00697**
**Theta EX2019**

Damascene Process Steps 90
4.2 Connecting Wires to Poly and Active ..... 92
Connecting the P-Substrate to Ground 93
Layout of an N-Well Resistor 94
Layout of an NMOS Device 95
Layout of a PMOS Device 96
A Comment Concerning MOSFET Symbols 96
Standard Cell Frame 97
Design Rules 98
4.3 Electrostatic Discharge (ESD) Protection ..... 100
Layout of the Diodes 100
**Chapter 5 Resistors, Capacitors, MOSFETs 105**
5.1 Resistors ..... 105
Temperature Coefficient (Temp Co) 105
Polarity of the Temp Co 106
Voltage Coefficient 107
Using Unit Elements 109
Guard Rings 110
Interdigitated Layout 110
Common-Centroid Layout 111
Dummy Elements 113
5.2 Capacitors ..... 113
Layout of the Poly-Poly Capacitor 114
Parasitics 115
Temperature Coefficient (Temp Co) 116
Voltage Coefficient 116
5.3 MOSFETs ..... 116
Lateral Diffusion 116
Oxide Encroachment 116
Source/Drain Depletion Capacitance 117
Source/Drain Parasitic Resistance 118
Layout of Long-Length MOSFETs 120
Layout of Large-Width MOSFETs 121
A Qualitative Description of MOSFET Capacitances 123
5.4 Layout Examples ..... 125
Metal Capacitors 125
Polysilicon Resistors 127

**IPR2023-00697**
**Theta EX2019**

**Chapter 6 MOSFET Operation 131**
6.1 MOSFET Capacitance Overview/Review ..... 132
Case I: Accumulation 132
Case II: Depletion 133
Case III: Strong Inversion 133
Summary 135
6.2 The Threshold Voltage ..... 135
Contact Potentials 137
Threshold Voltage Adjust 140
6.3 IV Characteristics of MOSFETs ..... 140
6.3.1 MOSFET Operation in the Triode Region 141
6.3.2 The Saturation Region 143
Cgs Calculation in the Saturation Region 145
6.4 SPICE Modeling of the MOSFET ..... 145
Model Parameters Related to $V_{THN}$ 146
Long-Channel MOSFET Models 146
Model Parameters Related to the Drain Current 146
SPICE Modeling of the Source and Drain Implants 147
Summary 147
6.4.1 Some SPICE Simulation Examples 148
Threshold Voltage and Body Effect 148
6.4.2 The Subthreshold Current 149
6.5 Short-Channel MOSFETs ..... 151
Hot Carriers 151
Lightly Doped Drain (LDD) 151
6.5.1 MOSFET Scaling 152
6.5.2 Short-Channel Effects 153
Negative Bias Temperature Instability (NBTI) 153
Oxide Breakdown 154
Drain-Induced Barrier Lowering 154
Gate-Induced Drain Leakage 154
Gate Tunnel Current 154
6.5.3 SPICE Models for Our Short-Channel CMOS Process 154
BSIM4 Model Listing (NMOS) 154
BSIM4 Model Listing (PMOS) 156
Simulation Results 157

**IPR2023-00697**
**Theta EX2019**

**Chapter 7 CMOS Fabrication *by Jeff Jessing*** **161**
7.1 CMOS Unit Processes ..... 161
7.1.1 Wafer Manufacture 161
Metallurgical Grade Silicon (MGS) 162
Electronic Grade Silicon (EGS) 162
Czochralski (CZ) Growth and Wafer Formation 162
7.1.2 Thermal Oxidation 163
7.1.3 Doping Processes 165
Ion Implantation 165
Solid State Diffusion 166
7.1.4 Photolithography 167
Resolution 168
Depth of Focus 168
Aligning Masks 170
7.1.5 Thin Film Removal 170
Thin Film Etching 170
Wet Etching 171
Dry Etching 171
Chemical Mechanical Polishing 173
7.1.6 Thin Film Deposition 173
Physical Vapor Deposition (PVD) 175
Chemical Vapor Depositon (CVD) 176
7.2 CMOS Process Integration ..... 177
FEOL 177
BEOL 177
CMOS Process Description 178
7.2.1 Frontend-of-the-Line Integration 180
Shallow Trench Isolation Module 181
Twin Tub Module 187
Gate Module 190
Source/Drain Module 193
7.2.2 Backend-of-the-Line Integration 199
Self-Aligned Silicide (Salicide) Module 199
Pre-Metal Dielectric 200
Contact Module 202
Metallization 1 203
Intra-Metal Dielectric 1 Deposition 205

**IPR2023-00697**
**Theta EX2019**

Via 1 Module 205
Metallization 2 207
Additional Metal/Dieletric Layers 208
Final Passivation 208
7.3 Backend Processes ..... 209
Wafer Probe 209
Die Separation 211
Packaging 211
Final Test and Burn-In 211
7.4 Summary ..... 211
**Chapter 8 Electrical Noise: An Overview 213**
8.1 Signals ..... 213
8.1.1 Power and Energy 213
Comments 215
8.1.2 Power Spectral Density 215
Spectrum Analyzers 216
8.2 Circuit Noise ..... 219
8.2.1 Calculating and Modeling Circuit Noise 219
Input-Referred Noise I 220
Noise Equivalent Bandwidth 220
Input-Referred Noise in Cascaded Amplifiers 223
Calculating $V_{onoise,RMS}$ from a Spectrum: A Summary 224
8.2.2 Thermal Noise 225
8.2.3 Signal-to-Noise Ratio 230
Input-Referred Noise II 231
Noise Figure 233
An Important Limitation of the Noise Figure 233
Optimum Source Resistance 236
Simulating Noiseless Resistors 236
Noise Temperature 239
Averaging White Noise 240
8.2.4 Shot Noise 242
8.2.5 Flicker Noise 244
8.2.6 Other Noise Sources 252
Random Telegraph Signal Noise 252
Excess Noise (Flicker Noise) 253
Avalanche Noise 253

IPR2023-00697
Theta EX2019

8.3 Discussion ..... 254
8.3.1 Correlation 254
Correlation of Input-Referred Noise Sources 256
Complex Input Impedance 256
8.3.2 Noise and Feedback 259
Op-Amp Noise Modeling 259
8.3.3 Some Final Notes Concerning Notation 262
**Chapter 9 Models for Analog Design 269**
9.1 Long-Channel MOSFETs ..... 269
9.1.1 The Square-Law Equations 271
PMOS Square-Law Equations 272
Qualitative Discussion 272
Threshold Voltage and Body Effect 276
Qualitative Discussion 276
The Triode Region 278
The Cutoff and Subthreshold Regions 278
9.1.2 Small Signal Models 279
Transconductance 280
AC Analysis 285
Transient Analysis 286
Body Effect Transconductance, $g_{mb}$ 287
Output Resistance 288
MOSFET Transition Frequency, $f_T$ 290
General Device Sizes for Analog Design 291
Subthreshold $g_m$ and $V_{THN}$ 292
9.1.3 Temperature Effects 293
Threshold Variation and Temperature 293
Mobility Variation with Temperature 295
Drain Current Change with Temperature 295
9.2 Short-Channel MOSFETs ..... 297
9.2.1 General Design (A Starting Point) 297
Output Resistance 298
Forward Transconductance 298
Transition Frequency 299
9.2.2 Specific Design (A Discussion) 300
9.3 MOSFET Noise Modeling ..... 302
Drain Current Noise Model 302

IPR2023-00697
Theta EX2019

**Chapter 10 Models for Digital Design** 311
Miller Capacitance 311
10.1 The Digital MOSFET Model ..... 312
Effective Switching Resistance 312
Short-Channel MOSFET Effective Switching Resistance 314
10.1.1 Capacitive Effects 315
10.1.2 Process Characteristic Time Constant 316
10.1.3 Delay and Transition Times 317
10.1.4 General Digital Design 320
10.2 The MOSFET Pass Gate ..... 321
The PMOS Pass Gate 322
10.2.1 Delay through a Pass Gate 323
The Transmission Gate (The TG) 324
10.2.2 Delay through Series-Connected PGs 325
10.3 A Final Comment Concerning Measurements ..... 326
**Chapter 11 The Inverter** 331
11.1 DC Characteristics ..... 331
Noise Margins 333
Inverter Switching Point 334
Ideal Inverter VTC and Noise Margins 334
11.2 Switching Characteristics ..... 337
The Ring Oscillator 339
Dynamic Power Dissipation 339
11.3 Layout of the Inverter ..... 341
Latch-Up 341
11.4 Sizing for Large Capacitive Loads ..... 344
Buffer Topology 344
Distributed Drivers 347
Driving Long Lines 348
11.5 Other Inverter Configurations ..... 349
NMOS-Only Output Drivers 350
Inverters with Tri-State Outputs 351
Additional Examples 351
**Chapter 12 Static Logic Gates** 353
12.1 DC Characteristics of the NAND and NOR Gates ..... 353
12.1.1 DC Characteristics of the NAND Gate 353

IPR2023-00697
Theta EX2019

12.1.2 DC Characteristics of the NOR Gate 356
A Practical Note Concerning $V_{SP}$ and Pass Gates 357
12.2 Layout of the NAND and NOR Gates ..... 358
12.3 Switching Characteristics ..... 358
Parallel Connection of MOSFETs 358
Series Connection of MOSFETs 359
12.3.1 NAND Gate 360
Quick Estimate of Delays 362
12.3.2 Number of Inputs 363
12.4 Complex CMOS Logic Gates ..... 364
Cascode Voltage Switch Logic 369
Differential Split-Level Logic 370
Tri-State Outputs 370
Additional Examples 370
**Chapter 13 Clocked Circuits 375**
13.1 The CMOS TG ..... 375
Series Connection of TGs 377
13.2 Applications of the Transmission Gate ..... 378
Path Selector 378
Static Circuits 379
13.3 Latches and Flip-Flops ..... 380
Basic Latches 380
An Arbiter 383
Flip-Flops and Flow-through Latches 383
An Edge-Triggered D-FF 386
Flip-Flop Timing 388
13.4 Examples ..... 389
**Chapter 14 Dynamic Logic Gates 397**
14.1 Fundamentals of Dynamic Logic ..... 397
14.1.1 Charge Leakage 398
14.1.2 Simulating Dynamic Circuits 401
14.1.3 Nonoverlapping Clock Generation 401
14.1.4 CMOS TG in Dynamic Circuits 402
14.2 Clocked CMOS Logic ..... 403
Clocked CMOS Latch 403
An Important Note 403
PE Logic 404

IPR2023-00697
Theta EX2019

Domino Logic 405
NP Logic (Zipper Logic) 407
Pipelining 407
**Chapter 15 VLSI Layout Examples 411**
15.1 Chip Layout ..... 412
Regularity 412
Standard Cell Examples 413
Power and Ground Considerations 417
An Adder Example 419
A 4-to-1 MUX/DEMUX 422
15.2 Layout Steps *by Dean Moriarty* ..... 422
Planning and Stick Diagrams 422
Device Placement 424
Polish 427
Standard Cells Versus Full-Custom Layout 427
**Chapter 16 Memory Circuits 433**
16.1 Array Architectures ..... 434
16.1.1 Sensing Basics 435
NMOS Sense Amplifier (NSA) 435
The Open Array Architecture 436
PMOS Sense Amplifier (PSA) 440
Refresh Operation 441
16.1.2 The Folded Array 441
Layout of the DRAM Memory Bit (Mbit) 443
16.1.3 Chip Organization 447
16.2 Peripheral Circuits ..... 448
16.2.1 Sense Amplifier Design 448
Kickback Noise and Clock Feedthrough 449
Memory 450
Current Draw 450
Contention Current (Switching Current) 450
Removing Sense Amplifier Memory 451
Creating an Imbalance and Reducing Kickback Noise 451
Increasing the Input Range 454
Simulation Examples 454
16.2.2 Row/Column Decoders 457
Global and Local Decoders 458

IPR2023-00697
Theta EX2019

Reducing Decoder Layout Area 460
16.2.3 Row Drivers 461
16.3 Memory Cells ..... 463
16.3.1 The SRAM Cell 463
16.3.2 Read-Only Memory (ROM) 464
16.3.3 Floating Gate Memory 466
The Threshold Voltage 467
Erasable Programmable Read-Only Memory 468
Two Important Notes 468
Flash Memory 469
**Chapter 17 Sensing Using ΔΣ Modulation 483**
17.1 Qualitative Discussion ..... 484
17.1.1 Examples of DSM 484
The Counter 485
Cup Size 486
Another Example 486
17.1.2 Using DSM for Sensing in Flash Memory 487
The Basic Idea 487
The Feedback Signal 492
Incomplete Settling 496
17.2 Sensing Resistive Memory ..... 497
The Bit Line Voltage 497
Adding an Offset to the Comparator 498
Schematic and Design Values 499
A Couple of Comments 502
17.3 Sensing in CMOS Imagers ..... 504
Resetting the Pixel 504
The Intensity Level 504
Sampling the Reference and Intensity Signals 505
Noise Issues 506
Subtracting $V_R$ from $V_S$ 508
Sensing Circuit Mismatches 517
**Chapter 18 Special Purpose CMOS Circuits 523**
18.1 The Schmitt Trigger ..... 523
18.1.1 Design of the Schmitt Trigger 524
Switching Characteristics 526
18.1.2 Applications of the Schmitt Trigger 527

IPR2023-00697
Theta EX2019

18.2 Multivibrator Circuits ..... 529
18.2.1 The Monostable Multivibrator 529
18.2.2 The Astable Multivibrator 530
18.3 Input Buffers ..... 531
18.3.1 Basic Circuits 531
Skew in Logic Gates 533
18.3.2 Differential Circuits 534
Transient Response 535
18.3.3 DC Reference 538
18.3.4 Reducing Buffer Input Resistance 541
18.4 Charge Pumps (Voltage Generators) ..... 542
Negative Voltages 543
Using MOSFETs for the Capacitors 544
18.4.1 Increasing the Output Voltage 544
18.4.2 Generating Higher Voltages: The Dickson Charge Pump 544
Clock Driver with a Pumped Output Voltage 546
NMOS Clock Driver 546
18.4.3 Example 547

**Chapter 19 Digital Phase-Locked Loops 551**

19.1 The Phase Detector ..... 553
19.1.1 The XOR Phase Detector 553
19.1.2 The Phase Frequency Detector 557
19.2 The Voltage-Controlled Oscillator ..... 561
19.2.1 The Current-Starved VCO 561
Linearizing the VCO's Gain 564
19.2.2 Source-Coupled VCOs 565
19.3 The Loop Filter ..... 567
19.3.1 XOR DPLL 568
Active-PI Loop Filter 573
19.3.2 PFD DPLL 575
Tri-State Output 575
Implementing the PFD in CMOS 576
PFD with a Charge Pump Output 578
Practical Implementation of the Charge Pump 579
Discussion 581
19.4 System Concerns ..... 582

IPR2023-00697
Theta EX2019

19.4.1 Clock Recovery from NRZ Data 584
The Hogge Phase Detector 588
Jitter 591
19.5 Delay-Locked Loops ..... 592
Delay Elements 595
Practical VCO and VCDL Design 596
19.6 Some Examples ..... 596
19.6.1 A 2 GHz DLL 596
19.6.2 A 1 Gbit/s Clock-Recovery Circuit 602
**Chapter 20 Current Mirrors 613**
20.1 The Basic Current Mirror ..... 613
20.1.1 Long-Channel Design 614
20.1.2 Matching Currents in the Mirror 616
Threshold Voltage Mismatch 616
Transconductance Parameter Mismatch 616
Drain-to-Source Voltage and Lambda 617
Layout Techniques to Improve Matching 617
Layout of the Mirror with Different Widths 620
20.1.3 Biasing the Current Mirror 621
Using a MOSFET-Only Reference Circuit 622
Supply Independent Biasing 624
20.1.4 Short-Channel Design 627
An Important Note 630
20.1.5 Temperature Behavior 631
Resistor-MOSFET Reference Circuit 631
MOSFET-Only Reference Circuit 633
Temperature Behavior of the Beta-Multiplier 634
Voltage Reference Using the Beta-Multiplier 634
20.1.6 Biasing in the Subthreshold Region 635
20.2 Cascoding the Current Mirror ..... 636
20.2.1 The Simple Cascode 636
DC Operation 637
Cascode Output Resistance 637
20.2.2 Low-Voltage (Wide-Swing) Cascode 639
An Important Practical Note 641
Layout Concerns 642
20.2.3 Wide-Swing, Short-Channel Design 642

IPR2023-00697
Theta EX2019

20.2.4 Regulated Drain Current Mirror 645
20.3 Biasing Circuits ..... 647
20.3.1 Long-Channel Biasing Circuits 647
Basic Cascode Biasing 648
The Folded-Cascode Structure 648
20.3.2 Short-Channel Biasing Circuits 650
Floating Current Sources 651
20.3.3 A Final Comment 651
**Chapter 21 Amplifiers 657**
21.1 Gate-Drain Connected Loads ..... 657
21.1.1 Common-Source (CS) Amplifiers 657
Miller's Theorem 660
Frequency Response 661
The Right-Hand Plane Zero 662
A Common-Source Current Amplifier 666
Common-Source Amplifier with Source Degeneration 667
Noise Performance of the CS Amplifier with Gate-Drain Load 669
21.1.2 The Source Follower (Common-Drain Amplifier) 670
21.1.3 Common Gate Amplifier 671
21.2 Current Source Loads ..... 671
21.2.1 Common-Source Amplifier 671
Class A Operation 672
Small-Signal Gain 673
Open Circuit Gain 673
High-Impedance and Low-Impedance Nodes 673
Frequency Response 674
Pole Splitting 676
Pole Splitting Summary 679
Canceling the RHP Zero 685
Noise Performance of the CS Amplifier with Current Source Load 686
21.2.2 The Cascode Amplifier 686
Frequency Response 687
Class A Operation 688
Noise Performance of the Cascode Amplifier 688
Operation as a Transimpedance Amplifier 688

IPR2023-00697
Theta EX2019

21.2.3 The Common-Gate Amplifier 689
21.2.4 The Source Follower (Common-Drain Amplifier) 690
Body Effect and Gain 691
Level Shifting 692
Input Capacitance 693
Noise Performance of the SF Amplifier 694
Frequency Behavior 694
SF as an Output Buffer 696
A Class AB Output Buffer Using SFs 697
21.3 The Push-Pull Amplifier ..... 698
21.3.1 DC Operation and Biasing 699
Power Conversion Efficiency 699
21.3.2 Small-Signal Analysis 702
21.3.3 Distortion 704
Modeling Distortion with SPICE 705
**Chapter 22 Differential Amplifiers 711**
22.1 The Source-Coupled Pair ..... 711
22.1.1 DC Operation 711
Maximum and Minimum Differential Input Voltage 712
Maximum and Minimum Common-Mode Input Voltage 713
Current Mirror Load 715
Biasing from the Current Mirror Load 717
Minimum Power Supply Voltage 717
22.1.2 AC Operation 718
AC Gain with a Current Mirror Load 719
22.1.3 Common-Mode Rejection Ratio 721
Input-Referred Offset from Finite CMRR 723
22.1.4 Matching Considerations 724
Input-Referred Offset with a Current Mirror Load 725
22.1.5 Noise Performance 726
22.1.6 Slew-Rate Limitations 727
22.2 The Source Cross-Coupled Pair ..... 727
Operation of the Diff-Amp 728
Input Signal Range 729
22.2.1 Current Source Load 731
Input Signal Range 732

IPR2023-00697
Theta EX2019

22.3 Cascode Loads (The Telescopic Diff-Amp) ..... 733
22.4 Wide-Swing Differential Amplifiers ..... 736
22.4.1 Current Differential Amplifier 737
22.4.2 Constant Transconductance Diff-Amp 738
Discussion 740
**Chapter 23 Voltage References 745**
23.1 MOSFET-Resistor Voltage References ..... 746
23.1.1 The Resistor-MOSFET Divider 746
23.1.2 The MOSFET-Only Voltage Divider 749
23.1.3 Self-Biased Voltage References 750
Forcing the Same Current through Each Side of the Reference 751
An Alternate Topology 756
23.2 Parasitic Diode-Based References ..... 757
Diode Behavior 758
The Bandgap Energy of Silicon 759
Lower Voltage Reference Design 760
23.2.1 Long-Channel BGR Design 761
Diode-Referenced Self-Biasing (CTAT) 761
Thermal Voltage-Referenced Self-Biasing (PTAT) 762
Bandgap Reference Design 765
Alternative BGR Topologies 766
23.2.2 Short-Channel BGR Design 768
The Added Amplifier 770
Lower Voltage Operation 770
**Chapter 24 Operational Amplifiers I 773**
24.1 The Two-Stage Op-Amp ..... 774
Low-Frequency, Open Loop Gain, $A_{OLDC}$ 774
Input Common-Mode Range 774
Power Dissipation 775
Output Swing and Current Source/Sinking Capability 775
Offsets 775
Compensating the Op-Amp 776
Gain and Phase Margins 781
Removing the Zero 782
Compensation for High-Speed Operation 783
Slew-Rate Limitations 787

IPR2023-00697
Theta EX2019

Common-Mode Rejection Ratio (CMRR) 789
Power Supply Rejection Ratio (PSRR) 790
Increasing the Input Common-Mode Voltage Range 791
Estimating Bandwidth in Op-Amps Circuits 792
24.2 An Op-Amp with Output Buffer ..... 793
Compensating the Op-Amp 794
24.3 The Operational Transconductance Amplifier (OTA) ..... 796
Unity-Gain Frequency, $f_{un}$ 797
Increasing the OTA Output Resistance 798
An Important Note 799
OTA with an Output Buffer (An Op-Amp) 800
The Folded-Cascode OTA and Op-Amp 803
24.4 Gain-Enhancement ..... 808
Bandwidth of the Added GE Amplifiers 809
Compensating the Added GE Amplifiers 811
24.5 Some Examples and Discussions ..... 812
A Voltage Regulator 812
Bad Output Stage Design 817
Three-Stage Op-Amp Design 820
**Chapter 25 Dynamic Analog Circuits 829**
25.1 The MOSFET Switch ..... 829
Charge Injection 830
Capacitive Feedthrough 831
Reduction of Charge Injection and Clock Feedthrough 832
kT/C Noise 833
25.1.1 Sample-and-Hold Circuits 834
25.2 Fully-Differential Circuits ..... 836
Gain 836
Common-Mode Feedback 837
Coupled Noise Rejection 838
Other Benefits of Fully-Differential Op-Amps 838
25.2.1 A Fully-Differential Sample-and-Hold 838
Connecting the Inputs to the Bottom (Poly1) Plate 840
Bottom Plate Sampling 841
SPICE Simulation 841
25.3 Switched-Capacitor Circuits ..... 843
25.3.1 Switched-Capacitor Integrator 845

IPR2023-00697
Theta EX2019

Parasitic Insensitive 846
Other Integrator Configurations 846
Exact Frequency Response of a Switched-Capacitor Integrator 849
Capacitor Layout 851
Op-Amp Settling Time 852
25.4 Circuits ..... 853
Reducing Offset Voltage of an Op-Amp 853
Dynamic Comparator 854
Dynamic Current Mirrors 856
Dynamic Amplifiers 858

**Chapter 26 Operational Amplifiers II 863**

26.1 Biasing for Power and Speed ..... 863
26.1.1 Device Characteristics 864
26.1.2 Biasing Circuit 865
Layout of Differential Op-Amps 865
Self-Biased Reference 866
26.2 Basic Concepts ..... 867
Modeling Offset 867
A Diff-Amp 867
A Single Bias Input Diff-Amp 868
The Diff-Amp's Tail Current Source 868
Using a CMFB Amplifier 869
Compensating the CMFB Loop 871
Extending the CMFB Amplifier Input Range 873
Dynamic CMFB 874
26.3 Basic Op-Amp Design ..... 876
The Differential Amplifier 877
Adding a Second Stage (Making an Op-Amp) 878
Step Response 880
Adding CMFB 881
CMFB Amplifier 882
The Two-Stage Op-Amp with CMFB 883
Origin of the Problem 884
Simulation Results 886
Using MOSFETs Operating in the Triode Region 887
Start-up Problems 887

IPR2023-00697
Theta EX2019

Lowering Input Capacitance 887
Making the Op-Amp More Practical 888
Increasing the Op-Amp's Open-Loop Gain 889
Offsets 892
Op-Amp Offset Effects on Outputs 893
Single-Ended to Differential Conversion 894
CMFB Settling Time 895
CMFB in the Output Buffer (Fig. 26.43) or the Diff-Amp (Fig. 26.40)? 895
26.4 Op-Amp Design Using Switched-Capacitor CMFB ..... 896
Clock Signals 896
Switched-Capacitor CMFB 896
The Op-Amp's First Stage 898
The Output Buffer 900
An Application of the Op-Amp 901
Simulation Results 902
A Final Note Concerning Biasing 904
**Chapter 27 Nonlinear Analog Circuits 909**
27.1 Basic CMOS Comparator Design ..... 909
Preamplification 910
Decision Circuit 910
Output Buffer 913
27.1.1 Characterizing the Comparator 915
Comparator DC Performance 915
Transient Response 916
Propagation Delay 918
Minimum Input Slew Rate 918
27.1.2 Clocked Comparators 918
27.1.3 Input Buffers Revisited 920
27.2 Adaptive Biasing ..... 920
27.3 Analog Multipliers ..... 923
27.3.1 The Multiplying Quad 924
Simulating the Operation of the Multiplier 926
27.3.2 Multiplier Design Using Squaring Circuits 928
**Chapter 28 Data Converter Fundamentals *by Harry Li* 931**
28.1 Analog Versus Discrete Time Signals ..... 931
28.2 Converting Analog Signals to Digital Signals ..... 932

**IPR2023-00697**
**Theta EX2019**

28.3 Sample-and-Hold (S/H) Characteristics ..... 935
Sample Mode 936
Hold Mode 937
Aperture Error 937
28.4 Digital-to-Analog Converter (DAC) Specifications ..... 938
Differential Nonlinearity 941
Integral Nonlinearity 943
Offset 945
Gain Error 945
Latency 945
Signal-to-Noise Ratio (SNR) 945
Dynamic Range 947
28.5 Analog-to-Digital Converter (ADC) Specifications ..... 947
Quantization Error 948
Differential Nonlinearity 950
Missing Codes 951
Integral Nonlinearity 951
Offset and Gain Error 953
Aliasing 953
Signal-to-Noise Ratio 956
Aperture Error 956
28.6 Mixed-Signal Layout Issues ..... 957
Floorplanning 958
Power Supply and Ground Issues 958
Fully Differential Design 960
Guard Rings 960
Shielding 961
Other Interconnect Considerations 962

**Chapter 29 Data Converter Architectures *by Harry Li* 965**

29.1 DAC Architectures ..... 965
29.1.1 Digital Input Code 965
29.1.2 Resistor String 966
Mismatch Errors Related to the Resistor-String DAC 967
Integral Nonlinearity of the Resistor-String DAC 969
Differential Nonlinearity of the Worst-Case Resistor-String DAC 970
29.1.3 R-2R Ladder Networks 971

IPR2023-00697
Theta EX2019

29.1.4 Current Steering 973
Mismatch Errors Related to Current-Steering DACs 976
29.1.5 Charge-Scaling DACs 978
Layout Considerations for a Binary-Weighted Capacitor Array 980
The Split Array 980
29.1.6 Cyclic DAC 982
29.1.7 Pipeline DAC 984
29.2 ADC Architectures ..... 985
29.2.1 Flash 985
Accuracy Issues for the Flash ADC 988
29.2.2 The Two-Step Flash ADC 990
Accuracy Issues Related to the Two-Step Flash Converter 992
Accuracy Issues Related to Operational Amplifiers 992
29.2.3 The Pipeline ADC 994
Accuracy Issues Related to the Pipeline Converter 996
29.2.4 Integrating ADCs 998
Single-Slope Architecture 998
Accuracy Issues Related to the Single-Slope ADC 1000
Dual-Slope Architecture 1000
Accuracy Issues Related to the Dual-Slope ADC 1002
29.2.5 The Successive Approximation ADC 1003
The Charge-Redistribution Successive Approximation ADC 1005
29.2.6 The Oversampling ADC 1007
Differences in Nyquist Rate and Oversampled ADCs 1007
The First-Order ΔΣ Modulator 1008
The Higher Order ΔΣ Modulators 1010

**Chapter 30 Implementing Data Converters 1023**

30.1 R-2R Topologies for DACs ..... 1024
30.1.1 The Current-Mode R-2R DAC 1024
30.1.2 The Voltage-Mode R-2R DAC 1025
30.1.3 A Wide-Swing Current-Mode R-2R DAC 1026
DNL Analysis 1029
INL Analysis 1029
Switches 1030
Experimental Results 1030

IPR2023-00697
Theta EX2019

Improving DNL (Segmentation) 1032
Trimming DAC Offset 1034
Trimming DAC Gain 1036
Improving INL by Calibration 1037
30.1.4 Topologies Without an Op-Amp 1038
The Voltage-Mode DAC 1038
Two Important Notes Concerning Glitches 1041
The Current-Mode (Current Steering) DAC 1042
30.2 Op-Amps in Data Converters ..... 1045
Gain Bandwidth Product of the Noninverting Op-Amp Topology 1045
Gain Bandwidth Product of the Inverting Op-Amp Topology 1046
30.2.1 Op-Amp Gain 1047
30.2.2 Op-Amp Unity Gain Frequency 1048
30.2.3 Op-Amp Offset 1049
Adding an Auxiliary Input Port 1049
30.3 Implementing ADCs ..... 1052
30.3.1 Implementing the S/H 1052
A Single-Ended to Differential Output S/H 1054
30.3.2 The Cyclic ADC 1059
Comparator Placement 1061
Implementing Subtraction in the S/H 1062
Understanding Output Swing 1065
30.3.3 The Pipeline ADC 1067
Using 1.5 Bits/Stage 1068
Capacitor Error Averaging 1075
Comparator Placement 1082
Clock Generation 1082
Offsets and Alternative Design Topologies 1084
Dynamic CMFB 1089
Layout of Pipelined ADCs 1090
**Chapter 31 Feedback Amplifiers *with Harry Li*** **1099**
31.1 The Feedback Equation ..... 1100
31.2 Properties of Negative Feedback on Amplifier Design ..... 1101
31.2.1 Gain Desensitivity 1101
31.2.2 Bandwidth Extension 1101

IPR2023-00697
Theta EX2019

31.2.3 Reduction in Nonlinear Distortion 1103
31.2.4 Input and Output Impedance Control 1104
31.3 Recognizing Feedback Topologies ..... 1105
31.3.1 Input Mixing 1106
31.3.2 Output Sampling 1106
31.3.3 The Feedback Network 1107
An Important Assumption 1107
Counting Inversions Around the Loop 1108
Examples of Recognizing Feedback Topologies 1109
31.3.4 Calculating Open-Loop Parameters 1110
31.3.5 Calculating Closed-Loop Parameters 1112
31.4 The Voltage Amp (Series-Shunt Feedback) ..... 1113
31.5 The Transimpedance Amp (Shunt-Shunt Feedback) ..... 1119
31.5.1 Simple Feedback Using a Gate-Drain Resistor 1125
31.6 The Transconductance Amp (Series-Series Feedback) .... 1128
31.7 The Current Amplifier (Shunt-Series Feedback) ..... 1132
31.8 Stability ..... 1135
31.8.1 The Return Ratio 1139
31.9 Design Examples 1141
31.9.1 Voltage Amplifiers 1141
Amplifiers with Gain 1143
31.9.2 A Transimpedance Amplifier ..... 1145
**Index** **1157**
**About the Author** **1174**

**IPR2023-00697**
**Theta EX2019**

# Preface

CMOS (complementary metal oxide semiconductor) technology continues to be the dominant technology for fabricating integrated circuits (ICs or chips). This dominance will likely continue for the next 25 years and perhaps even longer. Why? CMOS technology is reliable, manufacturable, low power, low cost, and, perhaps most importantly, scalable. The fact that silicon integrated circuit technology is scalable was observed and described in 1965 by Intel founder Gordon Moore. His observations are now referred to as *Moore's law* and state that the number of devices on a chip will double every 18 to 24 months. While originally not specific to CMOS, Moore's law has been fulfilled over the years by scaling down the feature size in CMOS technology. Whereas the gate lengths of early CMOS transistors were in the micrometer range (long-channel devices) the feature sizes of current CMOS devices are in the nanometer range (short-channel devices).

To encompass both the long- and short-channel CMOS technologies in this book, a two-path approach to custom CMOS integrated circuit design is adopted. Design techniques are developed for both and then compared. This comparison gives readers deep insight into the circuit design process. While the square-law equations used to describe MOSFET operation that students learn in an introductory course in microelectronics can be used for analog design in a long-channel CMOS process they are not useful when designing in short-channel, or nanometer, CMOS technology. The behavior of the devices in a nanometer CMOS process is quite complex. Simple equations to describe the devices' behavior are not possible. Rather electrical plots are used to estimate biasing points and operating behavior. It is still useful, however, for the student to use mathematical rigor when learning circuit analysis and design and, hence, the reason for the two-path approach. Hand calculations can be performed using a long-channel CMOS technology with the results then used to describe how to design in a nano-CMOS process.



IPR2023-00697
Theta EX2019

What's new in the third edition of CMOS? The information discussing computer-aided design (CAD) tools (e.g., Cadence, Electric, HSPICE, LASI, LTspice, and WinSpice) has been moved to the book's webpage, **http://CMOSedu.com**. In addition, chapters were added covering the implementation of data converters and feedback amplifiers. This additional, practical, information should make the book even more useful as an academic text and companion for the working design engineer. As in the earlier editions, the book is filled with practical design examples, discussions, and problems. The solutions to the end-of-chapter problems (for self-study) and the netlists used when simulating the circuits are found at CMOSedu.com. Additional problems are also found at this website. Those interested in gaining an in-depth knowledge of CMOS analog and digital design will be greatly aided by downloading, modifying, and simulating the design examples found in the book.

The assumed background of the reader is a knowledge of linear circuits (e.g., RC and RLC circuits, Bode plots, Laplace transforms, AC analysis, etc.), microelectronics (e.g., diodes, transistors, small-signal analysis, amplifiers, switching behavior, etc.), and digital logic design. Several courses can be taught using this book including VLSI or CMOS digital IC design (chapters 1–7 and 10–19), CMOS analog IC design (chapters 9 and 20–24), and advanced analog IC design (chapters 8 and 25–31).

***How will this book be useful to the student, researcher, or practicing engineer?***

A great deal of effort has gone into making this book useful to an eclectic audience. For the student, the book is filled with hundreds of examples, problems, and practical discussions (according to one student there can never be too many examples in a textbook). The layout discussions build a knowledge foundation important for troubleshooting and precision or high-speed design. Layout expertise is gained in a step-by-step fashion by including circuit design details, process steps, and simulation concerns (parasitics). Covering layout in a single chapter and decoupling the discussions from design and simulation was avoided. The digital design chapters emphasize real-world process parameters (e.g., $I_{off}$ , $I_{on}$ , $t_{ox}$ , $VDD$). The analog chapters provide coherent discussions about selecting device sizes and design considerations. Likewise "cookbook" design procedures for selecting the widths/lengths of MOSFETs and design using long-channel equations in a short-channel process are not present. The focus is on preparing the student to "hit the ground running" when they become a custom CMOS IC designer or product engineer.

For the researcher, topics in circuit design such as noise considerations and sensing using delta-sigma modulation (DSM) continue to be important in nanometer CMOS. For example, in Ch. 17 the use of DSM has been applied to CMOS image sensors, Flash memory, and memory using thin oxides (direct tunneling). Sensing using DSM is important because it makes use of the fact that as CMOS clock speeds are going up, the gain and matching of transistors is deteriorating. Further, Ch. 8 discusses noise-limited design issues such as, "Why can't I improve the signal-to-noise ratio in my imaging chip?" or "Why is integrating thermal or flicker noise harmful?"

For the working engineer, the book provides design and layout examples that will be immediately useful in products. At the risk of stating the obvious, matching, power, speed, process shifts, power supply voltage variations, and temperature behavior are extremely important in practical design. The discussions and examples found in this book

IPR2023-00697
Theta EX2019

are focused on these topics. Phase-locked loops, charge pumps, low-voltage references, single and fully-differential op-amp designs, continuous-time and clocked comparators, memory circuits, etc., are covered in detail with numerous examples. To ensure the most practical computer validation of the designs, the simulations for the nanometer designs (a 50 nm process) use the BSIM4 SPICE models. Again, all of the book's simulation examples are available at CMOSedu.com.

***Acknowledgments***

I would like to thank the reviewers, students, colleagues, and friends that have helped to make this book a possibility: Jenn Ambrose, Jeanne Audino, Rupa Balan, Sakkarapani Balagopal, Mahesh Balasubramanian, David Binkley, Jan Bissey, Bill Black, Lincoln Bollschweiler, Eric Booth, Dave Boyce, Elizabeth Brauer, John Brews, Ben Brown, J. W. Bruce, Prashanth Busa, Kris Campbell, John Chiasson, Kloy Debban, Ahmad Dowlatabadi, Robert Drost, Kevin Duesman, Krishna Duvvada, Mike Engelhardt, Surendranath Eruvuru, Cathy Faduska, Paul Furth, Chris Gagliano, Gilda Garretón, Neil Goldsman, Tyler Gomm, Shantanu Gupta, Kory Hall, Wes Hansford, David Harris, Qawi Harvard, Robert Hay, Jeff Jessing (for authoring Ch. 7), Adam Johnson, Brent Keeth, Howard Kirsch, Bill Knowlton, Bhavana Kollimarla, Harry Li (for authoring Chs. 28 and 29 and for coauthoring Ch. 31), Matthew Leslie, Song Liu, Mary Mann, Mary Miller, Amy Moll, Dennis Montierth, Dean Moriarty (for authoring Sec. 15.2) Sugato Mukherjee, Michael Newman, Ward Parkinson, Winway Pang, Priyanka Mukeshbhai Parikh, Andrew Prince, Mahyar Arjmand Rad, Avinash Rajagiri, Harikrishna Rapole, Steven Rubin, Vishal Saxena, Terry Sculley (for deriving the INL and DNL equations found in Ch. 29), Brian Shirley, Harish Singidi, Joseph Skudlarek, Mike Smith, Avani Falgun Trivedi, Mark Tuttle, Vance Tyree, Gary VanAckern, Lisa VanHorn, Indira Vemula, Tony VenGraitis, Joseph J. Walsh, Justin Wood, Kuangming Yap, and Geng Zheng.

R. Jacob (Jake) Baker

IPR2023-00697
Theta EX2019

This page intentionally left blank

Chapter

1

# Introduction to CMOS Design

This chapter provides a brief introduction to the CMOS (complementary metal oxide semiconductor) integrated circuit (IC) design process (the design of "chips"). CMOS is used in most very large scale integrated (VLSI) or ultra-large scale integrated (ULSI) circuit chips. The term "VLSI" is generally associated with chips containing thousands or millions of metal oxide semiconductor field effect transistors (MOSFETs). The term "ULSI" is generally associated with chips containing billions, or more, MOSFETs. We'll avoid the use of these descriptive terms in this book and focus simply on "digital and analog CMOS circuit design."

We'll also introduce circuit simulation using SPICE (simulation program with integrated circuit emphasis). The introduction will be used to review basic circuit analysis and to provide a quick reference for SPICE syntax.

## 1.1 The CMOS IC Design Process

The CMOS circuit design process consists of defining circuit inputs and outputs, hand calculations, circuit simulations, circuit layout, simulations including parasitics, reevaluation of circuit inputs and outputs, fabrication, and testing. A flowchart of this process is shown in Fig. 1.1. The circuit specifications are rarely set in concrete; that is, they can change as the project matures. This can be the result of trade-offs made between cost and performance, changes in the marketability of the chip, or simply changes in the customer's needs. In almost all cases, major changes after the chip has gone into production are not possible.

This text concentrates on custom IC design. Other (noncustom) methods of designing chips, including field-programmable-gate-arrays (FPGAs) and standard cell libraries, are used when low volume and quick design turnaround are important. Most chips that are mass produced, including microprocessors and memory, are examples of chips that are custom designed.

The task of laying out the IC is often given to a layout designer. However, it is extremely important that the engineer can lay out a chip (and can provide direction to the layout designer on how to layout a chip) and understand the parasitics involved in the

IPR2023-00697
Theta EX2019

layout. Parasitics are the stray capacitances, inductances, pn junctions, and bipolar transistors, with the associated problems (breakdown, stored charge, latch-up, etc.). A fundamental understanding of these problems is important in precision/high-speed design.

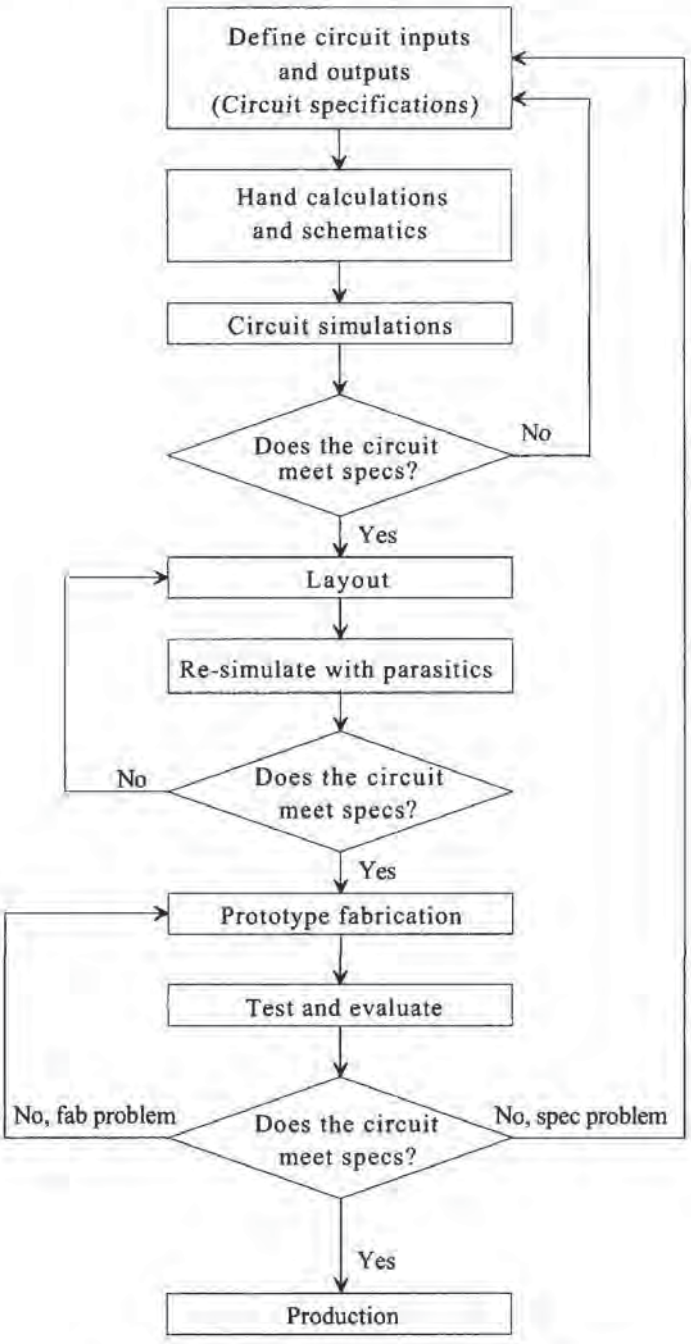



**Figure 1.1** Flowchart for the CMOS IC design process.

IPR2023-00697
Theta EX2019

### 1.1.1 Fabrication

CMOS integrated circuits are fabricated on thin circular slices of silicon called wafers. Each wafer contains several (perhaps hundreds or even thousands) of individual **chips or "die"** (Fig. 1.2). For production purposes, each die on a wafer is usually identical, as seen in the photograph in Fig. 1.2. Added to the wafer are test structures and process monitor plugs (sections of the wafer used to monitor process parameters). The most common wafer size (diameter) in production at the time of this writing is 300 mm (12 inch).

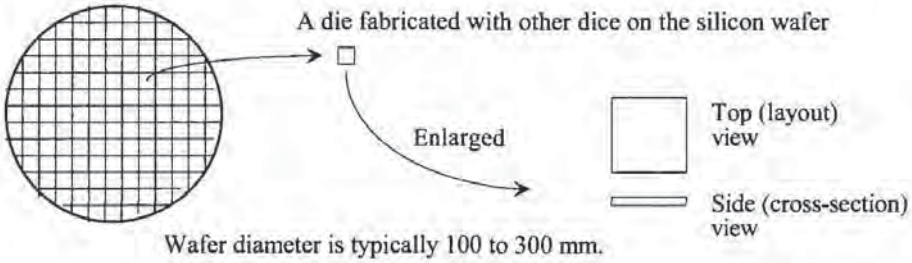





**Figure 1.2** CMOS integrated circuits are fabricated on and in a silicon wafer. Shown are 150, 200, and 300 mm diameter wafers. Notice the reflection of ceiling tiles in the 300 mm wafer.

The ICs we design and lay out using a layout program can be fabricated through MOSIS (http://mosis.com) on what is called a **multiproject wafer**; that is, a wafer that is comprised of chip designs of varying sizes from different sources (educational, private, government, etc.). MOSIS combines multiple chips on a wafer to split the fab cost among several designs to keep the cost low. MOSIS subcontracts the fabrication of the chip designs (multiproject wafer) out to one of many commercial manufacturers (vendors). MOSIS takes the wafers it receives from the vendors, after fabrication, and cuts them up to isolate the individual chip designs. The chips are then packaged and sent to the originator. A sample package (40-pin ceramic) from a MOSIS-submitted student design is seen in Fig. 1.3. Normally a cover (not shown) keeps the chip from being exposed to light or accidental damage.

IPR2023-00697
Theta EX2019

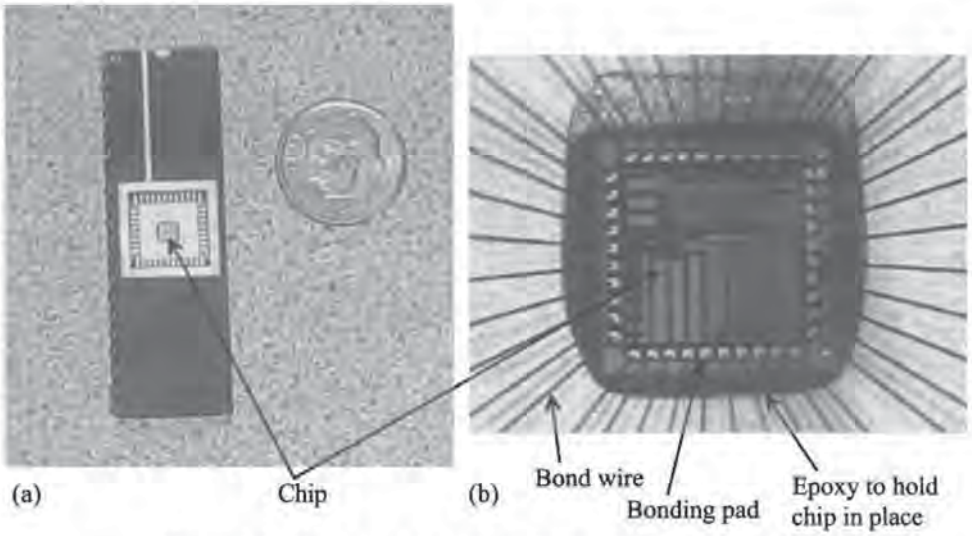



**Figure 1.3** How a chip is packaged (a) and (b) a closer view.

Note, in Fig. 1.3, that the chip's electrical signals are transmitted to the pins of the package through wires. These wires (called "bond wires") electrically bond the chip to the package so that a pin of the chip is electrically connected (shorted) to a piece of metal on the chip (called a bonding pad). The chip is held in the cavity of the package with an epoxy resin ("glue") as seen in Fig. 1.3b.

The ceramic package used in Fig. 1.3 isn't used for most mass-produced chips. Most chips that are mass produced use plastic packages. Exceptions to this statement are chips that dissipate a lot of heat or chips that are placed directly on a printed circuit board (where they are simply "packaged" using a glob of resin). Plastic packaged (encapsulated) chips place the die on a lead frame (Fig. 1.4) and then encapsulate the die and lead frame in plastic. The plastic is melted around the chip. After the chip is encapsulated, its leads are bent to the correct position. This is followed by printing information on the chip (the manufacturer, the chip type, and the lot number) and finally placing the chip in a tube or reel for shipping to a company that makes products that use the chips. Example products might include chips that are used in cell phones, computers, microwave ovens, printers.

*Layout and Cross Sectional Views*

The view that we see when laying out a chip is the top, or layout, view of the die. However, to understand the parasitics and how the circuits are connected together, it's important to understand the chip's cross-sectional view. Since we will often show a layout view followed by a cross-sectional view, let's make sure we understand the difference and how to draw a cross-section from a layout. Figure 1.5a shows the layout (top) view of a pie. In (b) we show the cross-section of the pie (without the pie tin) at the line indicated in (a). To "lay-out" a pie we might have layers called: crust, filling, caramel, whipped-cream, nuts, etc. We draw these layers to indicate how to assemble the pie (e.g., where to place nuts on the top). Note that *the order we draw the layers doesn't matter*. We could draw the nuts (on the top of the pie) first and then the crust. When we fabricate the pie, the order does matter (the crust is baked before the nuts are added).

IPR2023-00697
Theta EX2019

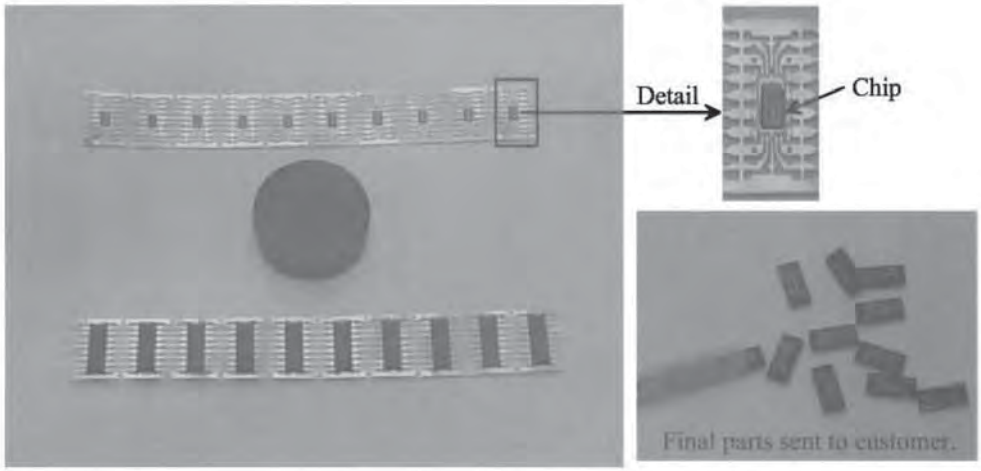



**Figure 1.4** Plastic packages are used (generally) when the chip is mass produced.

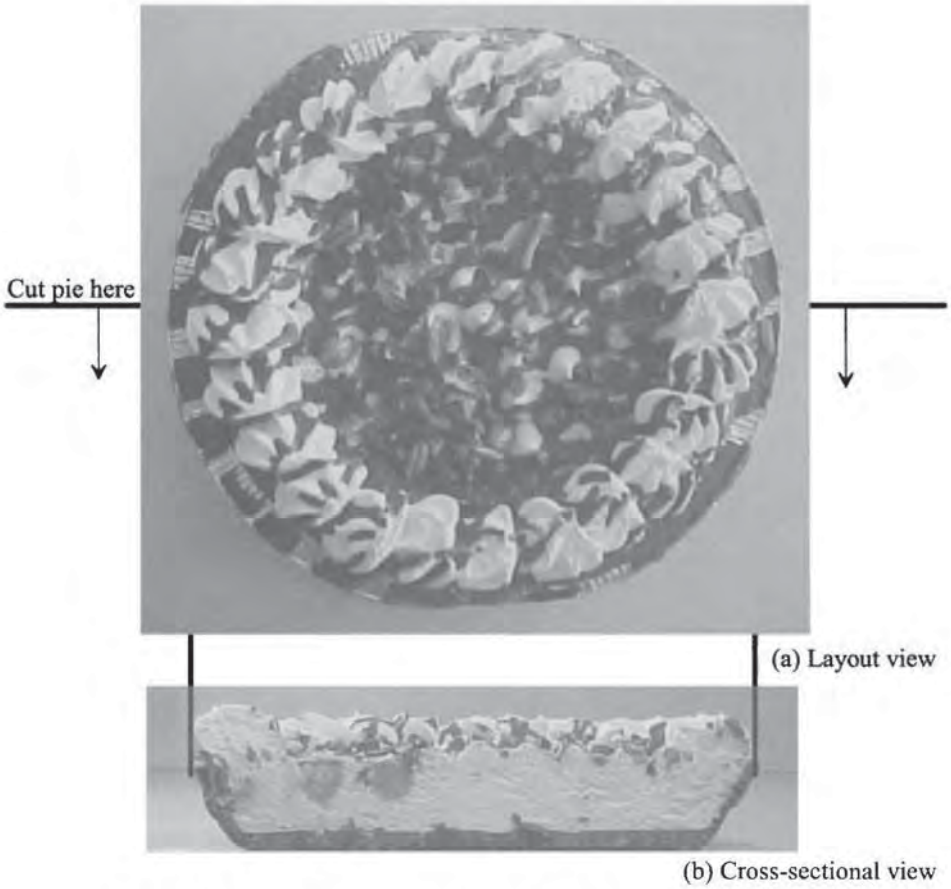



**Figure 1.5** Layout and cross sectional view of a pie (minus pie tin).

IPR2023-00697
Theta EX2019

## 1.2 CMOS Background

CMOS circuit design (the idea and basic concepts) was invented in 1963 by Frank Wanlass while at Fairchild Semiconductor, see US Patent 3,356,858, [5]. The idea that a circuit could be made with discrete complementary MOS devices, an NMOS (n-channel MOSFET) transistor (Fig. 1.6) and a PMOS (p-channel) transistor (Fig. 1.7) was quite novel at the time given the immaturity of MOS technology and the rising popularity of the bipolar junction transistor (BJT) as a replacement for the vacuum tube.

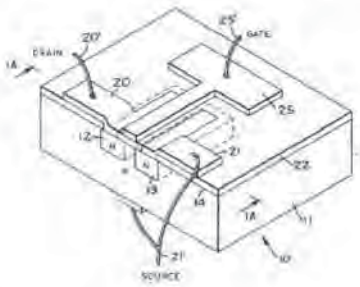


**Figure 1.6** Discrete NMOS device from US Patent 3,356,858 [5]. Note the metal gate and the connection to the MOSFET's body on the bottom of the device. Also note that the source and body are tied together.

*The CMOS Acronym*

Note in Figs. 1.6 and 1.7 the use of a metal gate and the connection to the MOSFET's body on the bottom of the transistor (these are discrete devices). As we'll see later in the book (e.g., Fig. 4.3) the gate material used in a modern MOSFET is no longer metal but rather polysilicon. Strictly speaking, modern technology is not CMOS then but rather CPOS (complementary-polysilicon-oxide-semiconductor). US Patent 3,356,858 refers to the use of insulated field effect transistors (IFETs). The acronym IFET is perhaps, even today, a more appropriate descriptive term than MOSFET. Others (see the footnote on page 154) have used the term IGFET (insulated-gate-field-effect-transistor) to describe the devices. We'll stick to the ubiquitous terms MOSFET and CMOS since they are standard terms that indicate devices, design, or technology using complementary field effect devices.

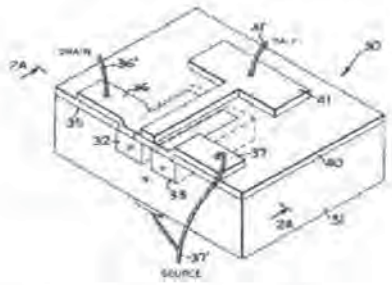


**Figure 1.7** Discrete PMOS device from US Patent 3,356,858 [5].

IPR2023-00697
Theta EX2019

*CMOS Inverter*

Figure 1.8 shows the schematic of a CMOS inverter. Note the use of a modified bipolar symbol for the MOSFET (see Fig. 4.14 and the associated discussion). Also note that the connections of the sources (the terminals with arrows) and drains are backwards from most circuit design and schematic drawing practices. Current flows from the top of the schematic to the bottom, and the arrow indicates the direction of current flow.

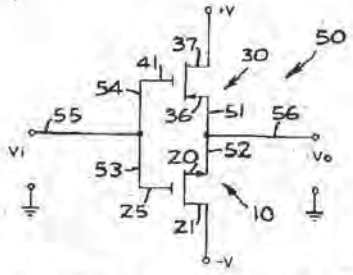



**Figure 1.8** Inverter schematic from US Patent 3,356,858 [5].

When the input voltage, $V_i$, is $-V$ (the negative supply rail), the output, $V_o$, goes to $+V$ (the positive supply voltage). The NMOS device (bottom) shuts off and the PMOS device (top) turns on. When the input goes to $+V$, the output goes to $-V$ turning on the NMOS and turning off the PMOS. So if a logic 0 corresponds to $-V$ and a logic 1 to $+V$, the circuit performs the logical inversion operation. This topology has several advantages over digital circuits implemented using BJTs including an output swing that goes to the power supply rails, very low static power dissipation, and no storage time delays (see Sec. 2.4.3).

*The First CMOS Circuits*

In 1968 a group led by Albert Medwin at RCA made the first commercial CMOS integrated circuits (the 4000 series of CMOS logic gates). At first CMOS circuits were a low-power, but slower, alternative to BJT logic circuits using TTL (transistor-transistor logic) digital logic. During the 1970s, the makers of watches used CMOS technology because of the importance of long battery life. Also during this period, MOS technology was used for computing processor development, which ultimately led to the creation of the personal computer market in the 1980s and the use of internet, or web, technology in the 1990s. It's likely that the MOS transistor is the most manufactured device in the history of mankind.

Currently more than 95% of integrated circuits are fabricated in CMOS. For the present, and foreseeable future, CMOS will remain the dominant technology used to fabricate integrated circuits. There are several reasons for this dominance. CMOS ICs can be laid out in a small area. They can handle very high operating speeds while dissipating relatively low power. Perhaps the most important aspect of CMOS's dominance is its manufacturability. CMOS circuits can be fabricated with few defects. Equally important, the cost to fabricate in CMOS has been kept low by shrinking devices (scaling) with each new generation of technology. This also, for digital circuits, is significant because in many cases the same layout can be used from one fabrication size (process technology node) to the next via simple scaling.

IPR2023-00697
Theta EX2019

*Analog Design in CMOS*

While initially CMOS was used exclusively for digital design, the constant push to lower costs and increase the functionality of ICs has resulted in it being used for analog-only, analog/digital, and mixed-signal (chips that combine analog circuits with digital signal processing) designs. The main concern when using CMOS for an analog design is matching. Matching is a term used to describe how well two identical transistors' characteristics match electrically. How well circuits "match" is often the limitation in the quality of a design (e.g., the clarity of a monitor, the accuracy of a measurement, etc.).

## 1.3 An Introduction to SPICE

The simulation program with an integrated circuit emphasis (SPICE) is a ubiquitous software tool for the simulation of circuits. In this section we'll provide an overview of SPICE. In addition, we'll provide some basic circuit analysis examples for quick reference or as a review. Note that the reader should review the links at CMOSedu.com for SPICE download and installation information. In addition, the examples from the book are available at this website. Note that all SPICE engines use a text file (a netlist) for simulation input.

*Generating a Netlist File*

We can use, among others, the Window's notepad or wordpad programs to create a SPICE netlist. SPICE likes to see files with "*.cir, *.sp, or *.spi" (among others) extensions. To save a file with these extensions, place the file name and extension in quotes, as seen in Fig. 1.9. If quotes are not used, then Windows may tack on ".txt" to the filename. This can make finding the file difficult when opening the netlist in SPICE.

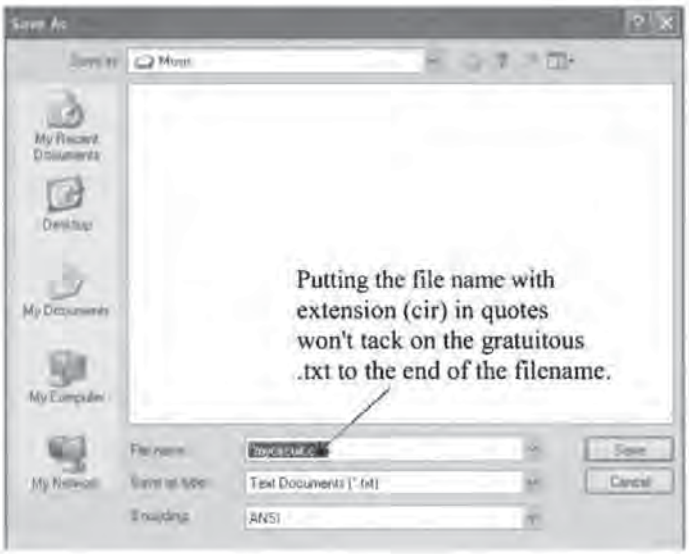



**Figure 1.9** Saving a text file with a ".cir" extension.

IPR2023-00697
Theta EX2019

*Operating Point*

The first SPICE simulation analysis we'll look at is the .op or operating point analysis. An operating point simulation's output data is not graphical but rather simply a list of node voltages, loop currents, and, when active elements are used, small-signal AC parameters. Consider the schematic seen in Fig. 1.10. The SPICE netlist used to simulate this circuit may look like the following (again, remember, that all of these simulation examples are available for download at CMOSedu.com):

```
*** Figure 1.10 CMOS: Circuit Design, Layout, and Simulation ***

*#destroy all
*#run
*#print all

.op

Vin    1     0     DC    1
R1     1     2     1k
R2     2     0     2k

.end
```

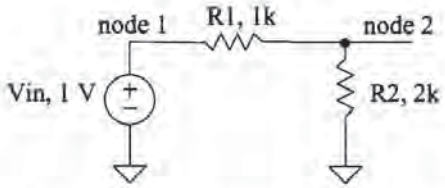



**Figure 1.10** Operation point simulation for a resistive divider.

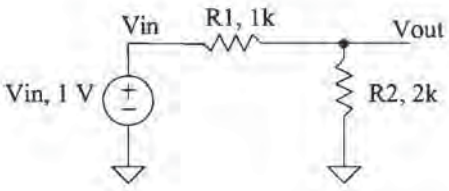
The first line in a netlist is a title line. SPICE ignores the first line (important to avoid frustration!). A comment line starts with an asterisk. SPICE ignores lines that start with a * (in most cases). In the netlist above, however, the lines that start with *# are command lines. These command lines are used for control in some SPICE simulation programs. In other SPICE programs, these lines are simply ignored. The commands in this netlist destroy previous simulation data (so we don't view the old data), run the simulation, and then print the simulation output data. SPICE analysis commands start with a period. Here we are performing an operating point analysis. Following the .op, we've specified an input voltage source called Vin (voltage source names must start with a V, resistor names must start with an R, etc.). connected from node 1 to ground (ground always has a node name of 0 [zero]). We then have a 1k resistor from node 1 to node 2 and a 2k resistor from node 2 to ground. Running the simulation gives the following output:

```
v(1) = 1.000000e+00
v(2) = 6.666667e-01
vin#branch = -3.33333e-04
```

The node voltages, as we would expect, are 1 V and 667 mV, respectively. The current flowing through Vin is 333 μA. Note that SPICE defines positive current flow as from the + terminal of the voltage source to the – terminal (hence, the current above is negative).

IPR2023-00697
Theta EX2019

It's often useful to use names for nodes that have meaning. In Fig. 1.11, we replaced the names node 1 and 2 with Vin and Vout. Vin corresponds to the input voltage source's name. This is useful when looking at a large amount of data. Also seen in Fig. 1.11 is the modified netlist.




```
*** Figure 1.11 CMOS ***
*#destroy all
*#run
*#print all
.op
Vin Vin 0 DC 1
R1 Vin Vout 1k
R2 Vout 0 2k
.end
```

**Figure 1.11** Operation point simulation for a resistive divider.

*Transfer Function Analysis*

The transfer function analysis can be used to find the DC input and output resistances of a circuit as well as the DC transfer characteristics. To give an example, let's replace, in the netlist seen above, .op with

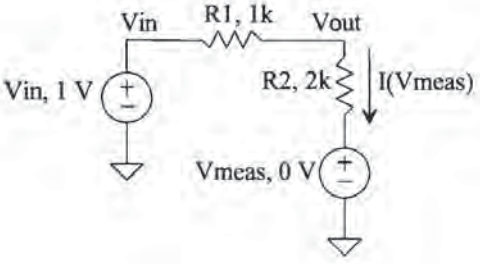
```
.TF     V(Vout,0)       Vin
```

The output is defined as the voltage between nodes Vout and 0 (ground). The input is a source (here a voltage source). When we run the simulation with this command line, we get an output of

```
transfer_function = 6.666667e-01
output_impedance_at_v(vout,0) = 6.666667e+02
vin#input_impedance = 3.000000e+03
```

As expected, the "gain" of this voltage divider is 2/3, the input resistance is 3k (1k + 2k), and the output resistance is 667 Ω (1k||2k).

As another example of the use of the .tf command consider adding the 0 V voltage source to Fig. 1.11, as seen in Fig. 1.12. Adding a 0 V source to a circuit is a common method to measure the current in an element (we plot or print I(Vmeas) for example).




```
*** Figure 1.12 CMOS ***
*#destroy all
*#run
*#print all
.TF I(Vmeas) Vin
Vin Vin 0 DC 1
R1 Vin Vout 1k
R2 Vout Vmeas 2k
Vmeas Vmeas 0 DC 0
.end
```

**Figure 1.12** Measuring the transfer function in a resistive divider when the output variable is the current through R2 and the input is Vin.

IPR2023-00697
Theta EX2019

Here, in the .tf analysis, we have defined the output variable as a current, I(Vmeas) and the input as the voltage, Vin. Running the simulation, we get an output of

```
transfer_function = 3.333333e-04
vin#input_impedance = 3.000000e+03
vmeas#output_impedance = 1.000000e+20
```

The gain is I(Vmeas)/Vin or 1/3k (= 333 μmhos), the input resistance is still 3k, and the output resistance is now an open (Vmeas is removed from the circuit).

*The Voltage-Controlled Voltage Source*

SPICE can be used to model voltage-controlled voltage sources (VCVS). Consider the circuit seen in Fig. 1.13. The specification for a VCVS starts with an E in SPICE. The netlist for this circuit is

```
*** Figure 1.13 CMOS: Circuit Design, Layout, and Simulation ***

*#destroy all
*#run
*#print all

.TF     V(Vout,0) Vin

Vin     Vin     0       DC      1
R1      Vb      0       3k
R2      Vt      Vout    1k
R3      Vout    0       2k
E1      Vt      Vb      Vin     0       23

.end
```

The first two nodes (Vt and Vb), following the VCVS name E1, are the VCVS outputs (the first node is the + output). The second two nodes (Vin and ground) are the controlling nodes. The gain of the VCVS is, in this example, 23. The voltage between Vt and Vb is 23·Vin. Running this simulation gives an output of

```
transfer_function = 7.666667e+00
output_impedance_at_v(vout,0) = 1.333333e+03
vin#input_impedance = 1.000000e+20
```

Notice that the input resistance is infinite.

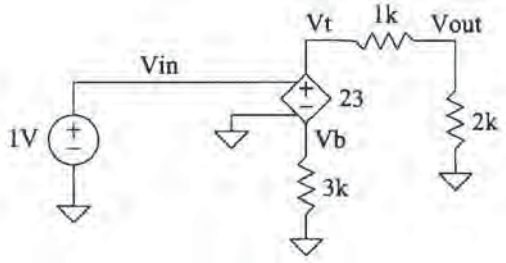
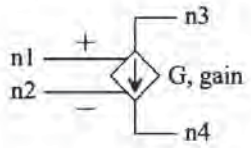



**Figure 1.13** Example using a voltage-controlled voltage source.

IPR2023-00697
Theta EX2019

*An Ideal Op-Amp*

We can implement a (near) ideal op-amp in SPICE with a VCVS or with a voltage-controlled current source (VCCS), Fig. 1.14. It turns out that using a VCCS to implement an op-amp in SPICE results, in general, in better simulation convergence. The input voltage, the difference between nodes n1 and n2 in Fig. 1.14, is multiplied by the transconductance G (units of amps/volts or mhos) to cause a current to flow between n3 and n4. Note that the input resistance of the VCCS, the resistance seen at n1 and n2, is infinite.




Voltage-Controlled Current Source (VCCS)
G1 n3 n4 n1 n2 G

**Figure 1.14** Voltage-controlled current source in SPICE.

Figure 1.15 shows the implementation of an ideal op-amp in SPICE along with an example circuit. The open-loop gain of the op-amp is a million (the product of the VCCS's transconductance with the 1-ohm resistor). Note how we've flipped the polarity of the (SPICE model of the) op-amp's input to ensure a rising voltage on the noninverting input (+ input) causes Vout to increase. The closed-loop gain is –3 (if this isn't obvious then the reader should revisit sophomore circuits before going too much further in the book).

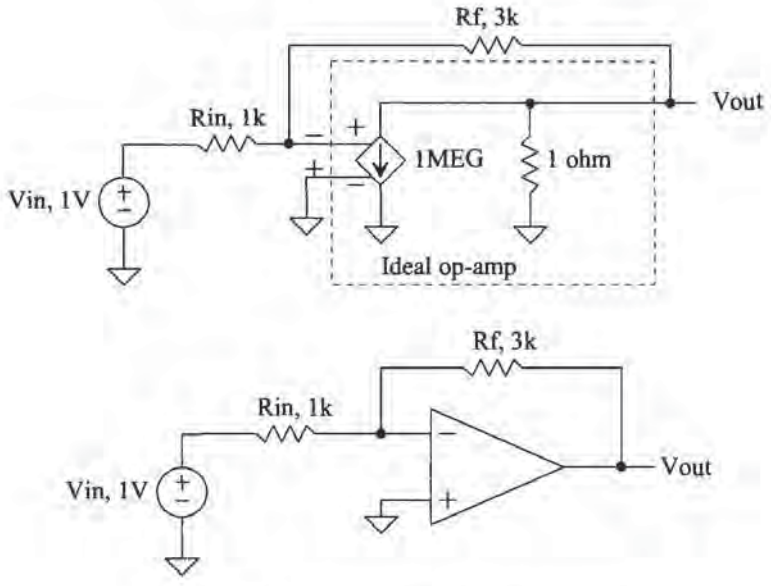



**Figure 1.15** An op-amp simulation example.

IPR2023-00697
Theta EX2019

*The Subcircuit*

In a simulation we may want to use a circuit, like an op-amp, more than once. In these situations we can generate a subcircuit and then, in the main part of the netlist, call the circuit as needed. Below is the netlist for simulating, using a transfer function analysis, the circuit in Fig. 1.15 where the op-amp is specified using a subcircuit call.

```
*** Figure 1.15 CMOS: Circuit Design, Layout, and Simulation ***

*#destroy all
*#run
*#print all

.TF     V(Vout,0) Vin

Vin     Vin     0       DC      1
Rin     Vin     Vm      1k
Rf      Vout    Vm      3k

X1      Vout    0       vm      Ideal_op_amp

.subckt Ideal_op_amp    Vout    Vp      Vm
G1      Vout    0       Vm      Vp      1MEG
RL      Vout    0       1
.ends
.end
```

Notice that a subcircuit call begins with the letter X. Note also how we've called the noninverting input (the + input) Vp and not V+ or +. Some SPICE simulators don't like + or – symbols used in a node's name. Further note that a subcircuit ends with .ends (end subckt). Care must be exercised with using either .end or .ends. If, for example, a .end is placed in the middle of the netlist all of the SPICE netlist information following this .end is ignored.

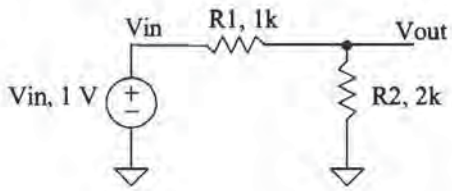
The output results for this simulation are seen below. Note how the ideal gain is –3 where the simulated gain is –2.99999. Our near-ideal op-amp has an open-loop gain of one million and thus the reason for the slight discrepancy between the simulated and calculated gains. Also note how the input resistance is 1k, and the output resistance, because of the feedback, is essentially zero.

```
transfer_function = -2.99999e+00
output_impedance_at_v(vout,0) = 3.999984e-06
vin#input_impedance = 1.000003e+03
```

*DC Analysis*

In both the operating point and transfer function analyses, the input to the circuit was constant. In a DC analysis, the input is varied and the circuit's node voltages and currents (through voltage sources) are simulated. A simple example is seen in Fig. 1.16. Note how we are now plotting, instead of printing, the node voltages. We could also plot the current through Vin (plot Vin#branch). The .dc command specifies that the input source, Vin, should be varied from 0 to 1 V in 1 mV steps. The x-axis of the simulation results seen in the figure is the variable we are sweeping, here Vin. Note that, as expected, the slope of the Vin curve is one (of course) and the slope of Vout is 2/3 (= Vout/Vin).

IPR2023-00697
Theta EX2019




```
*** Figure 1.16 CMOS ***
*#destroy all
*#run
*#plot Vin Vout
.dc Vin 0 1 1m
Vin Vin 0 DC 1
R1 Vin Vout 1k
R2 Vout 0 2k
.end
```

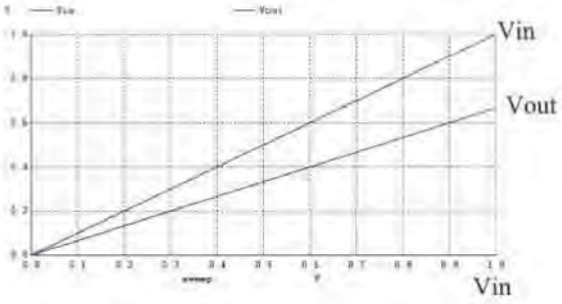
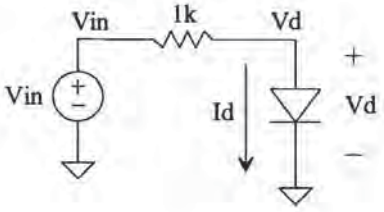



**Figure 1.16** DC analysis simulation for a resistive divider.

*Plotting IV Curves*

One of the simulations that is commonly performed using a DC analysis is plotting the current-voltage (IV) curves for an active device (e.g., diode or transistor). Examine the simulation seen in Fig. 1.17. The diode is named D1. (Diodes must have names that start with a D.) The diode's anode is connected to node Vd, while its cathode is connected to




```
*** Figure 1.17 CMOS ***
*#destroy all
*#run
*#let ID=-Vin#branch
*#plot ID
.dc Vin 0 1 1m
Vin Vin 0 DC 1
R1 Vin Vd 1k
D1 Vd 0 mydiode
.model mydiode D
.end
```

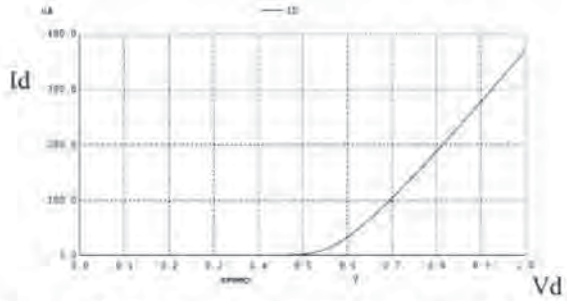



**Figure 1.17** Plotting the current-voltage curve for a diode.

IPR2023-00697
Theta EX2019

ground. This is our first introduction to the .model specification. Here our diode's model name is mydiode. The .model parameter D seen in the netlist simply indicates a diode model. We don't have any parameters after the D in this simulation, so SPICE uses default parameters. The interested reader is referred to Table 2.1 on page 47 for additional information concerning modeling diodes in SPICE. Note, again, that SPICE defines positive current through a voltage source as flowing from the + terminal to the – terminal (hence why we defined the diode current the way we did in the netlist).

*Dual Loop DC Analysis*

An outer loop can be added to a DC analysis, Fig. 1.18. In this simulation we start out by setting the base current to 5 μA and sweeping the collector-emitter voltage from 0 to 5 V in 1 mV steps. The output data for this particular simulation is the trace, seen in Fig. 1.18, with a label of "Ib=5u." The base current is then increased by 5 μA to 10 μA, and the collector-emitter voltage is stepped again (resulting in the trace labeled "Ib=10u". This continues until the final iteration when Ib is 25 μA. Other examples of using a dual-loop DC analysis for MOSFET IV curves are found in Figs. 6.11, 6.12, and 6.13.

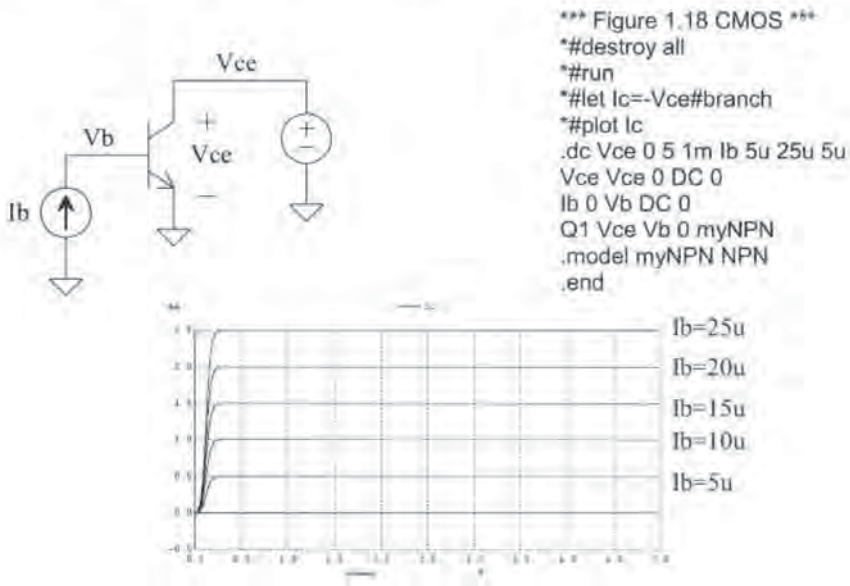



**Figure 1.18** Plotting the current-voltage curves for an NPN BJT.

*Transient Analysis*

The form of the transient analysis statement is

```
.tran tstep tstop <tstart> <tmax> <uic>
```

where the terms in < > are optional. The tstep term indicates the (suggested) time step to be used in the simulation. The parameter tstop indicates the simulation's stop time. The starting time of a simulation is always time equals zero. However, for very large (data) simulations, we can specify a time to start saving data, tstart. The tmax parameter is used to specify the maximum step size. If the plots start to look jagged (like a sinewave that isn't smooth), then tmax should be reduced.

IPR2023-00697
Theta EX2019

A SPICE transient analysis simulates circuits in the time domain (as in an oscilloscope, the x-axis is time). Let's simulate, using a transient analysis, the simple circuit seen back in Fig. 1.11. A simulation netlist may look like (see output in Fig. 1.19):

```
*** Figure 1.19 CMOS: Circuit Design, Layout, and Simulation ***

*#destroy all
*#run
*#plot vin vout

.tran 100p 100n

Vin     Vin     0       DC      1
R1      Vin     Vout    1k
R2      Vout    0       2k

.end
```

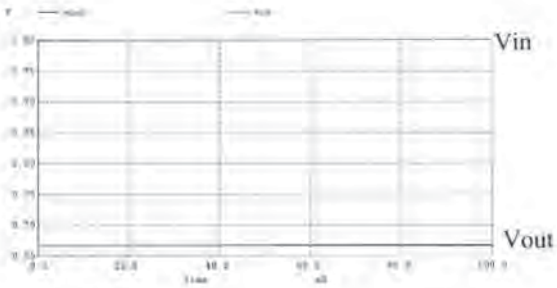



**Figure 1.19** Transient simulation for the circuit in Fig. 1.11.

*The SIN Source*

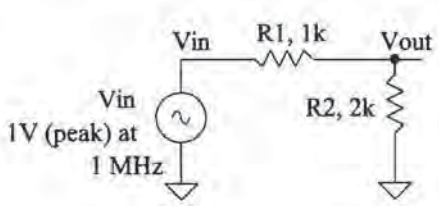
To illustrate a simulation using a sinewave, examine the schematic in Fig 1.20. The statement for a sinewave in SPICE is

```
SIN Vo Va freq <td> <theta>
```

The parameter Vo is the sinusoid's offset (the DC voltage in series with the sinewave). The parameter Va is the peak amplitude of the sinewave. Freq is the frequency of the sinewave, while td is the delay before the sinewave starts in the simulation. Finally, theta is used if the amplitude of the sinusoid has a damped nature. Figure 1.20 shows the netlist corresponding to the circuit seen in this figure and the simulation results.

Some key things to note in this simulation: (1) MEG is used to specify $10^6$. Using "m" or "M" indicates milli or $10^{-3}$. The parameter 1MHz indicates 1 milliHertz. Also, f indicates femto or $10^{-15}$. A capacitor value of 1f doesn't indicate one Farad but rather 1 femto Farad. (2) Note how we increased the simulation time to 3 μs. If we had a simulation time of 100 ns (as in the previous simulation), we wouldn't see much of the sinewave (one-tenth of the sinewave's period). (3) The "SIN" statement is used in a transient simulation analysis. The SIN specification is **not** used in an AC analysis (discussed later).

IPR2023-00697
Theta EX2019




```
*** Figure 1.20 ***
*#destroy all
*#run
*#plot vin vout
.tran 1n 3u
Vin Vin 0 DC 0 SIN 0 1 1MEG
R1 Vin Vout 1k
R2 Vout 0 2k
.end
```

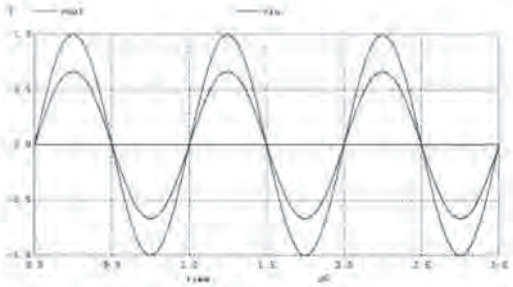


**Figure 1.20** Simulating a resistive divider with a sinusoidal input.

*An RC Circuit Example*

To illustrate the use of a .tran simulation let's determine the output of the RC circuit seen in Fig. 1.21 and compare our hand calculations to simulation results. The output voltage can be written in terms of the input voltage by

$$V_{out} = V_{in} \cdot \frac{1/j\omega C}{1/j\omega C + R} \text{ or } \frac{V_{out}}{V_{in}} = \frac{1}{1 + j\omega RC} \tag{1.1}$$

Taking the magnitude of this equation gives

$$\left|\frac{V_{out}}{V_{in}}\right| = \frac{1}{\sqrt{1 + (2\pi fRC)^2}} \tag{1.2}$$

and taking the phase gives

$$\angle\frac{V_{out}}{V_{in}} = -\tan^{-1}\frac{2\pi fRC}{1} \tag{1.3}$$

From the schematic the resistance is 1k, the capacitance is 1 μF, and the frequency is 200 Hz. Plugging these numbers into Eqs. (1.1) – (1.3) gives $\left|\frac{V_{out}}{V_{in}}\right| = 0.623$ and $\angle\frac{V_{out}}{V_{in}} = -0.898$ radians or −51.5 degrees. With a 1 V peak input then our output voltage is 623 mV (and as seen in Fig. 1.21, it is). Remembering that phase shift is simply an indication of time delay at a particular frequency,

$$\angle\,(\text{radians}) = \frac{t_d}{T} \cdot 2\pi \text{ or } \angle\,(\text{degrees}) = \frac{t_d}{T} \cdot 360 = t_d \cdot f \cdot 360 \tag{1.4}$$

The way to remember this equation is that the time delay, $t_d$, is a percentage of the period ($T$), $t_d/T$, multiplied by either 2π (radians) or 360 (degrees). For the present example, the time delay is 715 μs (again, see Fig. 1.21). Note that the minus sign indicates that the output is lagging (occurring later in time) the input (the input leads the output).

IPR2023-00697
Theta EX2019

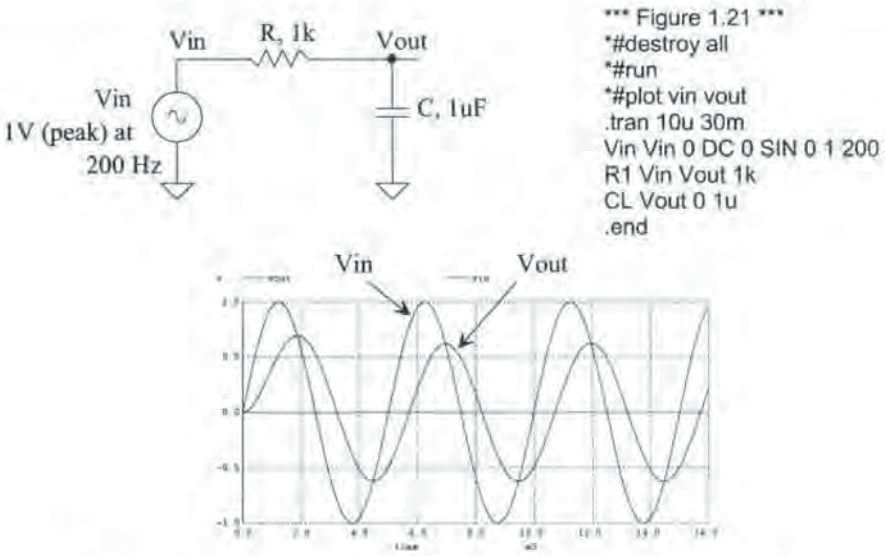



**Figure 1.21** Simulating the operation of an RC circuit using a .tran analysis.

*Another RC Circuit Example*

As one more example of simulating the operation of an RC circuit consider the circuit seen in Fig. 1.22. Combining the impedances of C1 and R, we get

$$Z = \frac{R/j\omega C_1}{R + 1/j\omega C_1} = \frac{R}{1 + j\omega RC_1} \tag{1.5}$$

The transfer function for this circuit is then

$$\frac{V_{out}}{V_{in}} = \frac{1/j\omega C_2}{1/j\omega C_2 + Z} = \frac{1 + j\omega RC_1}{1 + j\omega R(C_1 + C_2)} \tag{1.6}$$

The magnitude of this transfer function is

$$\left|\frac{V_{out}}{V_{in}}\right| = \frac{\sqrt{1 + (2\pi fRC_1)^2}}{\sqrt{1 + (2\pi fR \cdot (C_1 + C_2))^2}} \tag{1.7}$$

and the phase response is

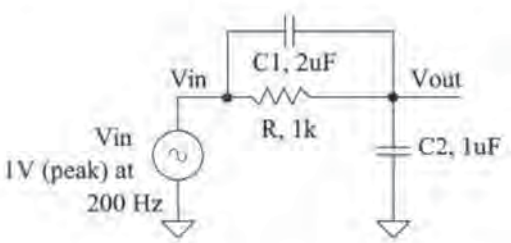
$$\angle\frac{V_{out}}{V_{in}} = \tan^{-1}\frac{2\pi fRC_1}{1} - \tan^{-1}\frac{2\pi fR(C_1 + C_2)}{1} \tag{1.8}$$

Plugging in the numbers from the schematic gives a magnitude response of 0.6 (which matches the simulation results) and a phase shift of – 0.119 radians or – 6.82 degrees. The amount of time the output is lagging the input is then

$$t_d = \frac{T \cdot \angle}{360} = \frac{\angle}{f \cdot 360} = \frac{-6.82}{200 \cdot 360} = -95\ \mu s \tag{1.9}$$

which is confirmed with the simulation results seen in Fig. 1.22.

IPR2023-00697
Theta EX2019




```
*** Figure 1.22 ***
*#destroy all
*#run
*#plot vin vout
.tran 10u 30m
Vin Vin 0 DC 0 SIN 0 1 200
R1 Vin Vout 1k
C1 Vin Vout 2u
C2 Vout 0 1u
.end
```

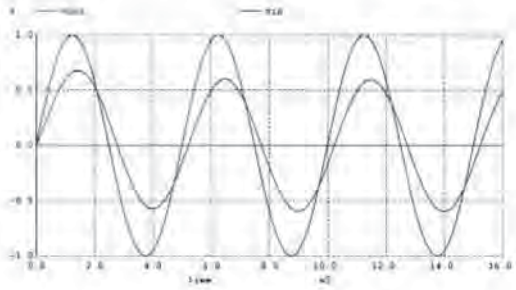


**Figure 1.22** Another RC circuit example.

*AC Analysis*

When performing a transient analysis (.tran) the x-axis is time. We can determine the frequency response of a circuit (the x-axis is frequency) using an AC analysis (.ac). An AC analysis is specified in SPICE using

```
.ac dec nd fstart fstop
```

The dec indicates that the x-axis should be plotted in decades. We could replace dec with lin (linear plot on the x-axis) or oct (octave). The term nd indicates the number of points per decade (say 100), while fstart and fstop indicate the start and stop frequencies (note that fstart cannot be zero, or DC, since this isn't an AC signal). The netlist used to simulate the AC response of the circuit in Fig. 1.21 follows. The simulation output is seen in Fig. 1.23, where we've pointed out the response at 200 Hz (the frequency used in Fig. 1.21 and used for calculations on page 17).

```
*** Figure 1.23 CMOS: Circuit Design, Layout, and Simulation ***

*#destroy all
*#run
*#plot db(vout/vin)
*#set units=degrees
*#plot ph(vout/vin)

.ac dec 100 1 10k

Vin     Vin     0       DC      0       SIN 0 1 200     AC 1
R1      Vin     Vout    1k
CL      Vout    0       1u

.end
```

IPR2023-00697
Theta EX2019

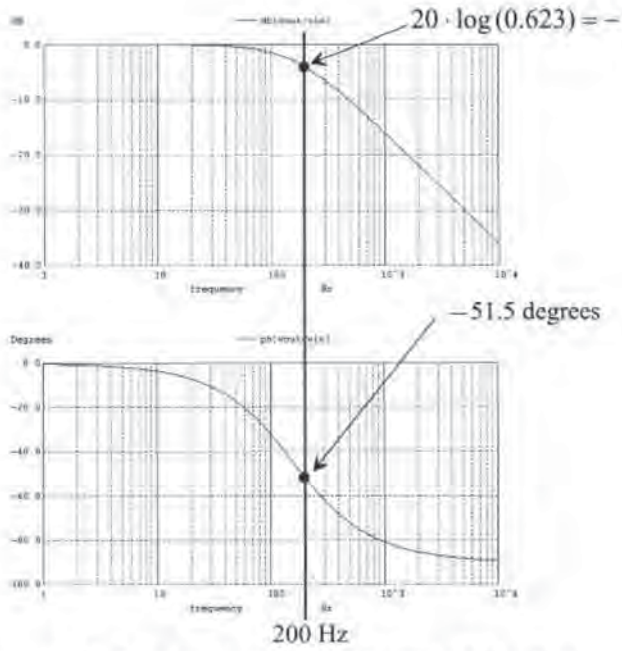



**Figure 1.23** AC simulation for the RC circuit in Fig. 1.21.

Note in this netlist that the SIN specification in Vin has nothing to do with an AC analysis (it's ignored for an AC analysis). For the AC analysis, we added, to the statement for Vin, the term AC 1 (specifying that the magnitude or peak of the AC signal is 1). We can add a phase shift of 45 degrees by using AC 1 45 in the statement.

*Decades and Octaves*

In the simulation results seen in Fig. 1.23 we used decades. When we talk about decades we either are multiplying or dividing by 10. One decade above 23 MHz is 230 MHz, while one decade below 1.2 kHz is 120 Hz.

When we talk about octaves, we talk about either multiplying or dividing by 2. One octave above 23 MHz is 46 MHz while one octave below 1.2 kHz is 600 Hz. Two octaves above 23 MHz is (multiply by 4) 92 MHz.

*Decibels*

When the magnitude response of a transfer function decreases by 10, it is said it goes down by −20 dB (divide by 10, $20 \cdot \log(0.1) = -20\,dB$). When the magnitude response increases by 10, it goes up by 20 dB (multiply by 10). For the frequency response in Fig. 1.23 (above 159 Hz, the −3 dB frequency, or here when the magnitude response is 0.707), the response is rolling off at −20 dB/decade. What this means is that if we increase the frequency by 10 the magnitude response decreases by 10. We could also say the response is rolling off at −6 dB/octave above 159 Hz (for every increase in frequency by 2 the magnitude response drops by a factor of 2). If a magnitude response is rolling off at −40 dB/decade, then for every increase in frequency by 10 the magnitude drops by 100. Similarly if a response rolls off at −12 dB/octave, for every doubling in frequency our response drops by 4. Note that −6 dB/octave is the same rate as −20 dB/decade.

IPR2023-00697
Theta EX2019

*Pulse Statement*

The SPICE pulse statement is used in transient simulations to specify pulses or clock signals. This statement has a format given by

```
pulse vinit vfinal td tr tf pw per
```

The pulse's initial voltage is vinit while vfinal is the pulse's final (or pulsed) value, td is the delay before the pulse starts, tr and tf are the rise and fall times, respectively, of the pulse (noting that when these are set to zero the step size used in the transient simulation is used), pw is the pulse's width; and per is the period of the pulse. Figure 1.24 provides an example of a simulation that uses the pulse statement. A section of the netlist used to generate the waveforms in this figure follows.

```
.tran 100p 30n

Vin    Vin    0      DC    0      pulse 0 1 6n 0 0 3n 10n
R1     Vin    Vout   1k
C1     Vout   0      1p
```

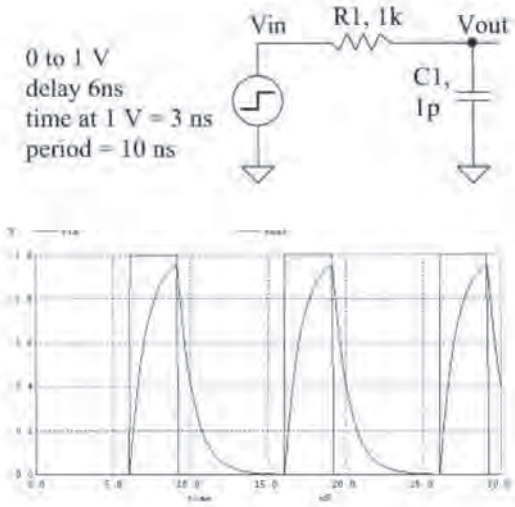



**Figure 1.24** Simulating the step response of an RC circuit using a pulsed source voltage.

*Finite Pulse Rise Time*

Notice, in the simulation results seen in Fig. 1.24, that the rise and fall times of the input pulse are not 0 as specified in the pulse statement but rather 100 ps as specified by the suggested maximum step size in the .tran statement. Figure 1.25 shows the simulation results if we change the pulse statement to

```
Vin    Vin    0      DC    0      pulse 0 1 6n 10p 10p 3n 10n
```

where we've specified 10 ps rise and fall times. Note that in some SPICE simulators you must specify a maximum step size in the .tran statement. You could do this in the .tran statement above by using .tran 10p 30n 0 **10p** (where the 10p is the maximum step size and the simulation starts saving data at 0.)

IPR2023-00697
Theta EX2019

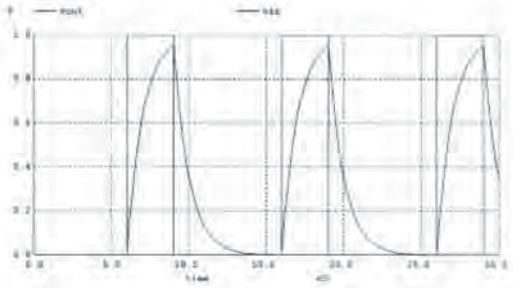


**Figure 1.25** Specifying a rise time in the pulse statement to avoid slow rise times (rise times set by the maximum step size in the .tran statement.)

*Step Response*

The pulse statement can also be used to generate a step function

```
Vin Vin 0 DC 0 pulse 0 1 2n 10p
```

We've reduced the delay to 2n and have specified (only) a rise time for the pulse. Since the pulse width isn't specified, the pulse transitions and then stays high for the extent of the simulation. Figure 1.26 shows the step response for the RC circuit seen in Fig. 1.24.

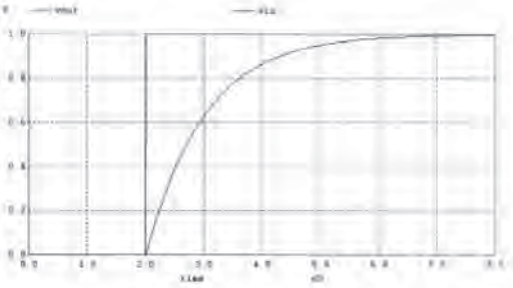


**Figure 1.26** Step response of an RC circuit.

*Delay and Rise Time in RC Circuits*

From the RC circuit review on page 50 we can write the delay time, the time it takes the pulse to reach 50% of its final value in an RC circuit, using

$$t_d \approx 0.7RC \tag{1.10}$$

and the rise time (or fall time) as

$$t_r \approx 2.2RC \tag{1.11}$$

Using the RC in Fig. 1.24 (1 ns), we get a (calculated) delay time of 700 ps and a rise time of 2.2 ns. These numbers are verified in Fig. 1.26. To show that the pulse statement can be used for other amplitude steps consider resimulating the circuit in Fig. 1.24 (see Fig. 1.27) with an input pulse that transitions from –1 to –2 V (note how the delay and transition times remain unchanged. The SPICE pulse statement is now

```
Vin Vin 0 DC 0 pulse -1 -2 2n 10p
```

IPR2023-00697
Theta EX2019

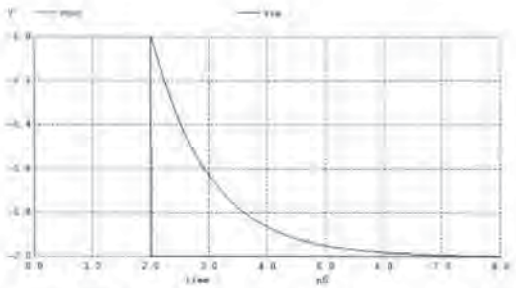


**Figure 1.27** Another step response (negative going) of an RC circuit.

*Piece-Wise Linear (PWL) Source*

The piece-wise linear (PWL) source specifies arbitrary waveform shapes. The SPICE statement for a PWL source is

```
pwl t1 v1 t2 v2 t3 v3 ... <rep>
```

To provide an example using a PWL voltage source, examine Fig. 1.28. The input waveform in this simulation is specified using

```
pwl 0 0.5 3n 1 5n 1 5.5n 0 7n 0
```

At 0 ns, the input voltage is 0.5 V. At 3 ns the input voltage is 1 V. Note the linear change between 0 and 3 ns. Each pair of numbers, the first the time and the second the voltage (or current if a current source is used) represent a point on the PWL waveform. Note that in some simulators the specification for a PWL source may be quite long. In these situations a text file is specified that contains the PWL for the simulation.

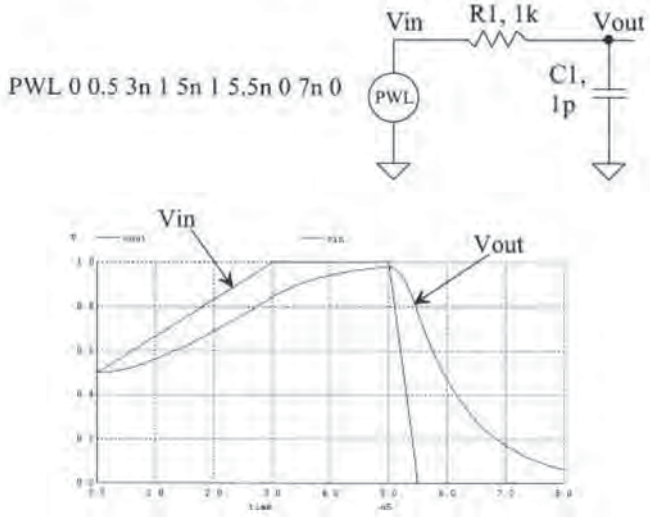



**Figure 1.28** Using a PWL source to drive an RC circuit.

IPR2023-00697
Theta EX2019

*Simulating Switches*

A switch can be simulated in SPICE using the following (for example) syntax

```
s1 node1 node2 controlp controlm switmod
.model switmod sw ron=1k
```

The name of a switch must start with an s. The switch is connected between node1 and node2, as seen in Fig. 1.29. When the voltage on node controlp is greater than the voltage on node controlm, the switch closes. The switch is modeled using the .model statement. As seen above, we are setting the series resistance of the switch to 1k.

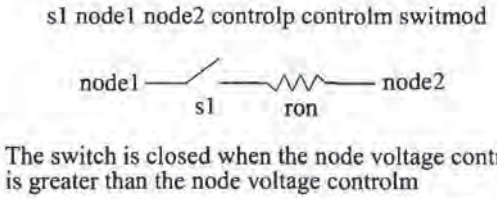
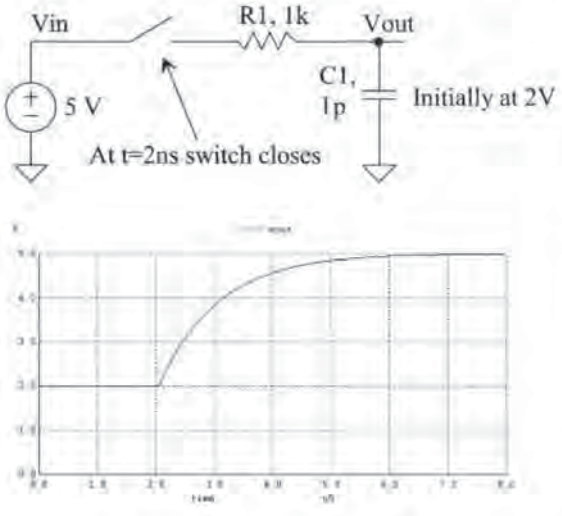



**Figure 1.29** Modeling a switch in SPICE.

*Initial Conditions on a Capacitor*

An example of a circuit that uses both a switch and an initial voltage on a capacitor is seen in Fig. 1.30. Notice, in the netlist, that we have added UIC to the end of the .tran statement. This addition makes SPICE "use initial conditions" or skip an initial operating point calculation. Also note that to set the initial voltage across the capacitor we simply added IC=2 to the end of the statement for a capacitor. To set a node to a voltage (that may have a capacitor connected to it or not), we can add, for example,

```
.ic v(vout)=2
```




```
*** Figure 1.30 ***

*#destroy all
*#run
*#plot vout

.tran 100p 8n UIC

Vclk clk 0 pulse -1 1 2n
Vin Vin 0 DC 5
S1 Vin Vouts clk 0 switmodel
R1 Vouts Vout 1k
C1 Vout 0 1p IC=2

.model switmodel sw ron=0.1

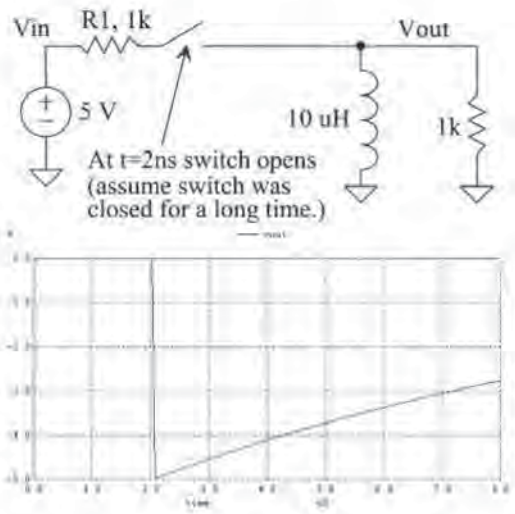
.end
```

**Figure 1.30** Using initial conditions and a switch in an RC circuit simulation.

IPR2023-00697
Theta EX2019

*Initial Conditions in an Inductor*

Consider the circuit seen in Fig. 1.31. Here we assume that the switch has been closed for a long period of time so that the circuit reaches steady-state. The inductor shorts the output to ground and the current flowing in the inductor is 5 mA. To simulate this initial condition, we set the current in the inductor using the IC statement as seen in the netlist (remembering to include the UIC in the .tran statement). At 2 ns after the simulation starts, we open the switch (the control voltage connections are switched from the previous simulations). Since we know we can't change the current through an inductor instantaneously (the inductor wants to keep pulling 5 mA), the voltage across the inductor will go from 0 to −5 V. The inductor will pull the 5 mA of current through the 1k resistor connected to the output node. Note that we select the transient simulation time by looking at the time constant, $L/R$, of the circuit (here 10 ns).




```
*** Figure 1.31 ***

*#destroy all
*#run
*#plot vout

.tran 100p 8n UIC

Vclk clk 0 pulse -1 1 2n
Vin Vin 0 DC 5
S1 Vin Vouts 0 clk switmodel
R1 Vouts Vout 1k
R2 Vout 0 1k
L1 Vout 0 10u IC=5m

.model switmodel sw ron=0.1

.end
```

**Figure 1.31** Using initial conditions in an inductive circuit.

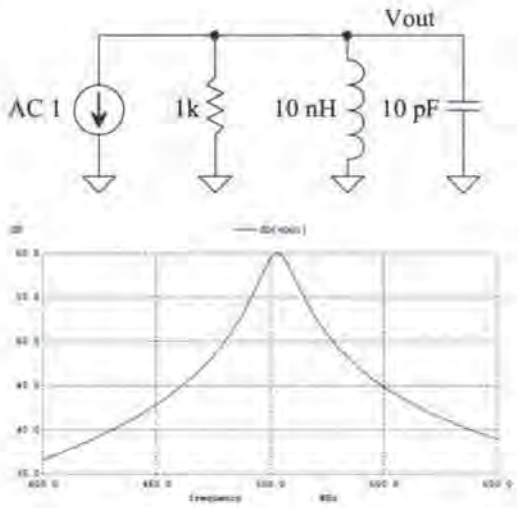
*Q of an LC Tank*

Figure 1.32 shows a simulation useful in determining the quality factor or $Q$ of a parallel LC circuit (a tank, used in communication circuits among others). The current source and resistor may model a transistor. The resistor can also be used to model the losses in the capacitor or inductor. Quality factor for a resonant circuit is defined as the ratio of the energy stored in the tank to the energy lost. Our circuit definition for $Q$ is the ratio of the center (resonant) frequency to the bandwidth of the response at the 3 dB points. We can write an equation for this circuit definition of $Q$ as

$$Q = \frac{f_{center}}{BW} = \frac{f_{center}}{f_{3dBhigh} - f_{3dBlow}} \tag{1.12}$$

The center frequency of the circuit in Fig. 1.32 is roughly 503 MHz, while the upper 3 dB frequency is 511.2 MHz and the lower 3 dB frequency is 494.8 MHz. The $Q$ is roughly 30. Note the use of linear plotting in the ac analysis statement.

IPR2023-00697
Theta EX2019




```
*** Figure 1.32 ***

*#destroy all
*#run
*#plot db(vout)

.AC lin 100 400MEG 600MEG

Iin Vout 0 DC 0 AC 1
R1 Vout 0 1k
L1 Vout 0 10n
C1 Vout 0 10p

.end
```

**Figure 1.32** Determining the Q, or quality factor, of an LC tank.

*Frequency Response of an Ideal Integrator*

The frequency response of the integrator seen in Fig. 1.33 can be determined knowing the op-amp keeps the inverting input terminal at the same potential as the non-inverting input (here ground). The current through the resistor must equal the current through the capacitor so

$$\frac{V_{in}}{R} + \frac{V_{out}}{1/j\omega C} = 0 \tag{1.13}$$

or

$$\frac{V_{out}}{V_{in}} = \frac{-1}{j\omega RC} = \frac{-(1 + j \cdot 0)}{0 + j\omega RC} \tag{1.14}$$

The magnitude of the integrator's transfer function is

$$\left|\frac{V_{out}}{V_{in}}\right| = \frac{\sqrt{(-1)^2 + (-0)^2}}{\sqrt{(0)^2 + (2\pi fRC)^2}} = \frac{1}{2\pi RCf} \tag{1.15}$$

while the phase shift through the integrator is

$$\angle\frac{V_{out}}{V_{in}} = \tan^{-1}\frac{-0}{-1} - \tan^{-1}\frac{2\pi RCf}{0} = -90° \tag{1.16}$$

Note that the gain of the integrator approaches infinity as the frequency decreases towards DC while the phase shift is constant.

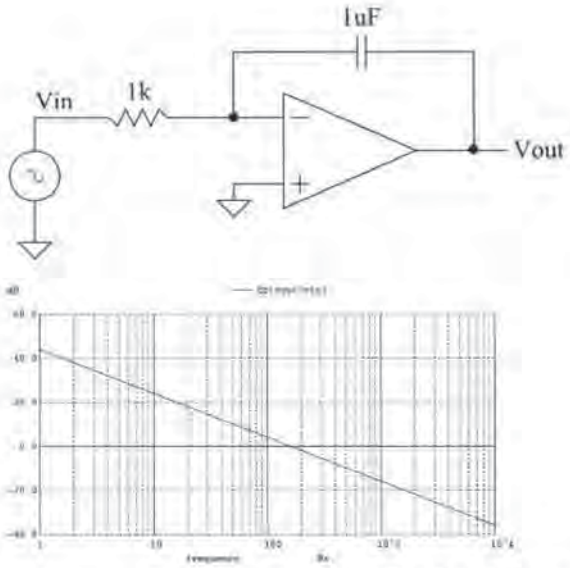
*Unity-Gain Frequency*

It's of interest to determine the frequency where the magnitude of the transfer function is unity (called the unity-gain frequency, $f_{un}$.) Using Eq. (1.15), we can write

IPR2023-00697
Theta EX2019

$$\left|\frac{V_{out}}{V_{in}}\right| = 1 = \frac{1}{2\pi RCf_{un}} \rightarrow f_{un} = \frac{1}{2\pi RC} \tag{1.17}$$

Using the values seen in the schematic, the unity-gain frequency is 159 Hz (as verified in the SPICE simulation seen in Fig. 1.33).




```
*** Figure 1.33 ***

*#destroy all
*#run
*#plot db(vout/vin)
*#set units=degrees
*#plot ph(vout/vin)

.ac dec 100 1 10k

Vin Vin 0 DC 1 AC 1
Rin Vin vm 1k
Cf Vout vm 1u

X1 Vout 0 vm Ideal_op_amp
.subckt Ideal_op_amp Vout Vp Vm
G1 Vout 0 Vm Vp 1MEG
RL Vout 0 1
.ends
.end
```

**Figure 1.33** An integrator example.

*Time-Domain Behavior of the Integrator*

The time-domain behavior of the integrator can be characterized, again, by equating the current in the resistor with the current in the capacitor

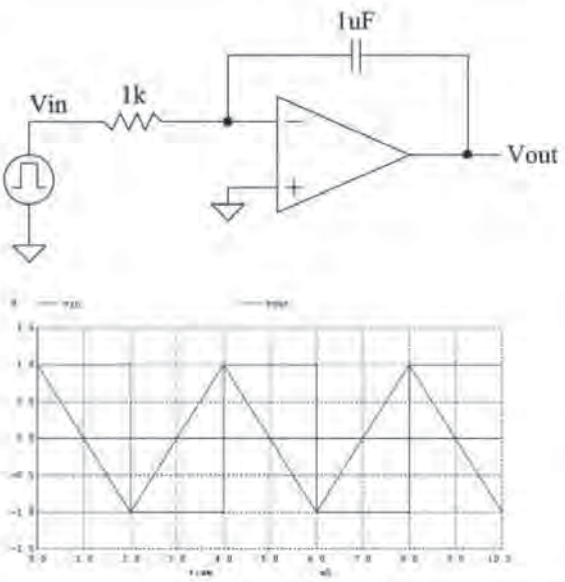
$$V_{out} = \frac{1}{C}\int \frac{V_{in}}{R} \cdot dt \tag{1.18}$$

If our input is a constant voltage, then the output is a linear ramp increasing (if the input is negative) or decreasing (if the input is positive) with time. If the input is a squarewave, with zero mean then the output will look like a triangle wave. Using the values seen in Fig. 1.33 for the time-domain simulation seen in Fig. 1.34, we can estimate that if a 1 V signal is applied to the integrator the output voltage will have a slope of

$$V_{out}(t) = \frac{V_{in}}{RC} = \frac{1}{1\ ms} \tag{1.19}$$

or 1 V/ms slope. This equation can be used to design a sawtooth waveform generator from an input squarewave. Note, however, there are several practical concerns. To begin, we set the output of the integrator, using the .ic statement, to ground at the beginning of the simulation. In a real circuit this may be challenging (one method is to add a reset switch across the capacitor). Another issue, discussed later in the book, is the op-amp's offset voltage. This will cause the outputs to move towards the power supply rails even with no input applied. Finally, notice that putting a + in the first column treats the SPICE code as if it were continued from the previous line.

IPR2023-00697
Theta EX2019




```
*** Figure 1.34 ***

*#destroy all
*#run
*#plot vout vin

.tran 10u 10m
.ic v(vout)=0

Vin Vin 0 DC 1
+ pulse -1 1 0 1u 1u 2m 4m
Rin Vin vm 1k
Cf Vout vm 1u

X1 Vout 0 vm Ideal_op_amp
.subckt Ideal_op_amp Vout Vp Vm
G1 Vout 0 Vm Vp 1MEG
RL Vout 0 1
.ends
.end
```

**Figure 1.34** Time-domain integrator example.

*Convergence*

A netlist that doesn't simulate isn't converging numerically. *Assuming* that the circuit contains no connection errors, there are basically three parameters that can be adjusted to help convergence: ABSTOL, VNTOL, and RELTOL.

ABSTOL is the absolute current tolerance. Its default value is 1 pA. This means that when a simulated circuit gets within 1 pA of its "actual" value, SPICE assumes that the current has converged and moves onto the next time step or AC/DC value. VNTOL is the node voltage tolerance, default value of 1 μV. RELTOL is the relative tolerance parameter, default value of 0.001 (0.1 percent). RELTOL is used to avoid problems with simulating large and small electrical values in the same circuit. For example, suppose the default value of RELTOL and VNTOL were used in a simulation where the actual node voltage is 1 V. The RELTOL parameter would signify an end to the simulation when the node voltage was within 1 mV of 1 V (1V·RELTOL), while the VNTOL parameter signifies an end when the node voltage is within 1 μV of 1 V. SPICE uses the larger of the two, in this case the RELTOL parameter results, to signify that the node has converged.

Increasing the value of these three parameters helps speed up the simulation and assists with convergence problems at the price of reduced accuracy. To help with convergence, the following statement can be added to a SPICE netlist:

```
.OPTIONS ABSTOL=1uA VNTOL=1mV RELTOL=0.01
```

To (hopefully) force convergence, these values can be increased to

```
.OPTIONS ABSTOL=1mA VNTOL=100mV RELTOL=0.1
```

IPR2023-00697
Theta EX2019

Note that in some high-gain circuits with feedback (like the op-amp's designed later in the book) decreasing these values can actually help convergence.

*Some Common Mistakes and Helpful Techniques*

The following is a list of helpful techniques for simulating circuits using SPICE.

1. The first line in a SPICE netlist must be a comment line. SPICE ignores the first line in a netlist file.
2. One megaohm is specified using 1MEG, not 1M, 1m, or 1 MEG.
3. One farad is specified by 1, not 1f or 1F. 1F means one femto-Farad or $10^{-15}$ farads.
4. Voltage source names should always be specified with a first letter of V. Current source names should always start with an I.
5. Transient simulations display time data; that is, the x-axis is time. A jagged plot such as a sinewave that looks like a triangle wave or is simply not smooth is the result of not specifying a maximum print step size.
6. Convergence with a transient simulation can usually be helped by adding a UIC (use initial conditions) to the end of a .tran statement.
7. A simulation using MOSFETs must include the scale factor in a .options statement unless the widths and lengths are specified with the actual (final) sizes.
8. In general, the body connection of a PMOS device is connected to *VDD*, and the body connection of an n-channel MOSFET is connected to ground. This is easily checked in the SPICE netlist.
9. Convergence in a DC sweep can often be helped by avoiding the power supply boundaries. For example, sweeping a circuit from 0 to 1 V may not converge, but sweeping from 0.05 to 0.95 will.
10. In any simulation adding .OPTIONS RSHUNT=1E8 (or some other value of resistor) can be used to help convergence. This statement adds a resistor in parallel with every node in the circuit (see the WinSPICE manual for information concerning the GMIN parameter). Using a value too small affects the simulation results.

**ADDITIONAL READING**

[1] R. J. Baker, *CMOS Mixed-Signal Circuit Design*, 2nd ed., John-Wiley and Sons, 2009. ISBN 978-0470290262

[2] S. M. Sandler and C. Hymowitz, *SPICE Circuit Handbook*, McGraw-Hill, 2006. ISBN 978-0071468572

[3] K. Kundert, *The Designer's Guide to SPICE and Spectre*, Springer, 1995. ISBN 978-0792395713

[4] A. Vladimirescu, *The SPICE Book*, John-Wiley and Sons, 1994. ISBN 978-0471609261

[5] F. M. Wanlass, "Low Standby-Power Complementary Field Effect Transistor," US Patent 3,356,858, filed June 18, 1963, and issued December 5, 1967.

IPR2023-00697
Theta EX2019

## PROBLEMS

**1.1** What would happen to the transfer function analysis results for the circuit in Fig. 1.11 if a capacitor were added in series with R1? Why? What about adding a capacitor in series with R2?

**1.2** Resimulate the op-amp circuit in Fig. 1.15 if the open-loop gain is increased to 100 million while, at the same time, the resistor used in the ideal op-amp is increased to 100 Ω. Does the output voltage move closer to the ideal value?

**1.3** Simulate the op-amp circuit in Fig. 1.15 if Vin is varied from −1 to +1V. Verify, with hand calculations, that the simulation output is correct.

**1.4** Regenerate IV curves, as seen in Fig. 1.18, for a PNP transistor.

**1.5** Resimulate the circuit in Fig. 1.20 if the sinewave doesn't start to oscillate until 1 μs after the simulation starts.

**1.6** At what frequency does the output voltage, in Fig. 1.21, become half of the input voltage? Verify your answer with SPICE.

**1.7** Determine the output of the circuit seen in Fig. 1.22 if a 1k resistor is added from the output of the circuit to ground. Verify your hand calculations using SPICE.

**1.8** Using an AC analysis verify the time domain results seen in Fig. 1.22.

**1.9** If the capacitor in Fig. 1.24 is increased to 1 μF simulate, similar to Fig. 1.26 but with a longer time scale, the step response of the circuit. Compare the simulation results to the hand-calculated values using Eqs. (1.10) and (1.11).

**1.10** Using a PWL source (instead of a pulse source), regenerate the simulation data seen in Fig. 1.26.

**1.11** Using the values seen in Fig. 1.32, for the inductor and capacitor determine the Q of a series resonant LC tank with a resistor value of 10 ohms. Note that the resistor is in series with the LC and that an input voltage source should be used (the voltage across the LC tank goes to zero at resonance.)

**1.12** Suppose the input voltage of the integrator in Fig. 1.34 is zero and that the op-amp has a 10 mV input-referred offset voltage. If the input-referred offset voltage is modeled using a 10 mV voltage source in series with the non-inverting (+) op-amp input then estimate the output voltage of the op-amp in the time-domain. Assume that at $t = 0$ $V_{out} = 0$. Verify your answer with SPICE.

IPR2023-00697
Theta EX2019

Chapter
2

# The Well

To develop a fundamental understanding of CMOS integrated circuit layout and design, we begin with a study of the well. The well is the first layer fabricated when making a CMOS IC. The approach of studying the details of each fabrication (layout) layer will build a solid foundation for understanding the performance limitations and parasitics (the pn junctions, capacitances, and resistances inherent in a CMOS circuit) of the CMOS process.

*The Substrate (The Unprocessed Wafer)*

CMOS circuits are fabricated on and in a silicon wafer, as discussed in Ch. 1. This wafer is doped with donor atoms, such as phosphorus for an n-type wafer, or acceptor atoms, such as boron for a p-type wafer. Our discussion centers around a p-type wafer (the most common substrate used in CMOS IC processing). When designing CMOS integrated circuits with a p-type wafer, n-channel MOSFETs (NMOS for short) are fabricated directly in the p-type wafer, while p-channel transistors, PMOS, are fabricated in an "n-well." The substrate or well are sometimes referred to as the bulk or body of a MOSFET. CMOS processes that fabricate MOSFETs in the bulk are known as "bulk CMOS processes." The well and the substrate are illustrated in Fig. 2.1, though not to scale.

Often an epitaxial layer is grown on the wafer. In this book we will not make a distinction between this layer and the substrate. Some processes use a p-well or both n- and p-wells (sometimes called twin tub processes). A process that uses a p-type (n-type) substrate with an n-well (p-well) is called an "n-well process" ("p-well process"). *We will assume, throughout this book, that an n-well process is used for the layout and design discussions.*

*A Parasitic Diode*

Notice, in Fig. 2.1, that the n-well and the p-substrate form a diode. In CMOS circuits, the substrate is usually tied to the lowest voltage in the circuit (generally, the substrate is grounded) to keep this diode from forward biasing. Ideally, zero current flows in the substrate. We won't concern ourselves with how the substrate is connected to ground at this point (see Ch. 4).

IPR2023-00697
Theta EX2019

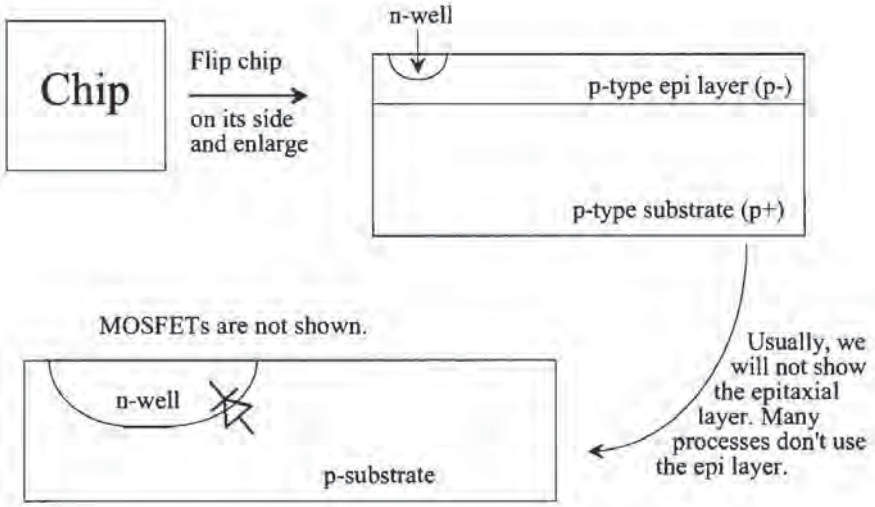



**Figure 2.1** The top (layout) and side (cross-sectional) view of a die.

*Using the N-well as a Resistor*

In addition to being used as the body for p-channel transistors, the n-well can be used as a resistor, Fig. 2.2. The voltage on either side of the resistor must be large enough to keep the substrate/well diode from forward biasing.

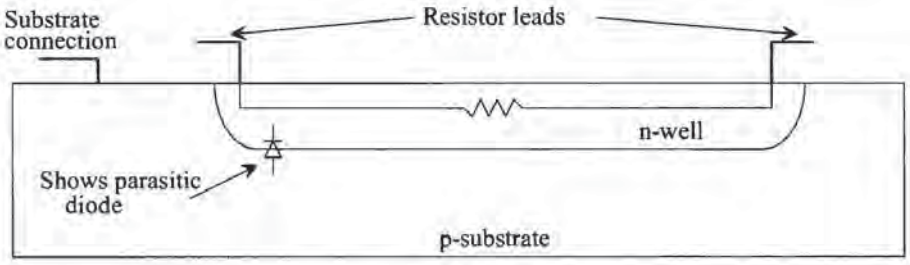



**Figure 2.2** The n-well can be used as a resistor.

## 2.1 Patterning

CMOS integrated circuits are formed by patterning different layers on and in the silicon wafer. Consider the following sequence of events that apply, in a fundamental way, to any layer that we need to pattern. We start out with a clean, bare wafer, as shown in Fig. 2.3a. The distance given by the line A to B will be used as a reference in Figs. 2.3b–j. Figures 2.3b–j are cross-sectional views of the dashed line shown in (a). The small box in Fig. 2.3a is drawn with a layout program (and used for mask generation) to indicate where to put the patterned layer.

IPR2023-00697
Theta EX2019

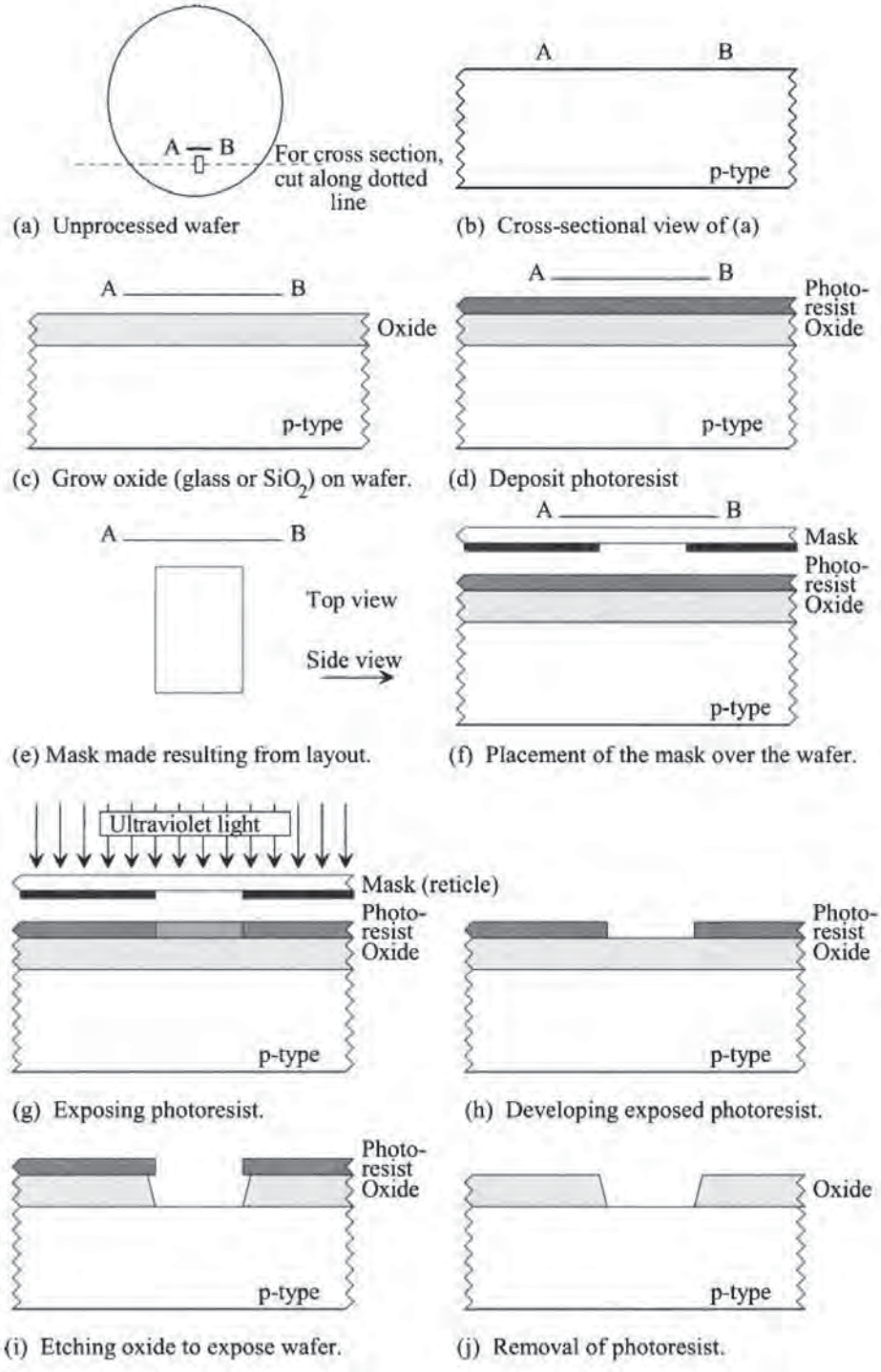



**Figure 2.3** Generic sequence of events used in photo patterning.

IPR2023-00697
Theta EX2019

The first step in our generic patterning discussion is to grow an oxide, $SiO_2$ or glass, a very good insulator, on the wafer. Simply exposing the wafer to air yields the reaction $Si + O_2 \rightarrow SiO_2$. However, semiconductor processes must have tightly controlled conditions to precisely set the thickness and purity of the oxide. We can grow the oxide using a reaction with steam, $H_2O$, or with $O_2$ alone. The oxide resulting from the reaction with steam is called a wet oxide, while the reaction with $O_2$ is a dry oxide. Both oxides are called thermal oxides due to the increased temperature used during oxide growth. The growth rate increases with temperature. The main benefit of the wet oxide is fast growing time. The main drawback of the wet oxide is the hydrogen byproduct. In general terms, the oxide grown using the wet techniques is not as pure as the dry oxide. The dry oxide generally takes a considerably longer time to grow. Both methods of growing oxide are found in CMOS processes. An important observation we should make when looking at Fig. 2.3c is that the oxide growth actually consumes silicon. This is illustrated in Fig. 2.4. The overall thickness of the oxide is related to thickness of the consumed silicon by

$$x_{Si} = 0.45 \cdot x_{ox} \tag{2.1}$$

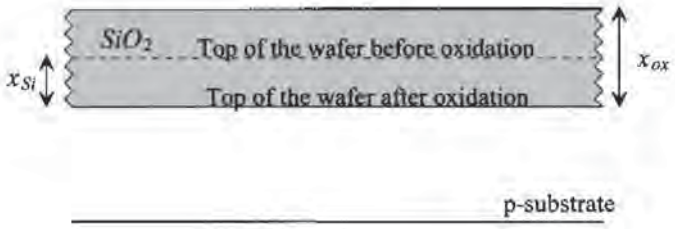



**Figure 2.4** How growing oxide consumes silicon.

The next step of the generic CMOS patterning process is to deposit a photosensitive resist layer across the wafer (see Fig. 2.3d). Keep in mind that the dimensions of the layers, that is, oxide, resist, and the wafer, are not drawn to scale. The thickness of a wafer is typically 500 μm, while the thickness of a grown oxide or a deposited resist may be only a μm ($10^{-6}$ m) or even less. After the resist is baked, the mask derived from the layout program, Figs. 2.3e and f, is used to selectively illuminate areas of the wafer, Fig. 2.3g. In practice, a single mask called a reticle, with openings several times larger than the final illuminated area on the wafer, is used to project the pattern and is stepped across the wafer with a machine called a stepper to generate the patterns needed to create multiple copies of a single chip. The light passing through the opening in the reticle is photographically reduced to illuminate the correct size area on the wafer.

The photoresist is developed (Fig. 2.3h), removing the areas that were illuminated. This process is called a positive resist process because the area that was illuminated was removed. A negative resist process removes the areas of resist that were not exposed to the light. Using both types of resist allows the process designer to cut down on the number of masks needed to define a CMOS process. Because creating the masks is expensive, lowering the number of masks is equated with lowering the cost of a process. This is also important in large manufacturing plants where fewer steps equal lower cost.

IPR2023-00697
Theta EX2019

The next step in the patterning process is to remove the exposed oxide areas (Fig. 2.3i). Notice that the etchant etches under the resist, causing the opening in the oxide to be larger than what was specified by the mask. Some manufacturers intentionally bloat (make larger) or shrink (make smaller) the masks as specified by the layout program. Figure 2.3j shows the cross-sectional view of the opening after the resist has been removed.

### 2.1.1 Patterning the N-well

At this point we can make an n-well by diffusing donor atoms, those with five valence electrons, as compared to the 4 four found in silicon, into the wafer. Referring to our generic patterning discussion given in Fig. 2.3, we begin by depositing a layer of resist directly on the wafer, Fig. 2.3d (without oxide). This is followed by exposing the resist to light through a mask (Figs. 2.3f and g) and developing or removing the resist (Fig. 2.3h). The mask used is generated with a layout program. The next step in fabricating the n-well is to expose the wafer to donor atoms. The resist blocks the diffusion of the atoms, while the openings allow the donor atoms to penetrate into the wafer. This is shown in Fig. 2.5a. After a certain amount of time, depending on the depth of the n-well desired, the diffusion source is removed (Fig. 2.5b). Notice that the n-well "outdiffuses" under the resist; that is, the final n-well size is not the same as the mask size. Again, the foundry where the chips are fabricated may bloat or shrink the mask to compensate for this lateral diffusion. The final step in making the n-well is the removal of the resist (Fig. 2.5c).

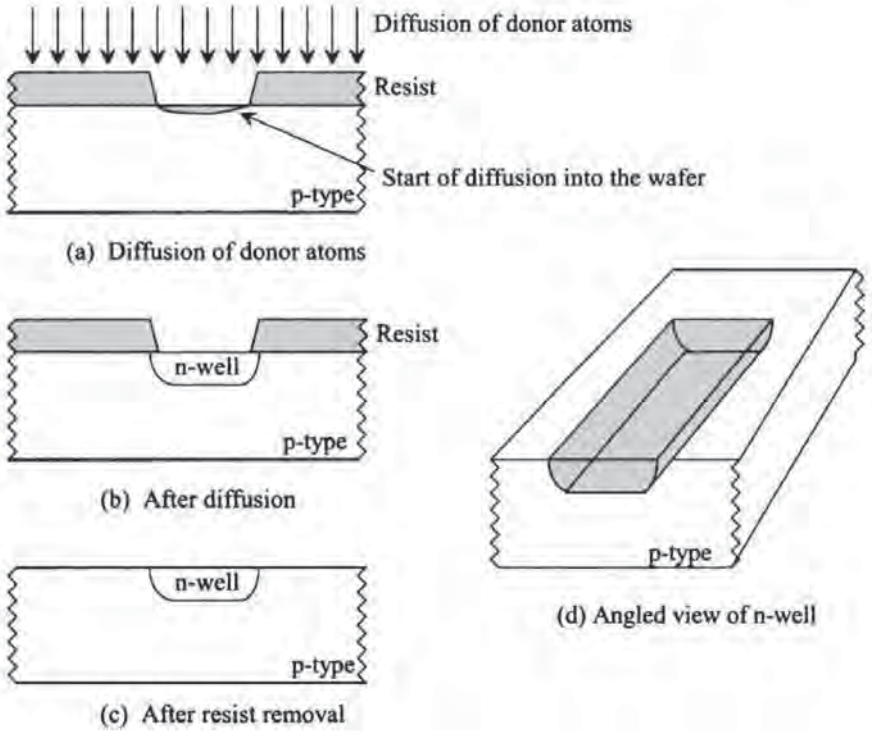



**Figure 2.5** Formation of the n-well.

IPR2023-00697
Theta EX2019

## 2.2 Laying Out the N-well

When we lay out the n-well, we are viewing the chip from the top. One of the key points in this discussion, as well as the discussions to follow, is that we do layout to a generic **scale factor**. If, for example, the minimum device dimensions are 50 nm (= 0.05 μm = $50 \times 10^{-9}$ m), then an n-well box **drawn** 10 by 10, see Fig. 2.6, has an actual size after it is fabricated of $10 \cdot 50\,nm$ or half a micron (0.5 μm = 500 nm), neglecting lateral diffusion or other process imperfections. We scale the layout when we generate the GDS (calma stream format) or CIF (Caltech-intermediate-format) file from a layout program. (A GDS or CIF file is what the mask maker uses to make reticles.) Using integers to do layout simplifies things. We'll see in a moment that many electrical parameters are ratios (such as resistance) and so the scale factor cancels out of the ratio.

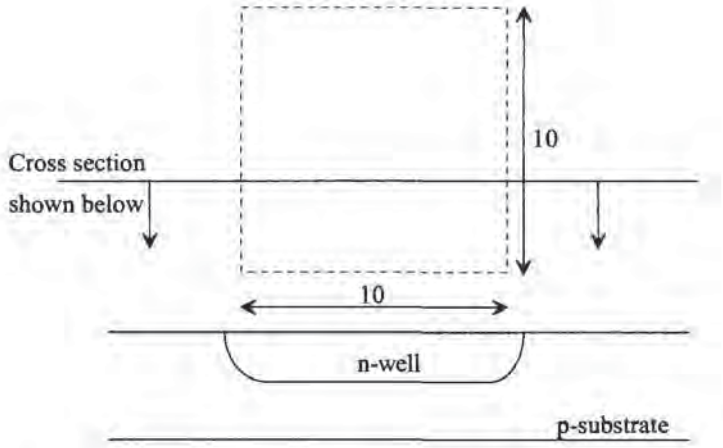



**Figure 2.6** Layout and cross-sectional view of a 10 by 10 (drawn) n-well.

### 2.2.1 Design Rules for the N-well

Now that we've laid out the n-well (drawn a box in a layout program), we might ask the question, "Are there any limitations or constraints on the size and spacing of the n-wells?" That is to say, "Can we make the n-well 2 by 2?" Can we make the distance between the n-wells 1? As we might expect, there are minimum spacing and size requirements for all layers in a CMOS process. Process engineers, who design the integrated circuit process, specify the design rules. The design rules vary from one process technology (say a process with a scale factor of 1 μm) to another (say a process with a scale factor of 50 nm).

Figure 2.7 shows sample design rules for the n-well. The minimum size (width or length) of any n-well is 6, while the minimum spacing between different n-wells is 9. As the layout becomes complicated, the need for a program that ensures that the design rules are not violated is needed. This program is called a *design rule checker* program (DRC program). Note that the minimum size may be set by the quality of patterning the resist (as seen in Fig. 2.5), while the spacing is set by the parasitic npn transistor seen in Fig. 2.7. (We don't want the n-wells interacting to prevent the parasitic npn from turning on.)

IPR2023-00697
Theta EX2019

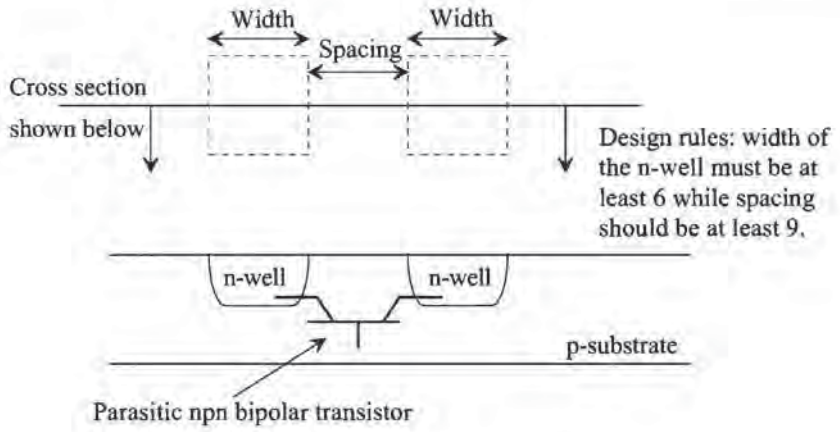



**Figure 2.7** Sample design rules for the n-well.

## 2.3 Resistance Calculation

In addition to serving as a region in which to build PMOS transistors (called the body or bulk of the PMOS devices), n-wells are often used to create resistors. The resistance of a material is a function of the material's resistivity, ρ, and the material's dimensions. For example, the slab of material in Fig. 2.8 between the two leads has a resistance given by

$$R = \frac{\rho}{t} \cdot \frac{L \cdot scale}{W \cdot scale} = \frac{\rho}{t} \cdot \frac{L}{W} \qquad (2.2)$$

In semiconductor processing, all of the fabricated thicknesses, $t$, seen in a cross-sectional view, such as the n-well's, are fixed in depth (**this is important**). When doing layout, we only have control over $W$ (width) and $L$ (length) of the material. The $W$ and $L$ are what we see from the top view, that is, the layout view. We can rewrite Eq. (2.2) as

$$R = R_{square} \cdot \frac{L}{W} \rightarrow R_{square} = \frac{\rho}{t} \qquad (2.3)$$

$R_{square}$ is the *sheet resistance* of the material in Ω/square (noting that when $L = W$ the layout is square and $R = R_{square}$).

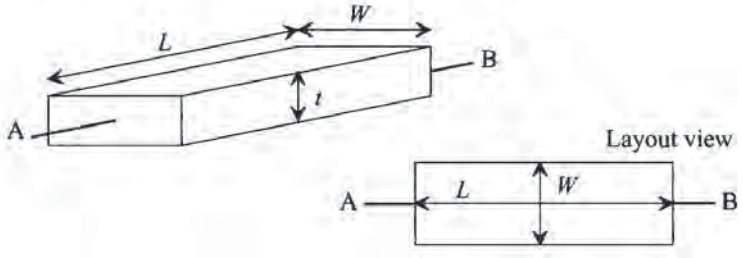



**Figure 2.8** Calculation of the resistance of a rectangular block of material.

IPR2023-00697
Theta EX2019

**Example 2.1**
Calculate the resistance of an n-well that is 10 wide and 100 long. Assume that the n-well's sheet resistance is typically 2 kΩ/square; however, it can vary with process shifts from 1.6 to 2.4 kΩ/square.

The typical resistance, using Eq. (2.3), between the ends of the n-well is

$$R = 2{,}000 \cdot \frac{100}{10} = 20\ k\Omega$$

The maximum value of the resistor is 24k, while the minimum value is 16k. ■

*Layout of Corners*

Often, to minimize space, resistors are laid out in a serpentine pattern. The corners, that is, where the layer bends, are not rectangular. This is shown in Fig. 2.9a. All sections in Fig 2.9a are square, so the resistance of sections 1 and 3 is $R_{square}$. The equivalent resistance of section 2 between the adjacent sides, however, is approximately 0.6 $R_{square}$. The overall resistance between points A and B is therefore 2.6 $\cdot R_{square}$. As seen in Ex. 2.1 the actual resistance value varies with process shifts. The layout shown in Fig. 2.9b uses wires to connect separate sections of unit resistors to avoid corners. Avoiding corners in a resistor is the (generally) preferred method of layout in analog circuit design where the ratio of two resistors is important. For example, the gain of an op-amp circuit may be $R_f/R_1$.

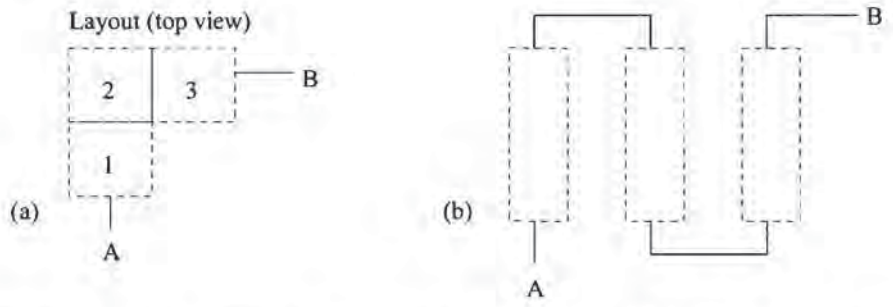



**Figure 2.9** (a) Calculating the resistance of a corner section and (b) layout to avoid corners.

### 2.3.1 The N-well Resistor

At this point, it is appropriate to show the actual cross-sectional view of the n-well after all processing steps are completed (Fig. 2.10). The n+ and p+ implants are used to increase the threshold voltage of the field devices; more will be said on this in Ch. 7. In all practical situations, the sheet resistance of the n-well is measured with the field implant in place, that is, with the n+ implant between the two metal connections in Fig. 2.10. Not shown in Fig. 2.10 is the connection to substrate. The field oxide (FOX; also known as ROX or recessed oxide) are discussed in Chs. 4 and 7 when we discuss the active and poly layers. The reader shouldn't, at this point, feel they should understand any of the cross-sectional layers in Fig. 2.10 except the n-well. Note that the field implants *aren't drawn* in the layout and so their existence is transparent to the designer.

IPR2023-00697
Theta EX2019

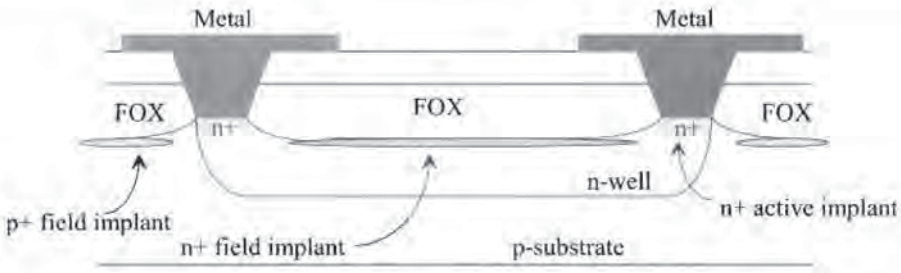



**Figure 2.10** Cross-sectional view of n-well showing field implant. The field implantation is sometimes called the "channel stop implant."

## 2.4 The N-well/Substrate Diode

As seen in Fig. 2.1, placing an n-well in the p-substrate forms a diode. It is important to understand how to model a diode for hand calculations and in SPICE simulations. In particular, let's discuss general diodes using the n-well/substrate pn junction as an example. The DC characteristics of the diode are given by the Shockley diode equation, or

$$I_D = I_S\left(e^{\frac{v_d}{nV_T}} - 1\right) \tag{2.4}$$

The current $I_D$ is the diode current; $I_S$ is the scale (saturation) current; $V_d$ is the voltage across the diode where the anode, A, (p-type material) is assumed positive with respect to the cathode, K, (n-type); and $V_T$ is the *thermal voltage*, which is given by $\frac{kT}{q}$ where $k$ = Boltzmann's constant ($1.3806 \times 10^{-23}$ Joules per degree Kelvin), $T$ is temperature in Kelvin, $n$ is the emission coefficient (a term that is related to the doping profile and affects both the exponential behavior of the diode and the diode's turn-on voltage), and $q$ is the electron charge of $1.6022 \times 10^{-19}$ coulombs. The scale current and thus the overall diode current are related in SPICE by an area factor (not associated with or to be confused with the scale term we use in layouts, Eq. (2.2)). The SPICE (Simulation Program with Integrated Circuit Emphasis) circuit simulation program assumes that the value of $I_s$ supplied in the model statement was measured for a device with a reference area of 1. If an area factor of 2 is supplied for a diode, then $I_S$ is doubled in Eq. (2.4).

### 2.4.1 A Brief Introduction to PN Junction Physics

A conducting material is made up of atoms that have easily shared orbiting electrons. As a simple example, copper is a better conductor than aluminum because the copper atom's electrons aren't as tightly coupled to its nucleus allowing its electrons to move around more easily. An insulator has, for example, eight valence electrons tightly coupled to the atom's nucleus. A significant electric field is required to break these electrons away from their nucleus (and thus for current conduction). A semiconductor, like silicon, has four valence electrons. Silicon's conductivity falls between an insulator and a conductor (and thus the name "semiconductor"). As silicon atoms are brought together, they form both a periodic crystal structure and bands of energy that restrict the allowable energies an electron can occupy. At absolute zero temperature, ($T = 0$ K), all of the valence electrons in the semiconductor crystal reside in the valence energy band, $E_v$. As temperature

IPR2023-00697
Theta EX2019

increases, the electrons gain energy (heat is absorbed by the silicon crystal), which causes some of the valence electrons to break free and move to a conducting energy level, $E_c$. Figure 2.11 shows the movement of an electron from the valence band to the conduction band. Note that there aren't any allowable energies between $E_v$ and $E_c$ in the silicon crystal structure (if the atom were by itself, that is, not in a crystal structure this exact limitation isn't present). Further note that when the **electron** moves from the valence energy band to the conduction energy band, a **hole** is left in the valence band. Having an electron in the conduction band increases the material's conductivity (the electron can move around easily in the semiconductor material because it's not tightly coupled to an atom's nucleus). At the same time a hole in the valence band increases the material's conductivity (electrons in the valence band can move around more easily by simply falling into the open hole). The *key point* is that increasing the number of electrons or holes increases the materials conductivity. Since the hole is more tightly coupled to the atom's nucleus (actually the electrons in the valence band), its mobility (ability to move around) is lower than the electron's mobility in the conduction band. This point is **fundamentally important**. The fact that the mobility of a hole is lower than the mobility of an electron (in silicon) results in, among other things, the size of PMOS devices being larger than the size of NMOS devices (when designing circuits) in order for each device to have the same drive strength.

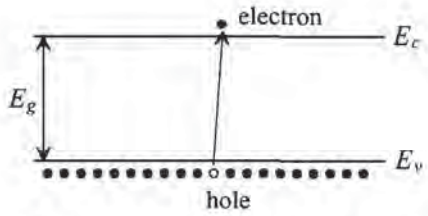



**Figure 2.11** An electron moving to the conduction band, leaving behind a hole in the valence band.

*Carrier Concentrations*

Pure silicon is often called *intrinsic* silicon. As the temperature of the silicon crystal is increased it absorbs heat. Some of the electrons in the valence band gain enough energy to jump the bandgap energy of silicon, $E_g$ (see also Eq. (23.21)), as seen in Fig. 2.11. This movement of an electron from the valence band to the conduction band is called *generation*. When the electron loses energy and falls back into the valence band, it is called *recombination*. The time the electron spends in the conduction band, before it recombines (drops back to the valence energy band), is random and often characterized by the carrier lifetime, $\tau_T$ (a root-mean-square, RMS, value of the random times the electrons spend in the conduction band of the silicon crystal). While discussing the actual processes involved with generation-recombination (GR) is outside the scope of this book, the carrier lifetime is a practical important parameter for circuit design. Another important parameter is the number of electrons in the conduction band (and thus the number of holes in the valence band) at a given time (again a random number). These carriers are called intrinsic carriers, $n_i$. At room temperature

$$n_i \approx 14.5 \times 10^9 \; carriers/cm^3 \tag{2.5}$$

IPR2023-00697
Theta EX2019

noting that $cm^3$ indicates a volume. If we call the number of free electrons (meaning electrons excited up in the conduction band of silicon) $n$ and the number of holes $p$, then for *intrinsic silicon*,

$$n = p = n_i \approx 14.5 \times 10^9 \; carriers/cm^3 \tag{2.6}$$

This may seem like a lot of carriers. However, the number of silicon atoms, $N_{Si}$, in a given volume of crystalline silicon is

$$N_{Si} = 50 \times 10^{21} \; atoms/cm^3 \tag{2.7}$$

so there is only one excited electron/hole pair for (roughly) every $10^{12}$ silicon atoms.

Next let's add different materials to intrinsic silicon (called *doping* the silicon) to change silicon's electrical properties. If we add a small amount of a material containing atoms with five valence electrons like phosphorous (silicon has four), then the added atom would bond with the silicon atoms and the *donated* electron would be free to move around (and easily excited to the conduction band). If we call the density of this added donor material $N_D$ with units of atoms/cm$^3$ and we assume the number of atoms added to the silicon is much larger than the intrinsic carrier concentration, then we can write the number of free electrons (the electron concentration $n$) in the material as

$$n \approx N_D \text{ when } N_{Si} >> N_D >> n_i \tag{2.8}$$

A material with added donor atoms is said to be an "n-type" material. Similarly, if we were to add a small material to silicon with atoms having three valence electrons (like boron), the added material would bond with the silicon resulting in a hole in the valence band. Again, this increases the conductivity of silicon because, now, the electrons in the valence band can move into the hole (having the effect of making it look like the hole is moving). The added material in this situation is said to be an *acceptor* material. The added material accepts an electron from the silicon crystal. If the density of the added acceptor material is labeled $N_A$, then the hole concentration, $p$, in the material is

$$p \approx N_A \text{ when } N_{Si} >> N_A >> n_i \tag{2.9}$$

A material with added acceptor atoms is said to be a "p-type" material.

If we dope a material with donor atoms, the number of free electrons in the material, $n$, goes up, as indicated by Eq. (2.8). We would expect, then, the number of free holes in the material to go down (some of those free electrons fall easily into the available holes reducing the number of holes in the material). The relationship between the number of holes, electrons, and intrinsic carrier concentration, is governed by the *mass-action law*

$$pn = n_i^2 \tag{2.10}$$

Consider the following example.

**Example 2.2**

Suppose silicon is doped with phosphorous having a density, $N_D$, of $10^{18}$ atoms/cm$^3$. Estimate the doped silicon's hole and electron concentration.

The electron concentration, from Eq. (2.8) is, $n = 10^{18}$ electrons/cm$^3$ (one electron for each donor atom). The hole concentration is found using the mass-action law as

IPR2023-00697
Theta EX2019

$$p = \frac{n_i^2}{n} = \frac{(14.5 \times 10^9)^2}{10^{18}} = 210 \text{ holes/cm}^3$$

Basically, all of the holes are filled. Note that with a doping density of $10^{18}$ there is one dopant atom for every 50,000 silicon atoms. If we continue to increase the doping concentration, our assumption that $N_{Si} >> N_D$ isn't valid and the material is said to be *degenerate* (no longer mainly silicon). A degenerate semiconductor doesn't follow the mass-action law (or any of the equations for silicon we present). ■

*Fermi Energy Level*

To describe the carrier concentration in a semiconductor, the Fermi energy level is often used. The Fermi energy level is useful when determining the contact potentials in materials. For example, the potential that you have to apply across a diode before it turns on is set by the p-type and n-type material contact potential difference. Also, the threshold voltage is determined, in part, by contact potentials.

The Fermi energy level simply indicates the energy level where the probability of occupation by a free electron is 50%. Figure 2.12a shows that for intrinsic silicon the (intrinsic) Fermi level, $E_i$ is close to the middle of the bandgap. In p-type silicon, the Fermi level, $E_f$, moves towards the valence band, Fig. 2.12b, since the number of free electrons, $n$, is reduced with the abundance of holes. Figure 2.12c shows the location of the Fermi energy level in n-type silicon. The Fermi level moves towards $E_c$ with the abundance of electrons in the conduction band.

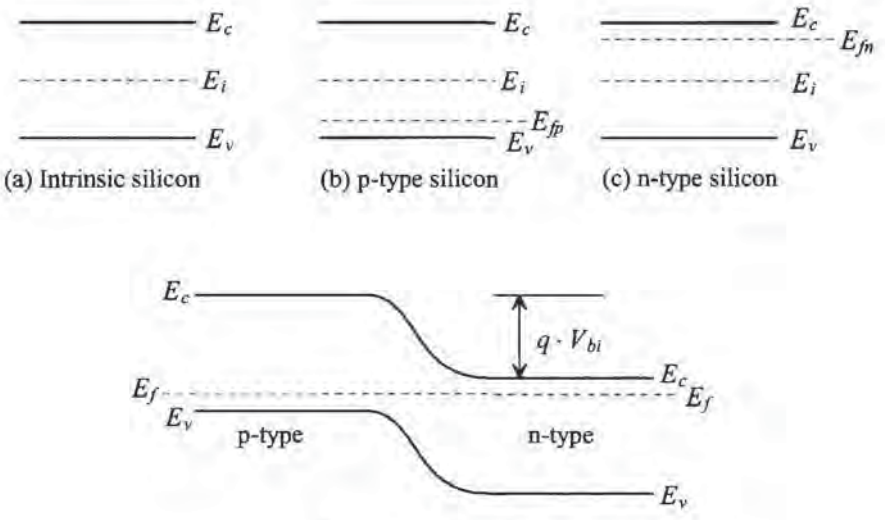



**Figure 2.12** The Fermi energy levels in various structures.

IPR2023-00697
Theta EX2019

The energy difference between the $E_i$ and $E_f$ is given, for a p-type semiconductor, by

$$E_i - E_{fp} = kT \cdot \ln \frac{N_A}{n_i} \tag{2.11}$$

and for an n-type semiconductor by

$$E_{fn} - E_i = kT \cdot \ln \frac{N_D}{n_i} \tag{2.12}$$

The band diagram of a pn junction (a diode) is seen in Fig. 2.12d. Note how the Fermi energy level is constant throughout the diode. A variation in $E_f$ would indicate a nonequilibrium situation (the diode has an external voltage applied across it). To get current to flow in a diode, we must apply an external potential that approaches the diode's contact potential (its built-in potential, $V_{bi}$ ). By applying a potential to forward bias the diode, the conduction energy levels in each side of the diode move closer to the same level. The voltage applied to the diode when the conduction energy levels are exactly at the same level is given by

$$V_{bi} = \frac{E_{fn} - E_{fp}}{q} = \frac{kT}{q} \cdot \ln \frac{N_A N_D}{n_i^2} \tag{2.13}$$

noting that $kT/q = V_T$ is the thermal voltage.

### 2.4.2 Depletion Layer Capacitance

We know that n-type silicon has a number of mobile electrons, while p-type silicon has a number of mobile holes (a vacancy of electrons in the valence band). Formation of a pn junction results in a depleted region at the p-n interface (Fig. 2.13). A depletion region is an area depleted of mobile holes or electrons. The mobile electrons move across the junction, leaving behind fixed donor atoms and thus a positive charge. The movement of holes across the junction, to the right in Fig. 2.13, occurs for the p-type semiconductor as well with a resulting negative charge. The fixed atoms on each side of the junction within the depleted region exert a force on the electrons or holes that have crossed the junction. This equalizes the charge distribution in the diode, preventing further charges from crossing the diode junction and also gives rise to a parasitic capacitance. This parasitic capacitance is called a *depletion* or *junction* capacitance.

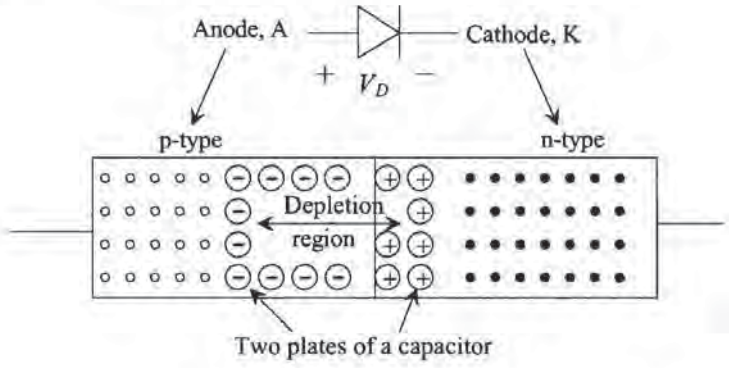



**Figure 2.13** Depletion region formation in a pn junction.

IPR2023-00697
Theta EX2019

The depletion capacitance, $C_j$, of a pn junction is modeled using

$$C_j = \frac{C_{j0}}{\left[1 - \left(\frac{V_D}{V_{bi}}\right)\right]^m} \tag{2.14}$$

$C_{j0}$ is the zero-bias capacitance of the pn junction, that is, the capacitance when the voltage across the diode is zero. $V_D$ is the voltage across the diode, $m$ is the grading coefficient (showing how the silicon changes from n- to p-type), and $V_{bi}$ is the built-in potential given by Eq. (2.13).

**Example 2.3**

Schematically sketch the depletion capacitance of an n-well/p-substrate diode 100 × 100 square (with a scale factor of 1 μm), given that the substrate doping is $10^{16}$ atoms/cm$^3$ and the well doping is $10^{17}$ atoms/cm$^3$. The measured zero-bias depletion capacitance of the junction is 100 aF/μm$^2$ (= $100 \times 10^{-18}$ $F/\mu m^2$), and the grading coefficient is 0.333. Assume the depth of the n-well is 3 μm.

The n-well doping (n-type side of the diode) is $N_D = 10^{17}$, while the substrate doping is $N_A$ is $10^{16}$. We can calculate the built-in potential using Eq. (2.13)

$$V_{bi} = 26\,mV \cdot \ln \frac{10^{16} \cdot 10^{17}}{(14.5 \times 10^9)^2} = 759\,mV$$

The depletion capacitance is made up of a *bottom* component and a *sidewall* component, as shown in Fig. 2.14 (see Eq. (5.17) for the more general form of Eq. (2.14)).

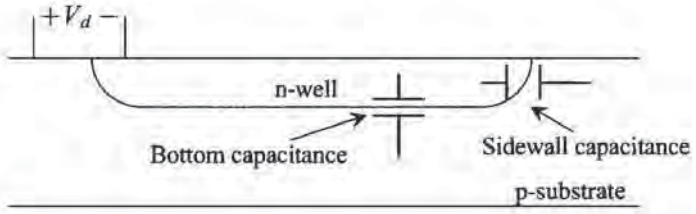



**Figure 2.14** A pn junction on the bottom and sides of the junction.

The bottom zero-bias depletion capacitance, $C_{j0b}$, is given by

$$C_{j0b} = (capacitance/area) \cdot (scale)^2 \cdot (bottom\ area) \tag{2.15}$$

which, for this example, is

$$C_{j0b} = (100\ aF/\mu m^2) \cdot (1\ \mu m)^2 \cdot (100)^2 = 1\ pF$$

The sidewall zero-bias depletion capacitance, $C_{j0s}$, is given by

$$C_{j0s} = (capacitance/area) \cdot (depth\ of\ the\ well) \cdot (perimeter\ of\ the\ well) \cdot (scale)^2$$

or (2.16)

$$C_{j0s} = (100\ aF/\mu m^2) \cdot (3) \cdot (400) \cdot (1\ \mu m)^2 = 120\,fF$$

IPR2023-00697
Theta EX2019

The total diode depletion capacitance between the n-well and the p-substrate is the parallel combination of the bottom and sidewall capacitances, or

$$C_j = \frac{C_{j0b}}{\left[1 - \left(\frac{V_D}{V_{bi}}\right)\right]^m} + \frac{C_{j0s}}{\left[1 - \left(\frac{V_D}{V_{bi}}\right)\right]^m} = \frac{C_{j0b} + C_{j0s}}{\left[1 - \left(\frac{V_D}{V_{bi}}\right)\right]^m} \qquad (2.17)$$

Substituting in the numbers, we get

$$C_j = \frac{1\,pF + 0.120\,pF}{\left(1 - \left(\frac{V_D}{0.759}\right)\right)^{0.33}} = \frac{1.120\,pF}{\left(1 - \left(\frac{V_D}{0.759}\right)\right)^{0.33}}$$

A sketch of how this capacitance changes with reverse potential is given in Fig. 2.15. Notice that when we discuss the depletion capacitance of a diode, it is usually with regard to a reverse bias ($V_D$ is negative). When the diode becomes forward-biased minority carriers, electrons in the p material and holes in the n material, injected across the junction, form a *stored* or *diffusion* charge in and around the junction and give rise to a storage or diffusion capacitance. This capacitance is usually much larger than the depletion capacitance. Furthermore, the time it takes to remove this stored charge can be significant. ■

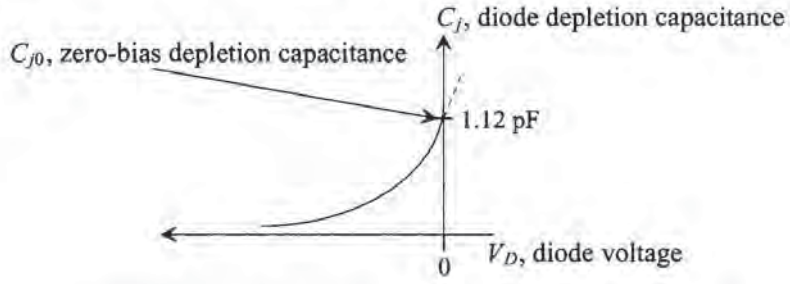



**Figure 2.15** Diode depletion capacitance against diode reverse voltage.

### 2.4.3 Storage or Diffusion Capacitance

Consider the charge distribution of the forward-biased diode shown in Fig. 2.16. When the diode becomes forward biased, electrons from the n-type side of the junction are attracted to the p-type side (and vice versa for the holes). After an electron drifts across the junction, it starts to diffuse toward the metal contact. If the electron recombines, that is, falls into a hole, before it hits the metal contact, the diode is called a *long base diode*. The time it takes an electron to diffuse from the junction to the point where it recombines is called the carrier lifetime, $\tau_T$ (see Sec. 2.4.1). For silicon this lifetime is on the order of 10 μs. If the distance between the junction and the metal contact is short, such that the electrons make it to the metal contact before recombining, the diode is said to be a *short base diode*. In either case, the time between crossing the junction and recombining will be labeled $\tau_T$ (transit time). A capacitance is formed between the electrons diffusing into the p-side and the holes diffusing into the n-side, that is, formed between the minority carriers. (Electrons are the minority carriers in the p-type semiconductor.) This capacitance is called a **diffusion** capacitance or **storage** capacitance due to the presence of the stored, or diffusing, minority carriers around the forward-biased pn junction.

IPR2023-00697
Theta EX2019

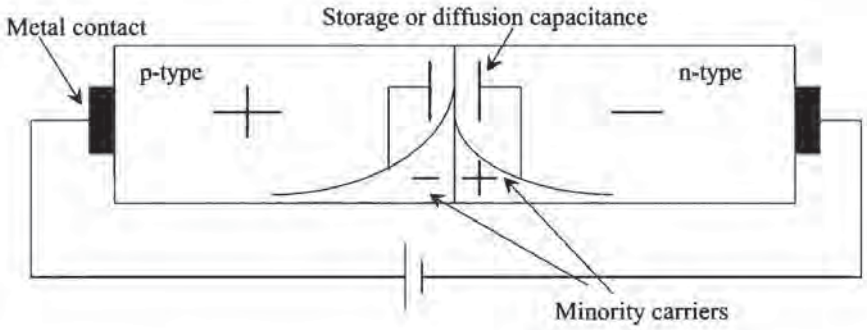



**Figure 2.16** Charge distribution in a forward-biased diode.

We can characterize the storage capacitance, $C_S$, in terms of the minority carrier lifetime. Under DC operating conditions, the storage capacitance is given by

$$C_S = \frac{I_D}{nV_T} \cdot \tau_T \tag{2.18}$$

$I_D$ is the DC current flowing through the forward-biased junction given by Eq. (2.4). Looking at the diode capacitance in this way is very useful for analog AC small-signal analysis. However, for digital applications, we are more interested in the large-signal switching behavior of the diode. It should be pointed out that, in general, for a CMOS process, it is undesirable to have a forward-biased pn junction. If we do have a forward-biased junction, it usually means that there is a problem. For example, electrostatic protection diodes are turning on (Fig. 4.17), capacitive feedthrough is possibly causing latch-up, or such. These topics appear in more detail later in the book.

Consider Fig. 2.17. In the following diode switching analysis, we assume that $V_F \gg 0.7$, $V_R < 0$ and that the voltage source has been at $V_F$ long enough to reach steady-state condition; that is, the minority carriers have diffused out to an equilibrium condition. At the time $t_1$, the input voltage source makes an abrupt transition from a forward voltage of $V_F$ to a reverse voltage of $V_R$, causing the current to change from $\frac{V_F - 0.7}{R}$ to $\frac{V_R - 0.7}{R}$. The diode voltage remains at 0.7 V, because the diode contains a stored charge that must be removed. At time $t_2$, the stored charge is removed. At this point, the diode basically looks like a voltage-dependent capacitor that follows Eq. (2.14). In other

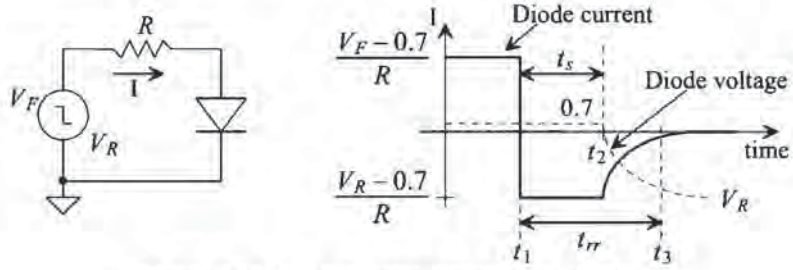



**Figure 2.17** Diode reverse recovery test circuit.

IPR2023-00697
Theta EX2019

words for $t > t_2$, the diode-depletion capacitance is charged through $R$ until the current in the circuit goes to zero and the voltage across the diode is $V_R$. This accounts for the exponential decay of the current and voltage shown in Fig. 2.17.

The diode storage time, the time it takes to remove the stored charge, $t_s$, is simply the difference in $t_2$ and $t_1$, or

$$t_s = t_2 - t_1 \tag{2.19}$$

This time is also given by

$$t_s = \tau_T \cdot \ln \frac{i_F - i_R}{-i_R} \tag{2.20}$$

where $\frac{V_F - 0.7}{R} = i_F$ and $\frac{V_R - 0.7}{R} = i_R$ = a negative number in this discussion. Note that it is quite easy to determine the minority carrier lifetime using this test setup.

Defining a time $t_3$, where $t_3 > t_2$, when the current in the diode becomes 10% of $\frac{V_R - 0.7}{R}$, we can define the diode reverse recovery time, or

$$t_{rr} = t_3 - t_1 \tag{2.21}$$

Note that the reverse recovery time (the time it takes to shut off a forward-biased diode) is one of the big reasons that digital circuits made using silicon bipolar transistors don't perform as well, in general, as their CMOS counterparts.

**Table 2.1** SPICE parameters related to diode.

| Name | SPICE | |
|---|---|---|
| $I_S$ | IS | Saturation current |
| $R_S$ | RS | Series resistance |
| $n$ | N | Emission coefficient |
| $V_{bd}$ | BV | Breakdown voltage |
| $I_{bd}$ | IBV | Current which flows during $V_{bd}$ |
| $C_{j0}$ | CJ0 | Zero-bias pn junction capacitance |
| $V_{bi}$ | VJ | Built-in potential |
| $m$ | M | Grading coefficient |
| $\tau_T$ | TT | Carrier transit time |

### 2.4.4 SPICE Modeling

The SPICE diode model parameters are listed in Table 2.1. The series resistance, $R_s$, results from the finite resistance of the semiconductor used in making the diode and the contact resistance, the resistance resulting from a metal contact to the semiconductor. At this point, we are only concerned with the resistance of the semiconductor. For a reverse-biased diode, the depletion layer width changes, increasing for larger reverse voltages (decreasing both the capacitance and series resistance, of the diode). However, when we model the series resistance, we use a constant value. In other words, SPICE will not show us the effects of a varying $R_S$.

IPR2023-00697
Theta EX2019

**Example 2.4**

Using SPICE, explain what happens when a diode with a carrier lifetime of 10 ns is taken from the forward-biased region to the reverse-biased region. Use the circuit shown in Fig. 2.18 to illustrate your understanding.

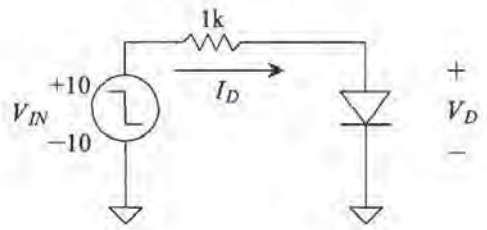



**Figure 2.18** Circuit used in Ex. 2.4 to demonstrate simulation of a diode's reverse recovery time.

We assume a zero-bias depletion capacitance of 1 pF. The SPICE netlist used to simulate the circuit in Fig. 2.18 is shown below.

```
*** Figure 2.19 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
let id=-i(vin)*1k
plot vd vin id
.endc

D1      vd      0       Dtrr
R1      vin     vd      1k
Vin     vin     0       DC      0       pulse 10 -10 10n .1n .1n 20n 40n

.Model Dtrr  D   is=1.0E-15 tt=10E-9 cj0=1E-12 vj=.7 m=0.33
.tran 100p 25n
.end
```

Figure 2.19 shows the current through the diode ($I_D$), the input voltage step ($V_{IN}$), and the voltage across the diode ($V_D$).

What's interesting to notice about this circuit is that current actually flows through the diode in the negative direction, even though the diode is forward biased (has a forward voltage drop of 0.7 V). During this time, the stored minority carrier charge (the diffusion charge) is removed from the junction. The storage time is estimated using Eq. (2.20)

$$t_s = 10ns \cdot \ln\frac{9.3+10.7}{10.7} = 6.25\ ns$$

which is close to the simulation results. Note that the input pulse doesn't change until 10 ns after the simulation starts. This ensures a steady-state condition when the input changes from 10 to −10 V. ■

IPR2023-00697
Theta EX2019

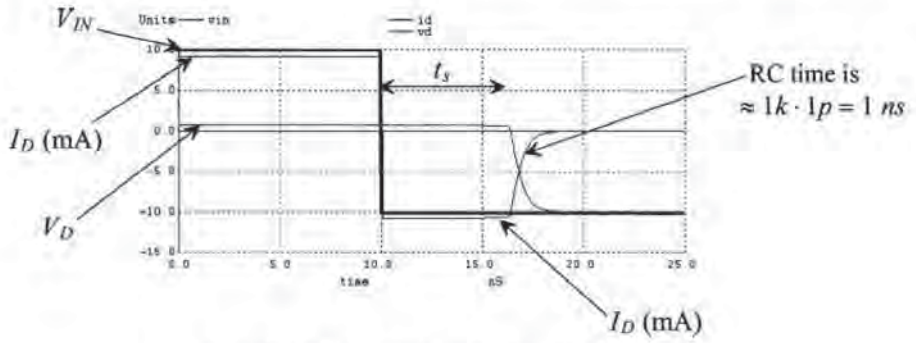



**Figure 2.19** The simulation results for Ex. 2.4.

## 2.5 The RC Delay through an N-well

At this point, we know that the n-well can function as a resistor and as a diode when used with the substrate. Figure 2.20a shows the parasitic capacitance and resistance associated with the n-well. Since there is a depletion capacitance from the n-well to the substrate, we could sketch the equivalent symbol for the n-well resistor, as shown in Fig. 2.20b. This is the basic form of an RC transmission line. If we put a voltage pulse into one side of the n-well resistor, then a finite time later, called the delay time and measured at the 50% points of the pulses, the pulse will appear.

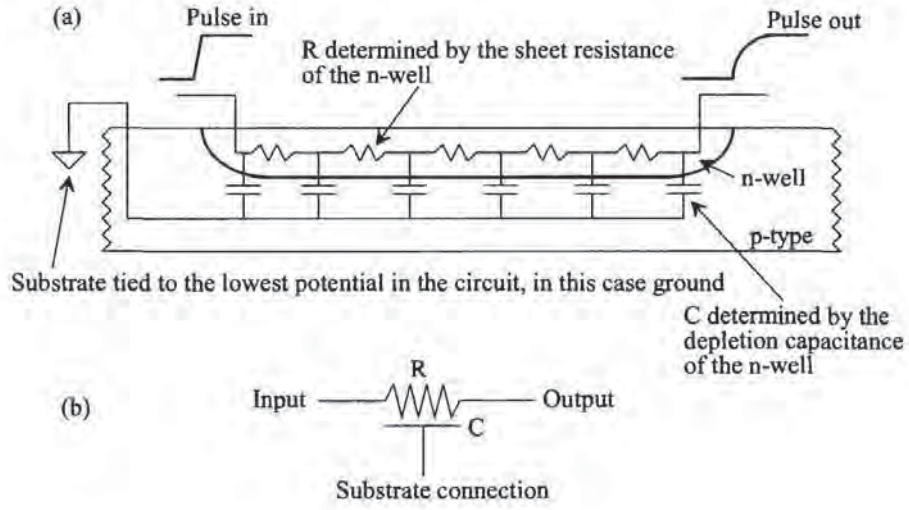



**Figure 2.20** (a) Parasitic resistance and capacitance of the n-well and (b) schematic symbol.

IPR2023-00697
Theta EX2019

*RC Circuit Review*

Figure 2.21 shows a simple RC circuit driven from a voltage pulse. If the input pulse transitions from 0 to $V_{pulse}$ at a time which we'll call zero, then the voltage across the capacitor (the output voltage) is given by

$$V_{out}(t) = V_{pulse}(1 - e^{-t/RC}) \tag{2.22}$$

The time it takes the output of the RC circuit to reach 50% of $V_{pulse}$ (defined as the circuit's delay time) is determined using

$$\frac{V_{pulse}}{2} = V_{pulse}(1 - e^{-t_d/RC}) \rightarrow t_d \approx 0.7RC \tag{2.23}$$

The rise time of the output pulse is defined as the time it takes the output to go from 10% of the final voltage to 90% of the final voltage ($V_{pulse}$). To determine the rise time in terms of the RC time constant, we can write

$$0.1V_{pulse} = V_{pulse}(1 - e^{-t_{10\%}/RC}) \tag{2.24}$$

and

$$0.9V_{pulse} = V_{pulse}(1 - e^{-t_{90\%}/RC}) \tag{2.25}$$

Solving these two equations for the rise time gives

$$t_r = t_{90\%} - t_{10\%} \approx 2.2RC \tag{2.26}$$

We will use these results *often* when designing digital circuits. **It's important** that any electrical engineer be able to derive Eqs. (2.23) and (2.26).

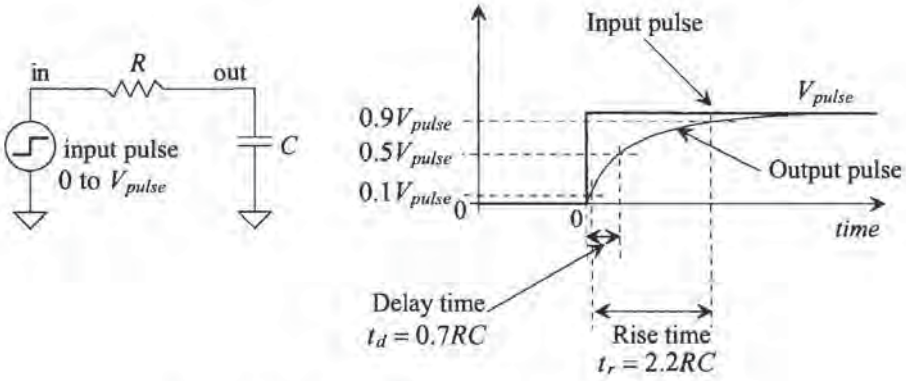



**Figure 2.21** Rise and delay times in an RC circuit.

*Distributed RC Delay*

The n-well resistor seen in Fig. 2.20 is an example of a distributed RC circuit (not a single, RC like the one seen in Fig. 2.21). In order to estimate the delay through a distributed RC, consider the circuit seen in Fig. 2.22. The delay to node A is estimated using

$$t_{dA} = 0.7R_{square}C_{square} \tag{2.27}$$

IPR2023-00697
Theta EX2019

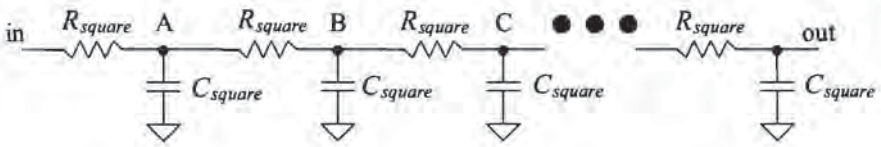



**Figure 2.22** Calculating the delay through a distributed RC delay.

The delay to node B is the sum of the delay to point A plus the delay associated with charging the capacitance at node B through $2R_{square}$ or

$$t_{dB} = 0.7(R_{square}C_{square} + 2R_{square}C_{square}) \tag{2.28}$$

Similarly, the delay to node C is

$$t_{dC} = 0.7(R_{square}C_{square} + 2R_{square}C_{square} + 3R_{square}C_{square}) \tag{2.29}$$

For a large number of sections, $l$, we can write the overall delay through the distributed RC delay as

$$t_d = 0.7R_{square}C_{square} \cdot (1 + 2 + 3 + 4 + ... + l) \tag{2.30}$$

The term in parentheses can be written as

$$(1 + 2 + 3 + 4 + ... + l) = \frac{l(l+1)}{2} \tag{2.31}$$

and so for a large number of sections $l$

$$t_d \approx 0.35 \cdot R_{square}C_{square} \cdot l^2 \tag{2.32}$$

where the $R_{square}$ and $C_{square}$ are the resistance and capacitance of each square of the distributed RC line.

**Example 2.5**

Estimate the delay through a 250 kΩ resistor made using an n-well with a width of 10 and a length of 500. Assume that the capacitance of a 10 by 10 square of n-well to substrate is 5 fF. Verify your answer with SPICE.

We can divide the n-well up into 50 squares each having a size of 10 by 10 and a resistance of 5 kΩ. The delay through the resistor is then (remembering 1 femto = $10^{-15}$)

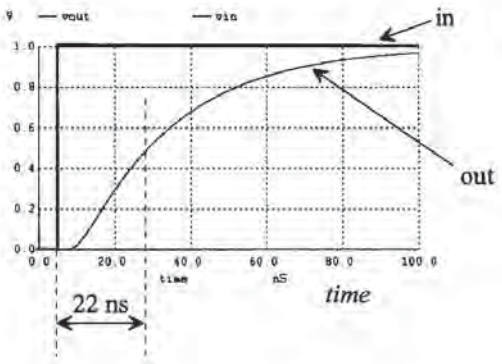
$$t_d = 0.35 \cdot (5k) \cdot (5f) \cdot (50)^2 \approx 22\ ns$$

Note that the total resistance value is $R_{square} \cdot l$ while the total capacitance of the resistor to substrate is $C_{square} \cdot l$. We can use this result to quickly estimate the delay of a distributed RC line (sometimes called an RC transmission line) as

$$t_d = 0.35 \cdot (\text{total resistance}) \cdot (\text{total capacitance}) \tag{2.33}$$

The simulation results are seen in Fig. 2.23. A SPICE lossy transmission line was used to model the distributed effects of the resistor. ■

IPR2023-00697
Theta EX2019




```
*** Figure 2.23 CMOS: Circuit Design, Layout, and Simulation ***
.control
destroy all
run
plot vin vout
.endc
.tran 100p 100n

O1 Vin 0 Vout 0 TRC
Rload Vout 0 1G
Vin vin 0 DC 0 pulse 0 1 5n 0
.model TRC ltra R=5k C=5f len=50
.end
```

**Figure 2.23** SPICE simulations showing the delay through an n-well resistor.

*Distributed RC Rise Time*

A similar analysis to what was used to arrive at Eq. (2.32) can be used to determine the rise time through a distributed RC line. The result is

$$t_r = 1.1 \cdot R_{square} C_{square} \cdot l^2 \qquad (2.34)$$

Using this equation in Ex. 2.5 results in an output rise time of approximately 69 ns. Comparing this estimate to the simulation results in Fig. 2.23, we see good agreement.

## 2.6 Twin Well Processes

Before going too much further, let's summarize some of the layout discussions presented in this chapter and discuss some concerns. Examine the cross-sectional views seen in Fig. 2.24. We know that the body of an NMOS transistor is p-type, while the body of a PMOS device is n-type. In the n-well process, Fig. 2.24a, the NMOS are fabricated directly in the p-type substrate and the PMOS are made in the n-well. For a p-well process, Fig. 2.24b, the NMOS are made in the p-well while the PMOS are fabricated in the n-type substrate. Note that sometimes the term **tub** is used in place of well (e.g., an n-tub process) because the resulting semiconductor area has a cross-sectional view like a bathtub's.

When implanting the n-well, in Fig. 2.24a for example, the substrate must be *counter-doped*. This means that the p-substrate must have n-type dopants (such as phosphorous or arsenic) added until its concentration changes from p- to n-type. The problem with counter-doping the p-substrate to make an n-well is that the quality of the resulting semiconductor isn't as good as it would be by simply taking intrinsic silicon and adding donor atoms. The acceptor atoms in the p-substrate become ionized (e.g., electrons fall into the holes) increasing scattering and reducing mobility. Sometimes the effects of counter doping are called *excessive doping effects*. In an n-well process, the PMOS devices suffer from excessive doping and so the quality of the device isn't as good as the quality of a PMOS device in a p-well process. For example, a PMOS device fabricated in an n-well is slower than a PMOS device fabricated in an n-substrate.

IPR2023-00697
Theta EX2019

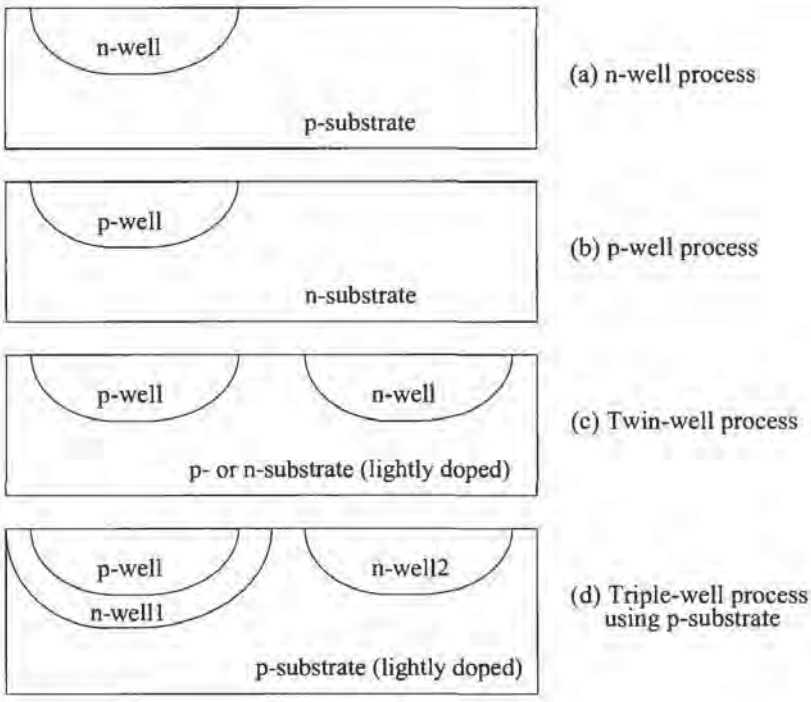



**Figure 2.24** The different possible wells used in a bulk CMOS process.

In an attempt to reduce excessive doping effects, a twin-well process, Fig. 2.24c, can be used. When using a lighter doped substrate, the amount of counter doping isn't as significant. We don't use an intrinsic silicon substrate because it is difficult to control the doping at such low levels. If a p-substrate is used, then the p-well is electrically connected to the substrate. The bodies of the NMOS are then all tied to the same potential, usually ground. To allow the bodies of the MOSFETs to be at different potentials a triple-well process can be used, Fig. 2.24d. The added n-well isolates the p-well from the substrate (the n-well1 and p-substrate form a diode that electrically isolates the p-well from the substrate). The p-well can exist directly in the substrate too.

*Design Rules for the Well*

Figure 2.25 shows the design rules, from MOSIS, for the well. Notice that there are four different sets of rules that the layout designer can follow. In this book **we will use the CMOSedu rules** to illustrate design examples. Before saying why, let's provide a little history and background. We know from Ch. 1 that MOSIS collects chip designs from various sources (education, private, not-for-profit, etc). These designs are put together to form the masks used for making the chips (on multiproject wafers). The actual vendors used by MOSIS to fabricate chips has changed throughout the years. To make the layouts transferable as well as scalable between different CMOS processes, MOSIS came up with the so-called SCMOS rules (scalable CMOS design rules). A parameter, $\lambda$, is used in the rules. All of the layouts are drawn on a $\lambda$ grid. When making the GDS file (or CIF file), the layout is scaled by this factor. For example, if an n-well box is drawn $10\lambda$ by $10\lambda$ with

IPR2023-00697
Theta EX2019

| Rule | Description | Lambda | | | Scale |
|---|---|---|---|---|---|
| | | SCMOS | SUBM | DEEP | CMOSedu |
| 1.1 | Minimum width | 10 | 12 | 12 | 6 |
| 1.2 | Minimum spacing between wells at different potential | 9 | 18 | 18 | 9 |
| 1.3 | Minimum spacing between wells at same potential | 6 | 6 | 6 | 3 |
| 1.4 | Minimum spacing between wells of different type (if both are drawn) | 0 | 0 | 0 | 0 |

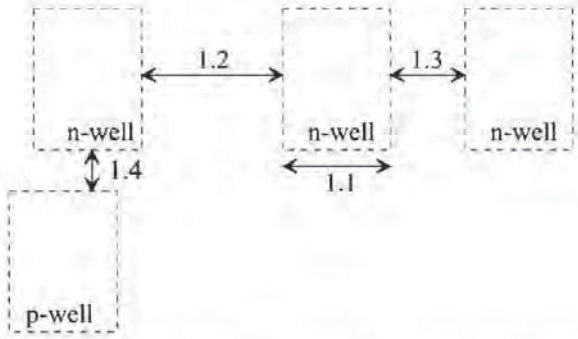



**Figure 2.25** Layouts showing the MOSIS design rules for the n-well.

a lambda of 0.3 μm, then when the GDS file is exported (called streaming the layout out) the actual size of the layout is 3 μm square. If the layout were used in a different process, only the value of λ would need to be changed. In other words, the exact same layout can be used in a different technology. Being able to use the same layout and simply *scale it* is a significant benefit of CMOS.

When the SCMOS rules were first introduced, the minimum size of a dimension in CMOS was approximately 1 μm. The "fabs" or vendors (the factory where the wafers are actually processed) also have a set of design rules. In general, the fab's design rules are tighter than the SCMOS rules. For example, one fab may specify a minimum n-well width of 3 μm, while another fab, in the same process technology, may specify 4 μm. The SCMOS rules may specify 5 μm to cover all possible situations (the price of using the SCMOS rules over the vendor's rules is larger layout areas). Unfortunately, as the process dimensions have shrunk over time, the MOSIS SCMOS rules weren't relaxed enough, making modifications necessary. This has led to the MOSIS submicron rules (SUBM) and the MOSIS deep-submicron (DEEP) rules seen in Fig. 2.25 (there are three sets of MOSIS scalable design rules, SCMOS, DEEP, and SUBM). Older processes still use the SCMOS rules, while the smaller technologies use the modified rules. Note that if a layout passes the DEEP rules, it will also pass the SCMOS rules (except for the exact via size).

Why use the CMOSedu rules in this book? Why not use one of the three sets of design rules from MOSIS? The answer comes from how we lay out MOSFETs. The **minimum length of a MOSFET using the MOSIS rules is 2** (2λ, keeping in mind that some scale factor is used when generating the GDS or CIF file). In the CMOSedu rules

IPR2023-00697
Theta EX2019

we took the MOSIS DEEP rules and divided by two. This means that layouts in the CMOSedu rules are *exactly the same* as the MOSIS DEEP rules except that they are scaled by a factor of 2. **Using the CMOSedu rules, the minimum length of a MOSFET is 1**. If MOSIS specifices a scale factor, λ, of 90 nm using the DEEP rules, where the minimum length is 2, then we would use a scale factor of 180 nm when using the CMOSedu rules with a minimum length of 1. In SPICE we use ".options scale=90n" when using the DEEP rules and ".options scale=180n" when using the CMOSedu rules.

*SEM Views of Wells*

Before leaving this chapter, let's show a scanning electron microscope (SEM) image of a well. In an SEM electrons are emitted from a cathode made with either tungsten or lanthanum hexaboride (LaB6). Tungsten is generally used for the cathode because it has a high melting point and a low vapor pressure. In some SEMs electrons are emitted via field emission. In either case an electron beam is formed and moved across the surface of an object. To move the electron beam it is passed through pairs of scanning coils and an objective lens. Varying the current through the coils deflects the beam, moving it across the surface of an object, and is used to form the image. The electrons are then attracted towards an anode and collected (as a varying output current).

Figure 2.26 shows an SEM image of a cross-sectional view of a portion of a CMOS memory chip. While most of the fabricated layers seen in this photograph will be covered in the next few chapters, we do point out, in the figure, a p-well (hard to see), a deep n-well, and the p-substrate. Note that in order to view different materials in a cross-section the sample is first stained prior to placement in the SEM and imaging.

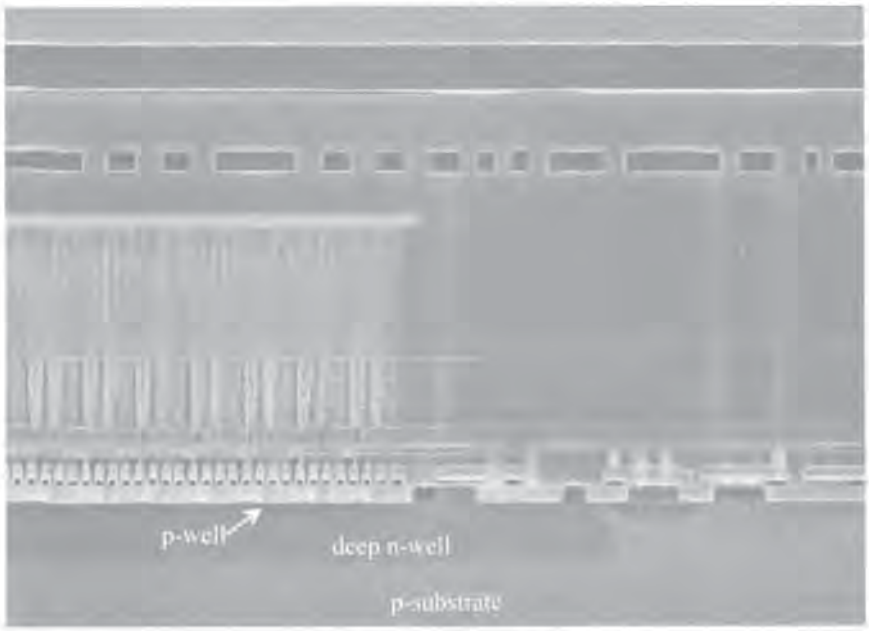



**Figure 2.26** SEM image showing the cross-section of a CMOS memory chip.

IPR2023-00697
Theta EX2019

## ADDITIONAL READING

[1] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, Second Edition, Cambridge University Press, 2010. ISBN 978-0521832946

[2] S. A. Campbell, *Fabrication Engineering at the Micro- and Nanoscale*, 3rd ed, Oxford University Press, 2008. ISBN 978-0195320176

[3] M. J. Madou, *Fundamentals of Microfabrication: The Science of Miniaturization*, 2nd ed., CRC Publisher, 2002. ISBN 978-0849308260

[4] R. C. Jaeger, *Introduction to Microelectronic Fabrication*, 2nd ed, volume 5 of the Modular Series on Solid State Devices, Prentice-Hall Publishers, 2002. ISBN 0-20-144494-1

[5] J. D. Plummer, M. D. Deal, and P. B. Griffin, *Silicon VLSI Technology, Fundamentals, Practice, and Modeling*, Prentice-Hall Publishers, 2000. ISBN 978-0130850379

## PROBLEMS

**2.1** For the layout seen in Fig. 2.27, sketch the cross-sectional views at the places indicated. Is there a parasitic pn junction in the layout? If so, where? Is there a parasitic bipolar transistor? If so, where?

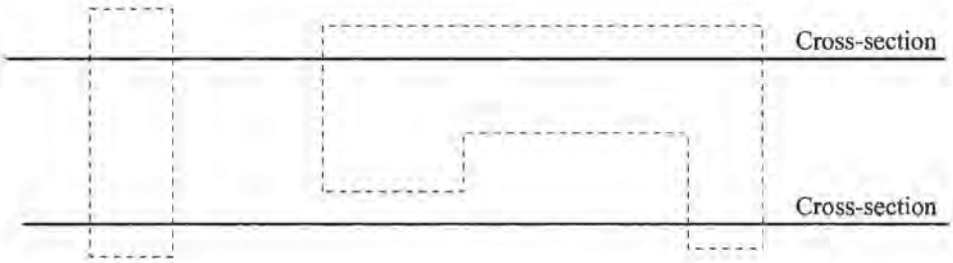



**Figure 2.27** Layout used in problem 2.1.

**2.2** Sketch (or use a layout tool) the layout of an n-well box that measures 100 by 10. If the scale factor is 50 nm, what is the actual size of the box after fabrication? What is the area before and after scaling? Neglect lateral diffusion or any other fabrication imperfections.

**2.3** Lay out a nominally 250 kΩ resistor using the n-well in a serpentine pattern similar to what's seen in Fig. 2.28. Assume that the maximum length of a segment is 100 and the sheet resistance is 2 kΩ/square. Design rule check the finished resistor. If the scale factor in the layout is 50 nm, estimate the fabricated size of the resistor.

**2.4** If the fabricated n-well depth, $t$, is 1 μm, then what are the minimum, typical, and maximum values of the n-well resistivity, ρ? Assume that the measured sheet resistances (minimum, typical, and maximum) are 1.6, 2.0, and 2.2 kΩ/square?

IPR2023-00697
Theta EX2019

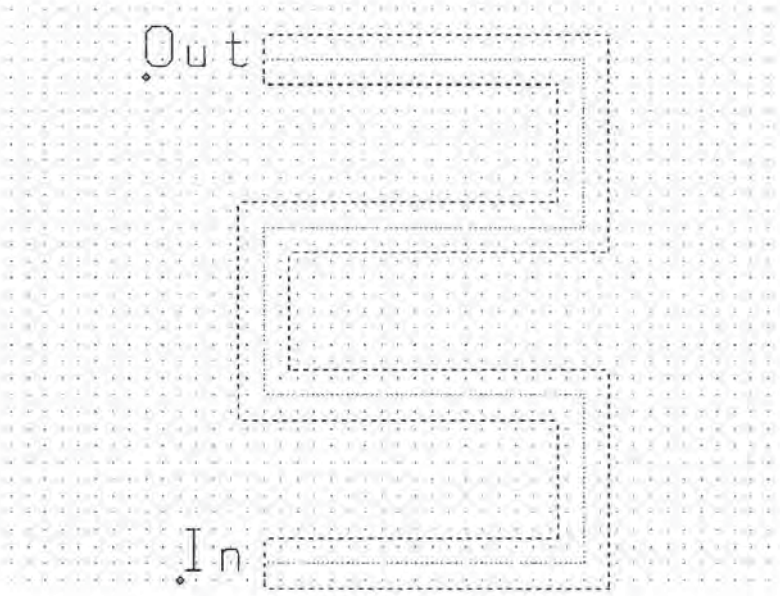



**Figure 2.28** Layout of an n-well resistor using a serpentine pattern.

**2.5** Normally, the scale current of a pn junction is specified in terms of a scale current density, $J_s$ (A/m$^2$), and the width and length of a junction (i.e., $I_s = J_s \cdot L \cdot W \cdot scale^2$ neglecting the sidewall component). Estimate the scale current for the diode of Ex. 2.3 if $J_s = 10^{-8}$ A/m$^2$.

**2.6** Repeat problem 2.5, including the sidewall component ($I_s = J_s \cdot L \cdot W \cdot scale^2 + J_s \cdot (2L+2W) \cdot scale \cdot$depth).

**2.7** Using the diode of Ex. 2.3 in the circuit of Fig. 2.29, estimate the frequency of the input signal when the AC component of $v_{out}$ is 707 μV (i.e., estimate the 3 dB frequency of the $|v_{out}/v_{in}|$).

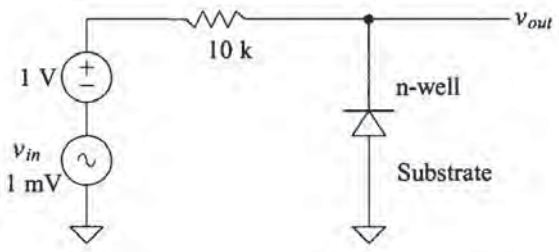



**Figure 2.29** Treating the diode as a capacitor. See Problem 2.7.

**2.8** Verify the answer given in problem 2.7 with SPICE.

IPR2023-00697
Theta EX2019

**2.9** Using SPICE, show that a diode can conduct significant current from its cathode to its anode when the diode is forward biased.

**2.10** Estimate the delay through a 1 MΩ resistor (10 by 2,000) using the values given in Ex. 2.5. Verify the estimate with SPICE.

**2.11** If one end of the resistor in problem 2.10 is tied to +1 V and the other end is tied to the substrate that is tied to ground, estimate the depletion capacitance ($F/m^2$) between the n-well and the substrate at the beginning, the middle, and the end of the resistor. Assume that the resistance does not vary with position along the resistor and that the scale factor is 50 nm, $C_{j0} = 25$ aF for a 10 by 10 square, m = 0.5, and $V_{bi} = 1$.

**2.12** The diode reverse breakdown current, that is, the current that flows when $|V_D| < BV$ (breakdown voltage), is modeled in SPICE by

$$I_D = IBV \cdot e^{-(V_D + BV)/V_T}$$

Assuming that 10 μA of current flows when the junction starts to break down at 10 V, simulate, using a SPICE DC sweep, the reverse breakdown characteristics of the diode. (The breakdown voltage, $BV$, is a positive number. When the diode starts to break down $-BV = V_D$. For this diode, breakdown occurs when $V_D = -10$ V.)

**2.13** Repeat Ex. 2.3 if the n-well/p-substrate diode is 50 square and the acceptor doping concentration is changed to $10^{15}$ atoms/$cm^3$.

**2.14** Estimate the storage time, that is, the time it takes to remove the stored charge in a diode, when $\tau_T = 5$ ns, $V_F = 5$ V, $V_R = -5$ V, $C_{j0} = 0.5$ pF, and $R = 1$k. Verify the estimate using SPICE.

**2.15** Repeat problem 2.14 if the resistor is increased to 10k. Comment on the difference in storage time between using a 1k and a 10k resistor. What dominates the increase the diode's reverse recovery time when using a 10k resistor instead of a 1k resistor?

IPR2023-00697
Theta EX2019

Chapter
3

# The Metal Layers

The metal layers in a CMOS integrated circuit connect circuit elements (MOSFETs, capacitors, and resistors). In the following discussion we'll discuss a generic CMOS process with two layers of metal. These levels of metal are named metal1 and metal2. The metal in a CMOS process is either aluminum or copper. In this chapter we look at the layout of the bonding pad, capacitances associated with the metal layers, crosstalk, sheet resistance, and electromigration.

## 3.1 The Bonding Pad

The bonding pad is at the interface between the die and the package or the outside world. One side of a wire is soldered to the pad, while the other side of the wire is connected to a lead frame, as was seen in Fig. 1.3. Figure 3.1 shows a close up of a bonding pad and wire. In this chapter we will not concern ourselves with electrostatic discharge (ESD) protection, which is an important design consideration when designing the pad.

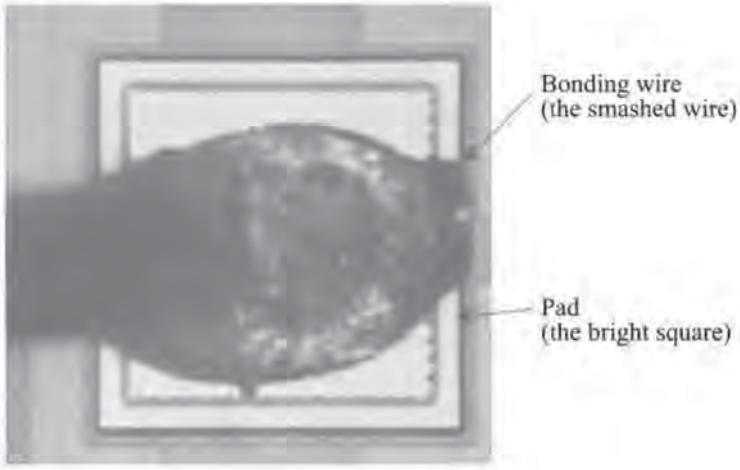



**Figure 3.1** The bonding wire connection to a pad.

IPR2023-00697
Theta EX2019

### 3.1.1 Laying Out the Pad I

The basic size of the bonding pad specified by MOSIS is a square 100 μm x 100 μm (actual size). For a probe pad, used to probe the circuit with a microprobe station, the size should be greater than 6 μm x 6 μm. In production chips the pads may vary in size (e.g., 75 x 100, or 50 x 75, etc.) depending on the manufacturer's design rules. *The final size of the pads are the only part of a layout that doesn't scale as process dimensions shrink.* The layout of a pad that uses metal2 is shown in Fig. 3.2. Notice, in the cross-sectional view, the layers of insulator ($SiO_2$ in most cases) under and above the metal2. These layers are used for isolation between the other layers in the CMOS process.

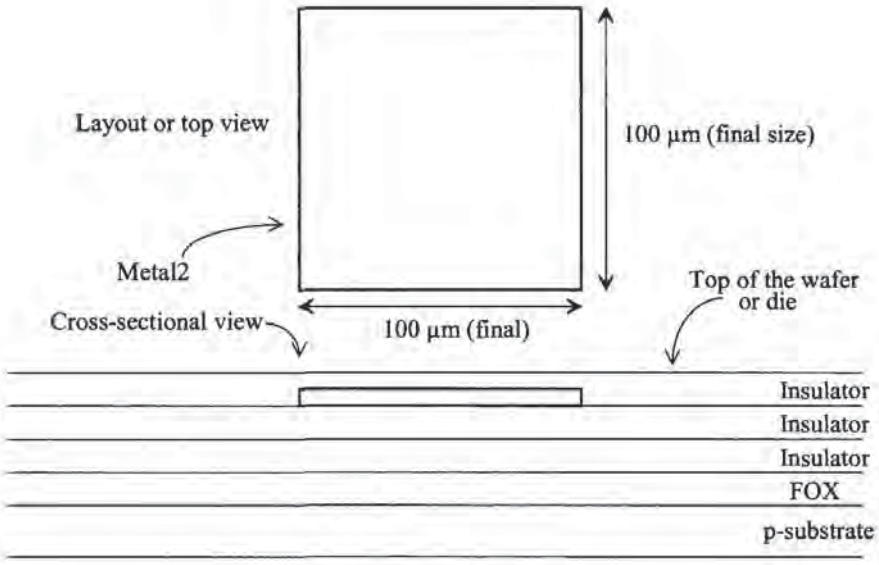



**Figure 3.2** Layout of metal2 used for bonding pad with associated cross-sectional view.

*Capacitance of Metal-to-Substrate*

Before proceeding any further, we might ask the question, "What is the capacitance from the metal2 box (pad) in Fig. 3.2 to the substrate?" The substrate is at ground potential and so, for all intents and purposes, it can be thought of as an equipotential plane. This is important because we have to drive this capacitance to get a signal off the chip. Table 3.1 gives typical values of parasitic capacitances for a CMOS process. Consider the following example.

**Example 3.1**

Estimate the parasitic capacitance associated with the pad in Fig. 3.2.

The capacitance associated with this pad is the sum of the plate (or bottom) capacitance and the fringe (or edge) capacitance. We can write

$$C_{pad,m2\rightarrow sub} = area \cdot C_{plate} + perimeter \cdot C_{fringe} \qquad (3.1)$$

IPR2023-00697
Theta EX2019

The area of the pad is 100 μm² square (100 μm by 100 μm), while the perimeter of the pad is 400 μm. Using the typical values of capacitance for metal2 to substrate in Table 3.1 gives

$$C_{pad,m2\rightarrow sub} = 10,000 \cdot 14\ aF + 400 \cdot 81\ aF = 172,400\ aF = 172.4\ fF = 0.172\ pF$$

A significant on-chip capacitance. ■

**Table 3.1** Typical parasitic capacitances in a CMOS process. Note that while the physical distance between the layers decreases, as process technology scales downwards, the dielectric constant used in between the layers can be decreased to keep the parasitic capacitances from becoming too significant. The values are representative of the parasitics in both long- and short-channel CMOS processes.

| | Plate Cap. aF/μm² | | | Fringe Cap. aF/μm | | |
|---|---|---|---|---|---|---|
| | min | typ | max | min | typ | max |
| Poly1 to subs. (FOX) | 53 | 58 | 63 | 85 | 88 | 92 |
| Metal1 to poly1 | 35 | 38 | 43 | 84 | 88 | 93 |
| Metal1 to substrate | 21 | 23 | 26 | 75 | 79 | 82 |
| Metal1 to diffusion | 35 | 38 | 43 | 84 | 88 | 93 |
| Metal2 to poly1 | 16 | 18 | 20 | 83 | 87 | 91 |
| Metal2 to substrate | 13 | 14 | 15 | 78 | 81 | 85 |
| Metal2 to diffusion | 16 | 18 | 20 | 83 | 87 | 91 |
| Metal2 to metal1 | 31 | 35 | 38 | 95 | 100 | 104 |

**Example 3.2**

The pad layout in Fig. 3.2 is the actual size. However, when we lay out the pad with the other circuit components, it must also be scaled when the layout is streamed out (see Sec. 1.2.3). If the scale factor in a design is 50 nm, what is the size of the box used for a pad that we draw with the layout program? Does the capacitance calculated in Ex. 3.1 change?

Because we want a final pad size of 100 μm by 100 μm, the drawn layout size of the box with a scale factor of 50 nm is

$$\frac{100\ \mu m}{0.05\ \mu m} = 2,000 \text{ (drawn size)}$$

Each side of the pad, in Fig. 3.2, is drawn with a size of 2,000 for a final (actual) size of 100 μm by 100 μm.

The capacitance calculated in Ex. 3.1 doesn't change. We can rewrite Eq. (3.1) as

$$C_{pad,m2\rightarrow sub} = area_{drawn} \cdot (scale)^2 \cdot C_{plate} + perimeter_{drawn} \cdot (scale) \cdot C_{fringe} \quad (3.2)$$

to use the drawn layout size. At this point there should be no confusion between the terms "drawn layout size" and "actual or final layout size." ■

IPR2023-00697
Theta EX2019

*Passivation*

Because an insulator is covering the pad (the piece of metal2) in Fig. 3.2, we can't bond (connect a wire) to it. The top layer insulator on the chip is also called passivation. The passivation helps protect the chip from contamination. Openings for bonding pads are called cuts in the passivation. To specify an opening or cut in the glass (insulator) covering the metal2, we use the overglass layer. The MOSIS rules specify 6 μm distance between the edge of the metal2 and the overglass box, as seen in Fig. 3.3. The drawn distance between the smaller overglass box and the larger metal2 box, with a scale factor of 50 nm, is 6/0.05 or 120.

There may be another layer in the MOSIS setups for the layout tool called the PAD layer. This layer has no fabrication significance but rather is used by the machine that bonds the chip to the lead frame to indicate the location of the pads. Since MOSIS takes designs of varying sizes and shapes, the locations of the pads change from one project to the next. This layer isn't really necessary since we can use the overglass layer (ensuring the overglass box has a drawn size of 1,760 square or a final size of 88 μm square) to indicate the location of the pads. We won't use the PAD layer in our layouts here.

*An Important Note*

Here we are using a CMOS process with (only) two layers of metal. In most modern CMOS processes, more than two layers of metal are used. If the process has five layers of metal, then the top layer (just like the top floor in a five-story building) is metal5. Therefore, metal5 is the layer the bonding wire is connected to.

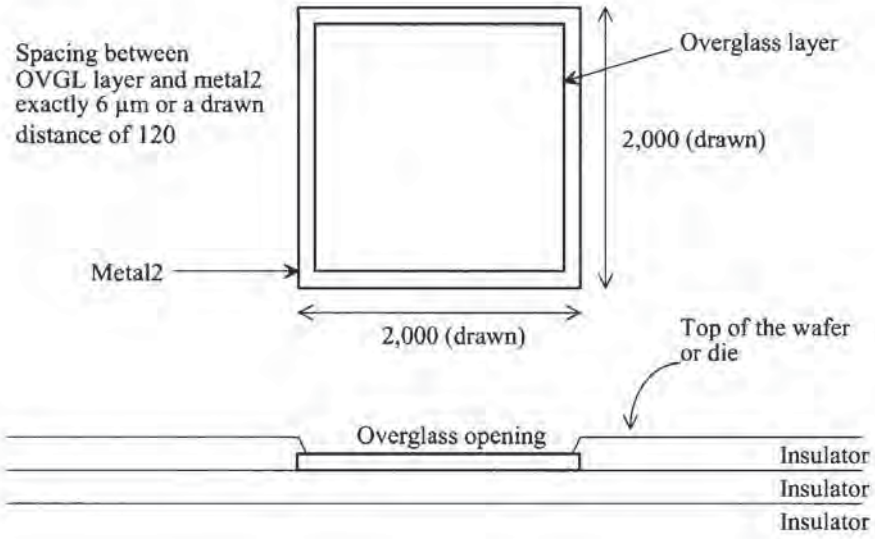



**Figure 3.3** Layout of a metal2 pad with pad opening for bonding connection in a 50 nm (scale factor) CMOS process.

IPR2023-00697
Theta EX2019

## 3.2 Design and Layout Using the Metal Layers

As mentioned earlier, the metal layers connect the resistors, capacitors, and MOSFETs in a CMOS integrated circuit. So far, in this book, we've learned about the layout layers n-well, metal2, overglass, and pad. In this section we'll also learn about the metal1 and via1 layers and the associated parasitic resistances and capacitances of these layers.

### 3.2.1 Metal1 and Via1

Metal1 is a layer of metal found directly below metal2. Figure 3.4 shows an example layout and cross-sectional view. The via1 layer connects metal1 and metal2. The via layer specifies that the insulator be removed in the location indicated. Then, for example, a tungsten "plug" is fabricated in the insulator's opening. When the metal2 is laid down, the plug provides a connection between the two metals. Note that if we were to use more than two layers of metal, then via2 would connect metal2 to metal3, via3 would connect metal3 to metal4, etc.

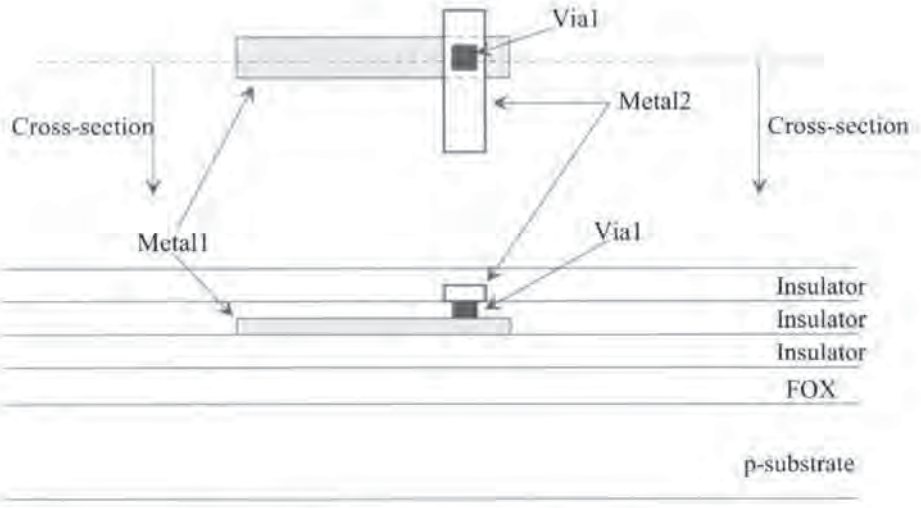



**Figure 3.4** Layout and cross-sectional views.

*An Example Layout*

Figure 3.5 shows an example layout using the n-well, metal1, via1, and metal2 layers. It's important that, before proceeding, this layout and the associated cross-sectional view are understood. For example, how would our cross-sectional view change if we moved the cross-sectional line used in Fig. 3.5 down slightly so that it only intersects the n-well and the metal2 layers? Answer: the cross-sectional view would be the same as seen in Fig. 3.5 except that the metal1 and via1 layers wouldn't be present.

IPR2023-00697
Theta EX2019

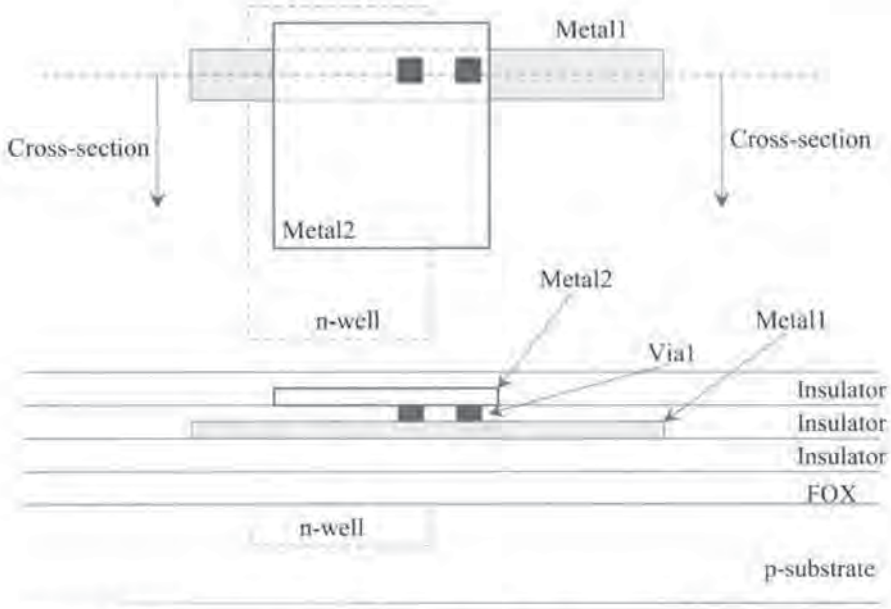



**Figure 3.5** An example layout and cross-sectional view using including the n-well.

### 3.2.2 Parasitics Associated with the Metal Layers

Associated with the metal layers are parasitic capacitances (see Table 3.1) and resistance. Like the n-well in the last chapter, the metal layers are characterized by a sheet resistance. However, the sheet resistance of the metal layers is considerably lower than the sheet resistance of the n-well. For the sake of examples in this book, we'll use metal sheet resistances of **0.1 Ω/square**. Also, there is a finite contact resistance of the via. The following examples illustrate some of the unwanted parasitics associated with these layers.

---

**Example 3.3**

Estimate the resistance of a piece of metal1 1 mm long and 200 nm wide. What is the drawn size of this metal line if the scale factor is 50 nm? Also estimate the delay through this piece of metal, treating the metal line as an RC transmission line. Verify your answer with a SPICE simulation.

The drawn size of the metal line is 1 mm/50 nm (= 20,000) by 200/50 (= 4). Figure 3.6 shows the layout of the metal wire (not to scale). The line consists of 1,000/0.2 = 20,000/4 = 5,000 squares of metal1.

To calculate the resistance of the metal line, we use Eq. (2.3)

$$R = \frac{0.1\ \Omega}{\text{square}} \cdot \frac{20,000}{4} = 0.1 \cdot 5,000 = 500\ \Omega$$

To calculate the capacitance, we use the information in Table 3.1 and either Eq. (3.1) or (3.2)

IPR2023-00697
Theta EX2019

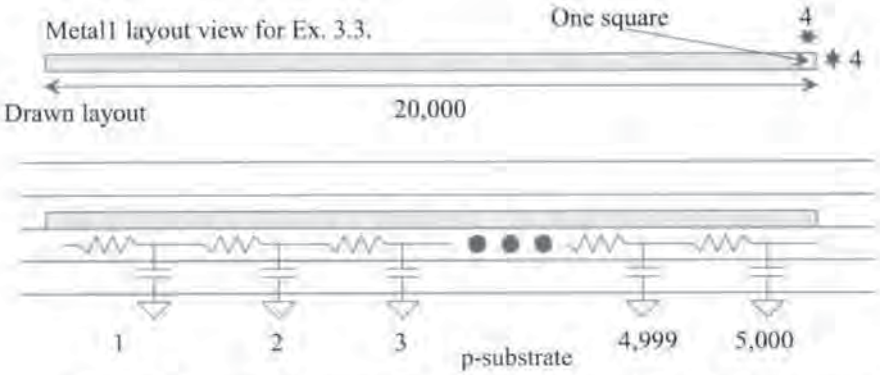



**Figure 3.6** Layout and cross-sectional view with parasitics for the metal line in Ex. 3.3.

$$C = (1,000 \cdot 0.2) \cdot 23\ aF + (2,000 \cdot 4) \cdot 79\ aF = 162\ fF$$

or the capacitance for each 200 nm by 200 nm square (4 by 4) of metal1 is

$$C_{square} = \frac{162\ fF}{5,000} = 32\ aF/\text{square}$$

The delay through the metal line is, using Eqs. (2.32) or (2.33)

$$t_d = 0.35 \cdot R_{square} C_{square} \cdot l^2 = 0.35(0.1)(32\ aF)(5,000)^2 = 28\ ps$$

or

$$t_d = 0.35RC = 0.35 \cdot 500 \cdot 162\ fF = 28\ ps$$

The delay of a metal1 line (with nothing connected to it) is 28 ps/mm when the parasitic capacitance and resistance are the limiting factors. The SPICE simulation results are seen in Fig. 3.7. ■

*Intrinsic Propagation Delay*

The result of this example (a metal delay of 28 ps/mm) should be compared to the intrinsic delay of a signal propagating in a material with a relative dielectric constant, $\varepsilon_r$ (no parasitic resistance). The velocity, $v$, of the signal in this situation is related to the speed of light, $c$, by

$$v = \frac{c}{\sqrt{\varepsilon_r}} \text{ (meters/second)} \tag{3.3}$$

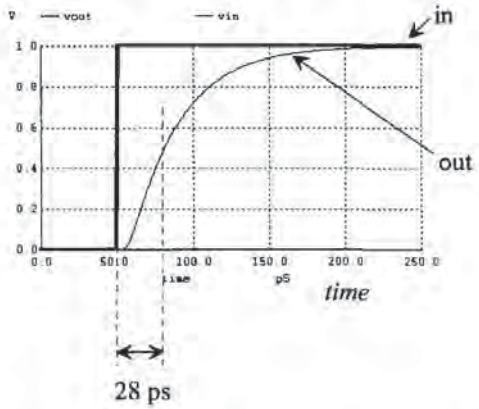
If we assume the signal is propagating in silicon dioxide ($SiO_2$) with a relative dielectric constant of roughly 4, then we can estimate the delay of the metal line as

$$\frac{t_d}{\text{meter}} = \frac{1}{v} = \frac{\sqrt{\varepsilon_r}}{c} = \frac{2}{3 \times 10^8\ m/s} = \frac{6.7\ ns}{\text{meter}} \tag{3.4}$$

or a delay of 6.7 ps/mm. For any practical integrated circuit wire in bulk CMOS, the parasitics (RC delay) dominate the propagation delays.

Note that increasing the width of the wire decreases its resistance and increases its capacitance (resulting in the delay staying relatively constant).

IPR2023-00697
Theta EX2019




```
*** Figure 3.7 CMOS: Circuit Design,
Layout, and Simulation ***
.control
destroy all
run
plot vin vout
.endc
.tran 1p 250p

O1 Vin 0 Vout 0 TRC
Rload Vout 0 1G
Vin vin 0 DC 0 pulse 0 1 50p 0
.model TRC ltra R=0.1 C=32e-18 len=5k
.end
```

**Figure 3.7** Simulating the delay through a 1 mm wire made using metal1.

**Example 3.4**

Estimate the capacitance between a 10 by 10 square piece of metal1 and an equal-size piece of metal2 placed exactly above the metal1 piece. Assume a scale factor of 50 nm. Sketch the layout and the cross-sectional views. Also sketch the symbol of a capacitor on the cross-sectional view.

The plate capacitance, from Table 3.1, between metal1 and metal2 is typically 35 aF/μm$^2$, while the fringe capacitance is typically 100 aF/μm. The two layers form a parallel plate capacitor, Fig. 3.8. The capacitance between the plates is given by the sum of the plate capacitance and the fringe capacitance, or

$$C_{12} = 100 \cdot (0.05)^2 \cdot 35\ aF + 40 \cdot (0.05) \cdot 100\ aF = 209\ aF$$

■

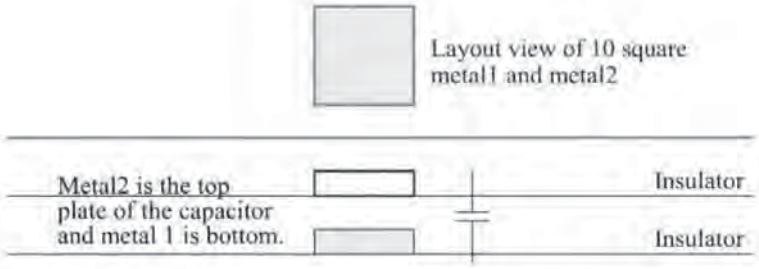



**Figure 3.8** Capacitance between metal1 and metal2.

**Example 3.5**

In the previous example, estimate the voltage change on metal1 when metal2 changes potential from 0 to 1 V. Verify the result with SPICE.

IPR2023-00697
Theta EX2019

The capacitance from metal2 to metal1 was calculated as 209 aF. The capacitance from metal1 to substrate is given by

$$C_{1sub} = 100 \cdot (0.05)^2 \cdot (23) + 40 \cdot (0.05) \cdot 79 = 164\ aF$$

The equivalent schematic is shown in Fig. 3.9. Since charge must be conserved we can write

$$C_{12} \cdot (\Delta V_{metal2} - \Delta V_{metal1}) = C_{1sub} \cdot \Delta V_{metal1}$$

The change in voltage on $C_{1sub}$ (metal1) is then given by

$$\Delta V_{metal1} = \Delta V_{metal2} \cdot \frac{C_{12}}{C_{12} + C_{1sub}} = 1 \cdot \frac{C_{12}}{C_{12} + C_{1sub}} = \frac{209}{209 + 164} = 560\ mV$$

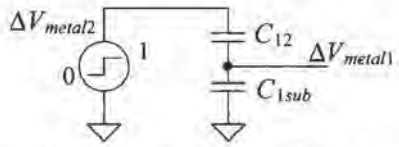
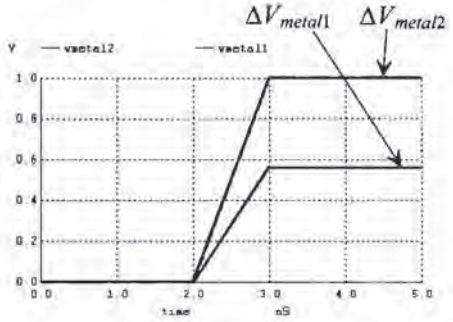



**Figure 3.9** Equivalent circuit used to calculate the change in metal1 voltage, see Ex. 3.5.

A displacement current flows through the capacitors, causing the potential on metal1 to change by 560 mV. This may seem significant at first glance. However, one must remember that most metal lines in a CMOS circuit are being driven from a low-impedance source; that is, the metal is not floating but is being held at some potential. This is not the case in some dynamic circuits or in circuits with high-impedance nodes or long metal runs. Figure 3.10 shows the SPICE simulation results and netlist. Notice how we used the "use initial conditions" (UIC) in the transient statement. This sets all nodes that aren't driven by a source to, initially (at the beginning of the simulation), zero volts. Note that older versions of SPICE don't recognize "a" as atto so we used "e-18." ■




```
*** Figure 3.10 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
plot vmetal2 vmetal1
.endc

.tran 10p 5n UIC

vmetal2 vmetal2 0 DC 0 pulse 0 1 2n 1n

C12 vmetal2 vmetal1 209e-18
C1sub vmetal1 0 164e-18

.end
```

**Figure 3.10** Simulating the operation of the circuit in Fig. 3.9.

IPR2023-00697
Theta EX2019

### 3.2.3 Current-Carrying Limitations

Now that we have some familiarity with the metal layers, we need to answer the question, "How much current can we carry on a given width or length of metal?" The factors that limit the amount of current on a metal wire or bus are metal electromigration and the maximum voltage drop across the wire or bus due to the resistance of the metal layer.

A conductor carrying too much current causes metal electromigration. This effect is similar to the erosion that occurs when a river carries too much water. The result is a change in the conductor dimensions, causing spots of higher resistance and eventually failure. If the current density is kept below the metal migration threshold current density, $J_{Al}$, metal electromigration will not occur. Typically, for aluminum, the current threshold for migration $J_{Al}$ is $1 \rightarrow 2 \frac{\text{mA}}{\mu\text{m}}$.

---

**Example 3.6**

Assuming a scale factor of 50 nm, estimate the maximum current a piece of metal1 with a drawn width of 3 can carry. Also estimate the maximum current a 100 by 100 $\mu m^2$ bonding pad can receive from a bonding wire. Assume that the metal wires are fabricated in aluminum.

The actual width of the metal1 wire in this example is 150 nm. Assuming that $J_{AL} = 1 \frac{\text{mA}}{\mu\text{m}}$, the maximum current on a 0.15 μm wide aluminum conductor is given by

$$I_{max} = J_{Al} \cdot W = 10^{-3} \cdot 0.15 = 150\ \mu A$$

The maximum current through a bonding pad is then 100 mA. ■

---

**Example 3.7**

Estimate the voltage drop across the conductor discussed in the previous example when the length of the conductor is 1 cm and the current flowing in the conductor is 150 μA ($I_{max}$).

The sheet resistance of metal1 is 0.1 Ω/square. The voltage drop across a metal1 wire that is 3 (0.15 μm) wide and 10,000 μm (1 cm) long carrying 150 μA is

$$V_{drop} = (0.1\ \Omega/\text{square}) \cdot \frac{10,000}{0.15} \cdot 150\ \mu A = 1\ V$$

or a significant voltage drop. If this conductor were used for power, we would want to increase the width significantly; however, if the conductor is used to route data, the size may be fine. ■

In general, the higher levels of metal (metal2, metal3, etc.) should be used for power routing. Metal2 is approximately twice as thick as metal1 and, therefore, has a lower sheet resistance. Metal3 is thicker than metal2, etc. When routing power, the more metal that is used, the fewer problems, in general, that will be encountered. If possible, a ground or power plane should be used across the entire die (entire levels of metal are used for *VDD* and ground). The more capacitance between the power and ground buses, the harder it is to induce a voltage change on the power plane; that is, the DC voltages will not vary.

IPR2023-00697
Theta EX2019

### 3.2.4 Design Rules for the Metal Layers

The design rules for the metal1, via1, and metal2 layers are seen in Fig. 3.11. Note that the via1 size must be exactly 1.5 by 1.5. Also note that the minimum allowable spacing between two wires using metal1 is 1.5, while the spacing between wires using metal2 is 2. There isn't a spacing rule between metal1 and metal2 because wires made with metal1 and metal2 are isolated by an insulator (sometimes called an interlayer dielectric, ILD).

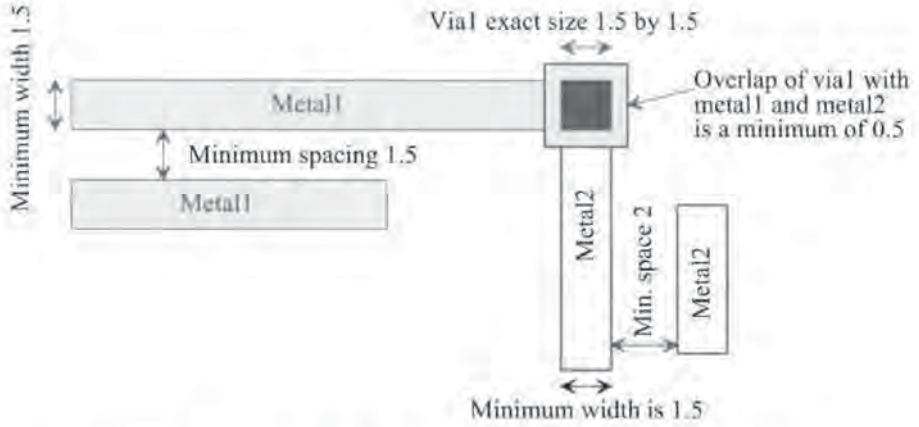



**Figure 3.11** Design rules for the metal layers using the CMOSEDU rules.

*Layout of Two Shapes or a Single Shape*

When learning to do layout, one may wonder about the equivalence of the two layouts seen in Fig. 3.12. In (a) two boxes are used while in (b) a single box is used. When the masks are made the layouts are equivalent.



(a) Layout using two boxes (b) Layout using a single box

**Figure 3.12** Equivalence of layouts drawn with a different number of shapes.

*A Layout Trick for the Metal Layers*

Notice that the size of the via is exactly 1.5 by 1.5 and that the minimum metal surrounding the via is 0.5. In order to save time when doing layout, a cell can be made called "via1." Instead of drawing boxes on the via1, metal1, and metal2 layers each time a connection between metal1 and metal2 is needed, we simply place the "via1" cell into the layout, Fig. 3.13.

IPR2023-00697
Theta EX2019



Three boxes: a via1 box that is 1.5 by 1.5, a metal1 box 2.5 by 2.5, and a metal2 box directly placed on the metal1 box that is 2.5 by 2.5.




Cross-sectional view of the via1 cell.

**Figure 3.13** Via1 cell with a rank of 1.

### 3.2.5 Contact Resistance

Associated with any contact to metal (or any other layer in a CMOS process for that matter) is an associated contact resistance. For the examples using metal layers in this book, we'll use a contact resistance of **10 Ω/contact**. Consider the following example.

**Example 3.8**

Sketch the equivalent electrical schematic for the layout depicted in Fig. 3.14a showing the via contact resistance. Estimate the voltage drop across the contact resistance of the via when 1 mA flows through the via. Repeat for the layout shown in Fig. 3.14b.

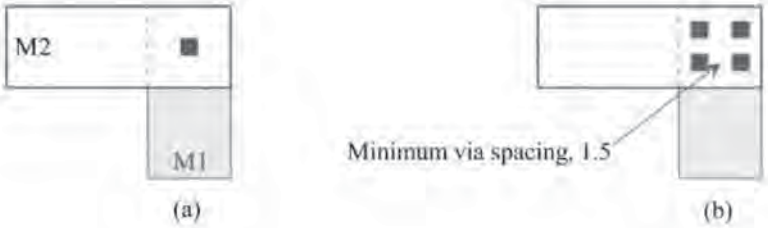
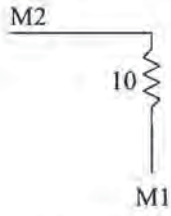


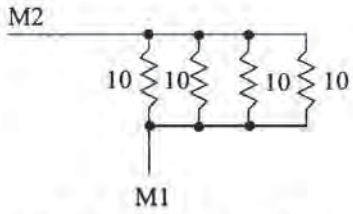

**Figure 3.14** Layouts used in Ex. 3.8.

The equivalent schematics are shown in Fig. 3.15a and (b) for the layouts in Figs. 3.14a and (b) respectively. If the via contact resistance is 10 Ω, and 1 mA flows through the via in (a), then a voltage drop of 10 mV results. Further, the reliability of the single via will be poor with 1 mA flowing through it due to electromigration effects. A "rule-of-thumb" is to allow no more than 100 μA of current flow per via. The four vias shown in Fig. 3.14(b) give an effective contact resistance of 10/4 or 2.5 Ω because the contact resistances of each of the vias are in parallel. The voltage drop across the vias decreases to 2.5 mV with 1 mA flowing in the wires. Increasing the metal overlap and the number of vias will further decrease the voltage drop (and electromigration effects). ■

IPR2023-00697
Theta EX2019




(a) The contact resistance of the via in Fig. 3.14a.




(b) The contact resistance of the four vias in Fig. 3.14b.

**Figure 3.15** The schematics of the contact resistances for the layouts in Fig. 3.14.

## 3.3 Crosstalk and Ground Bounce

Crosstalk is a term used to describe an unwanted interference from one conductor to another. Between two conductors there exists mutual capacitance and inductance, which give rise to signal feedthrough. Ground bounce (and *VDD* droop) are terms describing local variations in the power and ground supplies at a circuit. While crosstalk is only a problem for time-varying signals in a circuit, ground bounce can be problematic for both time varying and DC signals.

### 3.3.1 Crosstalk

Consider the two metal wires shown in Fig. 3.16. A signal voltage propagating on one of the conductors couples current onto the conductor. This current can be estimated using

$$I_m = C_m \frac{dV_A}{dt} \tag{3.5}$$

where $C_m$ is the mutual capacitance, $I_m$ is the coupled current, and $V_A$ is the signal voltage on the source conductor. Treating the capacitance between the two conductors in this

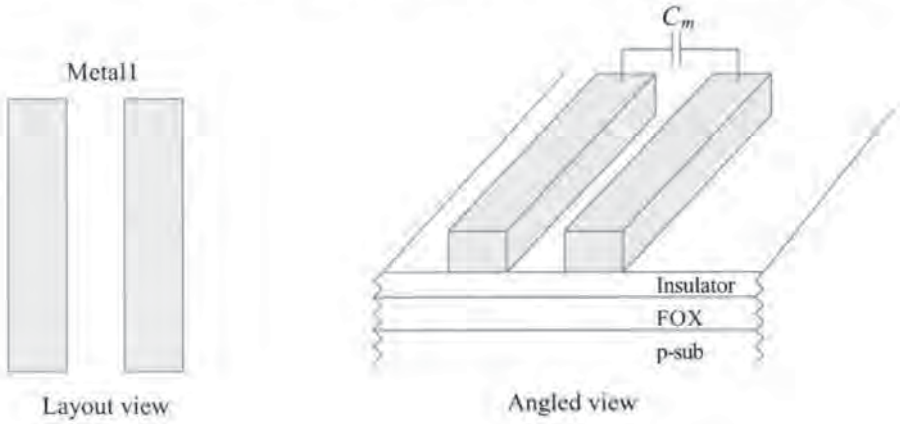



**Figure 3.16** Conductors used to illustrate crosstalk.

IPR2023-00697
Theta EX2019

simple manner is useful in most cases. Determining $C_m$ experimentally proceeds by applying a step voltage to one conductor while measuring the coupled voltage on the adjacent conductor. Since we know the capacitance of any conductor to substrate (see Table 3.1), we can write

$$\Delta V = V_A \cdot \frac{C_m}{C_m + C_{1sub}} \tag{3.6}$$

where $\Delta V$ is the coupled noise voltage to the adjacent conductor and $C_{1sub}$ is the capacitance of the adjacent conductor (in this case metal1) to ground (the substrate).

The adjacent metal lines shown in Fig. 3.16 also exhibit a mutual inductance. The effect can be thought of as connecting a miniature transformer between the two conductors. A current flowing on one of the conductors induces a voltage on the other conductor. Measuring the mutual inductance begins by injecting a current into one of the conductors. The voltage on the other conductor is measured. The mutual inductance is determined using

$$V_m = L_m \frac{dI_A}{dt} \tag{3.7}$$

where $I_A$ is the injected (time-varying) current (the input signal), $V_m$ is the induced voltage (the output signal), and $L_m$ is the mutual inductance.

Crosstalk can be reduced by increasing the distance between adjacent conductors. In many applications (e.g., DRAM), the design engineer has no control over the spacing (pitch) between conductors. The circuit designer then attempts to balance the signals on adjacent conductors (see, for example, the open and folded architectures in Ch. 16 concerning DRAM design).

### 3.3.2 Ground Bounce

*DC Problems*

Consider the schematic seen in Fig. 3.17a. In this schematic a circuit is connected to *VDD* and ground through two wires measuring 10,000 μm (10 mm) by 150 nm (with a resistance of 6.67 kΩ). Next consider, in (b), what happens if the circuit starts to pull a DC current of 50 μA. Instead of the circuit being connected to a *VDD* of 1 V the actual *VDD* drops to 667 mV. Further, the actual "ground" connected to the circuit increases to 333 mV. The voltage dropped across the circuit is the difference between the applied *VDD* and ground or only 333 mV (considerably less than the ideal 1 V). The obvious solution to making the supplied *VDD* and ground move closer to the ideal values is to increase the widths of the conductors supplying and returning currents to the circuit. This reduces the series resistance. The key point here is that *VDD* and ground are not fixed values; rather, they can vary depending how the circuit is laid out.

*AC Problems*

It is common, in CMOS circuit design, for a CMOS circuit to draw practically zero current in a static state (not doing anything). This is why, for example, it's possible to use solar power in a CMOS-based calculator. In this situation, conductors with small widths, as those in Fig. 3.17, may be fine. However, consider what happens if the circuit, for a short time, pulls 50 μA. As discussed above, the ground bounces up and *VDD* droops

IPR2023-00697
Theta EX2019

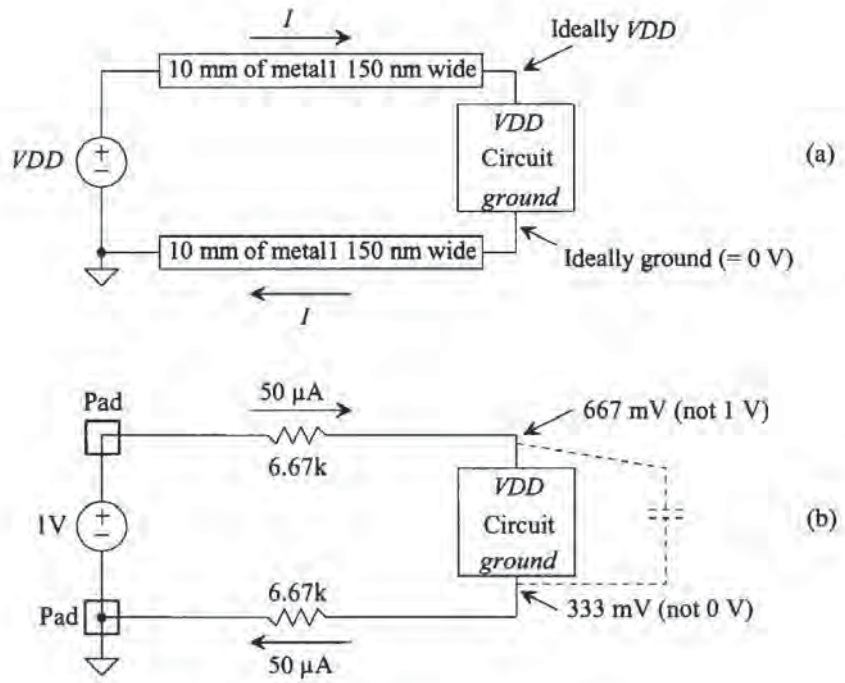



**Figure 3.17** Illustrating problems with incorrectly sized conductors.

down during this short time. The average current supplied by *VDD* may be well under a microamp; however, the occasional need for 50 μA still creates or causes problems. To circumvent these, consider adding an on-chip *decoupling* capacitor physically at the circuit between *VDD* and ground (see dotted lines in Fig. 3.17b). The added capacitor supplies the needed charge during the transient times and keeps the voltage applied across the circuit at *VDD*. Note that a decoupling capacitor should be used external to the chip as well. The capacitor is placed across the *VDD* and ground pins of the chip.

**Example 3.9**

Suppose that the circuit in Fig. 3.17b needs 50 μA of current for 10 ns. Estimate the size of the decoupling capacitor required if the voltage across the circuit should change by no more than 10 mV during this time.

We can write the charge supplied by the capacitor as

$$Q = I \cdot \Delta t = (50\ \mu A) \cdot 10\ ns = 500 \times 10^{-15}\ Coulumbs$$

The decoupling capacitor must supply this charge

$$\Delta V \cdot C = Q \rightarrow C \geq \frac{Q}{\Delta V} = \frac{I \cdot \Delta t}{\Delta V} = \frac{500 \times 10^{-15}}{10\ mV} \rightarrow C \geq 50\ pF \qquad (3.8)$$

A reasonably large capacitor. ■

IPR2023-00697
Theta EX2019

**Example 3.10**

To drive off-chip loads, an output buffer (see Ch. 11) is usually placed in between the on-chip logic and the large off-chip load, Fig. 3.18. If *VDD* is 1 V and it is desirable to drive the 30 pF off-chip load in Fig. 3.18 to 900 mV in 1 ns, estimate the size of the decoupling capacitor required. Assume that ground variations are not a concern.

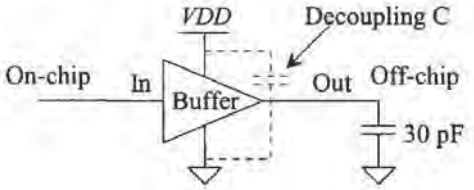



**Figure 3.18** Estimating the decoupling capacitance needed in an output buffer.

The charge supplied to the 30 pF capacitor (the load capacitance) by the output buffer is

$$Q = (900\ mV) \cdot (30\ pF) = 27\ pC$$

This charge is supplied by the decoupling capacitor (assuming that the conductors powering the buffer are narrow). Initially, the decoupling capacitor is charged to *VDD* (1 V). If *VDD* (actually the voltage across the decoupling capacitor) drops to 900 mV, then using Eq. (3.8) we can calculate the size of decoupling capacitor as

$$C \geq \frac{27\ pC}{100\ mV} = 270\ pF\ !!!$$

Not a practical value for an on-chip capacitor in most situations. The solution to this problem is to supply *VDD* and ground to the output buffers through wide conductors. Often separate power and ground pads (and very wide wires) are used to power the output buffers separately from the other on-chip circuitry. Using separate pads reduces the size of the decoupling capacitor required and eliminates the noise (ground bounce and *VDD* droop) from interfering with the operation of the other circuitry in the chip. Off-chip decoupling capacitors should still be used across the power and ground pins for the output buffers.

Note that if this buffer is running at 500 MHz (a clock period of 2 ns), the average current supplied to the load is

$$I_{avg} = \frac{27\ pC}{2\ ns} = 13.5\ mA$$

A significant value for a single chip output. ■

*A Final Comment*

It should be clear that some thought needs to go into the sizing of the metal layers and the number of vias used when transitioning from one metal layer to the next. Ignoring the parasitics associated with the wires used in an IC is an invitation for disaster.

IPR2023-00697
Theta EX2019

## 3.4 Layout Examples

In this section we provide some additional layout examples. In the first section we discuss laying out a pad and a padframe. In the following section we discuss laying out test structures to measure the parasitics associated with the metal layers.

### 3.4.1 Laying out the Pad II

Let's say we want to lay out a chip in a 50 nm process. Further let's say that the final die size (chip size) must be approximately 1 mm on a side with a pad size of 100 μm square (again, the pads can be smaller depending on the process). From the MOSIS design rules, the distance between pads must be at least 30 μm. Further let's assume a two-metal process (so metal2 is the top layer of metal the bonding wire drops down on). Table 3.2 summarizes the final and scaled sizes for our pads.

**Table 3.2** Sizes for an example 1 mm square chip with a scale factor of 50 nm.

| | **Final size** | **Scaled size** |
|---|---|---|
| Pad size | 100 μm by 100 μm | 2,000 by 2,000 |
| Pad spacing (center to center) | 130 μm | 2,600 |
| Number of pads on a side (corners empty) | 6 | 6 |
| Total number of pads | 24 | 24 |
| Overglass opening | 88 μm by 88 μm | 1,760 by 1,760 |

Let's start out by laying out a cell called "via1" like the one seen in Fig. 3.13. The resulting cell is seen in Fig. 3.19. We'll use this cell in our pad to connect metal1 to metal2. The bond wire will touch the top metal2. However, we'll place metal1 directly beneath the metal2 so that we can connect to the pad using either metal1 or metal2. The layout of the pad is seen in Fig. 3.20. The spacing between the pads is a minimum of 30 μm. We use an outline layer (no fabrication significance) to help when we place the pads together to form a padframe. We've assumed the distance from the pad metal to the edge of the chip is 15 μm.

Figure 3.21 shows the detail of how the overglass layer is placed in the pad metal area. Also seen in Fig. 3.21 is the placement of the via1 cell in Fig. 3.19 around the perimeter of the pad. This ensures metal1 is solidly shorted to metal2, Fig. 3.22.

Next let's calculate, assuming we want a chip size of approximately 1 mm on a side, the number of pads we can fit on the chip. The size of the pad in Fig. 3.20 is 130 μm square. To determine the number of pads we take the length of a side and divide by the size of a pad or

$$\#\text{ of pads} = \frac{1\ mm}{130\ \mu m} \approx 8 \qquad (3.9)$$

However, the corners don't contain a pad so the actual number of pads on a side is six as seen in Fig. 3.23. The CMOS circuits are placed in the area inside the padframe, while

IPR2023-00697
Theta EX2019

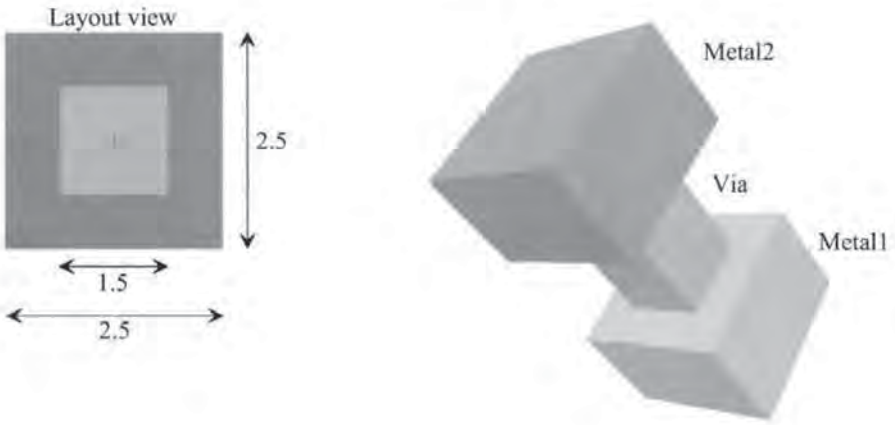



**Figure 3.19** Layout of a Via1 cell.

outside the padframe, the scribe, gets cut up when the chips are separated as seen in Fig. 1.2. Note that while this discussion assumed a two metal CMOS process it can be extended to a CMOS process with any number of metals.

The procedure to lay out probe pads, those pads that are not connected to a bonding wire but rather used for probing signals in, generally, unpackaged chips, follows basically the same procedure. The differences are that probe pads can reside anywhere on the chip and that they are smaller than bonding pads.

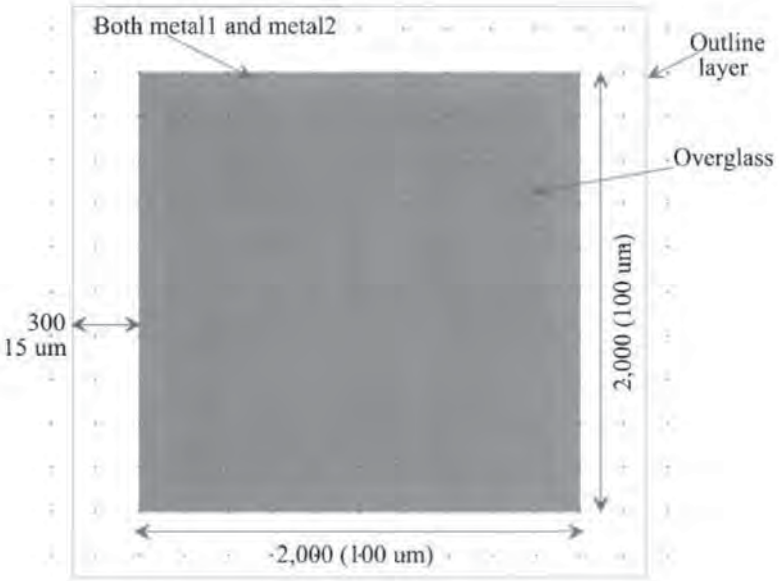



**Figure 3.20** Layout of the bonding pad.

IPR2023-00697
Theta EX2019

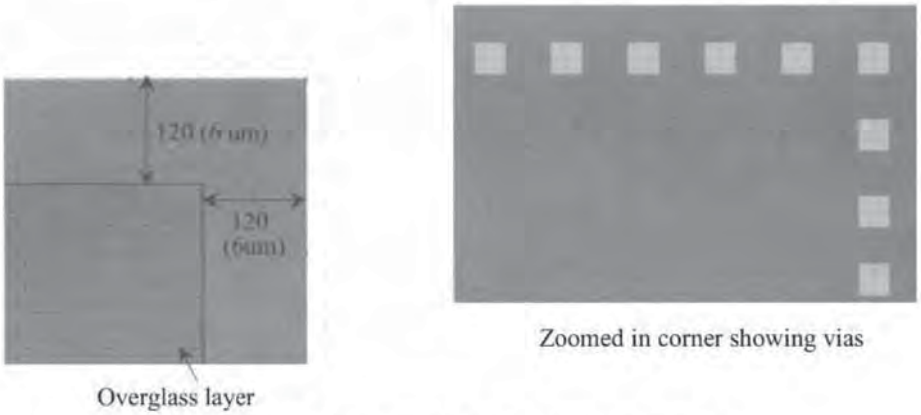



**Figure 3.21** Corner detail for the pad in Fig. 3.20.

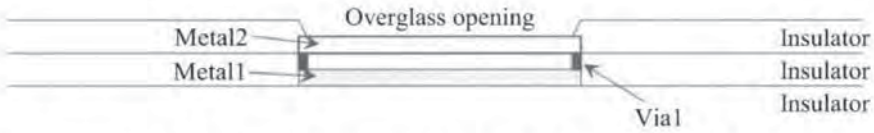



**Figure 3.22** Simplified cross-sectional view of the bonding pad discussed in this section.

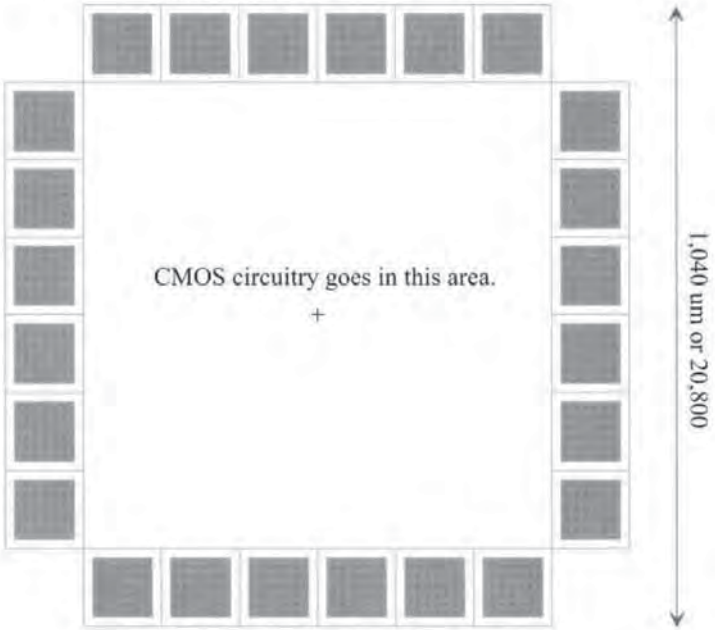



**Figure 3.23** The layout of a padframe.

IPR2023-00697
Theta EX2019

### 3.4.2 Laying Out Metal Test Structures

To characterize the sheet resistance, plate capacitance, fringe capacitance, and mutual capacitance associated with the metal layers, layouts called "test structures" are used. These layouts take one of two basic shapes. The first is the serpentine pattern seen in Fig. 3.24a (long perimeter while minimizing the area). This pattern is used for measuring sheet resistance or, with two serpentine layouts, mutual capacitance (see Figs. 3.16 and 3.24c). While it can be better to use a very long, straight, length of metal to measure resistance instead of a serpentine pattern (to avoid corners) the length of metal is generally limited by the chip size. By "snaking" the layout the length of metal can be made quite long. The layout seen in Fig. 3.24b, large area minimizing the perimeter, is useful for measuring plate capacitance. We don't use this type of layout to measure resistance because of the error associated with the connections to the metal.

To understand this last statement in more detail consider making a connection to points A and B in (a) to measure the line's resistance. A current is sent flowing in the metal line (say from A to B) and the voltage drop across the line is measured (hence why we can't use too short of a line [resistor], that is, the voltage drop would then be difficult to measure). At the source of current contact point, A, the current will spread out and then flow uniformly down the line where it converges to collection at the receiving contact point, B. In figure (b) the same thing happens; however, the height of the metal line is larger and thus allows for larger spreading/contraction resulting in more measurement error. Also, the wider width in (b) decreases the resistance, between points A and B, lowering the measured voltage and increasing the difficulty of the measurement.

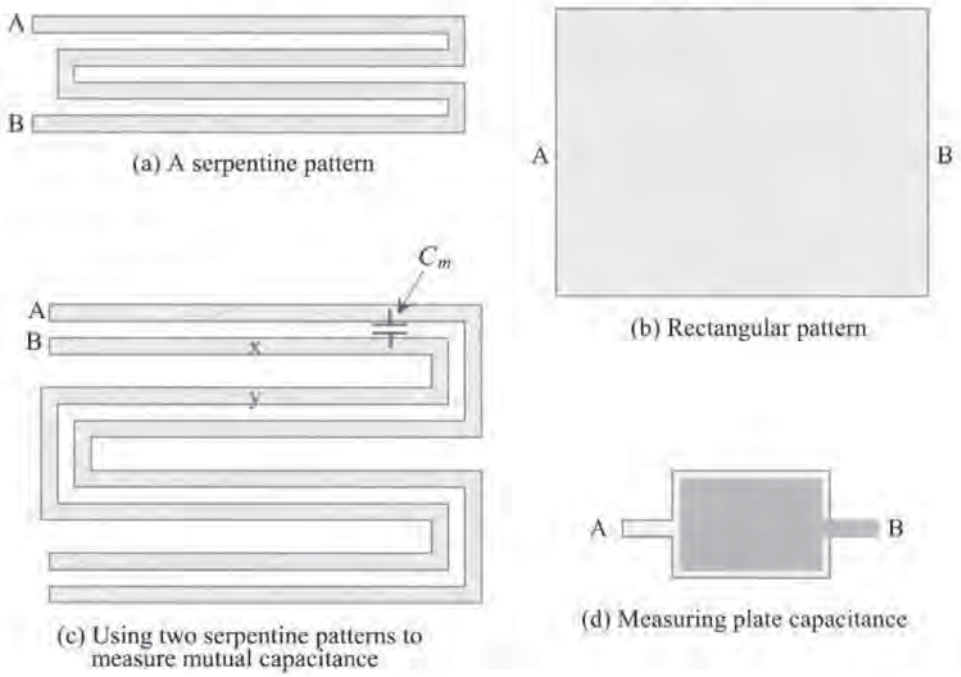



**Figure 3.24** Showing the layout of various patterns for measuring parasitics.

IPR2023-00697
Theta EX2019

In Fig. 3.24c two serpentine patterns are laid out adjacent to each other. This test structure is used to measure the mutual capacitance between like layers (as seen in Fig. 3.16). Again, a serpentine pattern is used to increase the measured variable (capacitance) between points A and B. Minimum spacing is used between the metal lines to maximize the capacitance and because this is the spacing where mutual capacitance has the most influence. To measure the capacitance a low frequency AC voltage is applied between A and B while the displacement current is measured. We need to use a low frequency source to avoid the distributed effects of the metal lines (the delay through the metal lines). Note that at low frequencies points x and y are at the same potential. The result is that the current we measure is restricted to (only) the displacement current between conductors A and B.

The test structure seen in Fig. 3.24d can be used to measure plate capacitance (the capacitance is measured between points A and B). Again large area structures are used to minimize the effects of the perimeter (fringe) capacitance. The test structure can be drawn so that both layers are the same size. To measure the fringe capacitance between two layers the rectangles in (d) are replaced with serpentine structures.

*SEM View of Metal*

Figure 3.25 shows an SEM image of a portion, a layout view, of a CMOS memory chip. The brighter areas of the image are metal1 while the bright, and circular-shaped, objects are contacts (discussed in the next chapter). Notice that none of the sections of metal or contacts are square or rectangular. While the layout can be square or rectangular the actual fabricated metal layers show rounding (tools such as optical proximity correction, OPC, are used for corrections). If the reader looks closely at the image, the misalignment between the metal and the contacts is seen (hence the reason for design rules).

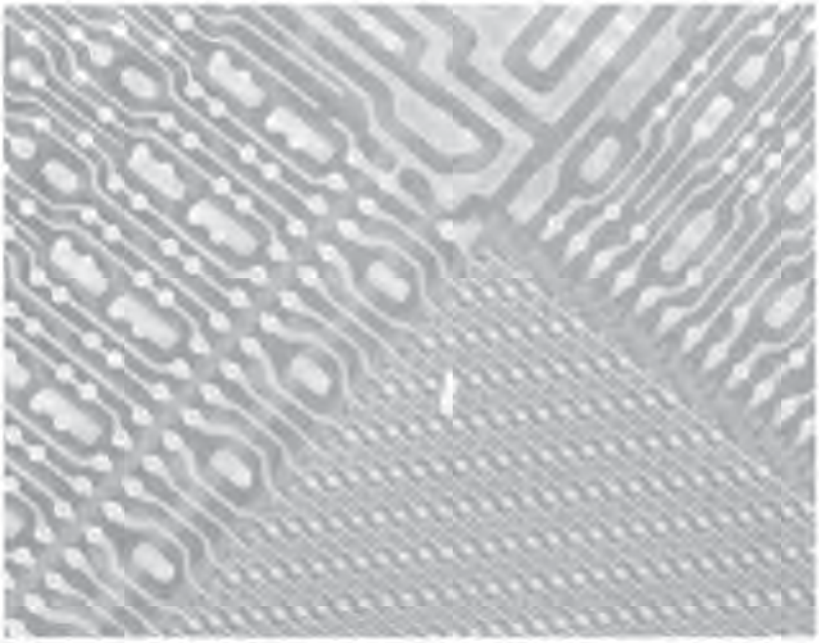


**Figure 3.25** SEM photo showing patterned metal layers.

IPR2023-00697
Theta EX2019

## ADDITIONAL READING

[1] R. S. Muller, T. I. Kamins, and M. Chan, *Device Electronics for Integrated Circuits*, John Wiley and Sons Publishers, 2002. ISBN 0-471-59398-2

[2] J. D. Plummer, M. D. Deal, and P. B. Griffin, *Silicon VLSI Technology, Fundamentals, Practice, and Modeling*, Prentice-Hall Publishers, 2000. ISBN 0-13-085037-3

## PROBLEMS

Unless otherwise indicated, use the data from Table 3.1, a metal sheet resistance of 0.1 Ω/square, and a metal contact resistance of 10 Ω.

**3.1** Redraw the layout and cross-sectional views of a pad, similar to Fig. 3.2, if the final pad size is 50 μm by 75 μm with a scale factor of 100 nm.

**3.2** Estimate the capacitance to ground of the pad in Fig. 3.20 made with both metal1 and metal2.

**3.3** Suppose a parallel plate capacitor was made by placing a 100 μm square piece of metal1 directly below the metal2 in Fig. 3.2. Estimate the capacitance between the two plates of the capacitor (metal1 and metal2). Estimate the capacitance from metal1 to substrate. The unwanted parasitic capacitance from metal1 to substrate is often called the **bottom plate** parasitic.

**3.4** Sketch the cross-sectional view for the layout seen in Fig. 3.26.

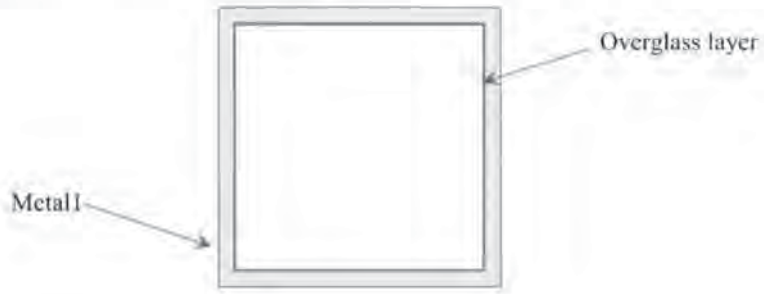



**Figure 3.26** Layout used in Problem 3.4.

**3.5** Sketch the cross-sectional view, at the dashed line, for the layout seen in Fig. 3.27. What is the contact resistance between metal3 and metal2?

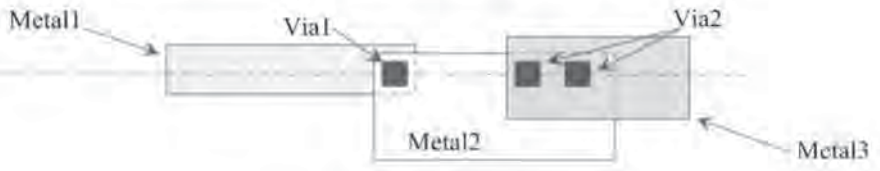



**Figure 3.27** Layout for Problem 3.5.

IPR2023-00697
Theta EX2019

**3.6** The insulator used between the metal layers (the interlayer dielectric, ILD) can have a relative dielectric constant well under the relative dielectric constant of $SiO_2$ (= 4). Estimate the intrinsic propagation delay through a metal line encapsulated in an ILD with a relative dielectric constant of 1.5. What value of metal sheet resistance, using the values from Ex. 3.3, would be required if the RC delay through the metal line is equal to the intrinsic delay?

**3.7** Using $CV = Q$, rederive the results in Ex. 3.5.

**3.8** For the layout seen in Fig. 3.28, sketch the cross-sectional view (along the dotted line) and estimate the resistance between points A and B. Remember that a via is sized 1.5 by 1.5.

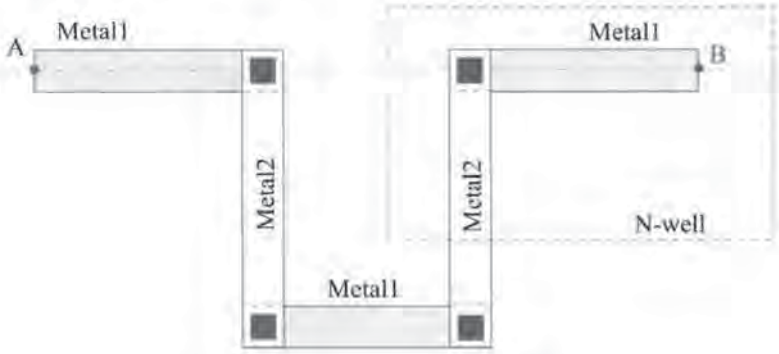



**Figure 3.28** Layout for Problem 3.8.

**3.9** Laying out two metal wires directly next to each other, and with minimum spacing, for a long distance increases the capacitance between the two conductors, $C_m$. If the two conductors are *VDD* and ground, is this a good idea? Why or why not?

**3.10** Consider the schematic seen in Fig. 3.29. This circuit can be used to model ground bounce and *VDD* droop. Show, using SPICE, that a decoupling capacitor can be used to reduce these effects for various amplitude and duration current pulses.

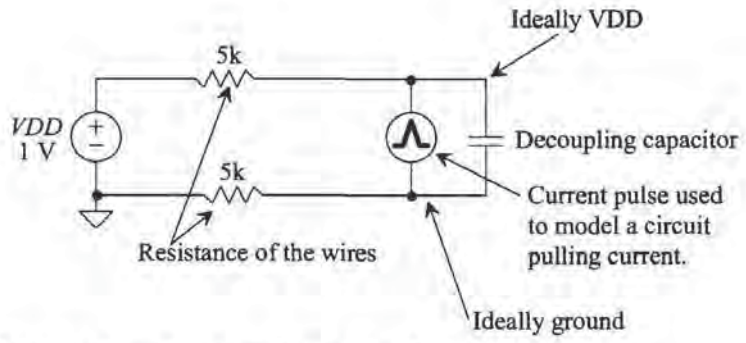



**Figure 3.29** Circuit used to show the benefits of a decoupling capacitor.

IPR2023-00697
Theta EX2019

**3.11** Lay out the padframe specified by the information in Table 3.2. Assume a 3 metal CMOS process is used. Comment on how the scale factor affects the (drawn) layout size.

**3.12** Propose, and lay out, a test structure to measure the sheet resistance of metal3. Comment on the trade-offs between accuracy and layout size. Using your test structure provide a numerical example of calculating sheet resistance for metal3.

IPR2023-00697
Theta EX2019

Chapter
4

# The Active and Poly Layers

The active, n-select, p-select, and poly layers are used to form n-channel and p-channel MOSFETs (NMOS and PMOS respectively) and so metal1 can make an ohmic contact to the substrate or well. The active layer, in a layout program, defines openings in the silicon dioxide covering the substrate (see Figs. 2.3 and 2.4). The n-select and p-select layers indicate where to implant n-type or p-type atoms, respectively. The active and select layers are always used together. The active defines an opening in the oxide and the select then dopes the semiconductor in the opening either n-type or p-type.

The poly layer forms the gate of the MOSFETs. Poly is a short name for polysilicon (not to be confused with the poly, or polygon, object in a layout program). Polysilicon is made up of small crystalline regions of silicon. Therefore, in the strictest sense, poly is not amorphous silicon (randomly organized atoms), and it is not crystalline silicon (an orderly arrangement of atoms in the material) such as the wafer.

## 4.1 Layout using the Active and Poly Layers

We've covered the following fabrication layers in Chs. 2 and 3: n-well, metal1, via1, metal2, and overglass. In this section we cover the following additional fabrication layers: active, n-select, p-select, poly1, silicide block, and contact.

*The Active Layer*

Examine the layout of a box and the corresponding angled view (the fabrication results) seen in Fig. 4.1. The box is drawn on the active layer and indicates where to open a hole in the field oxide (FOX). These openings are called active areas. The field area (the area that isn't the active area, which is the area where the FOX is grown) is used for routing wires (connecting the circuit together). The MOSFETs are fabricated in the bulk (the p-substrate or the n-well) in these active openings. The FOX is used to isolate the devices from one another (the active areas are isolated by the FOX). Note that there will be some resistive connection between active areas (either through the substrate or the n-well). However, the FOX is grown thick enough to keep the interactions between adjacent active areas to a minimum.

IPR2023-00697
Theta EX2019

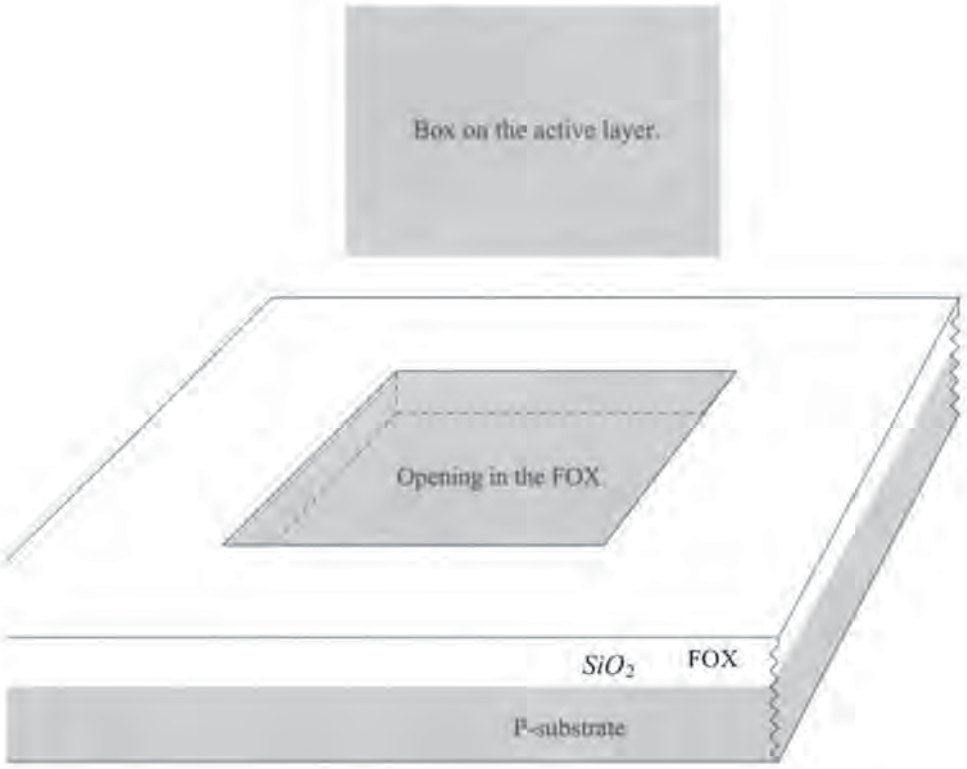



**Figure 4.1** How the active layer specifies where to open holes in the field oxide (FOX).

*The P- and N-Select Layers*

Surrounding the active layer with either the n-select or p-select layers dopes the semiconductor n- or p-type. Figure 4.2 shows several combinations of selects, n-well, and active layers. In (a) and (b) for example, the opening in the FOX is implanted p-type (in the location determined by the p-select mask). When learning to do design and layout, it's important to be able to see a layout and then visualize the corresponding cross-sectional views.

Also seen in this figure (see 4.2i and j) is how a single layer (called the n+ layer) can be used instead of two layers (the active and n-select layers). The n+ layer in (j) is used directly for the active mask (openings in the FOX) in (i). The n-select in (i) is a *derived* mask. It is derived by bloating the size of the n+ layer. The n-select mask must be larger than the active mask due to misalignment. If the implant (select) isn't aligned directly over the active opening in the FOX, then the semiconductor exposed in the active opening won't get doped. If the select and active masks could be aligned perfectly, the select layers wouldn't need to be larger than the active layer. Also note that using a select without active causes the implant to bombard the FOX. Since the FOX is thick, it keeps the implanted atoms from reaching the substrate.

IPR2023-00697
Theta EX2019

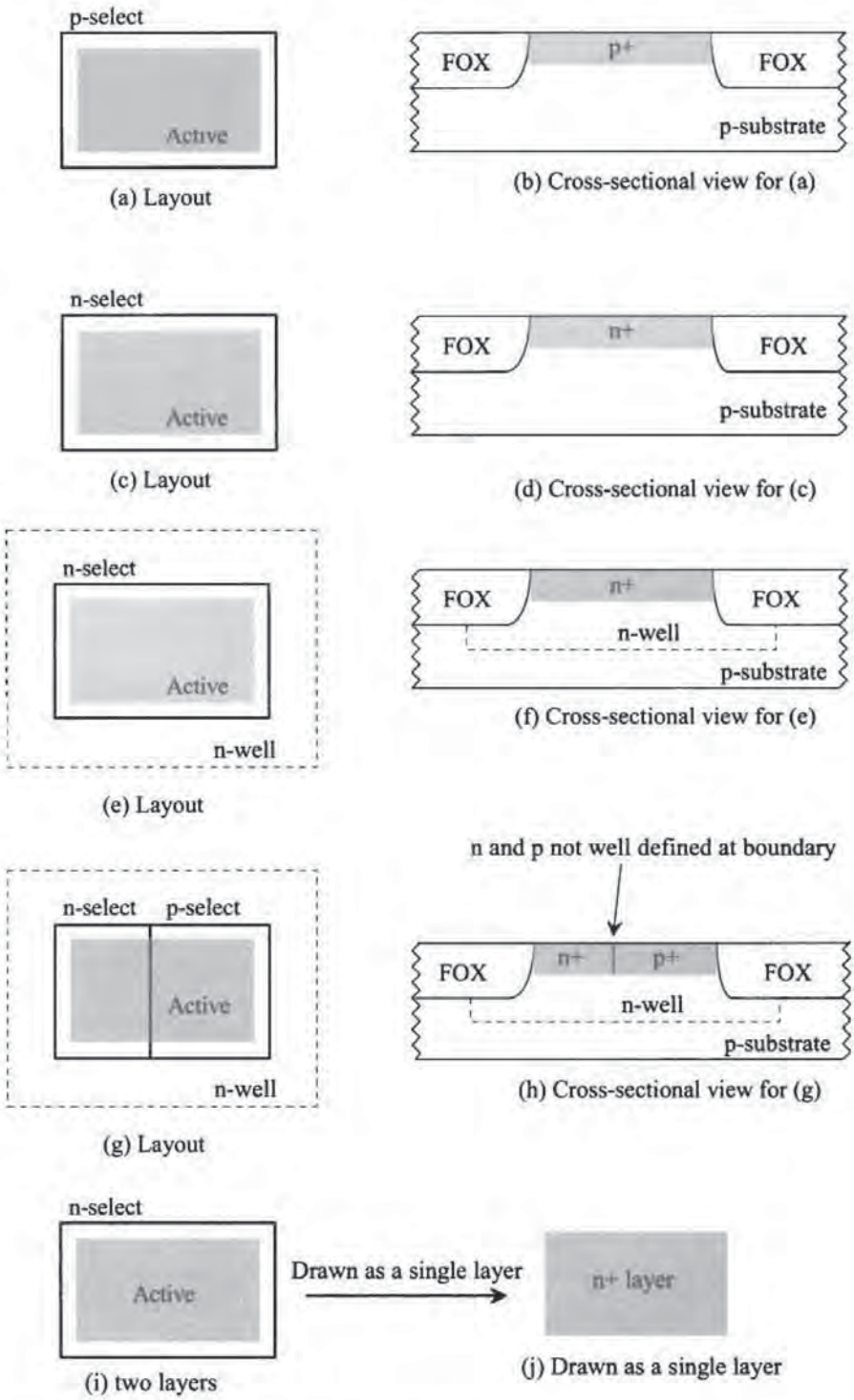



**Figure 4.2** Combinations of active, selects, and n-wells.

IPR2023-00697
Theta EX2019

*The Poly Layer*

The poly layer is used for MOSFET formation. Consider the layout seen in Fig. 4.3a. Drawing poly over active produces a MOSFET layout. If we see a complicated layout, it is straightforward to determine the number of MOSFETs in the layout simply by counting how many times poly crosses active. Note that the gate of the MOSFET is formed with the polysilicon, and the source and drain of the MOSFET are formed with the n+ implant. Further note that the source and drain of an integrated MOSFET (in a general CMOS process) are interchangeable. We are not showing the (required) contact to the substrate (the body of the MOSFET). The body connection will be covered in a moment (the MOSFET is a four-terminal device).

*Self-Aligned Gate*

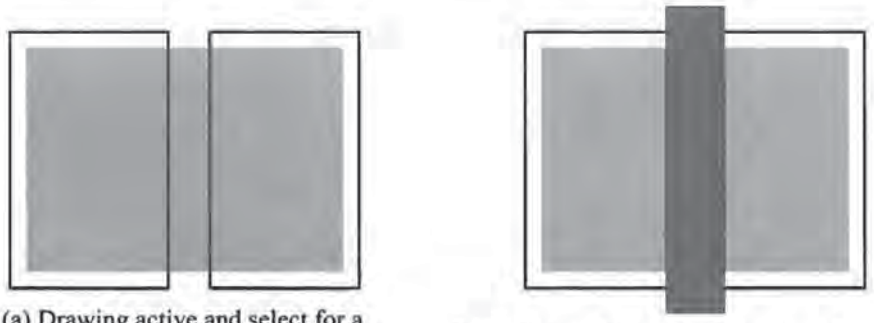
Notice how, Fig. 4.3b or c, the area under the poly gate isn't doped n+. After the opening in the FOX is formed with the active mask, a thin insulating oxide is grown over the opening. This is the MOSFET's gate oxide (GOX), Fig. 4.3b. Next, the poly mask specifies where to deposit the polysilicon gate material. This is followed by applying the implant in the areas specified by the n-select mask. The implant easily penetrates through the thin GOX into the source and drain areas. However, the polysilicon gate acts like a mask to keep the n+ atoms from penetrating under the MOSFET's gate (the poly is made thick enough to ensure the implant doesn't reach the GOX). Also, the drain and gate become self-aligned to the source/drain of the MOSFET. This is important because we know we can't perfectly align the poly mask to the active masks.

---

**Example 4.1**

Comment on the problems with the MOSFET layout seen in Fig. Ex4.1.

In Fig. Ex4.1a the active layer defines an opening in the field oxide. The select masks are placed exactly where the desired n+ implants will occur, as seen in Fig. 4.3. However, due to shifts in the select mask, relative to the poly mask, the area directly next to the gate will not get implanted. (Redraw Fig. Ex. 4.1b with the poly layer shifted left or right.) Notice how the incorrect layout in Fig. Ex4.1b looks exactly the same as the correct layout in Fig. 4.3a. ■

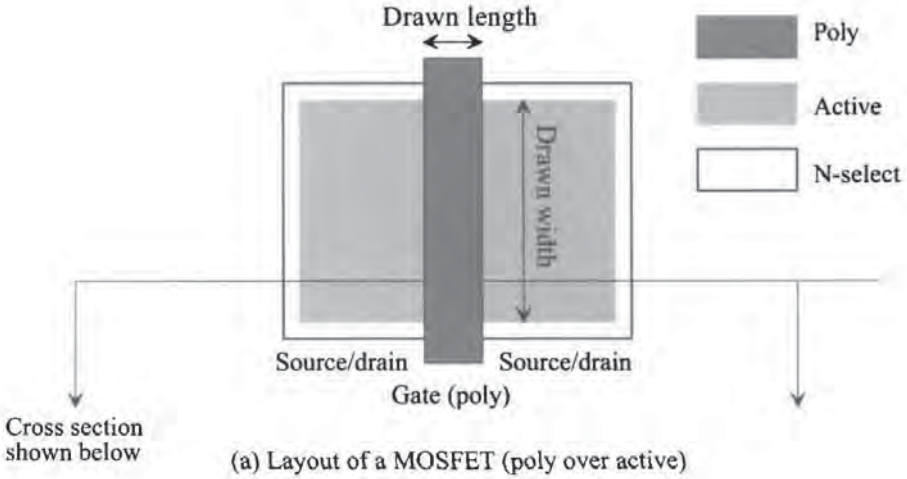


(a) Drawing active and select for a MOSFET (bad).

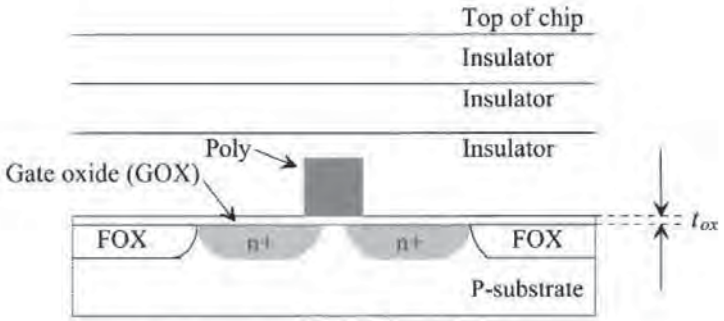
(b) Placing poly over the layout in (a) (bad).

**Figure Ex4.1** Bad layout examples (what NOT to do).

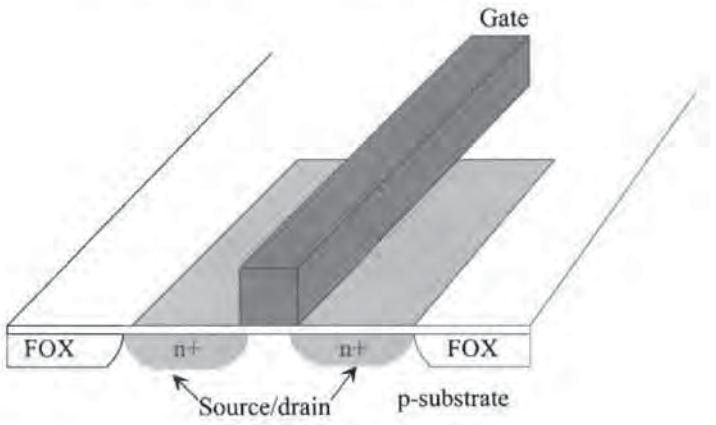
IPR2023-00697
Theta EX2019




(a) Layout of a MOSFET (poly over active)




(b) Cross-sectional view of the layout in (a).




(c) Angled view of the layout in (a)

**Figure 4.3** Layout and cross-sectional views of a MOSFET.

IPR2023-00697
Theta EX2019

*The Poly Wire*

The poly layer can also be used, like metal1, as a wire. Poly is routed on top of the FOX. The main limitation when using the poly layer for interconnection is its sheet resistance. As we saw in the last chapter, the sheet resistance of the metal layers is approximately 0.1 Ω/square. The sheet resistance of the doped poly can be on the order of 200 Ω/square. The capacitance to substrate is also larger for poly simply because it is closer to the substrate (see Table 3.1). Therefore, the delay through a poly line can be considerably longer than the delay through a metal line. To reduce the sheet resistance of poly (and of the implanted active regions), a silicide (a material that is a mixture of silicon and a refractory metal like tungsten) is deposited over the MOSFET and field region, Fig. 4.4. The silicide and poly gate sandwich is called a polycide.

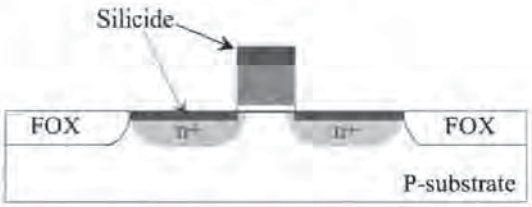



**Figure 4.4** How the gate and drain/source of a MOSFET are silicided to reduce sheet resistance.

Table 4.1 gives some typical values of sheet resistance, $R_{square}$, for well, poly, n+, and p+ in a nm CMOS process. When we use the poly as a mask to self-align the source and drain regions of the MOSFET to the gate, Fig. 4.3, we dope the poly either n-type (for an NMOS device) or p-type (for a PMOS device). Silicide is then used to avoid forming a pn-junction (diode) when the PMOS and NMOS polysilicon gates are connected together in the field region (the silicide electrically shorts the n- and p-type poly gates together). The other specifications seen in the table will be discussed in the next chapter.

**Table 4.1** Typical properties of resistive materials in a nm CMOS process.

| Silicide | Resistor type | $R_{square}$ (ohms/sq) AVG. | *TCR*1 (ppm/C) AVG. | *TCR*2 (ppm/C²) AVG. | *VCR*1 (ppm/V) AVG. | *VCR*2 (ppm/V²) AVG. | Mismatch % $\Delta R/R$ |
|---|---|---|---|---|---|---|---|
| N/A | well | 500 ± 10 | 2400 ± 50 | 7 ± 0.5 | 8000 ± 200 | 500 ± 50 | < 0.1 |
| No | n+ poly | 200 ± 1 | 20 ± 10 | 0.6 ± 0.03 | 700 ± 50 | 150 ± 15 | < 0.5 |
| No | p+ poly | 400 ± 5 | 160 ± 10 | 0.8 ± 0.03 | 600 ± 50 | 150 ± 15 | < 0.2 |
| No | n+ | 100 ± 2 | 1500 ± 10 | 0.04 ± 0.1 | 2500 ± 50 | 350 ± 20 | < 0.4 |
| No | p+ | 125 ± 3 | 1400 ± 20 | 0.4 ± 0.1 | 80 ± 80 | 100 ± 25 | < 0.6 |
| Yes | n+ poly | 5 ± 0.3 | 3300 ± 90 | 1.0 ± 0.2 | 2500 ± 125 | 3800 ± 400 | < 0.4 |
| Yes | p+ poly | 7 ± 0.1 | 3600 ± 50 | 1.0 ± 0.2 | 2500 ± 400 | 5500 ± 250 | < 0.7 |
| Yes | n+ | 10 ± 0.1 | 3700 ± 50 | 1.0 ± 0.2 | 350 ± 150 | 600 ± 60 | < 1.0 |
| Yes | p+ | 20 ± 0.1 | 3800 ± 40 | 1.0 ± 0.2 | 150 ± 50 | 800 ± 40 | < 1.0 |

IPR2023-00697
Theta EX2019

*Silicide Block*

In some situations (as in making a resistor), it is desirable to keep from depositing the silicide on the gate poly or source/drain regions. A layer called the silicide block can be used for this purpose. Consider the following example.

**Example 4.2**
Estimate the delay through the poly wire in Fig. 4.5 with and without silicide. The width of the wire is 1 and the length is 1,000. Use a scale factor of 50 nm and the values for capacitance in Table 3.1. Simulate the delay using SPICE.

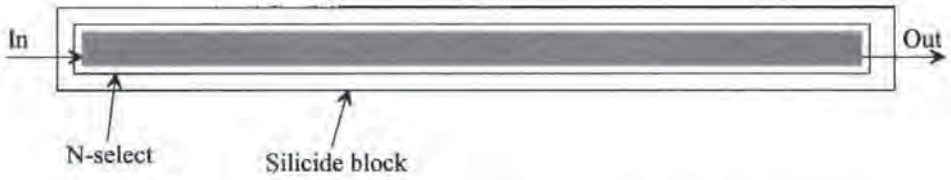



**Figure 4.5** Estimating the delay through a polysilicon line with and without a silicide.

The capacitance of the poly wire to substrate doesn't depend on the presence or absence of the silicide. Using the data from Table 3.1, we can estimate the capacitance of the poly wire to substrate as

$$C_{poly} = C_{plate} \cdot area_{drawn} \cdot (scale)^2 + C_{fringe} \cdot perimeter_{drawn} \cdot scale \qquad (4.1)$$

or

$$C_{poly} = (58\ aF) \cdot (1,000) \cdot (0.05)^2 + (88\ aF) \cdot (2,002) \cdot (0.05) \approx 9\ fF$$

The resistance of the poly wire is calculated using

$$R = R_{square} \cdot \frac{L}{W} \qquad (4.2)$$

From the data in Table 4.1, the resistance of the wire is either 200k (no silicide) or 5k (with silicide). The delays are then calculated as

$$t_d = 0.35 \cdot 9\ fF \cdot 200k = 630\ ps \text{ and } t_d = 0.35 \cdot 9\ fF \cdot 5k = 16\ ps$$

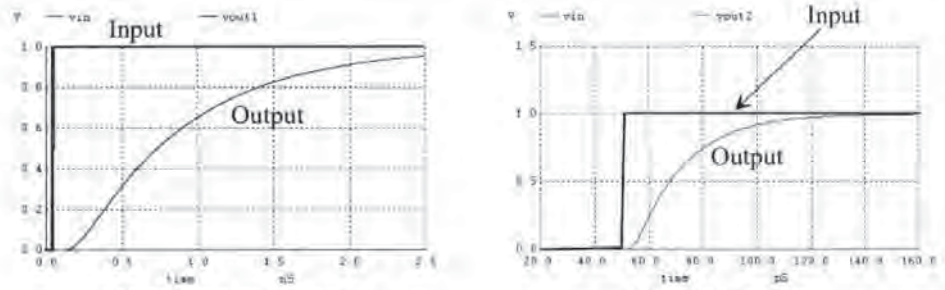
The simulation results are seen in Fig. 4.6. In the simulation netlist we divided the line up into 1,000 squares. The capacitance/square is 9 aF (remembering SPICE doesn't recognize "a" so we use e-18). The resistance/square is either 200 Ω (no silicide) or 5 Ω (with a silicide). ■

Note that the silicide block layer should not be placed under (surround) a contact to polysilicon or active else a rectifying contact may form (this is important).

### 4.1.1 Process Flow

A generic CMOS process flow is seen in Fig. 4.7. The fabrication of both PMOS and NMOS devices is detailed in this figure. This figure doesn't show the initial steps taken to fabricate the n-well (and/or p-well) but rather starts with the wells already fabricated.

IPR2023-00697
Theta EX2019




(a) Poly without silicide (b) Silicided poly

**Figure 4.6** Simulated delay through poly wires.

The first step, Fig. 4.7a, is to grow a thin pad oxide on top of the entire wafer. This is followed by depositing nitride (the pad oxide is used as a cushion for the nitride) and photoresist layers. The photoresist is then patterned using the active mask. The remaining photoresist, seen in Fig. 4.7a, ultimately defines the openings in the FOX.

In Fig. 4.7b the areas not covered by the photoresist are etched. The etching extends down into the wafer so that *shallow trenches* are formed. In (c) the shallow trenches are filled with $SiO_2$. These trenches isolate the active areas and form the field regions (FOX). This type of device isolation is called shallow trench isolation (STI).

In (d) two separate implants are performed to adjust the threshold voltages of the devices. A photoresist is patterned (twice) to select the areas for threshold voltage adjust.

Figure 4.7e shows the results after the deposition and patterning of polysilicon (for the MOSFET gate material). This is followed by several implants. In (f) we see a shallow implant to form the MOSFET's lightly doped drains (LDD). The LDD implants prevent the electric field directly next to the source/drain regions from becoming too high (this is discussed further in Ch. 6). Note that the poly gate is used as a mask during this step.

The next step is to grow a spacer oxide on the sides of the gate poly, Fig. 4.7g. After the spacer is grown, the n+/p+ implants are performed. This implant dopes the areas used for the source/drain of the MOSFETs as well as the gate poly. The last step is to silicide the source and drain regions of the MOSFET. This is important for reducing the sheet resistances of the polysilicon and n+/p+ materials, as indicated in Table 4.1.

Finally, note that the process sequence seen in Fig. 4.7 is often called, in the manufacturing process, the front-end of the line (FEOL). The fabrication of the metal layers and associated contacts/vias is called the back-end of the line (BEOL).

*Damascene Process Steps*

The process of: 1) making a trench, 2) (over) filling the trench with a material, and 3) grinding the material down until the top of the wafer is flat is called a Damascene process (a technique used, and invented, by craftsman in the city of Damascus to inlay gold or silver in swords). The STI process just described is a Damascene process. More often, though, the Damascene process is associated with the metal layers in a CMOS process. Trenches are formed in the insulators. Copper, for example, is then deposited in the trenches. The top of the wafer is then ground down until it is flat.

IPR2023-00697
Theta EX2019

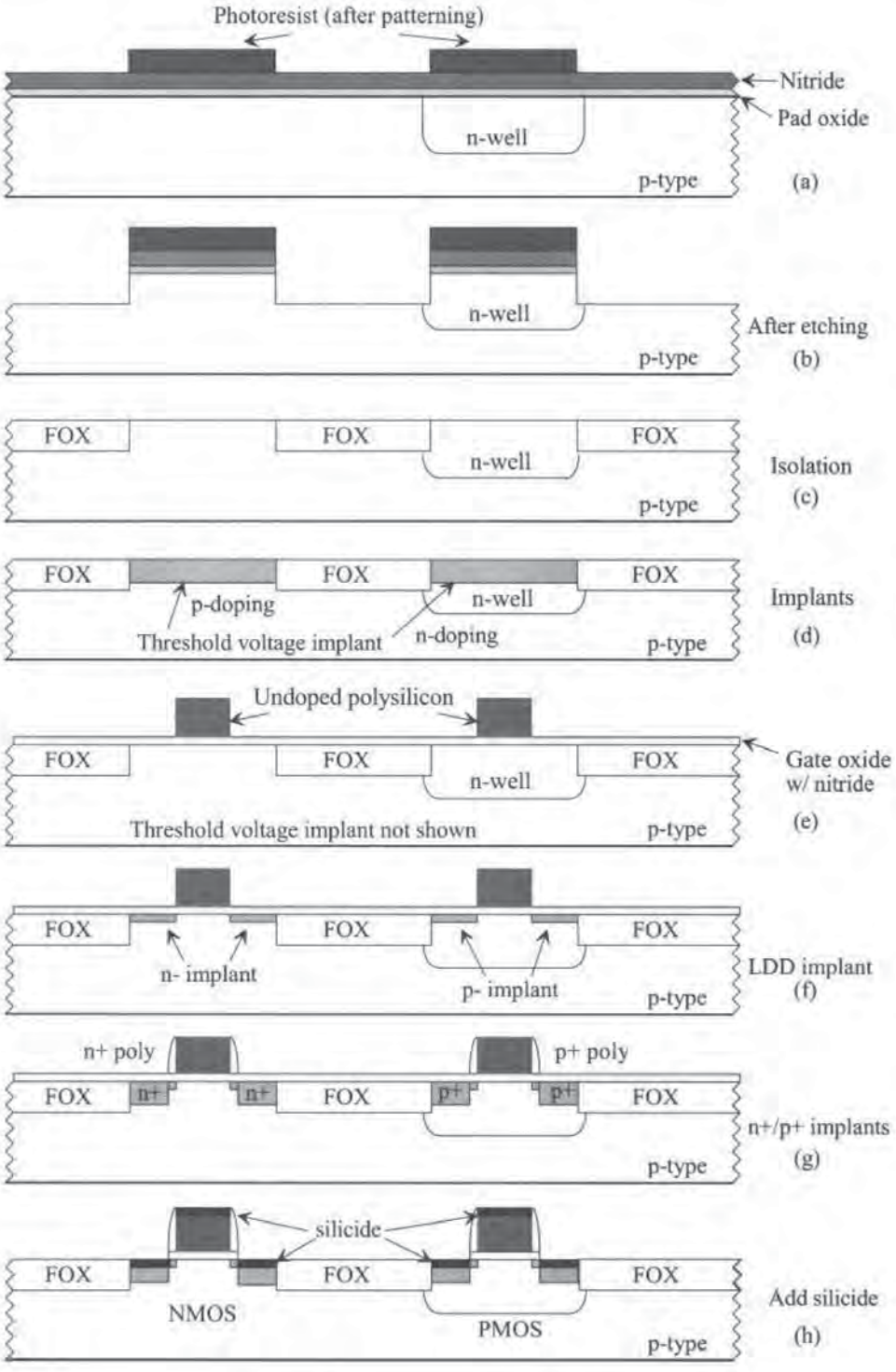


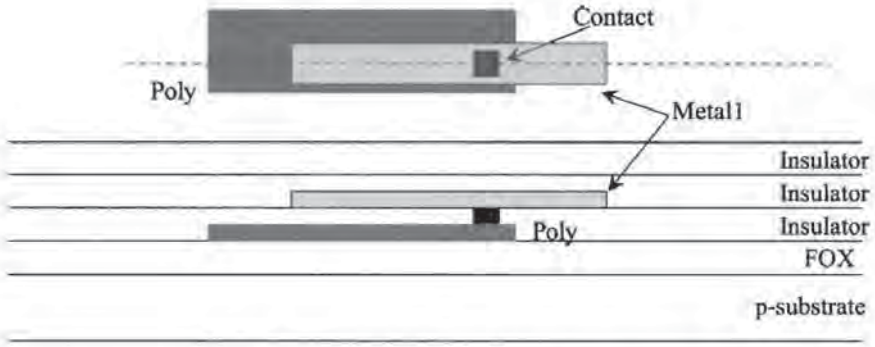

**Figure 4.7** General CMOS process flow.

IPR2023-00697
Theta EX2019

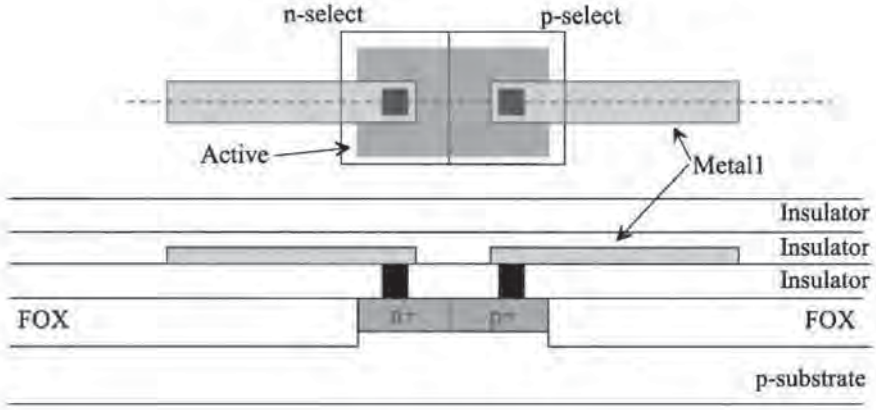
## 4.2 Connecting Wires to Poly and Active

In the last section we discussed how to lay out active areas and polysilicon. Here, in this section, let's discuss how to connect metal wires to poly and active. The contact layer connects metal1 to either active (n+/p+) or poly. Unless we want to form a rectifying contact (a Schottky diode), *we never connect metal directly to the substrate or well.* Further, we won't connect metal to poly without having the silicide in place. Never put a silicide block around a contact to poly.

Figure 4.8a shows a layout and corresponding cross-sectional view of the layers metal1, contact, and poly (a contact to poly). Figure 4.8b shows a connection to n+ and p+. Note that, like we did with the via cell in Fig. 3.13, we can layout contact cells to poly, n+, and p+. Further note that metal1 is connected to either metal2 (through via1) or poly/active. Metal2 can't be connected to active/poly without first connecting to metal1 and a contact.




(a) Metal1 connecting to poly through a contact.




(b) Contacts to active.

**Figure 4.8** How metal1 is connected to poly and active.

IPR2023-00697
Theta EX2019

When etching an opening for a contact to poly or active, an etchant stop layer is used. The etchant stop is put down directly on top of the FOX prior to depositing the insulator.

*Connecting the P-Substrate to Ground*

So far, throughout the book, we've said that the p-substrate is at ground potential. However, we haven't actually said how we connect the substrate to ground (the substrate must be connected to the ground pad through a wire). The substrate, as seen in Fig. 4.4, is the body of the NMOS devices and is common to all NMOS devices fabricated on the chip (assuming an n-well process, see Fig. 2.24). Towards connecting the substrate to ground, consider the layout seen in Fig. 4.9. Again note that we only connect metal1 to p+ (or n+/poly) and not directly to the substrate. Further notice that poly sits on the FOX while metal1 sits on the insulator above FOX.

An important consideration when "tying down the substrate" is the number of places, around the chip, the substrate is tied to ground. We don't just connect the substrate to ground with one connection, like the one seen in Fig. 4.9, and assume that the entire chip's substrate is grounded. The reason for this is that the substrate is a resistive material. The circuitry fabricated in the substrate (in the bulk) has leakage currents (DC and AC) that flow in the p-type semiconductor of the substrate. The result is an increase in the substrate's potential above ground in localized regions of the chip. Ideally, the current flowing in the substrate is zero. In reality it won't be zero but will have some value that depends on the location and the activity of the on-chip circuitry. A substrate connection provides a place to remove this substrate current (a point of exit), keeping the substrate potential at ground. In practice, substrate connections are used wherever possible (more on this topic when we cover standard cell frames later in the chapter).

The body for the PMOS devices is the n-well. The n-well must also be tied to a known voltage through n+ and metal1. For the PMOS's body connection, an n+ region is placed in the n-well and connected with a metal1 wire to (for digital design) *VDD*. If an n-well is laid out and used to make a resistor or for the body of a PMOS device, then there must be n+ in the n-well.

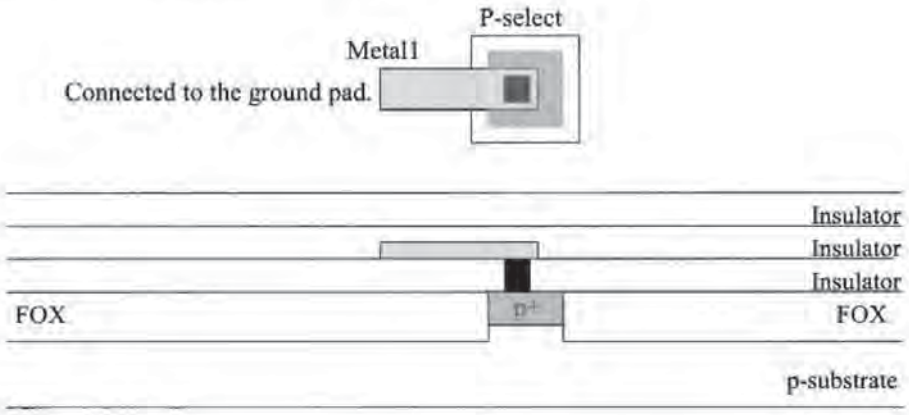



**Figure 4.9** Connecting the substrate to ground.

IPR2023-00697
Theta EX2019

*Layout of an N-Well Resistor*

The layout of an n-well resistor is seen in Fig. 4.10. On each side of the resistor, n+ regions are implanted so that we can drop metal1 down and make a connection. We aren't showing the silicide in the cross-sectional view (Fig. 4.4). Further, after reviewing Table 4.1, we see that the sheet resistance of the n+ regions is small compared to the sheet resistance of the n-well. When we calculate the number of squares, we measure between edges of active as seen in the figure. This results in a small error in the measured resistance compared to the actual resistance. The variation in the sheet resistance with process shifts (say 20%) makes this error insignificant. The next chapter will discuss the layout of resistors in more detail.

Notice that if the substrate is at ground, we can't apply a potential on either side of the wire less than, say, −0.5 V for fear of turning on the n-well to substrate parasitic diode. These parasitics, as discussed in Ch. 2, are an important concern when laying out resistors.

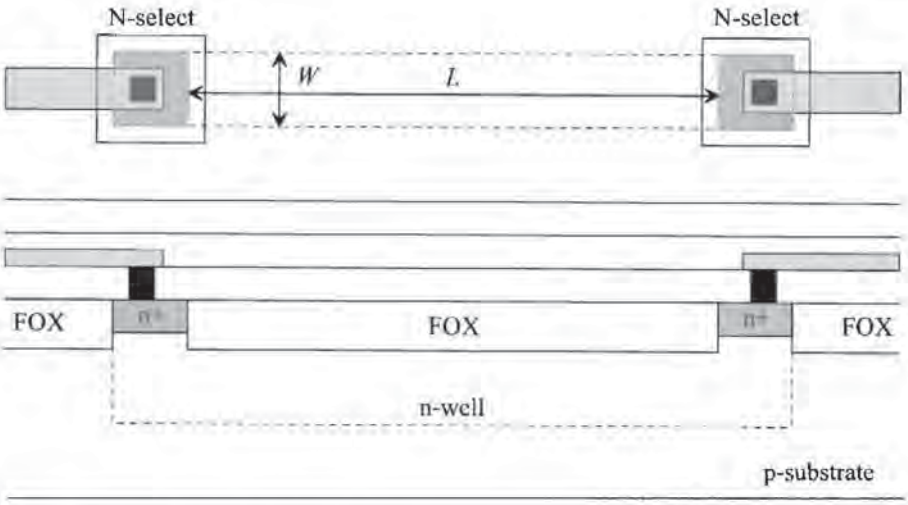



**Figure 4.10** Layout of an n-well resistor and the corresponding cross-sectional view.

**Example 4.3**

Consider the layout for a metal1 connection to an n-well resistor seen in Fig. 4.11. Will the extension of the n+ beyond the n-well affect the resistor's operation? Why or why not?

The cross-sectional view along the dotted line in the layout is also seen in the figure. The n+ active forms a diode with the p-substrate. As long as this diode doesn't forward bias, the connection to the n-well works just like the connection in Fig. 4.10. There is additional junction (depletion) capacitance with this connection (between the n+ and substrate). When the resistor in Fig. 4.10 is fabricated, the active mask won't be perfectly aligned with the n-well. As this example illustrates, it doesn't have to be perfectly aligned for a proper connection to the resistor. ■

IPR2023-00697
Theta EX2019

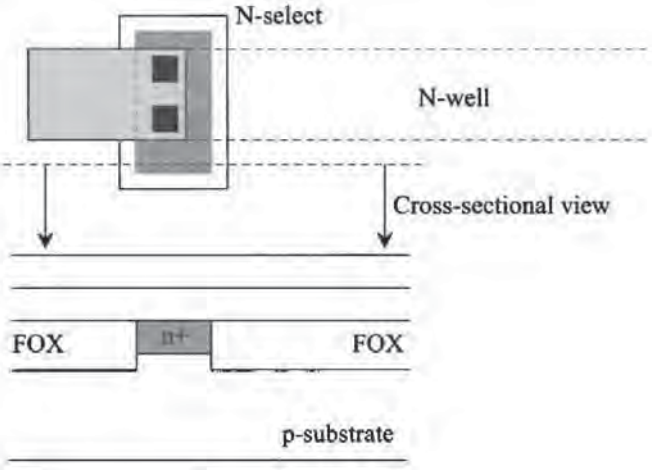



**Figure 4.11** Active extending beyond the edges of the n-well.

*Layout of an NMOS Device*

Figure 4.12 shows the layout, schematic representation, and cross-sectional views of an NMOS device. As seen in Fig. 4.3 and the associated discussion, **poly over active** forms a MOSFET. It's important to remember that the MOSFET is a four-terminal device. In

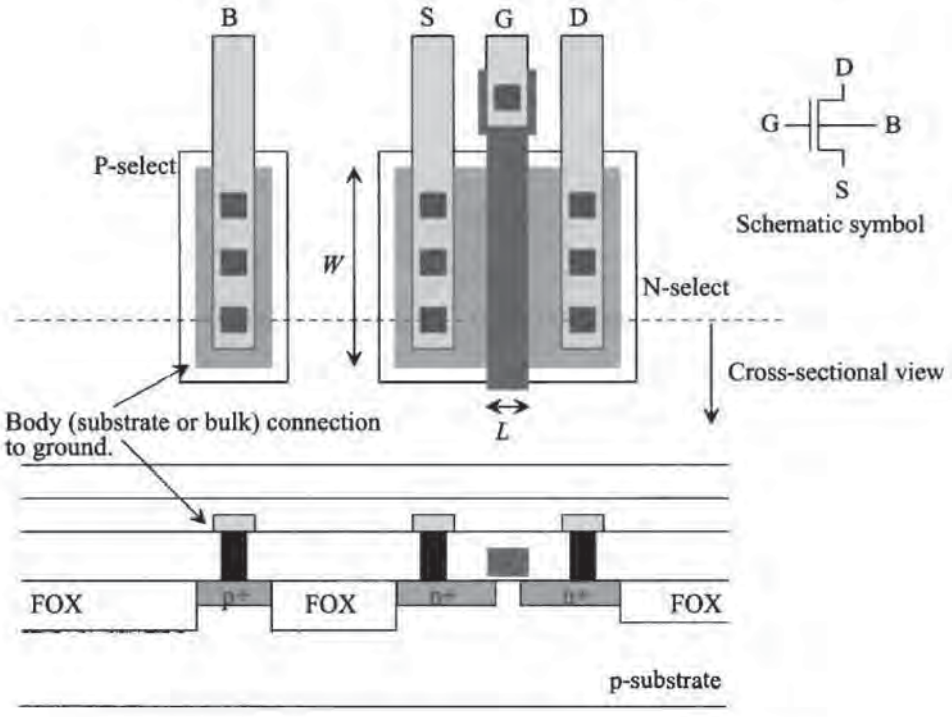



**Figure 4.12** Layout and cross-sectional views of an NMOS device.

IPR2023-00697
Theta EX2019

Fig. 4.12 we show the bulk connection in the layout and in the schematic. In an n-well process, the bulk is tied to ground so the bulk (or body) connection is normally not shown in the schematic symbol. Also note, again, that the source and drain are interchangeable.

*Layout of a PMOS Device*

Figure 4.13 shows the layout of a PMOS device. Note how we lay the device out in an n-well. Also seen in the figure is the schematic symbol for the PMOS device. Again, the source and drain of the MOSFET are interchangeable. The n-well is normally tied to the highest potential, *VDD*, in the circuit to keep the parasitic n-well/p-substrate diode from forward biasing.

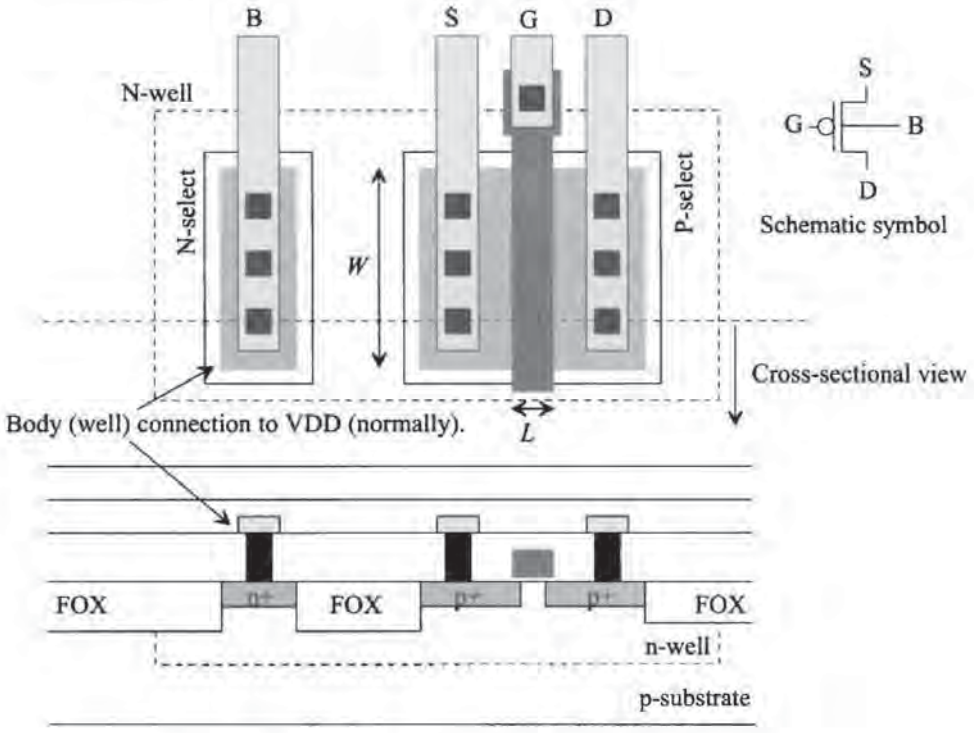
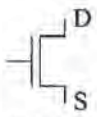



**Figure 4.13** Layout and cross-sectional views of an PMOS device.

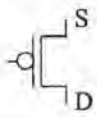
*A Comment Concerning MOSFET Symbols*

We'll use the symbols shown in Figs. 4.14a and b to represent NMOS and PMOS devices. Because the bulk terminals aren't drawn in these symbols, it's assumed they are connected to ground (NMOS device) or *VDD* (PMOS device). Notice that the drain and source in this symbol, like the actual layout, are interchangeable. The symbols seen in (c) and (d) are the bipolar-derived symbols where the arrow on the MOSFET's source represents the direction of drain current flow (derived from a bipolar junction transistor symbol). Since, for an integrated circuit MOSFET, the current can flow in either direction through the MOSFET, we'll avoid using the bipolar-derived symbols. Finally, the

IPR2023-00697
Theta EX2019




(a) NMOS device with body tied to ground.




(b) PMOS device with body tied to VDD.



(c) Bipolar-derived NMOS symbol with body tied to ground.



(d) Bipolar-derived PMOS symbol with body tied to VDD.



(e) NMOS symbol where the arrow indicates the p-substrate to n-channel diode.



(f) PMOS symbol where the arrow indicates the n-well to p-channel diode.

**Figure 4.14** Symbols commonly found in schematics to represent a MOSFET.

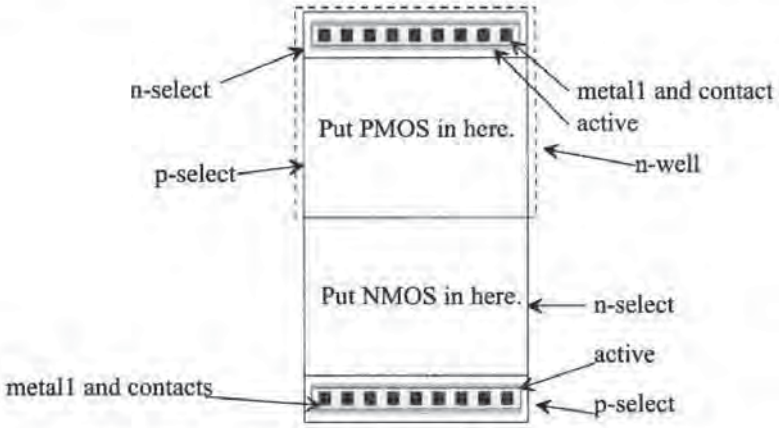
symbols seen in (e) and (f) are also used to draw schematics. The arrow in the bulk terminal represents the p-substrate (or p-well) to n-channel diode (NMOS) or the p-channel to n-well diode (PMOS).

*Standard Cell Frame*

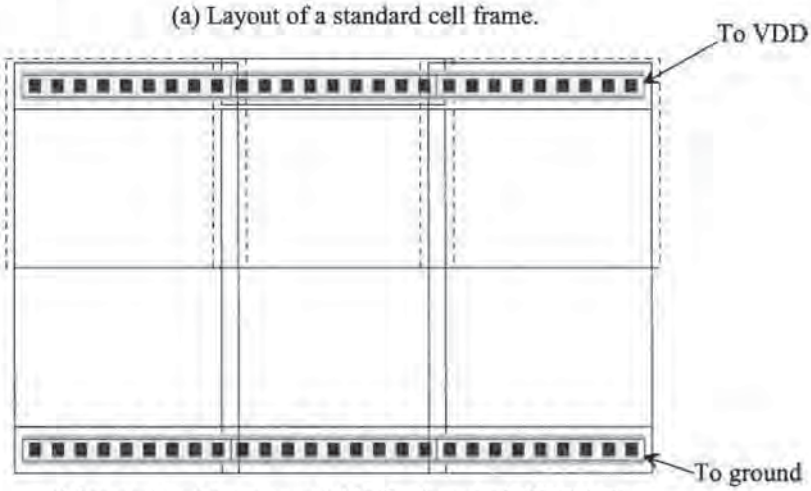
When doing layout, it can be convenient to route power and ground as well as substrate and well connections in a fashion that makes connecting the MOSFETs together simpler. We don't have to worry about the substrate and well connections or the routing of power. Towards this goal, consider the layout seen in Fig. 4.15a. The top half of the layout is an n-well. This n-well is where PMOS devices are placed, as indicated in the figure. The top of the n-well is tied to *VDD* through an n+ implant. In other words, the metal on the top of the layout has n+ underneath it and is routed to the *VDD* pad. Below this layout of the n+ well tie-down, and still in the n-well, is a p-select region. We can draw one or more PMOS devices with active, poly, contacts, and metal1 and place them in this area. The NMOS devices are laid out in the area below this, that is, the area of the n-select layer. Below the NMOS area is the substrate connection, p+, to ground. The metal, at the bottom of the layout, connects the p+ substrate contacts to the ground pad.

Figure 4.15b shows how the standard cell frames can be laid out end-to-end to increase the area available for the layout of the MOSFETs. Notice that the layout area is increased horizontally and not vertically. Standard cell frames have a standard height. Also notice how power, ground, well, and substrate connections are routed through the cell. Finally, note that overlapping layout layers, like the n-well, is fine. It simply indicates a larger layout area (of course the design rule check must be passed).

IPR2023-00697
Theta EX2019




(a) Layout of a standard cell frame.




(b) Increasing the area available for layout of MOSFETs.

**Figure 4.15** Layout of standard cell frames.

*Design Rules*

Figure 4.16 shows the design rules for the active, contact, poly, and select layers. More detailed information on the design rules can be found at MOSIS (http://mosis.org). Seen at the bottom of Fig. 4.16 is an alternative method of laying out an NMOS device. The same active area is used for both the MOSFET formation and the contact to the MOSFET's body (either p-substrate for NMOS or n-well for PMOS). In Fig. 4.12, for example, we laid out the substrate connection a distance away from the source and drain. (The source and drain of the MOSFET are interchangeable.) In the layout seen in Fig. 4.16, we *abut* the n-select directly next to the p-select. This minimizes the layout area. However, as seen in Figs. 4.2g and 4.2h, the n+ and p+ aren't well defined at the boundary between the two materials (the select masks can shift). The p+/n+ can, for all intents and purposes, be thought of as being shorted together. Since the substrate is tied to ground, the layout of the MOSFET in this way (to minimize area) **isn't symmetrical** and the source must also be tied to ground (the source and drain are not interchangeable).

IPR2023-00697
Theta EX2019

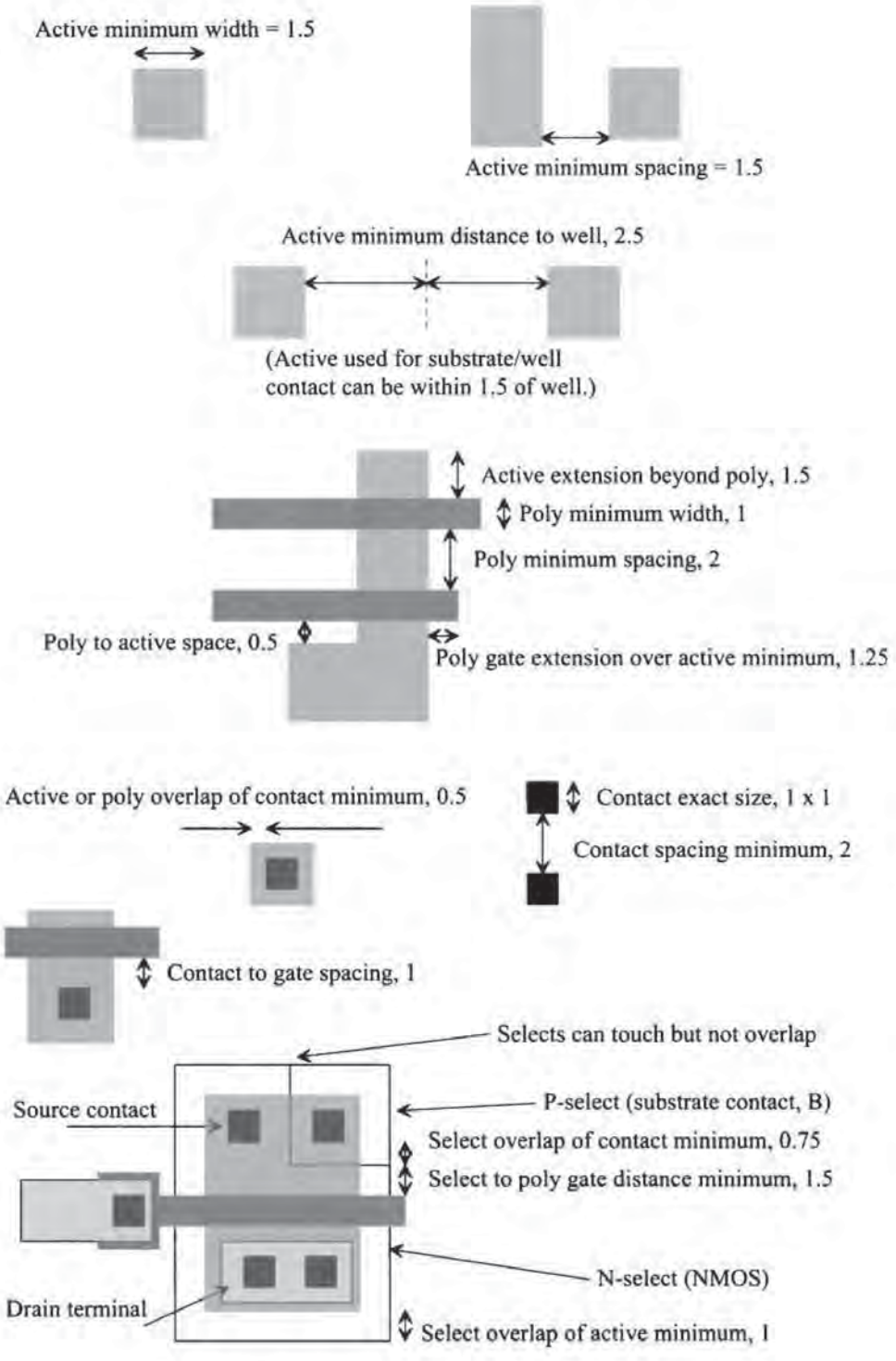



**Figure 4.16** Design rules for active, selects, poly, and contacts.

IPR2023-00697
Theta EX2019

## 4.3 Electrostatic Discharge (ESD) Protection

A major concern in CMOS technology is the protection of the thin gate oxides, Fig. 4.3b, from electrostatic discharge (ESD). While an in-depth presentation of techniques for preventing ESD damage are outside the scope of this book, here we briefly show how the active layers can be used to form a simple diode-protection structure.

Where does ESD come from? Walking across the floor, for example, causes the buildup of charge on the human body. Touching a conducting object can result in a transfer of charge or a static "shock." If the transfer of this charge (the discharge of the electrostatic charge buildup on the human body) is through the thin gate oxide (GOX) of a MOSFET, it is likely that the GOX will be damaged. While we use the human body as the location of the buildup of electrostatic charge, just about any object can build up charge. To keep from damaging the GOX, consider the circuit seen in Fig. 4.17. Here we connect two diodes to the input pad. If the signals applied to the pad are within ground and *VDD*, neither diode turns on. However, if the voltage on the pad starts to increase above *VDD* + 0.5 V or decrease below – 0.5 V, one of the diodes turns on and provides a low impedance "clamp" to keep the voltage on the GOX from becoming excessive.

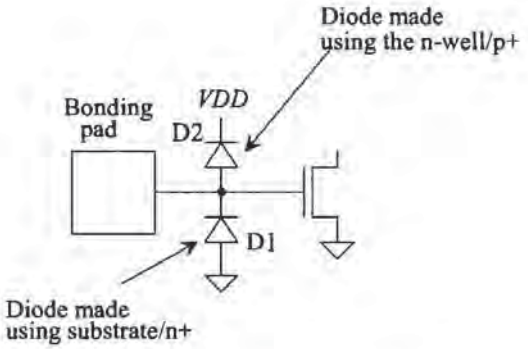



**Figure 4.17** Adding diodes to the pads to protect the MOSFET gates from ESD damage.

*Layout of the Diodes*

Examine the layout seen in Fig. 4.18. This layout shows a conceptual implementation of the diode protection circuitry seen in Fig. 4.17. Diode, D1, is implemented using p-substrate (for the anode) and n+ (for the cathode). The substrate is connected to ground through a metal1 ground bus as seen in the figure. It is desirable to place the substrate connection and n+ as close together as possible to minimize the resistance in series with the diode. Further, it is desirable to increase the size of this n+/p+ diode to both reduce the resistance of the diode and increase its current handling capability. The drawback of increasing the size is the larger capacitive loading on the pad (the depletion capacitances of the diodes). Diode D2 is implemented using p+ in the n-well (for the anode) and the n-well (for the cathode). An n+ implant is needed in the n-well to provide a connection to a wire that is connected to *VDD*. Again, the *VDD* is run through the pad on a bus (the horizontal metal line connected to *VDD* at the bottom of the layout). Again, it's desirable

IPR2023-00697
Theta EX2019

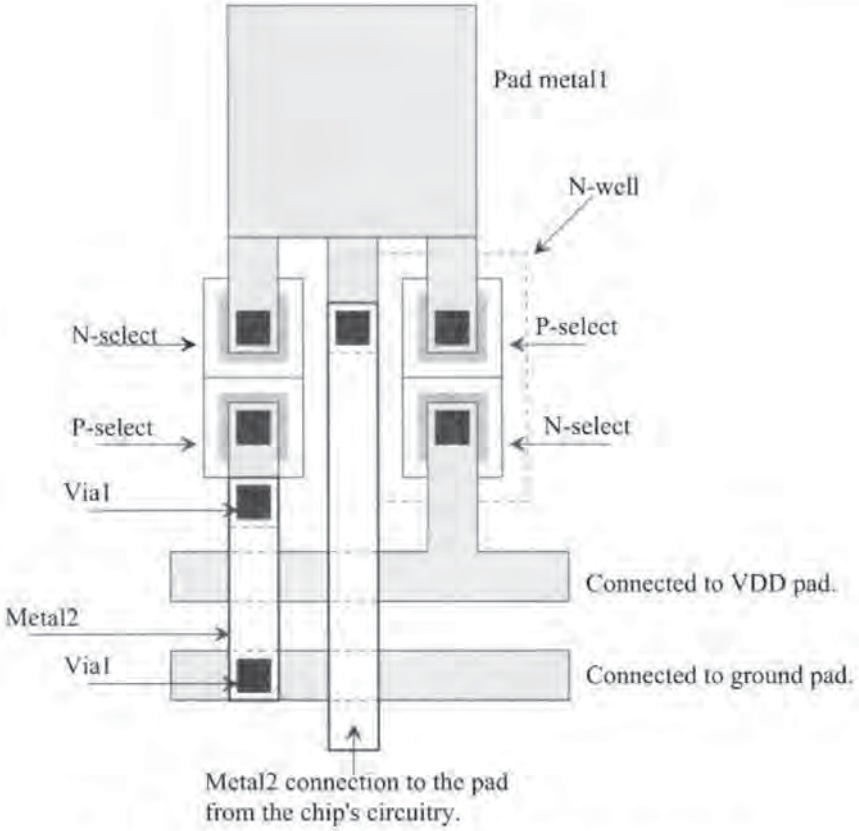



**Figure 4.18** Conceptual layout of a pad ESD protection circuit.

to minimize the series resistance between the p+ and n+/n-well. Normally, the ground bus at the bottom of the pad ties the substrate to ground (p+ is implanted under the bus and tied, with contacts, to the metal) and the *VDD* bus is connected to n+/n-well directly beneath it.

Figure 4.19 shows a more realistic view of an I/O (input/output) pad with ESD protection diodes. We've laid the implants directly next to each other to minimize parasitic resistance. We've also staggered the diodes, that is, D1, D2, D1, and D2 to increase their areas. In a real pad layout, the top layer of metal must be used (not metal1 as we've used in the figure, see Figs. 3.20 to 3.23). An example padframe layout is seen in Fig. 4.20. Two of the pads must be connected (separately) to the *VDD* and ground buses.

An important addition to most pads is a buffer to drive the large off-chip capacitances, which are relatively large compared to the on-chip capacitances. We briefly talked about output buffers in Sec. 3.3.2. The circuit design for I/O buffers is covered in Ch. 11 where we talk about inverters. While it's possible to design custom pads for a project, it's generally a better idea to get pads directly from the CMOS vendor where the chips will be fabricated (pads can be downloaded from MOSIS too).

IPR2023-00697
Theta EX2019

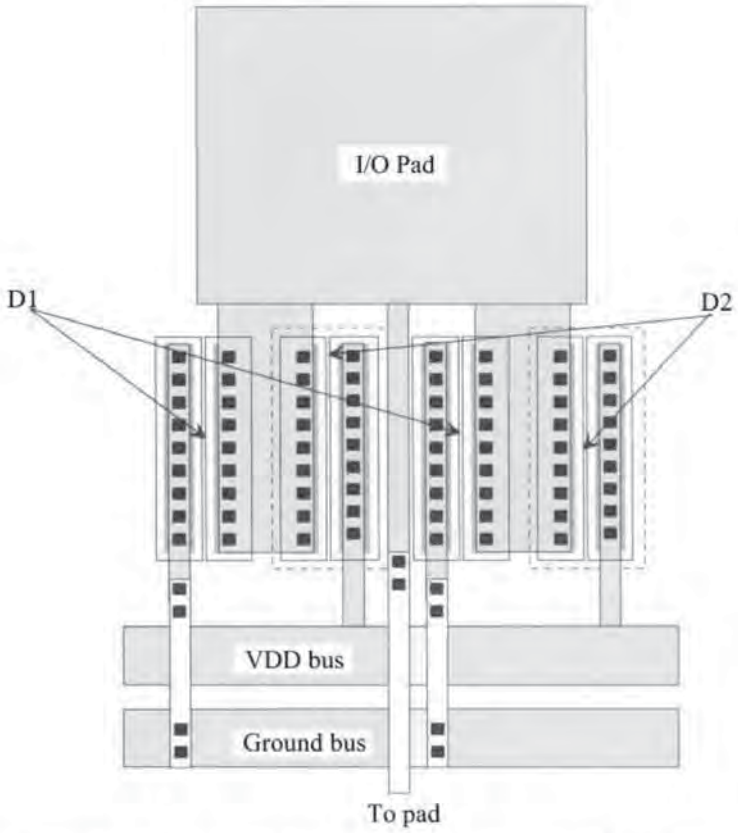



**Figure 4.19** More detailed implementation of a pad with ESD protection.

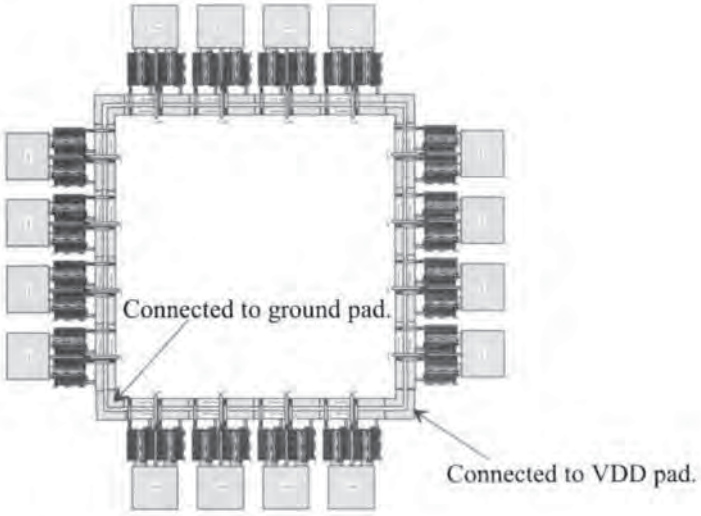



**Figure 4.20** Layout of a padframe using pads with ESD diodes.

IPR2023-00697
Theta EX2019

## ADDITIONAL READING

[1] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, Second Edition, Cambridge University Press, 2010. ISBN 978-0521832946

[2] A. Amerasekera and C. Duvvury, *ESD in Silicon Integrated Circuits*, John Wiley and Sons Publishers, 2002. ISBN 0-471-49871-8

[3] D. Clein, *CMOS IC Layout: Concepts, Methodologies, and Tools*, Newnes Publishers, 2000. ISBN 0-750-67194-7

[4] S. Dabral and T. J. Maloney, *Basic ESD and I/O Design*, John Wiley and Sons Publishers, 1999. ISBN 0-471-25359-6

## PROBLEMS

**4.1** Lay out a nominally 200 kΩ resistor with metal1 wire connections. DRC your layout. What would happen if the layout did not include n+ under the contacts? Use the sheet resistances from Table 4.1.

**4.2** Sketch the cross-sectional view along the line indicated in Fig. 4.21.

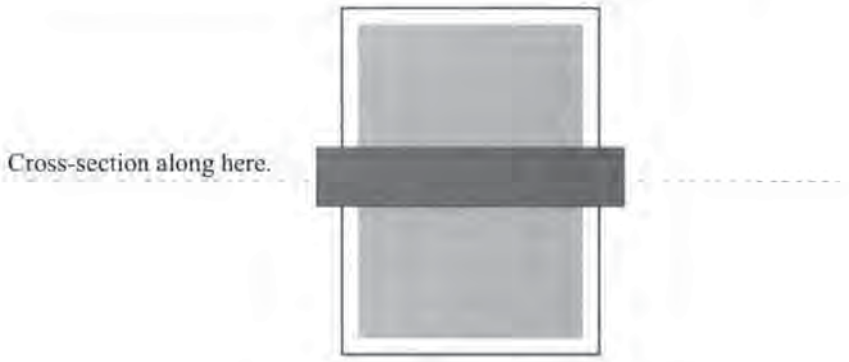



**Figure 4.21** Layout used in Problem 4.2.

**4.3** Sketch the cross-sectional views across the *VDD* and ground power buses in the standard cell frame of Fig. 4.15.

**4.4** Suppose the "bad" layout seen in Fig. Ex4.1 is used to fabricate an NMOS device. Will the poly be doped? Why or why not?

**4.5** Why is polysilicon's parasitic capacitance larger than metal1's?

**4.6** Lay out an NMOS device with an length of 1 and width of 10. Label all four of the MOSFET terminals.

**4.7** Lay out an PMOS device with a length of 1 and width of 20. Label all four of the MOSFET terminals.

IPR2023-00697
Theta EX2019

**4.8** Using the standard cell frame, lay out 10 (length) by 10 (width) NMOS and PMOS transistors. Sketch several cross-sectional views of the resulting layouts showing all four terminals of the MOSFETs.

**4.9** Repeat Ex. 4.2 for a poly wire that is 5 wide.

**4.10** Sketch the cross-sectional view at the line indicated in Fig. 4.22.

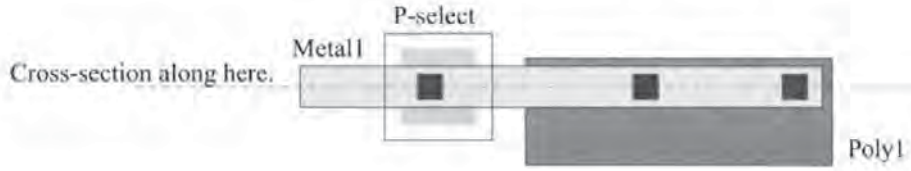



**Figure 4.22** Layout used in Problem 4.10.

**4.11** Sketch the cross-sectional views across the lines shown in Fig. 4.23.




**Figure 4.23** Layout used in Problem 4.11.

**4.12** The layout of a PMOS device is seen in Fig. 4.24 is incorrect. What is the (fatal) problem?

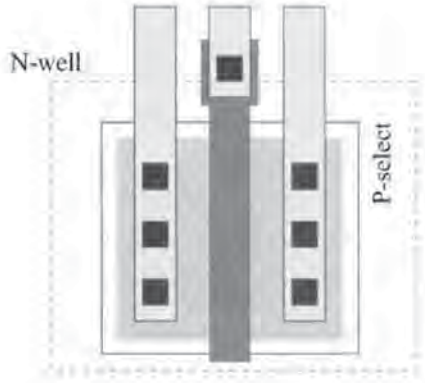



**Figure 4.24** Flawed layout of a PMOS device. What is the error?

IPR2023-00697
Theta EX2019

Chapter

5

# Resistors, Capacitors, MOSFETs

This chapter provides more information and examples related to the layout of resistors, capacitors, and MOSFETs. Layout using the poly2 layer and how poly2 is used to make poly-poly capacitors will be covered. We'll also introduce some fundamental layout techniques including using unit cells, layout for matching, and the layout of long length and wide MOSFETs. The temperature and voltage dependence of resistors and capacitors will also be covered.

## 5.1 Resistors

The resistors and capacitors in a CMOS process have values that change with voltage and temperature. The change is usually listed as ppm/°C (parts per million per degree C). The ppm/°C is equivalent to a multiplier of $10^{-6}$/°C.

*Temperature Coefficient (Temp Co)*

As temperature goes up, so does the value of a resistor, Fig. 5.1 (in general). Generally, the value of a resistor, $R(T_0)$, is specified at room temperature, $T_0$. To characterize the resistor we use

$$R(T) = R(T_0) \cdot (1 + TCR1 \cdot (T - T_0)) \qquad (5.1)$$

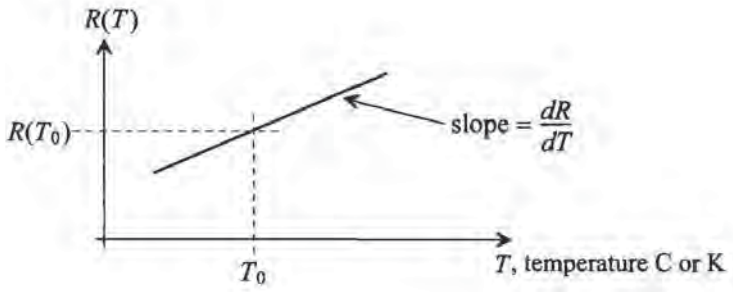



**Figure 5.1** Resistor's value change with temperature.

IPR2023-00697
Theta EX2019

where $T$ is the actual temperature of the resistor. Because of the difference in the equation, it doesn't matter if Celsius or Kelvin are used for the units of $T$ as long as the units match the units used for $T_0$. The first-order temperature coefficient of a resistor, $TCR1$, is given by

$$TCR1 = \frac{1}{R} \cdot \frac{dR}{dT} \tag{5.2}$$

Notice that the $TCR1$ changes with temperature. For most practical applications, the "temp co" is assumed to be linear. Typical values of $TCR1$ for different layers in the CMOS process are given in Table 4.1. SPICE uses a quadratic, in addition to this first order term, to model the temperature dependence of a resistor

$$R(T) = R(T_0) \cdot [1 + TCR1 \cdot (T - T_0) + TCR2 \cdot (T - T_0)^2] \tag{5.3}$$

For hand calculations, we assume $TCR2$ is 0.

**Example 5.1**

Using the values in Table 4.1, estimate the minimum and maximum resistance of an n-well resistor with a length of 100 and a width of 5 over a temperature range of 0 to 100 °C. Verify the hand calculations using SPICE.

The n-well resistor sheet resistance is approximately 500 Ω/square (at 27 °C). The value of the resistor in this example (20 squares) is 10 k at 27 °C. The temp co is 2400 ppm/°C (= 0.0024) The minimum resistance is then determined, using Eq. (5.1) by

$$R_{min} = 10,000 \cdot [1 + 0.0024 \cdot (0 - 27)] = 9.35\ k\Omega$$

and the maximum resistance is determined by

$$R_{max} = 10,000 \cdot [1 + 0.0024 \cdot (100 - 27)] = 11.75\ k\Omega$$

The SPICE simulations and netlist are shown in Fig. 5.2. ■

*Polarity of the Temp Co*

We made the comment that in general the temp co of a resistor is positive, which means that the resistor's value increases with increasing temperature. In other words, the resistivity of the silicon material used to fabricate the resistor increases with increasing temperature. If we were to look at the electron concentration in an n-well as temperature is increased, for example, we would see an exponential increase in the number of thermally generated carriers. More carriers, less resistivity, right? Looking at the carrier concentration alone, we would expect silicon's resistivity to decrease with increasing temperature. With the increase in the carrier concentration, however, we get a reduction in the carrier's mobility. The carrier-to-carrier interactions increase the scattering (the average distance a carrier can travel before colliding with some other carrier decreases with a larger number of free carriers). The material parameter "mobility" characterizes how easily carriers can move through a material. The mobility of an electron or hole, $\mu_n$ or $\mu_p$ respectively, is a material parameter that relates the applied electric field across the material to the velocity the carriers can drift through the material. High velocity indicates that the carriers have high-mobility and can move through the material quickly. Mobility is a measured parameter and is given by

IPR2023-00697
Theta EX2019

```
*** Figure 5.2 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
set temp=0
run
set temp=27
run
set temp=100
run
let iref0=-dc1.i(vr)
let iref27=-dc2.i(vr)
let iref100=-dc3.i(vr)
let r0=vr/iref0
let r27=vr/iref27
let r100=vr/iref100
plot r0 r27 r100
.endc

.dc Vr 0.9 1.1 1m
vr vr 0 DC 0

r1 vr 0 10k rmod L=5 W=100
.model rmod R RSH=500 TNOM=27 TC1=0.0024
.end
```

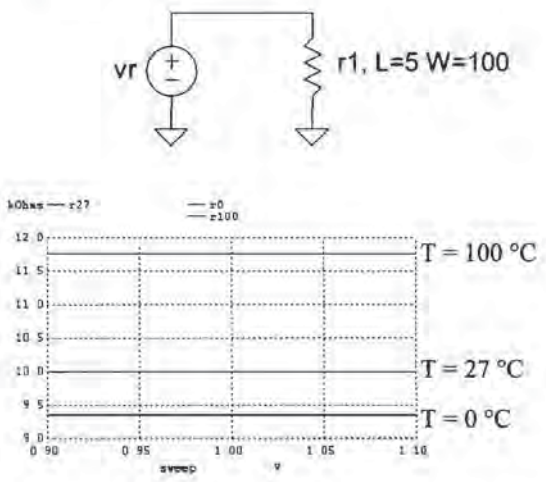



**Figure 5.2** Simulating the temperature dependence of an n-well resistor.

$$\mu_{n,p} = \frac{\text{Average velocity of carriers, } cm/s}{\text{Applied electric field, } V/cm} \tag{5.4}$$

noting mobility's units are $cm^2/V \cdot s$. The resistivity of the material depends on the number of free carriers (electrons/cm$^3$, $n$, and holes/cm$^3$, $p$) or

$$\rho = \frac{1}{q(\mu_n n + \mu_p p)} \tag{5.5}$$

where $q$ is the electron charge (see Eqs. [2.2] and [2.3]). This equation is important and shows why we can have a negative or positive resistor temperature coefficient. If the mobilities decrease faster than the carrier concentrations increase, we get a positive temp co. The resistor's value goes up with increasing temperature because the mobility is dropping faster than the carrier concentration is increasing. However, if the increase in carrier concentration with temperature is faster, the resistivity goes down with increasing temperature and we get a negative temp co (the resistor's value goes down with increasing temperature). At room temperature, a positive temperature coefficient is normally observed.

*Voltage Coefficient*

Another important contributor to a changing resistance is the voltage coefficient of the resistor given by

$$VCR = \frac{1}{R} \cdot \frac{dR}{dV} \tag{5.6}$$

IPR2023-00697
Theta EX2019

where $V$ is the average voltage applied to the resistor, that is, the sum of the voltages on each end of the resistor divided by two. The resistance as a function of voltage is then given by

$$R(V) = R(V_0) \cdot (1 + VCR1 \cdot (V - V_0) + VCR2 \cdot (V - V_0)^2) \qquad (5.7)$$

The value $R(V_0)$ is the value of the resistor at the voltage $V_0$. A typical value of $VCR1$ is 8000 ppm/$V$ for an n-well resistor. The main contributor to the voltage coefficient is the depletion layer width between the n-well and the p-substrate. The depletion layer extends into the n-well, resulting in an effective change in the sheet resistance. The thickness of the n-well available to conduct current decreases with increasing potential (reverse bias) between the n-well and the substrate.

**Example 5.2**

Estimate the average resistance of an n-well resistor with a typical value of 10k at 27 °C, for an average voltage across the resistor of 0, 5, and 10 V.

$$R(0) = 10,000 \cdot (1 + 0.008 \cdot 0) = 10\ k\Omega$$

$$R(5) = 10,000 \cdot (1 + 0.008 \cdot 5) = 10.4\ k\Omega$$

$$R(10) = 10,000 \cdot (1 + 0.008 \cdot 10) = 10.8\ k\Omega$$

This is a small change compared to the change due to temperature. However, as the next example will show, the change due to the applied voltage can have a greater influence on the circuit performance than the temperature. ■

**Example 5.3**

Compare the change in $V_{out}$ in the following circuit due to the $VCR$ with the change due to the $TCR$.

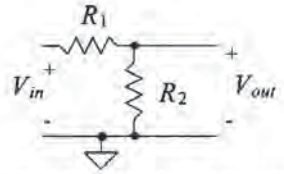



**Figure 5.3** Comparing a resistive divider's temperature performance to its voltage performance.

We start by writing the voltage divider equation and then substitute the temperature or voltage dependence. For the temperature dependence

$$V_{out} = V_{in} \cdot \frac{R2(T)}{R1(T) + R2(T)} = V_{in} \cdot \frac{R2 \cdot [1 + TCR(T - T_0)]}{(R1 + R2) \cdot [1 + TCR(T - T_0)]} = V_{in} \cdot \left[\frac{R2}{R1 + R2}\right]$$

showing that the divider is independent of temperature. For the voltage dependence

$$V_{out} = V_{in} \cdot \frac{R2(V)}{R1(V) + R2(V)} = V_{in} \cdot \frac{R2 \cdot (1 + VCR \cdot V_2)}{R1 \cdot (1 + VCR \cdot V_1) + R2 \cdot (1 + VCR \cdot V_2)}$$

where $V_1 = \frac{Vin + Vout}{2}$ and $V_2 = \frac{Vout}{2}$. These results show that the voltage divider has no temperature dependence but that it does have a voltage dependence. ■

IPR2023-00697
Theta EX2019

*Using Unit Elements*

The preceding example shows the advantage of ratioing components. For precision design over large temperature ranges, this fact is very important. Usually, a *unit resistor* is laid out with some nominal resistance. Figure 5.4a shows a unit cell resistor layout, say 5 kΩ, using the n-well. Figure 5.4b shows how the nominal 5 kΩ unit cell resistor implements a 4/5 voltage divider. Using the divider in Fig. 5.3 as a reference, the resistor $R_1$ is 5k and the resistor $R_2$ is 20k (notice the layout of the 20k resistor is 4 unit cells). The errors due to corners and differing perimeters using a serpentine pattern, see Fig. 2.28, are eliminated when using unit cell layout techniques. Also, the exact calculation of the resistor's value isn't important. Changes in the nominal resistance value because of process shifts, temperature, or how the number of squares is calculated only affect the current flowing in the divider (the power dissipation) and not the output voltage. Amplifier circuits, such as those using op-amps with feedback, are examples of circuits where ratioing is important.

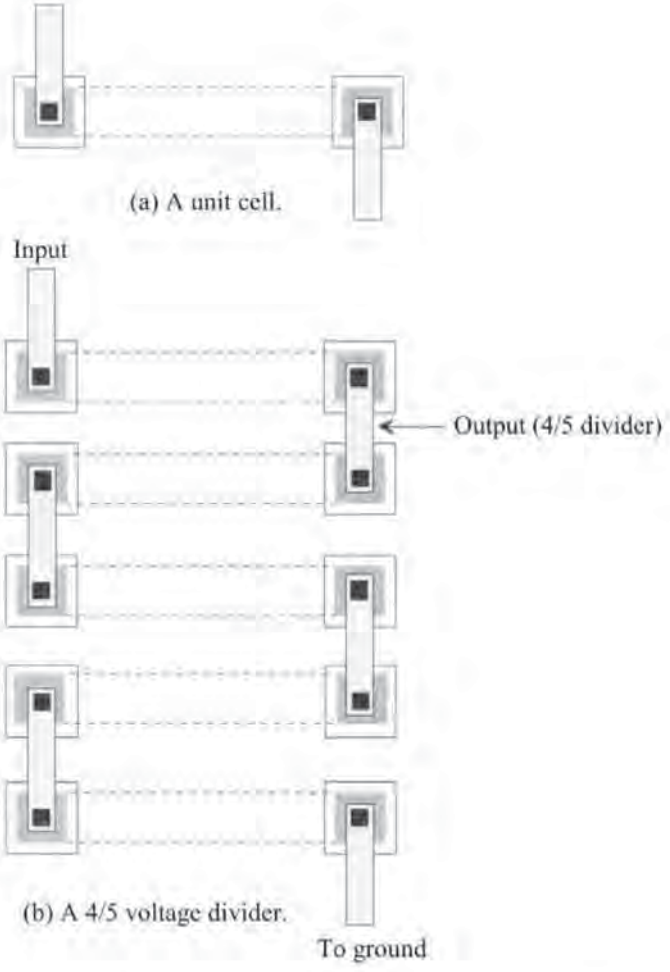



**Figure 5.4** (a) Layout of a unit resistor cell, and (b) layout of a divider.

IPR2023-00697
Theta EX2019

*Guard Rings*

Any precision circuit is susceptible to substrate noise. Substrate noise results from adjacent circuits injecting current into one another. The simplest method of reducing substrate noise between adjacent circuits is to place a p+ implant (a substrate contact for a p-substrate wafer) between the two circuits. The substrate contact removes the injected carriers and holds the substrate, ideally, at a fixed potential (ground). Figure 5.5 shows the basic idea for a resistor. The p+ implant guards the circuit against carrier injection. Because the implants are laid out in a ring, they are often called *guard rings*.

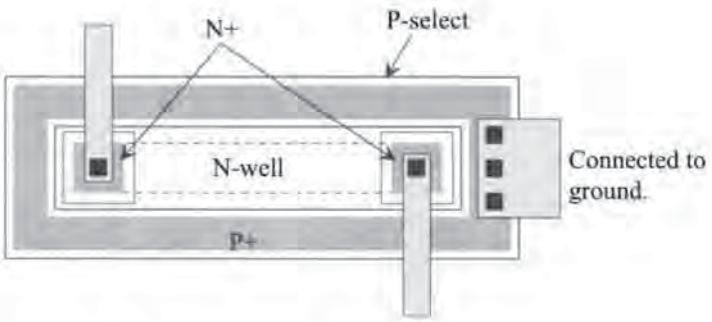



**Figure 5.5** Guard ringing an n-well resistor.

*Interdigitated Layout*

Matching between two different resistors can be improved by using the layout shown in Fig. 5.6. These resistors are said to be interdigitated. Process gradients, in this case changes in the n-well, n+, or p+ doping at different places on the die, are spread between the two resistors more evenly. Notice that the orientation of the resistors is consistent between unit cells; that is, all cells are laid out (here) vertically. Also, each resistor has essentially the same parasitics.

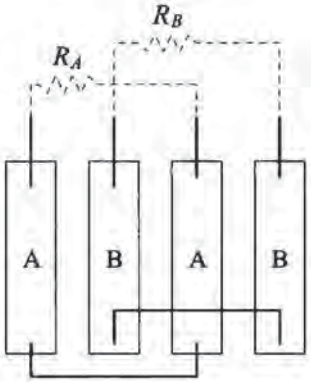



**Figure 5.6** Layout of interdigitated resistors.

IPR2023-00697
Theta EX2019

*Common-Centroid Layout*

Common-centroid (common center) layout helps improve the matching between two resistors (at the cost of uneven parasitics between the two elements). Consider the arrangement of unit resistors shown in Fig. 5.7a. This interdigitated resistor is the same layout style as just discussed in Fig. 5.6. Consider the effects of a linearly varying sheet resistance on the overall value of each resistor. If we assign a normalized value to each unit resistor, as shown, then resistor A has a value of 16 and resistor B has a value of 20. Ideally, the resistor values are equal.

Next consider the common-centroid layout shown in Fig. 5.7b, noting that resistors A and B share a common center. The value of either resistor in this figure is 18. In other words, the use of a common center (ABBAABBA) will give better matching than the interdigitized layout (ABABABAB). Figure 5.8 shows the common-centroid layout (two different possibilities) of four matched resistors. Note that common-centroid layout can also improve matching in MOSFETs or capacitors.

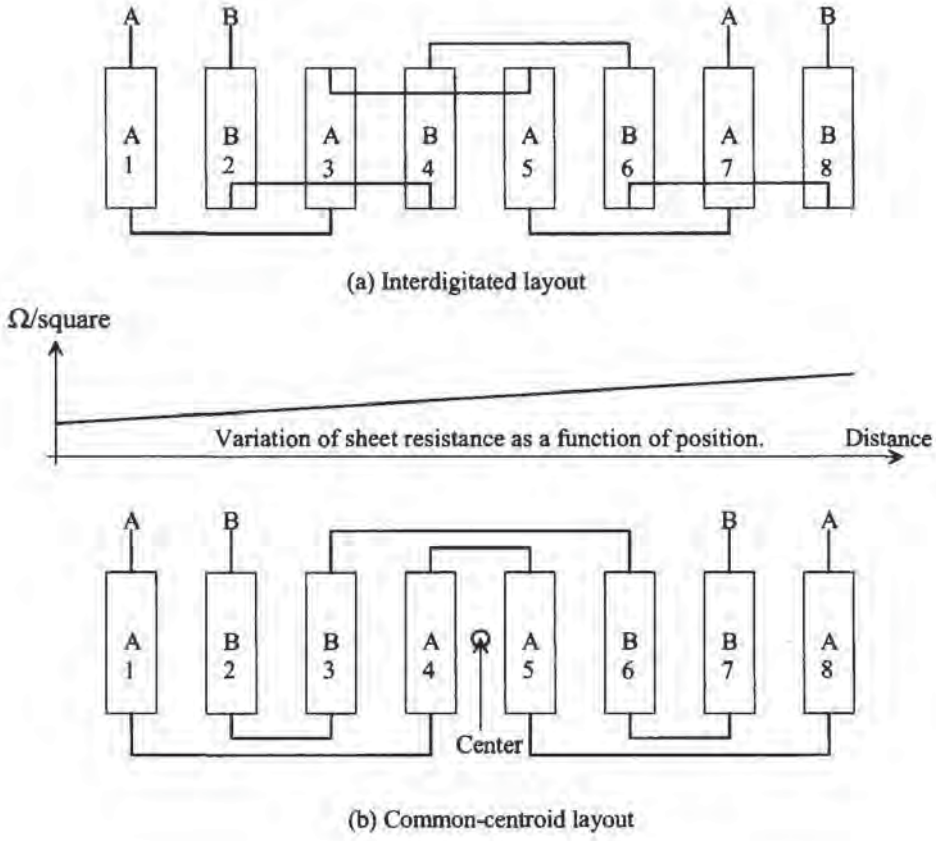



**Figure 5.7** (a) Interdigitated layout and (b) common-centroid layout.

IPR2023-00697
Theta EX2019

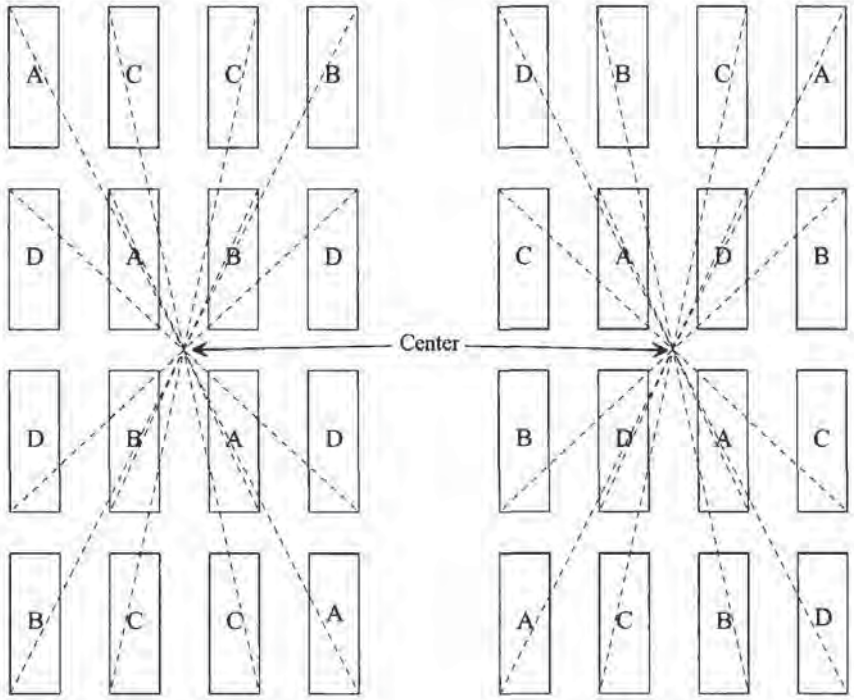



**Figure 5.8** Common-centroid layout of four matched resistors (or elements).

**Example 5.4**

Suppose that the nominal value of the unit resistor, A, on the left side of the layouts in Fig. 5.7 is 5k (a nominal value because we know the actual sheet resistance varies with process shifts and temperature). If the sheet resistance linearly varies across the layout, from the left to the right, and the final resistor's value is 5.07k, compare the interdigitated layout to the common-centroid layout.

The farthest left resistor in the layout has a value of 5k (given). The second resistor's value, because of the linear variation in the sheet resistance from the left to the right in the layout, is 5.01k. The third resistor's value is 5.02k, etc. For the interdigitated layout, the value of resistor A is

$$R_A = 5.0 + 5.02 + 5.04 + 5.06 = 20.12\ k\Omega$$

and the value of resistor B is

$$R_B = 5.01 + 5.03 + 5.05 + 5.07 = 20.16\ k\Omega$$

For the common-centroid layout, resistor A's value is

$$R_A = 5.0 + 5.03 + 5.04 + 5.07 = 20.14\ k\Omega$$

and resistor B's value is

$$R_B = 5.01 + 5.02 + 5.05 + 5.06 = 20.14\ k\Omega$$

IPR2023-00697
Theta EX2019

exactly the same result. If we had laid out all of the unit elements for resistor A on the left and the all of the unit elements for resistor B on the right, we would get $R_A = 20.06\ k\Omega$ and $R_B = 20.22\ k\Omega$. This shows that the interdigitated layout does help with matching. ■

*Dummy Elements*

Another technique that improves the matching between two or more elements is the use of dummy elements. Consider the cross-sectional views of the three n-wells shown in Fig. 5.9a. The final amount of diffusion under the resist, on the edges, is different between the outer and the inner unit cells. This is the result of differing dopant concentrations at differing points on the surface (during the diffusion process). This difference results in a mismatch between unit resistors. To compensate for this effect, dummy elements can be added (see Fig. 5.9b) to an interdigitated or common-centroid layout. The dummy element does nothing electrically. It simply ensures that the unit resistors of matched resistors see the same adjacent structures. Normally, these dummy elements are tied off to either ground or *VDD* rather than left floating.

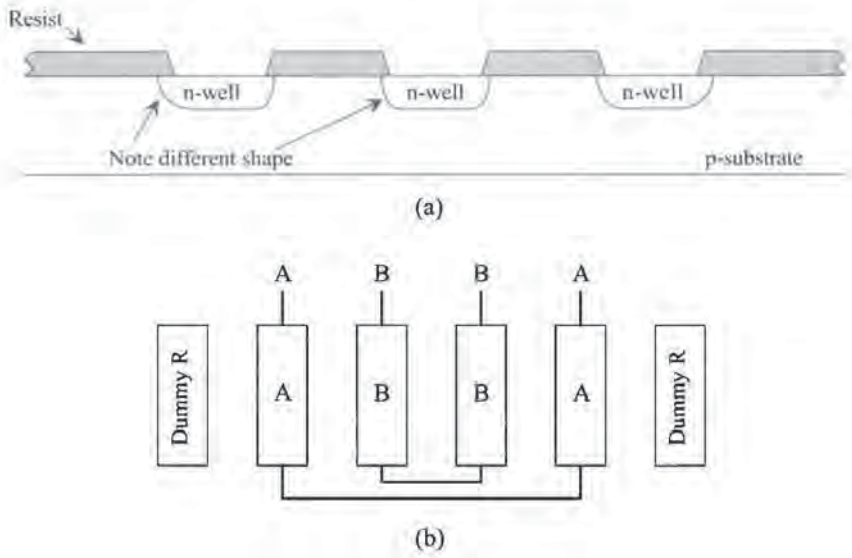



**Figure 5.9** (a) Edge effects and (b) a common-centroid layout with dummy R.

## 5.2 Capacitors

An added layer of polysilicon, called poly2, can be present in a CMOS process for capacitor formation (called a poly-poly capacitor), for MOSFET formation (we can use poly2 instead of poly1 for the gate of a MOSFET) and for creating a floating gate device (see Ch. 16's discussion of Flash memory technology for example). In this section we discuss the layout of capacitors using poly2, the parasitics present, and the temperature behavior of the poly-poly capacitor. Many of the layout techniques discussed in the last section can be used when laying out capacitors.

IPR2023-00697
Theta EX2019

*Layout of the Poly-Poly Capacitor*

Layout and cross-sectional views of a capacitor using the poly1 and poly2 layers are shown in Fig. 5.10. The silicon dioxide dielectric between the two layers of poly is roughly the same thickness as the gate oxide (GOX), $t_{ox}$ in a MOSFET (see Fig. 4.3b). Table 5.1 shows **typical values for the $t_{ox}$ that we'll use in the 1 μm and 50 nm processes, referred to as long- and short-channel CMOS processes, in this book.** Also seen in the table is the oxide capacitance per area calculated using

$$C'_{ox} = \frac{\varepsilon_r \cdot \varepsilon_0}{t_{ox}} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{5.8}$$

where $\varepsilon_0 = 8.85 \times 10^{-18}\ F/\mu m = 8.85\ aF/\mu m$ and the relative dielectric constant of $SiO_2$ is 3.97 ($= \varepsilon_r$). To calculate the value of a capacitor, we look at the area where poly1 and poly2 intersect, $A$, or

$$C_{ox} = C'_{ox} \cdot A \tag{5.9}$$

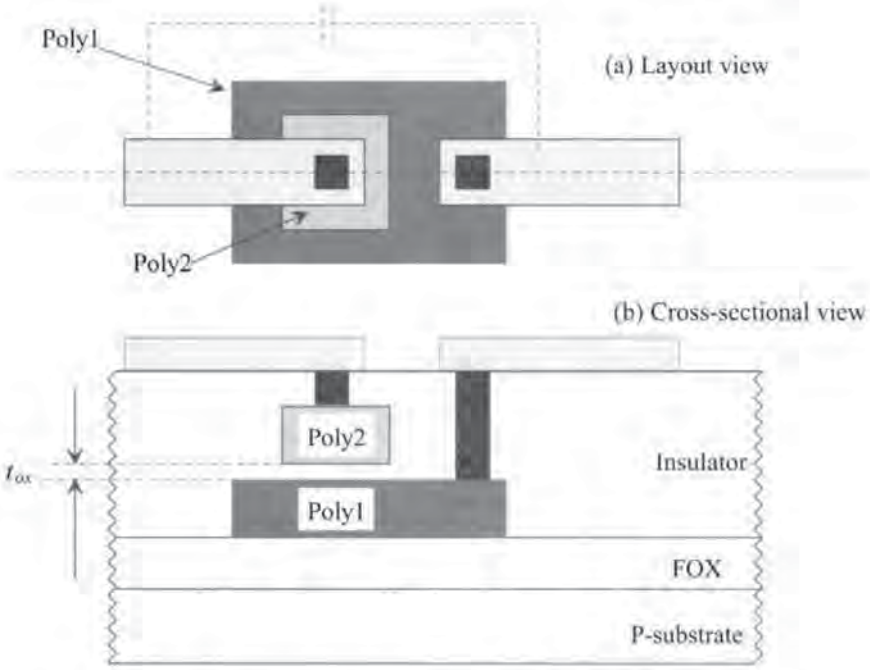



**Figure 5.10** Layout and cross-sectional view of a poly-poly capacitor.

If the scale factor is included, then we can write this equation as

$$C_{ox} = C'_{ox} \cdot A \cdot (scale)^2 \tag{5.10}$$

If, in the 50 nm process, a poly-poly capacitor is formed with an intersection of the poly1 and poly2 that measures 10 by 20 then the capacitor's value is

$$C = C_{ox} = 25\ fF/\mu m^2 \cdot 200 \cdot (0.05\ \mu m)^2 = 12.5\ fF \tag{5.11}$$

IPR2023-00697
Theta EX2019

**Table 5.1** Oxide thicknesses and oxide capacitances for the long- and short-channel CMOS processes used in this book.

| CMOS technology | Oxide thickness, $t_{ox}$ | $C'_{ox}$ |
|---|---|---|
| 1 μm (long channel) | 200 Å | 1.75 *fF*/μm² |
| 50 nm (short channel) | 14 Å | 25 *fF*/μm² |

Note that when the poly2 is used to form a floating gate device (see Ch. 16) poly2 can be called an electrode (see the MOSIS design rules). Also note that the practical minimum capacitor value one should try using (laying out) in the long- and short-channel processes described in Table 5.1 is approximately 100 *fF* and 10 *fF*, respectively.

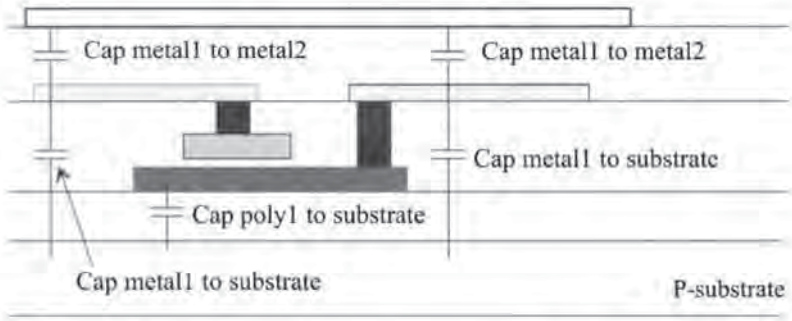



**Figure 5.11** The parasitic capacitances of the poly-poly capacitor.

*Parasitics*

As with any layout structure, we must be concerned with the parasitics. Figure 5.11 shows the parasitics associated with the poly-poly capacitor. The most important (largest) parasitic is the capacitance from poly1 to substrate. This capacitance is called the **bottom plate parasitic** capacitance. Reviewing Table 3.1, this parasitic (plate) capacitance is 58 *aF*/μm² (there are fringe capacitances as well that can be considered). Keeping in mind that the poly1 area is larger than the poly2 layout area (and the intersection of poly1 and poly2 is the desired capacitance), the bottom plate capacitance can be 20% (or more) of the desired capacitance value. In analog circuits the bottom plate of a capacitor is indicated, as seen in Fig. 5.12, see Ch. 25.

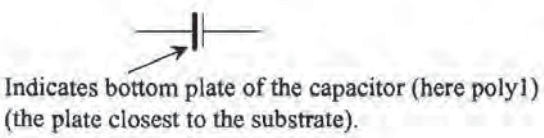



**Figure 5.12** The bottom plate parasitic of a capacitor.

IPR2023-00697
Theta EX2019

*Temperature Coefficient (Temp Co)*

The first-order temperature coefficient of a capacitor, *TCC*, is given by

$$TCC = \frac{1}{C} \cdot \frac{dC}{dT} \tag{5.12}$$

A typical value of *TCC* for a poly-poly capacitor is 20 ppm/°C. Matching of large area poly-poly capacitors on a die is typically better than 0.1% with good layout techniques. The capacitance as a function of temperature is given by

$$C(T) = C(T_0) \cdot [1 + TCC \cdot (T - T_0)] \tag{5.13}$$

where $C(T_0)$ is the capacitance at $T_0$.

*Voltage Coefficient*

The voltage coefficient of a capacitor is given by

$$VCC = \frac{1}{C} \cdot \frac{dC}{dV} \tag{5.14}$$

The voltage coefficient of the poly-poly capacitor is in the neighborhood of 10 ppm/V (for a long-channel process). The capacitance as a function of voltage is given by

$$C(V) = C(V_0) \cdot (1 + VCC \cdot V) \tag{5.15}$$

where $C(V_0)$ is the capacitance between the two poly layers with zero applied voltage, and $V$ is the voltage between the two plates.

## 5.3 MOSFETs

We introduced the layout of MOSFETs in the last chapter. In this section we discuss some electrical parameters of interest and how to lay out MOSFETs to minimize parasitics or with a long length or width.

*Lateral Diffusion*

When the drain and source regions are implanted, some of the implant dose laterally diffuses out underneath the gate poly, as depicted in Fig. 5.13. If the drawn length is called $L_{drawn}$, we can write the effective length as

$$L_{effective} = L_{drawn} - 2L_{diff} \tag{5.16}$$

where $L_{diff}$ is the length of the lateral diffusion. To minimize the lateral diffusion, a spacer is normally deposited adjacent to the poly after a light implant (see Figs. 4.7f and g) and before the heavy source/drain implants. This type of structure is called a lightly doped drain (LDD).

*Oxide Encroachment*

There are also imperfections associated with the width of the MOSFET, Fig. 5.13. When the active area is defined, Fig. 4.3, the FOX won't be precisely patterned as specified by the active mask layer. The oxide may encroach on the active area (called oxide encroachment) and reduce the active opening area. The drawn width, $W_{drawn}$, of the MOSFET will be different from the effective width, $W_{effective}$ by $2W_{enc}$. To compensate for oxide encroachment, the layout may be bloated before making the active mask.

IPR2023-00697
Theta EX2019

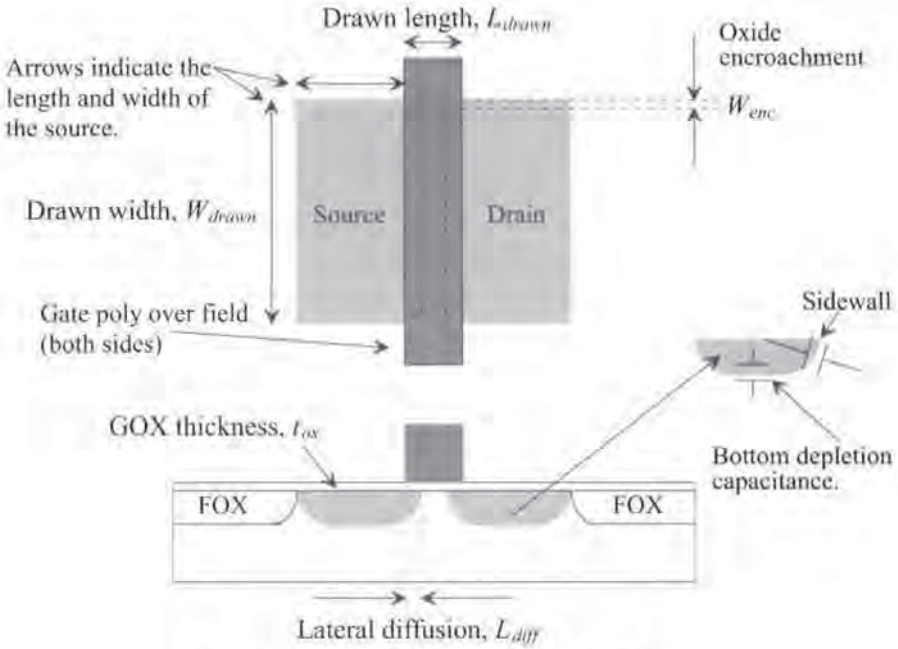



**Figure 5.13** Lateral diffusion in a MOSFET.

*Source/Drain Depletion Capacitance*

As we saw in Ex. 2.3, a pn junction fabricated in the bulk has a depletion capacitance consisting of a bottom component and a sidewall component. For the source and drain junction (depletion) capacitances, this can be written in terms of SPICE parameters as

$$C_{js,d} = \frac{cj \cdot A_{S,D} \cdot (scale)^2}{\left(1 + \frac{V_{S,DB}}{pb}\right)^{mj}} + \frac{cjsw \cdot P_{S,D} \cdot (scale)}{\left(1 + \frac{V_{S,DB}}{pbsw}\right)^{mjsw}} \qquad (5.17)$$

where *cj* is the zero-bias bottom depletion capacitance, $A_{S,D}$ is the area of the source or drain implant, *scale* is the scale factor (1 μm for the long-channel process and 50 nm for the short-channel process), *cjsw* is the zero-bias depletion capacitance for the sidewalls, *pb* and *pbsw* are the built-in potentials for the bottom and sidewall components, respectively, *mj* and *mjsw* are the grading coefficients for the bottom and sidewall components, respectively, and finally $V_{S,DB}$ is the potential from the source or drain to the MOSFET's body (the substrate for the NMOS and the well for the PMOS).

It's very common to call the source/drain depletion capacitance (incorrectly) *diffusion capacitance*. As discussed in Sec. 2.4.3, a pn junction only exhibits diffusion capacitance when it becomes forward biased. When CMOS technology was first introduced, the source/drain regions were formed using a diffusion process step (perhaps the reason for the incorrect labeling). CMOS technology developed from, approximately, the mid-80s has used implantation to form the source/drain regions. In any case, this diode junction capacitance should be called either a "junction capacitance" or "depletion capacitance," unless the diode is forward biased.

IPR2023-00697
Theta EX2019

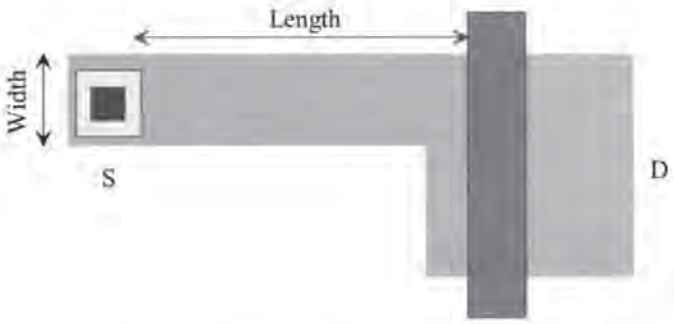



**Figure 5.14** Determining a resistance in series with the drain of the MOSFET.

*Source/Drain Parasitic Resistance*

Examine the layout seen in Fig. 5.14. The active area, in this layout, is used to move the contact to the MOSFET's source or drain away from the gate poly. This can be useful if metal needs to be run over the area directly next to the gate poly. The length of active adds a parasitic resistance in series with the MOSFET. This resistance is determined by specifying the number of squares in the source/drain (NRD/NRS). An estimate for the number of squares, assuming that the extension is on what we label the source of the MOSFET, for the layout in Fig. 5.14, is

$$NRS = \frac{Length}{Width} \tag{5.18}$$

For the other side of the MOSFET (here we call this side the drain), the length is less than the width so we can set *NRD* to zero (or not specify it). To calculate the resistance in series with a MOSFET's source or drain, Fig. 5.15, we use

$$R_S = NRS \cdot R_{sh} \text{ or } R_D = NRD \cdot R_{sh} \tag{5.19}$$

In a SPICE model the paramter rsh is used to specify sheet resistance of the n+ (for the NMOS model) or p+ (for the PMOS model).

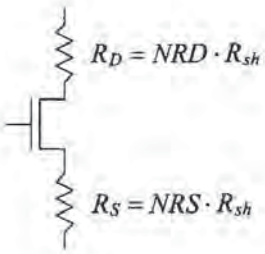



**Figure 5.15** How source/drain series parasitic resistance is modeled in SPICE.

IPR2023-00697
Theta EX2019

**Example 5.5**
For the MOSFET layout seen in Fig. 5.16, write the SPICE statement including the areas and number of squares in the source/drain regions with or without a scale factor of 50 nm.

Estimates for the areas of the drain/source are

$$A_D = 40 \text{ and } A_S = 45$$

with units of area squared. Note how we didn't include the small area directly adjacent to the gate on the source side. Again, we are estimating (or, more appropriately, approximating) the area. The actual values of capacitance (or resistance) vary with process shifts.

The perimeters of the drain/source are estimated as

$$P_D = 28 \text{ and } P_S = 36$$

The length of the MOSFET is 1 and the width is 10, or,

$$L = 1 \text{ and } W = 10$$

The number of squares of n+ in series with the drain/source is

$$NRD = \frac{4}{10} \rightarrow 0 \text{ and } NRS = \frac{11}{3} \approx 4$$

The SPICE statement for the MOSFET is

```
M1  D G S B NMOS L=1 W=10 AD=40 AS=45 PD=28 PS=36 NRD=0 NRS=4
```

The MOSFET device name always starts with an M (a voltage source device name always starts with a V, a resistor R, etc.) The drain, gate, source, and bulk nodes (in this statement) are labeled D, G, S, B, respectively. Note that in an n-well process the bulk is always tied to ground (which is universally zero, 0, in SPICE). The MOSFET's model name is NMOS (same as the technology to make

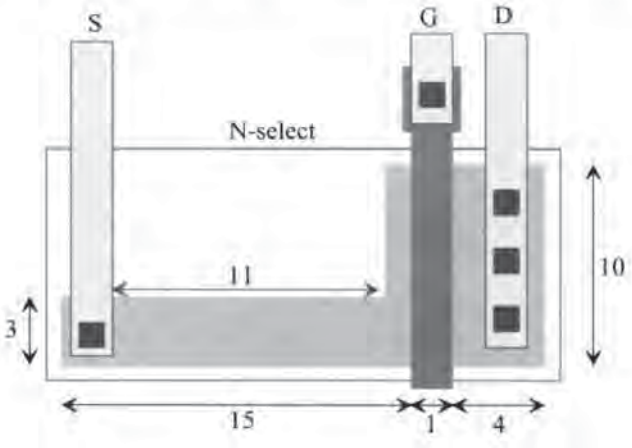



**Figure 5.16** Layout of the MOSFET used in Ex. 5.5.

IPR2023-00697
Theta EX2019

it easier to remember). The statement above is used if we employ a scale factor in the layouts and simulations. Using this MOSFET specification, we would also have to include, in the SPICE netlist,

```
.options scale=50n
```

If we didn't include this statement, then the MOSFET specification would be

```
M1  D G S B NMOS L=50n W=500n AD=100f AS=112.5f PD=1.4u
+ PS=1.8u NRD=0 NRS=4
```

noting that the "+" symbol in the first column of a line indicates that the previous line is continued on the line. **We use drawn sizes in the layout and circuits in this book.** If a SPICE netlist using MOSFETs with drawn sizes doesn't include the .options statement with the scale parameter, then the simulation output is likely flawed. ■

*Layout of Long-Length MOSFETs*

Figure 5.17 shows the layout of a MOSFET with a long length. The active layer is "snaked" back-and-forth under the poly. Each side of this active is contacted to metal1 for the source and drain connections. The width of the MOSFET is equal to the width of the active under the poly. The length of the MOSFET is estimated by looking at the length of the active underneath the poly between the drain and source. In other words, we know that the drain current flows from the drain to the source. Further we know that the drain current flows in the active area. The poly controls the drain current but is isolated from it by the gate oxide. The length is determined by the intersection of the poly and active between the source and drain contacts (the length of the dotted arrows in the figure). Long-length MOSFETs have, as we'll see, a higher effective switching resistance.

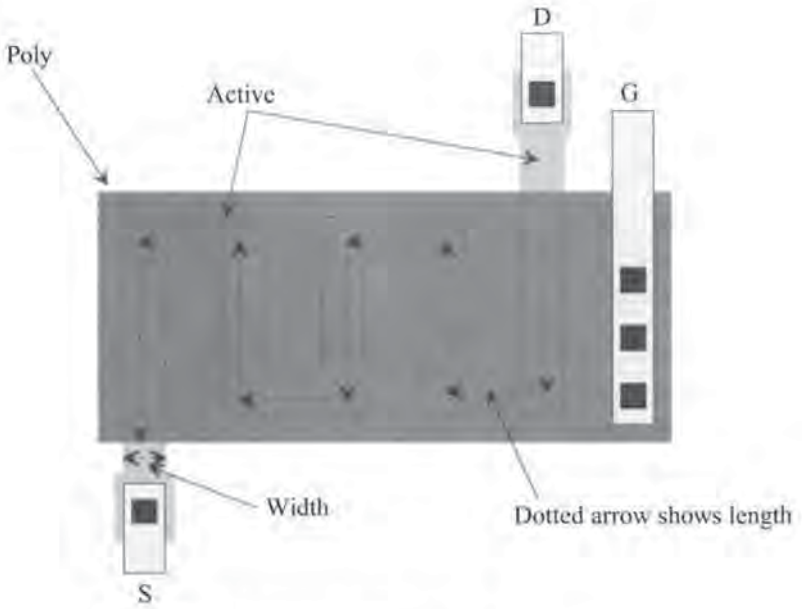



**Figure 5.17** Layout of a long-length MOSFET.

IPR2023-00697
Theta EX2019

*Layout of Large-Width MOSFETs*

Figure 5.18 shows the layout of a large MOSFET. The length of the MOSFET, $L$, is still set by the length of poly, as seen in the figure. The MOSFET's overall width is set by the width of poly over active, $W$, times the number of poly "fingers"

$$\text{Width of MOSFET} = (\text{number of fingers}) \cdot W \tag{5.20}$$

This layout minimizes area by sharing drain and source connections between MOSFETs. Notice how the widths of MOSFET's laid out in parallel (with the gates tied together) add to form an equivalent (to the sum) width MOSFET. This concept can also be used for MOSFET's laid out in series, Fig. 5.19. MOSFETs laid out in series (with their gates tied together) form a MOSFET with an effective length equal to the sum of the individual MOSFET's lengths. We use these concepts often when doing design. For example, we can lay out a MOSFET with a

$$\text{Width-to-length ratio} = W/L = 10/2 \tag{5.21}$$

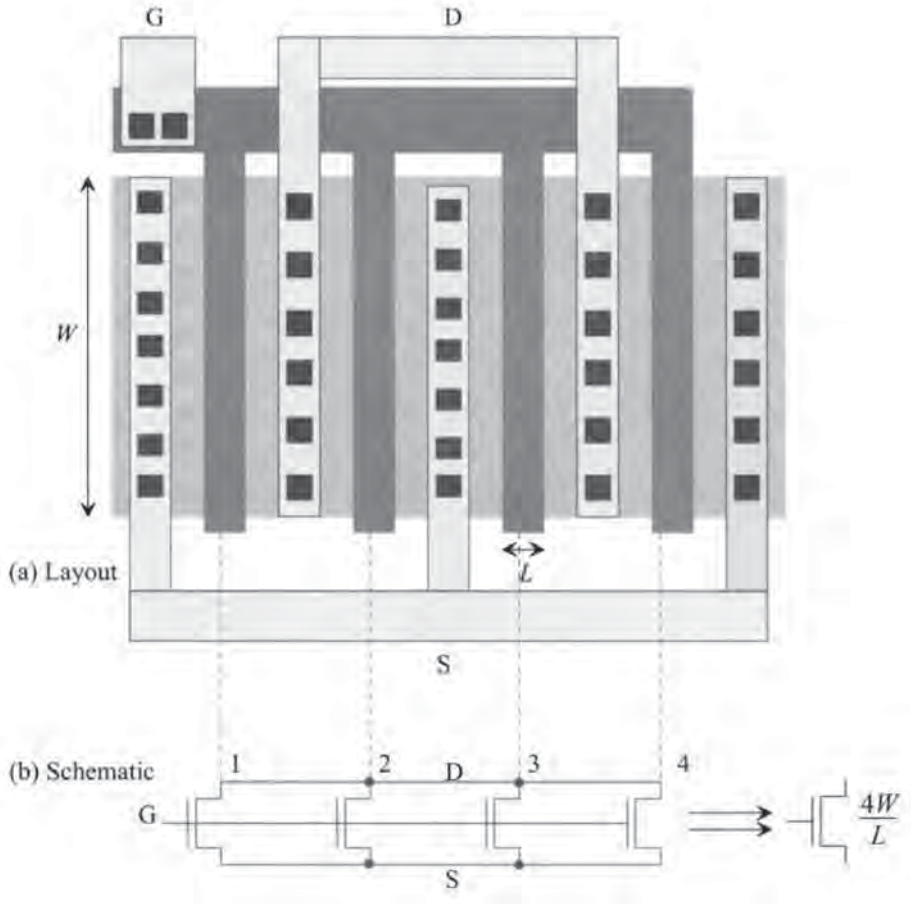



**Figure 5.18** Layout and equivalent schematic of a large-width MOSFET.

IPR2023-00697
Theta EX2019

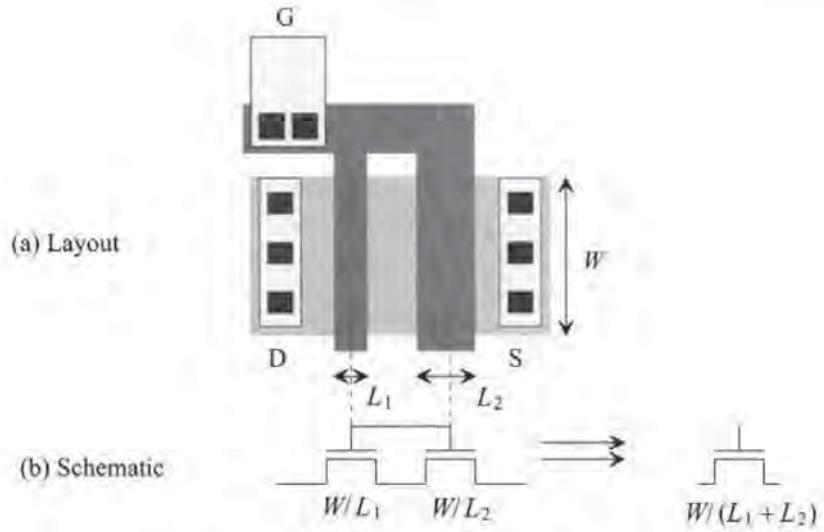



**Figure 5.19** MOSFETs in series with gates tied together behave as a single MOSFET with the sum of the lengths.

as a MOSFET with a width of 10 and a length of 2 or as two MOSFETs in series with lengths of 1 and widths of 10. **Notice** that when we write, for a MOSFET size, 10/2, $\frac{15}{3}$, 25/10, or 10/100 the first or top number always specifies the width of the MOSFET, while the second number specifies the MOSFET's length.

Notice how, when there is an even number of fingers in a large width MOSFET (Fig. 5.18), the area of the active is larger on one side of the poly (on one side of the MOSFET) than on the other side. This leads to a larger parasitic depletion capacitance. Consider the MOSFET schematics in Fig. 5.20. In (a) of this figure, we've drawn the parasitic capacitance on the MOSFET symbol. Generally, for high-speed design, we want to place the MOSFET terminal with the larger parasitic capacitance closest to ground (for

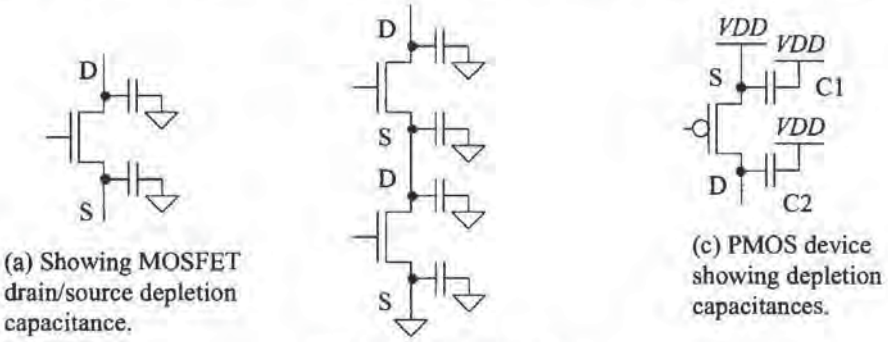



**Figure 5.20** Showing depletion capacitance on the MOSFET symbols.

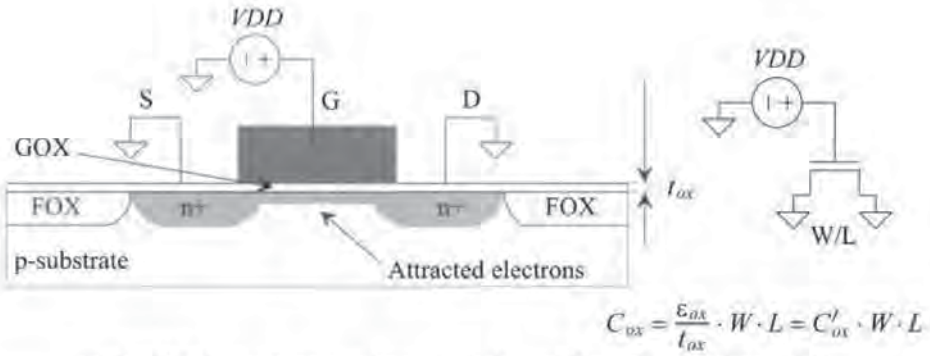
IPR2023-00697
Theta EX2019

the NMOS) or *VDD* for the PMOS. To understand this, consider the MOSFETs seen in Fig. 5.20b. Thinking of the two NMOS devices as switches, we see that when both switches turn on, the top parasitic capacitance is discharged through both switches. The middle two parasitic capacitors discharge through one switch and the bottom parasitic doesn't charge or discharge (both sides are always tied to ground). The smallest parasitic capacitance should be at the top of the switches because it has the highest resistance discharge path (in this example through two MOSFETs). For the PMOS device in Fig. 5.20c, we would want the larger of C1 or C2 to be called the source terminal and connected to *VDD*.

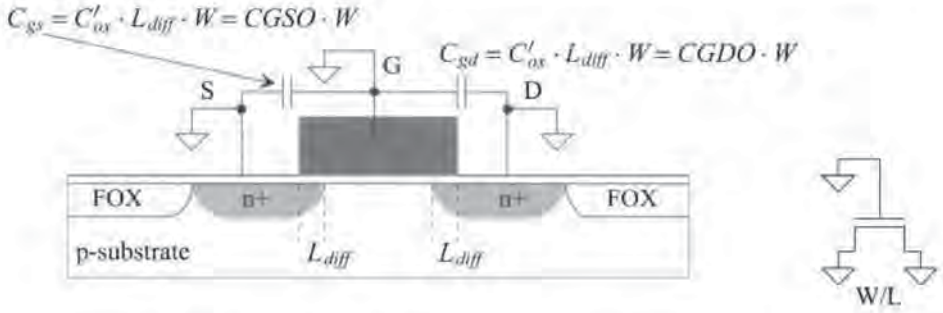
*A Qualitative Description of MOSFET Capacitances*

Consider the NMOS device in Fig. 5.21a. Here we apply a voltage to the gate of the NMOS device (the most positive voltage in the circuit, the power supply voltage, *VDD*) while holding the source and drain at the same potential as the substrate (ground). The application of this voltage attracts electrons under the gate oxide. This creates a continuous channel of electrons, effectively shorting the drain/source implants. The capacitance from the gate terminal to ground is calculated as

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \cdot W \cdot L = C'_{ox} \cdot W \cdot L \tag{5.22}$$




(a) NMOS device in strong inversion (channel formed under the oxide).




(b) NMOS device in depletion (no channel formed under the oxide).

**Figure 5.21** Qualitative description of MOSFET capacitances.

IPR2023-00697
Theta EX2019

The device is said to be operating in the strong inversion region. The surface under the GOX is p-substrate. When we apply a large positive potential to the gate, we change this material from p-type to n-type (we invert the surface). Note that the oxide capacitance, $C_{ox}$, does not depend on the extent of the lateral diffusion.

Next examine the configuration in Fig. 5.21b. In this figure all of the MOSFET terminals are grounded. No channel is formed under the GOX. Because of contact potentials, discussed in the next chapter, the area under the GOX is depleted of free carriers. Under these conditions, the MOSFET is operating in the depletion region. The source and drain are not connected as they were in Fig. 5.21a. The capacitance from the gate to the source (or drain) depends on the lateral diffusion and is given by

$$C_{gs} = C'_{ox} \cdot L_{diff} \cdot W = C_{gd} = CGDO \cdot W = CGSO \cdot W \qquad (5.23)$$

The parameter *CGDO* (or *CGSO)* is SPICE parameter called the gate-drain (gate-source) overlap capacitance and is given by

$$CGDO = CGSO = C'_{ox} \cdot L_{diff} \qquad (5.24)$$

When the MOSFET is off, as it is in Fig. 5.21b, the overlap of the gate over the source/drain implant region (the overlap capacitance) is an important component of the capacitance at the MOSFET's gate terminal.

Before leaving this section let's show an SEM photo of a cross-sectional view of a MOSFET (actually 3 MOSFETs), Fig. 5.22. For more information about MOSFET formation see Fig. 6.18 and the associated discussion.

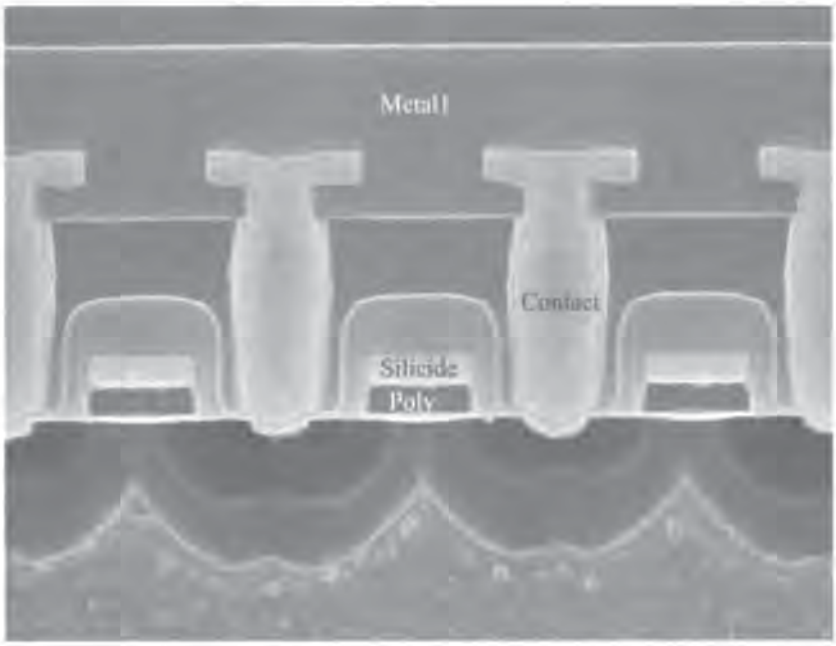



**Figure 5.22** SEM image of the cross-section of three MOSFETs.

IPR2023-00697
Theta EX2019

## 5.4 Layout Examples

In this section we provide some additional layout examples with a focus on implementing capacitors using (only) metal. Metal-only capacitors are important in CMOS processes that only have one layer of poly (common in most digital CMOS processes). We end the chapter with some discussions about laying out polysilicon resistors.

*Metal Capacitors*

One method of forming capacitors in a single-poly CMOS process uses the metal layers. Consider the cross-sectional view of a parallel plate capacitor shown in Fig. 5.23. If the plate capacitance between the metal1 and metal2 dominates because the metals have a large layout area (that is, the fringe capacitance contribution is small), then the capacitance can be estimated using

$$C_{12} = \text{Area} \cdot (\text{capacitance per area}) \tag{5.25}$$

If the capacitance per area is 50 aF/μm$^2$, then it would take an area of 100 μm by 200 μm to implement a 1 pF capacitor. While large area is a problem, it isn't the main problem with a metal parallel-plate capacitor. The main problem occurs from the extremely large bottom plate parasitic capacitance, that is, the capacitance from metal1 to substrate. This parasitic capacitance can be anywhere from 80 to 100% of the desired capacitance. Further it usually slows the circuit response and results in a waste of power (see Fig. 25.16 and the associated discussion).

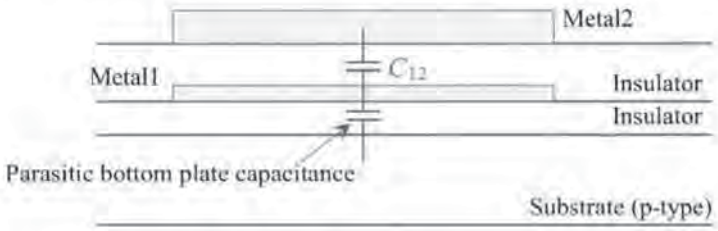



**Figure 5.23** Parallel-plate capacitor using metal1 and metal2.

To help decrease the bottom plate's percentage of the desired capacitor value, consider the cross-sectional view shown in Fig. 5.24, where four layers of metal implement a capacitor. The capacitance of this structure can be estimated using

$$C = C_{12} + C_{23} + C_{34} \tag{5.26}$$

If plate capacitance between each metal layer is, again, 50 aF/μm$^2$, then the area required to implement a 1 pF capacitor is 100 μm by 66 μm. The area needed is reduced by one-third of the area used in the metal1/metal2-only capacitor. While we used the same plate capacitance value in between each level the actual value will vary because of the differing thickness in between the metals. The absolute value of the capacitors, in most circuit design situations (as we'll see later in the book), isn't important but rather the ratio of capacitors is the important parameter (see for example, Eq. [25.19] and the associated discussion). Also notice how the thickness of the metals (made most often now with copper) increases as we move away from the substrate.

IPR2023-00697
Theta EX2019

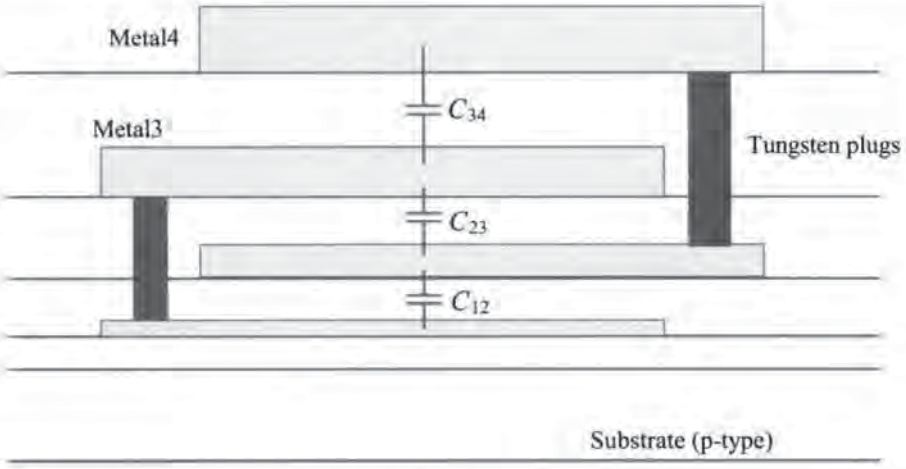



**Figure 5.24** Cross-sectional view of a parallel plate capacitor using metal1-metal4.

The value of the capacitors in Figs. 5.23 and 5.24 was set by the areas of the metals and the corresponding plate capacitance. We assumed the perimeter of the metals and the resulting fringe capacitance was a small contributor. Figure 5.25 shows typical minimum sizes and distances between pieces of metal1 where the fringe capacitance dominates. We can make a capacitor using the two pieces of metal1 shown in this figure. A typical value of capacitance per length is 50 aF/μm. The parasitic bottom plate capacitance is half of this value or 25 aF/μm. Since, as seen in the figure, the electric fields can terminate on the close adjacent metal, the bottom component is a smaller percentage than it was when the plate capacitance dominated.

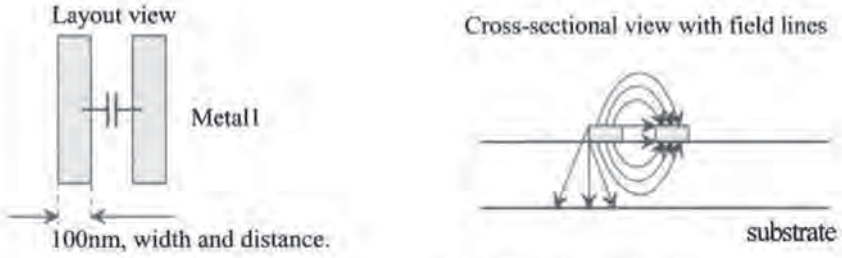



**Figure 5.25** Typical size when fringing capacitance dominates.

Next consider the use of metal2 and via1 in the implementation of a capacitor as seen in Fig. 5.26. While the fringe capacitance is still a major component in this capacitor because of the addition of the via between the metals, it is sometimes called a lateral capacitor (there exists a "plate" capacitance between the vias). A typical value of capacitance for this structure is 500 aF/μm. The bottom plate capacitance remains approximately 25 aF/μm. Using additional vias and metal layers will increase the capacitance but generally not linearly. The higher levels of metal, e.g. metal4 or metal5, generally have larger spacing and width design rules than do the lower levels of metal.

IPR2023-00697
Theta EX2019

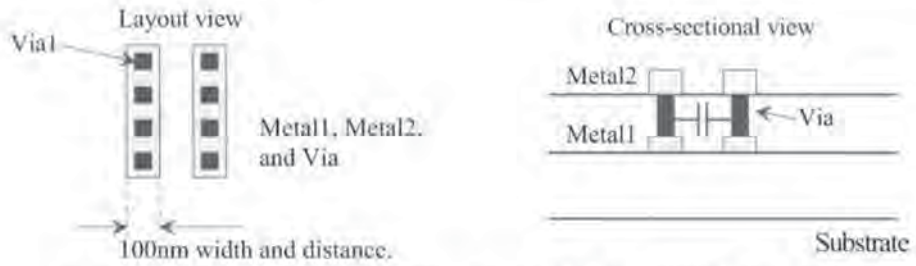



**Figure 5.26** Using two layers of metal and the via to implement a lateral capacitor.

It's important to note that while we've concentrated on the bottom plate parasitic, it is also possible to have a top plate parasitic. Often, to avoid coupling noise into the relatively large area occupied by the capacitor, a ground plate is placed above the capacitor. This would allow noisy digital signals to be routed above the capacitor, as seen in Fig. 5.27.

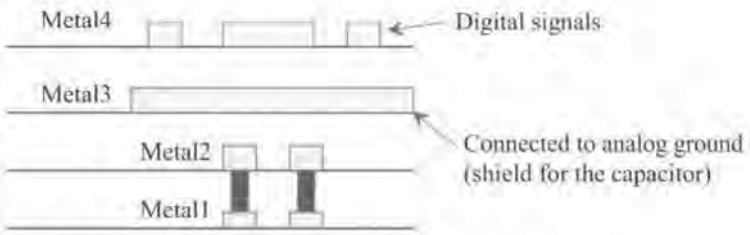



**Figure 5.27** Using a metal3 shield to isolate the lateral capacitor.

*Polysilicon Resistors*

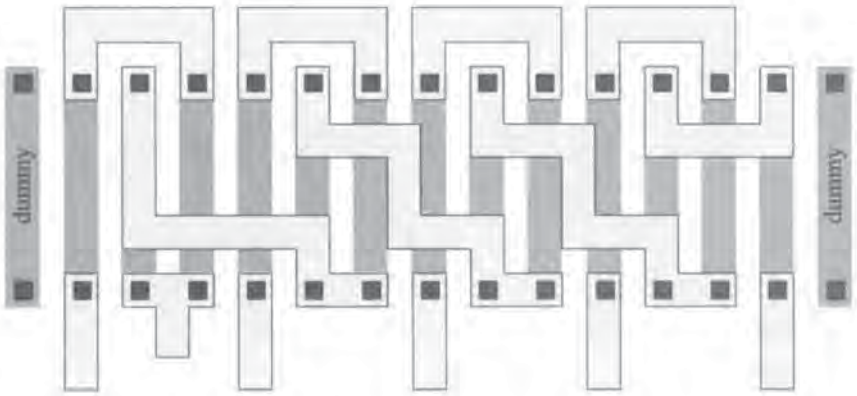
In general, polysilicon resistors offer the best performance when precision resistive ratios are required. This is mainly due to the fact that the polysilicon material sits on top of the FOX while the other resistive materials reside in the bulk (and thus form a pn junction that gives rise to a large voltage coefficient.) Since the matching characteristics, temperature behavior, and voltage coefficient are, overall, better for the polysilicon resistors, they are generally preferred in the implementation of precision circuits such as data converters.

In general, a resistor's width and length should be at least 10 and 100 times the minimum feature size of the process, respectively. For example, if $L_{min}$ is 50 nm, then the minimum width of the resistor should be 500 nm (or wider!). Using larger widths and lengths for the resistors is important both for matching and to ensure that the self-heating, which occurs because of the different current densities flowing in the resistors, doesn't cause any noticeable differences in linearity. In simple terms, the larger resistor area dissipates heat better than the same valued resistor in a smaller area.

Figure 5.28a shows the conceptual layout of an $R$-$2R$ resistor string in a minimum area (common resistor topology in an R-2R data converter, see Fig. 29.5.) Figure 5.28b

IPR2023-00697
Theta EX2019

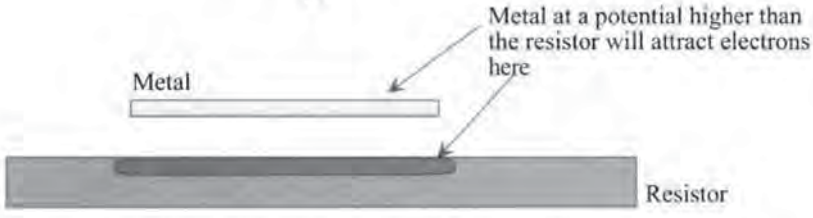
shows the actual layout of the resistors having large width and length along with a large number of contacts to reduce metal/resistive material contact resistance. Figure 5.28c shows the problem of laying out metal over the resistive material, that is, resistor *conductivity modulation*. The figure shows what happens when a metal, having a potential greater than the potential of the underlying resistor is laid out directly over a resistor. Electrons are attracted towards the surface of the resistor causing spots of lower




(a)




(b)




(c) Cross sectional view of metal over resistor.

**Figure 5.28** (a) Minimal layout of R-2R string, (b) actual layout of resistor, and (c) conductivity modulation of the resistor value.

IPR2023-00697
Theta EX2019

resistivity. The solutions to avoiding or reducing conductivity modulation are (1) avoiding running metal over the resistors, (2) using higher levels of metal to route the resistive signals so as to increase the distance between the resistor and the overlaying metal (remembering vias and contacts must be plentiful to avoid adding unwanted series resistance), or (3) inserting a conducting "shield" connected to analog ground and made with metal1 between the resistors and the routing wires above the *R*-2*R* resistor array.

Finally, to conclude this subsection, we ask, "What is the best method of laying out the resistors in an *R*-2*R* string to avoid process gradients and achieve good matching?" While there are no absolute answers, we will discuss a possibility where layout area is a concern. In other words, we won't discuss methods that use a large amount of layout area to average out process variations but will limit our averaging to at most twice the layout area of the *R*-2*R* string shown in Fig. 5.28.

Figure 5.29 shows one possibility for averaging process gradients using a common-centroid configuration with two *R*-2*R* strings connected in series. In this figure we are assuming that the process variations change linearly with position. For example, the first resistor in the string may have an effective value of 1k, while the second's value may be 1.01k, and the third's value is 1.02k, etc. The normalized change in the resistance value is shown in the figure using numbers. However, we could show that the process gradients average out no matter what numbers are used, when using this layout topology, as long as the sheet resistance varies linearly with position. For example, the MSB 2*R* in

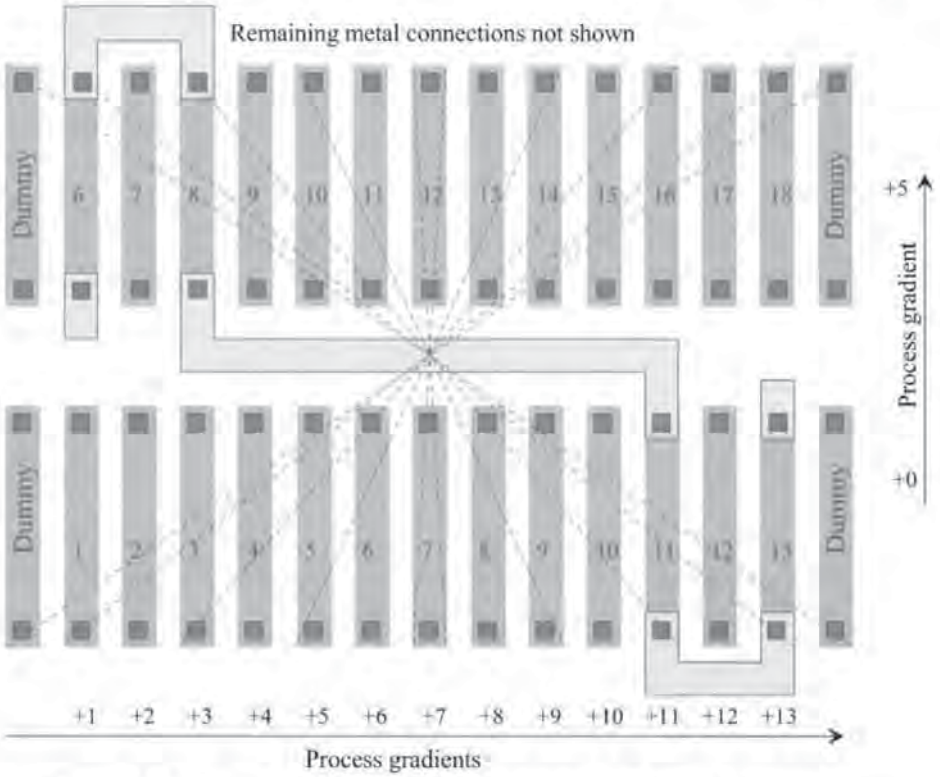



**Figure 5.29** Two-string layout for improving matching of R-2Rs. Assumes the resistors are connected together on higher levels of metal to avoid conductivity modulation.

IPR2023-00697
Theta EX2019

the top string of Fig. 5.29 (on the left) has a value of 14 (6 + 8). The MSB $2R$ in the bottom string (on the right) has a value of 24. Adding the values of the two resistors, by connecting them in series, results in a resistor value of 38 ($2R$ = 38 while $R$ = 19). The middle resistor value in the top string has a value of 12, while the bottom resistor has a value of 7. Again, adding the two resistors results in a value of 19. Fundamentally, the limiting factor in matching then becomes the voltage and temperature (because of the different current densities through the resistors) coefficients of the resistors.

## ADDITIONAL READING

[1] R. A. Pease, J. D. Bruce, H. W. Li, and R. J. Baker, "Comments on Analog Layout Using ALAS!" *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, September 1996, pp. 1364–1365.

[2] D. J. Allstot and W. C. Black, "Technology Design Considerations for Monolithic MOS Switched-Capacitor Filtering Systems," *Proceedings of the IEEE*, vol. 71, no. 8, August 1983, pp. 967–986.

## PROBLEMS

**5.1** Suppose a current in a circuit is given by

$$I = \frac{V_{REF}}{R}$$

If the voltage, $V_{REF}$, comes from a precision voltage reference and doesn't change with temperature, determine the temperature coefficient of the current in terms of the resistor's temperature coefficient. If the resistor is fabricated using the n-well plot, similar to Fig. 5.1, the current's change with temperature. Use the TCR1 given in Table 4.1.

**5.2** Suppose a silicided n+ resistor with a value of 100 Ω is used. Using the data from Table 4.1, sketch the layout and cross-sectional views of the resistor. The current in the resistor flows mainly in the silicide. Suppose the mobility of free carriers in the silicide is constant with increasing temperature. Would the temperature coefficient of the resistor be positive or negative? Why?

**5.3** Using a layout program, make a schematic and layout for the 1/5 voltage divider seen in Fig. 5.4 if the resistor's value is 5k. Use n-well resistors and DRC/LVS the final layout and schematic.

**5.4** Using a layout program, make a schematic and layout for an RC lowpass circuit where the resistor's value is 10k and the poly-poly capacitor's value is 100 fF. Use the 50 nm process (see Table 5.1). DRC/LVS the final layout and schematic. Simulate the operation of the circuit with a pulse input (see Fig. 1.27).

**5.5** Estimate the areas and perimeters of the source/drain in the layout seen in Fig. 5.18 if the length of the device, $L$, is 2 and the width of a finger, $W$, is 20.

**5.6** Provide a qualitative discussion for the capacitances of the PMOS device similar to the discussion associated with Fig. 5.21 for the NMOS device. Make sure the descriptions of operation in the strong inversion and depletion regions are clear. Draw the equivalent (to Fig. 5.21) figure for the PMOS devices.

IPR2023-00697
Theta EX2019

Chapter

6

# MOSFET Operation

In this chapter we discuss MOSFET operation. Figure 6.1 shows how we define the voltages, currents, and terminal designations for a MOSFET. When the substrate is connected to ground and the well is tied to *VDD*, we use the simplified models shown at the bottom of the figure. It is important to keep in mind that the MOSFET is a four-terminal device and that the source and drain of the MOSFET are interchangeable. Note that **all voltages and currents are positive** using the naming convention seen in the figure. If we say "the $V_{SG}$ of the MOSFET is...," we know that we are talking about a PMOS device. Also note, the drain current flows from the top of the symbol to the bottom. For the NMOS, the drain is at the top of the symbol, while for the PMOS the source is at the top of the symbol. The devices are complementary.

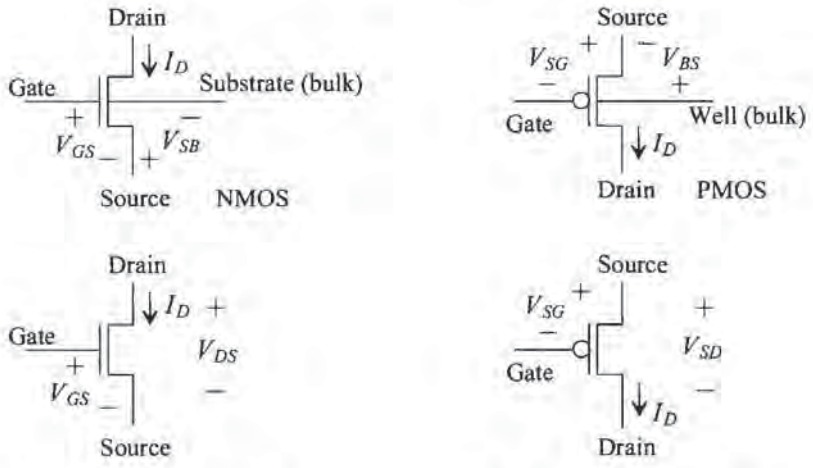



**Figure 6.1** Voltage and current designations for MOSFETs in this chapter.

IPR2023-00697
Theta EX2019

## 6.1 MOSFET Capacitance Overview/Review

In this section we'll discuss and review the capacitances of a MOSFET operating in the accumulation, depletion (weak inversion), and strong inversion regions.

*Case I: Accumulation*

Examine the cross-sectional view seen in Fig. 6.2. When $V_{GS} < 0$, mobile holes from the substrate are attracted (or *accumulated*) under the gate oxide. Remembering from Eq. (5.8) that

$$C'_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{6.1}$$

the capacitance between the *gate* electrode and the *substrate* electrode is given by

$$C_{gb} = \frac{\varepsilon_{ox}}{t_{ox}} \cdot (L_{drawn} - 2 \cdot L_{diff}) \cdot W_{drawn} \cdot (scale)^2 = C'_{ox} \cdot L_{eff} \cdot W_{drawn} \cdot (scale)^2 \tag{6.2}$$

where $\varepsilon_{ox}$ (= 3.97 · 8.85 aF/μm) is the dielectric constant of the gate oxide, $W_{drawn}$ is the drawn width (neglecting oxide encroachment), and $L_{drawn} - 2 \cdot L_{diff}$ is the effective channel length, $L_{eff}$. The capacitance between the gate and drain/source (the overlap capacitances, see Eq. [5.23]) is given by

$$C_{gs} = C'_{ox} \cdot L_{diff} \cdot W_{drawn} \cdot (scale)^2 = C_{gd} \tag{6.3}$$

neglecting oxide encroachment on the width of the MOSFET. The gate-drain overlap capacitance is present in a MOSFET regardless of the biasing conditions.

The total capacitance between the gate and ground in the circuit of Fig. 6.2 is the sum of $C_{gd}$, $C_{gs}$, and $C_{gb}$ and is given by

$$C_{gs} + C_{gb} + C_{gd} = C_{ox} = C'_{ox} \cdot L_{drawn} \cdot W_{drawn} \cdot (scale)^2 \tag{6.4}$$

There is a significant resistance in series with $C_{gb}$. The resistance comes from the physical distance between the substrate connection and the area under the gate oxide. The resistivity of the n+ source and drain regions in series with $C_{gs}$ and $C_{gd}$ tends to be small enough to neglect in most circuit design applications.

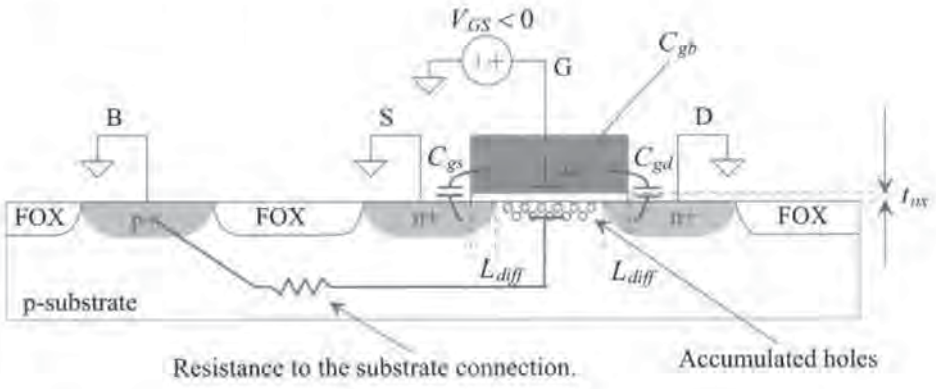



**Figure 6.2** Cross-sectional view of a MOSFET operating in accumulation.

IPR2023-00697
Theta EX2019

*Case II: Depletion*

Referring again to Fig. 6.2, let's consider the case when $V_{GS}$ is not negative enough to attract a large number of holes under the oxide and not positive enough to attract a large number of electrons. Under these conditions, the surface under the gate is said to be nearly depleted (depeleted of free electrons and holes). Consider the cross-sectional view seen in Fig. 6.3. As $V_{GS}$ is increased from some negative voltage, holes will be displaced under the gate, leaving immobile acceptor ions that contribute a negative charge. We see that as we increase $V_{GS}$ a capacitance between the gate and the induced (n) channel under the oxide exists. Also, a depletion capacitance between the depleted channel and the substrate is formed. The capacitance between the gate and the source/drain is simply the overlap capacitance, while the capacitance between the gate and the substrate is the oxide capacitance *in series* with the depletion capacitance. The depletion layer shown in Fig. 6.3 is formed between the substrate and the induced channel. The MOSFET operated in this region is said to be in *weak inversion* or the *subthreshold region* because the surface under the oxide is not heavily n+.

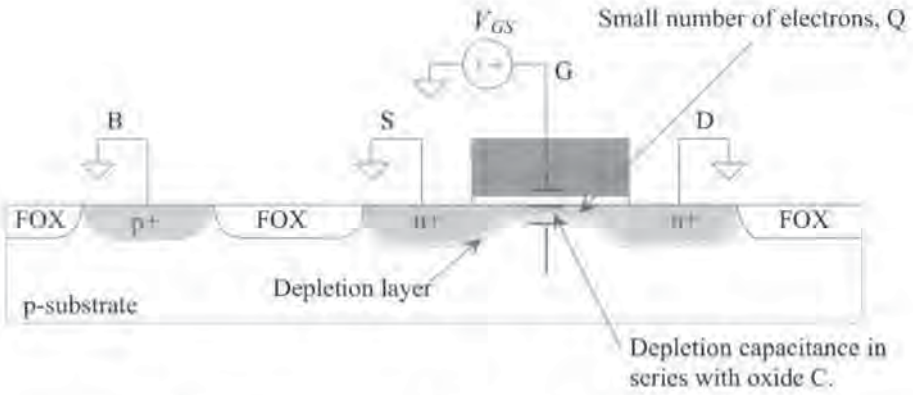



**Figure 6.3** Cross-sectional view of a MOSFET operating in depletion.

*Case III: Strong Inversion*

When $V_{GS}$ is sufficiently large (>> $V_{THN}$, the threshold voltage of the NMOS device) so that a large number of electrons are attracted under the gate, the surface is said to be inverted, that is, no longer p-type. Figure 6.4 shows how the capacitance at the gate changes as $V_{GS}$ is varied, for an NMOS device, when the source, drain, and bulk are grounded. This figure can be misleading. It may appear that we can operate the MOSFET in accumulation if we need a good capacitor. Remembering that when the MOSFET is in the accumulation region the majority of the capacitance to ground, $C_{gb}$, runs through the large parasitic resistance of the substrate, we see that to operate the MOSFET in this region we need plentiful substrate connections around the gate oxide (to reduce this parasitic substrate resistance). It's preferable to operate the MOSFET in strong inversion when we need a capacitor. The attracted electrons under the gate oxide short the drain and source together forming a low-resistance bottom plate for the capacitor. We will make a capacitor in this fashion many times when designing circuits.

IPR2023-00697
Theta EX2019

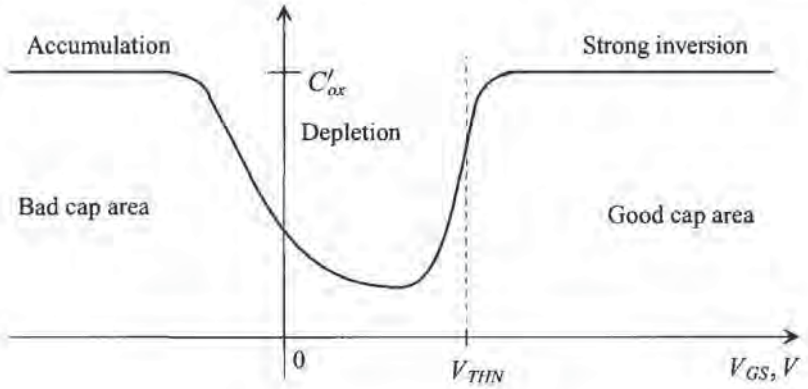



**Figure 6.4** The variation of the gate capacitance with DC gate-source voltage.

**Example 6.1**

Suppose the MOSFET configuration seen in Fig. 6.5 is to be used as a capacitor. If the width and length of the MOSFET are both 100, estimate the capacitance between the gate and the source/drain terminals. Use the long-channel process oxide capacitance listed in Table 5.1. Are there any restrictions on the voltages we can use across the capacitor?

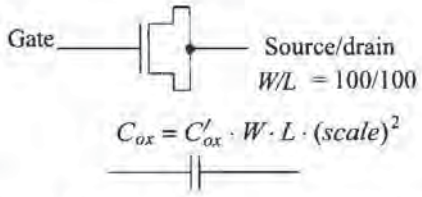



**Figure 6.5** Using the MOSFET as a capacitor.

Since the MOSFET is to be used as a capacitor, we require operation in the strong inversion region, that is, $V_{GS} >> V_{THN}$ (the gate potential at least a threshold voltage plus 5% of *VDD* above the source/drain potentials). The capacitance between the gate and the source/drain is then $C_{tot} = C_{ox} = C'_{ox} \cdot W \cdot L$ or from Table 5.1

$$C_{tot} = (1.75\,fF/\mu m^2)(100\ \mu m)(100\ \mu m) = 17.5\,pF$$

Note that we did not concern ourselves with the substrate connection. Since we are assuming strong inversion, the bulk (substrate) connection only affects the capacitances from the drain/source to substrate (those of the source/drain implant regions). We see, however, that the connection of the substrate significantly affects the threshold voltage of the devices and thus the point we label strong inversion. ■

IPR2023-00697
Theta EX2019

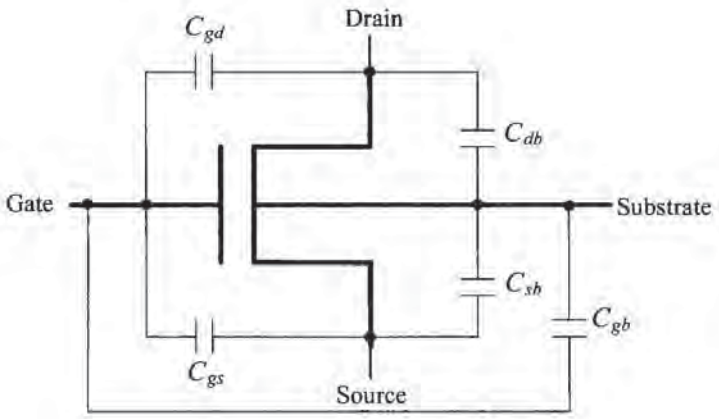



**Figure 6.6** MOSFET capacitances.

*Summary*

Figure 6.6 shows our MOSFET symbol with capacitances. Table 6.1 lists the capacitances based on the region of operation (without a scale factor). The capacitance *CGBO* is the capacitance associated with the gate poly extension over the field region (Fig. 5.13). The gate-drain capacitance, $C_{gd}$, and the gate-source capacitance, $C_{gs}$, are determined by the region of operation. For example, as we'll see later, when the MOSFET operates in the triode region, the inverted channel extends between the source and drain implants (the channel resistively connects the source and drain together). The capacitance between the gate and this channel is the oxide capacitance, $C_{ox}$. We assume that half of this capacitance is between the drain and the other half is between the source.

**Table 6.1** MOSFET capacitances.

| Name | Off | Triode | Saturation |
|---|---|---|---|
| $C_{gd}$ | $CGDO \cdot W$ | $\frac{1}{2} \cdot W \cdot L \cdot C'_{ox}$ | $CGDO \cdot W$ |
| $C_{db}$ | $C_{jd}$ | $C_{jd}$ | $C_{jd}$ |
| $C_{gb}$ | $C'_{ox} WL_{eff} + CGBO \cdot L$ | $CGBO \cdot L$ | $CGBO \cdot L$ |
| $C_{gs}$ | $CGSO \cdot W$ | $\frac{1}{2} \cdot W \cdot L \cdot C'_{ox}$ | $\frac{2}{3} \cdot W \cdot L \cdot C'_{ox}$ |
| $C_{sb}$ | $C_{js}$ | $C_{js}$ | $C_{js}$ |

## 6.2 The Threshold Voltage

In the last section we said that the semiconductor/oxide surface is inverted when $V_{GS}$ is greater than the threshold voltage $V_{THN}$. Under these conditions a channel of electrons is formed under the gate oxide. Below this channel, electrons fill the holes in the substrate

IPR2023-00697
Theta EX2019

giving rise to a depletion region (depleted of free carriers). The thickness of the depletion region (Fig. 6.7) is given from pn junction theory by

$$X_d = \sqrt{\frac{2\varepsilon_{si}|V_s - V_{fp}|}{qN_A}} \tag{6.5}$$

where $N_A$ is the number of acceptor atoms in the substrate, $V_s$ is the electrostatic potential at the oxide-silicon interface (the channel), and the electrostatic potential of the p-type substrate is given by (see Eq. [2.11])

$$V_{fp} = -\frac{E_i - E_{fp}}{q} = -\frac{kT}{q}\ln\frac{N_A}{n_i} \tag{6.6}$$

noting that this is a negative number. As seen in Fig. 6.7, one edge of the depletion region is the MOSFET's gate oxide, while the other edge is the p-substrate (holes). The positive potential on the gate attracts electrons under the gate oxide. This charge is equal and opposite to the charge in the polysilicon gate material. The charge/unit area is given by

$$Q'_b = qN_A X_d = \sqrt{2\varepsilon_{si} qN_A |V_s - V_{fp}|} \tag{6.7}$$

If the surface electrostatic potential at the oxide interface, $V_s$, is the same as the bulk electrostatic potential $V_{fp}$ (i.e., $V_s = V_{fp}$ and then $Q'_b = 0$), the MOSFET is operating in the accumulation mode, or the MOSFET is OFF in circuit terms. At this point the number of holes at the oxide-semiconductor surface is $N_A$, the same concentration as the bulk.

As $V_{GS}$ is increased, the surface potential becomes more positive. When $V_s = 0$, the surface under the oxide has become depleted (the carrier concentration is $n_i$). When $V_s = -V_{fp}$ (a positive number), the channel is inverted (electrons are pulled under the oxide forming a channel), and the electron concentration at the semiconductor-oxide interface is equal to the substrate doping concentration. The value of $V_{GS}$ when $V_s = -V_{fp}$ is arbitrarily defined as the threshold voltage, $V_{THN}$. Note that the surface potential changed a total of $2|V_{fp}|$ between the strong inversion and accumulation cases.

For $V_{GS} = V_{THN}$ ($V_s = -V_{fp}$), the negative charge under the gate oxide is given by

$$Q'_{bo} = \sqrt{2qN_A\varepsilon_{si}|-2V_{fp}|} \tag{6.8}$$

with units of Coulombs/m$^2$. Up to this point we have assumed that the substrate and source were tied together to ground. If the source of the NMOS device is at a higher potential than the substrate, the potential difference is given by $V_{SB}$; the negative charge under the gate oxide becomes

$$Q'_b = \sqrt{2qN_A\varepsilon_{si}|-2V_{fp}+V_{SB}|} \tag{6.9}$$

**Example 6.2**

For a substrate doping of $10^{15}$ atoms/cm$^3$, $V_{GS} = V_{THN}$ and $V_{SB} = 0$, estimate the electrostatic potential in the substrate region, $V_{fp}$, and at the oxide-semiconductor interface, $V_s$, the depletion layer width, $X_d$, and the charge contained in the depletion region, $Q'_b$, and thus the inverted region under the gate.

The electrostatic potential of the substrate is

IPR2023-00697
Theta EX2019

$$V_{fp} = -\frac{kT}{q}\ln\frac{N_A}{n_i} = -26\text{ mV}\cdot\ln\frac{10^{15}}{14.5\times10^{9}} = -290\text{ mV}$$

and therefore the electrostatic potential at the oxide semiconductor interface ($V_{GS} = V_{THN}$), $V_s$, is 290 mV. The depletion layer thickness is given by

$$X_d = \sqrt{\frac{2\cdot11.7\cdot(8.85\times10^{-18}F/\mu m)(2\cdot0.29V)}{(1.6\times10^{-19}\frac{C}{atom})(10^{15}\frac{atoms}{cm^3})(\frac{cm^3}{10^{12}\mu m^3})}} = 0.866\ \mu m$$

and the charge contained in this region, from Eq. (6.7) or (6.8) with $V_s = -V_{fp}$, by

$$Q'_{bo} = qN_AX_d = \left(1.6\times10^{-19}\frac{C}{atom}\right)\left(10^{15}\frac{atoms}{cm^3}\right)\left(\frac{cm^3}{10^{12}\mu m^3}\right)(0.866\,\mu m)$$

$$= 139\ \frac{aC}{\mu m^2}$$

Note that this is true only when $V_{GS} = V_{THN}$. ■

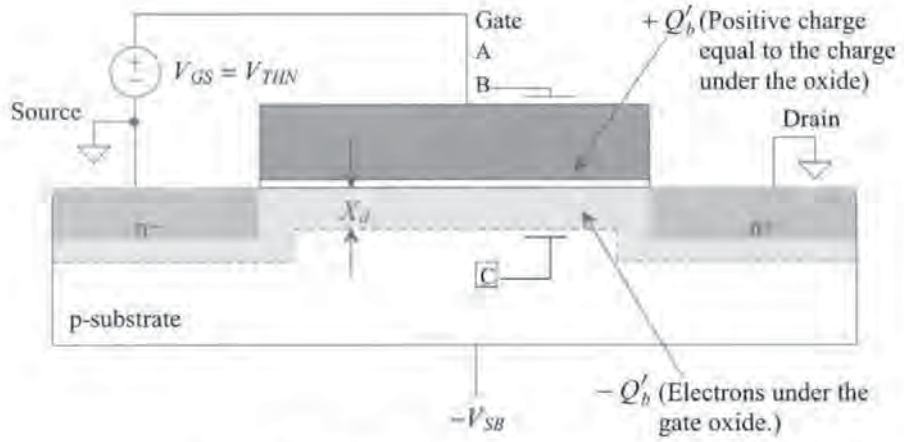



**Figure 6.7** Calculation of the threshold voltage.

*Contact Potentials*

Again, consider the MOSFET shown in Fig. 6.7. We assume that the applied $V_{GS} = V_{THN}$ so that the preceding discussions and assumptions hold. The potential across the gate-oxide capacitance, $C'_{ox}$, is simply

$$V_{BC} = \frac{Q'_b}{C'_{ox}} \tag{6.10}$$

The surface potential *change*, $V_C$ (= $\Delta V_s$), from the equilibrium case is $|2V_{fp}|$. (The absolute voltage of the channel is 0 V; that is, the source, drain, and channel are at ground.) The potential needed to change the surface potential and fill the fixed holes in the substrate is

IPR2023-00697
Theta EX2019

$$V_B = \frac{Q'_b}{C'_{ox}} - 2V_{fp} \tag{6.11}$$

An additional charge, $Q'_{ss}$ (columbs/area), can be used to model surface states (aka interface trapped charges, $Q'_{it}$) that may exist because of dangling bonds at the oxide-silicon interface. Here we assume electrons are attracted to the surface of the semiconductor (and trapped directly under the oxide) causing the threshold voltage to decrease. Equation (6.11) may be rewritten to include these surface-state charges as

$$V_B = \frac{Q'_b - Q'_{ss}}{C'_{ox}} - 2V_{fp} \tag{6.12}$$

The final component needed to determine the threshold voltage is the contact potential between point C (the bulk) and point A (the gate material) in Fig. 6.7. The potential difference between the gate and bulk (p-substrate) can be determined by summing the difference between the materials in the MOS system shown in Fig. 6.8. Adding the contact potentials, we get $(V_G - V_{ox}) + (V_{ox} - V_{fp}) = V_G - V_{fp}$. Note that if we were to include the surface electrostatic potential, $V_s$, the result would be the same, that is, only the two outer materials are of concern when calculating the contact potential difference. The contact potential between the bulk and the gate poly, we will assume n+ poly with doping concentration $N_{D,\,poly}$, is given by

$$V_{ms} = V_G - V_{fp} = \frac{kT}{q}\ln\left(\frac{N_{D,poly}}{n_i}\right) + \frac{kT}{q}\ln\frac{N_A}{n_i} \tag{6.13}$$

The threshold voltage, $V_{THN}$, is given by

$$V_{THN} = \frac{Q'_b - Q'_{ss}}{C'_{ox}} - 2V_{fp} - V_{ms} \tag{6.14}$$

$$= -V_{ms} - 2V_{fp} + \frac{Q'_{bo} - Q'_{ss}}{C'_{ox}} - \frac{Q'_{bo} - Q'_b}{C'_{ox}} \tag{6.15}$$

$$= -V_{ms} - 2V_{fp} + \frac{Q'_{bo} - Q'_{ss}}{C'_{ox}} + \frac{\sqrt{2q\varepsilon_{si}N_A}}{C'_{ox}}\left[\sqrt{|2V_{fp}| + V_{SB}} - \sqrt{|2V_{fp}|}\right] \tag{6.16}$$

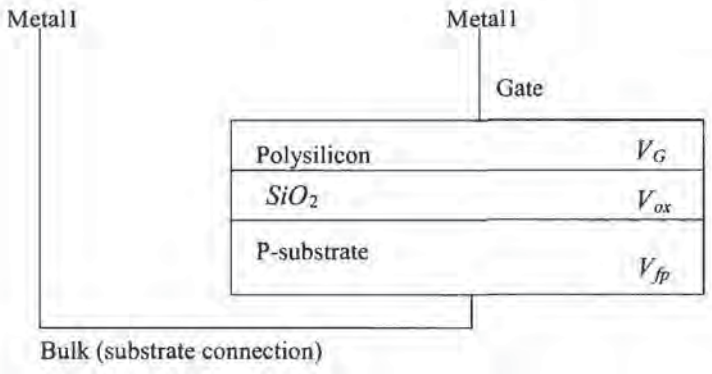



**Figure 6.8** Determining the contact potential between poly and substrate.

IPR2023-00697
Theta EX2019

When the source is shorted to the substrate, $V_{SB} = 0$, we can define the zero-bias threshold voltage as

$$V_{THN0} = -V_{ms} - 2V_{fp} + \frac{Q'_{bo} - Q'_{ss}}{C'_{ox}} \tag{6.17}$$

We can define a body effect coefficient or body factor by

$$\gamma = \frac{\sqrt{2q\varepsilon_{si}N_A}}{C'_{ox}} \tag{6.18}$$

Equation (6.16) can now be written as

$$V_{THN} = V_{THN0} + \gamma\left(\sqrt{|2V_{fp}| + V_{SB}} - \sqrt{|2V_{fp}|}\right) \tag{6.19}$$

It is interesting to note that a voltage, called the flatband voltage $V_{FB}$, must be applied for the oxide-semiconductor interface surface potential, $V_s$, to become the same potential as the bulk surface potential, $V_{fp}$. The flatband voltage is given by

$$V_{FB} = -V_{ms} - \frac{Q'_{ss}}{C'_{ox}} \tag{6.20}$$

The zero-bias threshold voltage may then be written in terms of the flatband voltage as

$$V_{THN0} = V_{FB} - 2V_{fp} + \frac{Q'_{bo}}{C'_{ox}} \tag{6.21}$$

These equations describe how the threshold voltage of the MOSFET is affected by substrate doping, oxide thickness, source/substrate bias, gate material, and surface charge density.

**Example 6.3**

Assuming $N_A = 10^{16}$ atoms/cm$^3$ and $C'_{ox} = 1.75$ fF/μm$^2$, estimate γ (GAMMA, the body effect coefficient).

From Eq. (6.18) the calculated γ is

$$\gamma = \frac{\sqrt{2 \cdot 1.6 \times 10^{-19} \frac{columbs}{atom} \cdot 11.7 \cdot 8.85 \frac{aF}{\mu m} \cdot 10^{16} \frac{atoms}{cm^3} \cdot \frac{cm^3}{10^{12} \mu m^3}}}{1.75 \frac{fF}{\mu m^2}} = 0.330\ V^{1/2} \blacksquare$$

**Example 6.4**

Estimate the zero-bias threshold voltage for the MOSFET of Ex. 6.2. Assume that the poly doping level is $10^{20}$ atoms/cm$^3$. What happens to the threshold voltage if sodium contamination causes an impurity of 40 aC/μm$^2$ at the oxide-semiconductor interface with $C'_{ox} = 1.75\ fF/\mu m^2$?

The electrostatic potential between the gate and substrate is given by

$$-V_{ms} = V_{fp} - V_G = -290\ mV - 26\ mV \cdot \ln\frac{10^{20}}{14.5 \times 10^9} = -879\ mV$$

$$-2V_{fp} = 580\ mV$$

IPR2023-00697
Theta EX2019

$$\frac{Q'_{bo}}{C'_{ox}} = \frac{139\ aC/\mu m^2}{1.75\ fF/\mu m^2} = 79\ mV$$

$$\frac{Q'_{ss}}{C'_{ox}} = 23\ mV$$

The threshold voltage, from Eq. (6.17) without the sodium contamination, is –220 mV; with the sodium contamination the threshold voltage is –243 mV. ■

*Threshold Voltage Adjust*

These threshold voltages would correspond to *depletion devices* (a negative threshold voltage), that is, MOSFETs that conduct when the $V_{GS} = 0$. In CMOS applications, this is highly undesirable. We normally use *enhancement devices* (devices with positive threshold voltages that are off with $V_{GS} = V_{SG} = 0$). To compensate or adjust the value of the threshold voltage (the channel, the area under the gate poly) can be implanted with p+ ions. This effectively increases the value of the threshold voltage by $Q'_c/C'_{ox}$, where $Q'_c$ is the charge density/unit area due to the implant. If $N_I$ is the ion implant dose in atoms/unit area, then we can write

$$Q'_c = q \cdot N_I \tag{6.22}$$

and the threshold voltage by

$$V_{THN0} = -V_{ms} - 2V_{fp} + \frac{Q'_{bo} - Q'_{ss} + Q'_c}{C'_{ox}} \tag{6.23}$$

**Example 6.5**
Estimate the ion implant dose required to change the threshold voltage in Ex. 6.4 without sodium contamination, to 1 V.

From Eqs. (6.22) and (6.23) and the results of Ex. 6.4

$$V_{THN0} = -220\ mV + \frac{qN_I}{C'_{ox}} = 1\ V$$

This gives $N_I = 1.3 \times 10^{12}$ atoms/cm$^2$. ■

These calculations lend some insight into how the threshold voltage is affected by the different process parameters. In practice, the results of these calculations do not exactly match the measured threshold voltage. From a circuit design engineer's point of view, the threshold voltage and the body factor are measured in the laboratory when the SPICE models are extracted.

## 6.3 IV Characteristics of MOSFETs

Now that we have some familiarity with the factors influencing the threshold voltage of a MOSFET, let's derive the large-signal IV (current/voltage) characteristics of the MOSFET, namely operation in the triode and the saturation regions. The following derivation is sometimes referred to as the *gradual-channel approximation*. The electric field variation in the channel between the source and drain (the y-direction) doesn't vary significantly when compared to the variation in the direction perpendicular to the channel (the x-direction).

IPR2023-00697
Theta EX2019

### 6.3.1 MOSFET Operation in the Triode Region

Consider Fig. 6.9, where $V_{GS} > V_{THN}$, so that the surface under the oxide is inverted and $V_{DS} > 0$, causing a drift current to flow from the drain to the source. In our initial analysis, we assume that $V_{DS}$ is sufficiently small so that the threshold voltage and the depletion layer width are approximately constant.

Initially, we must find the charge stored on the oxide capacitance $C'_{ox}$. The voltage, with respect to the source of the MOSFET, of the channel a distance $y$ away from the source is labeled $V(y)$. The potential difference between the gate electrode and the channel is then $V_{GS} - V(y)$. The charge/unit area in the inversion layer is given by

$$Q'_{ch} = C'_{ox} \cdot [V_{GS} - V(y)] \tag{6.24}$$

However, we know that a charge $Q'_b$ is present in the inversion layer from the application of the threshold voltage, $V_{THN}$, necessary for conduction between the drain and the source. This charge is given by

$$Q'_b = C'_{ox} \cdot V_{THN} \tag{6.25}$$

The total charge available in the inverted channel, for conduction of a current between the drain and the source, is given by the difference in these two equations, or

$$Q'_I(y) = C'_{ox} \cdot (V_{GS} - V(y) - V_{THN}) \tag{6.26}$$

The differential resistance of the channel region with a length $dy$ and a width $W$ is given by

$$dR = \overbrace{\frac{1}{\mu_n Q'_I(y)}}^{\text{eff. sheet Res.}} \cdot \frac{dy}{W} \tag{6.27}$$

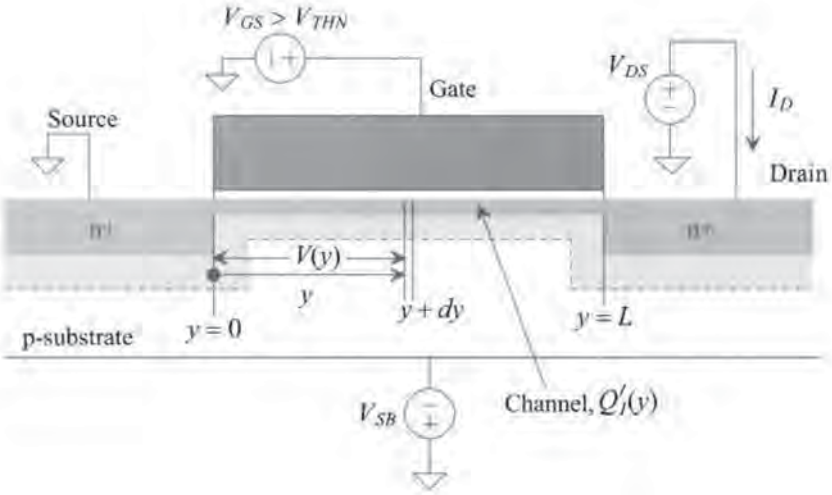



**Figure 6.9** Calculation of the large-signal behavior of the MOSFET in the triode (ohmic) region.

IPR2023-00697
Theta EX2019

where $\mu_n$ is the average electron mobility through the channel with units of $cm^2/V \cdot sec$ (see Eq. [5.4]). The mobility is simply a ratio of the electron (or hole) velocity cm/sec to the electric field, V/cm. For short-channel devices, the mobility decreases when the velocity of the carriers starts to saturate. This causes the effective sheet resistance in Eq. (6.27) to increase, resulting in a lowering of the drain current. This (velocity saturation) is discussed in more detail later in the chapter.

The differential voltage drop across this differential resistance is given by

$$dV(y) = I_D \cdot dR = \frac{I_D}{W\mu_n Q_I'(y)} \cdot dy \tag{6.28}$$

or substituting Eq. (6.26) and rearranging

$$I_D \cdot dy = W\mu_n C_{ox}'(V_{GS} - V(y) - V_{THN}) \cdot dV(y) \tag{6.29}$$

At this point, let's define the transconductance parameter, *KP*, for a MOSFET. For an n-channel MOSFET, this parameter is given by

$$KP_n = \mu_n \cdot C_{ox}' = \mu_n \cdot \frac{\varepsilon_{ox}}{t_{ox}} \tag{6.30}$$

and for a p-channel MOSFET, it is given by

$$KP_p = \mu_p \cdot C_{ox}' = \mu_p \cdot \frac{\varepsilon_{ox}}{t_{ox}} \tag{6.31}$$

where $\mu_p$ is the mobility of the holes in a PMOS transistor. Typical values of *KP* in the *long-channel* process (with a minimum length of 1 μm) used in this book are 120 $\mu A/V^2$ and 40 $\mu A/V^2$ for n- and p-channel transistors, respectively.

The current can be obtained by integrating the left side of Eq. (6.29) from the source to the drain, that is, from 0 to *L* and the right side from 0 to $V_{DS}$. This is shown below:

$$I_D \int_0^L dy = W \cdot KP_n \cdot \int_0^{V_{DS}} (V_{GS} - V(y) - V_{THN}) \cdot dV(y) \tag{6.32}$$

or

$$I_D = KP_n \cdot \frac{W}{L} \cdot \left[ (V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2} \right] \text{ for } V_{GS} \geq V_{THN} \text{ and } V_{DS} \leq V_{GS} - V_{THN} \tag{6.33}$$

This equation is valid when the MOSFET is operating in the triode (aka **linear** or **ohmic**) region. This is the case when the induced channel extends from the source to the drain. Furthermore, we can rewrite Eq. (6.33) defining the transconductance parameter as

$$\beta = KP_n \cdot \frac{W}{L} \tag{6.34}$$

or

$$I_D = \beta \cdot \left[ (V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2} \right] \tag{6.35}$$

IPR2023-00697
Theta EX2019

The equivalent equation for the PMOS device operating in the triode region is

$$I_D = KP_p \cdot \frac{W}{L} \cdot \left[ (V_{SG} - V_{THP}) V_{SD} - \frac{V_{SD}^2}{2} \right] \text{ for } V_{SG} \geq V_{THP} \text{ and } V_{SD} \leq V_{SG} - V_{THP} \quad (6.36)$$

where the threshold voltage of the p-channel MOSFET is positive (noting, again, that from our sign convention in Fig. 6.1 all voltages and currents are positive.)

### 6.3.2 The Saturation Region

The voltage at $V(y)$ when $y = L$, that is, $V(L)$, in Eq. (6.26) is simply $V_{DS}$. In the previous subsection, we said that $V_{DS}$ is always less than $V_{GS} - V_{THN}$ so that at no point along the channel is the inversion charge zero. When $V_{DS} = V_{GS} - V_{THN}$, the inversion charge under the gate at $y = L$ (the drain-channel junction) is zero, Eq. (6.26). This drain-source voltage is called $V_{DS,sat}$ $(= V_{GS} - V_{THN})$, and indicates when the channel charge becomes *pinched off* at the drain-channel interface. Increases in $V_{DS}$ beyond $V_{DS,sat}$ attract the fixed channel charge to the drain terminal depleting the charge in the channel directly adjacent to the drain. Further increases in $V_{DS}$ do not cause an increase in the drain current[1]. In other words the *current saturates* and thus stops increasing.

Figure 6.10 shows the depletion region, with a thickness of $X_{dl}$, between the drain and channel. An increase in $V_{DS}$ results in an increase in $X_{dl}$. If $V_{DS}$ is increased until $X_{dl}$ extends from the drain to the source, the device is said to be *punched through*. Large currents can flow under these conditions, causing device failure. The maximum voltage, for near minimum-size channel lengths, that can be applied between the drain and source of a MOSFET is set by the "punchthrough" voltage. For long-channel lengths, the maximum voltage is set by the breakdown voltage of the drain (n+) to substrate diode.

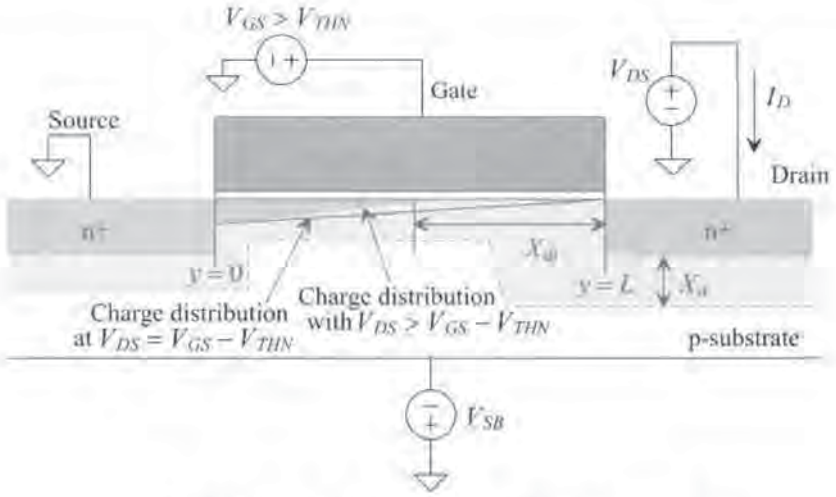



**Figure 6.10** The MOSFET in saturation (pinched off).

[1] We will see on the next page that this is not entirely true. An effect called *channel length modulation* causes the drain current to increase slightly with increasing drain-source voltage.

IPR2023-00697
Theta EX2019

When a MOSFET's channel is pinched off, that is, $V_{DS} \geq V_{GS} - V_{THN}$ and $V_{GS} \geq V_{THN}$, it is operating in the saturation region. Substitution of $V_{DS,sat}$ into Eq. (6.33) yields

$$I_{D,sat} = \frac{KP_n}{2} \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN})^2 = \frac{\beta}{2}(V_{GS} - V_{THN})^2$$

$$\text{for } V_{DS} \geq V_{GS} - V_{THN} \text{ and } V_{GS} \geq V_{THN} \qquad (6.37)$$

We can define an electrical channel length, $L_{elec}$, of the MOSFET as the difference between the drawn channel length, $L$, neglecting lateral diffusion, and the depletion layer width, $X_{dl}$

$$L_{elec} = L - X_{dl} \qquad (6.38)$$

Substituting this into Eq. (6.37), we obtain a better representation of the drain current

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L_{elec}}(V_{GS} - V_{THN})^2 \qquad (6.39)$$

Qualitatively, this means that since the depletion layer width increases with increasing $V_{DS}$, the drain current increases as well. This effect is called *channel length modulation* (CLM). As $L$ is increased the effects of $X_{dl}$ changing (CLM) become negligible.

To determine the change in output current with drain-source voltage, we take the derivative of Eq. (6.39) with respect to $V_{DS}$ around $V_{DS,sat}$ (when $L \approx L_{elec}$)

$$\frac{dI_D}{dV_{DS}} = -\frac{KP_n}{2}\frac{W}{L_{elec}^2}(V_{GS} - V_{THN})^2 \cdot \frac{dL_{elec}}{dV_{DS}} = I_{D,sat} \cdot \left[\frac{1}{L}\frac{dX_{dl}}{dV_{DS}}\right] \qquad (6.40)$$

where it's common to define $\lambda$, the channel length modulation parameter, as

$$\lambda = \frac{1}{L} \cdot \frac{dX_{dl}}{dV_{DS}} \qquad (6.41)$$

Typical values for $\lambda$ range from greater than 0.1 $V^{-1}$ to less than 0.01 $V^{-1}$ (ideally $\lambda = 0$). Note that the units of $dV_{DS}/dX_{dl}$ are V/m and that this term grows as process technology shrinks. The smaller devices are designed to drop larger voltages over smaller distances. This last point is why *a nanometer device can't be made to behave like a long-channel device simply by increasing its length.*

Equation (6.37) can be rewritten to account for CLM as

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - V_{THN})^2[1 + \lambda(V_{DS} - V_{DS,sat})] = I_{D,sat} \cdot [1 + \lambda(V_{DS} - V_{DS,sat})]$$

$$\text{for } V_{DS} > V_{DS,sat} = V_{GS} - V_{THN} \text{ and } V_{GS} > V_{THN} \qquad (6.42)$$

The drain current at the triode/saturation region border occurs when

$$I_D = I_{D,sat} \text{ and } V_{DS} = V_{DS,sat} = V_{GS} - V_{THN} \qquad (6.43)$$

In these equations we assumed that the mobility does not vary with $V_{DS}$. Later in the chapter (in the short-channel MOSFET discussion, Sec. 6.5.2) we'll see that the mobility does indeed vary with $V_{DS}$ making characterizing $I_D$ considerably more challenging. Figure 6.11 shows typical curves for an n-channel MOSFET. Notice how the device *appears to go into saturation earlier* than predicted by $V_{DS,sat} = V_{GS} - V_{THN}$. The bold line in the figure separates the actual triode and saturation regions (and also indicates $I_{D,sat}$).

IPR2023-00697
Theta EX2019

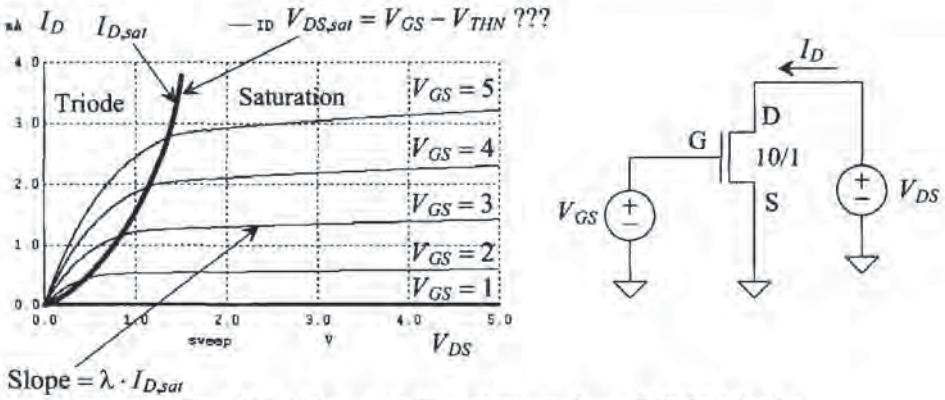



**Figure 6.11** Characterisitics of a long-channel NMOS device.

For example, in the simulation results at $V_{GS} = 5$ V (with $V_{THN}$ roughly equal to 1 V), we see that $V_{DS,sat}$ is 1.4 V (not the 4 V we calculate using $V_{GS} - V_{THN}$) The actual charge distribution in the channel is not constant but rather a function of $V_{DS}$. $Q'_I(y)$ decreases as we move away from the source of the MOSFET, causing $Q'_I(L)$ to become zero earlier, see Fig. 6.10.

*$C_{gs}$ Calculation in the Saturation Region*

The gate-to-source capacitance of a MOSFET operating in the saturation region can be determined by solving Eq. (6.26) for the total charge in the inverted channel,

$$Q_I = \int_0^L W \cdot Q'_I(y) \cdot dy = WC'_{ox} \int_0^L (V_{GS} - V(y) - V_{THN})\, dy \tag{6.44}$$

or solving Eq. (6.29) for $dy$ and substituting yields

$$Q_I = \frac{(W \cdot C'_{ox})^2 \cdot \mu_n}{I_D} \int_0^{V_{GS}-V_{THN}} (V_{GS} - V(y) - V_{THN})^2 \cdot dV(y) \tag{6.45}$$

where it was used that $Q_I$ goes to zero when $y = L$ occurs when $V_{DS} = V_{GS} - V_{THN}$. Solving this equation using Eqs. (6.30) and (6.37) yields

$$Q_I = \frac{2}{3} \cdot W \cdot L \cdot C'_{ox} \cdot (V_{GS} - V_{THN}) \tag{6.46}$$

We can determine the gate-to-source capacitance while in the saturation region by,

$$C_{gs} = \frac{\partial Q_I}{\partial V_{GS}} = \frac{2}{3} \cdot W \cdot L \cdot C'_{ox} \tag{6.47}$$

See the entry in Table 6.1 (note the discontinuity between saturation and triode).

## 6.4 SPICE Modeling of the MOSFET

In this section we list the level 1, 2, and 3 SPICE model parameters and their relationship to the equations derived in the last section. The level 1 model is a subset of the level 2 and 3 models which have more elaborate mobility modeling. All three models are based on Eq. (6.42). A level 3 model for a long-channel CMOS process is also presented.

IPR2023-00697
Theta EX2019

*Model Parameters Related to $V_{THN}$*

The following SPICE model parameters are related to the calculation of $V_{THN}$,

| Symbol | Name | Description | Default | Typ. | Units |
|---|---|---|---|---|---|
| $V_{THN0}$ | VTO | Zero-bias threshold voltage | 1.0 | 0.8 | Volts |
| $\gamma$ | GAMMA | Body-effect parameter | 0 | 0.4 | $V^{1/2}$ |
| $2\lvert V_{fp}\rvert$ | PHI | Surface to bulk potential | 0.65 | 0.58 | V |
| $N_A$ | NSUB | Substrate doping | 0 | 1E15 | $cm^{-3}$ |
| $Q'_{ss}/q$ | NSS | Surface state density | 0 | 1E10 | $cm^{-2}$ |
| | TPG | Type of gate material | 1 | 1 | |

Using Eq. (6.19), we can calculate the threshold voltage, $V_{THN}$, given the above parameters. If $V_{THN0}$ or $\gamma$ are not given, then SPICE calculates them using the above information and Eqs. (6.19) – (6.21). TPG specifies the type of gate material: 1 opposite to substrate, –1 same as substrate, and 0 for aluminum gate.

*Long-Channel MOSFET Models*

The SPICE models used in this book for the "long-channel CMOS process" follow. The scale factor is 1 μm (= minimum drawn channel length).

```
* 1 um Level 3 models
* Don't forget the .options scale=1u if using an Lmin of 1
* 1<L<200 and 10<W<10000 Vdd=5V

.MODEL NMOS NMOS LEVEL = 3
+ TOX  = 200E-10      NSUB  = 1E17       GAMMA = 0.5
+ PHI  = 0.7          VTO   = 0.8        DELTA = 3.0
+ UO   = 650          ETA   = 3.0E-6     THETA = 0.1
+ KP   = 120E-6       VMAX  = 1E5        KAPPA = 0.3
+ RSH  = 0            NFS   = 1E12       TPG   = 1
+ XJ   = 500E-9       LD    = 100E-9
+ CGDO = 200E-12      CGSO  = 200E-12    CGBO  = 1E-10
+ CJ   = 400E-6       PB    = 1          MJ    = 0.5
+ CJSW = 300E-12      MJSW  = 0.5
*
.MODEL PMOS PMOS LEVEL = 3
+ TOX  = 200E-10      NSUB  = 1E17       GAMMA = 0.6
+ PHI  = 0.7          VTO   = -0.9       DELTA = 0.1
+ UO   = 250          ETA   = 0          THETA = 0.1
+ KP   = 40E-6        VMAX  = 5E4        KAPPA = 1
+ RSH  = 0            NFS   = 1E12       TPG   = -1
+ XJ   = 500E-9       LD    = 100E-9
+ CGDO = 200E-12      CGSO  = 200E-12    CGBO  = 1E-10
+ CJ   = 400E-6       PB    = 1          MJ    = 0.5
+ CJSW = 300E-12      MJSW  = 0.5
```

*Model Parameters Related to the Drain Current*

The SPICE implemention of the square-law equations, Eqs. (6.35) and (6.42), is slightly different [8] than our derivations. Equation [6.42] is implemented in SPICE with $V_{DS,sat} = 0$. To avoid a discontinuity then Eq. (6.35) is multiplied by $(1 + \lambda \cdot V_{DS})$.

IPR2023-00697
Theta EX2019

| Symbol | Name | Description | Default | Typ. | Units |
|---|---|---|---|---|---|
| $KP$ | KP | Transconductance parameter | 20E-6 | 50E-6 | $A/V^2$ |
| $t_{ox}$ | TOX | Gate-oxide thickness | 1E-7 | 40E-10 | m |
| $\lambda$ | Lambda | Channel-length modulation | 0 | 0.01 | $V^{-1}$ |
| $L_{diff}$ | LD | Lateral diffusion | 0 | 2.5E-7 | m |
| $\mu_{n,p}$ | UO | Surface mobility | 600 | 580 | $cm^2/Vs$ |

*SPICE Modeling of the Source and Drain Implants*

| Name | Description | Default | Typical | Units |
|---|---|---|---|---|
| RD | Drain contact resistance | 0 | 40 | Ω |
| RS | Source contact resistance | 0 | 40 | Ω |
| RSH | Source/drain sheet resistance | 0 | 50 | Ω/sq. |
| CGBO | Gate-bulk overlap capacitance | 0 | 4E-10 | F/m |
| CGDO | Gate-drain overlap capacitance | 0 | 4E-10 | F/m |
| CGSO | Gate-source overlap capacitance | 0 | 4E-10 | F/m |
| PB, PBSW | Bottom, sidewall built-in potential | 0.8 | 0.8 | V |
| MJ, MJSW | Bottom, sidewall grading coefficient | 0.6 | 0.6 | |
| CJ | Bottom zero-bias depletion capacitance | 0 | 3E-4 | $F/m^2$ |
| CJSW | Sidewall zero-bias depletion capacitance | 0 | 2.5E-10 | F/m |
| IS | Bulk-junction saturation current | 1E-14 | 1E-14 | A |
| JS | Bulk-junction saturation current density | 0 | 1E-8 | $A/m^2$ |
| FC | Bulk-junction forward bias coefficient | 0.6 | 0.6 | |

*Summary*

Table 6.2 lists the characteristics of the long-channel CMOS process used in this book.

**Table 6.2** Summary of device characteristics for the long-channel CMOS process.

| Long-channel MOSFET parameters used in this book. The $VDD$ = 5 V and the scale factor is **1 μm** ($scale$ = 1e–6) | | | |
|---|---|---|---|
| **Parameter** | **NMOS** | **PMOS** | **Comments** |
| $V_{THN}$ and $V_{THP}$ | 800 mV | 900 mV | Typical |
| $KP_n$ and $KP_p$ | 120 μA/V² | 40 μA/V² | $t_{ox}$ = 200 Å |
| $C'_{ox} = \varepsilon_{ox}/t_{ox}$ | 1.75 $f$F/μm² | 1.75 $f$F/μm² | $C_{ox} = C'_{ox}WL \cdot (scale)^2$ |
| $\lambda_n$ and $\lambda_p$ | 0.01 $V^{-1}$ | 0.0125 $V^{-1}$ | at $L$ = 2 |
| $\gamma_n$ and $\gamma_p$ | 0.5 $V^{-1/2}$ | 0.6 $V^{-1/2}$ | Body factor |

IPR2023-00697
Theta EX2019

### 6.4.1 Some SPICE Simulation Examples

Figure 6.11 shows the $I_D$-$V_{DS}$ characteristics of an NMOS device in the long-channel process. Figure 6.12 shows the equivalent PMOS device. Notice that the devices are the same size, 10/1, in the two simulations; however, the drain current of the PMOS is less than half the drain current of the NMOS. This is related to the mobility of the holes being two to three times lower than the mobility of the electrons. Electrons in the valence band are more tightly coupled to the nucleus of an atom than are electrons in the conduction band. Because apparent movement of holes is actually the result of electrons moving in the valence band, the hole mobility is lower than electron (in conduction band) mobility.

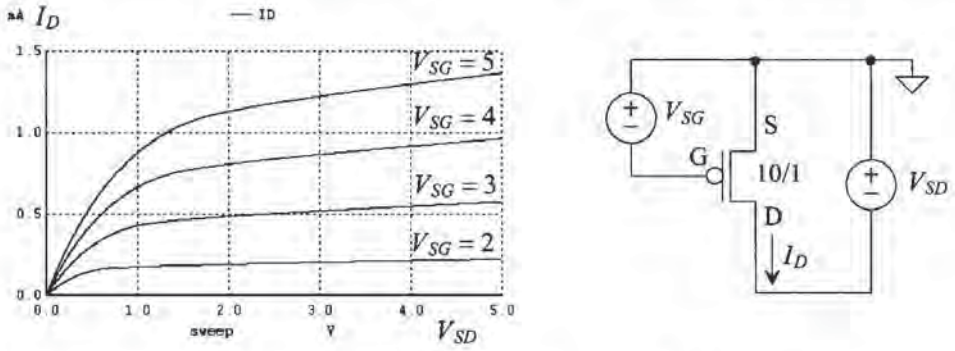



**Figure 6.12** Characterisitics of a long-channel PMOS device.

*Threshold Voltage and Body Effect*

In simple terms, the threshold voltage is the voltage that turns the device on and allows drain current to flow from the drain to the source. Figure 6.13 shows the $I_D$-$V_{GS}$ curves for an NMOS device. When the source and substrate are at the same potential, $V_{SB} = 0$, the threshold voltage is labeled $V_{THN0}$ (see Eq. [6.17]). When $V_{SB}$ starts to increase, the threshold voltage goes up. This is called the body effect. Figure 6.14 shows two MOSFETs: one with and one without body effect (substrate, or body, is grounded).

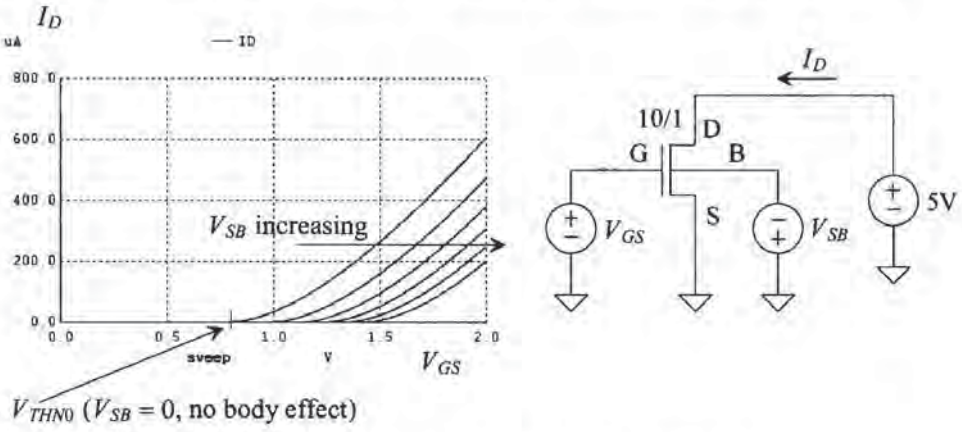



**Figure 6.13** Threshold voltage and body effect.

IPR2023-00697
Theta EX2019

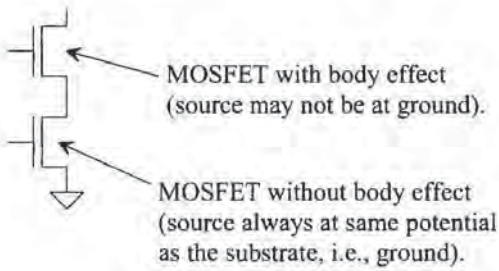



**Figure 6.14** How an NMOS can have body effect.

To qualitatively understand the origin of the body effect, consider the MOSFET cross-sectional view seen in Fig. 6.15. As the source potential rises above the bulk (substrate) potential (represented by $V_{SB}$ in the figure), electrons are attracted towards the positive terminal of $V_{SB}$ from the MOSFET's channel. To keep the surface inverted, a larger $V_{GS}$ must be applied to the MOSFET. Thus the effect of the body stealing charge from the channel is an increase in the MOSFET's threshold voltage.

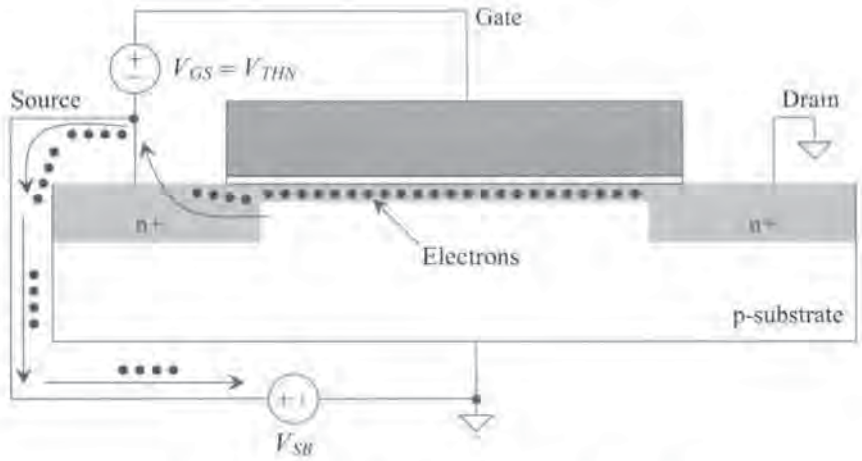



**Figure 6.15** Qualitative description of body effect.

### 6.4.2 The Subthreshold Current

In the last section we said that the MOSFET starts to conduct a current when $V_{GS} = V_{THN}$. In reality there is a drain current, albeit small, when $V_{GS} < V_{THN}$. This current is called subthreshold current. When the MOSFET is operating in the weak inversion region it can also be said to be operating in the subthreshold region. Subthreshold operation can be very useful for low-power operation. Solar-powered calculators, CMOS imagers, or battery- operated watches are examples of devices using CMOS ICs operating in the subthreshold region. The main problems that plague circuits designed to operate in the subthreshold region are matching, noise, and bandwidth. For example, since the drain current is exponentially related to the gate-source voltage (as we'll soon see), any mismatch in these voltages can cause significant differences in the drain current.

IPR2023-00697
Theta EX2019

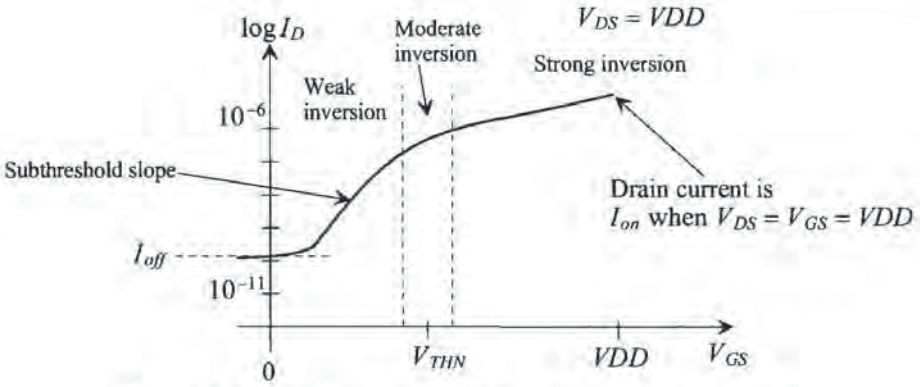



**Figure 6.16** Drain current plotted from weak to strong inversion.

The subthreshold region is often characterized using the log $I_D$ plotted against $V_{GS}$ (see Fig. 6.16). The current transport from the drain to source in Sec. 6.3 was via drift. An applied electric field, when the MOSFET is operating in the strong inversion region, causes carriers to drift from the channel to the drain across the depletion region (and hence why we talk about mobility). In the weak inversion, or subthreshold region, the carriers diffuse from the source to the drain just like carrier movement in a bipolar junction transistor, BJT. In a BJT the carriers are emitted from the emitter, diffuse across the base, and are collected at the collector. In a MOSFET operating in subthreshold, the carriers are emitted by the source, diffuse across the body of the device (under the gate oxide) and are collected at the drain. We can write the drain current of the MOSFET in the subthreshold region as

$$I_D = I_{D0} \cdot \frac{W}{L} \cdot e^{q(V_{GS} - V_{THN})/(n \cdot kT)} \tag{6.48}$$

Taking the log of both sides with $V_T = kT/q$ (the thermal voltage), we get

$$\log I_D = \log \frac{W}{L} + \log I_{D0} + -\frac{V_{THN}}{nV_T} \cdot \log e + \overbrace{\left[\frac{1}{V_T \cdot n} \cdot \log e\right]}^{\text{subthreshold slope}} \cdot V_{GS} \tag{6.49}$$

The reciprocal of the subthreshold slope is given by

$$\text{Subthreshold slope}^{-1} = \frac{V_T \cdot n}{\log e} \text{ (mV/decade)} \tag{6.50}$$

If $kT/q = 0.026$ V $= V_T$ and $n$ (the slope parameter) = 1, the reciprocal of the subthreshold slope is 60 mV/decade (it can be said the subthreshold slope is 60 mV/decade and it is understood it is actually one over the slope). In bulk CMOS $n$ is around 1.6 and the subthreshold slope is 100 mV/decade at room temperature. For the ideal MOSFET used as a switch when $V_{GS}$ is less than the threshold voltage, the drain current goes to zero. The slope of the curve below $V_{THN}$ in Fig. 6.16 is then infinite (corresponding to zero subthreshold slope$^{-1}$). The subthreshold slope can be a very important MOSFET parameter in many applications (the design of dynamic circuits). Notice that the drain current that flows with $V_{GS} = 0$ is called $I_{off}$ (with $V_{DS} = VDD$). The drain current that flows when $V_{GS} = V_{DS} = VDD$ (in the strong inversion region) is called $I_{on}$.

IPR2023-00697
Theta EX2019

## 6.5 Short-Channel MOSFETs

The long-channel CMOS process used in the first part of this chapter is useful for illustrating the fundamentals of MOSFET operation. However, modern CMOS transistors have channel lengths that are well below the 1 μm minimum length of this process. The gradual channel approximation used earlier to develop the square-law current-voltage characteristics of the MOSFET falls apart for modern short-channel devices. The electric field under the gate oxide can no longer be treated in a single dimension. In addition, the velocity of the carriers drifting between the channel and the drain of the MOSFET can saturate, Fig. 6.17, an effect called *carrier velocity saturation*, $v_{sat}$. Typical values for the low electric field mobilities (electric fields, $E$, less than the critical electric field, $E_{crit}$, where the velocity saturates) for electrons and holes are 600 cm$^2$/Vs ($\mu_n$) and 250 cm$^2$/Vs ($\mu_p$). See text associated with Eq. (5.4) for additional information.

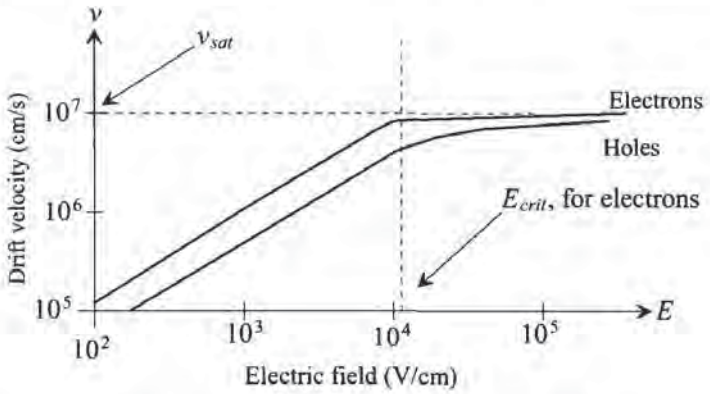



**Figure 6.17** Drift velocity plotted against electric field. The slope of these curves is the mobility of the carriers, see Eq. (5.4)

*Hot Carriers*

In very small devices some of the carriers drifting near the drain can obtain energies much larger than the thermal energy of carriers under equilibrium conditions. These carriers are termed *hot carriers*. The carrier scattering events are no longer localized allowing the carriers to have higher, than the values mentioned above, mobilities. The velocity of these carriers can exceed the saturation velocity indicated in Fig. 6.17. This effect is called *velocity overshoot* and it can enhance the speed and transconductance of the MOSFETs (which is good). Unfortunately, hot carriers can also tunnel through the gate oxide and cause gate current or become trapped in the gate oxide, having the effect of changing the MOSFET's threshold voltage. Hot carriers can also cause impact ionization (avalanche multiplication).

*Lightly-Doped Drain (LDD)*

A cross-sectional view of an NMOS device using a lightly doped drain (LDD) structure is shown in Fig. 6.18 (and Fig. 4.7). The addition of the lightly doped n- provides a resistive buffer between the channel and the higher-doped source/drain. The effect of the LDD structure is to increase the voltage dropped across the drain-channel interface, $dV_{DS}/dX_{dl}$.

IPR2023-00697
Theta EX2019

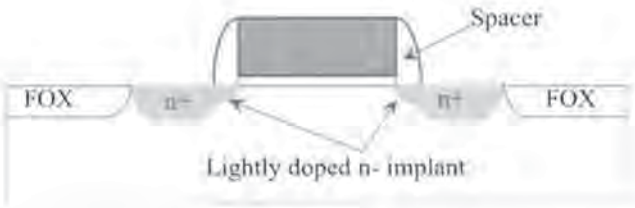



**Figure 6.18** Lightly doped drain (LDD) implant.

### 6.5.1 MOSFET Scaling

Reducing the channel length of a MOSFET can be described in terms of scaling theory. A scaling parameter $S$ ($S < 1$) is used to scale the dimensions of a MOSFET. The value of $S$ is typically in the neighborhood of 0.7 from one CMOS technology generation to the next. For example, if a process uses a $VDD$ of 2 V, a next generation process would use a $VDD$ of 1.4 V. In other words

$$VDD' = VDD \cdot S \tag{6.51}$$

The channel length of the scaled process is reduced to

$$L' = L \cdot S \tag{6.52}$$

while the width is reduced to

$$W' = W \cdot S \tag{6.53}$$

Table 6.3 describes how $S$ affects the MOSFET parameters. The main benefits of scaling are (1) smaller device sizes and thus reduced chip size (increased yield and more parts per wafer), (2) lower gate delays, allowing higher frequency operation, and (3) reduction in power dissipation. Associated with these benefits are some unwanted side effects referred to as short-channel effects. These unwanted effects are discussed in the next section.

**Table 6.3** CMOS scaling relationships.

| Parameter | Scaling |
|---|---|
| Supply voltage ($VDD$) | S |
| Channel length ($L_{min}$) | S |
| Channel width ($W_{min}$) | S |
| Gate-oxide thickness ($t_{ox}$) | S |
| Substrate doping ($N_A$) | $S^{-1}$ |
| On current ($I_{on}$) | S |
| Gate capacitance ($C_{ox}$) | S |
| Gate delay | S |
| Active power | $S^3$ |

IPR2023-00697
Theta EX2019

### 6.5.2 Short-Channel Effects

The average drift velocity, $v$, of an electron plotted against electric field, $E$, was shown in Fig. 6.17. When the electric field reaches a critical value, labeled $E_{crit}$, the velocity saturates at a value $v_{sat}$, that is, the velocity ceases to increase with increasing electric field (note that here we are neglecting the potential for velocity overshoot). The ratio of electron drift velocity to applied electric field is the electron mobility (Eq. 5.4), or, again

$$\mu_n = \frac{v}{E} \tag{6.54}$$

Above the critical electric field, the mobility starts to decrease, whereas below $E_{crit}$, the mobility is essentially constant. Rewriting Eq. (6.29) to determine how the mobility changes with $V(y)$ results in

$$I_D = \mu_n \cdot \frac{dV(y)}{dy} \cdot W \cdot C'_{ox}[V_{GS} - V_{THN} - V(y)] \tag{6.55}$$

We are interested in determining how the drain current of a short-channel MOSFET changes with $V_{GS}$ when operating in the saturation region. (The charge under the gate oxide at the drain channel interface is zero, and the channel is pinched off.) The MOSFET enters the saturation region when $V(L) = V_{DS,sat}$. At high electric fields, the mobility can be approximated by

$$\mu_n = \frac{v_{sat}}{E} = \frac{v_{sat}}{dV(y)/dy} \tag{6.56}$$

so that Eq. (6.55) can be written as

$$I_D = W \cdot v_{sat} \cdot C'_{ox}(V_{GS} - V_{THN} - V_{DS,sat}) \tag{6.57}$$

The drain current of a short-channel MOSFET operating in the saturation region increases linearly with $V_{GS}$. The long-channel theory, Eq. (6.37), shows the drain current increasing with the square of the gate-source voltage. This result also presents a practical relative figure of merit for the modern CMOS process, the drive current per width of a MOSFET. The on or drive current, $I_{on}$ or $I_{drive}$ (μA/μm), is given by

$$I_{on} = I_{drive} = v_{sat} \cdot C'_{ox}(V_{GS} - V_{THN} - V_{DS,sat}) \tag{6.58}$$

and therefore, see Fig. (6.16),

$$I_D = I_{on} \cdot W = I_{drive} \cdot W \text{ with } V_{GS} = V_{DS} = VDD \tag{6.59}$$

The on (drive) current can be estimated using these equations; however, it is normally measured.

*Negative Bias Temperature Instability (NBTI)*

In a modern PMOS device when the gate voltage is driven below its source voltage ($V_{SG} > 0$) crucial device parameters, such as the threshold voltage, are observed to shift over time. Historically, both the trapping of holes in oxide defects and the creation of interface states have been suspected to be the cause of the shift. NBTI can be a significant reliability concern in $SiO_2$ gate dielectrics due to time and temperature-dependent fluctuations in device parameters during both on and off states of operation. NBTI is also present in NMOS devices but it is considerably more pronounced in the PMOS transistor.

IPR2023-00697
Theta EX2019

*Oxide Breakdown*

For reliable device operation, the maximum electric field across a device gate oxide should be limited to 10 MV/cm. This translates into 1V / 10 Å of gate oxide. A device with $t_{ox}$ of 20 Å should limit the applied gate voltages to 2 V for reliable long-term operation.

*Drain-Induced Barrier Lowering*

Drain-induced barrier lowering (DIBL, pronounced "dibble") causes a threshold voltage reduction with the application of a drain-source voltage. The positive potential at the drain terminal helps to attract electrons under the gate oxide and thus increase the surface potential $V_s$. In other words $V_{DS}$ helps to invert the channel on the drain side of the device, causing a reduction in the threshold voltage. Since $V_{THN}$ decreases with increasing $V_{DS}$, the result is an increase in drain current and thus a decrease in the MOSFET's output resistance.

*Gate-Induced Drain Leakage*

Gate-Induced Drain Leakage (GIDL, pronounced "giddle") is a term used to describe a component of the drain to substrate leakage current. When the device is in accumulation (e.g. the gate of an NMOS device is at ground) the surface and substrate potentials are nearly the same. In this situation there can be a dramatic increase in avalanche multiplication or band-to-band tunneling when the drain is at a higher potential. Minority carriers underneath the gate are swept to the substrate creating the leakage current.

*Gate Tunnel Current*

As the oxide thickness scales downwards, the probability of carriers directly tunneling through the gate oxide increases. For oxide thicknesses less than 15 Å, this gate current can be significant. To reduce the tunnel current, various sandwiches of dielectrics are being explored. Figure 16.67 later in the book presents some results showing values for direct tunnel currents under various operating conditions.

### 6.5.3 SPICE Models for Our Short-Channel CMOS Process

Section 6.4 presented some SPICE models for the long-channel CMOS process used in this book. In this section we give the BSIM4[2] models for the 50 nm process we use in the book with *VDD* = 1 V, see also Table 5.2. The model listing is given below.

*BSIM4 Model Listing (NMOS)*

```
* 50nm BSIM4 models
*
* Don't forget the .options scale=50nm if using an Lmin of 1
* 1<Ldrawn<200   10<Wdrawn<10000 Vdd=1V
*
.model        nmos      nmos      level = 54
+binunit = 1            paramchk= 1         mobmod  = 0
+capmod  = 2            igcmod  = 1         igbmod  = 1         geomod  = 1
+diomod  = 1            rdsmod  = 0         rbodymod= 1         rgatemod= 1
```

[2] BSIM4 is a fourth generation MOSFET model developed at the University of California, Berkeley. The acronym stands for Berkeley Short-channel IGFET (insulated gate FET) Model. For more information see: http://www-device.eecs.berkeley.edu

IPR2023-00697
Theta EX2019

```
+permod  = 1          acnqsmod= 0          trnqsmod= 0

+tnom    = 27         toxe    = 1.4e-009   toxp    = 7e-010    toxm    = 1.4e-009
+epsrox  = 3.9        wint    = 5e-009     lint    = 1.2e-008
+ll      = 0          wl      = 0          lln     = 1         wln     = 1
+lw      = 0          ww      = 0          lwn     = 1         wwn     = 1
+lwl     = 0          wwl     = 0          xpart   = 0         toxref  = 1.4e-009

+vth0    = 0.22       k1      = 0.35       k2      = 0.05      k3      = 0
+k3b     = 0          w0      = 2.5e-006   dvt0    = 2.8       dvt1    = 0.52
+dvt2    = -0.032     dvt0w   = 0          dvt1w   = 0         dvt2w   = 0
+dsub    = 2          minv    = 0.05       voffl   = 0         dvtp0   = 1e-007
+dvtp1   = 0.05       lpe0    = 5.75e-008  lpeb    = 2.3e-010  xj      = 2e-008
+ngate   = 5e+020     ndep    = 2.8e+018   nsd     = 1e+020    phin    = 0
+cdsc    = 0.0002     cdscb   = 0          cdscd   = 0         cit     = 0
+voff    = -0.15      nfactor = 1.2        eta0    = 0.15      etab    = 0
+vfb     = -0.55      u0      = 0.032      ua      = 1.6e-010  ub      = 1.1e-017
+uc      = -3e-011    vsat    = 1.1e+005   a0      = 2         ags     = 1e-020
+a1      = 0          a2      = 1          b0      = -1e-020   b1      = 0
+keta    = 0.04       dwg     = 0          dwb     = 0         pclm    = 0.18
+pdiblc1 = 0.028      pdiblc2 = 0.022      pdiblcb = -0.005    drout   = 0.45
+pvag    = 1e-020     delta   = 0.01       pscbe1  = 8.14e+8   pscbe2  = 1e-007
+fprout  = 0.2        pdits   = 0.2        pditsd  = 0.23      pditsl  = 2.3e+006
+rsh     = 3          rdsw    = 150        rsw     = 150       rdw     = 150
+rdswmin = 0          rdwmin  = 0          rswmin  = 0         prwg    = 0
+prwb    = 6.8e-011   wr      = 1          alpha0  = 0.074     alpha1  = 0.005
+beta0   = 30         agidl   = 0.0002     bgidl   = 2.1e+009  cgidl   = 0.0002
+egidl   = 0.8

+aigbacc = 0.012      bigbacc = 0.0028     cigbacc = 0.002
+nigbacc = 1          aigbinv = 0.014      bigbinv = 0.004     cigbinv = 0.004
+eigbinv = 1.1        nigbinv = 3          aigc    = 0.017     bigc    = 0.0028
+cigc    = 0.002      aigsd   = 0.017      bigsd   = 0.0028    cigsd   = 0.002
+nigc    = 1          poxedge = 1          pigcd   = 1         ntox    = 1

+xrcrg1  = 12         xrcrg2  = 5
+cgso    = 6.238e-010 cgdo    = 6.238e-010 cgbo    = 2.56e-011 cgdl    = 2.495e-10
+cgsl    = 2.495e-10  ckappas = 0.02       ckappad = 0.02      acde    = 1
+moin    = 15         noff    = 0.9        voffcv  = 0.02

+kt1     = -0.21      kt1l    = 0.0        kt2     = -0.042    ute     = -1.5
+ua1     = 1e-009     ub1     = -3.5e-019  uc1     = 0         prt     = 0
+at      = 53000

+fnoimod = 1          tnoimod = 0

+jss     = 0.0001     jsws    = 1e-011     jswgs   = 1e-010    njs     = 1
+ijthsfwd= 0.01       ijthsrev= 0.001      bvs     = 10        xjbvs   = 1
+jsd     = 0.0001     jswd    = 1e-011     jswgd   = 1e-010    njd     = 1
+ijthdfwd= 0.01       ijthdrev= 0.001      bvd     = 10        xjbvd   = 1
+pbs     = 1          cjs     = 0.0005     mjs     = 0.5       pbsws   = 1
+cjsws   = 5e-010     mjsws   = 0.33       pbswgs  = 1         cjswgs  = 3e-010
+mjswgs  = 0.33       pbd     = 1          cjd     = 0.0005    mjd     = 0.5
+pbswd   = 1          cjswd   = 5e-010     mjswd   = 0.33      pbswgd  = 1
+cjswgd  = 5e-010     mjswgd  = 0.33       tpb     = 0.005     tcj     = 0.001
+tpbsw   = 0.005      tcjsw   = 0.001      tpbswg  = 0.005     tcjswg  = 0.001
+xtis    = 3          xtid    = 3


+dmcg    = 0e-006     dmci    = 0e-006     dmdg    = 0e-006    dmcgt   = 0e-007
```

IPR2023-00697
Theta EX2019

```
+dwj    = 0.0e-008      xgw    = 0e-007      xgl    = 0e-008

+rshg   = 0.4           gbmin  = 1e-010      rbpb   = 5          rbpd   = 15
+rbps   = 15            rbdb   = 15          rbsb   = 15         ngcon  = 1
```

*BSIM4 Model Listing (PMOS)*

```
.model      pmos        pmos      level = 54

+binunit = 1            paramchk= 1          mobmod  = 0
+capmod  = 2            igcmod  = 1          igbmod  = 1         geomod  = 1
+diomod  = 1            rdsmod  = 0          rbodymod= 1         rgatemod= 1
+permod  = 1            acnqsmod= 0          trnqsmod= 0

+tnom    = 27           toxe    = 1.4e-009   toxp    = 7e-010    toxm    = 1.4e-009
+epsrox  = 3.9          wint    = 5e-009     lint    = 1.2e-008
+ll      = 0            wl      = 0          lln     = 1         wln     = 1
+lw      = 0            ww      = 0          lwn     = 1         wwn     = 1
+lwl     = 0            wwl     = 0          xpart   = 0         toxref  = 1.4e-009

+vth0    = -0.22        k1      = 0.39       k2      = 0.05      k3      = 0
+k3b     = 0            w0      = 2.5e-006   dvt0    = 3.9       dvt1    = 0.635
+dvt2    = -0.032       dvt0w   = 0          dvt1w   = 0         dvt2w   = 0
+dsub    = 0.7          minv    = 0.05       voffl   = 0         dvtp0   = 0.5e-008
+dvtp1   = 0.05         lpe0    = 5.75e-008  lpeb    = 2.3e-010  xj      = 2e-008
+ngate   = 5e+020       ndep    = 2.8e+018   nsd     = 1e+020    phin    = 0
+cdsc    = 0.000258     cdscb   = 0          cdscd   = 6.1e-008  cit     = 0
+voff    = -0.15        nfactor = 2          eta0    = 0.15      etab    = 0
+vfb     = 0.55         u0      = 0.0095     ua      = 1.6e-009  ub      = 8e-018
+uc      = 4.6e-013     vsat    = 90000      a0      = 1.2       ags     = 1e-020
+a1      = 0            a2      = 1          b0      = -1e-020   b1      = 0
+keta    = -0.047       dwg     = 0          dwb     = 0         pclm    = 0.55
+pdiblc1 = 0.03         pdiblc2 = 0.0055     pdiblcb = 3.4e-008  drout   = 0.56
+pvag    = 1e-020       delta   = 0.014      pscbe1  = 8.14e+08  pscbe2  = 9.58e-07
+fprout  = 0.2          pdits   = 0.2        pditsd  = 0.23      pditsl  = 2.3e+006
+rsh     = 3            rdsw    = 250        rsw     = 160       rdw     = 160
+rdswmin = 0            rdwmin  = 0          rswmin  = 0         prwg    = 3.22e-008
+prwb    = 6.8e-011     wr      = 1          alpha0  = 0.074     alpha1  = 0.005
+beta0   = 30           agidl   = 0.0002     bgidl   = 2.1e+009  cgidl   = 0.0002
+egidl   = 0.8

+aigbacc = 0.012        bigbacc = 0.0028     cigbacc = 0.002
+nigbacc = 1            aigbinv = 0.014      bigbinv = 0.004     cigbinv = 0.004
+eigbinv = 1.1          nigbinv = 3          aigc    = 0.69      bigc    = 0.0012
+cigc    = 0.0008       aigsd   = 0.0087     bigsd   = 0.0012    cigsd   = 0.0008
+nigc    = 1            poxedge = 1          pigcd   = 1         ntox    = 1

+xrcrg1  = 12           xrcrg2  = 5
+cgso    = 7.43e-010    cgdo    = 7.43e-010  cgbo    = 2.56e-011 cgdl    = 1e-014
+cgsl    = 1e-014       ckappas = 0.5        ckappad = 0.5       acde    = 1
+moin    = 15           noff    = 0.9        voffcv  = 0.02

+kt1     = -0.19        kt1l    = 0          kt2     = -0.052    ute     = -1.5
+ua1     = -1e-009      ub1     = 2e-018     uc1     = 0         prt     = 0
+at      = 33000

+fnoimod = 1            tnoimod = 0

+jss     = 0.0001       jsws    = 1e-011     jswgs   = 1e-010    njs     = 1
```

IPR2023-00697
Theta EX2019

| | | | |
|---|---|---|---|
| +ijthsfwd= 0.01 | ijthsrev= 0.001 | bvs = 10 | xjbvs = 1 |
| +jsd = 0.0001 | jswd = 1e-011 | jswgd = 1e-010 | njd = 1 |
| +ijthdfwd= 0.01 | ijthdrev= 0.001 | bvd = 10 | xjbvd = 1 |
| +pbs = 1 | cjs = 0.0005 | mjs = 0.5 | pbsws = 1 |
| +cjsws = 5e-010 | mjsws = 0.33 | pbswgs = 1 | cjswgs = 3e-010 |
| +mjswgs = 0.33 | pbd = 1 | cjd = 0.0005 | mjd = 0.5 |
| +pbswd = 1 | cjswd = 5e-010 | mjswd = 0.33 | pbswgd = 1 |
| +cjswgd = 5e-010 | mjswgd = 0.33 | tpb = 0.005 | tcj = 0.001 |
| +tpbsw = 0.005 | tcjsw = 0.001 | tpbswg = 0.005 | tcjswg = 0.001 |
| +xtis = 3 | xtid = 3 | | |
| +dmcg = 5e-006 | dmci = 5e-006 | dmdg = 5e-006 | dmcgt = 6e-007 |
| +dwj = 4.5e-008 | xgw = 3e-007 | xgl = 4e-008 | |
| +rshg = 0.4 | gbmin = 1e-010 | rbpb = 5 | rbpd = 15 |
| +rbps = 15 | rbdb = 15 | rbsb = 15 | ngcon = 1 |

*Simulation Results*

Figure 6.19 shows 10/1 PMOS and NMOS device simulation results using the topologies seen in Figs. 6.11–6.13. The actual device sizes are 500 nm (width) by 50 nm (length). From the information in this figure and knowing *VDD* is 1 V, we can estimate the on currents for the MOSFETs. For the NMOS device

$$I_{on,n} \approx 300\ \mu A/(W \cdot scale) = 600\ \mu A/\mu m \tag{6.60}$$

For the PMOS device

$$I_{on,p} \approx 150\ \mu A/(W \cdot scale) = 300\ \mu A/\mu m \tag{6.61}$$

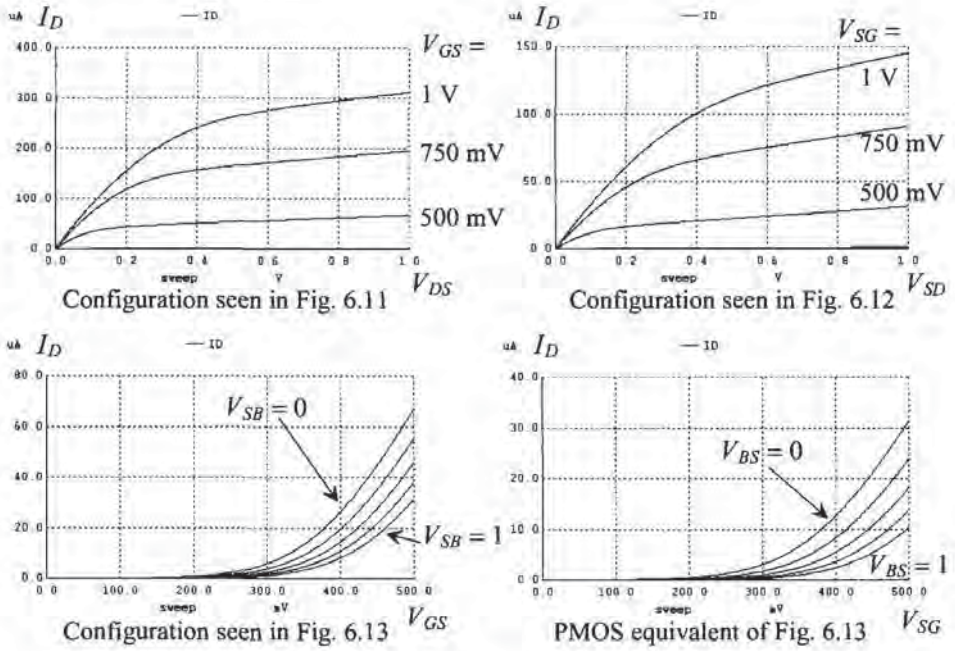



**Figure 6.19** Current-voltage characteristics for 50 nm MOSFETs.

IPR2023-00697
Theta EX2019

The threshold voltages can be estimated as 280 mV. (See Fig. 9.27 and the associated discussion for more information on determining the threshold voltages.) Note that it **doesn't make sense** to try to define a transconductance parameter, *KP*, for a short-channel process (the MOSFETs don't follow the square-law equations, Eqs. [6.33] and [6.37].) Instead we use $I_{on}$, $I_{off}$, $t_{ox}$, *W*, *scale*, $V_{THN,P}$, *VDD*, and plots of measured data. Table 6.4 shows some of the device characteristics for the short-channel CMOS process used in this book. Note that we **don't confuse** $I_{on}$ (Fig. 6.16) with $I_{D,sat}$ (the current at the border between triode and saturation, Figs. 6.11 or 9.4 and Eq. [6.43]).

**Table 6.4** Summary of device characteristics for the short-channel CMOS process.

| *Short*-channel MOSFET parameters used in this book. The *VDD* = 1 V and the scale factor is **50 nm** (*scale* = 50e–9) | | | |
|---|---|---|---|
| **Parameter** | **NMOS** | **PMOS** | **Comments** |
| $V_{THN}$ and $V_{THP}$ | 280 mV | 280 mV | Typical |
| $t_{ox}$ | 14 Å | 14 Å | See also Table 5.1 |
| $C'_{ox} = \varepsilon_{ox}/t_{ox}$ | $25 f$F/μm$^2$ | $25 f$F/μm$^2$ | $C_{ox} = C'_{ox}WL \cdot (scale)^2$ |
| $\lambda_n$ and $\lambda_p$ | 0.6 V$^{-1}$ | 0.3 V$^{-1}$ | At $L = 2$ |
| $I_{on,n}$ and $I_{on,p}$ | 600 μA/μm | 300 μA/μm | On current |
| $I_{off,n}$ and $I_{off,p}$ | 7.1 nA/μm | 10 nA/μm | Off current, see Fig. 14.2 |

## ADDITIONAL READING


[1] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, Second Edition, Cambridge University Press, 2010. ISBN 978-0521832946

[2] J.R. Brews, MOSFET Hand Analysis Using BSIM, *IEEE Circuits and Devices Magazine*, vol. 22, No. 1, pp. 28-36, January/February 2006.

[3] R. S. Muller, T. I. Kamins, and M. Chan, *Device Electronics for Integrated Circuits*, John Wiley and Sons Publishers, 2002. ISBN 0-471-59398-2

[4] R. C. Jaeger, *Introduction to Microelectronic Fabrication*, 2nd ed, volume 5 of the Modular Series on Solid State Devices, Prentice-Hall Publishers, 2002. ISBN 0-20-144494-1

[5] W. Liu, *MOSFET Models for Spice Simulation, Including BSIM3v3 and BSIM4*, John Wiley and Sons Publishers, 2001. ISBN 0-471-39697-4

[6] Y. P. Tsividis, *Operation and Modeling of the MOS Transistor*, 2nd ed., Oxford University Press, 1999. ISBN 978-0195170146.

[7] M. Bohr, "MOS Transistors: Scaling and Performance Trends," *Semiconductor International*, pp. 75–79, June 1995.

[8] G. Massobrio and P. Antognetti, *Semiconductor Device Modeling with SPICE, Second Edition*, McGraw-Hill, 1993. Excellent reference for SPICE modeling.

[9] D. A. Neamen, *Semiconductor Physics and Devices-Basic Principles*, Richard D. Irwin, 1992. ISBN 0-256-08405-X.


IPR2023-00697
Theta EX2019

[10] D. K. Schroder, *Modular Series on Solid State Devices-Advanced MOS Devices,* Addison-Wesley, 1987.

[11] S. M. Sze, *Physics of Semiconductor Devices,* 2nd ed., John-Wiley and Sons, 1981. ISBN 0-471-05661-8.

**PROBLEMS**

**6.1** Plot the magnitude and phase of $v_{out}$ (AC) in the following circuit, Fig. 6.20. Assume that the MOSFET was fabricated using the 50 nm process (see Table 5.1) and is operating in strong inversion. Verify your answer with SPICE.

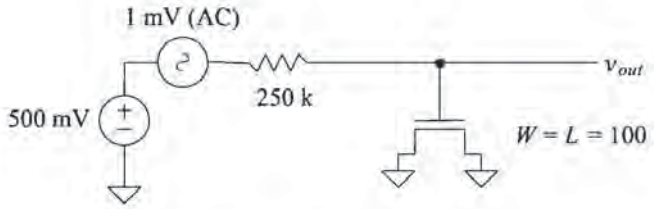



**Figure 6.20** Circuit used in Problem 6.1.

**6.2** If a MOSFET is used as a capacitor in the strong inversion region where the gate is one electrode and the source/drain is the other electrode, does the gate overlap of the source/drain change the capacitance? Why or not? What is the capacitance?

**6.3** Repeat Problem 6.2 when the MOSFET is operating in the accumulation region. Keep in mind that the question is not asking for the capacitance from gate to substrate.

**6.4** If the oxide thickness of a MOSFET is 40 Å, what is $C'_{ox}$?

**6.5** Repeat Ex. 6.2 when $V_{SB} = 1$ V.

**6.6** Repeat Ex. 6.3 for a p-channel device with a well doping concentration of $10^{16}$ atoms/cm$^3$.

**6.7** What is the electrostatic potential of the oxide-semiconductor interface when $V_{GS} = V_{THN0}$?

**6.8** Repeat Ex. 6.5 to get a threshold voltage of 0.8 V.

**6.9** What happens to the threshold voltage in Problem 6.8 if sodium contamination of $100 \times 10^9$ sodium ions/cm$^2$ is present at the oxide-semiconductor interface?

**6.10** How much charge (enhanced electrons) is available under the gate for conducting a drain current at the drain-channel interface when $V_{DS} = V_{GS} - V_{THN}$? Why? Assume that the MOSFET is operating in strong inversion, $V_{GS} > V_{THN}$.

**6.11** Show the details of the derivation for Eq. (6.33) for the PMOS device.

IPR2023-00697
Theta EX2019

**6.12** Using Eq. (6.35), estimate the small-signal channel resistance (the change in the drain current with changes in the drain-source voltage) of a MOSFET operating in the triode region (the resistance between the drain and source).

**6.13** Show, using Eqs. (6.33) and (6.37), that the parallel connection of MOSFETs shown in Fig. 5.18 behave as a single MOSFET with a width equal to the sum of each individual MOSFET width.

**6.14** Show that the bottom MOSFET, Fig. 6.21, in a series connection of two MOSFETs cannot operate in the saturation region. Neglect the body effect. *Hint*: Show that M1 is always either in cutoff ($V_{GS1} < V_{THN}$) or triode ($V_{DS1} < V_{GS1} - V_{THN}$).

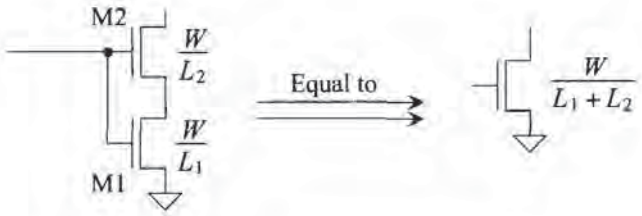



**Figure 6.21** MOSFETs operating in series.

**6.15** Show that the series connection of MOSFETs shown in Fig. 6.21 behaves as a single MOSFET with twice the length of the individual MOSFETs. Again, neglect the body effect.

IPR2023-00697
Theta EX2019

Chapter
7

# CMOS Fabrication

This chapter presents a brief overview of CMOS process integration. Process integration refers to the well-defined collection of semiconductor processes required to fabricate CMOS integrated circuits starting from virgin silicon wafers. This overview is intended to give the reader a fundamental understanding of the processes required in CMOS integrated circuit fabrication. Moreover, there are strong interactions between circuit design and process integration. For instance, the typical design rule set is determined in large part by the limitations in the fabrication processes. Hence, circuit designers, process engineers, and integration engineers are required to communicate effectively. To this end, we first examine the fundamental processes, called unit processes, required for CMOS fabrication. The primary focus is the qualitative understanding of the processes with limited introduction of quantitative expressions. The unit processes are combined in a deliberate sequence to fabricate CMOS. Additionally, the unit processes are typically repeated numerous times in a given process sequence. Here we present a representative modern CMOS process sequence, also called a process flow.

## 7.1 CMOS Unit Processes

In this section we introduce each of the major processes required in the fabrication of CMOS integrated circuits. We first discuss wafer production. Although wafer production is not a unit process, it is nonetheless important to present the production method which is used by wafer manufacturers. All subsequent discussions are focused on the unit processes incorporated by fabrication facilities to produce integrated circuits. The unit processes are grouped by functionality. Thermal oxidation, doping processes, photolithography, thin-film removal, and thin-film deposition techniques are presented.

### 7.1.1 Wafer Manufacture

Silicon is the second most abundant element in the Earth's crust; however, it occurs exclusively in compounds. In fact, elemental silicon is a man-made material that is refined from these various compounds. The most common is silica (impure $SiO_2$). Modern integrated circuits must be fabricated on ultrapure, defect-free slices of single crystalline silicon called wafers, as discussed in Ch. 1.

IPR2023-00697
Theta EX2019

*Metallurgical Grade Silicon (MGS)*

Wafer production requires three general processes: silicon refinement, crystal growth, and wafer formation. Silicon refinement begins with the reduction of silica in an arc furnace at roughly 2000 °C with a carbon source. The carbon effectively "pulls" the oxygen from the $SiO_2$ molecules, thus chemically reducing the $SiO_2$ into roughly 98% pure silicon referred to as metallurgical grade silicon (MGS). The overall reduction is governed by the following equation

$$SiO_2 \text{ (solid)} + 2C \text{ (solid)} \rightarrow Si \text{ (liquid)} + 2CO \text{ (gas)} \qquad (7.1)$$

*Electronic Grade Silicon (EGS)*

MGS is not sufficiently pure for microelectronic device applications. The reason is that the electronic properties of a semiconductor such as silicon are extremely sensitive to impurity concentrations. Impurity levels measured at parts per million or less can have dramatic effects on carrier mobilities, lifetimes, etc. It is therefore necessary to further purify the MGS in what is known as electronic grade silicon (EGS). EGS is produced from the chlorination of grounded MGS as

$$MGS \text{ (solid)} + HCL \text{ (gas)} \rightarrow \text{silane } (SiH_4) \text{ (liquid)} + \text{trichlorosilane } (SiHCL_3) \text{ (liquid)} \qquad (7.2)$$

Because the reaction products are liquids at room temperature, ultrapure EGS can be obtained from fractional distillation and chemical reduction processes. The resultant EGS is in the form of polycrystalline chunks.

*Czochralski (CZ) Growth and Wafer Formation*

To achieve a single crystalline form, the EGS must be subjected to a process called Czochralski (CZ) growth. A schematic representation of the CZ growth process is shown in Fig. 7.1. The polycrystalline EGS is melted in a large quartz crucible where a small seed crystal of known orientation is introduced into the surface of the silicon melt. The seed crystal, rotating in one direction, is slowly pulled from the silicon melt, rotating in the opposite direction. Solidification of the silicon onto the seed forms a growing crystal

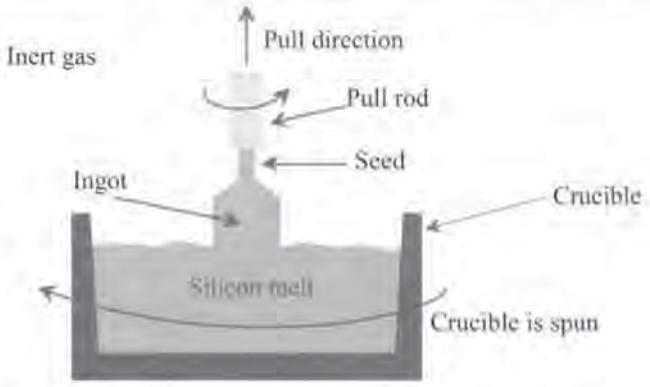



**Figure 7.1** Simplified diagram showing Czochralski (CZ) crystal growth.

IPR2023-00697
Theta EX2019

(called a boule or ingot) that assumes the crystallographic orientation of the seed. In general, the slower the pull-rate (typically mm/hour), the larger the diameter of the silicon crystal. Following CZ growth, the silicon boule is trimmed down to the appropriate diameter. *Flats* or *notches* are ground into the surface of the boule to indicate a precise crystal orientation. Using a special diamond saw, the silicon boule is cut into thin wafers. The wafers are finished by using a chemical mechanical polishing (CMP) process to yield a mirror-like finish on one side of the wafer. Although devices are fabricated entirely within the top couple of micrometers of the wafer, final wafer thicknesses (increasing with wafer diameter) are up to roughly one millimeter for adequate mechanical support.

### 7.1.2 Thermal Oxidation

Silicon, when exposed to an oxidant at elevated temperatures, readily forms a thin layer of oxide at all exposed surfaces. The native oxide of silicon is in the form of silicon dioxide ($SiO_2$). With respect to CMOS fabrication, $SiO_2$ can serve as a high quality dielectric in device structures such as gate oxides. Moreover, during processing, thermally grown oxides can be used as implantation, diffusion, and etch masks. The dominance of silicon as a microelectronic material can be attributed to the existence of this high quality native oxide and the resultant near ideal silicon/oxide interface.

Figure 7.2 depicts the basic thermal oxidation process. The silicon wafer is exposed at high temperatures (typically 900 °C–1200 °C) to a gaseous oxidant such as molecular oxygen ($O_2$) and/or water vapor ($H_2O$). For obvious reasons, oxidation in $O_2$ is called dry oxidation, whereas in $H_2O$ it is called wet oxidation, as discussed in Sec. 2.1. The gas/solid interface forms a stagnant layer through which the oxidant must diffuse to reach the surface of the wafer. Once at the surface, the oxidant again must diffuse through the existing oxide layer that is present. As the oxidant species reaches the silicon/oxide interface, one of two reactions occur

$$Si + O_2 \rightarrow SiO_2 \text{ (dry oxidation)} \tag{7.3}$$

$$Si + 2H_2O \rightarrow SiO_2 + 2H_2 \text{ (wet oxidation)} \tag{7.4}$$

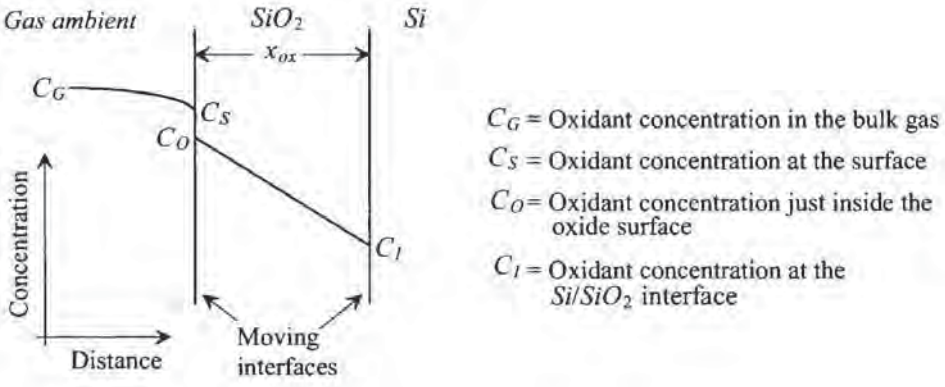



**Figure 7.2** A simple model for thermal oxidation of silicon. Notice the oxidant concentrations (boundary conditions) in the gas, oxide, and silicon.

IPR2023-00697
Theta EX2019

It should be emphasized that reactions specified by Eqs. (7.3) and (7.4) occur at the silicon/oxide interface where silicon is consumed in the reaction. As Fig. 7.3 illustrates, with respect to the original silicon surface, approximately 45% of the oxide thickness is accounted for by consumption of silicon.

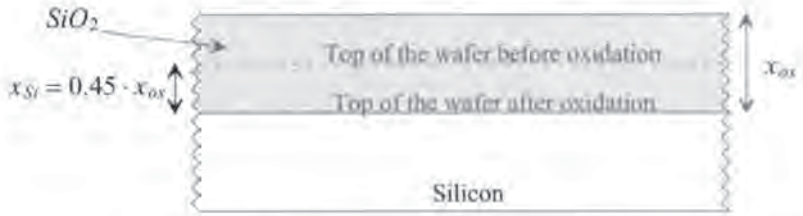



**Figure 7.3** Silicon/oxide growth interface. See also Fig. 2.4.

The rate of thermal oxidation is a function of temperature and rate constants. The rate is directly proportional to temperature. The rate constants are, in turn, a function of gas partial pressures, oxidant-type, and silicon wafer characteristics such as doping type, doping concentration, and crystallographic orientation. In general, dry oxidation yields a denser and thus higher quality oxide than does a wet oxidation. However, wet oxidation occurs at a much higher rate compared to dry oxidation. Depending on the temperature and existing thickness of oxide present, the overall oxidation rate can be either diffusion limited (e.g., thick oxides at high temperatures) or reaction rate limited (e.g., thin oxides at low temperatures). Practically, oxide thicknesses are limited to less than a few thousand angstroms and to less than a micron for dry and wet oxidation, respectively.

In a modern fabrication facility, oxidation occurs in either a tube furnace or in a rapid thermal processing (RTP) tool, as schematically shown in Fig. 7.4. The tube furnaces consist of quartz tubes surrounded by heating element coils. The wafers are loaded in the heated tubes where oxidants can be introduced through inlets. The function of the RTP is similar to the tube furnace with the exception that the thermal source is heating lamps.

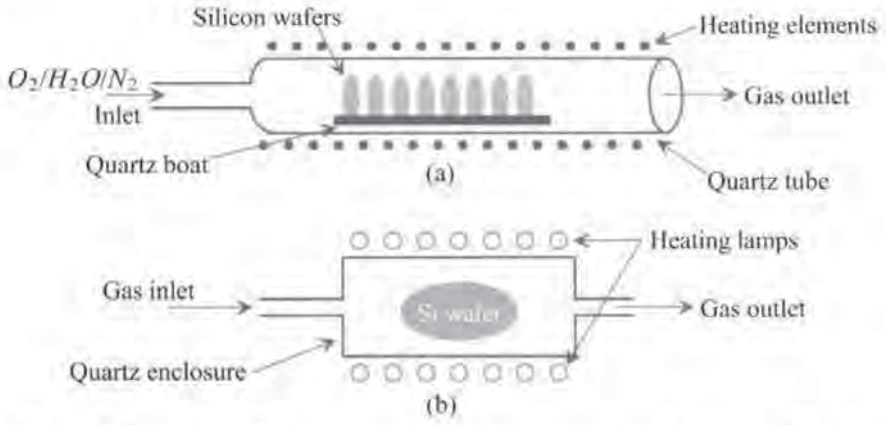



**Figure 7.4** (a) Simplified representation of an oxidation tube furnace and (b) simplified diagram for rapid thermal processing.

IPR2023-00697
Theta EX2019

### 7.1.3 Doping Processes

The controlled introduction of dopant impurities into silicon is necessary to affect majority carrier type, carrier concentration, carrier mobility, carrier lifetime, and internal electric fields. The two primary methods of dopant introduction are solid state diffusion and ion implantation. Historically, solid state diffusion has been an important doping process; however, ion implantation is the preferred method in modern CMOS fabrication.

*Ion Implantation*

The workhorse method of introducing dopants into the near-surface region of wafers is a process called ion implantation. In ion implantation, dopant atoms (or molecules) are ionized and then accelerated through a large electric potential (a few kV to MV) towards a wafer. The highly energetic ions bombard and thus implant into the surface. Obviously, this process leads to a high degree of lattice damage, which is generally repaired by annealing at high temperatures. Moreover, the ions do not necessarily come to rest at a lattice site, hence an anneal is required to electrically activate (i.e., thermally agitate the impurities into lattice sites) the dopant impurities.

Figure 7.5 shows a schematic diagram of an ion implanter. The ions are generated by an RF field in the ion source where they are subsequently extracted to a mass spectrometer. The spectrometer only allows ions with a user-selected mass to enter the accelerator, where the ions are passed through a large potential field. The ions are then scanned via electrostatic lens across the surface of the wafer.

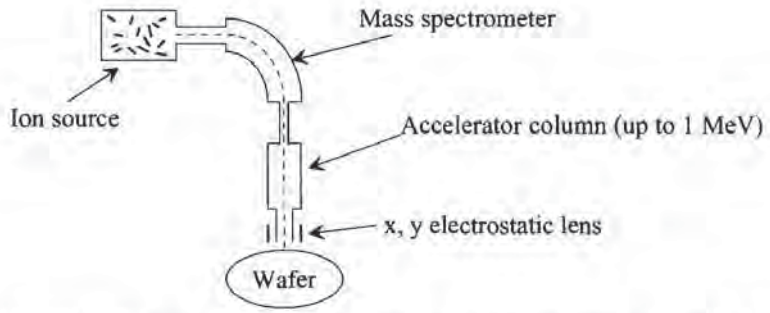



**Figure 7.5** Simplified diagram of an ion implanter. The ions are created by an RF field where they are extracted into a mass spectrometer. An electrostatic lens scans the ion beam on the surface of a wafer to achieve the appropriate dose. Electrostatically, the ions can be counted to provide the real-time dose.

A first-order model for an implant doping profile is given by a Gaussian distribution described mathematically as

$$N(x) = N_p \exp\left[-(x - R_p)^2 / 2\Delta R_p^2\right] \tag{7.5}$$

where $N_p$ is the peak concentration, $R_p$ is the projected range, and $\Delta R_p$ is called the straggle. By inspection, $R_p$ should be identified as the mean distance the ions travel into the silicon and $\Delta R_p$ as the associated standard deviation. Figure 7.6 illustrates a typical ion implant profile. Obviously, $N_p$ occurs at a depth of $R_p$. Moreover, the area under the

IPR2023-00697
Theta EX2019

implant curve corresponds to what is referred to as the implant dose $Q_{imp}$, given mathematically as

$$Q_{imp} = \int_0^\infty N(x) \cdot dx \tag{7.6}$$

Localized implantation is achieved by masking off regions of the wafer with an appropriately thick material such as oxide, silicon nitride, polysilicon, or photoresist. Since implantation occurs in the masking layer, the thickness must be of sufficient magnitude to stop the ions prior to reaching the silicon substrate. In comparison to solid state diffusion, ion implantation has the advantage of being a low temperature and highly controlled process.

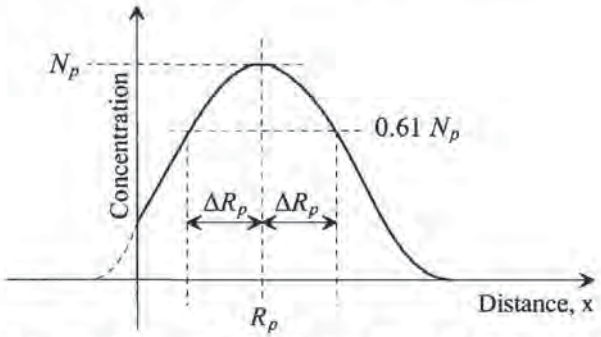



**Figure 7.6** Ideal implant profile representing Eq. (7.5). Notice that the peak concentration occurs below the surface and depends on the implant energy.

*Solid State Diffusion*

Solid state diffusion is a method for introducing and/or redistributing dopants. In this section, we study solid state diffusion primarily to gain insight into "parasitic" dopant redistribution during thermal processes. In typical CMOS process flows, dopants are introduced into localized regions via ion implantation where the subsequent processing often consists of high temperature processing. Solid state diffusion inherently occurs in these high temperature steps, thus spreading out the implant profile in three dimensions. The net effect is to shift the boundary of the implant from its original implant-defined position, both laterally and vertically. This thermal smearing of the implant profiles must be accounted for during CMOS process flow development. If not, the final device characteristics can differ significantly from what was expected.

Solid state diffusion (or simply diffusion) requires two conditions: 1) a dopant concentration gradient, and 2) thermal energy. Diffusion is directly proportional to both. An implanted profile (approximated by a delta function at the surface of the wafer) diffuses to first order as

$$N(x,t) = \frac{Q_{imp}}{\sqrt{\pi \cdot D \cdot t} \cdot \exp(-x^2/4Dt)} \tag{7.7}$$

where $Q_{imp}$ is the implant dose, $D$ is the diffusivity of the dopant, and $t$ is the diffusion time. Figure 7.7 illustrates limited-source diffusion of a one-dimensional implant profile. Notice that the areas under the respective curves for a given time are equal.

IPR2023-00697
Theta EX2019

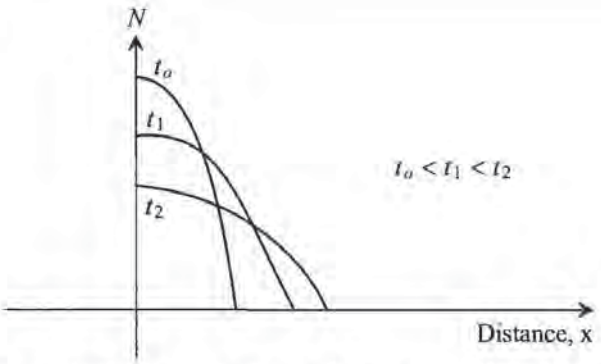



**Figure 7.7** Idealized limited-source diffusion profile showing the effects of drive-in time on the profile. Notice that the peak concentration occurs at the surface of the substrate (x = 0) and that the area under the curves is constant.

### 7.1.4 Photolithography

In the fabrication of CMOS, it is necessary to localize processing effects to form a multitude of features simultaneously on the surface of the wafer. The collection of processes that accomplish this important task using an ultraviolet light, a mask, and a light-sensitive chemical resistant polymer is called *photolithography*. Although there are many different categories of photolithography, they all share the same basic processing steps that result in micron-to-submicron features generated in the light-sensitive polymer called photoresist. The photoresist patterns can then serve as ion implantation masks and etch masks during subsequent processing steps.

Figure 7.8 outlines the major steps required to implement photolithography patterning of a thermally grown oxide. Photoresist, a viscous liquid polymer, is applied to the top surface of the oxidized wafer. The application typically occurs by dropping (or spraying) a small volume of photoresist to a rapidly rotating wafer yielding a uniform thin film on the surface. Following spinning, the coated wafer is softbaked on a hot plate which drives out most of the solvents from the photoresist and improves the adhesion to the underlying substrate. Next, the wafers are exposed to ultraviolet light through a mask (or reticle) that contains the layout patterns for a given drawn layer. Unless the first layer is being printed, the exposure must be preceded by a careful alignment of mask features to existing patterns on the wafer. There are three general methods of exposing (patterning) the photoresist: contact, proximity, and projection photolithography. In both contact and proximity photolithography, the mask and the wafers are in contact and in close proximity, respectively, to the surface of the photoresist. Here the mask features are of the same scale as the features to be exposed on the surface.

In projection photolithography, the dominant type of patterning technology, the mask features are on a larger scale (e.g., 5X or 10X) relative to the features exposed on the surface. This is accomplished with a projection *stepper*. Using reduction with optics, the stepper projects an image through the mask to the photoresist on the surface. For positive-tone photoresist, the ultraviolet light breaks molecular bonds, making the exposed regions more soluble in the developer. In contrast, for negative-tone photoresist, the exposure causes polymerization and thus less solubility. The exposed resist-coated wafer is developed in an alkaline solution. Depending on the formulation of the photoresist, a positive or negative image relative to the mask patterns can be generated.

IPR2023-00697
Theta EX2019

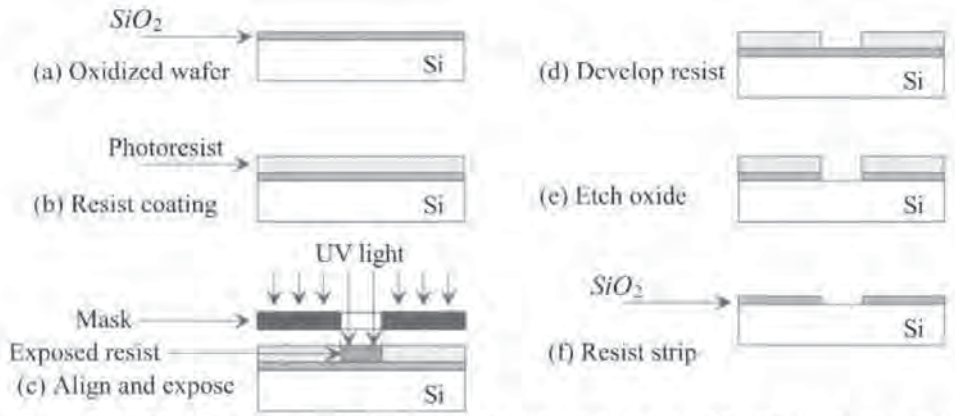



**Figure 7.8** Simplified representation of the primary steps required for the implementation of photolighography and pattern transfer.

To harden the photoresist for improved etch-resistance and to improve adhesion, the newly developed wafers are often hardbaked. At this point, the wafer can be etched to transfer the photoresist pattern into the underlying oxide film.

*Resolution*

In general, there are three critical parameters associated with a given projection stepper: resolution, depth of focus, and pattern registration. The diffraction of light caused by the various interfaces in its path limits the minimum printable feature size as depicted in Fig. 7.9. Resolution is defined as the minimum feature size, *M*, that can be printed on the surface of the wafer given by

$$M = \frac{c_1 \cdot \lambda}{NA} \tag{7.8}$$

where lambda, $\lambda$, is the wavelength of the ultraviolet light source, *NA* is the numerical aperture of the projection lens, and $c_1$ is a constant whose value ranges from 0.5 to 1. The *NA* of a lens is illustrated in Figure 7.10 and is given mathematically as

$$NA = n \cdot \sin\theta \tag{7.9}$$

where $n$ is the index of refraction of the space between the wafer and the lens and $\theta$ is the acute angle between the focal point on the surface of the wafer and the edge of the lens radii. Notice that M is directly proportional to wavelength, hence diffraction effects are the primary limitation in printable feature size. To a limit, the *NA* of the projection lens can be increased to help combat the diffraction effects because large *NA* optics have an increased ability to capture diffracted light. At first glance, the minimum feature size cannot be less than the wavelength of the light, however, advance techniques such as optical proximity correction (OPC) and wavefront engineering of the photomasks have been developed to push the resolution limits below the wavelength.

*Depth of Focus (DOF)*

The depth of focus (DOF) of the projection optics limits one's ability to pattern features at different heights, as illustrated in Fig. 7.11. Mathematically, DOF is given by

$$\text{DOF} = \frac{c_2 \lambda}{NA^2} \tag{7.10}$$

IPR2023-00697
Theta EX2019

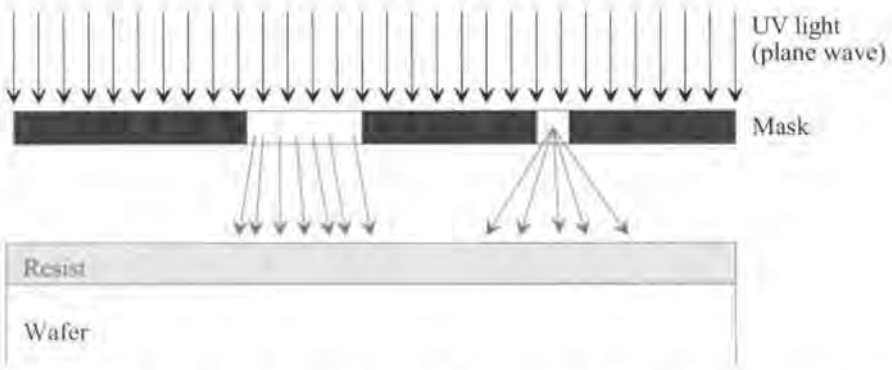



**Figure 7.9** The diffraction effects become significant as the mask feature dimensions approach the wavelength of UV light. Notice that the diffraction angle is larger for the smaller opening.

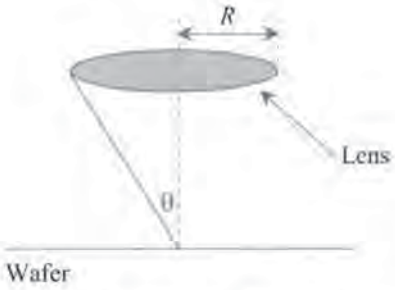



**Figure 7.10** The relationship of the lens radii to the angle used to compute NA.

where $c_2$ is a constant ranging in value from 0.5 to 1. As apparent from Eq. (7.8) and Eq. (7.10), there exists a fundamental trade-off between minimum feature size and DOF. In other words, to print the smallest possible features, the surface topography must be minimized.

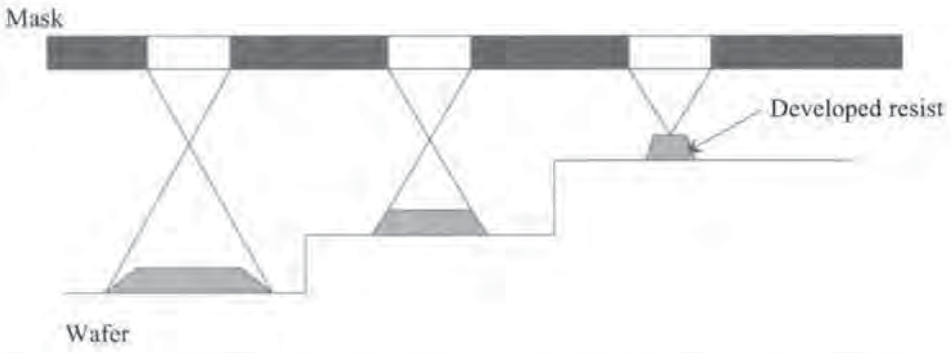
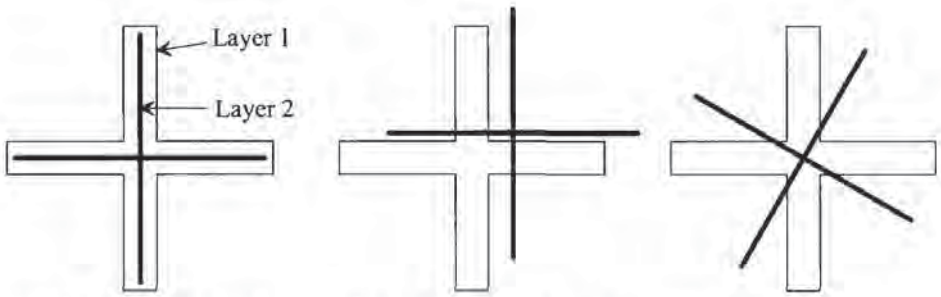



**Figure 7.11** Depth of focus diagram illustrating the need to have planar surfaces (minimized topography) during high resolution photopatterning.

IPR2023-00697
Theta EX2019

*Aligning Masks*

During CMOS fabrication, numerous mask levels (e.g., active, poly, contacts, etc.) are printed on the wafer. Each of these levels must be accurately aligned to one another. Registration is a measure of the level-to-level alignment error. Registration errors occur in x, y, and z-rotations, as illustrated in Fig. 7.12.




(a) No registration error (b) x-y registration error (c) z-rotation registration error.

**Figure 7.12** Simple registration errors that can occur during wafer-to-mask alignment in photolithography. Other registration errors exist but are not discussed here.

### 7.1.5 Thin Film Removal

Typically one of two processes are performed following photolithography. One is thin film etching used to transfer the photoresist patterns to the underlying thin film(s). The other is ion implantation using the photoresist patterns to block the dopants from select regions on the surface of the wafer. In this section, we discuss thin-film etching processes based on wet chemical etching and dry etching techniques. Additionally, we discuss a process used to remove unpatterned thin-films called chemical mechanical polishing (CMP).

*Thin Film Etching*

Once a photoresist pattern is generated there are two commonly employed approaches, wet etching and dry etching, to transfer patterns into underlying films. Etch rate (thickness removed per unit time), selectivity, and degree of anisotropy are key parameters for both wet and dry etching. Etch rates are typically strong functions of solution concentration and temperature. Selectivity, $S$, is defined as the etch rate ratio of one material to another given by the selectivity equation

$$S = \frac{R_2}{R_1} \tag{7.11}$$

where $R_2$ is the etch rate of the material intended to be removed and $R_1$ is the etch rate of the underlying, masking, or adjacent material not intended to be removed. The degree of anisotropy, $A_f$, is a measure of how rapidly an etchant removes material in different directions, mathematically given by

$$A_f = 1 - \frac{R_l}{R_v} \tag{7.12}$$

IPR2023-00697
Theta EX2019

where $R_l$ is the lateral etch rate and $R_v$ is the vertical etch rate. Notice that if $A_f = 1$, then the etchant is completely anisotropic. However, if $A_f = 0$, then the etchant is completely isotropic. In conjunction with photolithography, the degree of anisotropy is a major factor in the achievable resolution. Figure 7.13 illustrates the effects of etch bias (i.e., $d_{film} - d_{mask}$) on the final feature size. For the submicron features that are required in CMOS, dry etch techniques are preferred over wet etch processes. This is due to the fact that dry etch techniques can, in general, have a higher degree of anisotropy. Both wet and dry etching are applied to the removal of metals, semiconductors, and insulators.

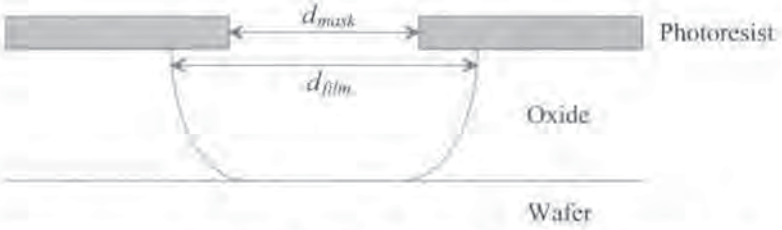



**Figure 7.13** Diagram showing a post-etch profile. Notice that because of isotropy in the etch process the mask opening does not match the fabricated opening in the underlying oxide film. The difference between these dimensions is called etch-bias.

*Wet Etching*

Wet etching consists of using a chemical solution to remove material. In CMOS fabrication, wet processes are used both for cleaning of wafers and for thin-film removal. Wet cleaning processes are repeated numerous times throughout a process flow. Some cleaning processes are targeted to particulate removal, while others are for organic and/or inorganic surface contaminants. Wet etchants can be isotropic (i.e., etch rate is the same in all directions) or anisotropic (i.e., etch rate differs in different directions) although most of the wet etchants used in CMOS fabrication are isotropic. In general, wet etchants tend to be highly selective compared to dry etch processes. A schematic diagram of a wet etch tank is shown in Fig. 7.14. To improve the etch uniformity and to aid particulate removal, it is common to ultrasonically vibrate the etchant, as shown in the figure. Furthermore, microcontrollers accurately control the temperature of the bath. Once the etch is completed, the wafers are rinsed in deionized (DI) water, then spun dried.

*Dry Etching*

In CMOS fabrication, there are three general categories of dry etch techniques: sputter etching, plasma etching, and reactive ion etching (RIE). Figure 7.15 schematically illustrates a sputter etch process. An inert gas (e.g., argon) is ionized where the ions are accelerated through an electric field established between two conductive electrodes, called the anode and the cathode. A vacuum in the range of millitorr must exist between the plates to allow the appropriate ionization and transfer of ions. Under these conditions, a glow discharge, or plasma, is formed between the electrodes. In simple terms, the plasma consists of positively charged ions and electrons, which respond oppositely to the electric field. The wafer sits on the cathode where it is bombarded by the positively charged ions, causing material to be ejected off of the surface. Essentially, sputter etching

IPR2023-00697
Theta EX2019

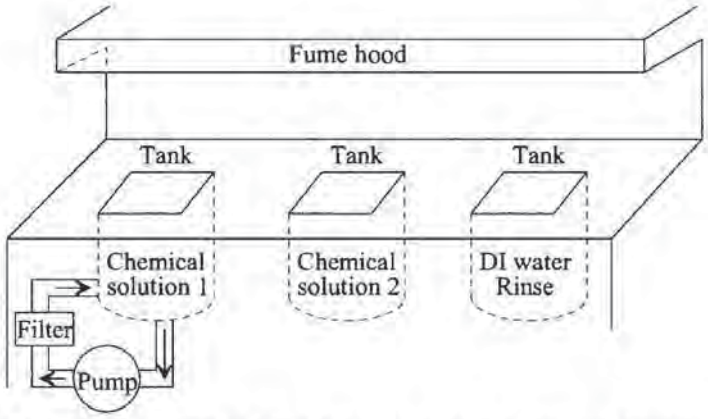



**Figure 7.14** Simplfied diagram of a wet bench used for wet chemical cleaning and etching

is atomic-scale sandblasting. A DC power supply can be used for sputter etching conductive substrates, while an RF supply is required through capacitive coupling for etching non-conductive substrates. Sputter etching tends not to be selective, but it is very anisotropic.

A simplified diagram of a plasma etch system is shown in Figure 7.16. A gas or mixture of gases (e.g., halogens) are ionized, producing reactive species called radicals. A glow discharge or plasma is formed between the electrodes. The radicals chemically react with the surface material forming reaction products in the gas phase which are pumped away through a vacuum system. Plasma etching can be very selective, but is typically highly isotropic.

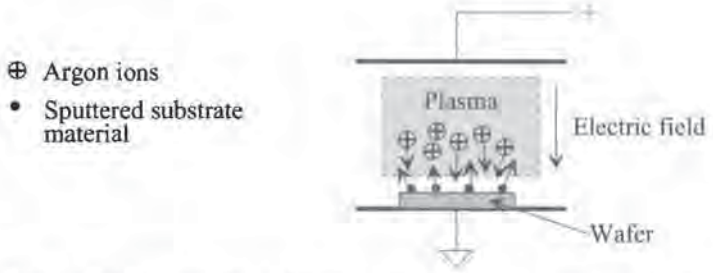



**Figure 7.15** Simplified schematic diagram of the sputter etch process. This process is dominated by the physical bombardment of ions on a substrate.

While sputter etching is a purely mechanical process and plasma etching is purely chemical, RIE is a combination of sputter etching and plasma etching, as schematically shown in Figure 7.17. In RIE, a gas or mixture of gases (e.g., fluorocarbons) are ionized where radicals and ionized species are generated, both of which interact with the surface of the wafer. RIE is the dominant etch process because it can provide the benefits of both sputter etching and plasma etching. In other words, RIE can be highly selective and highly anisotropic.

IPR2023-00697
Theta EX2019

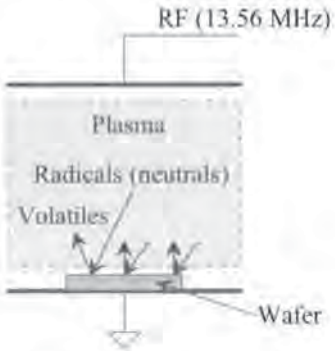



**Figure 7.16** Simplified schematic diagram of a plasma etch process. This process is dominated by the chemical reactions of radicals at the surface of the substrate.

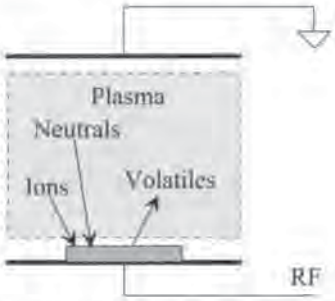



**Figure 7.17** Simplified schematic diagram of an RIE etch process. This process has both physical (ion bombardment) and chemical (reaction of radicals) components.

*Chemical Mechanical Polishing*

Figure 7.18 depicts the key features of chemical mechanical polishing (CMP). In CMP, an abrasive chemical solution, called a slurry, is introduced between a polishing pad and the wafer. Material on the surface of the wafer is removed by both a mechanical polishing component and a chemical reaction component. In modern CMOS fabrication, CMP is a critical process that is used to planarize the surface of the wafer prior to photolithography. The planar surface allows the printed feature size to be decreased. CMP can be used to remove metals, semiconductors, and insulators.

### 7.1.6 Thin Film Deposition

Insulators, conductors, and semiconductors are all required for CMOS integrated circuits. Semiconductors, such as crystalline silicon for the active areas and polycrystalline silicon for the gate electrodes/local area interconnects, are generally required. Insulators such as $Si_3N_4$, $SiO_2$ and doped glasses are used for gate dielectrics, device isolation, metal-to-substrate isolation, metal-to-metal isolation, passivation, etch masks, implantation masks, diffusion barriers, and sidewall spacers. Conductors such as aluminum, copper, cobalt, titanium, tungsten, and titanium nitride are used for local interconnects, contacts, vias, diffusion barriers, global interconnects, and bond pads. In this section, we discuss the various methods to deposit thin films of insulators, conductors, and semiconductors. We

IPR2023-00697
Theta EX2019

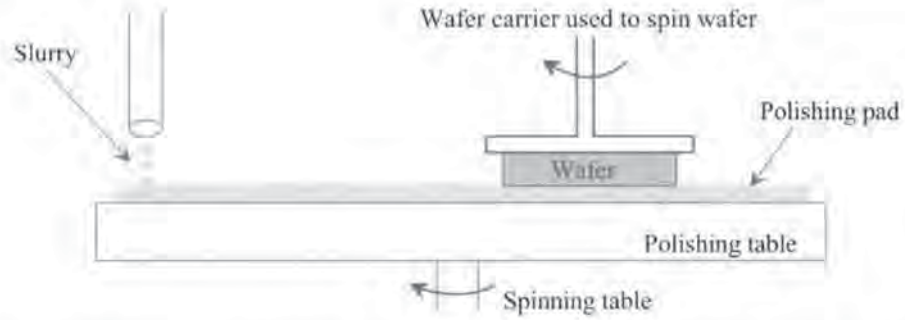



**Figure 7.18** Simplified representation of a chemical mechanical polishing process used in the fabrication process.

present two primary categories of thin film deposition: physical vapor deposition and chemical vapor deposition. A third less common category, electrodeposition, for depositing copper for backend interconnects will not be addressed here.

Deposited films are often characterized by several factors. Inherent film quality related to the compositional control, low contamination levels, low defect density, and predictable and stable electrical and mechanical properties are of prime importance. Moreover, film thickness uniformity must be understood and controlled to high levels. To achieve highly uniform CMOS parameters across a wafer, it is common to control the film thickness uniformity to less then +/– 5 nm across the wafer diameter. In addition, film uniformity over topographical features is of critical importance. A measure of this is called step coverage, as depicted in Fig. 7.19. As illustrated, good step coverage results in uniform thickness over all surfaces. In contrast, poor step coverage results in significantly reduced thickness on vertical surfaces relative to surfaces parallel with the surface of the wafer. Related to step coverage is what is referred to as gap fill. Gap fill applies to the deposition of a material into a high aspect ratio opening, such as contacts or gaps between adjacent metal lines. Figure 7.20 illustrates a deposition with good gap fill and a deposition that yields a poor gap fill (also called a keyhole or void).




(a) Good step coverage

(b) Poor step coverage

**Figure 7.19** Extremes in thin-film deposition coverage over a pre-existing oxide step.

IPR2023-00697
Theta EX2019

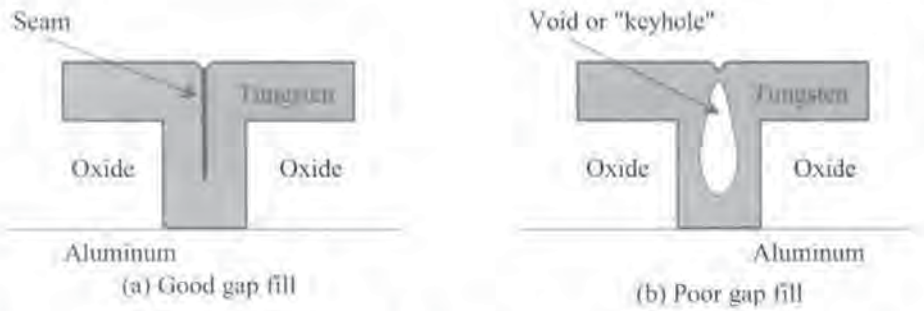



**Figure 7.20** Gap-fill profiles (good and bad) of a high aspect ratio opening filled with a deposited film.

*Physical Vapor Deposition (PVD)*

In physical vapor deposition (PVD), physical processes produce the constituent atoms (or molecules), which pass through a low-pressure gas phase and subsequently condense on the surface of the substrate. The common PVD processes are evaporation and sputter deposition, both of which can be used to deposit a wide range of insulating, conductive, and semiconductive materials. One of the drawbacks of PVD is that the resultant films often have poor step coverage.

Evaporation is one of the oldest methods of depositing thin-films of metals, insulators, and semiconductors. The basic process of evaporation is shown in Fig. 7.21. The material to be deposited is heated past its melting point in a high vacuum chamber where the vapor form of the material coats all exposed surfaces within the mean free path of the evaporant. The heat source can be of one of two types: heating filament or focused electron beam.

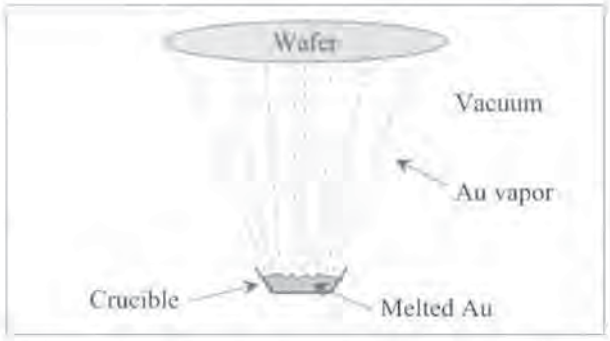



**Figure 7.21** Simplified diagram of an evaporation deposition process.

In simple terms, sputter deposition is similar to sputter etching, as discussed in Scc. 7.1.5, with the exception that the wafer serves as the anode, and the cathode is the target material to be deposited. Figure 7.22 outlines a simplified sputter deposition process. An inert gas such as argon is ionized in a low pressure ambient where the positively charged ions are accelerated through the electric field towards the target. The

IPR2023-00697
Theta EX2019

target is an ultra-high purity disk of material to be deposited. The bombardment of ions with the target sputter (or eject) target atoms (or molecules), where they transit to the surface of the wafer forming a thin-film. Similar to sputter etching, a DC supply can be used for sputtering conductors; however, a capacitively coupled RF supply must be used for depositing non-conductive materials.

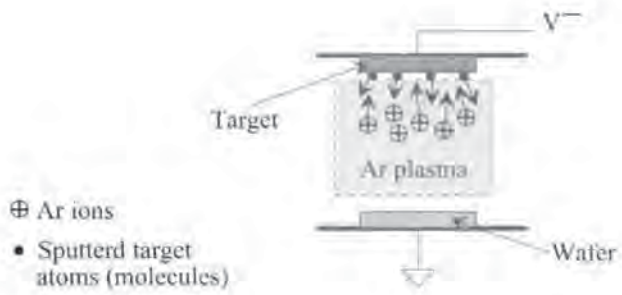



**Figure 7.22** Simplified diagram of a sputter deposition process.

*Chemical Vapor Deposition (CVD)*

In chemical vapor deposition (CVD), reactant gases are introduced into a chamber where chemical reactions between the gases at the surface of the substrate produce a desired film. The common CVD processes are atmospheric pressure (APCVD), low pressure (LPCVD), and plasma enhanced (PECVD). Again, there are a wide variety of insulators, conductors, and semiconductors that can be deposited by CVD. Most importantly, the resultant films tend to have good step coverage compared to PVD processes. APCVD occurs in an apparatus similar to an oxidation tube furnace (see Fig 7.4); however, an appropriate reactive gas is flowed over the wafers. APCVD is performed at relatively low temperatures. As depicted in Fig. 7.23, LPCVD occurs in a reactor in the pressure range of milliTorr to a few Torr. Compared to APCVD, the low pressure process can yield

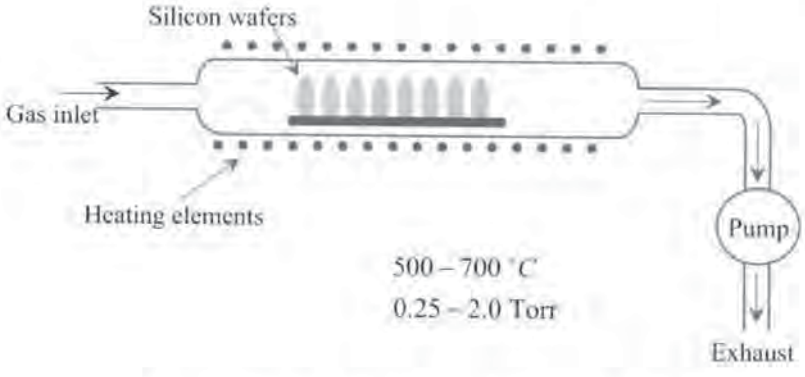



**Figure 7.23** Simplified schematic diagram of a LPCVD.

IPR2023-00697
Theta EX2019

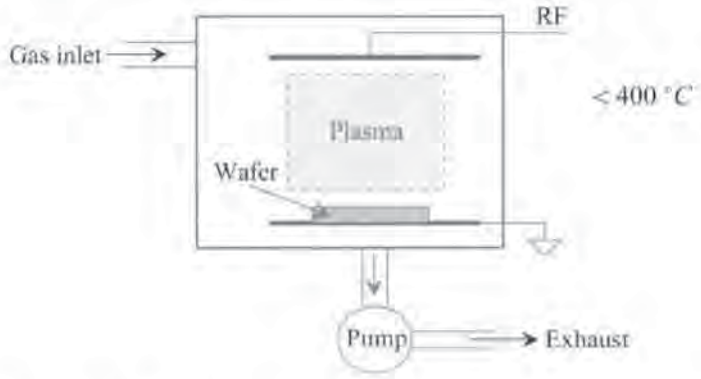



**Figure 7.24** Simplified schematic diagram of a PECVD reactor.

highly conformal films, but at the expense of a higher deposition temperature. Fig. 7.24 shows a schematic diagram of a PECVD reactor. In PECVD, a plasma imparts energy for the surface reactions, allowing for lower temperature deposition. By comparison, PECVD has the advantage of being low temperature and highly conformal.

## 7.2 CMOS Process Integration

Process integration is the task of combining a deliberate sequence of unit processes to fabricate integrated microelectronic circuits (e.g., MOSFETs, resistors, capacitors, etc.) A typical CMOS technology consists of a complex arrangement of unit processes where several hundred steps are required to manufacture integrated circuits on a silicon wafer. Groups of unit processes are combined to form integration modules. For example, the gate module would include a specific sequence of unit processes for yielding a gate electrode on a thin, gate dielectric. The modules could then be combined to yield the overall process flow (or process sequence). The process flow can be divided into frontend-of-the-line (FEOL) and backend-of-the-line (BEOL) processes. A typical process flow consisting of numerous modules is shown in block diagram form in Fig. 7.25.

*FEOL*

Generally, FEOL refers to all processes preceding salicidation (i.e., silicide formation, see Fig. 4.4). FEOL includes all processes required to fully form isolated CMOS transistors. In Fig. 7.25 we see that the FEOL begins with the selection of the starting material (i.e., type of silicon wafer to be used). Then, the shallow trench isolation (STI) module is implemented to form the regions of dielectric between regions of active area. Next, the wells (or tubs) are formed followed by the gate module, which includes all processes to properly define gate electrodes on a thin oxide. Finally, the FEOL concludes with the source/drain module, which includes the processes required for the formation of the low-doped drain extensions and the source/drain regions themselves.

*BEOL*

BEOL refers to all processes subsequent to source/drain formation. Hence, BEOL processes are used to "wire" the transistors together using multiple layers of dielectrics

IPR2023-00697
Theta EX2019

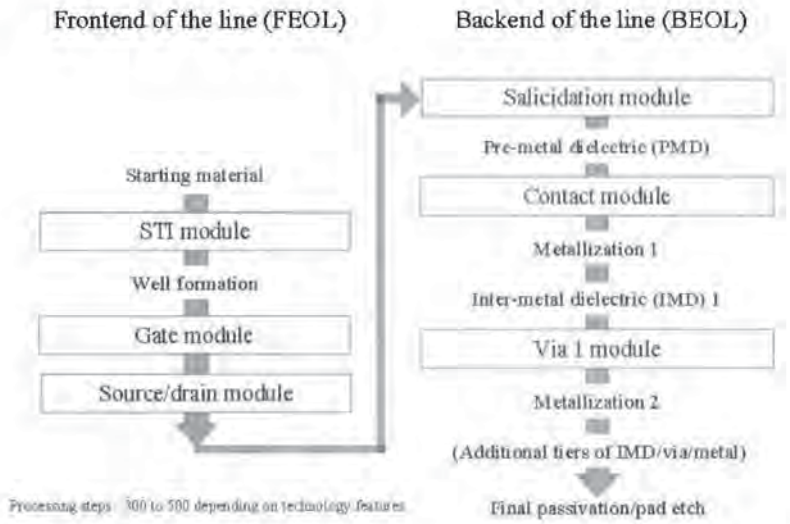



**Figure 7.25** A typical CMOS process flow illustrating the difference between FEOL and

and metals. The BEOL begins with the salicidation of the polysilicon and source/drain regions. The remaining BEOL processes proceed in repetitive sets of modules to yield lateral and vertical interconnects isolated from one another with dielectrics. It is important to understand that there is a high degree of inter-relationship between unit processes within each module and between modules themselves. A seemingly "trivial" change in one unit process in a given module can have dramatic effects on processes in other modules. In other words, **there is no trivial process change**.

*CMOS Process Description*

Even with a single device type, there are numerous variations in process schemes for achieving similar structures. Hence, it would be virtually impossible to outline all schemes. Therefore, we will describe a generic (but representative) deep-submicron CMOS process flow (deep indicating that a deep, or short wavelength, ultraviolet light source is used when patterning the wafers). Our CMOS technology will have the following features:

1. Frontend of the line (FEOL)
    (a) Shallow trench isolation (STI)
    (b) Twin-tubs
    (c) Single-level polysilicon
    (d) Low-doped drain extensions
2. Backend of the line (BEOL)
    (a) Fully planarized dielectrics
    (b) Planarized tungsten contacts and via plugs
    (c) Aluminum metallization

IPR2023-00697
Theta EX2019

Following each major process step, cross sections will be shown. The cross sections were generated with a technology computer-assisted design (TCAD) package called Tsuprem-4 and Taurus-Visual (2D) released by Technology Modeling Associates, Inc. These tools simulate a defined sequence of unit processes, thus allowing one to model a process flow prior to its actual implementation. In the interest of brevity, there are several omissions and consolidations in the process description. These include:

1. Wafer cleaning performed immediately prior to all thermal processes, metal depositions, and photoresist removals.
2. Individual photolithographic process steps such as dehydration bake, wafer priming, photoresist application, softbake, alignment, exposure, photoresist development, hardbake, inspection, registration measurement, and critical dimension (CD) measurement.
3. Backside film removal following select CVD processes.
4. Metrology to measure particle levels, film thickness, and post-etch CDs.

To implement our CMOS technology, we employ a reticle set as outlined in Table 7.1. The masks are labeled as having either a clear or a dark field. Clear field masks are masks with opaque features totaling less than 50% of the mask area. In contrast, dark field refers to a mask that has opaque features that account for greater than 50% of the total area. Using this mask set, we assume the exclusive use of positive tone photoresist processing. When appropriate, representative mask features that yield isolated, complementary transistors adjacent to one another are shown.

**Table 7.1** Masks used in our generic CMOS process.

| Layer name | Mask | Aligns to level | Times used | Purpose |
|---|---|---|---|---|
| 1 (active) | Clear | aligns to notch | 1 | Defines active areas |
| 2 (p-well) | Clear | 1 | 2 | Defines NMOS sidewall implants and p-well |
| 3 (n-well) | Dark | 1 | 2 | Defines PMOS sidewall implants and n-well |
| 4 (poly1) | Clear | 1 | 1 | Defines polysilicon |
| 5 (n-select) | Dark | 1 | 2 | Defines nLDD and n+ |
| 6 (p-select) | Dark | 1 | 2 | Defines pLDD and P+ |
| 7 (contact) | Dark | 4 | 1 | Defines contact to poly and actives areas |
| 8 (metal1) | Clear | 7 | 1 | Defines metal1 |
| 9 (via1) | Dark | 8 | 1 | Defines via1 (connects M1 to M2) |
| 10 (metal2) | Clear | 9 | 1 | Defines metal2 |
| passivation | Dark | Top-level metal | 1 | Defines bond pad opening in passivation |

IPR2023-00697
Theta EX2019

### 7.2.1 Frontend-of-the-Line Integration

As previously stated, FEOL encompasses all processing required to fabricate the fully-formed, isolated CMOS transistors. In this subsection we discuss the modules and unit processes required for a representative CMOS process flow.

*Starting Material*

The choice of substrate is strongly influenced by the application and characteristics of the CMOS ICs to be fabricated. Bulk silicon is the least costly but may not be the optimal choice in high performance or harsh environment CMOS applications. Epitaxial (Epi) wafers are heavily doped bulk wafers with a thin, moderately to lightly doped epitaxial silicon layer grown on the surface. The primary advantage of Epi wafers is for immunity to latch-up. Silicon-on-insulator wafers increase performance and eliminate latch-up. However, SOI CMOS is more costly to implement than bulk or epi technologies. The three general types of silicon substrates are shown in Fig. 7.26. In current manufacturing, wafer diameters typically range from 100 to 300 mm. The wafer thickness correspondingly increases with diameter to allow for greater rigidity. The actual CMOS is constructed in the top one micron or less of the wafer, whereas the remaining hundreds of microns are used solely for mechanical support during device fabrication.

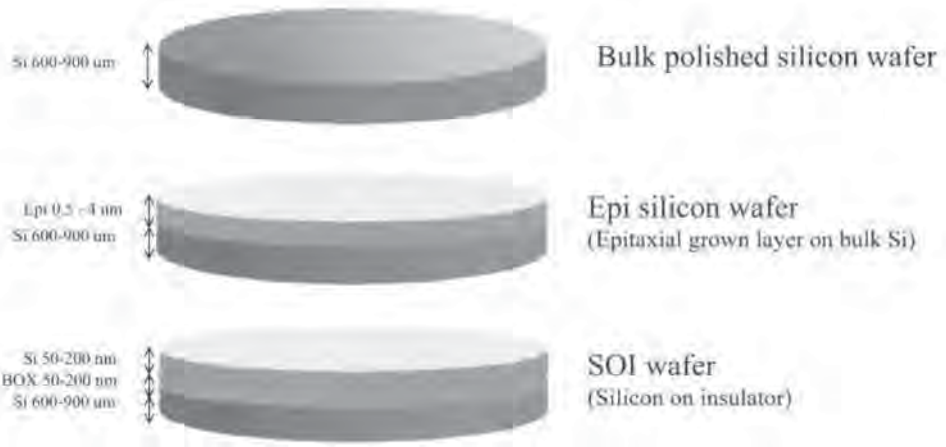



**Figure 7.26** The three general types of silicon wafers used for CMOS fabrication.

We use bulk silicon wafers for our CMOS technology in this section. It should be noted that with relatively minor process and integration adjustments our technology could be applicable to epi or SOI CMOS processes. The first of many simulated cross sections are shown in Fig. 7.27. Here only the top two microns of silicon are shown. At the beginning of the fabrication process, the wafer characteristics such as resistivity, sheet resistance, crystallographic orientation, and bow and warp are measured and/or recorded. Furthermore, the wafers are scribed, usually with a laser, with a number which identifies the wafer's lot and number.

IPR2023-00697
Theta EX2019



**Figure 7.27** Simulated cross-sectional view (the top 2 μm) of the bulk wafer in Fig. 7.26.

*Shallow Trench Isolation Module*

Devices (e.g., PMOS and NMOS) must be electrically isolated from one another. This isolation is of primary importance for suppressing leakage current between both like and dissimilar devices.

One of the simplest methods of isolation is to fabricate the CMOS such that a reversed-bias pn-junction is formed between the transistors. Oppositely doped regions (e.g., n-well adjacent to a p-well) can be electrically isolated by tying the n-region to the most positive potential in the circuit and the p-region to the most negative. As long as the reverse-bias is maintained and the breakdown voltage is not exceeded for all operating conditions, a small diode reverse saturation current accounts for the leakage current. This junction leakage current is directly proportional to the junction area; hence, for the large p- and n-regions in modern devices, junction isolation alone is not adequate.

The second general method of isolation is related to the formation of thick dielectric regions, called field regions, between transistors. The region without the thick dielectric is where the transistors reside and is known as the active area. The relatively thick oxide that forms between the active areas is called field oxide (FOX). Interconnections of polysilicon are formed over the field regions to provide localized electrical continuity between transistors. This arrangement inherently leads to the formation of parasitic field effect transistors. Effectively, the FOX increases the parasitic transistor's threshold voltage such that the device always remains in the off state. Further, this threshold voltage can be increased by increasing the surface doping concentration, called channel stops, under the FOX. There are two general approaches to forming field oxide regions: LOCOS and STI.

LOCal Oxidation of Silicon (LOCOS) has been used extensively for half-micron or larger minimum linewidth CMOS technologies. In LOCOS, a diffusion barrier of silicon nitride blocks the thermal oxidation of specific regions on the surface of a wafer. Both oxygen and water diffuse slowly through silicon nitride. Hence, nitride can be deposited and patterned to define active and field oxide areas. The primary limitation to LOCOS is bird's beak encroachment, where the lateral diffusion of the oxidant forms an oxide feature that in cross-section resembles a bird's beak. The bird's beak encroaches into the active area, thereby reducing the achievable circuit packing density. Moreover, LOCOS requires a long, high-temperature process, which can result in significant diffusion of previously introduced dopants.

Shallow trench isolation (STI) is the dominant isolation technology for sub-half micron CMOS technologies. As the name implies, a shallow trench is etched into the surface of the wafer and then filled with a dielectric serving as the FOX. A typical STI process sequence follows. From a processing perspective, STI is complex; however, it

IPR2023-00697
Theta EX2019

can be implemented with minimal active-area encroachment. Moreover, it has a relatively low thermal budget.

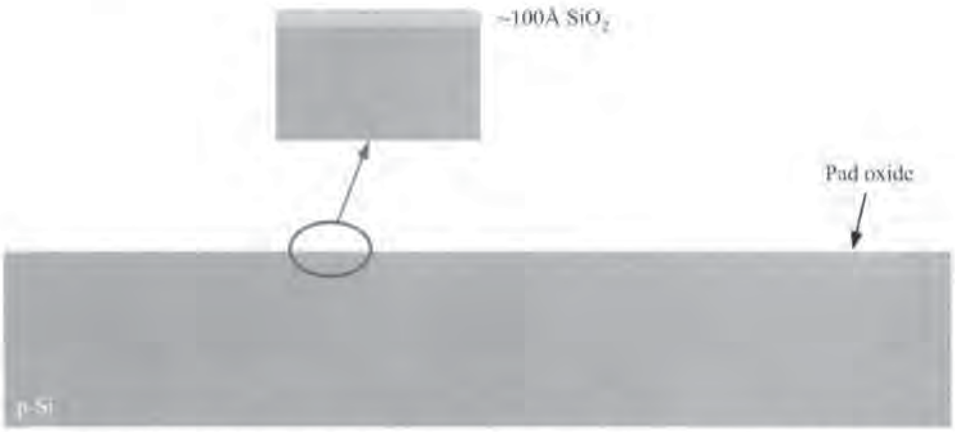



**Figure 7.28** STI film stack. The oxide is thermally grown at approximately 900 °C with dry $O_2$.

The CMOS technology outlined in this chapter uses STI to achieve device isolation. The STI module begins with the thermal oxidation of the wafer surface, as shown in Fig. 7.28. The resultant oxide serves as a film-stress buffer, called a pad oxide, between the silicon and the subsequently deposited $Si_3N_4$ layer. (In addition, it is also used following the post-CMP nitride strip as an ion implant sacrificial oxide.) Next, silicon nitride is deposited on the oxidized wafer by LPCVD, as shown in Fig. 7.29. Later, this nitride serves as both an implant mask and a CMP stop-layer. As shown in Fig. 7.30, the photolithography (mask layer 1) produces the appropriate patterns in photoresist for defining the active areas. Then, with end-point-detected RIE, the photoresist pattern is transferred into the underlying film stack of nitride and oxide. In Fig. 7.30 notice that the PMOS and NMOS devices will be fabricated on the left side and right side, respectively, under the photoresist. The region cleared of photoresist corresponds to the isolation regions. The 0.4 μm deep silicon trenches are formed by timed RIE with the photoresist




**Figure 7.29** STI film stack. Silicon nitride deposited by LPCVD at approximately 800 °C.

IPR2023-00697
Theta EX2019

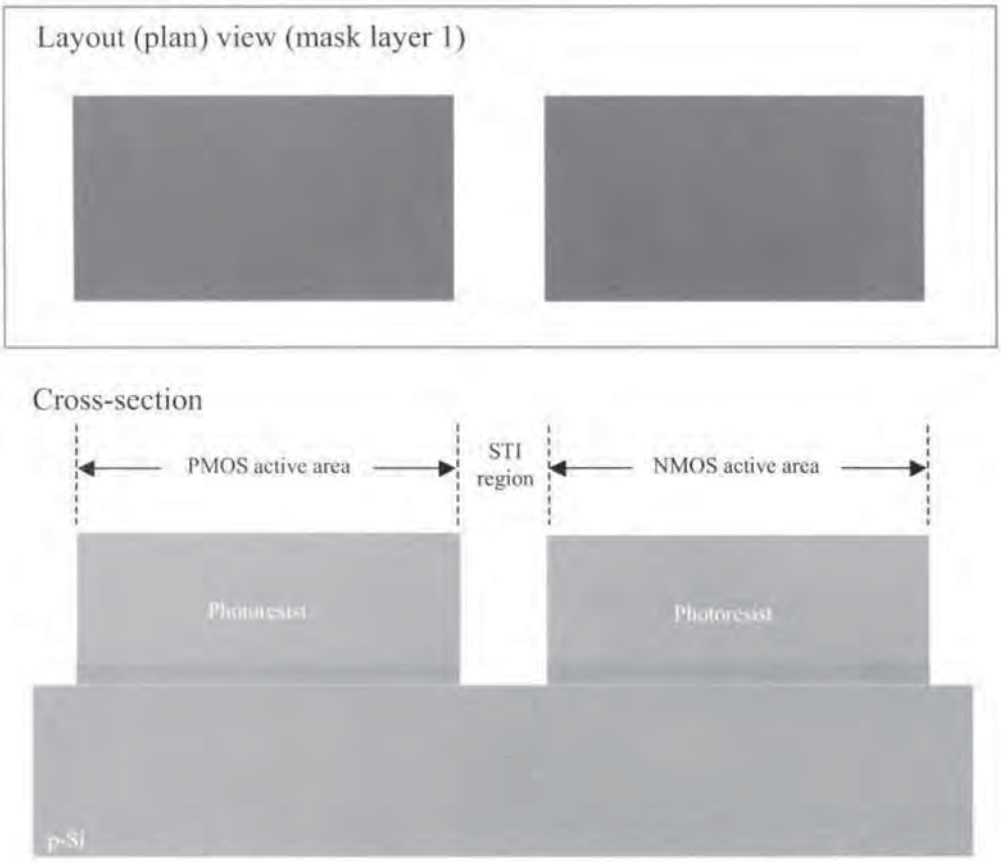



**Figure 7.30** STI definition, photolithography and nitride/pad oxide etch with fluorocarbon- based RIE.

softmask present, as shown in Fig 7.31. Although the etching can proceed without the resist, the sidewall profile can be tailored to a specific slope with the presence of the polymer during the etch process. At the cost of reduced packing density, the sloped sidewalls aid in the reduction of leakage current from the parasitic corner transistors.




**Figure 7.31** Timed silicon trench reactive ion etch.

IPR2023-00697
Theta EX2019

Following the silicon etch, $O_2$ plasma and wet processing strip the photoresist and etch by-products from the surface of the wafer. At this point, the general structural form of the STI is finished.

The next series of processes are used to improve effectiveness of the STI to suppress leakage currents. As shown in the expanded view of the trench seen in Fig. 7.32, a thin, sacrificial oxide is thermally grown on the exposed silicon. It should be noted that the nitride provides a barrier to the diffusion of oxygen, hence the oxidation occurs only in the exposed silicon regions. This oxide serves as a sacrificial oxide for subsequent ion implantation and aids in softening the corner of the trench. In general, implant sacrificial oxides are used to (1) suppress ion channeling in the crystal lattice, (2) minimize lattice damage from the ion bombardment, and (3) protect the silicon surface from contamination. Photolithography (mask layer 2) patterns resist to protect the PMOS sides of the trench during the p-wall implant. A shallow $BF_2$ implant is performed to dope what will eventually become the p-*well* trench sidewalls (called the p-*walls*), as seen in Fig. 7.33. The p-wall implant increases the threshold voltage of the parasitic corner transistor and minimizes leakage under the trench. The $BF_2$ implanted resist is stripped using $O_2$ plasma and wet processing, as shown in Fig. 7.34. Again, photolithography (mask layer 3), Fig. 7.35, is used to produce the complementary pattern for the n-wall implant. For the same reasons, this shallow phosphorous implant is introduced into what will become the n-well trench sidewalls. The phosphorous-implanted resist is stripped using $O_2$ plasma and wet processing, yielding the structure depicted in Fig. 7.36.

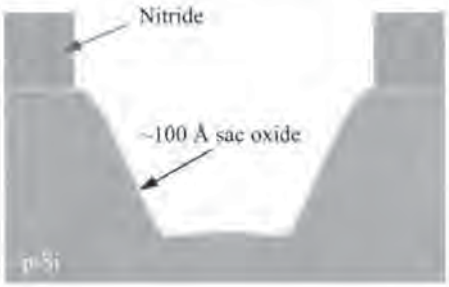



**Figure 7.32** Cross-section showing post STI resist strip followed by the dry thermal oxidation (at 900 °C) of a sacrificial (sac) oxide in the trench.

At this point, the sacrificial oxide has been degraded by the implantations and is likewise stripped using a buffered hydrofluoric acid solution. A thin, high-quality thermal oxide is regrown in the trenches to form what is called a *trench liner*. In general, the liner oxide improves the interface quality between the silicon and the subsequent trench fill, thus suppressing the interface leakage current. Specifically, the formation of the trench liner oxide (1) "cleans" the surface prior to trench fill, (2) anneals sidewall implant damage, and (3) passivates interface states to minimize parasitic leakage paths.

Once the liner oxide is grown, CVD is used to overfill the trenches with a dielectric, as shown in Fig. 7.37. The trench fill provides the field isolation required to increase the threshold voltage of the parasitic field transistors. Further, it blocks

IPR2023-00697
Theta EX2019

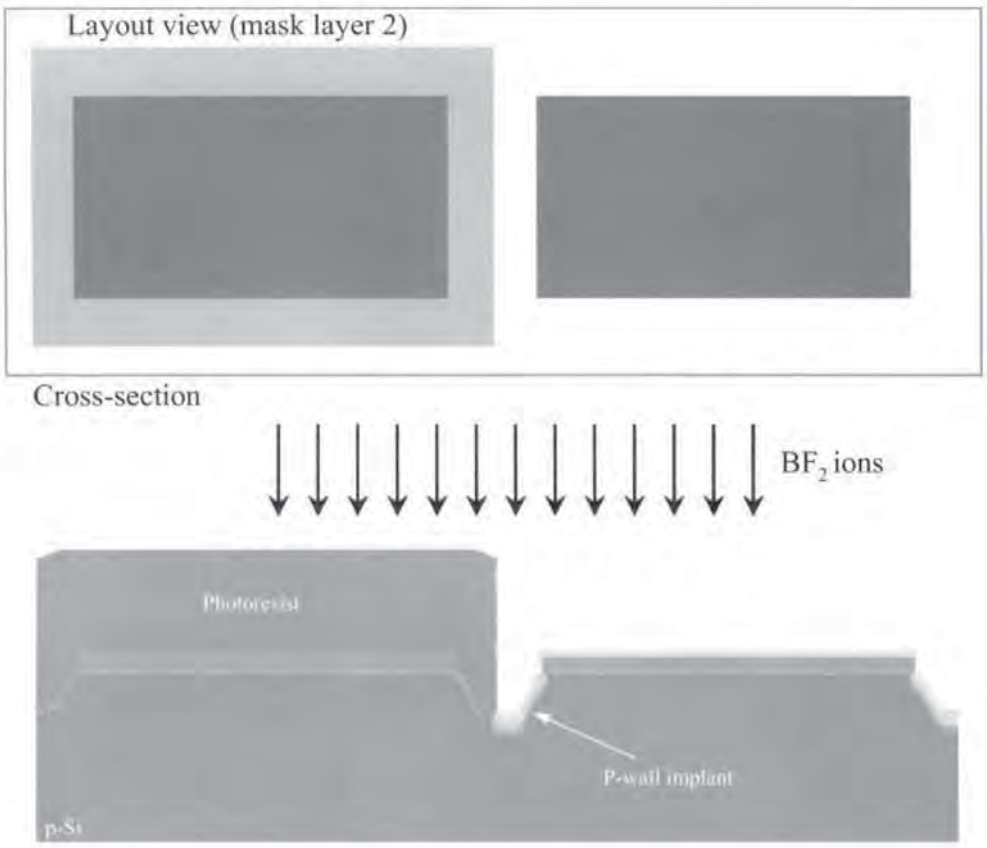



**Figure 7.33** P-wall sidewall formation via photolithography and $BF_2$ implantation.

subsequent ion implants. Although not shown, it is common to use a "block-out" pattern to improve the uniformity of the STI CMP. In Fig. 7.38a, CMP is performed to remove the CVD overfill. The nitride is used as a polish-stop layer. Next, a brief buffered oxide etch removes oxide that may have formed on top of the nitride. Then, the nitride is

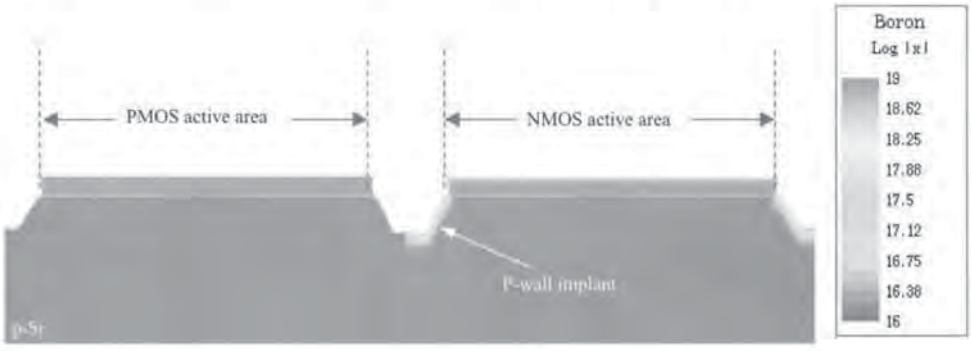



**Figure 7.34** Post p-wall photoresist strip using $O_2$ plasma and wet processing.

IPR2023-00697
Theta EX2019

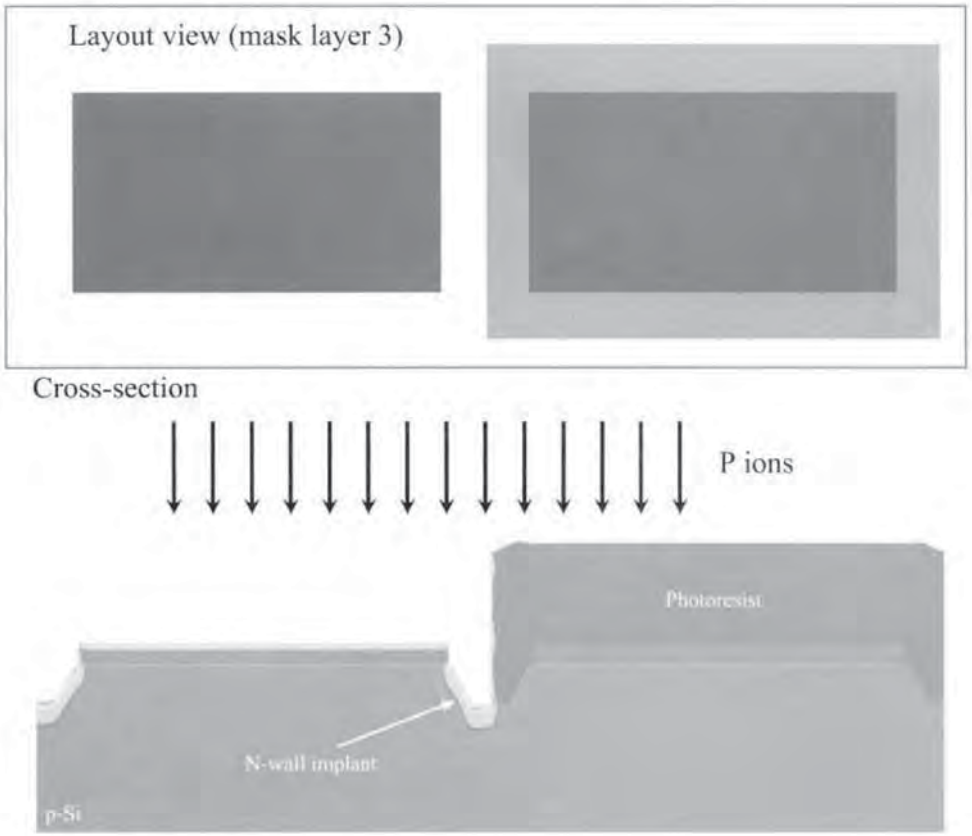



**Figure 7.35** N-wall sidewall formation via photolithography and *P* implantation.

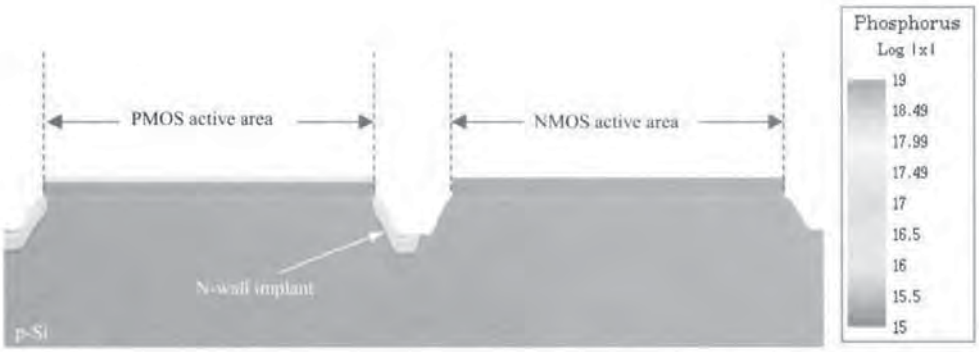



**Figure 7.36** Post n-wall photoresist strip using $O_2$ plasma and wet processing.

IPR2023-00697
Theta EX2019




**Figure 7.37** High quality, 100 Å thick liner oxide is thermally grown at 900 °C. High density plasma (HDP) CVD trench fill at room temperature. Notice that the trenches are overfilled.

removed from the active areas by using a wet or dry etch process, as illustrated in Fig. 7.38b. Notice that the pad oxide remains after this step. At this point, the STI is fully formed.

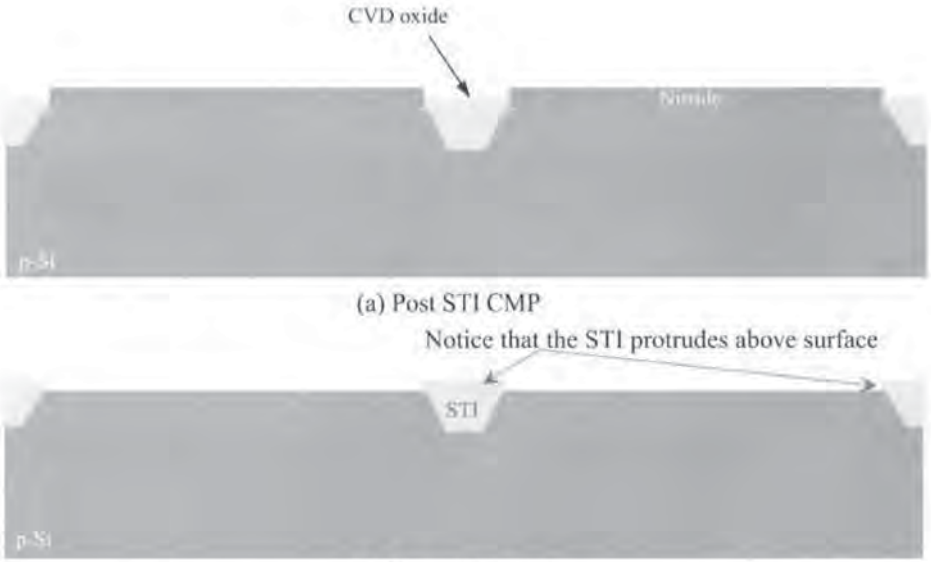



**Figure 7.38** STI CMP (a) where the nitride acts like a polish stop and (b) wet nitride etch in hot phosphoric acid and/or dry nitride etch in $NF_3$/Ar/NO. The remaining oxide will be used as a sacrificial oxide for subsequent implants.

*Twin-Tub Module*

As explained in Sec. 2.5, CMOS can be implemented in four general forms: n-well, p-well, twin-well (called twin-tub), and triple-well. The CMOS technology discussed in this chapter uses a twin-well approach. The p-well and n-well provide the appropriate

IPR2023-00697
Theta EX2019

dopants for the NMOS and PMOS, respectively. Modern wells are implanted with retrograde profiles to maximize transistor performance and reliability.

Following the STI module, the twin-tub module begins with p-well photolithography (mask layer 2, second use) to generate a resist pattern that covers the PMOS active regions, but exposes the NMOS active areas, as depicted in Fig. 7.39. A relatively high energy boron implant is performed into the NMOS active areas. Here the implant is blocked from the PMOS active area. The pad oxide that remained from the STI module now serves as the implant sacrificial oxide for the well implants. It should be pointed out that the p-well may be formed by a composition of several implants at different doses and energies to achieve the desired retrograde profile. Following the p-well implant, the resist is removed using $O_2$ plasma and wet processing, resulting in the structure shown in Fig. 7.40.

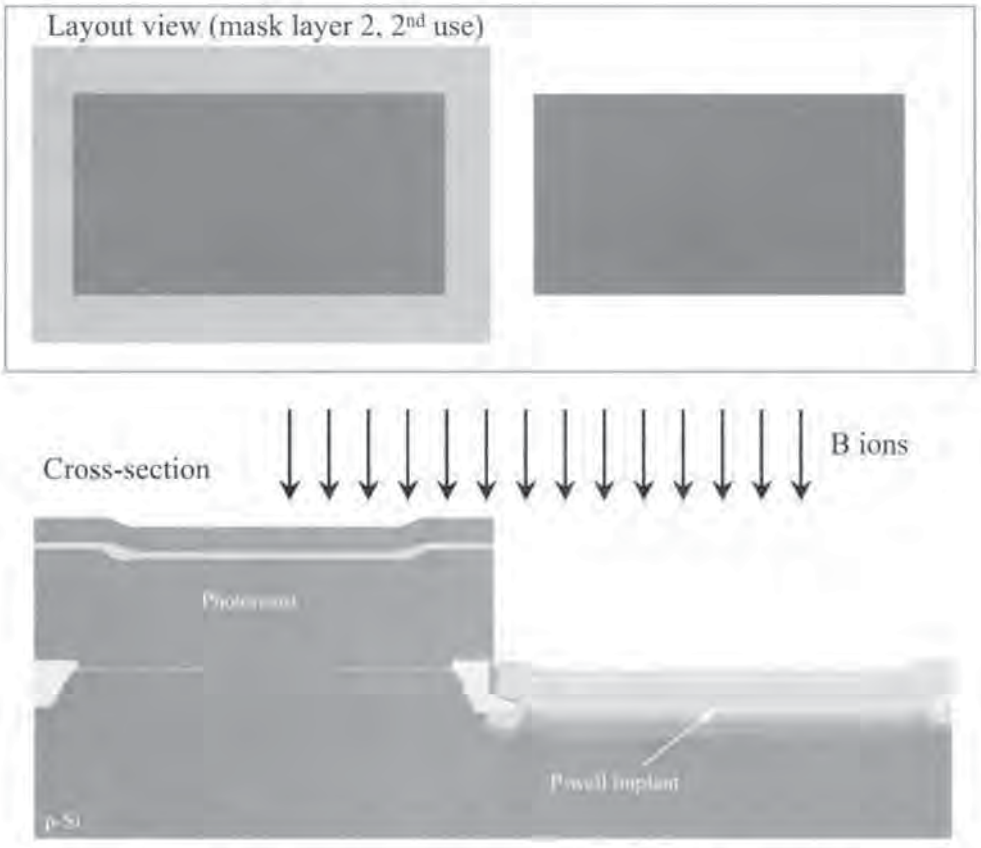



**Figure 7.39** P-well formation via photolithography and *B* implantation.

Next, a complementary resist pattern is formed using the n-well mask and photolithography (mask layer 3, second use), as seen in Fig. 7.41. Again, a relatively high energy implant, this time using phosphorus, is performed to generate the n-well. Similar to the p-well, a multitude of implants may be used to achieve the desired retrograde profile. Following the n-well implant, the resist is stripped using $O_2$ plasma and wet

IPR2023-00697
Theta EX2019

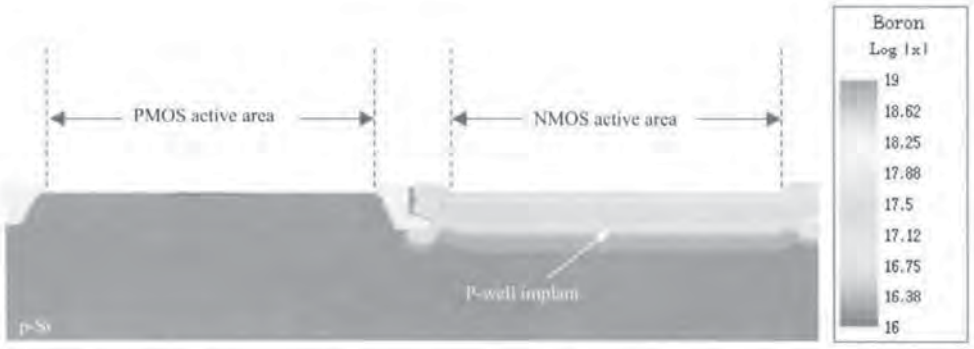



**Figure 7.40** Post p-well photoresist strip using $O_2$ plasma and wet processing.

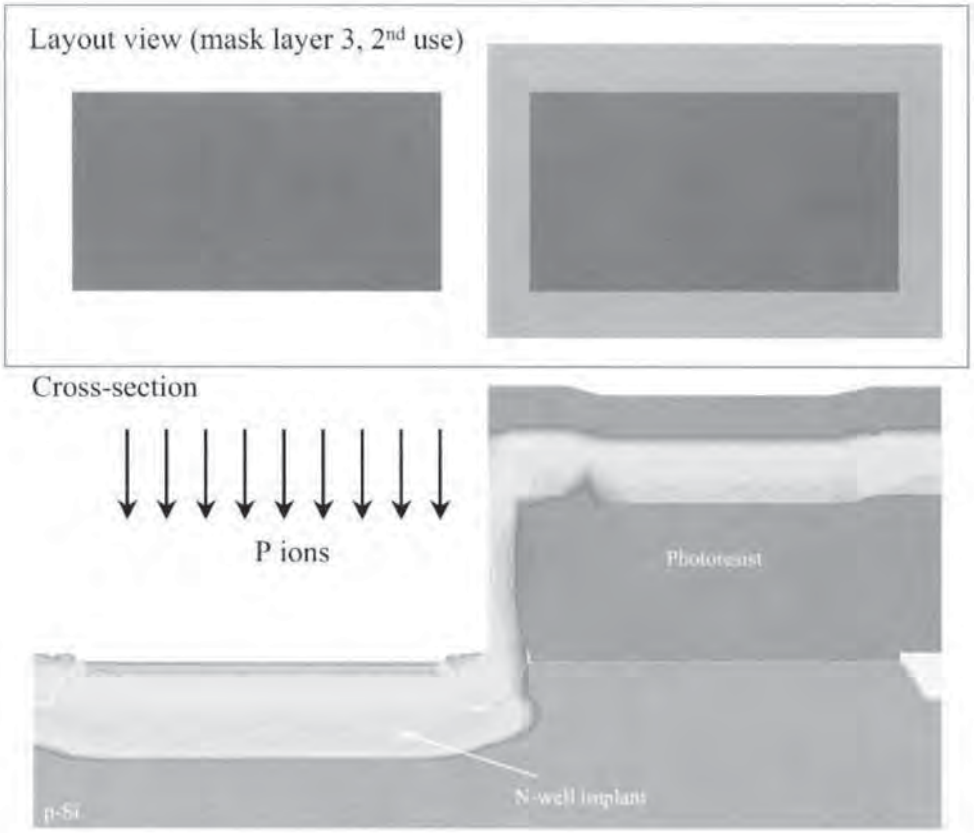



**Figure 7.41** N-well formation via photolithography and *P* implantation.

processing, yielding the structure in Fig. 7.42. At this point, both the isolation and the wells are fully formed. Figure 7.43 shows the cross section of the substrate following the twin-tub module. Notice that the net doping profile is given, thus highlighting both well and wall implants simultaneously. It should be emphasized that the PMOS are fabricated in the n-wells; the NMOS, in the p-wells.

IPR2023-00697
Theta EX2019

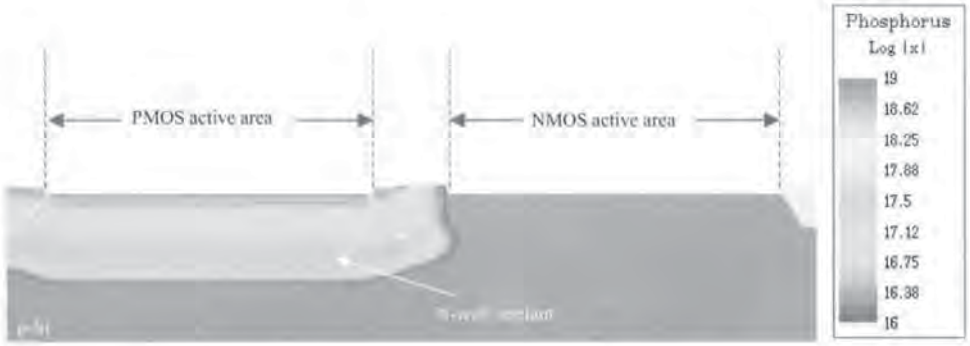



**Figure 7.42** Post n-well photoresist strip using $O_2$ plasma and wet processing.

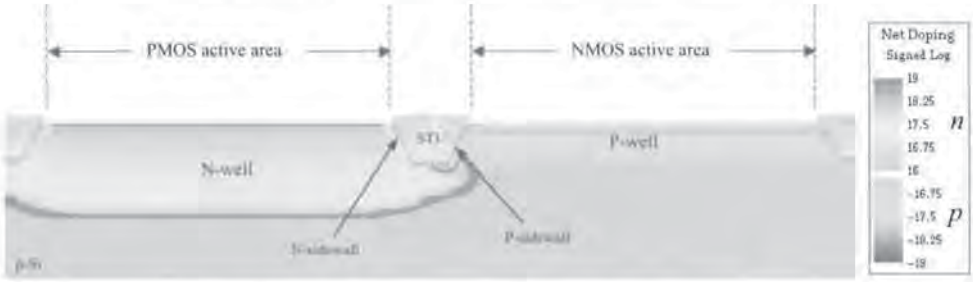



**Figure 7.43** Net doping profile of both the n-well and the p-well.

*Gate Module*

As depicted in Fig. 7.44, we begin the gate module with the buffered oxide etching of the remaining thin oxide in the active areas from the twin-tub module. Then, a sacrificial oxide is thermally grown. This oxide serves as a threshold adjust implant oxide and a pre-gate oxidation "clean-up." Next, a blanket (unpatterned), low energy $BF_2$ threshold adjust implant is performed, as shown in Fig. 7.45. This implant allows for the "tuning" of both the PMOS and NMOS threshold voltages. The single boron implant is common

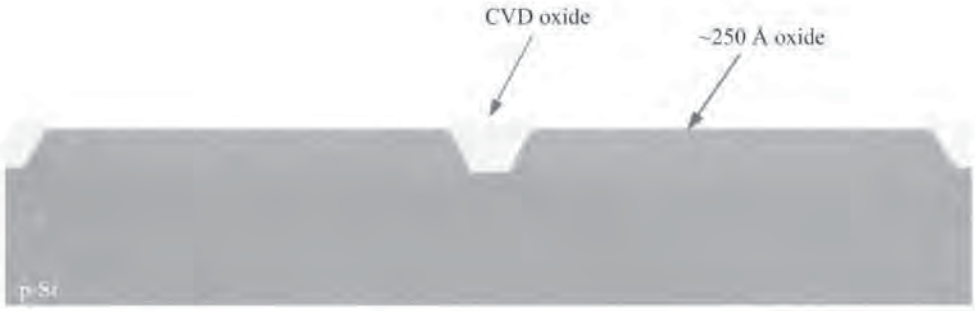



**Figure 7.44** Wet etch the remaining trench stack oxide using buffered *HF*. Sacrificial oxide formation using dry thermal oxidation at approximately 900 °C.

IPR2023-00697
Theta EX2019

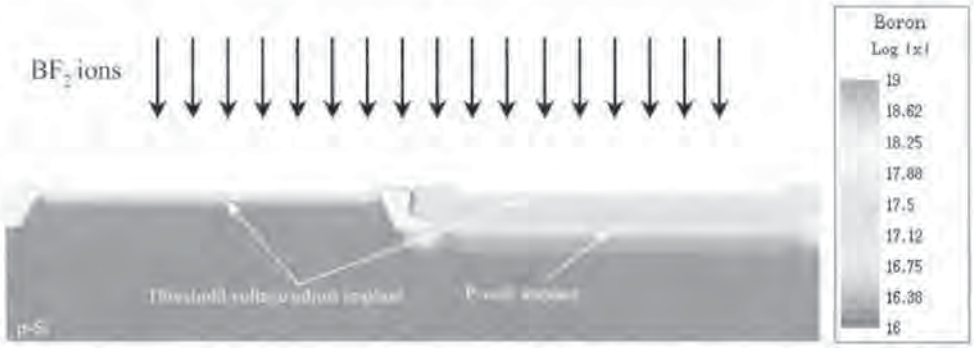



**Figure 7.45** Blanket low-energy $BF_2$ implant for NMOS and PMOS threshold voltage adjust.

for single workfunction gates. However, for dual workfunction gates (common in technologies with minimum gate lengths of 250 nm or less), separate p-type and n-type implants are required for the threshold adjustment in the NMOS and PMOS, respectively.

To form the gate stack (i.e., the gate dielectric and polysilicon gate electrode) the next set of processes are required. Of course, the gate stack provides for the capacitive coupling to the channel. Using wet processing, the sacrificial oxide is stripped from the active areas. As shown in Fig. 7.46, a high quality, thin oxide is thermally grown, which serves as the gate dielectric. In modern CMOS, it is common to use nitrided gate oxide by performing the oxidation in $O_2$ and $NO$ or $N_2O$. It can be argued that the gate oxidation is the most critical step in the entire process sequence, as the characteristic of the resultant film greatly determines the behavior of the CMOS transistors. The gate oxidation is immediately followed by an LPCVD polysilicon deposition, as depicted in Fig. 7.47. For single workfunction gates, the polysilicon can be doped with phosphorous during poly deposition or subsequently implanted. For dual workfunction gates, the NMOS and PMOS can be doped during the n+ and p+ source/drain implants, respectively.




**Figure 7.46** Removal of sacrificial oxide using buffered $HF$ followed by gate dielectric formation using dry oxidation in an ambient of $O_2$, $NO$ and/or $N_2O$.

Once the gate stack is formed, the transistor gates and local interconnects are patterned using photolithography (mask layer 4) to generate the appropriate patterns in photoresist, as seen in Fig. 7.48. The gate patterning must be precisely controlled as it

IPR2023-00697
Theta EX2019




**Figure 7.47** Polysilicon deposition via LPCVD at approximately 550 °C. Note that the polysilicon deposition must occur immediately following gate oxidation.

determines the gate lengths. Deviations in the resultant physical gate lengths can cause severe performance issues with the CMOS. Seen in Fig. 7.48 are the ideal gate profiles following the RIE of polysilicon and subsequent resist strip.

The gate module concludes with the poly reoxidation as shown in Fig. 7.49. Here the thermal oxidation of the polysilicon and active silicon is performed to (1) grow a buffer pad oxide for the subsequent nitride spacer deposition and (2) electrically activate the implanted dopants in the polysilicon. Notice that since the polysilicon oxidizes at a faster rate than the crystalline silicon, the resultant oxide thickness is greater on the polysilicon than the active silicon.

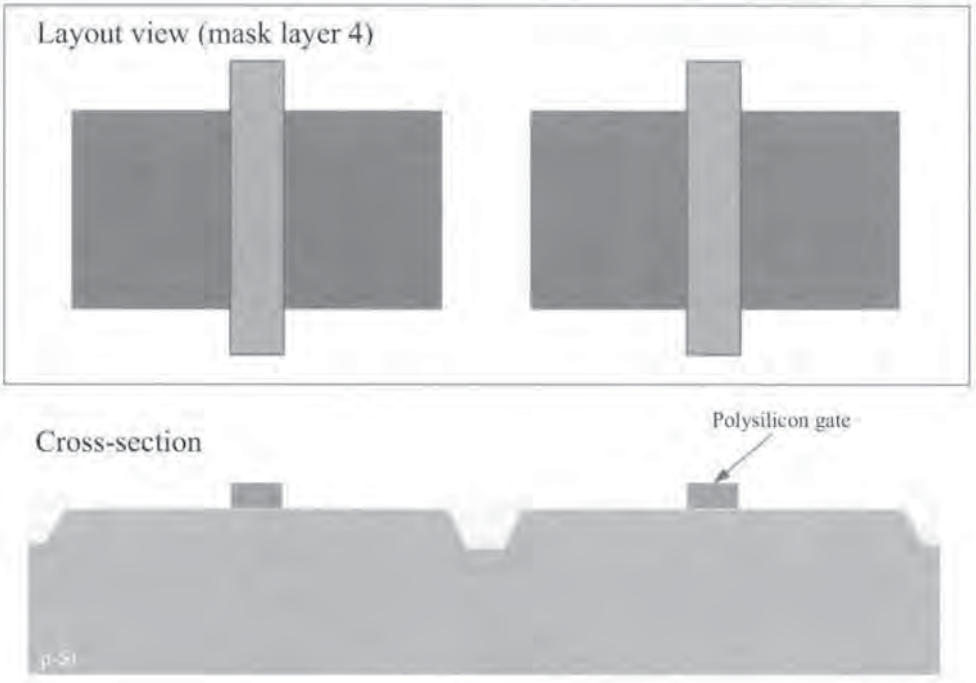



**Figure 7.48** Gate electrode and local interconnect photolithography and polysilicon reactive ion etching.

IPR2023-00697
Theta EX2019

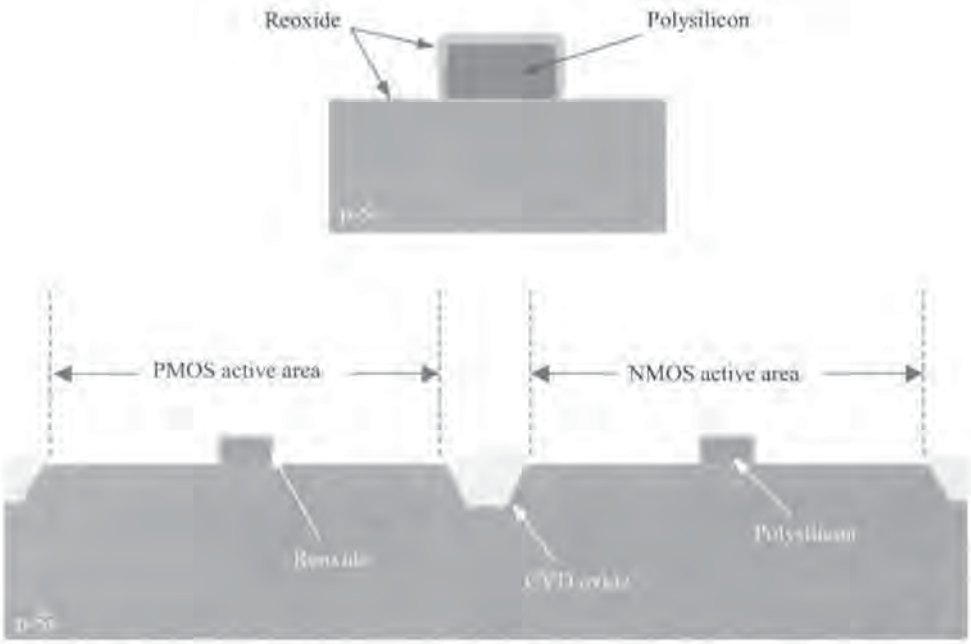



**Figure 7.49** Polysilcion reoxidation using dry $O_2$ at approximately 900 °C. Notice that the resulting oxide is thicker on the polysilicon than on the active silicon.

*Source/Drain Module*

At the onset of the source/drain module, the source/drain extensions are formed by a series of processes. Photolithography (mask layer 5) is used to pattern resist such that the NMOS devices are exposed, as shown in Fig. 7.50. Then, a low energy phosphorus implant is performed to form the n-channel, low doped drain (nLDD) extensions. Notice that the presence of the polysilicon gate inherently leads to the self-alignment of the extensions with respect to the gate. The nLDD suppresses hot carrier injection into the gate and reduces short-channel effects in the NMOS. At this point in the process sequence, a deep boron pocket implant is often used to prevent source/drain punchthrough in the NMOS. The photoresist is stripped. The resultant structure is shown in Fig. 7.51.

In a similar manner, the p-channel source/drain extensions are formed. Photolithography (mask layer 6) is used to protect NMOS devices with resist, as shown in Fig. 7.52. Boron is implanted at low energy to form the p-channel low doped drain (pLDD) extensions. Again, the polysilicon serves to self-align the implant with respect to the gate electrodes. As was the case with the NMOS, it is common to use a deep phosphorus pocket implant to suppress PMOS punchthrough. Once the photoresist is stripped, the cross-section shown in Fig. 7.53 is achieved.

To complete the source/drain extensions, the gate sidewall spacers must be formed prior to the actual source/drain implants. As shown in Fig. 7.54, conformal silicon nitride is deposited using LPCVD. A CVD oxide may be used in lieu of the nitride. Following the subsequent nitride etch, this nitride will form the LDD sidewall spacers.

IPR2023-00697
Theta EX2019

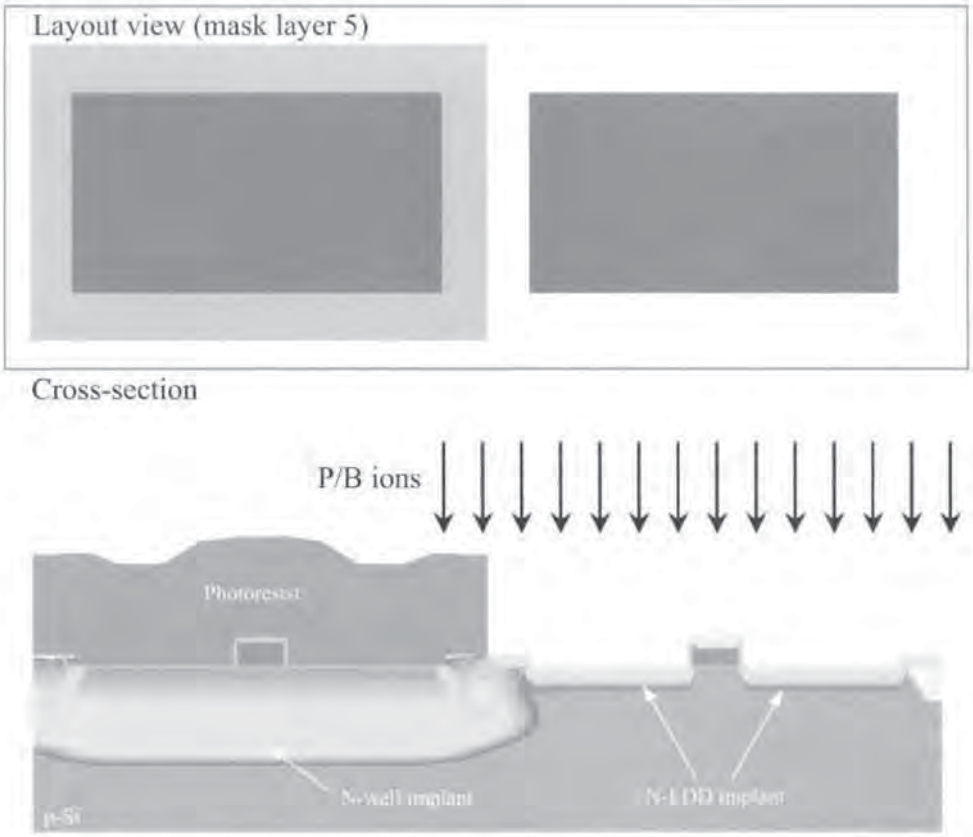



**Figure 7.50** N-LDD/n-pocket formation using low energy implantation of *P* and *B*, respectively.

The spacers function as (1) a mask to the source/drain implants and (2) a barrier to the subsequent salicide formation. The actual spacers are formed by an unpatterned anisotropic RIE of nitride, as illustrated in Fig. 7.55. The spacer etch is end-pointed on the underlying oxide. Notice that since the nitride is thickest along the polysilicon

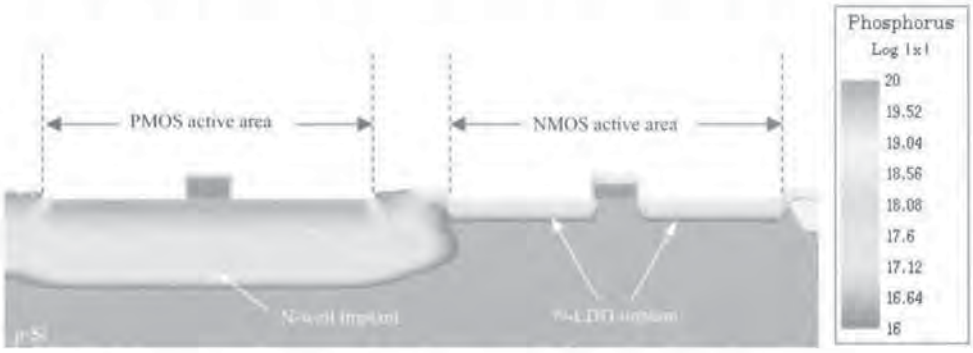



**Figure 7.51** Post-n-LDD resist strip using $O_2$ plasma and wet processing.

IPR2023-00697
Theta EX2019

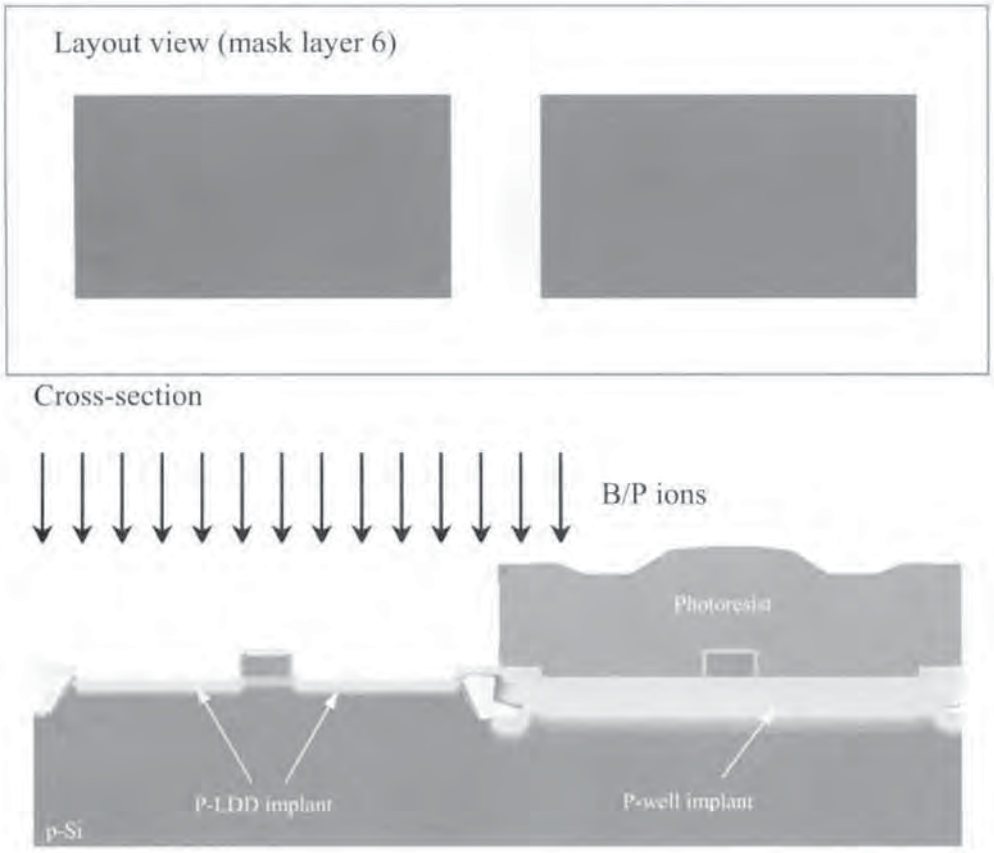



**Figure 7.52** P-LDD/p-pocket formation using low energy implantation of *B* and *P*, respectively.

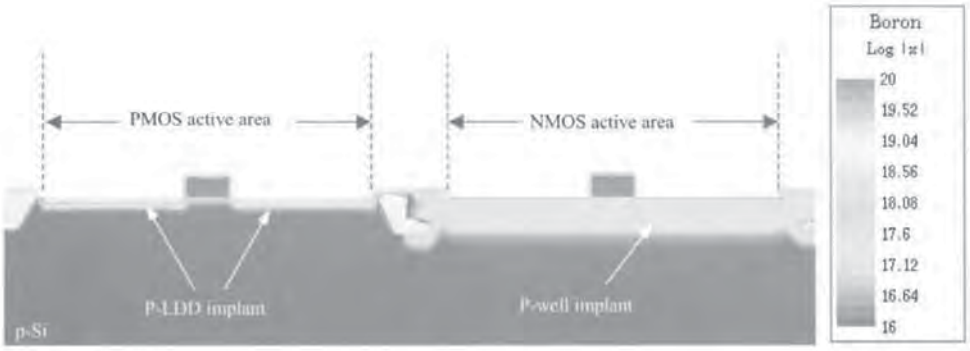



**Figure 7.53** Post-p-LDD resist strip using $O_2$ plasma and wet processing.

IPR2023-00697
Theta EX2019




**Figure 7.54** Sidewall spacer nitride deposition using LPCVD at approximately 800 °C.

sidewall, a well-formed insulating region remains on both sides of the polysilicon. This structure is called a spacer.

During the source/drain implants, the combination of the polysilicon and spacers block the implantation, thus allowing for self-alignment to not only the gate but also to the LDD extensions. With this stated, the NMOS source/drains are formed, as shown in Fig. 7.56. Photolithography (mask layer 5, second use) protects the PMOS with resist while exposing the NMOS. A relatively low energy, high dose arsenic implant is performed to form the n+ regions. The resist is stripped yielding the structure in Fig. 7.57. In addition to the source/drain formation, this implant forms the necessary n+ ohmic contacts.




**Figure 7.55** Dry, anisotropic, end-pointed reactive ion etch of spacer nitride yielding gate sidewall spacers.

The PMOS source/drains and p+ ohmic contacts are formed in a similar manner. Photolithography (mask layer 6, second use) and a low energy, high dose $BF_2$ implant is used, as illustrated in Fig. 7.58. The resist is stripped resulting in the structure shown in Fig. 7.59. The source/drain module concludes with a high temperature anneal that electrically activates the implants and re-crystallizes the damaged silicon. In modern CMOS, the primary reason that polysilicon is chosen as the gate electrode material as opposed to metal is that the poly can withstand the high temperatures required to activate the source/drain implants.

At this point in our CMOS process sequence we have fully formed the CMOS transistors and their isolation. This marks the completion of the FEOL. Figure 7.60 provides a summary of the main features generated in the FEOL.

IPR2023-00697
Theta EX2019

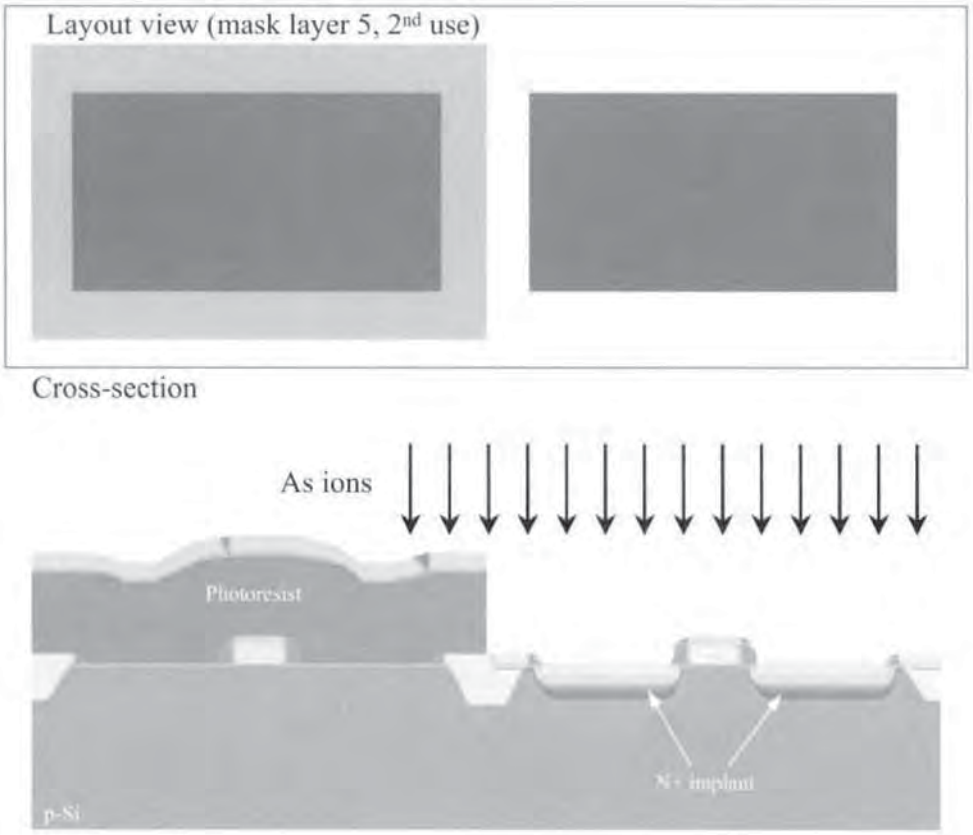



**Figure 7.56** N+ source/drain formation using a low energy, high dose implantation of *As*.

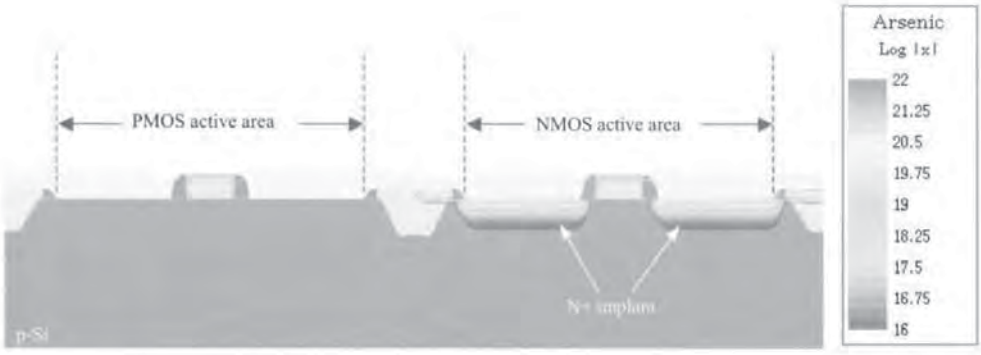



**Figure 7.57** Post-n+ resist strip using $O_2$ plasma and wet processing.

IPR2023-00697
Theta EX2019

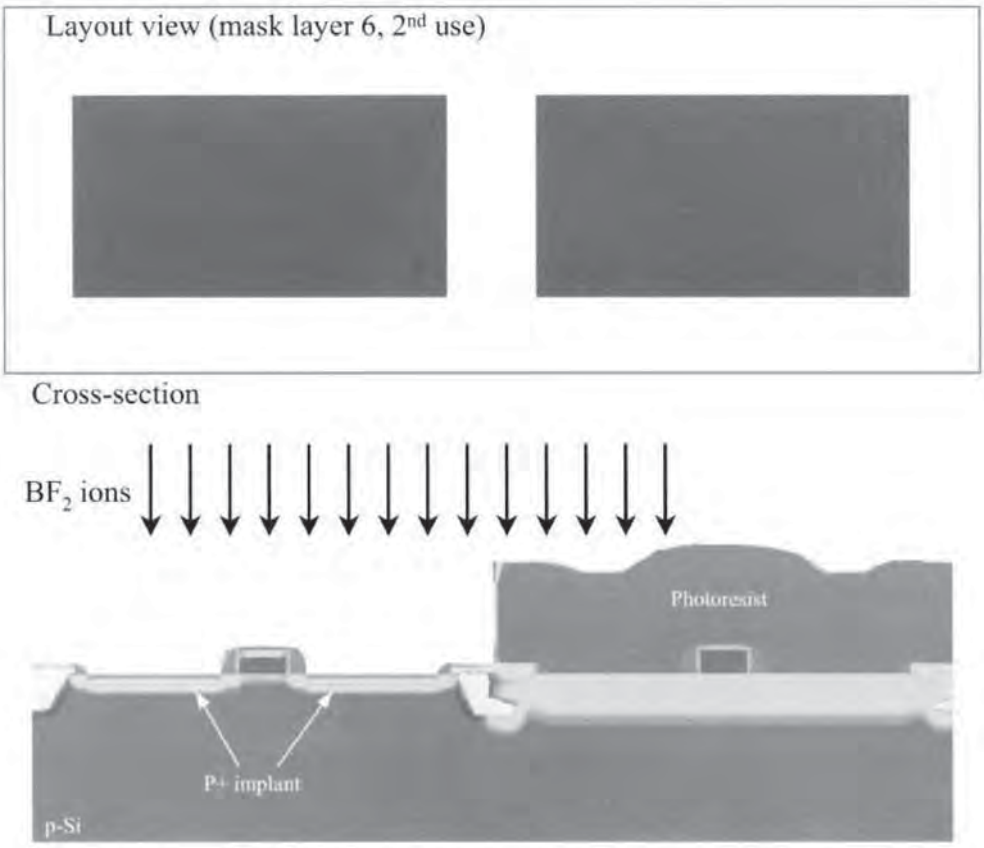



**Figure 7.58** P+ source/drain formation using a low energy, high dose implantation of $BF_2$.

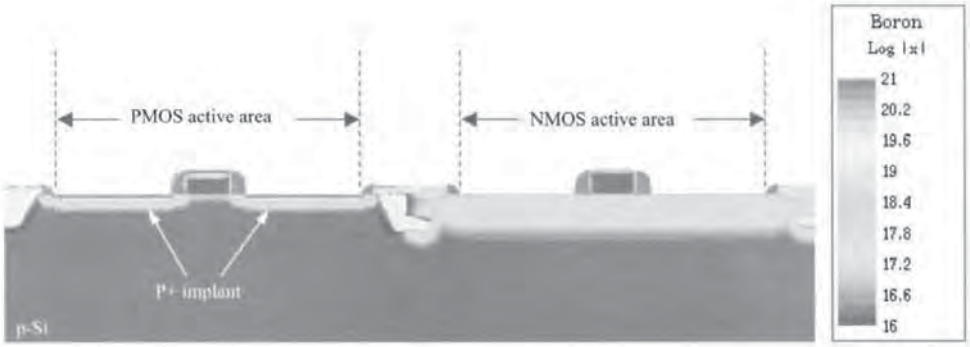



**Figure 7.59** Post p+ resist strip using $O_2$ plasma and wet processing.

IPR2023-00697
Theta EX2019

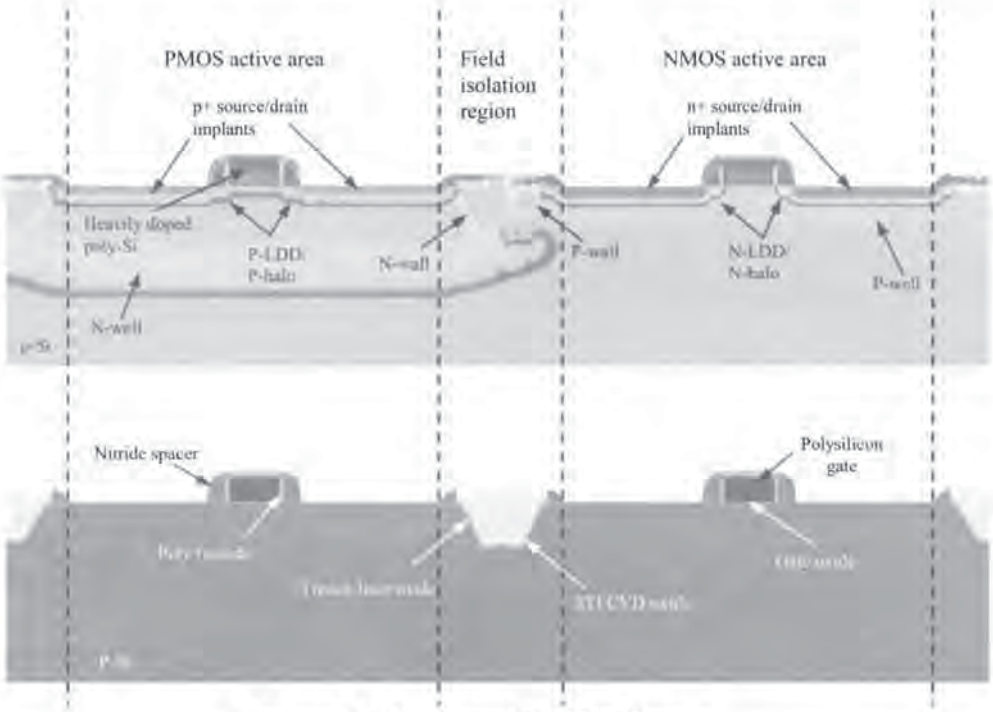



**Figure 7.60** Summary of the FEOL features.

### 7.2.2 Backend-of-the-Line Integration

In this section we continue our CMOS process flow through the BEOL. The BEOL encompasses all processes required to "wire" the transistors to one another and to the bond pads. CMOS requires several metal layers to achieve the interconnects necessary for modern designs. We discuss the processing through the first two metal layers to give an appreciation of the overall BEOL integration.

*Self-Aligned Silicide (Salicide) Module*

At the boundary of the FEOL and BEOL is the self-aligned silicide, called salicide, formation. Silicide lowers the sheet resistance of the polysilicon and active silicon regions. The self-aligned silicide (salicide) relies on the fact that metal silicide will generally not form over dielectric materials such as silicon nitride. Therefore, a metal such as titanium or cobalt can be deposited over the entire surface of the wafer, then annealed to selectively form silicide over exposed polysilicon and silicon. Because of the presence of the trench fill and sidewall spacers, the silicide become self-aligned without the need for photopatterning.

The salicide module begins with the removal of the thin oxide, present from the FEOL, using buffered *HF*, as shown in Fig. 7.61. Next, a refractory metal (e.g., titanium or cobalt) is deposited by sputtering, as depicted in Fig. 7.62. To minimize contamination, a thin layer of *TiN* is deposited as a cap. A relatively low temperature, nitrogen ambient, rapid thermal annealing (RTA) is used to react titanium (or cobalt) with the silicon, forming $TiSi_2$ (C49 phase) or ($CoSi_x$). The resultant silicide (i.e., C49) is a high resistivity

IPR2023-00697
Theta EX2019




**Figure 7.61** Removal of the exposed reoxide (present from the FEOL) using buffered-*HF*.

phase. Also, notice that the underlying nitride and oxide serves to block the formation of the silicide from the sidewalls and trenches, respectively.

To prevent spacer overgrowth of the silicide, the low resistivity phase is achieved by processing with two separate anneals. The first, as described above, forms the high resistivity phase without the risk of silicide formation on the nitride. The second, as described below, occurs following the wet chemical etching of the unreacted titanium (or cobalt) using a higher temperature, which causes a phase change (C49 to C54 for $TiSi_2$) with a much lower resistivity. If one high temperature anneal was originally performed to achieve the low resistivity phase, then significant overgrowth could occur, leading to leakage current from the source and drain to the gate of the transistors.

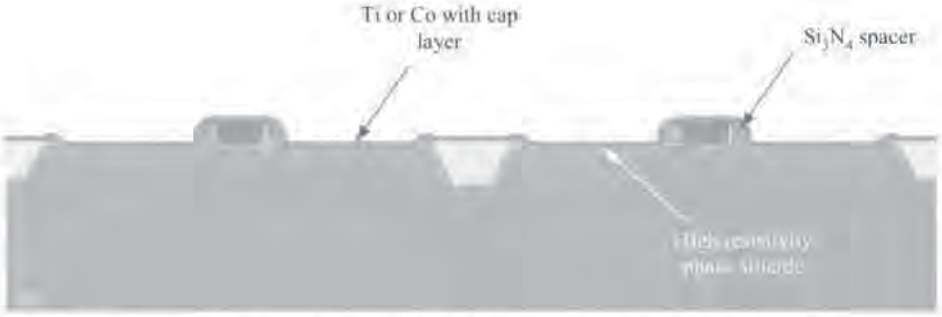



**Figure 7.62** Titanium or cobalt deposited by PVD followed by the first salicide rapid thermal anneal.

To continue the salicide module, following the first anneal, the unreacted titanium (or cobalt) is wet chemically etched from the wafer. The second RTA, in argon ambient at a slightly higher temperature, achieves the low resistivity phase, as shown in Fig. 7.63.

*Pre-Metal Dielectric*

The pre-metal dielectric (PMD) provides electrical isolation between metal1 and polysilicon/silicon. To aid the subsequent contact etch process, a thin layer of silicon nitride is deposited as an etch stop. This is followed by a high density plasma deposition of the PMD oxide, as shown in Fig. 7.64. The resultant surface of the PMD must be

IPR2023-00697
Theta EX2019




**Figure 7.63** Wet chemical etch of the unreacted titanium or cobalt followed by the second salicide rapid thermal anneal.

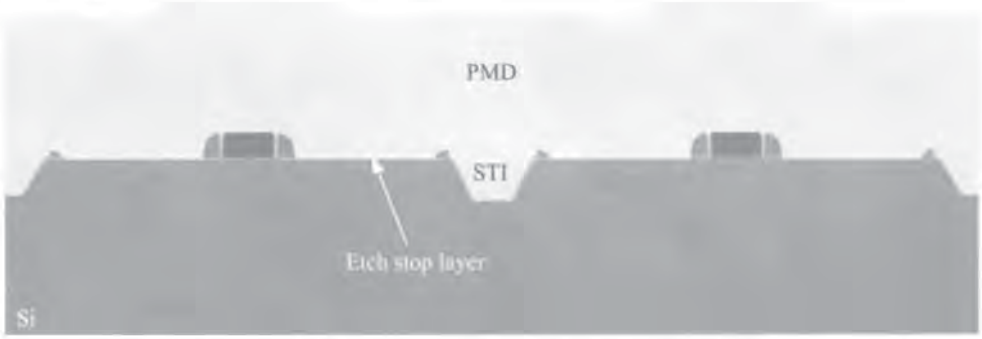



**Figure 7.64** Pre-metal dielectric (PMD) deposition using high density plasma

planarized to allow for improved depth-of-focus for the subsequent high resolution photopatterning of metal1. With CMP, the PMD is planarized, producing the cross-section shown in Fig. 7.65.

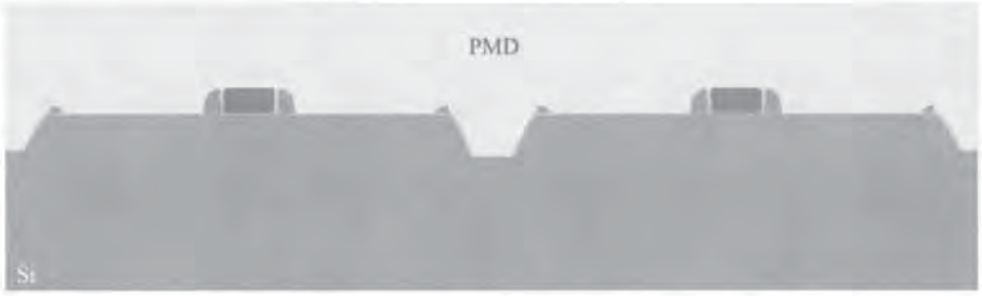



**Figure 7.65** Planarizing of the PMD using CMP.

IPR2023-00697
Theta EX2019

*Contact Module*

The contacts provide the electrical coupling between metal1 and polysilicon/silicon. The first BEOL photolithography (mask layer 7) step patterns contact openings in the resist. The PMD and nitride is then dry etched using the nitride as an etch-stop layer. The resist is stripped from the wafer, resulting in the structure shown in Fig. 7.66. Contact openings to the source and drains are shown; however, contacts to polysilicon (not shown) over field oxide are simultaneously formed.

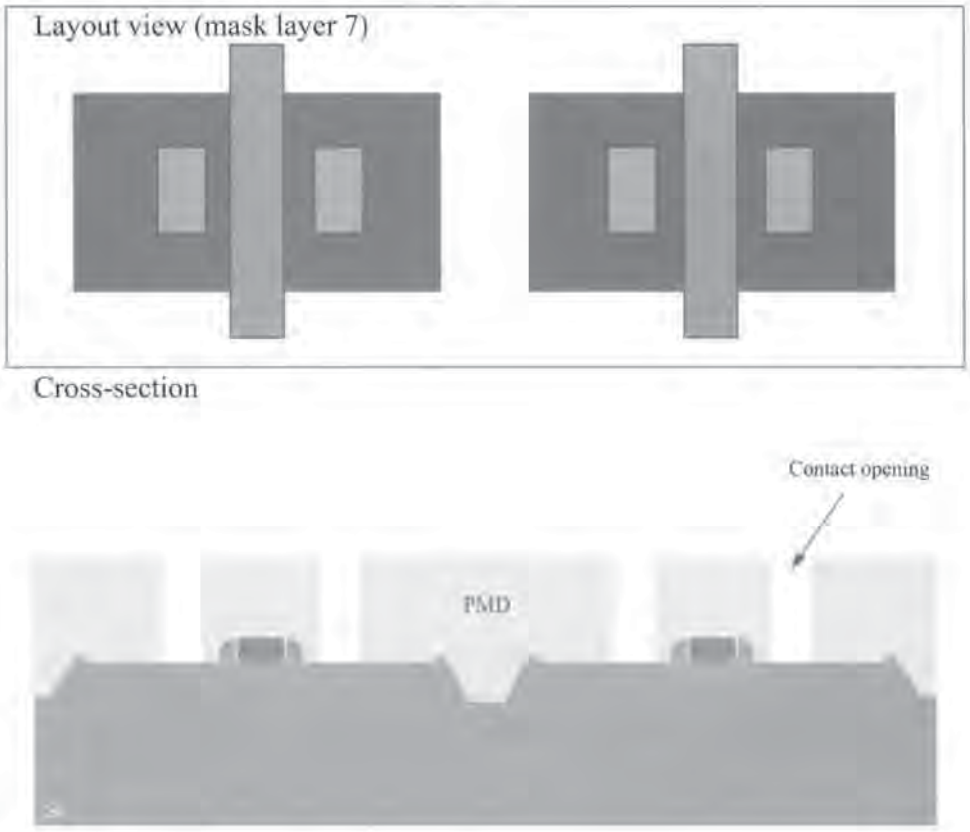



**Figure 7.66** Contact definition using photolithography and RIE of the PMD and stop layer. Notice that the contacts to the poly are formed at the same time but not shown.

Next, a thin layer of titanium is deposited by ionized metal plasma (IMP) sputtering, preceded by an in-situ argon sputter etch to clean the bottom of the high aspect ratio contact openings. The titanium is the first component of the contact liner and functions to chemically reduce oxides at the bottom of the contacts. The second component of the liner is a thin CVD *TiN*. The primary purpose of this layer is to act as a diffusion barrier to the fluorine (which readily etches silicon) used in the subsequent tungsten deposition. The contact openings are filled (actually overfilled) with tungsten using a $WF_6$ CVD process, as shown in Fig. 7.67 As depicted in Fig. 7.68, the overfilled tungsten is polished back to the top of the planarized PMD using CMP. At this point, the surface of the wafer is ultra-smooth and essentially free of all topography. To aid the

IPR2023-00697
Theta EX2019




**Figure 7.67** *Ti/TiN* liner deposition using IMP and CVD, respectively. *W* contact fill deposition using $WF_6$ CVD.




**Figure 7.68** *W* CMP to form defined contacts.

photolithographic alignment of the metal layer to the contacts, a tungsten recess etch is performed to provide adequate alignment reference. The recessed contacts are shown in Fig. 7.69.

*Metallization 1*

To allow for electrical signal transmission from contact-to-contact and from contact-to-via1, defined metallization must be formed. Following the recess etch, sputtering is used to deposit a film stack consisting of *Ti/TiN/Al/TiN*, as seen in Fig. 7.70. The Ti provides adhesion of the TiN and reduction in electromigration problems. The bottom TiN serves primarily as a diffusion barrier to $TiAl_3$ formation. The topmost TiN acts as an anti- reflective coating for the metal photolithography as well as an etch stop for the subsequent via formation.

Using photolithography (mask layer 8), the metal 1 pattern in resist is generated, as depicted in Fig. 7.71. A dry metal etch transfers the pattern into the metal. To prevent metal corrosion, the resist is plasma stripped in an $O_2/N_2/H_2O$ ambient, producing the cross-section shown in Fig. 7.71.

IPR2023-00697
Theta EX2019

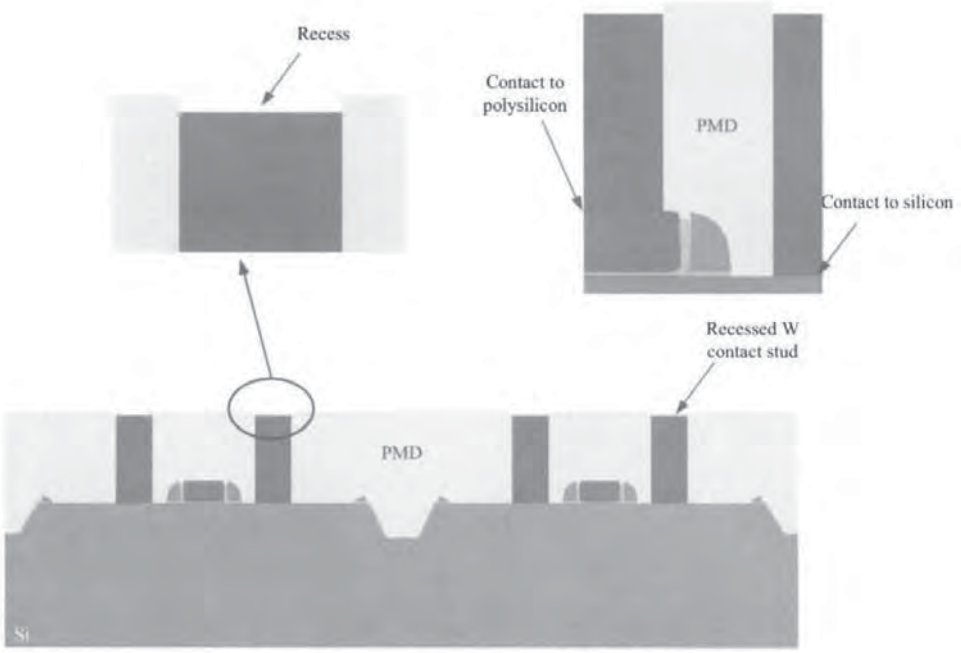



**Figure 7.69** *W* recess etch using "buff" polish or dry *W* etch.

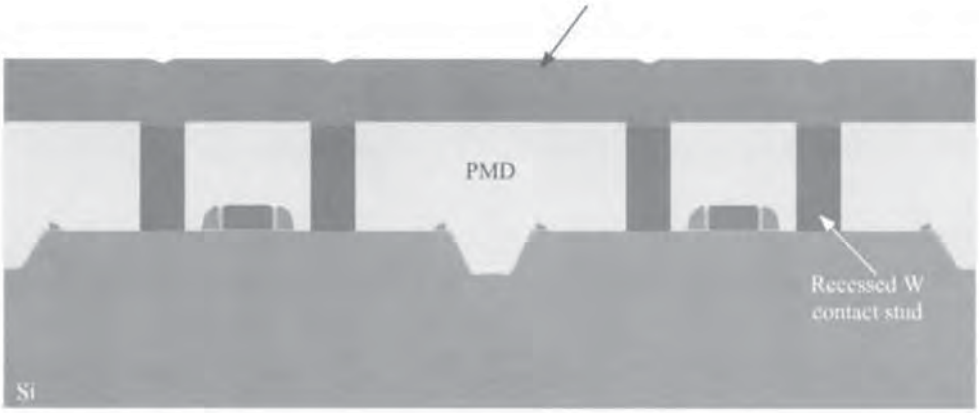



**Figure 7.70** Metal1 stack deposition using PVD.

IPR2023-00697
Theta EX2019

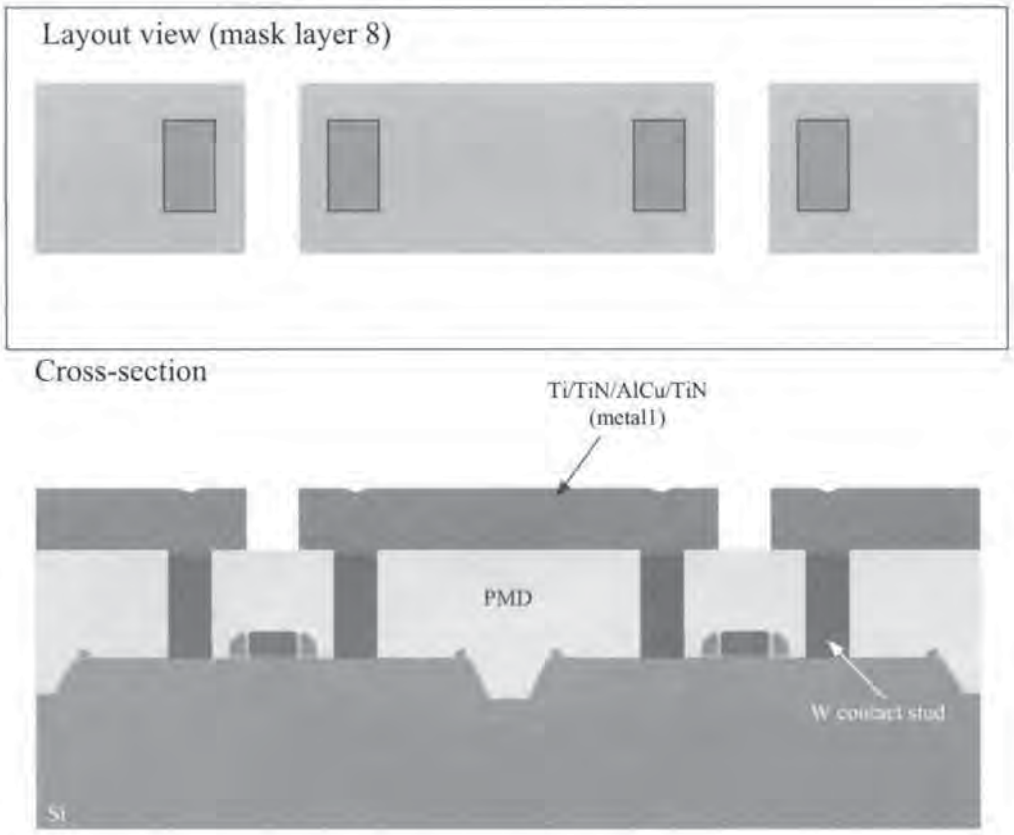



**Figure 7.71** Metal1 definition using photolithography and dry metal etch.

*Intra-Metal Dielectric 1 Deposition*

The intra-metal dielectric 1 (IMD1) provides the electrical isolation between metal1 and metal2. It is common to deposit this film using high density plasma CVD, as shown in Fig. 7.72. As can be seen, the conformal deposition results in a surface topography that must be planarized by CMP. The depth-of-focus is improved for the subsequent photolithographic steps as it is for the PMD planarization.

*Via 1 Module*

Electrical coupling between metal1 and metal2 is achieved by the via1 module. The planarized IMD is photolithographically defined (mask layer 9), and an RIE opens the vias. The resist is stripped using $O_2$ plasma and wet processing, producing the cross-section shown in Fig. 7.73. Next, similar to the contact fill, an argon sputter etch is performed followed by the deposition of thin layers of IMP titanium and CVD *TiN*. Using a $WF_6$ CVD process, the vias are deposited (overfilled), as seen in Fig. 7.74. The excess tungsten is removed by CMP utilizing the IMD1 as a "polish stop," as depicted in Fig. 7.75. Again, as to provide observable alignment features, a tungsten recess etch is often required (also shown in Fig. 7.75).

IPR2023-00697
Theta EX2019

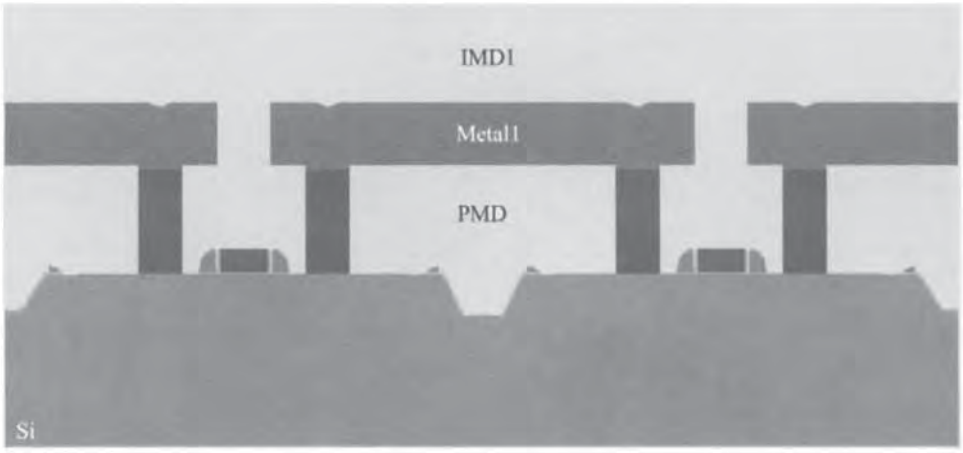



**Figure 7.72** Intra-metal dielectric 1 (IMD1) deposition using HDP CVD. This is followed by IMD1 planarization using CMP.

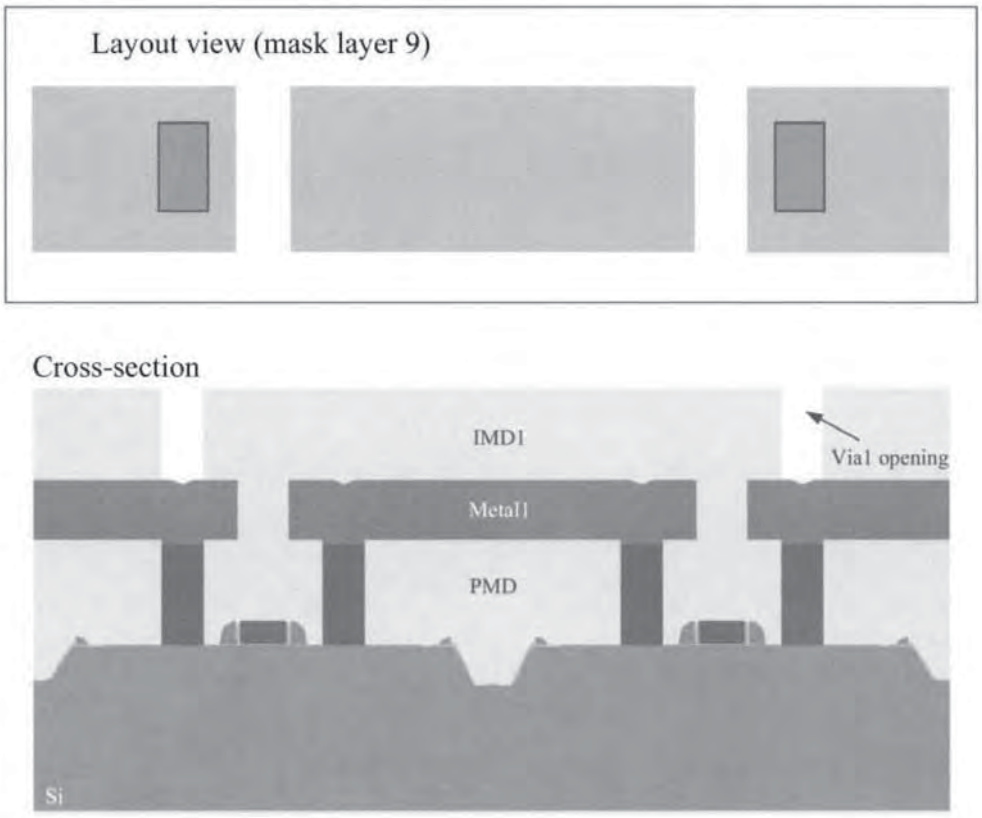



**Figure 7.73** Via1 definition using photolithography and dry IMD1 etch.

IPR2023-00697
Theta EX2019

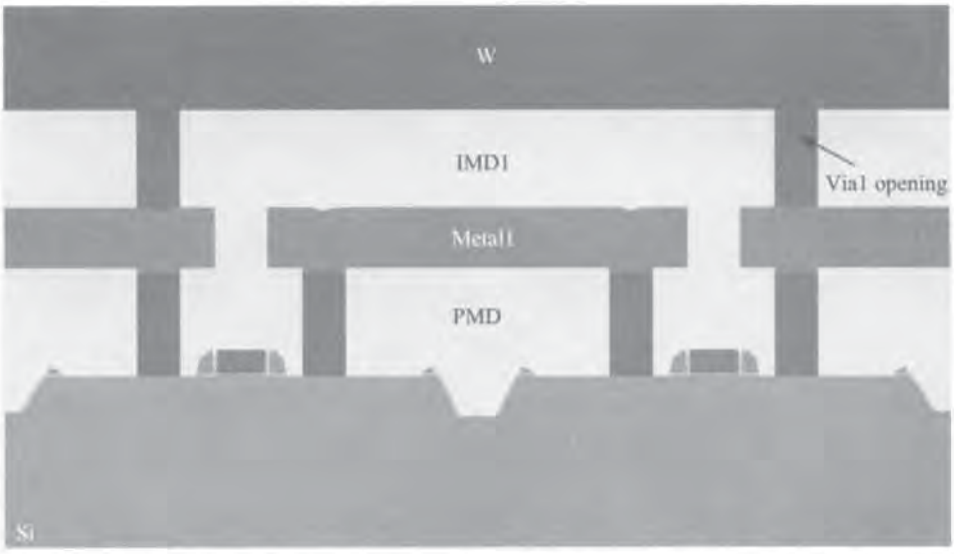



**Figure 7.74** *Ti/TiN* liner deposition using IMP and CVD, respectively. *W* via1 fill deposition using $WF_6$ CVD.

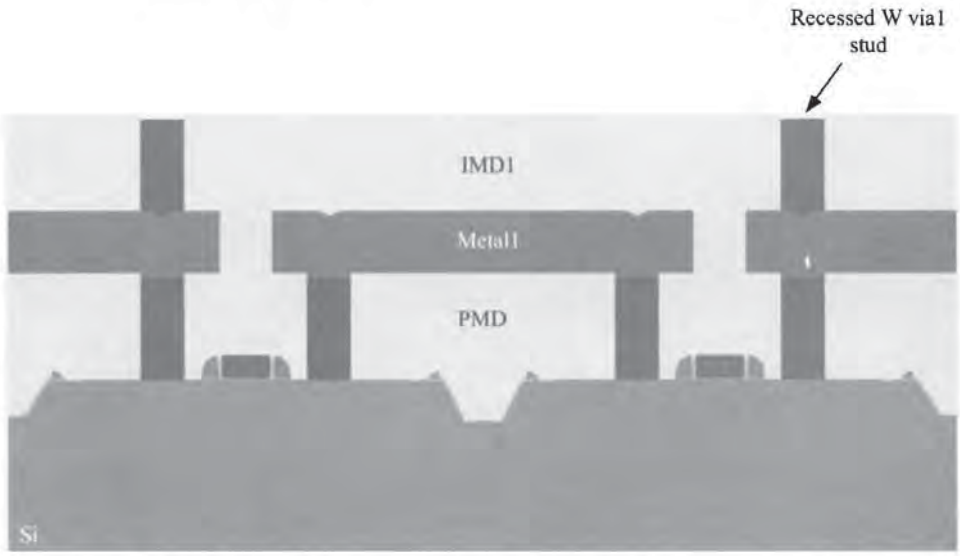



**Figure 7.75** *W* CMP to form defined vias. This is followed by *W* recess etch using "buff" polish or dry *W* etch.

*Metallization 2*

In a similar manner as the metal1 process, the metal 2 stack is deposited (Fig. 7.76) and photolithographically (mask layer 10) defined, as shown in Fig. 7.77. Note that we are not discussing metal implementation using copper. Copper wiring is often implemented with dual-Damascene techniques, see Ch. 4, where both vias and metal layers are simultaneously formed in a series of process steps.

IPR2023-00697
Theta EX2019

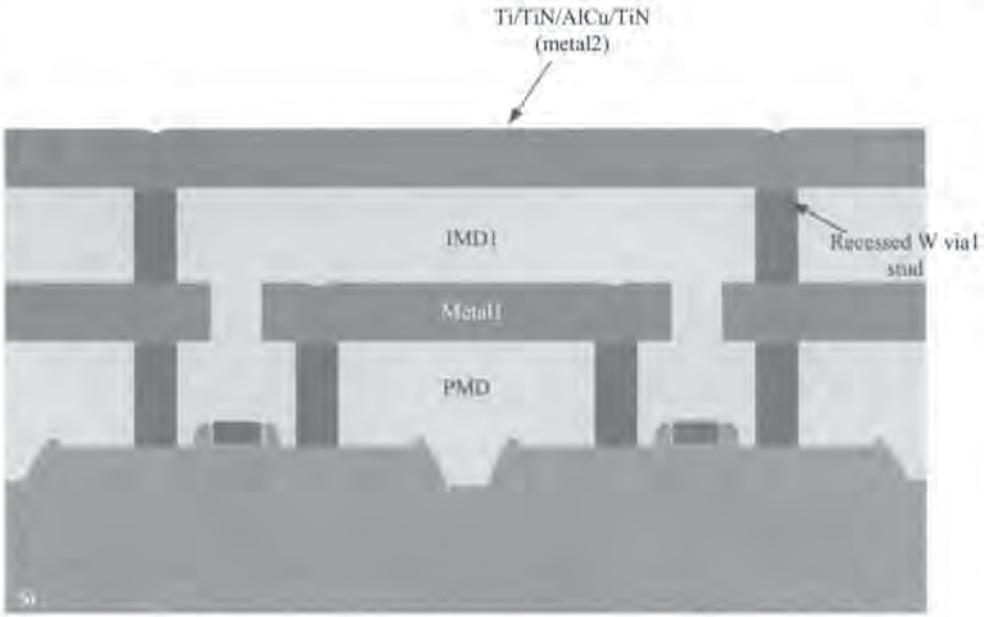



**Figure 7.76** Metal2 stack deposition using PVD.

*Additional Metal/Dielectric Layers*

At this point, additional tiers of dielectric/metal layers can be formed by replicating the aforementioned processes. In modern CMOS there may be more than eight metal layers. It should be noted that as dielectric/metal layers are added, the cumulative film stresses can cause significant bow/warp in the wafers. Hence, great effort is expended to minimize the stresses in the BEOL films.

*Final Passivation*

To protect the CMOS from mechanical abrasion during probe and packaging and to provide a barrier to contaminants (e.g., $H_2O$, ionic salts), a final passivation layer must be deposited. The passivation type is determined in large part by the type of package in which the CMOS IC will be placed. Common passivation layers are (1) doped glass and (2) silicon nitride on deposited oxide. Figure 7.78 shows the cross-section of our CMOS process flow following the deposition of the passivation. Finally, the bond pads are opened using photolithography (mask layer n) and by dry etching the passivation. Following photoresist strip, the final CMOS cross-section is shown in Fig. 7.79.

## 7.3 Backend Processes

Following completion of the final passivation, the wafers are removed from the cleanroom in preparation for a series of backend (i.e., post-fab) processes. These processes include wafer probe, die separation, packaging, and final test/burn-in.

*Wafer Probe*

Generally, dedicated die with parametric structures and devices are stepped into various positions of the wafer. Alternatively, parametric structures are placed in the dividing

IPR2023-00697
Theta EX2019

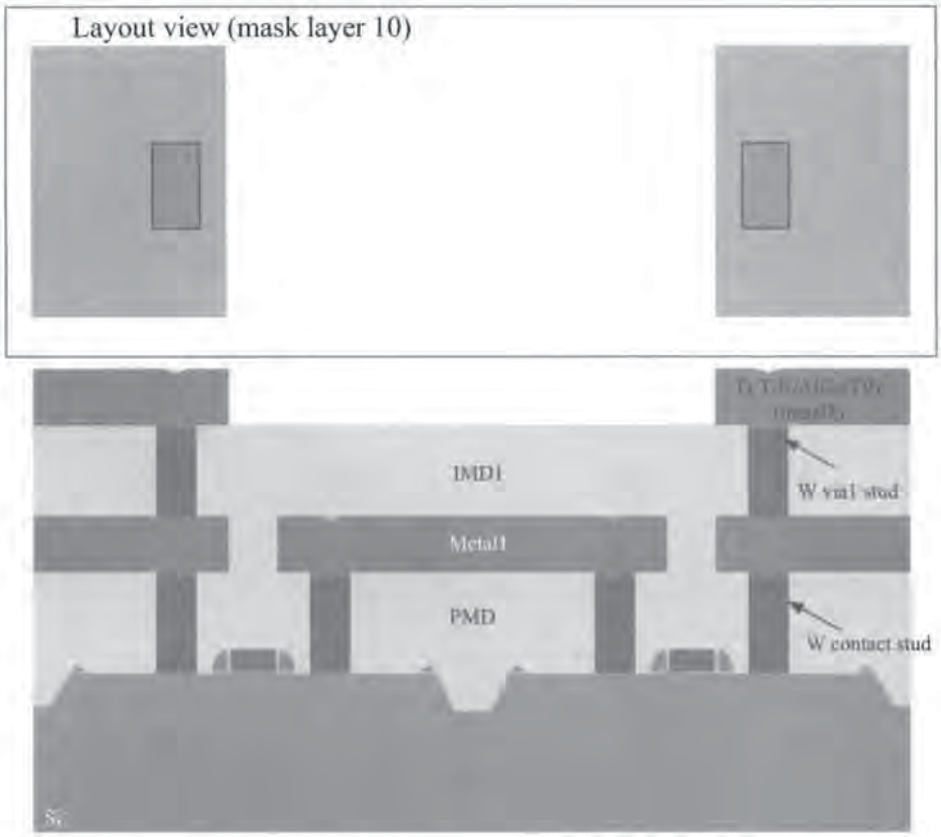



**Figure 7.77** Metal2 definition using photolithography and dry metal etch.

regions, called *streets* or *scribe lines*, between die. Electrical characterization of these parametric structures and devices is often performed at select points in the fabrication process flow (such as following metal 1 patterning). Parameters such as contact resistance, sheet resistance, transistor threshold voltage, saturated drain current, off-current, sub-threshold slope, etc., are measured. If problems are observed from the in-line parametric tests, troubleshooting can begin sooner than if the testing were only performed following final passivation. Furthermore, wafers that do not meet the parametric standards can be removed (or "killed") from the fabrication sequence (and thus money is saved).

After the completed wafers are removed from the fab, wafer-level probing is performed to check final device parameters and to check CMOS integrated circuit functionality and performance. Wafer probe is accomplished by using sophisticated testers that can probe individual die (or sets of die) and apply test vectors to determine circuit behavior. Inevitablity, there are a percentage of die that will not pass all the vector sets and thus are considered failed die. The ratio of good die-to-total die represents the wafer *yield* given by

$$Y = \frac{\#\text{ of die passing all tests}}{\text{total }\#\text{ of die}} \tag{7.13}$$

IPR2023-00697
Theta EX2019

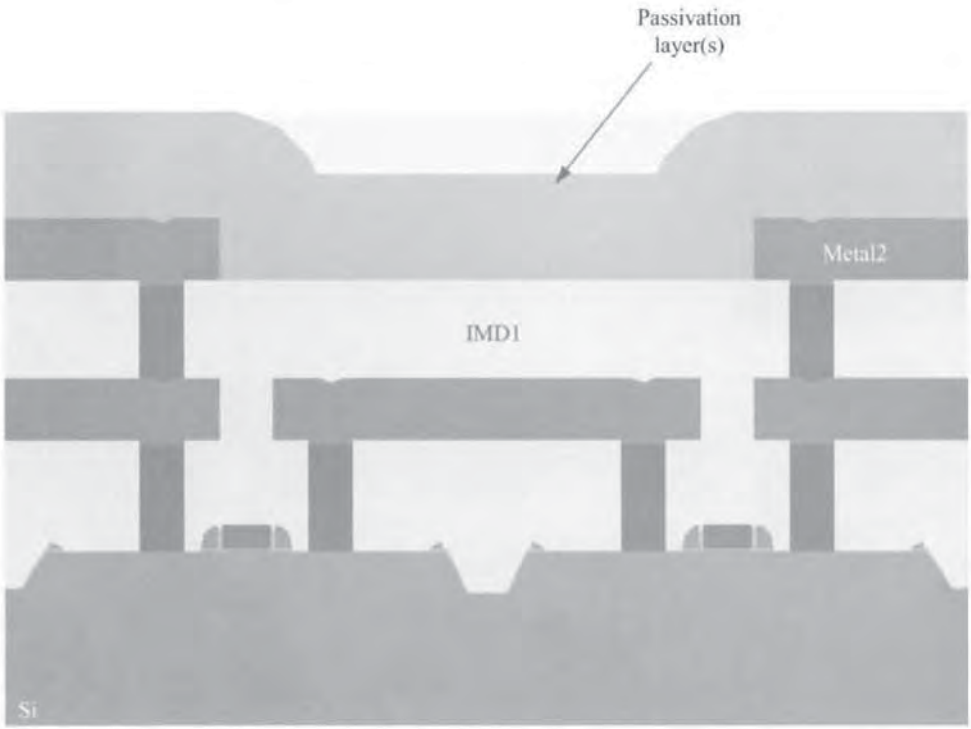



**Figure 7.78** Deposition of final

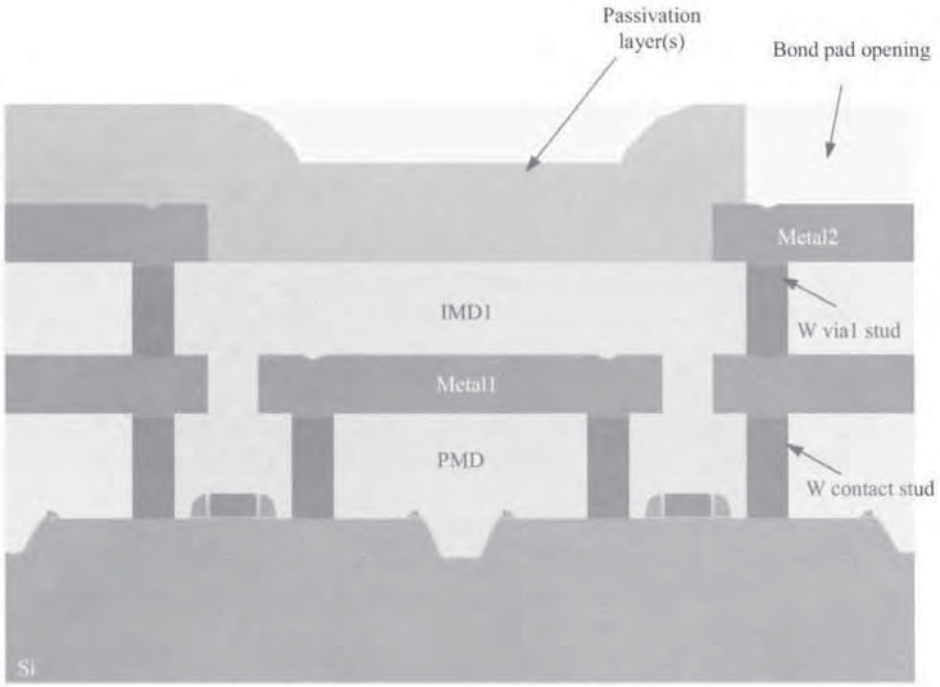



**Figure 7.79** Bond pad definition using photolithography and dry etch of passivation.

IPR2023-00697
Theta EX2019

In these cases, the die are marked, often with an ink dot, to indicate a nonfunctional circuit. Since the processing costs of a wafer are fixed, higher yield equates to higher profit.

*Die Separation*

Prior to separating the individual die, the backs of the wafers are often thinned using a lapping process similar to the CMP process discussed in Sec. 7.1. This thinning is often required for specific types of packages. Furthermore, thinning can aid in improving the removal of heat from the CMOS circuits.

Next, the die are separated from the wafers using a dicing saw comprised of a diamond-coated blade. The cutting paths are aligned to the *streets* or *scribe lines* on the wafer. Great care is given to minimize damage to the die during this mechanical separation. Along with the inked die, die that are observed to be damaged from the dicing are discarded. Obviously, the separation process can reduce the yield.

*Packaging*

The good die are now attached to a header in the appropriate package type. The die attach can be accomplished by either eutectic or epoxy attachment. Next, the bond pads are wired to the leads of the package. Common wire bonding techniques include thermocompression, thermosonic, and ultrasonic bonding. At this point, the packaging is completed by a wide range of different processes, greatly dependent on the type of package in which the IC resides. For instance, in plastic dual in-line packages (DIPs), a process similar to injection molding is performed to form a relatively inexpensive package. If ceramic packaging is required, then the attached, wire bonded die will reside in a cavity that is sealed by a metal lid. In general, plastic (or epoxy) packages are inexpensive, but do not provide a hermetic seal. On the contrary, ceramic packages are more costly, but do provide a hermetic seal. For information on other packaging schemes, the reader is referred to the list of additional reading at the end of the chapter. Finally, it should be noted that the packaging process can add to the overall yield loss.

*Final Test and Burn-In*

Once packaged, the CMOS parts are tested for final functionality and performance. When this is completed, it is common for the parts to go through a *burn-in* step. Here they are operated at extreme temperatures and voltages to weed out infant failures. Additional yield loss can be observed.

## 7.4 Summary

In this chapter the fundamental unit processes required in the manufacture of CMOS integrated circuits were introduced. These unit processes include thermal oxidation, solid state diffusion, ion implantation, photolithography, wet chemical etching, dry (plasma) etching, chemical mechanical polishing, physical vapor deposition, and chemical vapor deposition. Additionally, a brief overview of substrate preparation was given. With this foundation, a representative, deep-submicron CMOS process flow was provided and the significant issues in both the FEOL and BEOL integration were discussed. Finally, an overview of the backend processes was presented including wafer probe, die separation, packaging, and final test and burn-in.

IPR2023-00697
Theta EX2019

## ADDITIONAL READING

[1] S. A. Campbell, *Fabrication Engineering at the Micro- and Nanoscale*, 3rd ed., Oxford University Press, 2008. ISBN 978-0195320176

[2] M. J. Madou, *Fundamentals of Microfabrication: The Science of Miniaturization*, 2nd ed., CRC Publisher, 2002. ISBN 978-0849308260

[3] R. C. Jaeger, *Introduction to Microelectronic Fabrication*, 2nd ed., volume 5 of the Modular Series on Solid State Devices, Prentice-Hall Publishers, 2002. ISBN 0-20-144494-1

[4] S. A. Campbell, *The Science and Engineering of Micrelectronic Fabrication*, 2nd ed., Oxford University Press, 2001. ISBN 0-19-513605-5

[5] J. D. Plummer, M. D. Deal, and P. B. Griffin, *Silicon VLSI Technology, Fundamentals, Practice, and Modeling*, Prentice-Hall Publishers, 2000. ISBN 978-0130850379

IPR2023-00697
Theta EX2019

Chapter

8

# Electrical Noise: An Overview

The word "noise," in everyday use, is any wanted or unwanted sound that can be heard. Electrical noise has a significantly different meaning. Electrical noise is a current or voltage signal that is *unwanted* in an electrical circuit. Real signals are the sum of this unwanted noise and the desired signals. Types of noise include 1) noise resulting from the discrete and random movement of charge in a wire or device (which we'll call inherent circuit noise, e.g., thermal noise, shot noise, or flicker noise), 2) quantization noise (resulting from the finite digital word size when changing an analog signal into a digital signal), and 3) coupled noise (resulting from the signals of adjacent circuits feeding into each other and interfering). We focus the discussion in this chapter on inherent noise. Quantization and coupled noise are discussed in other parts of the book, e.g., Ch. 28.

Electrical noise is introduced early in our study of circuits in this book because it is often the limiting factor in a system's performance. For example, the distance a radio receiver can pick up a transmitted signal increases as the noise performance of the receiver improves. When studying noise, the units, terminology, and procedures used in a noise analysis can present a barrier to understanding what circuit element limits the noise performance (that is, how the circuit's noise performance can be improved). To obviate this barrier, we'll take an unusual approach: we'll use a few simple signals and measurements to illustrate some of the noise analysis procedures.

## 8.1 Signals

In this section we briefly review some fundamentals related to electrical signals.

### 8.1.1 Power and Energy

Figure 8.1 shows a simple sinewave. At the risk of stating the obvious, the average value of this signal is zero. Does this mean that if this signal is applied across a resistor zero average power is dissipated? Of course not (or else many people wouldn't be able to cook food or heat their homes). In order to characterize the *average* amount of power dissipated by a resistor when this sinusoidal signal is applied across it, Fig. 8.2, we can begin by writing the instantaneous power dissipated by the resistor (the power dissipated by the resistor at a specific time $t$) as

IPR2023-00697
Theta EX2019

$$P_{inst}(t) = \frac{v^2(t)}{R} = \frac{V_P^2 \sin^2(2\pi \cdot \frac{t}{T})}{R} \tag{8.1}$$

Summing this power over time results in

$$Energy = \int_0^{\infty} \frac{V_P^2 \sin^2(2\pi \cdot \frac{t}{T})}{R} \cdot dt \tag{8.2}$$

with units of $W \cdot s$ or Joules (energy). Clearly, summing the instantaneous power doesn't result in the average power dissipated by the resistor but rather the energy the sinewave source supplies to the resistor. Because we didn't place an upper bound on the integration in Eq. (8.2), the energy supplied tends to infinity. (A resistor dissipating power over an infinite period of time is being driven by a source with an infinite amount of energy.)

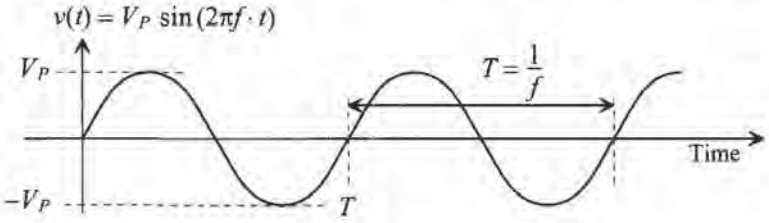



**Figure 8.1** A sinewave.

Examining Fig. 8.1 for a moment, we realize that because the sinewave is periodic, we need only estimate the average power over one cycle (say from time zero to $T$) to estimate the average power supplied to a resistor by the entire waveform. We can do this by summing the instantaneous power for one cycle (which results in the energy supplied by the sinewave source during this cycle) and then dividing by the cycle time $T$. This can be written as

$$\text{Average power dissipated} = \frac{1}{T}\int_0^T \frac{V_P^2 \sin^2(2\pi \cdot \frac{t}{T})}{R} \cdot dt \tag{8.3}$$

noting that the resistor's value can be moved outside of the integration. This result, with units of watts, is the mean (average) squared voltage of the sine wave signal dropped across the resistor divided by the resistor's value. For the sine wave in Figs. 8.1 and 8.2, we can write

$$\text{Mean-squared voltage} = \overline{v^2} = \frac{1}{T}\int_0^T V_P^2 \sin^2(2\pi \cdot \frac{t}{T}) \cdot dt = \frac{V_P^2}{2} \tag{8.4}$$

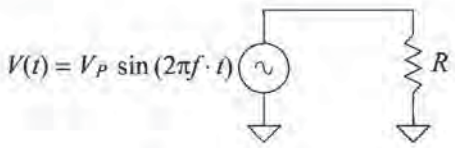



**Figure 8.2** A sinewave driving a resistor.

IPR2023-00697
Theta EX2019

with units of volts squared. Equation (8.3) simplifies to the mean-squared voltage divided by $R$, i.e., $\overline{v^2}/R$, or, for a sinusoid $V_P^2/2R$. This should be compared to the power dissipated by a resistor with a DC voltage across it, i.e., $V_{DC}^2/R$. When $V_P^2/2 = V_{DC}^2$, the power dissipation is the same. If we take the square root of the mean-squared voltage (the result is called the root-mean-squared, RMS, voltage) or

$$\text{RMS voltage of a sinewave, } \sqrt{\overline{v^2}} = V_{RMS} = \sqrt{\frac{1}{T}\int_0^T V_P^2 \sin^2(2\pi f \cdot t) \cdot dt} = \frac{V_P}{\sqrt{2}} \quad (8.5)$$

then we can say that when $V_{RMS} = V_{DC}$, the power dissipated by the resistor with either a sinusoidal or DC source is the same.

*Comments*

Power is heat. If a circuit is battery-powered, then the less heat it generates the longer the battery life. Batteries are often characterized by their capacity to supply power (the potential energy they hold with units of Joules, watthours, or if the battery voltage is known, mA·hours). A battery can also be characterized by its ability to supply a given amount of power in a period of time (kinetic energy).

For example, two 1.5 V batteries may be rated with the same capacity, i.e., potential energies (say 1 W·hr, 3600 J, or 667 mA·hr) but one may only be capable of supplying a maximum current of 1 mA, while the other may supply a maximum of 10 mA. In a one-hour period, and at maximum current draw, the first battery can supply a power of 1.5 mW to a circuit, while the second battery can supply up to 15 mW. The energy (kinetic) supplied at these maximums in one hour is then 5.40 J (the battery supplying 1.5 mW for one hour) and 54 J (the battery supplying 15 mW for one hour). Question: Which battery will discharge first? Answer: The one supplying 10 mA of current will discharge first because both started with the same capacity. It will discharge in 66.7 hours.

When a voltage signal containing many sinusoids (and perhaps DC) is applied across a resistor, we add the power from each sinusoid (and DC) to obtain the total power dissipated by the resistor. In other words, we sum the mean-squared voltages from each sinewave (the mean-squared voltage of a DC source, $V_{DC}$, is $V_{DC}$) and divide by the resistor's value to obtain the total power dissipated. Taking the square-root of the sum of the mean-squared voltages, we obtain the RMS (root OF THE mean OF THE square) value of the signal applied to the resistor. This is *important* because we will regularly sum mean-squared voltages when performing noise analysis. (We regularly add the power [mean-squared value] contributed from each noise source to a circuit.) We *never* sum RMS voltages.

### 8.1.2 Power Spectral Density

Figure 8.3a shows spectral plots of the sinewave in Fig. 8.1. Notice how we use either a single line or a dot to indicate spectral content at a frequency of $f$. Figure 8.3b shows the spectrum of a signal containing two sinewaves and a DC component (with arbitrary amplitudes) using lines and dots. In Fig. 8.3c we show a signal where the spectrum is occupied with sinusoids from DC to some maximum $f_{max}$. Even though we show a constant amplitude (a continuous spectral density from DC to $f_{max}$), it is understood that we are representing a spectrum with spectral components that are, or may be, changing.

IPR2023-00697
Theta EX2019

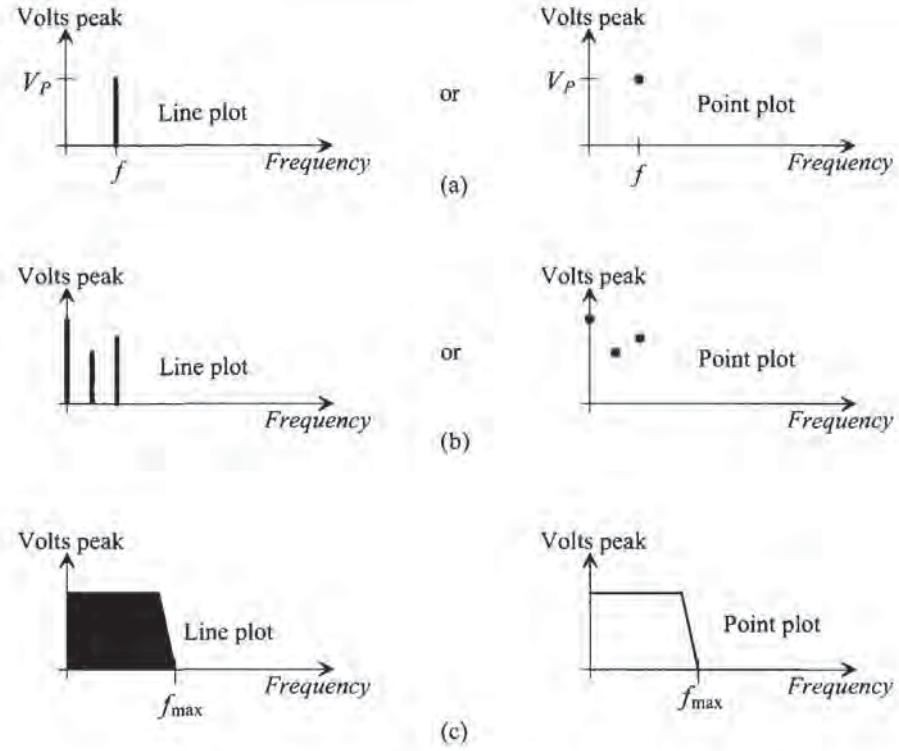



**Figure 8.3** Spectrums of various signals (see text).

*Spectrum Analyzers*

A spectrum analyzer (SA) is an instrument for evaluating the spectral content of a signal. The SA outputs plots similar to those seen in Fig. 8.3 (point-by-point). A block diagram of an SA is seen in Fig. 8.4. The input of the SA is multiplied by a sinusoid to frequency shift the input signal. After the multiplication, the bandpass filter limits the range of frequencies to a bandwidth of $f_{res}$ (the resolution bandwidth of the measurement). The power in the output signal of the bandpass filter is then measured. The counting index, $n$, varies the measurement from a start frequency, $f_{start}$, to a stop frequency, $f_{stop}$. This range of frequencies makes up the x-axis in our spectral plots (as seen in Fig. 8.3). The power measured at each corresponding point sets the y-axis values.

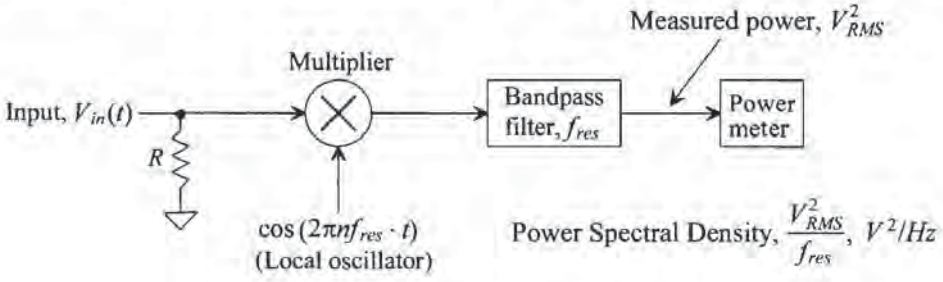



**Figure 8.4** Block diagram of a spectrum analyzer.

IPR2023-00697
Theta EX2019

Let's make some assumptions and give an example to illustrate how the SA operates. Let's assume that we want to look at the spectrum of a signal from DC ($f = 0 = f_{start}$) to 10 kHz ($= f_{stop}$) with a resolution of 100 Hz ($= f_{res}$). Further let's assume that the signal we want to look at is a 1 V peak amplitude sinewave at 4.05 kHz in series with 1 V DC, that is, $V_{in}(t) = 1 + \sin 2\pi \cdot 4.05\,kHz \cdot t$ V. To begin, we calculate the range of the counting index, $n$,

$$f_{stop} = 10\,kHz = n_{\max} \cdot f_{res} \rightarrow n_{\max} = 100 \qquad (8.6)$$

The counting index will vary from 0 to 100. In other words, the frequency of the cosine signal in Fig. 8.4 will vary from 0 to 10 kHz in 100 Hz steps. Our plot will have 101 points.

With our input signal applied, the power at the first point, $n = 0$ (or for the x-axis $f = 0$), is measured. The cosine term applied to the multiplier is 1. The entire input signal is applied to the bandpass filter. In a general SA, a bandpass filter is used. In this example, we use a lowpass filter that passes DC to < 100 Hz. The signal applied to the power meter is $(1\text{ V})^2$. We are ready to plot the first point in the spectral analysis. For this first point at $f = 0$ (and $n = 0$), the y-axis units can be peak voltages (= 1 V), mean-squared (= 1/2 $V^2$), RMS ($= 1/\sqrt{2}$ V $= V_{RMS}$), power ($= 1/2R = V_{RMS}^2/R$ watts where the $R$ is the input resistance of the SA seen in Fig. 8.4), power spectral density (PSD) ($= 1/[2Rf_{res}]$ with units of W/Hz or Joules[1]), or voltage spectral density ($= V_{RMS}/\sqrt{f_{res}} = 1/\sqrt{2 \cdot 100}$ with units of $V/\sqrt{Hz}$ ).

Often we'll use a PSD with units of $V^2/Hz$. This is the result of eliminating the resistance from the calculation **so that the PSD is** $V_{RMS}^2/f_{res}$ ($= 1/2f_{res} = 5 \times 10^{-3}$ $V^2$/Hz for this first point). We use the PSD in noise analysis (and elsewhere) because, for a continuous spectrum (something that we don't have in this example but that is common for noise signals), increasing $f_{res}$ increases the power we measure ($V_{RMS}^2$), resulting in a constant number when we take the ratio $V_{RMS}^2/f_{res}$. (The PSD of a continuous spectrum doesn't change with measurement resolution.) Unfortunately, when a single sinewave is measured, increasing $f_{res}$ decreases the sinewave's amplitude PSD amplitude, as seen in Fig. 8.5.

Continuing on with our measurement when $n = 1$ and the multiplying (local oscillator) frequency is 100 Hz, the output of the multiplier, in Fig. 8.4, is

$$\cos(2\pi \cdot 100 \cdot t) \cdot (1 + \sin[2\pi \cdot 4.05\,kHz \cdot t]) \text{ volts} \qquad (8.7)$$

or knowing $\cos A \cdot \sin B = \frac{1}{2}(\sin[B - A] + \sin[A + B])$, we get

$$\cos(2\pi \cdot 100t) + \tfrac{1}{2}(\sin[2\pi(3.95k)t] + \sin[2\pi(4.15k)t]) \text{ volts} \qquad (8.8)$$

This multiplier output signal (remembering $n = 1$) contains three sinusoids with frequencies of 100 Hz, 3.95 kHz, and 4.15 kHz. Passing this signal through the filter, which allows frequencies from DC to < 100 Hz through, results in zero measured power. This assumes an ideal filter that doesn't pass 100 Hz (see the comment at the end of this

[1] From the units, W/Hz = W·s = J, we can tell that this is the amount of power dissipated by $R$ (for the frequency range of sine waves that pass through the bandpass filter) in one second (the energy used). This shouldn't be confused with energy spectral density (ESD) which has units of $V^2$/Hz/s (or $V^2$), for transient signals (for a finite amount of time, i.e., one second).

IPR2023-00697
Theta EX2019

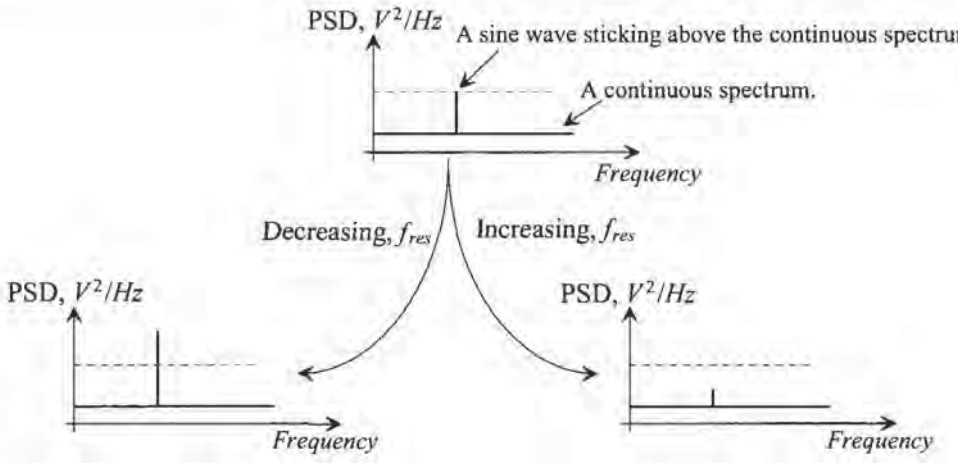



**Figure 8.5** Changing the resolution bandwidth doesn't affect the PSD amplitude of the continuous spectrum but does affect the amplitude of the sine wave sticking up above the continuous spectrum.

section). In fact, it's easy to show that we get zero measured signal until $nf_{res} = 4\,kHz$. When the local oscillator frequency is 4 kHz, the output of the multiplier is

$$\cos(2\pi \cdot 4kt) + \tfrac{1}{2}(\sin[2\pi(50)t] + \sin[2\pi(8.05k)t]) \text{ volts} \qquad (8.9a)$$

that is, a signal containing power at frequencies of 50 Hz, 4 kHz, and 8.05 kHz. Again, the filter passes the signal content in the range of $0 \le f < 100\,Hz$ to the power meter. What comes out of the filter is a sinewave at 50 Hz with a peak amplitude of 0.5 V. The next measured point occurs when $nf_{res} = 4.1\,kHz$. The signal applied to the filter (the output of the multiplier) is

$$\cos(2\pi \cdot 4.1kt) + \tfrac{1}{2}(\sin[2\pi(-50)t] + \sin[2\pi(8.15k)t]) \text{ volts} \qquad (8.9b)$$

The *negative frequency* signal present here can be thought of as a phase-shifted positive frequency signal. Because the sine function is an odd function, $\sin(-2\pi f \cdot t) = -\sin(2\pi f \cdot t) = \sin(2\pi f \cdot t \pm \pi)$, there is still spectral content at 50 Hz. The signal applied to the meter is, again, a 50 Hz sinewave with an amplitude of 0.5 V. The total (measured) signal amplitude from the 4.05 kHz component of the input is the sum of the measured signal amplitudes when $nf_{res}$ is 4 kHz and 4.1 kHz (that is, a sum of 1 V). When plotting points, the amplitudes at measured adjacent points are summed. For example, we can plot a point at 4.05 kHz with an amplitude of 1 V by summing the measured signals when $nf_{res}$ is 4 kHz and 4.1 kHz. We can plot points at 3.95 and 4.15 kHz with amplitudes of 0.5 V because at 3.9 kHz and 4.2 kHz ($= nf_{res}$), we have zero measured signal.

Looking at the spectrums of signals using a discrete Fourier transform (DFT) of analog waveforms is analogous to using an SA, to look at spectrums. For more detailed information on using both a DFT and an SA the interested reader is referred to the book *CMOS Mixed-Signal Circuit Design*. In this book additional information and discussions are given on these topics.

IPR2023-00697
Theta EX2019

## 8.2 Circuit Noise

The electrical noise coming out of a circuit can be measured using a SA, as seen in Fig. 8.6. The circuit under test (CUT) is connected to a spectrum analyzer with no source signal applied. If the noise coming out of a CUT is lower than the noise floor of the SA, a low-noise amplifier (LNA) is inserted between the CUT and SA to help with the noise measurement (the measured noise is then increased by the LNA's gain). As mentioned in the previous section, we generally plot PSD versus frequency to ultimately characterize the power in a signal or the signal's RMS value.

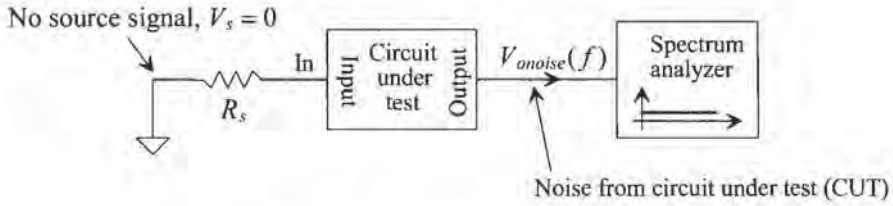



**Figure 8.6** Circuit used for noise measurement.

### 8.2.1 Calculating and Modeling Circuit Noise

Figure 8.7 shows a time domain representation of a noise signal. If this noise signal's PSD is called $V^2_{noise}(f)$ (units, $V^2/Hz$), we can determine its RMS value using

$$V_{RMS} = \sqrt{\int_{f_L}^{f_H} V^2_{noise}(f) \cdot df} \quad \text{Volts} \tag{8.10}$$

where $f_L$ and $f_H$ are the lowest and highest frequencies of interest. If the noise PSD is flat (often called a "white noise spectrum" after white light that contains spectral content at all visible wavelengths), then this equation reduces to

$$V_{RMS} = \sqrt{(f_H - f_L) \cdot V^2_{noise}(f)} = \sqrt{B} \cdot \sqrt{V^2_{noise}(f)} \tag{8.11}$$

where the bandwidth of the measurement, $B$, is $f_H - f_L$.

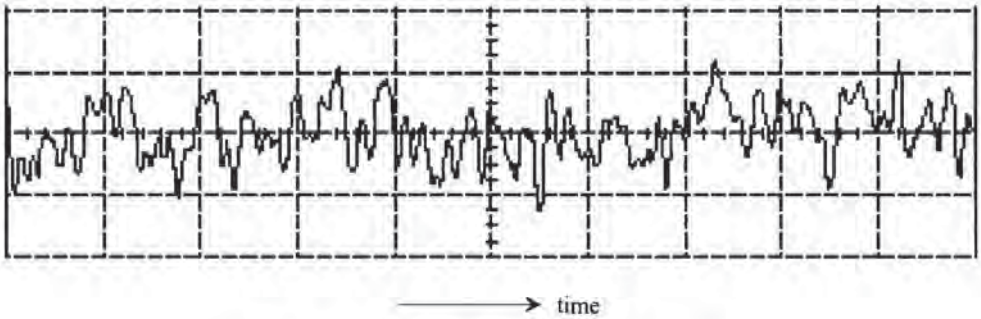



**Figure 8.7** Time domain view of circuit noise.

IPR2023-00697
Theta EX2019

*Input-Referred Noise I*

Noise is measured on the output of a circuit. It can, however, be referred back to the input of the circuit for comparison with an input signal. This input-referred noise isn't really present on the input of the CUT. This is especially true for CMOS circuits where the input to a CUT may be the polysilicon gate of a MOSFET. While a direct tunneling gate current noise may be present it's generally much less important than the average (DC) tunneling gate current and usually a very small contributor to the measured output noise.

Consider the amplifier models seen in Fig. 8.8. Figure 8.8a shows the measured output noise PSD. We can calculate the input-referred PSD, Fig. 8.8b, by simply dividing the output PSD by the gain, $A$, of the amplifier squared (or divide the voltage spectral density, = $\sqrt{PSD}$ , by the gain).

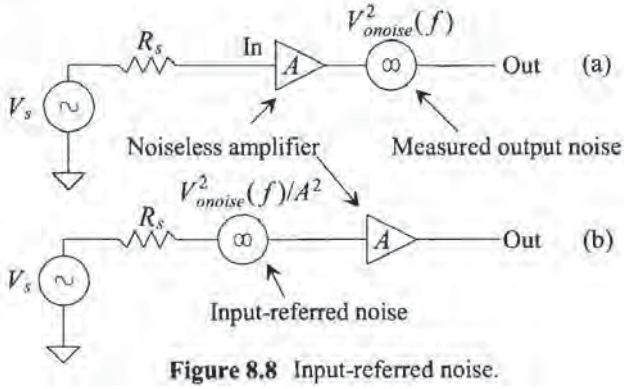



**Figure 8.8** Input-referred noise.

**Example 8.1**

Suppose the measured noise voltage spectral density on the output of an amplifier is a white spectrum from DC to 100 MHz ($= B = f_H - f_L$) and has a value of 10 $nV/\sqrt{Hz}$ . Estimate the amplifier's input-referred noise if its gain is 100.

The amplifier's output noise PSD is

$$V^2_{onoise}(f) = \left(10\ nV/\sqrt{Hz}\right)^2 = 100 \times 10^{-18}\ V^2/Hz$$

The RMS value of this spectral density is

$$V_{onoise,RMS} = \sqrt{\int_0^{100MHz} 100 \times 10^{-18} df} = 100\ \mu V$$

Referring this noise back to the input of the amplifier results in

$$V_{inoise,RMS} = \frac{V_{onoise,RMS}}{A} = 1\ \mu V \qquad (8.12)$$

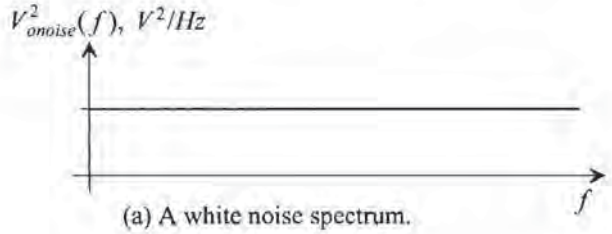
■

*Noise Equivalent Bandwidth*

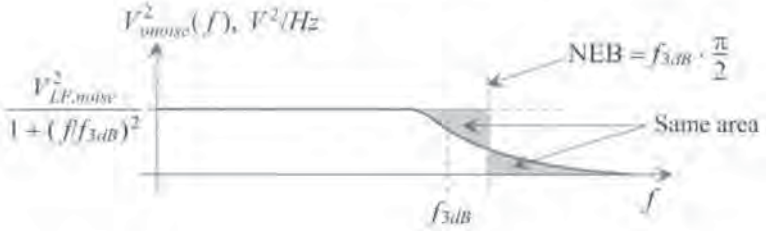
In the previous example, the bandwidth of the noise calculation was given. However, we may wonder, in a real circuit, how we determine $f_L$ and $f_H$. In the ideal situation, we would

IPR2023-00697
Theta EX2019

use an infinite bandwidth when calculating the noise. Figure 8.9a shows a white noise spectrum. If we were to calculate the RMS value of the noise voltage from this spectrum, over an infinite bandwidth, we would end up with an infinite RMS noise voltage. In real circuits the signals, and noise, are bandlimited (their spectral content goes to zero at some frequencies). This bandlimiting can be the result of intentionally added or parasitic capacitances present in the circuit. Figure 8.9b shows a noise spectrum if the CUT shows a simple single-pole roll-off. In this case, our noise is no longer white at high frequencies but rolls off above the 3-dB frequency of the circuit, $f_{3dB}$.




(a) A white noise spectrum.




(b) White noise rolling off with amplifier bandwidth.

**Figure 8.9** Noise PSDs with (a) a white spectrum and (b) a spectrum that rolls-off with circuit bandwidth (assuming a dominant pole circuit).

If we assume that a CUT has a single-pole roll-off, as seen in Fig. 8.9b, and the low-frequency noise of the CUT is white (again as seen in Fig. 8.9b), then we can calculate the RMS output noise using

$$V^2_{onoise,RMS} = \int_0^\infty \overbrace{\frac{V^2_{LF,noise}}{\left(\sqrt{1+(f/f_{3dB})^2}\right)^2}}^{V^2_{onoise}(f)} df \tag{8.13}$$

If we use

$$\int \frac{du}{a^2+u^2} = \frac{1}{a}\tan^{-1}\frac{u}{a} + C \tag{8.14}$$

then

$$V^2_{onoise,RMS} = V^2_{LF,noise} \cdot f_{3dB} \cdot [\tan^{-1} 2\pi f/f_{3dB}]_0^\infty$$

$$= V^2_{LF,noise} \cdot \overbrace{f_{3dB} \cdot \frac{\pi}{2}}^{NEB} \tag{8.15}$$

IPR2023-00697
Theta EX2019

where *NEB* is the noise-equivalent bandwidth. Rewriting Eq. (8.11), we get

$$V_{onoise,RMS} = \sqrt{NEB} \cdot \sqrt{V^2_{LF,noise}} \tag{8.16}$$

Of course this assumes $V^2_{LF,noise}$ is constant until the frequency approaches $f_{3dB}$.

**Example 8.2**

Repeat Ex. 8.1 if the amplifier's output noise power spectral density starts to roll off at 1 MHz as seen in Fig. 8.10.

We begin by using Eq. (8.16) directly, that is,

$$V_{onoise,RMS} = \sqrt{f_{3dB} \cdot \frac{\pi}{2}} \cdot \sqrt{V^2_{LF,noise}} \tag{8.17}$$

which evaluates to 12.5 μV. The gain of the amplifier is 100 so $V_{inoise,RMS}$ = 125 nV. ■

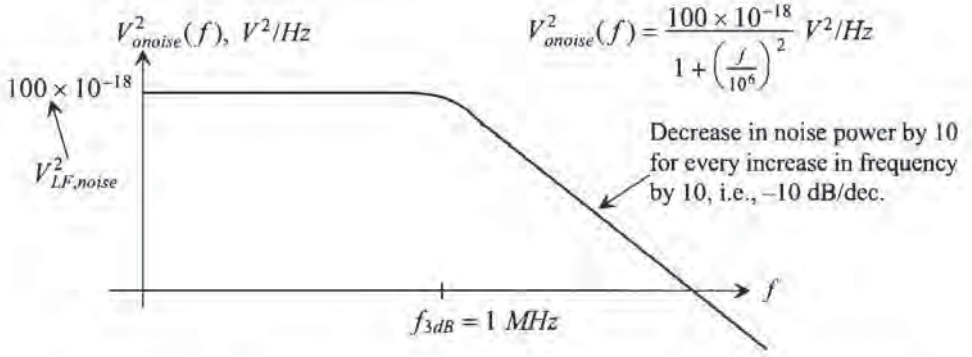



**Figure 8.10** Measured output noise spectrum for the CUT discussed in Ex. 8.2.

In this example, we assumed the voltage gain of the amplifier was 100 independent of frequency when we calculated the input-referred RMS noise voltage. In practice, the frequency response of the noise may follow (be correlated with) the amplifier's frequency response. If this is the case, we can write the frequency response of the amplifier in Ex. 8.2 as

$$A(f) = \frac{V_{out}}{V_{in}} = \frac{A_{DC}}{1 + j\frac{f}{f_{3dB}}} = \frac{100}{1 + j\frac{f}{f_{3dB}}} \tag{8.18}$$

where $f_{3dB}$ = 1 MHz. We can determine the input-referred noise PSD using

$$V^2_{inoise}(f) = \frac{V^2_{onoise}(f)}{|A(f)|^2} = \frac{V^2_{LF,noise}}{1 + (f/f_{3dB})^2} \cdot \frac{1 + (f/f_{3dB})^2}{A^2_{DC}} = \frac{V^2_{LF,noise}}{A^2_{DC}} \tag{8.19}$$

showing the input-referred PSD of the output noise is a flat spectrum from DC to infinity! The problem with this is that if, to determine the RMS input-referred noise, we integrate this PSD over an infinite bandwidth, we get an infinite RMS voltage. We must remember that the input-referred noise is used to model (with obvious limitations) the circuit's output noise (again, you can't measure input-referred noise because it isn't really there!).

IPR2023-00697
Theta EX2019

Input-referred noise is only used to get an idea of how a circuit will corrupt an input signal. This latter point is important because two amplifiers with identical bandwidths may have the same output noise PSD but different gains. The amplifier with the larger gain will have a smaller input-referred noise and thus result in an output signal with a better signal-to-noise ratio. Knowing that the output noise is bandlimited and that the input-referred RMS voltage should reflect this, we can write

$$V_{inoise,RMS} = \frac{\sqrt{NEB \cdot V_{LF,noise}^2}}{A_{DC}} = \frac{V_{onoise,RMS}}{A_{DC}} \qquad (8.20)$$

which is, of course, what we used in Ex. 8.2.

*Input-Referred Noise in Cascaded Amplifiers*

Figure 8.11a shows a cascade of noisy amplifiers with corresponding input-referred noise sources. The output noise power in (a) can be written as

$$V_{onoise,RMS}^2 = (A_1A_2A_3)^2 V_{inoise,RMS1}^2 + (A_2A_3)^2 V_{inoise,RMS2}^2 + A_3^2 V_{inoise,RMS3}^2 \quad (8.21)$$

again remembering that we add the noise powers from each amplifier. The input-referred noise in (b) is given by

$$V_{inoise,RMS}^2 = V_{inoise,RMS1}^2 + V_{inoise,RMS2}^2/A_1^2 + V_{inoise,RMS3}^2/(A_1A_2)^2 \qquad (8.22)$$

The key point to notice here is that the noise of the first amplifier has the largest effect on the noise performance of the amplifier chain. For good noise performance, the design of the first stage is critical.

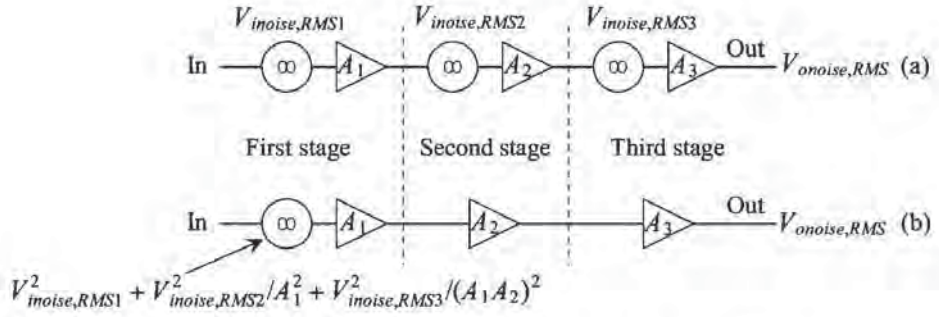



**Figure 8.11** Noise performance of a cascade of amplifiers.

**Example 8.3**

Comment on the limitations of Eqs. (8.21) and (8.22) for calculating noise.

Measured output noise usually includes the thermal noise of the source resistance. If the effective source resistance changes when we cascade the amplifiers, the value calculated for input-referred noise, $V_{inoise,RMS}$, will also change.

A perhaps more important concern is the change in the bandwidth of the noise. Cascading amplifiers results in a reduction in the circuit's bandwidth. The point of Eqs. (8.21) and (8.22) is still valid, that is, that the first stage's output noise and

IPR2023-00697
Theta EX2019

gain (equivalent to saying "input-referred noise") are critical for overall low-noise performance. However, to accurately determine the input-referred noise, it is best to measure the noise on the output of the cascade and then refer it back to the cascade's input. At the risk of stating the obvious, the gain of the cascaded amplifiers is determined by applying a small sinewave signal to the cascade's input at a frequency that falls within the amplifier's passband (not too high or too low). Taking the ratio of the cascade's output sinewave amplitude to the input sinewave amplitude is the gain. The input-referred noise is then the output RMS noise divided by the gain of the overall cascade. ■

*Calculating $V_{onoise,RMS}$ from a Spectrum: A Summary*

Before leaving this section, let's show, Fig. 8.12, a summary of how the output RMS noise voltage is calculated from a noise spectrum.

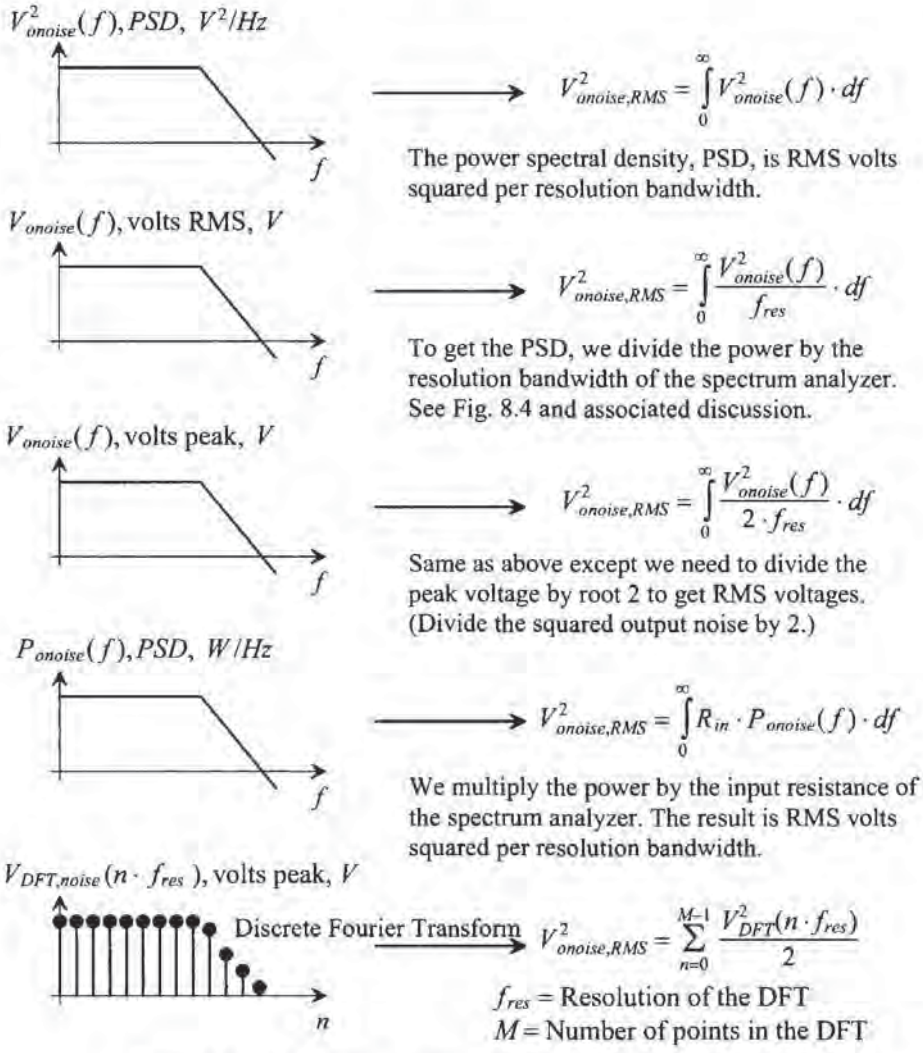



**Figure 8.12** Calculating RMS output noise from a noise spectrum.

IPR2023-00697
Theta EX2019

### 8.2.2 Thermal Noise

Noise in a resistor is primarily the result of random motion of electrons due to thermal effects. This type of noise is termed *thermal noise*, or Johnson noise, after J. B. Johnson who is credited for first observing the effect. Thermal noise in a resistor can be characterized by a PSD of

$$V_R^2(f) = 4kTR \text{ with units of V}^2\text{/Hz} \tag{8.23}$$

(noting thermal noise is white and independent of frequency) where

$k$ = Boltzmann's constant = 13.8 x $10^{-24}$ Watt·sec/° K (or J/° K)

$T$ = temperature in °K

$R$ = resistance in Ω

**Example 8.4**

Verify that the thermal noise PSD given by Eq. (8.23) does indeed have units of $V^2$/Hz. The term $kT$ has units of Joules (or Watt·sec) and can be thought of as thermal energy. The units for Joules can be written as

$$kT, \text{ units} = \text{Joules} = \text{Watt·sec} = \text{volts·amps·sec} = \text{volts·columbs}$$

knowing amps = columbs/sec. The units of resistance, $R$, can be written as

$$R, \text{ units of } \Omega = \frac{\text{volts}}{\text{amps}} = \frac{\text{volts}}{\text{columbs/sec}} = \frac{\text{volts}}{\text{columbs} \cdot \text{Hz}}$$

The units for thermal noise PSD, $4kTR$, are then $V^2$/Hz. ■

Figure 8.13 shows how the thermal noise from a resistor is modeled and added to a circuit. The output noise PSD in (a) is $(4kT/R) \cdot R^2$ (we multiply by $R^2$). Heat (power) is absorbed by the resistor when it is not at 0° K. The heat causes lattice vibrations in the resistive material and thus randomizes the motion of electrons moving in or traveling through the resistor. The net current that flows out of the resistor due to heat is zero but,

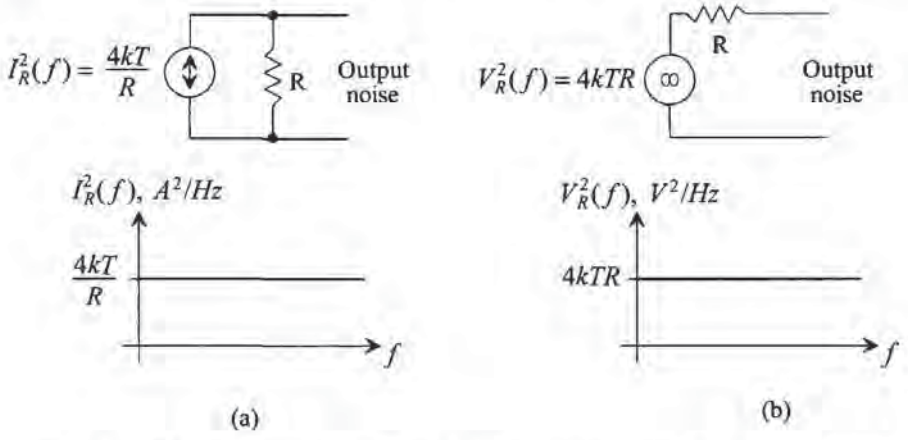



**Figure 8.13** Circuit representations and their corresponding PSDs for thermal noise.

IPR2023-00697
Theta EX2019

over a finite period of time, the current crossing the terminal of a resistor may have a net nonzero value (either into or out of the resistor). In other words, electrons move back and forth from the resistor to the metal wire with an overall net zero charge transfer but, over a short finite period of time, a net nonzero current flow is possible.

The term $kT$, at 300 °K, is $4.14 \times 10^{-21}$ Joules. For a one-second time frame, again at 300 °K, the resistor dissipates $4.14 \times 10^{-21}$ Watts (absorbs this much heat from a heat source). For one hour, the energy supplied to keep the resistor heated to 300 °K is $(3600\text{ s}) \cdot (4.14 \times 10^{-21}\text{ watts}) = 14.9 \times 10^{-18}$ Joules.

**Example 8.5**

For the circuit shown in Fig. 8.14, determine the RMS output and input-referred noise over a bandwidth from DC to 1 kHz. Verify your answer with SPICE.

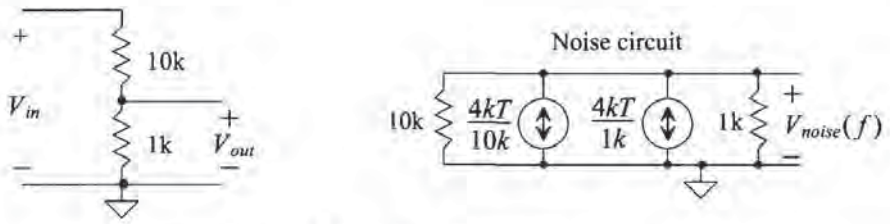



**Figure 8.14** Circuit used in Ex. 8.5.

The PSD of the noise currents are given by (again noting, as seen in Fig. 8.6, that we apply zero volts to the input when calculating output noise PSD, that is, we ground the input)

$$I_{10k}^2(f) = \frac{4 \cdot (13.8 \times 10^{-24}) \cdot (300)}{10,000} = 1.66 \times 10^{-24}\,\frac{A^2}{Hz} \rightarrow I_{10k}(f) = 1.29 \times 10^{-12}\frac{A}{\sqrt{Hz}}$$

$$I_{1k}^2(f) = 16.66 \times 10^{-24}\,\frac{A^2}{Hz} \rightarrow I_{1k}(f) = 4.1 \times 10^{-12}\frac{A}{\sqrt{Hz}}$$

The output thermal noise PSD due to the 10k resistor is

$$I_{10k}^2(f) \cdot \left[\frac{1k \cdot 10k}{1k + 10k}\right]^2 \text{ units } V^2/Hz$$

In order to avoid analyzing circuits in a different way, (for example, $V^2 = I^2R^2$, $i_d^2 = g_m^2 v_{gs}^2$, $V^2 = I^2(R_1 + R_2)^2$, etc.), we can use the voltage or current spectral densities (square-root of PSD) of the noise when doing circuit noise calculations

$$V_{10k}(f) = I_{10k}(f) \cdot \frac{1k \cdot 10k}{1k + 10k} \text{ units } V/\sqrt{Hz}$$

which evaluates to 1.2 $nV/\sqrt{Hz}$. The output noise voltage-spectral density $V_{1k}(f)$ from the 1k resistor is 3.7 $nV/\sqrt{Hz}$. The total output noise PSD, $V_{onoise}^2(f)$, is the sum of $V_{10k}^2(f)$ and $V_{1k}^2(f)$. Again, we sum the PSDs (or power) but **not** the voltage-spectral densities (or the RMS voltages).

IPR2023-00697
Theta EX2019

The mean-squared output noise voltage over a bandwidth of 1 kHz is

$$V^2_{onoise,RMS} = \int_{f_L}^{f_H} V^2_{onoise}(f) \cdot df = \int_{0}^{1kHz} \left( V^2_{10k}(f) + V^2_{1k}(f) \right) \cdot df = 15.1 \times 10^{-15}\ V^2$$

The RMS output noise voltage is

$$V_{onoise,RMS} = 123\ nV$$

The input-referred noise voltage is

$$V_{inoise,RMS} = 123nV \cdot \frac{10k + 1k}{1k} = 1.35\ \mu V$$

SPICE simulation gives an output mean squared noise of $1.5053 \times 10^{-14}$ $V^2$ (= 123 nV RMS) and an input-referred noise referenced to an input voltage of 1 V of $1.8215 \times 10^{-12}$ (= 1.35 μV RMS). The SPICE netlist is shown below.

```
*** Example 8.5 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
print all
.endc

.noise   v(2,0)  Vin dec  100 1 1k
R1       1 2 10k
R2       2 0 1k
Vin      1 0     dc 0 ac 1
.print noise all
.end
```

The SPICE output is seen below

```
TEMP=27 deg C
Noise analysis ... 100%
inoise_total = 1.821510e-12
onoise_total = 1.505380e-14
```

A bandwidth of 1 to 1,000 Hz was used in this simulation rather than DC to 1 kHz. Also note that the reference supply, Vin in the netlist, was a voltage, so the units of the SPICE output are $V^2$. The input AC source has a magnitude of 1 V, so the input noise SPICE gives is divided by 1 V squared. If we had used a 1 mV AC supply for Vin, then the inoise_total above would be $1.8215 \times 10^{-6}$ $V^2$.

Here, in SPICE, we used a single input-referred voltage source to model the input-referred noise, see Eq. (8.22) and the associated discussion. In SPICE we always refer the input noise to a source (voltage or current) and not to a node (like we do in our analysis). This means that the input-referred noise in SPICE is always a single source in series (voltage input) or parallel (current input) with the input of the circuit. ■

IPR2023-00697
Theta EX2019

**Example 8.6**
Estimate $V_{onoise,RMS}$ for the circuit seen in Fig. 8.15. Verify the answer with SPICE.

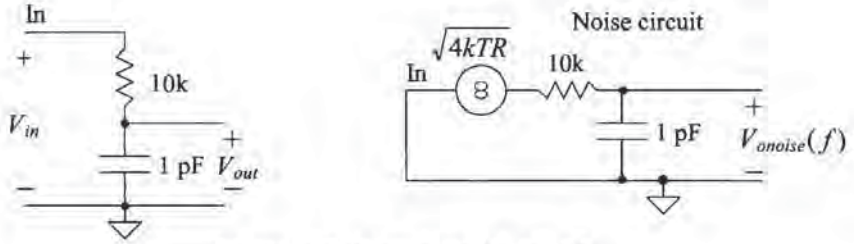



**Figure 8.15** Circuit used in Ex. 8.6.

The only element in this circuit that generates noise is the resistor. It generates thermal noise. The resistor's noise voltage spectral density is $\sqrt{4kTR}$ . The output noise spectral density is then

$$V_{onoise}(f) = \sqrt{4kTR}\,\frac{1/j\omega C}{1/j\omega C + R} = \frac{\sqrt{4kTR}}{1 + j\frac{f}{f_{3dB}}} \text{ units, } V/\sqrt{Hz}$$

or

$$V^2_{onoise}(f) = \frac{4kTR}{|1 + j\frac{f}{f_{3dB}}|^2} = \frac{4kTR}{\left(\sqrt{1 + (f/f_{3dB})^2}\right)^2} = \frac{4kTR}{1 + (f/f_{3dB})^2} \text{ units, } V^2/Hz$$

where $f_{3dB} = 1/2\pi RC$. This single-pole roll-off was the reason we discussed noise-equivalent bandwidth (NEB) earlier, Ex. 8.2. Using Eq. (8.17), the output RMS noise voltage is

$$V_{onoise,RMS} = \sqrt{\frac{1}{2\pi RC} \cdot \frac{\pi}{2} \cdot 4kTR} = \sqrt{\frac{kT}{C}} \qquad (8.24)$$

The RMS value of the thermal noise in this circuit is limited by the size of the capacitor and independent of the size of the resistor. This result is very useful. This "Kay Tee over Cee" noise is frequently used to determine the size of the capacitors used in filtering or sampling circuits.

The SPICE netlist and output are seen below.

```
*** Example 8.6 CMOS: Circuit Design, Layout, and Simulation ***

.noise   v(Vout,0)      Vin     dec     100     1       1G
R1       Vin            Vout    10k
C1       Vout    0      1p
Vin      Vin     0      dc      0       ac      1
.print noise all
.end

TEMP=27 deg C
Noise analysis ... 100%
```

IPR2023-00697
Theta EX2019

```
inoise_total = 1.695223e-07
onoise_total = 4.101864e-09
```

Using Eq. (8.24) at 300 °K, we get $V^2_{onoise,RMS} = 4.14 \times 10^{-9}\ V^2$. This is close to what SPICE gives above for onoise_total (close but not exact; we stopped the simulation at 1 GHz not infinity). The output RMS noise is 64 μV (keeping in mind that the peak-to-peak value of the thermal noise will be larger than this).

The input-referred noise, inoise_total, in this simulation example is somewhat meaningless. Changing the stop frequency in the simulation from 1 GHz to 10 GHz has little effect on the output RMS noise but does cause the input-referred noise to increase. As indicated in Eq. (8.19) and the associated discussion, the input-referred noise spectral density increases indefinitely. Integrating this spectral density from DC to infinity results in an infinite RMS input-referred voltage. Since the low-frequency gain here is one, we would specify $V_{inoise,RMS} = V_{onoise,RMS}$. ■

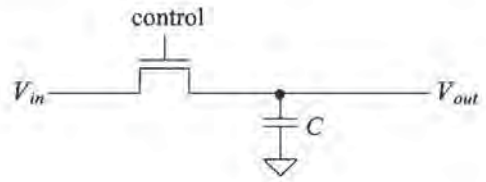



**Figure 8.16** A sample-and-hold circuit.

For an example of the usefulness of Eq. (8.24), consider the sample-and-hold circuit seen in Fig. 8.16. When the MOSFET gate is driven to *VDD,* the MOSFET behaves like a resistor and permits $V_{in}$ to charge the capacitor. When the MOSFET is off, the capacitor remains charged and the MOSFET behaves like an open. The voltage on the capacitor doesn't change again until the MOSFET turns back on. We can think of the sampling operation as sampling both the input signal and the *kT/C* noise onto the capacitor. Table 8.1 provides a comparison between capacitor sizes and *kT/C* noise.

**Table 8.1** Capacitor size and corresponding *kT/C* noise at 300 °K.

| **Capacitor size, pF** | $\sqrt{kT/C}$, μV |
|---|---|
| 0.01 | 640 |
| 0.1 | 200 |
| 1 | 64 |
| 10 | 20 |
| 100 | 6.4 |

IPR2023-00697
Theta EX2019

### 8.2.3 Signal-to-Noise Ratio

Signal-to-noise ratio, *SNR*, can be defined in general terms by

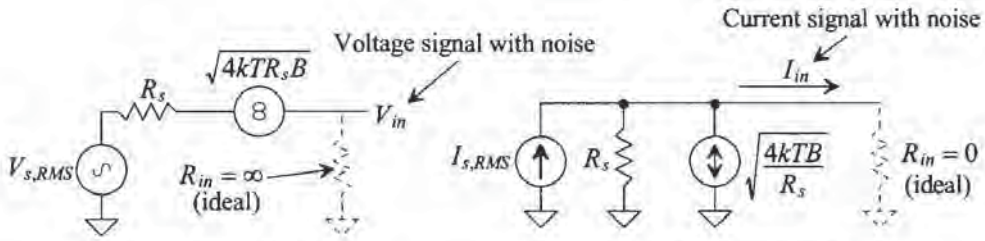
$$SNR = \frac{\text{desired signal power, } P_s}{\text{undesired signal power (noise), } P_{noise}} \tag{8.25}$$

*SNR* can be specified using dB as

$$SNR = 10\log\frac{P_s}{P_{noise}} \tag{8.26}$$

If the power is normalized to a 1-Ω load (e.g., $P_s = V_{s,RMS}^2/(1\ \Omega)$), we can write

$$SNR = 10\log\frac{P_s}{P_{noise}} = 20\log\frac{\sqrt{P_s}}{\sqrt{P_{noise}}} = 20\log\frac{V_{s,RMS}}{V_{noise,RMS}} \tag{8.27}$$




(a) Input voltage source (b) Input current source

**Figure 8.17** Calculating input SNR.

Figure 8.17 shows two equivalent input sources with associated thermal noise models. With the ideal $R_{in}$ the *SNR* associated with these circuits is (noting $V_{s,RMS}^2 = I_{s,RMS}^2 \cdot R_s^2$) an open (voltage input) or a short (current input)

$$SNR_{in} = \frac{V_{s,RMS}^2}{4kTR_sB} = \frac{I_{s,RMS}^2}{4kTB/R_s} \tag{8.28}$$

In Fig. 8.17a, $V_{in}$ is dropped across an infinite $R_{in}$ (an open). In Fig. 8.17b, $I_{in}$ drives zero $R_{in}$ (a short). In the practical case, $R_{in}$ is finite and nonzero. For Fig. 8.17a, the input signal and noise are attenuated by the voltage divider formed between $R_s$ and $R_{in}$

$$SNR_{in} = \frac{V_{s,RMS}^2 \cdot \left[\frac{R_{in}}{R_{in}+R_s}\right]^2}{4kTR_sB \cdot \left[\frac{R_{in}}{R_{in}+R_s}\right]^2} = \frac{V_{s,RMS}^2}{4kTR_sB} \tag{8.29}$$

which has no effect on the $SNR_{in}$. We can also show that there is no change in $SNR_{in}$ if $R_{in}$ is nonzero in Fig. 8.17b.

Examine the amplifier model with noise seen in Fig. 8.18. The output noise, $V_{onoise,RMS}$, includes the thermal noise contributions from $R_s$. We measured the output noise with the source connected to the amplifier (see Figs. 8.6 and 8.8). The *SNR* associated with the output of the amplifier (where the noise is due to the amplifier and the thermal noise from $R_s$) is then

IPR2023-00697
Theta EX2019

$$SNR_{out} = \frac{V_{s,RMS}^2 \cdot \left[\frac{R_{in}}{R_{in}+R_s}\right]^2 \cdot A^2}{V_{onoise,RMS}^2} \tag{8.30}$$

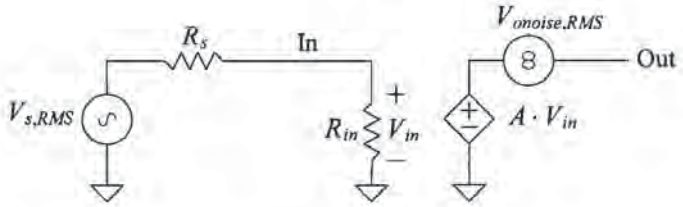



**Figure 8.18** Calculating output SNR.

*Input-Referred Noise II*

When we discussed input-referred noise in Fig. 8.8, we tacitly assumed that the input resistance of the amplifier $R_{in}$ was infinite. The input-referred noise could then be modeled with a single voltage source in series with the input signal source $V_s$. Keeping in mind that we are measuring $V_{onoise,RMS}$ at a fixed value of $R_s$, Fig. 8.19a, we can show that if the amplifier's input resistance $R_{in}$ is not infinite, this single input-referred voltage source can still manage to model the amplifier's output noise. Looking at Fig. 8.19b, we can write

$$A \cdot V_{inoise,RMS} \cdot \frac{R_{in}}{R_{in}+R_s} = V_{onoise,RMS} \tag{8.31}$$

If the input resistance $R_{in}$ goes to infinity (common for low-frequency CMOS amplifiers) or $R_s$ is zero (a voltage source is connected to the amplifier's input), a single input-referred noise voltage is ideal for modeling the output noise. (And we'll use this simpler model in most of our noise discussions in this book.)

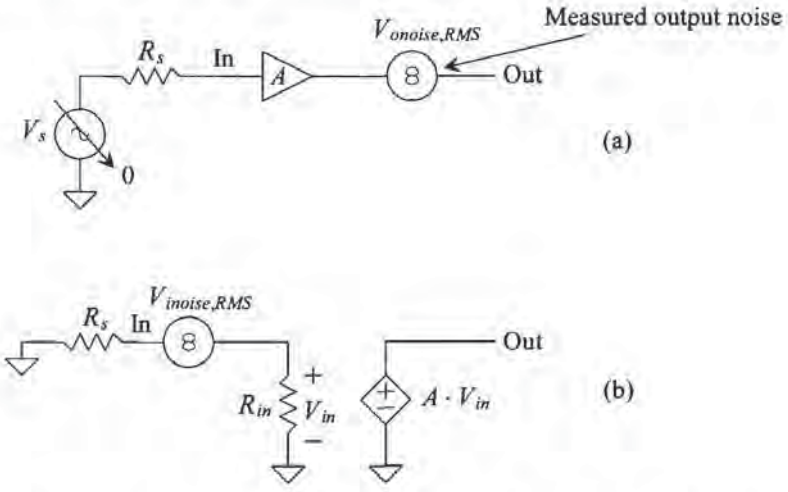


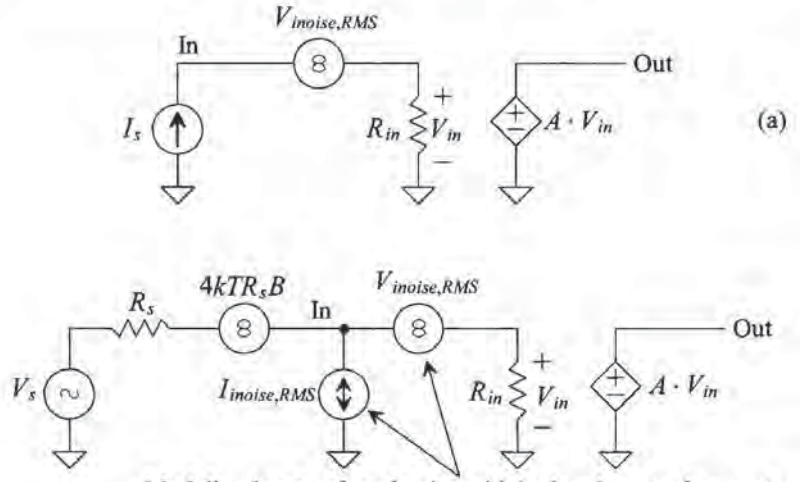

**Figure 8.19** Modeling measured output noise with a single input-referred source.

IPR2023-00697
Theta EX2019

Using a single voltage source to model input-referred noise does have some limitations though. For example, what happens if our input signal is a current (meaning $R_s$ is infinite), as seen in Fig. 8.20a? In this case, the input-referred noise voltage is irrelevant. The input voltage is independent of $V_{inoise,RMS}$ (the output of the circuit is free of noise). To avoid this situation and to make the input-referred noise sources *independent of* $R_s$, we can use the noise model seen in Fig. 8.20b. The output noise using this model is

$$V_{onoise,RMS}^2 = 4kTR_sB \cdot \left(\frac{AR_{in}}{R_s + R_{in}}\right)^2 + I_{inoise,RMS}^2 \cdot \left(\frac{AR_sR_{in}}{R_s + R_{in}}\right)^2 + V_{inoise,RMS}^2 \cdot \left(\frac{AR_{in}}{R_s + R_{in}}\right)^2 \tag{8.32}$$

noting that we sum the power from each source to get the total output noise power. It's important to note that our input-referred sources no longer depend on the thermal noise contributions from the source resistance (we've added it separately). Looking at Eq. (8.32), we see that if $R_s$ is zero or infinite, the thermal noise contributions from $R_s$ to the output noise are zero. We should also see that if $R_{in}$ is large (or $R_s$ is small), $V_{inoise,RMS}$ alone is sufficient to model the input-referred noise. Similarly, if $R_s$ is large (or $R_{in}$ is small), $I_{inoise,RMS}$ alone can be used to model the input-referred noise.




Modeling input-referred noise with both voltage and current noise sources.

**Figure 8.20** Using two input-referred noise sources to model an amplifier's output noise.

**Example 8.7**

Discuss how to determine $V_{inoise,RMS}$ and $I_{inoise,RMS}$.

Looking at Eq. (8.32), we see that if $R_s$ is zero, then $V_{inoise,RMS}$ is sufficient to model the output noise. We can therefore write

$$V_{inoise,RMS} = \frac{V_{onoise,RMS,Rs=0}}{A} \; (V_{onoise,RMS} \text{ measured with } R_s \text{ shorted}) \tag{8.33}$$

IPR2023-00697
Theta EX2019

If $R_s$ is infinite (think of the input as seen in Fig. 8.17b), then $I_{inoise,RMS}$ is sufficient to model the amplifier's output noise

$$I_{inoise,RMS} = \frac{V_{onoise,RMS,Rs=\infty}}{AR_{in}} \; (V_{onoise,RMS} \text{ measured with } R_s \text{ opened}) \tag{8.34}$$

■

*Noise Figure*

The noise figure, *NF*, of an amplifier is given by

$$NF = 10\log\frac{SNR_{in}}{SNR_{out}} = 10\log[\text{noise factor, } F] \tag{8.35}$$

If the amplifier doesn't degrade the *SNR*, then the input and output *SNR*s are the same, the noise factor, $F$, is 1, and the *NF* is 0 dB. Using Eqs. (8.29) and (8.30), we can write the *NF* as

$$NF = 10\log\frac{V^2_{onoise,RMS}}{4kTR_sB\cdot\left[\frac{R_{in}}{R_{in}+R_s}\right]^2\cdot A^2} \tag{8.36}$$

When the measuring bandwidth $B$ is small (common for narrowband amplifiers used in communication circuits), we are measuring the spot *NF* (noting, from Eq. [8.10], the bandwidth used to calculate $V_{onoise,RMS}$ is reduced). If the measuring bandwidth is large (common in general analog CMOS design), we are measuring the average *NF*.

The numerator in Eq. (8.36) is the total output noise power and the denominator is the output noise power due to the source resistance, that is,

$$F = \frac{\text{total output noise power}}{\text{output noise power due to source resistance}} \tag{8.37}$$

If the amplifier is noise-free ($V_{onoise,RMS} = \sqrt{4kTR_sB}\cdot A\cdot R_{in}/[R_{in}+R_s]$, that is, the output contains only the thermal noise from the source resistance), the *NF* is, again, 0 dB. For a low-noise amplifier (LNA), the *NF* may typically range from 0.5 to 5 dB.

Note that if $SNR_{in}$ is infinite, the *NF* is meaningless. Infinite $SNR_{in}$ could occur if we use the impractical case, see Fig. 8.17, of $R_s = 0$ (an ideal voltage source input where, also, ideally $R_{in} = \infty$) or $R_s = \infty$ (an ideal current source input where, also, ideally $R_{in} = 0$).

*An Important Limitation of the Noise Figure*

Looking at Eq. (8.36), we see that if $R_{in}$ is large compared to $R_s$ ($R_{in} >> R_s$), $F$ approaches

$$F = \frac{V^2_{onoise,RMS}}{4kTR_sB\cdot A^2} \tag{8.38}$$

This equation shows why CMOS amplifiers work so well for low-noise amplification with large source resistances. At low frequencies, the gate of the MOSFET is an open. If we take $R_s \to \infty$, then $F$ goes to 1 which is a perfect noiseless amplifier right? After reviewing how $R_s$ affects $SNR_{in}$ in Eq. (8.29), we see that the cost for this good amplifier *NF* is an $SNR_{in}$ that approaches zero. The input voltage signal is so noisy that it makes the amplifier noise irrelevant and thus the *NF* is 0 dB. Note that if $R_{in}$ is infinite, we must use the model seen in Fig. 8.17a. In Fig. 8.17b $I_{in}$ would be zero if $R_{in} = \infty$.

IPR2023-00697
Theta EX2019

**Example 8.8**
Suppose an input signal source has an *SNR* of 60 dB and a source resistance $R_s$. Further, suppose this signal is amplified with an amplifier having an *NF* of 1 dB when driven from a source with a resistance $R_s$. Estimate the *SNR* on the output of the amplifier.

We can begin by rewriting Eq. (8.31)

$$NF = 10\log SNR_{in} - 10\log SNR_{out} \tag{8.39}$$

illustrating the beauty of using the *NF*. That is, $SNR_{out}$ is calculated by subtracting the *NF* from $SNR_{in}$. In this case, the $SNR_{out}$ is 59 dB. ■

**Example 8.9**
Calculate the input-referred noise, *F*, and *SNR*s for the circuit seen in Fig. 8.21.

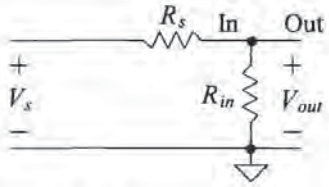



**Figure 8.21** Circuit used in Ex. 8.9.

Let's begin by adding the noise voltage spectral density to the circuit, Fig. 8.22a. The output noise PSD is

$$V^2_{onoise}(f) = 4kTR_s\left[\frac{R_{in}}{R_{in}+R_s}\right]^2 + 4kTR_{in}\left[\frac{R_s}{R_{in}+R_s}\right]^2$$

To determine $V_{onoise,RMS}$, we integrate this PSD over the bandwidth of interest *B* or

$$V^2_{onoise,RMS} = \int_{f_L}^{f_H} V^2_{onoise}(f)\cdot df = 4kTBR_s\left[\frac{R_{in}}{R_{in}+R_s}\right]^2 + 4kTBR_{in}\left[\frac{R_s}{R_{in}+R_s}\right]^2$$

Noting our gain *A* (= $V_{out}/V_{in}$ not $V_{out}/V_s$) is one, we can use the model shown in Fig. 8.22b. To determine the input-referred noise sources, we can use Eq. (8.32) and the results in Ex. 8.7. To determine $V_{inoise,RMS}$, we short the input to ground ($R_s = 0$ in Fig. 8.21 and the equation above), Fig. 8.22c, and equate the circuit output to $V_{onoise,RMS}$. This gives

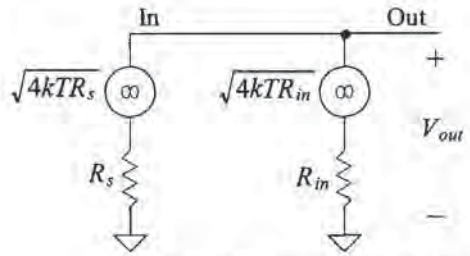
$$V_{onoise,RMS,Rs=0} = V_{inoise,RMS} = 0$$

To determine $I_{inoise,RMS}$, we open the input ($R_s = \infty$), Fig. 8.22d, and equate the circuit's output to $V_{onoise,RMS}$ (from the equation above). This gives

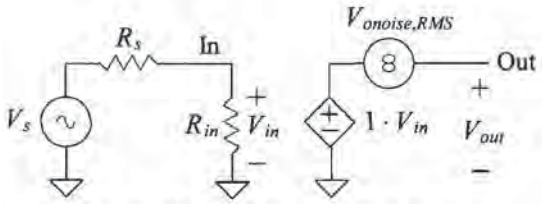
$$R^2_{in}\cdot I^2_{inoise,RMS} = V^2_{onoise,RMS,Rs=\infty} = 4kTBR_{in} \rightarrow I_{inoise,RMS} = \sqrt{\frac{4kTB}{R_{in}}}$$

The input *SNR* is given in Eq. (8.29). The output *SNR*, Fig. 8.22e, is

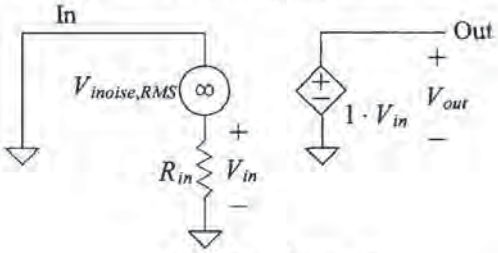
IPR2023-00697
Theta EX2019



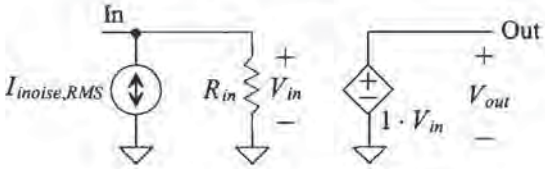

(a) Adding noise sources to the circuit.



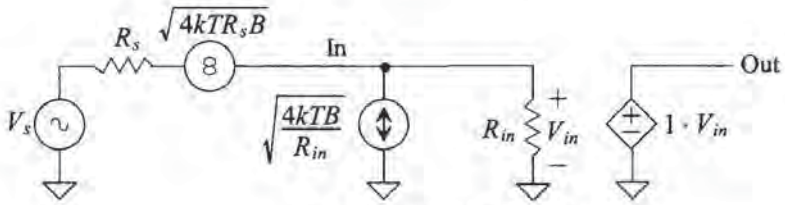

(b) Model showing output noise.




(c) Shorting the input.




(d) Opening the input.




(e) Input-referred noise model.

**Figure 8.22** Noise analysis procedure for Ex. 8.9.

IPR2023-00697
Theta EX2019

$$SNR_{out} = \frac{V_{s,RMS}^2 \cdot \left[\frac{R_{in}}{R_s+R_{in}}\right]^2}{V_{onoise,RMS}^2} = \frac{V_{s,RMS}^2}{4kTB \cdot R_s(1+R_s/R_{in})} \tag{8.40}$$

The noise factor is then

$$F = 1 + \frac{R_s}{R_{in}} \tag{8.41}$$

To minimize the NF, we can decrease $R_s$ or increase $R_{in}$. Decreasing $R_s$ causes $SNR_{in}$ and $SNR_{out}$ to increase, as seen in Eqs. (8.29) and (8.40). At the same time, increasing $R_{in}$ causes $SNR_{out}$ to move towards $SNR_{in}$, Eq. (8.40), resulting in $F$ moving towards 1. ■

*Optimum Source Resistance*

Looking at Eq. (8.36), notice that if $R_s$ approaches a short or an open, the $NF$ gets really large. To determine the optimum source resistance (to minimize $NF$), let's write $F$ using Eqs. (8.32) and (8.36) as

$$F = \frac{4kTR_sB + I_{inoise,RMS}^2 \cdot R_s^2 + V_{inoise,RMS}^2}{4kTR_sB} \tag{8.42}$$

Taking the derivative with respect to $R_s$ and setting the result equal to zero (to determine the minimum), results in

$$I_{inoise,RMS}^2 + 2 \cdot I_{inoise,RMS} \cdot R_s \cdot \frac{\partial I_{inoise,RMS}}{\partial R_s} - \frac{V_{inoise,RMS}^2}{R_s^2} + \frac{2V_{inoise,RMS}}{R_s} \cdot \frac{\partial V_{inoise,RMS}}{\partial R_s} = 0 \tag{8.43}$$

Since the input-referred noise sources do not vary with $R_s$ (the derivatives are zero), we can write the optimum source $R_{s,opt}$ as

$$R_{s,opt} = \frac{V_{inoise,RMS}}{I_{inoise,RMS}} \tag{8.44}$$

From Ex. 8.9 we can use this equation to see that $R_{s,opt} = 0$.

To determine the optimum noise factor $F_{opt}$, we can substitute Eq. (8.44) into Eq. (8.42) to get

$$F_{opt} = 1 + \frac{(V_{inoise,RMS} \cdot I_{inoise,RMS})/2}{kTB} \tag{8.45}$$

The best noise performance is characterized by a ratio of the input-referred noise power of the amplifier, $(V_{inoise,RMS} \cdot I_{inoise,RMS})/2$, to the thermal power (the thermal energy $kT$ in a bandwidth of $B$). This result isn't too useful because it indicates that for good noise performance we need to use an amplifier with low input-referred noise power (which is as profound as saying "to get good noise performance use a low-noise amplifier").

*Simulating Noiseless Resistors*

To verify the $NF$ of a circuit using SPICE, it may be useful to replace one or more resistors with elements that don't generate noise. To simulate a noiseless resistor, a voltage-controlled current source (VCCS) can be used, Fig. 8.23. Current flows from

IPR2023-00697
Theta EX2019

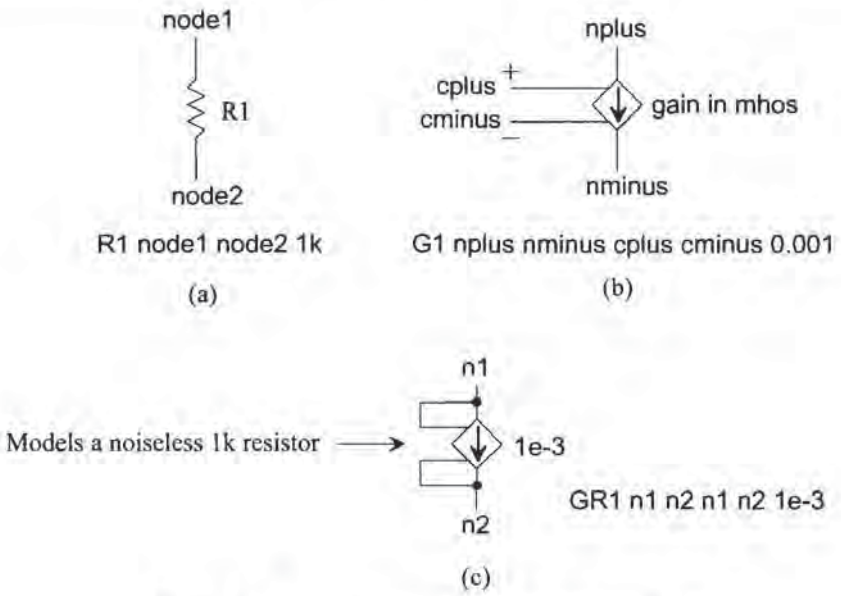



**Figure 8.23** Modeling a noiseless resistor in SPICE.

node1 to node2 in the resistor of (a) while it flows from nplus to nminus in the VCCS in (b). The current in (b) is controlled by the voltage applied to the controlling nodes cplus and cminus. The input to the VCCS is a voltage, and the output is a current (meaning the gain is current/volts or transconductance). To implement a noiseless resistor, we can use the VCCS where the controlling nodes are tied across the VCCS, as seen in (c). The value of the resistor is one over the transconductance of the VCCS.

---

**Example 8.10**

Suppose, in Fig. 8.21, $R_s$ is 10k and $R_{in}$ is 1k. Use SPICE to verify the value of noise factor derived in Ex. 8.9 (that is, Eq. [8.41]).

Using Eq. (8.41), the noise factor is 11 (NF = 10.4 dB). Note that Eq. (8.41) is not dependent on bandwidth. We simulated this circuit already in Ex. 8.5 over a bandwidth of 1 to 1kHz. The result was a $V^2_{onoise,RMS}$ of $1.5053 \times 10^{-14} V^2$. Referring to Eq. (8.37), this is the "total output noise power." To determine the "output noise power due to source resistance," we can replace $R_{in}$ with a noiseless resistor, as seen in Fig. 8.24. The SPICE netlist is given below.

```
*** Example 8.10 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
print all
.endc

.noise   V(Vout,0)     Vs     dec    100    1      1k

Rs       Vs     Vout   10k
Gin      Vout   0      Vout   0      1e-3
```

IPR2023-00697
Theta EX2019

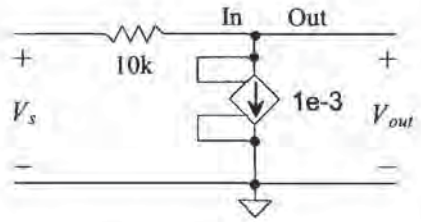



**Figure 8.24** Replacing Rin with a noiseless resistor.

```
Vs      Vs    0      dc     0      ac     1

.print noise all
.end

TEMP=27 deg C
Noise analysis ... 100%
inoise_total = 1.655918e-13
onoise_total = 1.368527e-15
```

The noise factor is calculated from the simulation results as

$$F = \frac{1.5053 \times 10^{-14} V^2}{1.3685 \times 10^{-15} V^2} = 11$$

which is what we calculated using Eq. (8.41). ■

In a general SPICE simulation, we can determine $F$ by eliminating the noise contributed by $R_s$. To do so, we first need to rewrite Eq. (8.37) as

$$F = \frac{\text{total output noise power}}{\text{total output noise power} - \text{total output noise power with } R_s \text{ noiseless}} \tag{8.46}$$

---

**Example 8.11**

Using SPICE and Eq. (8.46), calculate the noise factor of the circuit in Ex. 8.10 (verify, once again, that it is 11).

Again, as determined in Ex. 8.5, the total output noise power for this circuit is $1.5053 \times 10^{-14} V^2$. If we simulate the noise performance of the circuit in Fig. 8.21 with $R_s$ being noiseless, we get

```
TEMP=27 deg C
Noise analysis ... 100%
inoise_total = 1.655918e-12
onoise_total = 1.368527e-14
```

Using Eq. (8.46), we get

$$F = \frac{1.5053 \times 10^{-14}}{1.5053 \times 10^{-14} - 1.3685 \times 10^{-14}} = 11$$

Simulating with a noiseless source resistor is the preferred way to simulate NF with SPICE. ■

IPR2023-00697
Theta EX2019

*Noise Temperature*

Sometimes the term *noise temperature* is used to characterize the noise performance of an amplifier. In Fig. 8.25a we measure an amplifier's output RMS noise, $V_{onoise,RMS}$, with the input source connected but with $V_s = 0$ (the source contributes thermal noise to the output noise PSD). In Fig. 8.25b we refer this noise back to the input of the amplifier (this input-referred voltage includes both the amplifier and source resistance noise). After referring the noise back to the input, we think of the amplifier as being noiseless. In Fig. 8.23c we remove the input-referred source but, to have the same output noise as in (a) or (b), we change the effective temperature of the source's resistor and include its corresponding thermal noise model. By adjusting the temperature of the source resistor in our calculations (called the *noise temperature*, $T_n$), we can get the same total output noise power as in (a) or (b). We can relate the noise factor to the noise temperature by re-writing Eq. (8.36) as

$$F = \frac{\overbrace{4kTR_sB \cdot A^2 \cdot \left(\frac{R_{in}}{R_{in}+R_s}\right)^2}^{\text{Output noise from } Rs} + \overbrace{4kT_nR_sB \cdot A^2 \cdot \left(\frac{R_{in}}{R_{in}+R_s}\right)^2}^{\text{Amplifier's output noise alone}}}{\underbrace{4kTR_sB \cdot A^2 \cdot \left(\frac{R_{in}}{R_{in}+R_s}\right)^2}_{\text{Output noise from } Rs}} = 1 + \frac{T_n}{T} \quad \text{where } T_n \geq 0 \tag{8.47}$$

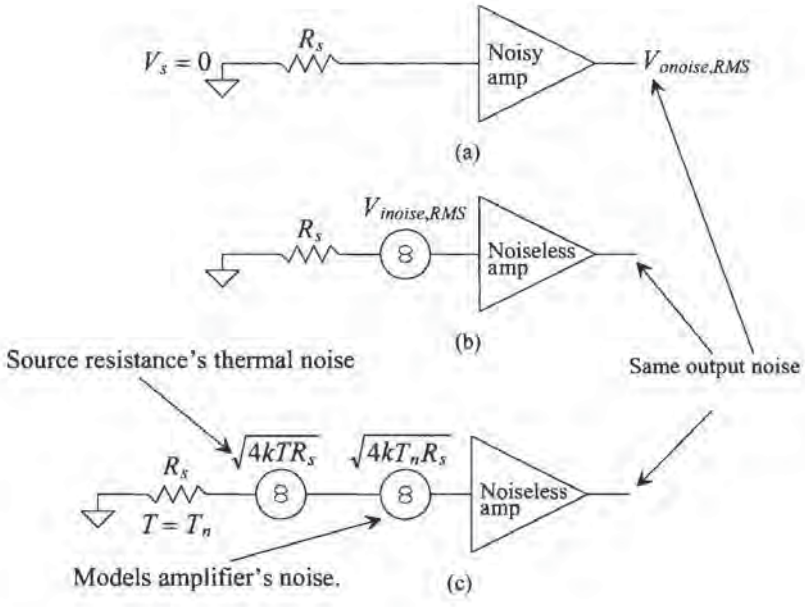



**Figure 8.25** Determining the noise temperature of an amplifier.

IPR2023-00697
Theta EX2019

where $T$ is the temperature, in Kelvin, at which the total output noise was measured in (a). We know that if the amplifier is noiseless, $F = 1$. A noiseless amplifier has $T_n = 0$. Note that we could now relate Eq. (8.47) to Eq. (8.45) to write the optimum noise temperature of an amplifier in terms of $R_{s,opt}$. Again this wouldn't be too useful because it would state that we should use an amplifier with a low input-referred noise power to get a low value of $T_n$ (good amplifier noise performance).

Table 8.2 provides a listing of *NF*s and the corresponding values of $F$ and $T_n$. Note that an *NF*, less than 3 dB indicates that more than half of the output noise is due to the thermal noise of the source resistance.

**Table 8.2** *NF*s and corresponding values of $F$ and $T_n$ (with $T = 300°$ K).

| **Noise figure, *NF*, dB** | **Noise factor, *F*** | **Noise temp, $T_n$, ° K** |
|---|---|---|
| 0 | 1 | 0 |
| 0.5 | 1.12 | 36.61 |
| 0.75 | 1.19 | 56.55 |
| 1 | 1.26 | 77.68 |
| 1.5 | 1.41 | 123.8 |
| 2 | 1.59 | 175.5 |
| 3 | 2 | 298.6 |
| 4 | 2.51 | 453.6 |
| 5 | 3.16 | 648.7 |

*Averaging White Noise*

Before moving on, let's comment on what happens if we average white noise (like thermal noise or shot noise discussed next). If the number of samples averaged is $K$, then the spectral density, and RMS value, of the noise is reduced by $K$. For thermal noise, we would rewrite Eq. (8.23) as

$$V_{RMS}^2 = \frac{4kTRB}{K} \text{ with units of } V^2 \qquad (8.48)$$

Averaging can be thought of as lowpass filtering the random signal. Looking at Eq. (8.48), we see that the RMS value of the noise is reduced by the square root of the number of points averaged. Reviewing Ex. 8.6 ($kT/C$ noise), we see the same result. Increasing $C$ reduces the bandwidth of the thermal noise in the output of the RC circuit. This then causes the RMS value of the output noise to decrease by the root of $C$.

Figure 8.26a shows a noisy voltage signal with no averaging. The peak-to-peak amplitude of this waveform is roughly 5 divisions. In (b) we are averaging 4 of the waveforms in (a). We expect the amplitude to go to $5/\sqrt{4}$ or 2.5 divisions. If 16 averages are used, we should get an amplitude of 1.25 divisions, Fig. 8.26c. Finally, Fig. 8.26d shows how the noise is reduced if 64 waveforms are averaged. We would expect an amplitude of $5/\sqrt{64}$ or 0.625 divisions.

IPR2023-00697
Theta EX2019

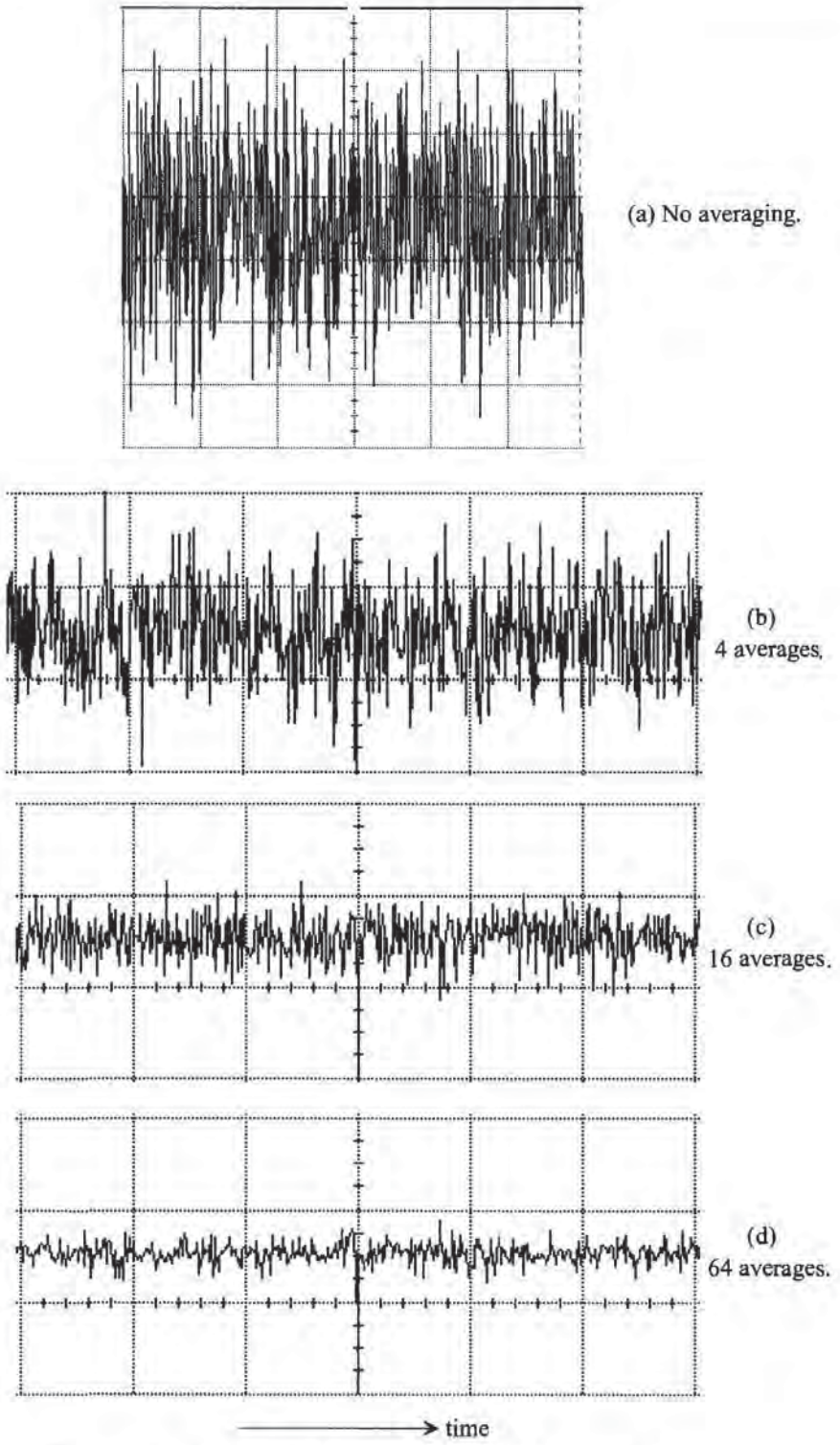



**Figure 8.26** Averaging reduces the amplitude of white noise.

IPR2023-00697
Theta EX2019

### 8.2.4 Shot Noise

Shot noise results from the discrete movement of charge across a potential barrier (e.g., a diode). To understand the origin of shot noise, imagine dumping marbles onto a table. As the marbles move around on the table, some will fall to the floor. The rate the marbles drop to the floor is random even though the rate we dump them onto the table may be constant. When current flows in a diode (when we dump the marbles onto the table), the movement of charge across the depletion region is random (the rate the marbles fall to the floor is random). For example, when a majority carrier, say an electron (a marble on top of the table), moves into the space-charge region (falls off the table), the electric field sweeps it across the junction (gravity causes the marble to fall to the floor). On the p-type side (the floor), where it is now a minority carrier, it diffuses out until it recombines (the marble moves until it stops). Note the name "shot" has to do with using shot from a shotgun shell (little balls of lead or, nowadays for waterfowl, steel used in shotgun shells) in the example above instead of marbles.

The power spectral density of shot noise can be determined empirically and is given by

$$I_{shot}^2(f) = 2qI_{DC} \text{ with units of } A^2/Hz \qquad (8.49)$$

Shot noise is different from thermal noise (although both are modeled with a white noise PSD). Recall that the carriers can randomly move either into or out of the resistive material (with net zero charge transfer) due to thermal noise. Thermal noise is present even without a current flowing in the resistor. For shot noise to be present, we must have both a potential barrier and a current flowing. (There is net charge transfer with an average equal to the current flowing across the barrier.) The movement across the barrier is random and in one direction (the marbles don't hop back up on the table).

Shot noise is not present in long-channel MOSFETs (however, see the discussion in [4]). Shot noise is present in short-channel MOSFETs ($t_{ox} < 20$ nm) because of the gate tunneling current. When a long-channel MOSFET is operating in the saturation region, the entrance of charge into the depletion region, between the channel and the drain, has a discrete nature. However, this discrete movement is the result of thermal variations in the MOSFET's channel resistance and not carriers crossing a potential barrier and being swept up by an electric field.

---

**Example 8.12**

Estimate the output noise in the circuit seen in Fig. 8.27a. Verify your answer with SPICE. Assume that the diode's minority carrier lifetime (see Ch. 2) is 10 ns.

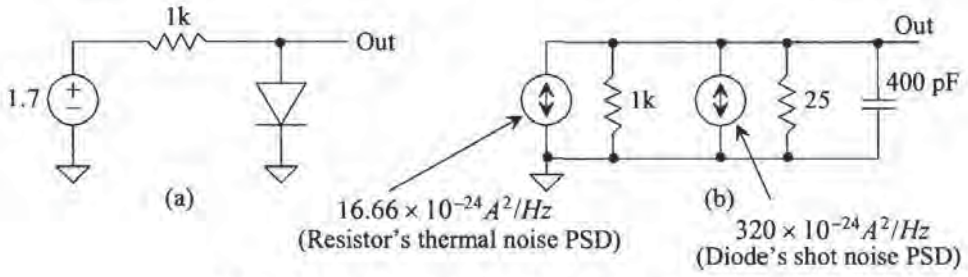



**Figure 8.27** Calculating output noise for Ex. 8.12.

IPR2023-00697
Theta EX2019

Figure 8.27b shows the circuit used for noise analysis. The diode is forward-biased at roughly 0.7 V. One volt is dropped across the 1k resistor and so 1 mA of current flows in the circuit. The diode's small-signal resistance is

$$r_d = \frac{V_T}{I_{DC}} = \frac{kT}{qI_{DC}} \approx 25\ \Omega$$

noting that the small-signal resistance of the diode does not generate thermal noise (it's a model) but that any diode series resistance will. (We set the diode's series resistance to zero in the SPICE model statement for this example.) The diode's storage capacitance is

$$C_S = \frac{I_{DC}}{V_T} \cdot \tau_T = \frac{\tau_T}{r_d} = 400\ pF$$

The diode's shot noise PSD is

$$I^2_{shot}(f) = 2qI_{DC} = 2 \cdot (1.6 \times 10^{-19}) \cdot 1mA = 320 \times 10^{-24} A^2/Hz$$

The resistor's thermal noise PSD was calculated in Ex. 8.5 as $16.66 \times 10^{-24} A^2/Hz$. The circuit output PSD is then the sum of the thermal and the shot noise contributions times the parallel connection of the 1k and 25 ohm resistors or

$$V^2_{onoise}(f) = 336.6 \times 10^{-24} \cdot (25||1k)^2 = 200 \times 10^{-21} V^2/Hz$$

To calculate the output RMS noise voltage, we need to integrate this PSD from DC to infinity as indicated in Fig. 8.12. However, notice that this circuit has a single time constant of $(1k||25) \cdot 400pF$ or approximately 10 ns (the diodes minority carrier lifetime). The noise equivalent bandwidth, *NEB* from Eq. (8.15), is roughly

$$NEB = \frac{1}{2\pi \cdot 10ns} \cdot \frac{\pi}{2} = 25\ MHz$$

The RMS output noise is then

$$V_{onoise,RMS} = \sqrt{25 \times 10^6 \times 200 \times 10^{-21}} = 2.23\ \mu V$$

The SPICE netlist and simulation output are seen below.

```
*** Example 8.12 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
print all
.endc

.noise  V(Vout,0)       Vs      dec     100     1       100G

Vs      Vs      0       dc      1.7     ac      1
Rs      Vs      Vout    1k
D1      Vout    0       Diode

.model Diode D TT=10n Rs=0
.print noise all
.end
```

IPR2023-00697
Theta EX2019

```
TEMP=27 deg C
Noise analysis ... 100%
inoise_total = 3.592225e-05
onoise_total = 5.257612e-12
```

The RMS output noise calculated using SPICE is $\sqrt{5.26 \times 10^{-12}} = 2.28\ \mu V$. Again note that the input-referred noise is meaningless in this example, as seen in Eq. (8.19) and the associated discussion. ■

**Example 8.13**

Estimate the output noise in the circuit seen in Fig. 8.27a if the diode's anode and cathode are swapped. Assume that the diode's zero bias depletion capacitance CJ0 is 25 fF, its built-in potential VJ is 1 V, and its grading coefficient m is 0.5.

The diode is now reverse-biased. The small reverse leakage current that flows in the diode will result in shot noise but it will be tiny compared to the thermal noise generated by the resistor. To calculate the shot noise PSD, we use the reverse leakage current for $I_{DC}$ in Eq. (8.49). Because the diode is reverse-biased, it can be thought of as a capacitor. The value of the diode's depletion capacitance is calculated as

$$C_j = \frac{C_{j0}}{\left(1 + \frac{V_d}{VJ}\right)^m} = \frac{25\,fF}{\sqrt{1 + \frac{1.7}{1}}} = 15.2\,fF$$

We can use the circuit seen in Fig. 8.28 for the noise analysis. After examining this figure for a moment, we see that the output RMS noise is simply, from Ex. 8.6 and Eq. (8.24), $kT/C$ noise. For this example then, $V_{onoise,RMS} = 521\ \mu V$ (verified with SPICE). The fundamental way to reduce the noise is to decrease the bandwidth of the circuit by increasing the capacitance shunting the output. Changing the resistor size has practically no effect on the RMS output noise. ■

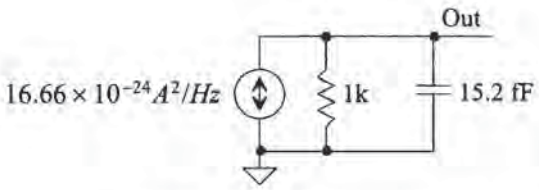



**Figure 8.28** Calculating output noise for Ex. 8.12.

### 8.2.5 Flicker Noise

Flicker noise is a low-frequency noise and is probably the most important, and most misunderstood, noise source in CMOS circuit design. Flicker noise is also known as pink noise (a reddish color present in the lower range of the visible spectrum) or $1/f$ noise (pronounced "one over f" because its PSD, as we shall see, is inversely proportional to frequency). Before going too much further, let's do an example.

IPR2023-00697
Theta EX2019

**Example 8.14**
Estimate the output noise PSD and the RMS value for the integrator circuit seen in Fig. 8.29 (which includes the only noise source, that is, the thermal noise from the resistor). Assume that the op-amp is ideal (noiseless, infinite gain, and infinite bandwidth).

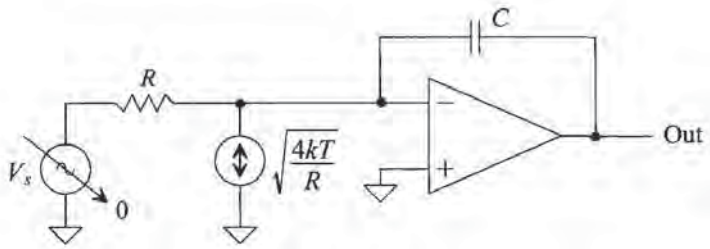



**Figure 8.29** Integrating thermal noise.

The op-amp will keep the inverting input at the same potential as the noninverting input, that is, ground. This means that all of the thermal noise from the resistor will flow through the feedback capacitor (both sides of R are at ground). The integrator's output noise PSD is then

$$V^2_{onoise}(f) = \left|\frac{1}{j\omega C}\right|^2 \cdot \frac{4kT}{R} = \frac{4kT}{R} \cdot \frac{1}{(2\pi C)^2} \cdot \frac{1}{f^2} \text{ units of } V^2/Hz$$

Noise with a $1/f^2$ PSD shape is often called red noise, Fig. 8.30. Looking at this spectrum, we see that the noise PSD becomes infinite as we approach DC. Remember, from our discussion at the beginning of the chapter concerning how a spectrum analyzer operates, that a point representing the PSD at a particular frequency is the measured power, $V^2_{RMS}$, divided by the resolution bandwidth of the spectrum analyzer, $f_{res}$ (that is, $V^2_{RMS}/f_{res}$). To make the low-frequency measurements, $f_{res}$ must decrease, e.g., go from 1 Hz to 0.01 Hz to 0.00001 Hz, etc. A DC signal (or a sinewave for that matter, see Fig. 8.5) results in an infinite PSD point if we take $f_{res} \rightarrow 0$.

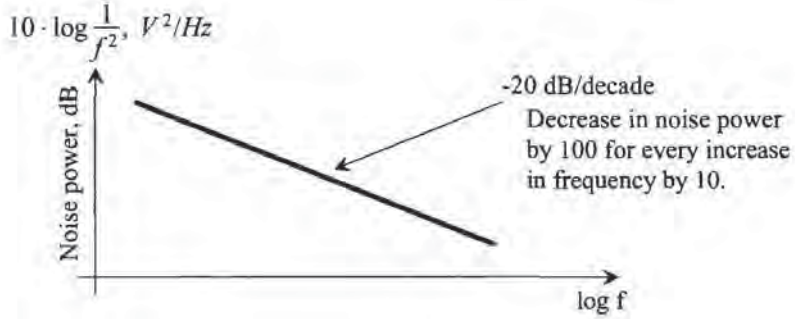



**Figure 8.30** PSD of integrated thermal noise (red noise).

IPR2023-00697
Theta EX2019

Let's use Eq. (8.10) to determine the RMS output noise voltage for this circuit

$$V_{onoise,RMS}^2 = \int_{f_L}^{f_H} V_{onoise}^2(f) \cdot df = \frac{4kT}{R} \cdot \frac{1}{(2\pi C)^2} \cdot \left(\frac{1}{f_L} - \frac{1}{f_H}\right) \tag{8.50}$$

If we take $f_H \to \infty$, this equation becomes

$$V_{onoise,RMS} = \sqrt{\frac{4kT}{R}} \cdot \frac{1}{2\pi C} \cdot \frac{1}{\sqrt{f_L}} \tag{8.51}$$

If we call the length of time we integrate (or measure an input signal since integrators are often used for sensing) $T_{meas}$, then we can get an approximation for lower integration frequency as

$$T_{meas} \approx \frac{1}{f_L} \tag{8.52}$$

or, rewriting Eq. (8.51),

$$V_{onoise,RMS} \approx \sqrt{\frac{4kT}{R}} \cdot \frac{1}{2\pi C} \cdot \sqrt{T_{meas}} \tag{8.53}$$

This result is practically very important. If we average (the same as integrating, which is just summing a variable) a signal with a $1/f^2$ noise spectrum (integrated thermal noise), the RMS output noise voltage actually gets bigger the longer we average (unlike averaging thermal noise, see Eq. [8.48]). A signal containing $1/f^3$ noise (e.g., integrating flicker noise) shows a linear increase in its RMS noise voltage with measurement time. The result is an *SNR* that doesn't get better, and may get worse, by increasing the measurement time. These results pose a limiting factor when imaging in astronomy. The night sky contains this type of noise (flicker, red, etc.). We can make images brighter by exposing our imaging system for longer periods of time but the images don't get clearer. ■

Flicker noise (a name used because when viewed in a light source flickering is observed) is modeled using

$$I_{1/f}^2(f) \text{ or } V_{1/f}^2(f) = \frac{FNN}{f} \text{ units of } A^2/Hz \text{ or } V^2/Hz \tag{8.54}$$

where *FNN* is the flicker noise numerator. The RMS value of a $1/f$ noise signal is

$$V_{noise,RMS}^2 = \int_{f_L}^{f_H} \frac{FNN}{f} \cdot df = FNN \cdot \ln\frac{f_H}{f_L} \text{ units of } V^2 \tag{8.55}$$

If $f_H$ is 100 GHz and $f_L$ is once every $10^{-10}$ Hz (roughly once every 320 years), this equation becomes

$$V_{noise,RMS} = 7 \cdot \sqrt{FNN} \text{ Volts} \tag{8.56}$$

where *FNN* has units of $V^2$. After reviewing Ex. 8.14, we can write

$$V_{noise,RMS} \propto \sqrt{\ln T_{meas}} \tag{8.57}$$

For all intents and purposes, $V_{noise,RMS}$ stabilizes as measurement time increases.

IPR2023-00697
Theta EX2019

**Example 8.15**

The input-referred noise voltage spectral density of the TLC220x (a low-noise CMOS op-amp) is seen in Fig. 8.31. The input-referred noise current is not shown but listed on the data sheet as typically $0.6\,fA/\sqrt{Hz}$ (a tiny number and so it will be neglected in the following example). For the unity follower op-amp seen in Fig. 8.32, estimate the RMS output noise voltage.

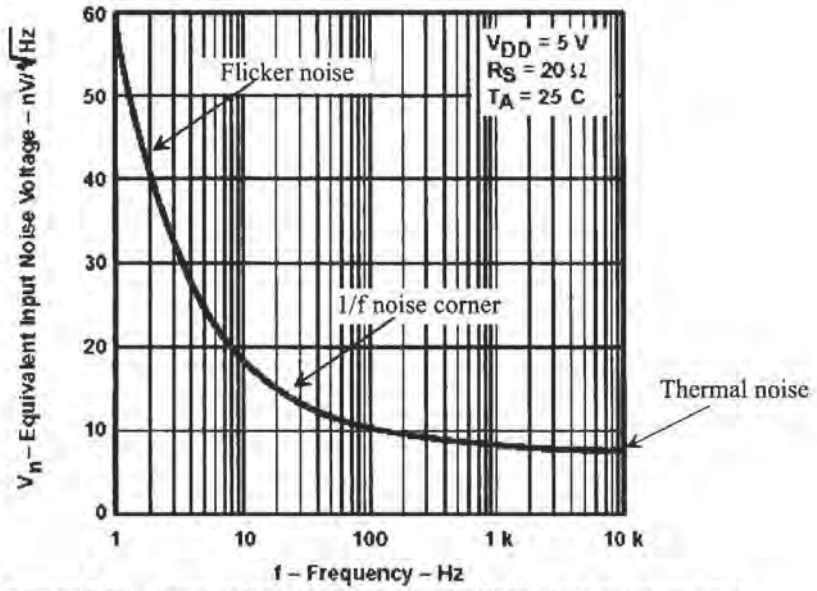



**Figure 8.31** Input-referred noise for the TLC220x low-noise op-amp.

The input-referred thermal noise PSD is roughly $\left(8\,nV/\sqrt{Hz}\right)^2$. The square root of the flicker noise input-referred PSD is

$$V_{inoise,1/f}(f) = \sqrt{\frac{FNN}{f}} = \frac{56\,nV}{\sqrt{Hz}} \text{ by looking at } f = 1 \text{ Hz}$$

then

$$FNN = (56\,nV)^2 = 3.14 \times 10^{-15}\,V^2$$

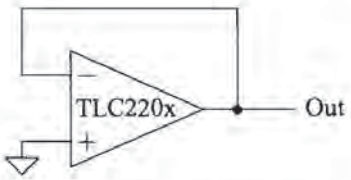



**Figure 8.32** Estimating RMS output noise in a voltage follower.

IPR2023-00697
Theta EX2019

As a quick check we see, in Fig. 8.31, that at 4 Hz the spectral density is roughly 28 nV/$\sqrt{Hz}$ or

$$V_{inoise,1/f}(f) = \sqrt{\frac{FNN}{f}} = \sqrt{\frac{3.14 \times 10^{-15} V^2}{4\ Hz}} = \frac{56\ nV}{\sqrt{4\ Hz}} = 28\ nV/\sqrt{Hz}$$

Note, in Fig. 8.31, that the point where the thermal noise PSD gets comparable to the $1/f$ noise PSD is sometimes called the $1/f$ noise corner. Because the gain of the op-amp is one, we can write the input or output noise PSD as

$$V^2_{onoise}(f) = V^2_{inoise}(f) = \overbrace{\frac{3.14 \times 10^{-15}}{f}}^{V^2_{inoise,1/f}(f),\ \text{flicker noise}} + \overbrace{64 \times 10^{-18}}^{\text{thermal noise}} \text{ with units of } V^2/Hz$$

To determine the output or input-referred RMS noise voltage, we integrate this noise spectrum. The gain-bandwidth product of the op-amp is roughly 2 MHz. This means, because the closed-loop gain of the op-amp is one, its $f_{3dB}$ frequency is 2 MHz. Knowing the $f_{3dB}$ of the circuit, at least for the thermal noise, we can use Eq. (8.15). With the help of Eq. (8.56) for the flicker noise contributions, we can then write

$$V^2_{onoise,RMS} = V^2_{inoise,RMS} = \overbrace{49 \cdot (3.14 \times 10^{-15})}^{\text{using Eq. (8.56)}} + \overbrace{(64 \times 10^{-18}) \cdot (2 \times 10^6) \cdot \frac{\pi}{2}}^{\text{using Eq. (8.15)}}$$

or noting the main contributor here is thermal noise. (The contributions from flicker noise, relative to the thermal noise present, fall off quickly above 10 Hz. Therefore, not using the amplifier's bandwidth when calculating the flicker noise contributions doesn't result in any significant error.) Solving this equation gives

$$V_{onoise,RMS} = V_{inoise,RMS} = 15\,\mu V$$

or roughly 100 μV peak-to-peak if the noise has a Gaussian probability distribution function, Fig. 8.33, (where the standard deviation 1σ is the RMS value of the noise voltage and the peak-to-peak value is roughly 6σ).

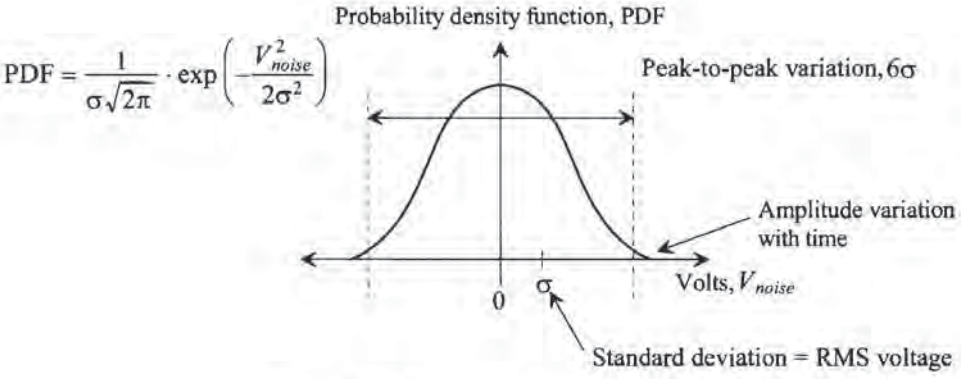



**Figure 8.33** Gaussian probability distribution.

IPR2023-00697
Theta EX2019

Note that this last result is useful for any of the RMS values we've calculated. To get an estimate for the peak-to-peak amplitude of the noise, we simply multiply by 6. Going the other way, we can estimate the RMS value of a noise waveform by looking at its peak-to-peak amplitude in the time domain and dividing by 6. ■

**Example 8.16**

On the TLC220x data sheet, the equivalent RMS input noise voltage for frequencies between 0.1 Hz and 1 Hz is 0.5 μV peak-to-peak. Verify this result with the data given in Fig. 8.31.

Using the numbers from the previous example, we see, in this 0.9 Hz bandwidth, that the contributions from thermal noise are negligible. The contributions from Flicker noise, using Eq. (8.55), are

$$V_{inoise,RMS}^2 = \int_{f_L}^{f_H} \frac{3.14 \times 10^{-15}}{f} \cdot df = 3.14 \times 10^{-15} \cdot \ln \frac{1}{0.1}$$

or

$$V_{inoise,RMS} = 85\ nV$$

The peak-to-peak input-referred noise voltage is roughly six times this or 510 nV (verifying the specification matches the data we get with the plot). ■

**Example 8.17**

Estimate the input-referred noise for the circuit seen in Fig. 8.34.

This amplifier is a transimpedance amplifier, that is, current input and voltage output. The amplifier takes the current generated by the photodiode and converts it into an output voltage. The low-frequency gain of the amplifier is $R_F$ or 100k. The 1,000 pF feedback capacitor limits the bandwidth of the circuit to reduce noise. The photodiode is reverse-biased. Its noise contributions are related to the conversion of light to electrons (which we won't add to our noise analysis). The diode's reverse leakage current will be small and so the shot noise contributions to the circuit noise will be insignificant. The op-amp holds the diode's anode at a fixed potential (ground). The diode's junction capacitance won't affect the noise in the circuit (both sides of the capacitance are at AC ground).

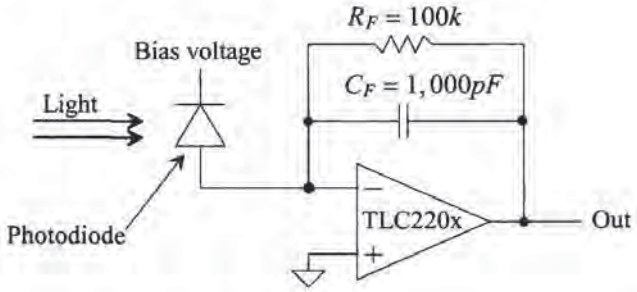



**Figure 8.34** Estimating RMS output noise in a transimpedance amplifier.

IPR2023-00697
Theta EX2019

The basic noise circuit for Fig. 8.34 is seen in Fig. 8.35. All of the output is fed back to the input so that the op-amp's thermal and flicker noise sources appear directly in the output signal (to keep the inverting and noninverting terminals at the same potential). The resistor's thermal noise (keeping in mind that we only look at one noise source at a time so that the inverting and noninverting inputs of the op-amp are at 0V), will create an output voltage that is the product of the noise current and the parallel combination of the 100k and 1,000 pF.

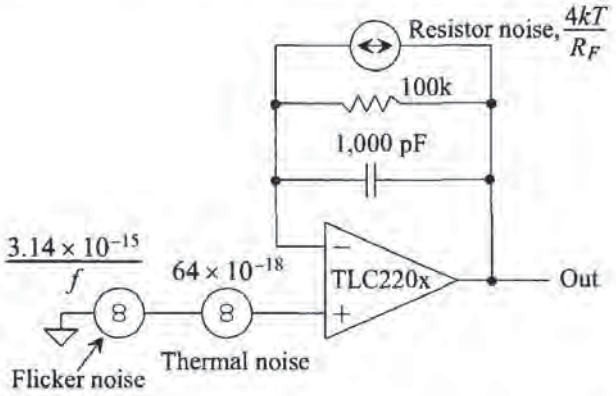



**Figure 8.35** Noise model for the transimpedance amplifier in Fig. 8.34.

The output noise PSD can then be written as

$$V^2_{onoise}(f) = \frac{3.14 \times 10^{-15}}{f} + 64 \times 10^{18} + \overbrace{\frac{4kT}{R_F} \cdot \left| \frac{R_F \cdot 1/j\omega C_F}{R_F + 1/j\omega C_F} \right|^2}^{\frac{4kTR_F}{1+(2\pi f \cdot R_F C_F)^2}}$$

We should recognize the last term in this PSD as simply $kT/C$ noise (Ex. 8.6). The bandwidth of the amplifier can be determined using Fig. 8.36 as

$$A(f) = R_m = \frac{v_{out}}{i_s} = R_F \| \frac{1}{j\omega C_F} = \frac{R_F}{1 + j\omega R_F C_F}$$

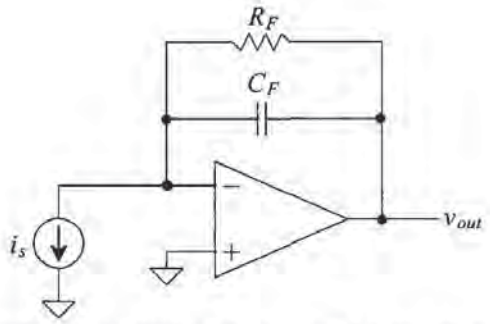



**Figure 8.36** Determining the transfer function of a transimpedance amplifier.

IPR2023-00697
Theta EX2019

or using

$$f_{3dB} = \frac{1}{2\pi R_F C_F}$$

which, for the present example, is 1.59 kHz. We can estimate the output RMS noise remembering the gain-bandwidth product, $f_{un}$, of the op-amp is 2 MHz and noting that the transfer function of the input-referred noise to the output is one as

$$V^2_{onoise,RMS} = \overbrace{49 \cdot 3.14 \times 10^{-15}}^{=154\times10^{-15}} + \overbrace{64 \times 10^{-18} \cdot \frac{\pi}{2} \cdot 2\ MHz}^{=200\times10^{-12}} + \overbrace{\frac{kT}{C_F}}^{=4\times10^{-12}}$$

$$V_{onoise,RMS} \approx \sqrt{(0.392)^2 + (14.14)^2 + (2.07)^2} = 14.3\ \mu V$$

noting that the op-amp's thermal noise dominates the output RMS noise. Our estimate for the flicker noise is high. However, it's easy to throw some numbers in for high and low frequencies in Eq. (8.55) and convince ourselves that the result isn't impacted too much. For example, we used $\ln \frac{10^{11}}{10^{-10}} \approx 49$ (yes, it's closer to 48 but we use 49 because the square-root is a whole number, i.e., 7) when we derived Eq. (8.55). If we were to use instead $\ln \frac{10^5}{10^{-10}}$ (a million times smaller upper frequency), we get 34.5 (a square root of roughly 6). The impact on the result is too small to worry about (unless, of course, the bandwidth is very narrow as in Ex. 8.16).

Dropping the feedback capacitance to 1 pF increases the bandwidth to 1.59 MHz and increases the RMS output noise due to $R_F$ to 64 μV.

Our output signal is determined using $i_s \cdot R_F$. An $SNR_{out}$ of 0 dB, in this example with $C_F$ = 1,000 pF, would correspond to an $i_s$ of roughly 143 pA. If we wanted an input signal to produce an output that is larger than the peak-to-peak value of the output noise (here, from Fig. 8.33, $> 6 \times 14.3\,\mu V$), then $i_s > 858\,pA$. ■

**Example 8.18**

Verify Ex. 8.17 using SPICE. Use a voltage-controlled voltage source for the op-amp (neglect the op-amp noise).

The schematic used for simulations is seen in Fig. 8.37. Instead of connecting the op-amp's noninverting input (the + input) to ground directly, we connect it to

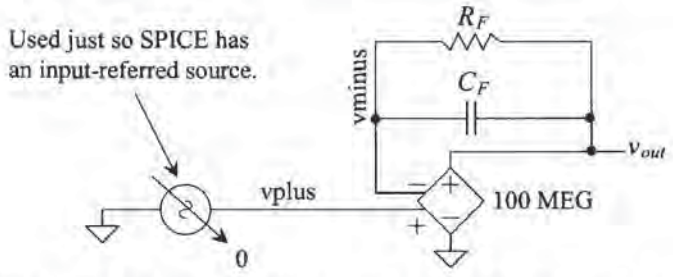



**Figure 8.37** Using a voltage-controlled voltage source for an op-amp in SPICE.

IPR2023-00697
Theta EX2019

ground via a zero volt DC supply so that, in the simulation, we have an "input." (Recall comments about how the AC magnitude of this input source affects the input-referred noise calculation on the bottom of page 227.) The SPICE netlist is

```
Comment line; the first line of a netlist is ignored by SPICE
.noise  V(Vout,0)       Vplus   dec     100     1       1000G
Vplus   Vplus   0       dc      0       ac      1
Rf      Vout    Vminus  100k
Cf      Vout    Vminus          1000pf
Eopamp          Vout    0       Vplus   Vminus  100MEG
.print noise all
.end
```

The SPICE output is

```
TEMP=27 deg C
inoise_total = 4.142190e-12
onoise_total = 4.142190e-12
```

For the 1,000 pF capacitor, this gives 2.03 μV. Resimulating with the 1 pF capacitor, gives 64.3 μV. ■

### 8.2.6 Other Noise Sources

Thermal and flicker noise are the dominant noise sources in CMOS circuit design; however, there are other noise mechanisms. In this section we briefly discuss other types of noise, including random telegraph signal (aka burst or popcorn) and avalanche noise.

*Random Telegraph Signal Noise*

Random telegraph signal (RTS) noise is seen, in the time-domain, in Fig. 8.38. RTS noise is often called popcorn or burst noise. When a signal containing RTS noise is played on a speaker, the RTS noise component sounds like popping corn or the sound a telegraph makes. The origin of RTS noise is thought to be related to how generation-recombination, GR, centers affect the diffusion of carriers in a pn junction. If gold, for example, is introduced into a semiconductor, its inclusion results in nonuniformity in the silicon crystal lattice and thus additional GR sites. With a large number of these sites present, the variation in the diffusion rate, that is, the rate that the carriers diffuse away from a depletion region, can be significant. Since an RTS noise burst, Fig. 8.38, can last hundreds of microseconds, it is likely that at times more sites are filled and thus more carriers can diffuse further into the diode with a smaller number landing in a recombination site. This results in a build up of stored charge in the diode. The popping in the signal occurs when this additional stored charge exits the diode.

Although RTS noise doesn't have a Gaussian noise distribution, its PSD can be modeled using

$$I_{RTS}^2(f) = \frac{K_{RTS} \cdot I_{DC}}{1 + \left(\frac{f}{f_{3dB}}\right)^2} \text{ units of } A^2/Hz \qquad (8.58)$$

where $K_{RTS}$ has units of $A/Hz$ and $f_{3dB}$ (e.g., 100 Hz) is the point where the noise starts to roll off and is related to the number of pops per second, Fig. 8.39. Note that at very low frequencies, RTS noise is white (like thermal and shot noises). However, at higher frequencies, the noise rolls off as $1/f^2$.

IPR2023-00697
Theta EX2019

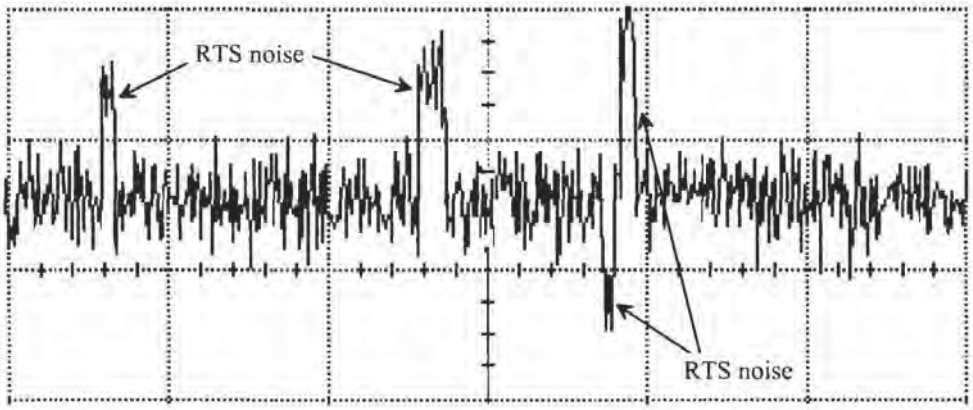



**Figure 8.38** RTS noise viewed in the time domain.

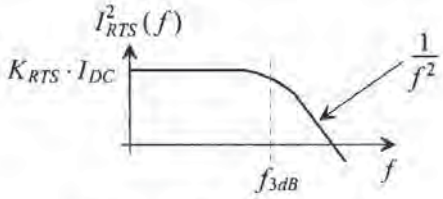



**Figure 8.39** PSD of RTS noise.

*Excess Noise (Flicker Noise)*

Thermal noise in a resistor is present independent of the amount of current flowing in the resistor, as seen in Eq. (8.23). However, there is also flicker noise present in the resistor, dubbed *excess* noise, see Eq. (8.54). Flicker noise is present whenever a direct current flows in a discontinuous material (e.g., a material's surface or the interface between a MOSFET's gate oxide and the silicon used as the channel of the MOSFET). The electrons "jump" from one location to the next while sometimes being randomly trapped and released. Excess (flicker) noise is not dependent on temperature, unlike thermal noise, and so it can be especially troublesome when making low noise, temperature, and frequency measurements.

*Avalanche Noise*

When the electric field in a reverse-biased pn junction gets really large, carriers are accelerated to a point where they can strike the lattice and dislodge additional electron-hole pairs. The result is an increase in diode current. This additional generation of current is termed avalanche multiplication and is used in many Zener diodes to break the diode down. Tunneling is used in the lower voltage, < 6 V, Zener diodes which may have a shot noise spectrum because carriers tunnel across a potential barrier. Avalanche breakdown is a very noisy process. A typical PSD (noting a dependence on the current flowing in the reverse-biased junction) is $10^{-16} V^2/Hz$. The PSD is generally white, which has led to the use of Zener diodes as white noise generators, Fig. 8.40. In low-noise design it can be challenging to design a low-noise voltage reference using a Zener diode.

IPR2023-00697
Theta EX2019

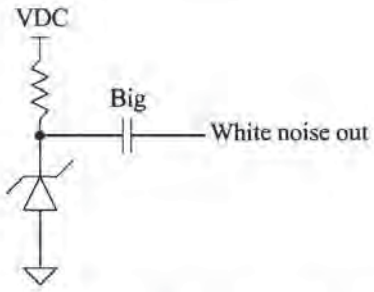



**Figure 8.40** A Zener diode-based white noise generator.

Avalanche noise can also be troublesome in small geometry CMOS processes where a large drain-source voltage can lead to avalanche multiplication (and thus make MOSFET thermal noise appear to be very large).

## 8.3 Discussion

In this section we'll provide additional discussions and examples to help explain electrical noise.

### 8.3.1 Correlation

Consider the two signals seen in Fig. 8.41. If the RMS values of $v_1(t)$ and $v_2(t)$ are called $V_{1RMS}$ and $V_{2RMS}$, respectively, then the power dissipated by the resistor is

$$P_{AVG} = \frac{V_{1RMS}^2 + V_{2RMS}^2}{R} \tag{8.59}$$

as discussed in Sec. 8.1.1 and used throughout the chapter. Let's look at this in more detail because it's easy to show that under certain circumstances this equation is misleading (and wrong!). In particular, if the two voltages sources are correlated, we can get a totally different value for the power the resistor dissipates. For example, we know that the voltage dropped across the resistor is $v_1(t)+v_2(t)$. What happens if $v_1(t) = V_P \cdot \sin(2\pi f \cdot t)$ and $v_2(t) = V_P \cdot \sin(2\pi f \cdot t + \pi)$ (which, of course, is the same as $v_2(t) = -V_P \cdot \sin[2\pi f \cdot t]$). The two sinewaves sum to zero. The voltage dropped across the resistor is zero and so is the power dissipated by the resistor. In a noise analysis, if our

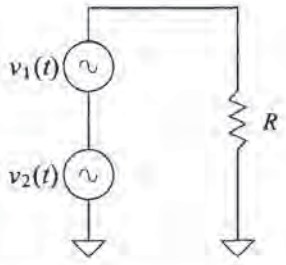



**Figure 8.41** Series combination of voltage signals.

IPR2023-00697
Theta EX2019

noise sources are correlated, then we can't simply add their noise contributions on the output of a network as we've done throughout the chapter. If the noise sources are correlated, the analysis can be much more challenging.

To describe how signals can be correlated in a little more detail, we can write, from Eq. (8.1),

$$P_{inst}(t) = \frac{V^2(t)}{R} = \frac{[v_1(t) + v_2(t)]^2}{R} \tag{8.60}$$

$$= \frac{v_1^2(t)}{R} + \frac{v_2^2(t)}{R} + \frac{2v_1(t)v_2(t)}{R} \tag{8.61}$$

The first two terms in the equation are independent. We can calculate the RMS value of these terms, and thus the average power supplied to $R$ from each source, using

$$P_{AVG} = \frac{V_{RMS}^2}{R} = \frac{1}{T_{meas}} \cdot \int_0^{T_{meas}} \frac{v^2(t)}{R} \cdot dt \tag{8.62}$$

where $T_{meas}$ is the amount of time used to measure the average power. Of course, we use this equation twice (once for each source) and then sum the average powers to get the average power supplied to $R$ (which is Eq. [8.59]). If no correlation exists between $v_1(t)$ and $v_2(t)$, then the third term in Eq. (8.61) is zero. A simple example of two signals with no correlation are sine waves with different frequencies. Because each sinewave will go positive and negative (the signals are no longer squared and thus always positive), the summation of the signal (the integration) results in zero power added or subtracted from the total power. An example of 100% correlation is when

$$v_1(t) = v_2(t) = V_P \cdot \sin 2\pi f \cdot t \tag{8.63}$$

The signal applied to $R$ in Fig. 8.41 has a peak amplitude of $2V_P$, an RMS value of $2V_P/\sqrt{2}$, and supplies a power to $R$ of $2V_P^2/R$. Using Eqs. (8.61) and (8.62), we can write

$$P_{AVG} = \frac{V_P^2}{2R} + \frac{V_P^2}{2R} + \overbrace{\frac{2V_P^2}{2R}}^{\text{100\% correlation}} = \frac{2V_P^2}{R} \tag{8.64}$$

We showed an example of −100% correlation a moment ago when

$$v_1(t) = V_P \cdot \sin 2\pi f \cdot t \text{ and } v_2(t) = -V_P \sin 2\pi f \cdot t \tag{8.65}$$

or

$$P_{AVG} = \frac{V_P^2}{2R} + \frac{V_P^2}{2R} + \overbrace{\frac{2(V_P)(-V_P)}{2R}}^{\text{−100\% correlation}} = 0 \tag{8.66}$$

In general, the correlation between two waveforms is specified using $C$ as

$$V_{1RMS}^2 + V_{2RMS}^2 + 2CV_{1RMS}V_{2RMS} \tag{8.67}$$

where $-1 \le C \le 1$. If $C$ is zero, no correlation exists (again what we've used throughout the chapter). An example of two signals partially correlated are two sine waves at the same frequency but different amplitudes or phases. Note that this discussion used RMS values for the signals. However, it is also possible to discuss correlation in PSDs.

IPR2023-00697
Theta EX2019

*Correlation of Input-Referred Noise Sources*

When we discussed input-referred noise and derived Eq. (8.32), we assumed zero correlation between $I_{inoise,RMS}$ and $V_{inoise,RMS}$. We can rewrite Eq. (8.32) without the thermal noise from the source resistance and with the help of Fig. 8.42 as

$$V_{onoise,RMS}^2 = \left(\frac{AR_{in}}{R_s + R_{in}}\right)^2 [V_{inoise,RMS} + I_{inoise,RMS} \cdot R_s]^2 \qquad (8.68)$$

If the correlation is zero, Eq. (8.68) is equivalent to Eq. (8.32) (without thermal noise contributions from $R_s$). With correlation, we can write

$$V_{onoise,RMS}^2 = \left(\frac{AR_{in}}{R_s + R_{in}}\right)^2 \left[V_{inoise,RMS}^2 + I_{inoise,RMS}^2 \cdot R_s^2 + 2C \cdot V_{inoise,RMS} \cdot I_{inoise,RMS}\right] \qquad (8.69)$$

To use the input-referred noise models, we generally assume no correlation ($C = 0$) between $I_{inoise,RMS}$ and $V_{inoise,RMS}$. In a practical circuit, the correlation can be significant because both are derived from the same noise mechanisms.

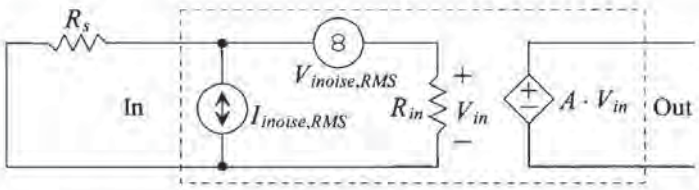



**Figure 8.42** Correlation calculation of input-referred noise sources.

*Complex Input Impedance*

We've used a real input resistance in all of our calculations in this chapter. It's straightforward to extend our discussions and derivation to a general input impedance, $Z_{in}$. Our input voltage divider changes to

$$\frac{R_{in}}{R_s + R_{in}} \rightarrow \frac{Z_{in}}{R_s + Z_{in}} \qquad (8.70)$$

When we multiply this divider by, say, $V_s$, to get $V_{in}$, we use

$$V_{in} = V_s \cdot \frac{|Z_{in}|}{|R_s + Z_{in}|} \text{ or } V_{in}^2 = V_s^2 \cdot \frac{|Z_{in}|^2}{|R_s + Z_{in}|^2} \qquad (8.71)$$

It's important to take the magnitude before squaring. For example, if $Z_{in} = 1/j\omega C_{in} = -j/\omega C_{in}$(the input impedance is a capacitor), then the correct way of calculating the denominator in Eq. (8.71) is

$$|R_s + Z_{in}|^2 = \left(\sqrt{R_s^2 + (1/\omega C_{in})^2}\right)^2 = R_s^2 + (1/\omega C_{in})^2 \qquad (8.72)$$

IPR2023-00697
Theta EX2019

The incorrect way is

$$|R_s + Z_{in}|^2 \neq |(R_s + 1/j\omega C_{in})(R_s + 1/j\omega C_{in})| = |\overbrace{R_s^2 + 1/\omega^2 C_{in}^2}^{\text{real term}} + j\cdot\overbrace{(-2/\omega C_{in})}^{\text{imaginary term}}|$$
$$= \sqrt{\left(R_s^2 + 1/\omega^2 C_{in}^2\right)^2 + (2/\omega C_{in})^2} \text{ (again wrong!)} \tag{8.73}$$

It appears that there may be correlation in the equation above but it is really just bad complex algebra. We used the magnitude squared earlier in Eq. (8.13), that is, $|1 + j(f/f_{3dB})|^2$ or $1 + (f/f_{3dB})^2$. The point is to always take the magnitude of the expressions prior to squaring.

Let's use a complex input impedance, $Z_{in}$, to derive the optimum source resistance, Eq. (8.44). As we saw in Ex. 8.7 and 8.9, we measure the output noise with $R_s$ shorted (or a big capacitor placed across the input of the circuit, which is essentially a short for all frequencies of interest) or opened to determine the input-referred noise sources. If the measured output noise (excluding thermal noise contributions from $R_s$) doesn't change with variations in $R_s$ (quite common in amplifiers whose bias is independent of $R_s$ like AC-coupled input amplifiers, but not the case for the circuit in Ex. 8.9), that is,

$$V_{onoise,RMS,Rs=\infty} = V_{onoise,RMS,Rs=0} = V_{onoise,RMS} \tag{8.74}$$

and the input-referred noise sources are uncorrelated then, for a real input resistance,

$$R_{s,opt} = R_{in} \tag{8.75}$$

We may think that we can get both maximum power transfer and the best noise performance by matching the input resistance to the source resistance. However, if we use a complex input impedance, Eq. (8.75) becomes

$$R_{s,opt} = |Z_{in}| = |R_{in} + jX_{in}| = \sqrt{R_{in}^2 + X_{in}^2} \tag{8.76}$$

which can be valid without regard for maximum power transfer. Knowing that the requirement for maximum power transfer is making the source impedance the complex conjugate of the load impedance, we see that if the source impedance is purely real, it will be impossible to achieve maximum power transfer when the input resistance has an imaginary component. In radio-frequency circuits (narrow bandwidths where the impedances are relatively constant), impedance transformation circuits are used to improve both power transfer and noise performance. From the source side, the input impedance is transformed to maximize the transfer of power. From the input side, the source impedance is transformed to provide the best noise performance.

---

**Example 8.19**

Suppose an amplifier has an output noise PSD, $V_{onoise}(f)$, of 1 $\mu V/\sqrt{Hz}$ (we get this PSD if $R_s$ is a short or an open), a gain of 100, and an input capacitance of 1 pF (infinite input resistance). Determine the input-referred noise sources for the amplifier.

IPR2023-00697
Theta EX2019

With the input shorted, Fig. 8.43a, we get an input-referred noise voltage spectral density of

$$A \cdot V_{inoise}(f) = V_{onoise}(f) = 1\ \mu V/\sqrt{Hz}$$

or since $A = 100$

$$V_{inoise}(f) = 10\ nV/\sqrt{Hz}$$

With the input opened, Fig. 8.43b, we get

$$I_{inoise}(f) \cdot |Z_{in}| \cdot A = \frac{I_{inoise}(f)}{2\pi f \cdot 1pF} \cdot 100 = 1\ \mu V/\sqrt{Hz}$$

or

$$I_{inoise}(f) = 62.8 \times 10^{-21} \cdot f\ A/\sqrt{Hz}$$

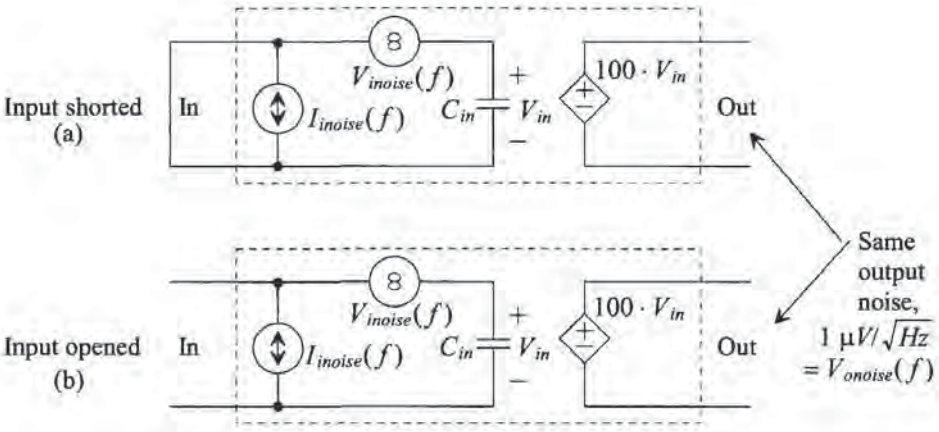



**Figure 8.43** Determining input-referred noise in Ex. 8.19.

Note that the spectral density increases with $f$ (Fig. 8.44). The noise current contributions to the output noise is insignificant until the frequency gets comparable to $1/(2\pi R_s C_{in})$ (noting that squaring $I_{inoise}(f)$ and integrating to find the RMS value results in an $f^3$ term). Unless the source resistance, frequencies of interest, or input capacitance are relatively large, the single input-referred noise voltage source is all that is needed to model the amplifier's output noise. ■

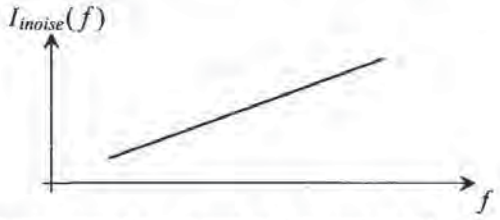



**Figure 8.44** Input-referred noise current increasing with frequency when the input impedance is capacitive.

IPR2023-00697
Theta EX2019

### 8.3.2 Noise and Feedback

Figure 8.45a shows the basic block diagram of a feedback circuit. In Fig. 8.45b we show the input-referred noise source. The output of the summer in (a) or (b) can be written as $V_{in} - \beta \cdot V_{out}$. The output of the circuit in (b) is

$$V_{out} = [(V_{in} - \beta V_{out}) + V_{inoise,RMS}] \cdot A \tag{8.77}$$

or

$$V_{out} = \frac{A}{1 + \beta A} \cdot (V_{in} + V_{inoise,RMS}) \tag{8.78}$$

In other words, feedback doesn't affect the circuit's noise performance. The input-referred noise adds directly to the input signal independent of the feedback. In practical circuits, the addition of resistors to provide feedback degrades the noise performance of the amplifier.

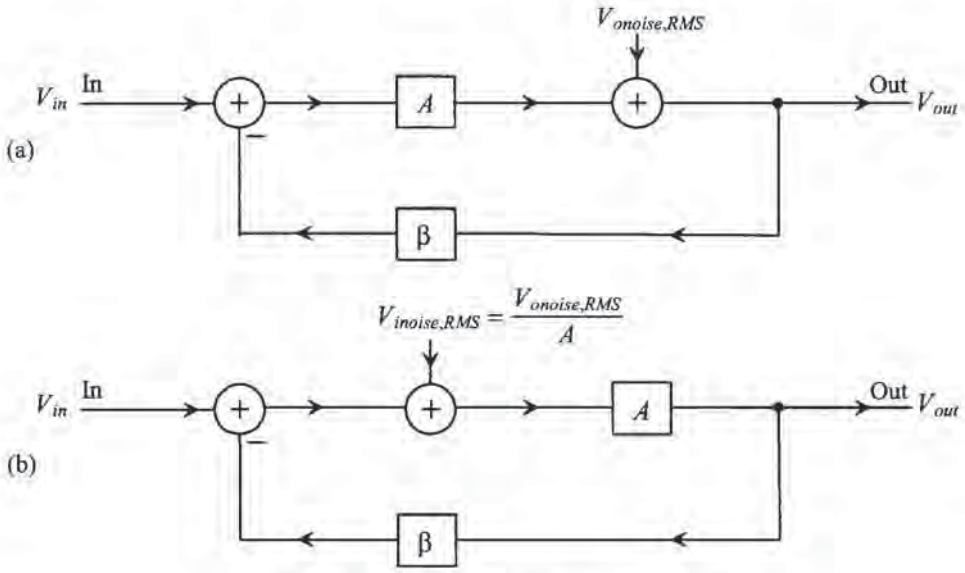



**Figure 8.45** Noise in a feedback circuit.

*Op-Amp Noise Modeling*

Figure 8.46a shows how input-referred noise sources are added to an op-amp to model output noise. The input-referred noise current is connected to the inverting terminal of the op-amp, while the input-referred noise voltage is connected in series with the noninverting terminal. Figure 8.46b shows the general implementation of a feedback amplifier using an op-amp, including noise sources modeling the resistor's thermal noise. If the input to the op-amp circuit is on the left side of $R_1$, then the gain to the output is inverting, $(-R_2/R_1)$. If the input is connected directly to the noninverting terminal of the op-amp, then the gain is positive, $(1 + R_2/R_1)$. Remember, we look at only one noise source at a time (superposition) when we calculate the total output noise. Also, remember that the op-amp, through the feedback action, holds the inverting terminal at ground.

IPR2023-00697
Theta EX2019

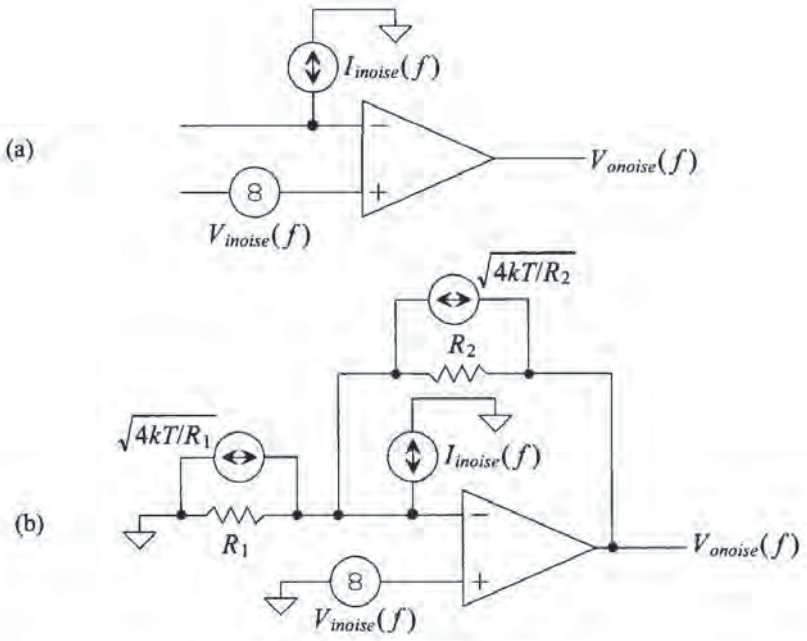



**Figure 8.46** Modeling op-amp noise.

Let's write the output noise, for Fig. 8.46b, in terms of the input-referred noise sources and the thermal noise from the resistors. Because both sides of $R_1$ are at ground (one side is physically tied to ground while the other side is held at ground by the op-amp), we can write

$$V_{onoise}^2(f) = V_{inoise}^2(f) \cdot \left(1 + \frac{R_2}{R_1}\right)^2 + \left[I_{inoise}^2(f) + \frac{4kT}{R_1} + \frac{4kT}{R_2}\right] \cdot R_2^2 \qquad (8.79)$$

The gain of this circuit for the sake of calculating bandwidth is $(1 + R_2/R_1)$. The closed-loop value of bandwidth is written in terms of the op-amp's gain-bandwidth product, $f_{un}$ (= the frequency when the open-loop gain of the op-amp is unity, see pages 792-793) as

$$\text{op-amp's gain-bandwidth} = f_{un} = \left(1 + \frac{R_2}{R_1}\right) \cdot f_{3dB} \qquad (8.80)$$

The NEB, Eq. (8.15), can be written as

$$NEB = \frac{\pi}{2} \cdot f_{un} \cdot \frac{R_1}{R_1 + R_2} \qquad (8.81)$$

Assuming that the input-referred sources are not correlated and the NEB of the output noise limits the RMS values, see Eq. (8.15) and associated discussion, we can write

$$V_{onoise,RMS} = \sqrt{V_{onoise}^2(f) \cdot \frac{\pi}{2} \cdot f_{un} \frac{R_1}{R_1 + R_2}} \qquad (8.82)$$

where $V_{onoise}^2(f)$ is assumed to be white until it roll-offs with the amplifer's bandwidth, Fig. 8.9b.

IPR2023-00697
Theta EX2019

As an example, Fig. 8.47 shows the input-referred noise sources for the LT1364 op-amp. Neglecting the low-frequency (flicker) noise, we see that $V_{inoise}(f) = 9\ nV/\sqrt{Hz}$ ($= e_n$ in Fig. 8.47) and $I_{inoise}(f) = 0.85\ pA/\sqrt{Hz}$ ($= i_n$ in Fig. 8.47). From the datasheet, the gain-bandwidth product ($f_{un}$) of this op-amp is nominally 70 MHz (and so an op-amp in a gain of 10 ($= 1 + R_2/R_1$) configuration has $f_{3dB} = 7$ MHz).

Equations (8.79) and (8.82) can be interpreted in many ways. If our op-amp is used as a transimpedance amplifier as in Fig. 8.34 (Ex. 8.17), then we want $R_1$ to be infinite to minimize the output noise. We think of the input source, $i_s$, as a current connected to the inverting op-amp input, the amplifier as having a gain of $|R_2|$, and the output voltage as $i_s \cdot R_2$. (The inverting input to the op-amp is a virtual ground, which is the ideal input resistance for an input current signal.) Looking at the $SNR_{out}$ or $(i_s \cdot R_2)/V_{onoise,RMS}$, we see that reducing $R_2$ lowers both the gain and the noise while at the same time causing the $SNR_{out}$ to drop.

If the op-amp is used as an inverting voltage amplifier with a gain of $-R_2/R_1$ (input signal connected to the left side of $R_1$), or as a noninverting voltage amplifier with a gain of $1 + R_2/R_1$ (input signal connected to the noninverting op-amp input terminal) then increasing $R_1$, as seen in Eq. (8.79), reduces the output noise. If $R_2$ is held constant then increasing $R_1$ also reduces the amplifier gain causing the $SNR_{out}$ to decrease (where $SNR_{out} = (v_s \cdot R_2/R_1)/V_{onoise,RMS}$ or $(v_s \cdot [1 + R_2/R_1])/V_{onoise,RMS}$. In this situation, for large $SNR_{out}$, we want both $R_1$ (for large gain) and $R_2$ (to minimize the output noise as seen in Eq. [8.79]) to be as small as possible. For example, if we want a gain of 10, it is preferable to use an $R_1$ and $R_2$ of 10k and 1k over using 100k and 10k. Of course, using lower values results in more power dissipation in these feedback resistors.

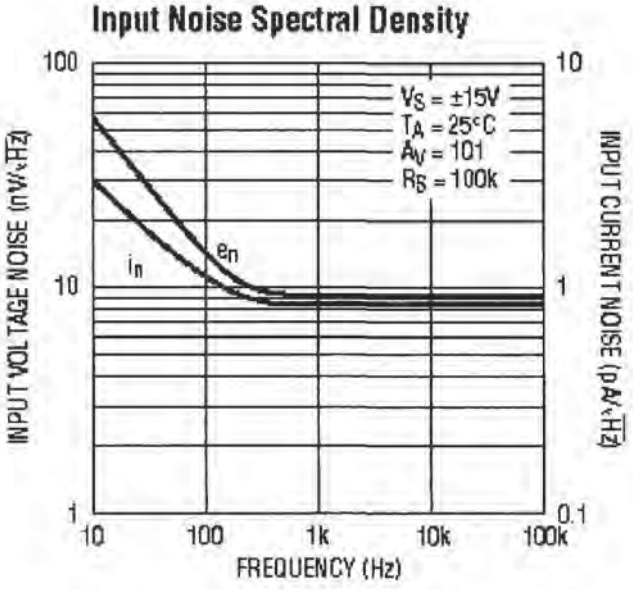



**Figure 8.47** Input-referred noise sources for the LT1364 high-speed op-amp.

IPR2023-00697
Theta EX2019

### 8.3.3 Some Final Notes Concerning Notation

When we calculate the mean-squared value of a noise voltage signal (which means that the desired signal components are not present in the signal), we used

$$\overline{v^2} = V_{RMS}^2 = \frac{1}{T_{meas}} \int_0^{T_{meas}} v^2(t) \cdot dt \tag{8.83}$$

The square root of the mean-squared voltage is the RMS value of a noise waveform. In Ex. 8.14 we showed that the RMS value of the integrated thermal noise increases with measuring time. Does this mean that if we measure, or look, at the value of this noise signal at a specific time it can't be zero? As seen in Fig. 8.26, a noise voltage can be positive, zero, or negative. What we are saying when we say the RMS value is increasing is that the voltage excursions away from zero are growing (but $v$(t) is still zero at some times).

The next thing we should note is that the mean-squared value is a number not a spectral density. Strictly speaking, it is incorrect to write, for example, the thermal noise of a resistor as

$$\overline{v^2} = 4kTR \text{ or } \overline{i^2} = \frac{4kT}{R} \tag{8.84}$$

A number (the left side of the equation) is not equal to a spectrum (the right side of the equation is a PSD). When we talked about measuring PSD with a spectrum analyzer in Sec. 8.1.2, we said that we plot a point by dividing the measured power by the resolution bandwidth, that is, $V_{RMS}^2/f_{res}$, at a particular frequency. Sometimes the notation, $\Delta f$, is used for the resolution bandwidth $f_{res}$. It is correct to write the PSD of thermal noise as

$$V_R^2(f) = \frac{V_{RMS}^2}{f_{res}} = \frac{\overline{v^2}}{\Delta f} = 4kTR \text{ units of } \frac{V^2}{Hz} \tag{8.85}$$

It is also correct to write

$$\overline{v^2} = 4kTR \cdot \Delta f \text{ volts} \tag{8.86}$$

where the resolution bandwidth is assumed narrow (for a general noise signal that may or may not be white). What makes this more confusing is that the mean-squared values are used for both narrow band signals, as seen in Eqs. (8.85) and (8.86), and for entire noise spectrums. For example, to calculate the RMS value of a wideband noise signal, we use

$$V_{RMS}^2 = \overline{v^2} = \int_{f_L}^{f_H} V_R{}^2(f) \cdot df = 4kTR \cdot B \tag{8.87}$$

where $B = f_H - f_L \neq \Delta f = f_{res}$. Note that $\Delta f$ has nothing to do with *NEB* or *B*.

As another example, if we were to measure the PSD of flicker noise, we would get a value of

$$V_{1/f}^2(f) = \frac{\overline{v^2}}{\Delta f} = \frac{FNN}{f} \tag{8.88}$$

at each frequency $f$. The measured power in the narrow bandwidth $\Delta f$ ($= f_{res}$), $\overline{v^2}$ ($= V_{RMS}^2$), decreases as $f$ goes up resulting in the $1/f$ PSD.

IPR2023-00697
Theta EX2019

## ADDITIONAL READING AND/OR INFORMATION

[1] R. J. Baker, *CMOS: Mixed-Signal Circuit Design, Second Edition*, Wiley-IEEE Press, 2009. Covers noise/averaging and the affects on signal-to-noise ratio. Good complement to the material in this chapter.

[2] W. T. (Tim) Holman, Private communication, Feb. 2002. Information on how the RMS value of a signal containing $1/f$ noise (and $1/f^2$, $1/f^3$, etc.) grows with measuring time.

[3] C. D. Motchenbacher and J. A. Connelly, *Low-Noise Electronic System Design*, John Wiley and Sons, 1993. ISBN 0-471-57742-1. Excellent reference for low-noise design.

[4] R. Sarpeshkar, T. Delbrück, and C. A. Mead, "White Noise in MOS Transistors and Resistors," *IEEE Circuits and Devices Magazine*, Nov. 1993. Shows that both shot and thermal noise mechanisms in MOSFETs have the same origins.

[5] D. M. Binkley, J. M. Rochelle, M. J. Paulus, and M. E. Casey, "A Low-Noise, Wideband, Integrated CMOS Transimpedance Preamplifier for Photodiode Applications," in *IEEE Transactions on Nuclear Science*, vol. NS-39, no. 4, August 1992, pp. 747–752. Covers the design of low-noise transimpedance amplifiers.

[6] H. L. Krauss, C. W. Bostian, and F. H. Raab, *Solid State Radio Engineering*, John Wiley and Sons, 1980. ISBN 0-471-03018-X. Reference for the design of radio circuits.

[7] W. R. Bennett, *Electrical Noise*, McGraw-Hill, 1960. Fundamental reference on electrical noise. Written by a pioneer in noise theory. No longer in print but available used through various book sellers.

[8] A. Van der Ziel, *Noise; Sources, Characterization, Measurement*, Prentice-Hall. Also written by a pioneer in noise theory.

[9] H. F. Friis, "Noise Figures of Radio Receivers," in *Proceedings of the IRE* (Institute of Radio Engineers), vol. 32, no. 7, 1944, pp. 419–422. Fundamental paper covering noise figure.

## LIST OF SYMBOLS/ACRONYMS

$A$ - Voltage gain of a circuit.

$A_{DC}$ - DC gain of a circuit.

$B$ - Bandwidth of a noise measurement, i.e., $f_H - f_L$.

$C$ - Correlation term, see Eq. (8.67).

$C_{in}$ - Input capacitance.

$C_j$ - Diodes junction capacitance (also called depletion capacitance).

$C_{j0}$ - Depletion capacitance measured with zero volts across the diode.

$CUT$ - Circuit under test.

IPR2023-00697
Theta EX2019

$\Delta f$ - Resolution bandwidth, same as $f_{res}$.

DFT - Discrete Fourier transform.

ESD - Energy spectral density.

$F$ - Noise factor.

$FNN$ - Flicker noise numerator.

$F_{opt}$ - Optimum noise factor.

$f$ - Frequency of a sinewave.

$f_{3dB}$ - 3 dB frequency of a circuit (the power is half the low frequency value at $f_{3dB}$ or the voltage is 0.707 its low-frequency value at a frequency of $f_{3dB}$).

$f_H$ - Higher frequency in an RMS noise calculation, see Eq. (8.10).

$f_L$ - Lower frequency in an RMS noise calculation, see Eq. (8.10).

$f_{res}$ - Resolution bandwidth of a spectrum analyzer or a DFT (same as $\Delta f$).

$f_{start}$ - Starting frequency in a DFT or a spectrum analyzer measurement.

$f_{stop}$ - Final frequency in a DFT or a spectrum analyzer measurement.

$f_{un}$ - Unity gain frequency of an op-amp (the open-loop gain is 1).

GR - Generation-recombination.

$\overline{i^2}$ - Mean-squared current$\left(=I_{RMS}^2\right)$.

$I(f)$ - Square root of a current PSD, units of $A/\sqrt{Hz}$.

$I^2(f)$ - PSD of a noise current, units of $A^2/Hz$.

$I_{1/f}^2(f)$ - PSD of flicker noise current, units of $A^2/Hz$.

$I_{inoise}^2(f)$ - PSD of the input-referred noise of a circuit, units of $A^2/Hz$.

$I_{inoise,RMS}$ - Input-referred RMS noise current, unit of $A$.

$I_{RMS}$ - Root mean-squared value of a current waveform, units of $A$, also written in this chapter as $\sqrt{\overline{i^2}}$.

$I_R^2(f)$ - PSD of thermal noise, units of $A^2/Hz$.

$I_{RMS}^2$ - Mean-squared value of a current waveform, units of $A^2$, also written as $\overline{i^2}$.

$I_{RTS}^2(f)$ - PSD of random telegraph signal noise current, units of $A^2/Hz$.

$I_{shot}^2(f)$ - Shot noise current PSD, units of $A^2/Hz$.

$j$ - $\sqrt{-1}$.

$K$ - Number of averages, see Eq. (8.48).

$k$ - Boltzmann's constant, $13.8 \times 10^{-24} J/K$.

$LNA$ - Low-noise amplifier.

IPR2023-00697
Theta EX2019

$n$ - A counting index.

$NEB$ - Noise-equivalent bandwidth, see Eq. (8.15).

$NF$ - Noise figure, see Eq. (8.35).

$P_{noise}(f)$ - PSD of a signal with units of $W/Hz$.

$P_{AVG}$ - Average power dissipated.

PDF - Probability density function, see Fig. 8.33.

$P_{inst}(t)$ - Instantaneous power dissipation (the power dissipated at a specific time). Units of watts (Joules/s).

PSD - Power spectral density, units of $V^2/Hz$.

$q$ - Electron charge of $1.6 \times 10^{-19}$ coulombs.

$R$ - Resistance in ohms.

$R_s$ - Resistance of an input source signal.

$RMS$ - Root mean square.

SA - Spectrum analyzer.

$SNR_{in}$ - Input signal-to-noise ratio, see Eq. (8.28).

$SNR_{out}$ - Output signal-to-noise ratio, see Eq. (8.40).

$t$ - Time in seconds.

$T$ - Temperature in Kelvin.

$T_{meas}$ - Time a measurement is taken, units of seconds.

$\overline{v^2}$ - Mean-squared voltage $\left(= V_{RMS}^2\right)$.

$V(f)$ - Square root of a PSD, units of $V/\sqrt{Hz}$ (also called a voltage spectral density).

$v(t)$ - Time domain voltage waveform.

$V_{in}$ - Input voltage.

$V_{inoise}^2(f)$ - PSD of the input-referred noise of a circuit.

$V_{inoise,RMS}$ - Input-referred RMS noise voltage, units of $V$.

$V_{LF,noise}^2(f)$ - Low-frequency white noise, see Fig. 8.9.

$V_{noise}^2(f)$ - PSD of a noise signal.

$V_{onoise}^2(f)$ - PSD of the output noise of a circuit.

$V_{onoise,RMS}$ - Output RMS noise voltage.

$V_p$ - Peak voltage of a sine waves wave.

$V_R^2(f)$ - PSD of thermal noise, units of $V^2/Hz$.

IPR2023-00697
Theta EX2019

$V_{1/f}^2(f)$ - PSD of flicker noise voltage, units of $V^2/Hz$.

$V_{RMS}$ - Root mean-squared value of a voltage waveform, units of $V$, also written in this chapter as $\sqrt{\overline{v^2}}$ .

$V_{RMS}^2$ - Mean-squared value of a voltage waveform, units of $V^2$, also written in this chapter as $\overline{v^2}$.

$V_s$ - Source input voltage.

$V_T$ - Thermal voltage, $kT/q$ or 26 mV at room temperature.

$\omega$ - $2\pi f$ .

$Z_{in}$ - A complex input impedance.

**PROBLEMS**

**8.1** Suppose that the power company is charging $ 0.25 for a kilowatthour (1,000 W of power supplied for one hour) of energy. How much are you paying for a Joule of energy? How much energy, in Joules, is supplied to a 100 W lightbulb in one hour? Is it potential energy or kinetic energy the power company sells?

**8.2** The power company attempts to hold, in the United States, the RMS value of the voltage supplied to households to 120 V RMS with $\pm 3V$ at a frequency of 60 Hz. What is the peak-to-peak value of this voltage? If we were to look at the PSD of this signal in a 10 Hz bandwidth, what amplitude would we see?

**8.3** Suppose a SA measures a noise voltage spectral density of 1 $\mu V/\sqrt{Hz}$ . What is the RMS value of this noisy signal over a bandwidth of DC to 1 MHz?

**8.4** Suppose the noise signal in problem 3 shows a 3-dB frequency of 5 MHz. What is its RMS value over an infinite bandwidth?

**8.5** Estimate the RMS output noise in the following circuit over a bandwidth of 1 to 1kHz. Verify your answer with SPICE. (Hint: simplify the circuit by combining resistors.)

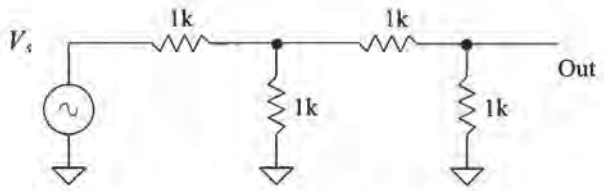



**Figure 8.48** Circuit for Problem 8.5.

**8.6** Estimate the RMS output noise over an infinite bandwidth for the circuit in Fig. 8.48 if the output is shunted with a 1 pF capacitor.

**8.7** Show, Fig. 8.49, that the calculation of $SNR_{in}$ results in the same value independent of treating the input as a voltage or a current.

IPR2023-00697
Theta EX2019

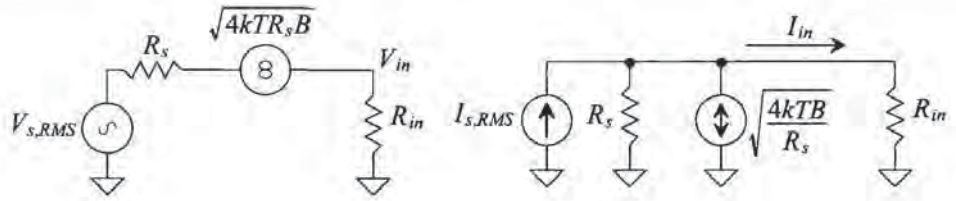



**Figure 8.49** Calculating input SNR in Problem 8.7.

**8.8** Using the input-referred noise model seen in Fig. 8.20b, verify that if the input resistance becomes infinite, the output noise is adequately modeled using a single input-referred noise voltage.

**8.9** If an amplifier has a 0 dB NF, does that indicate the amplifier's output is free of noise? Why or why not?

**8.10** Verify, as seen in Eq. (8.45), that the input-referred noise power of an amplifier is $(V_{inoise,RMS} \cdot I_{inoise,RMS})/2$.

**8.11** Verify that the units for the shot noise PSD seen in Eq. (8.49) are indeed $A^2/Hz$.

**8.12** Repeat Ex. 8.12 if the input voltage is reduced from 1.7 V to 1 V.

**8.13** Verify the comment made in Sec. 8.2.5 that the RMS value of integrated flicker noise signal increases linearly with measurement time.

**8.14** If the maximum allowable RMS output noise of a transimpedance amplifier built using the TLC220x is 100 μV in a bandwidth of 1 MHz is needed, what are the maximum values of $C_F$ and $R_F$? What is the peak-to-peak value of the noise in the time domain?

**8.15** Estimate the RMS output noise for the circuit seen in Fig. 8.50. Compare the hand-calculated result to SPICE simulation results using an ideal op-amp (as used in Ex. 8.18).

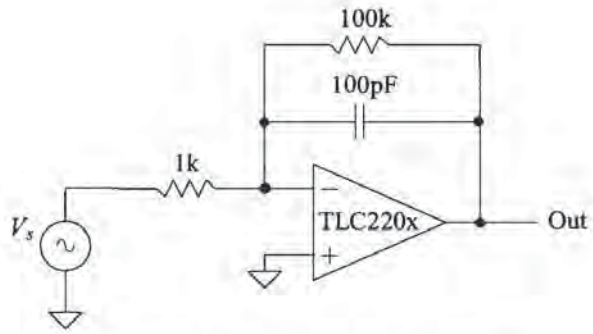



**Figure 8.50** Circuit for Problem 8.15.

IPR2023-00697
Theta EX2019

**8.16** Suppose a 4.7 V Zener diode is used in a noise generator, as seen in Fig. 8.40. Estimate the output noise voltage PSD in terms of the diode's noise current and the biasing resistor (use 1k) if the power supply is 9 V. (Note: the noise mechanism is shot noise not avalanche noise.)

**8.17** Rewrite Eq. (8.72) if $Z_{in} = R_{in} + 1/j\omega C_{in}$.

**8.18** Estimate the RMS output noise for the amplifier seen in Fig. 8.51. What would placing a capacitor across the feedback resistor do to the RMS output noise and to the speed (bandwidth) of the amplifier? Name two ways to lower the RMS output noise for this amplifier. What is the cost for the lower output noise?

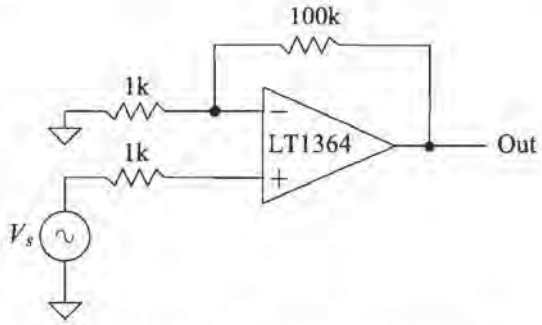



**Figure 8.51** Circuit for Problem 8.18.

IPR2023-00697
Theta EX2019

Chapter

9

# Models for Analog Design

In this chapter we develop models for analog design using the MOSFET. We'll break this discussion up into three sections. The first section covers long-channel MOSFET models with the assumption that the MOSFET follows the "square-law" equations derived in Ch. 6. In the second section we discuss models using modern MOSFETs with short-channel lengths (< 1 μm). Models for these short-channel MOSFETs are developed with graphs showing device characteristics, (e.g., output resistance, transconductance, etc.). Finally, at the end of the chapter, we introduce MOSFET noise modeling.

## 9.1 Long-Channel MOSFETs

When we do analog design we often say things like "the MOSFET looks like a current source when operating in the saturation region" or "it looks like a resistor." Before going too far, let's make sure that we understand these statements. Examine the current-voltage (IV) plot in Fig. 9.1. In this figure we've plotted the (DC) current-voltage characteristics of a resistor, a current source, and a voltage source. Often the controlling parameter in a semiconductor device is a voltage. The controlled parameter is then the device's output current (and this is why current is on the y-axis and voltage is on the x-axis in an IV plot).

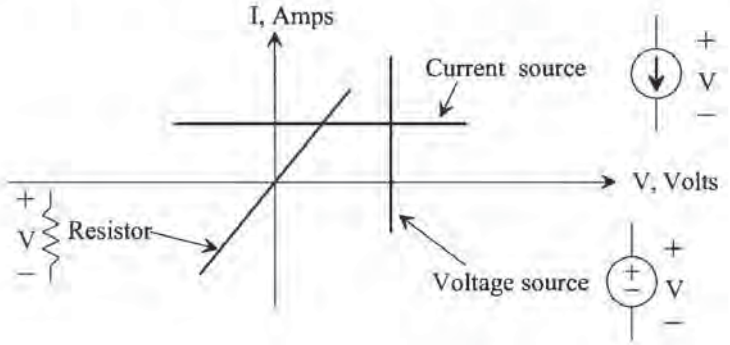



**Figure 9.1** Current-voltage (IV) plots for various electrical components.

IPR2023-00697
Theta EX2019

The voltage across the voltage source, for example, doesn't vary with changes in current running through it. The voltage across the resistor is linearly related to the current flowing through the resistor (Ohm's law). An important thing to note is that resistance can be calculated by taking the reciprocal of the IV plot slope. (So the voltage source in this figure has zero resistance; the current source, infinite resistance.) Also note that the x-axis corresponds to plotting the IV characteristics of an open circuit (no current with changes in voltage). The y-axis corresponds to a short (no changes in the voltage across a wire [a short], with changing current).

**Example 9.1**
Plot the IV characteristics for the circuit seen in Fig. 9.2.

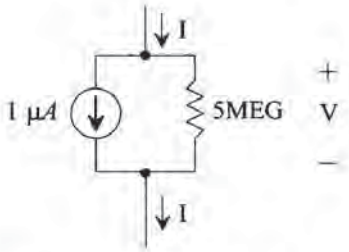



**Figure 9.2** Circuit for Ex. 9.1.

Figure 9.3a shows the IV curves for each component of the circuit where we use a single quadrant of the IV plotting plane. The slope of the resistor is 200 $nA/1\ V$. The resistance value is the reciprocal of this slope (5MEG). The combined IV curve is seen in Fig. 9.3b. ■

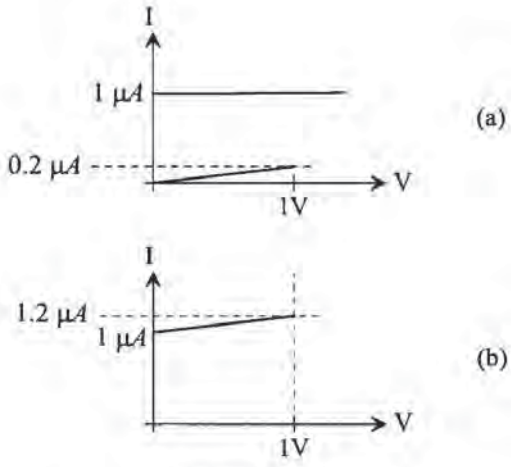



**Figure 9.3** IV plots for Ex. 9.1.

IPR2023-00697
Theta EX2019

**Example 9.2**
Figure 9.4 shows the IV curves (drain current versus drain-source voltage with constant gate source voltage) for a MOSFET. Comment on what the MOSFET looks like in the triode and saturation regions.

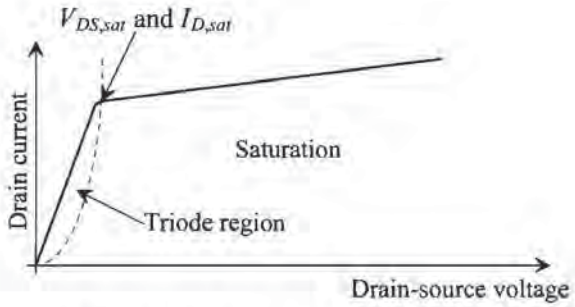



**Figure 9.4** IV curves of a MOSFET.

Clearly, in the triode region (also known as the linear or ohmic region), the MOSFET behaves like a resistor. In the saturation region, the MOSFET behaves, as seen in Fig. 9.3b, like a current source in parallel with a resistor. The resistive component, whether in the triode or saturation regions, is often called the MOSFET's output resistance. ■

### 9.1.1 The Square-Law Equations

The drain current, $I_D$, is related to the gate-source voltage, $V_{GS}$, and the drain-source voltage, $V_{DS}$, using

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - V_{THN})^2(1 + \lambda(V_{DS} - V_{DS,sat})) \qquad (9.1)$$

for $V_{DS} \geq V_{DS,sat}$ and $V_{GS} \geq V_{THN}$. As seen in Fig. 9.4, $V_{DS,sat}$ is the voltage where the MOSFET moves from the triode region to the saturation region. For long-channel MOSFETs, this can be written as

$$V_{DS,sat} = V_{GS} - V_{THN} \qquad (9.2)$$

This term is very important and will be used frequently when doing analog design. Notice that $V_{DS,sat}$ represents the amount of gate-source voltage that we have in *excess* or *over* the threshold voltage. For this reason, it is sometimes called

$$V_{DS,sat} = \text{excess gate voltage} = \text{gate overdrive voltage} \qquad (9.3)$$

Note that while **Eq. (9.2) is only valid for long-channel MOSFETs**, the voltage, $V_{DS,sat}$, simply indicates, for long- or short-channel MOSFETs, the $V_{DS}$ at the boundary between triode and saturation. When $V_{DS} = V_{DS,sat}$, the drain current is labeled $I_{D,sat}$ or

$$I_{D,sat} = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - V_{THN})^2 = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{DS,sat})^2 \qquad (9.4)$$

Equation (9.1) can be rewritten as

IPR2023-00697
Theta EX2019

$$I_D = I_{D,sat} + I_{D,sat}\lambda \cdot (V_{DS} - V_{DS,sat}) \tag{9.5}$$

Using the results from Exs. 9.1 and 9.2, we see that the MOSFET behaves, while in the saturation region, like a current source $I_{D,sat}$ in parallel with a resistor of value

$$r_o = \frac{1}{\lambda I_{D,sat}} \tag{9.6}$$

Figure 9.5 shows a gate-drain connected MOSFET, something we'll see often in analog design. Notice that $V_{GS} = V_{DS}$. If $V_{GS} > V_{THN}$ (a current is flowing through the device), then, for the MOSFET to operate in the saturation region, we must have $V_{DS} \geq V_{GS} - V_{THN}$ or $0 \geq -V_{THN}$ (indicating that a gate-drain-connected MOSFET with a current flowing through it is always operating in the saturation region [remember this]). We can also write the requirement for operation in the saturation region as

$$\overbrace{V_D - V_S}^{V_{DS}} \geq \overbrace{V_G - V_S}^{V_{GS}} - V_{THN} \text{ or } V_D \geq V_G - V_{THN} \tag{9.7}$$

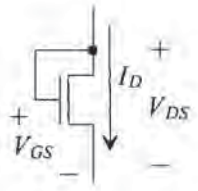



**Figure 9.5** Gate-drain connected MOSFET. Also known as a diode-connected MOSFET.

*PMOS Square-Law Equations*

The PMOS equivalents of Eqs. (9.1) and (9.2) are (for completeness)

$$I_D = \frac{KP_p}{2} \cdot \frac{W}{L}(V_{SG} - V_{THP})^2(1 + \lambda(V_{SD} - V_{SD,sat})) \tag{9.8}$$

and

$$V_{SD,sat} = V_{SG} - V_{THP} \tag{9.9}$$

Note that all we did was swap the subscripts of the symbols used in the NMOS equations. Using this notation, the terminal currents and voltages of the MOSFET (both NMOS and PMOS) **are always positive**.

*Qualitative Discussion*

To develop some intuitive understanding for MOSFET operation, let's look at the circuits in Fig. 9.6. In the following discussion it is assumed that the MOSFETs are operating in the saturation region and that the terminal voltages ($V_{GS}$ and $V_{DS}$) of the device do not exceed the power supply rails.

Imagine injecting a current into the drain of the NMOS device in Fig. 9.6a. What happens to the device's drain current? Answer: it goes up. What happens to the device's $V_{DS}$? Answer: it goes up. If we hold $V_{GS}$ constant then, as seen in Eq. (9.1), we must see an increase in $V_{DS}$ if $I_D$ increases. Stealing current from the NMOS's drain (pulling more of

IPR2023-00697
Theta EX2019

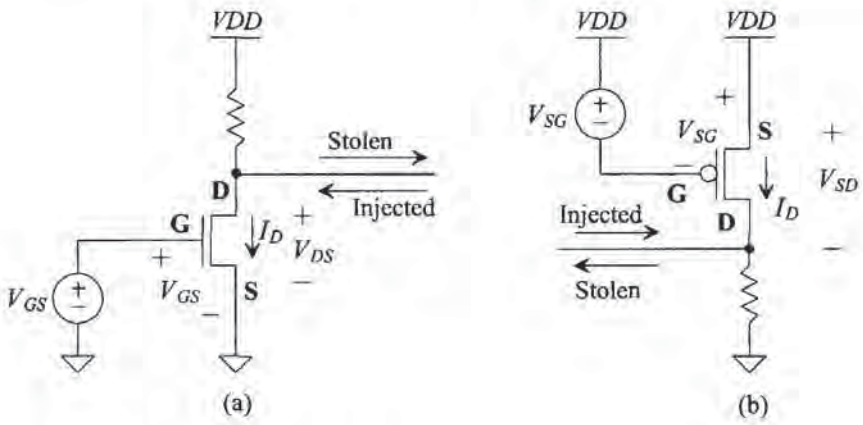



**Figure 9.6** Movement of voltages and currents in a MOSFET, see text.

the current flowing down from the resistor away from the MOSFET's drain terminal) results in both the drain current and $V_{DS}$ going down (until the device enters the triode region and ultimately turns off). Make sure that these statements are clearly understood. We will frequently sum or take the difference of currents using MOSFETs; how the voltages and currents change should be "felt" when looking at an analog design, without referring to the equations.

For the PMOS device in Fig. 9.6b, injecting a current into the drain causes the drain current to go down. In the PMOS device, drain current flows out of the drain terminal of the MOSFET. Injecting a current into the drain results in the drain current going down. When we inject this current into the PMOS drain, the source-drain voltage, $V_{SD}$, decreases as well (meaning that the drain voltage moves towards the power supply voltage *VDD*). Ultimately the device will shut off and the drain current will go to zero. If we steal current from the drain, the drain voltage will move towards ground (noting that $V_{SD} = VDD - V_D$) and the PMOS drain current will increase.

Question: If the channel length modulation parameter, $\lambda$, is zero and the MOSFETs stay in the saturation region, will the drain current change with drain-source voltage? Answer: no, the drain current is then independent of $V_{DS}$, Eq. (9.1). We won't be able to inject or steal a current from the MOSFET without either pushing the MOSFET into triode or moving the drain terminals beyond *VDD* or ground until the MOSFET breaks down. As seen in Eq. (9.1), with $\lambda = 0$, $I_D$ is only dependent on $V_{GS}$ as long as the MOSFET is in saturation.

---

**Example 9.3**

For the circuit in Fig. 9.7, describe qualitatively what will happen if we inject a current at the location seen in the figure of $-1\ \mu A \le I \le 1\ \mu A$. Verify your answer with SPICE. (Use the long-channel CMOS models from Ch. 6.)

This circuit, as we will see in Ch. 20, is a cascode current mirror. M1 and M3 are operating in the saturation region with a drain current of 1 μA. Neglecting body effect (the change in threshold voltage when the source and body of the MOSFET aren't at the same potential), we know $V_{GS1} = V_{GS3}$ when M1 and M3 are sized

IPR2023-00697
Theta EX2019

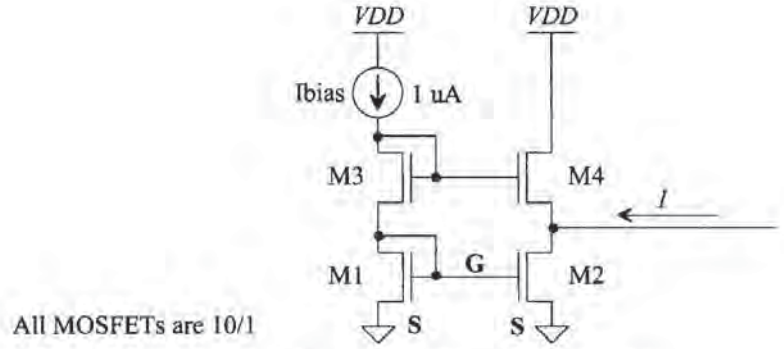



**Figure 9.7** Schematic for Ex. 9.3.

the same and have the same drain currents. Because the gates and sources of M2 and M1 are physically tied together, we can write $V_{GS1} = V_{GS2} = V_{GS3} = V_{GS4}$, meaning that 1 μA flows in all MOSFETs.

For the injected current seen in the figure, a positive 1 μA indicates that current flows to the left into the drain of M2 (the source of M4). A positive current also indicates that we are injecting a current into the node. A −1 μA indicates that current flows to the right out of the node (we are stealing current from the node).

If we inject 1 μA into the drain of M2, then M2's drain voltage and M4's source voltage increase causing $V_{GS4}$ to decrease. Since M2 wants to (meaning its $V_{GS}$ is setting) sink a current of 1 μA, our injected current goes entirely through M2 to ground. M4 will shut off ($V_{GS4} \leq V_{THN}$).

If we steal 1 μA from the node ($I = -1$ μA), then the drain voltage of M2 moves downwards towards ground. Then M4 supplies 2 μA of current ($V_{GS4}$ increases).

Figure 9.8 shows the SPICE simulation results. As discussed, M4 starts to shut off (its gate-source voltage falls below $V_{THN}$) when we inject 1 μA of current. When we steal 1 μA of current (the left side of the plot), $V_{GS4}$ increases to supply 2 μA of current.

To look at the currents flowing in the circuit, zero-volt DC sources can be added. Then, in SPICE, the current through these added sources can be plotted. An example is seen in Fig. 9.9. Simulating the circuit in Fig. 9.7 with the added DC source, we can plot the current through M4, as seen in Fig. 9.10. As mentioned, the drain current shuts off when 1 μA is injected into the node, and it goes to 2 μA when we pull (steal) 1 μA from the node. To monitor the current flowing in M2, we can either add another zero-volt DC source or simply move the injection point to the top of the added DC voltage source in Fig. 9.9.

It's highly recommended that the reader simulate the circuit in Fig. 9.7 under various operating conditions until its operation is fully understood. ■

IPR2023-00697
Theta EX2019

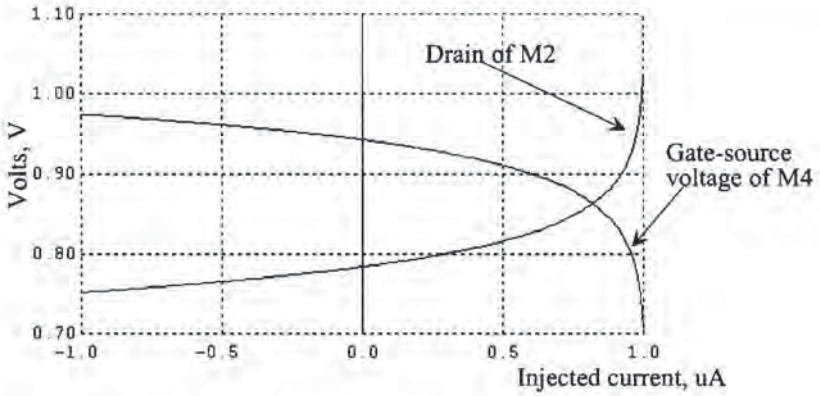



**Figure 9.8** SPICE simulations verifying the discussions in Ex. 9.3.

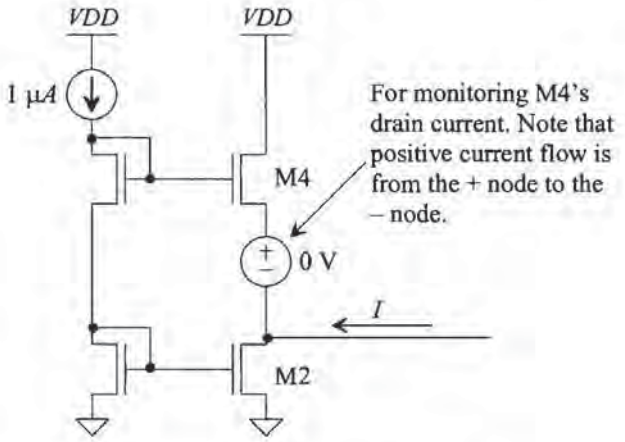



**Figure 9.9** Adding zero volt sources to monitor currents in SPICE.

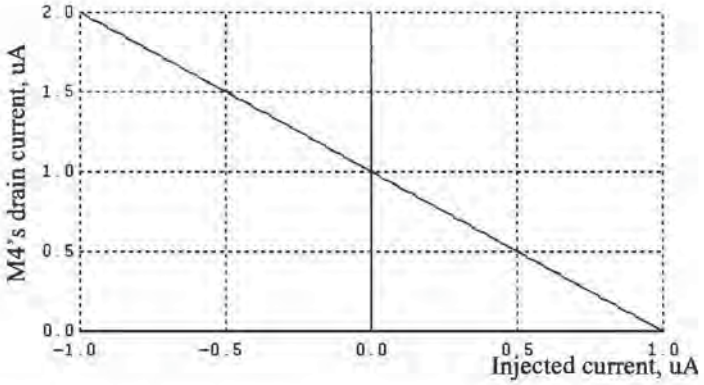



**Figure 9.10** How the drain current of M4 changes with injected current.

IPR2023-00697
Theta EX2019

*Threshold Voltage and Body Effect*

For an NMOS device, the threshold voltage increases when the source is at a higher potential than the NMOS body (the p-substrate, or ground, in this book). This change in threshold voltage is called the *body effect*. A simple example of a MOSFET operating with body effect (an increased threshold voltage) is seen in Fig. 9.7. Both M3 and M4 have a larger threshold voltage than M1 and M2.

From Ch. 6 the change in threshold voltage can be written as a function of the source to bulk potential, $V_{SB}$, Fig. 9.11, as

$$V_{THN}(V_{SB}) = V_{THN0} + \gamma_n\left(\sqrt{2|V_{fp}| + V_{SB}} - \sqrt{2|V_{fp}|}\right) \qquad (9.10)$$

and for the PMOS (again we simply switch the subscripts so that all voltages and currents are positive)

$$V_{THP}(V_{BS}) = V_{THP0} + \gamma_p\left(\sqrt{2|V_{fn}| + V_{BS}} - \sqrt{2|V_{fn}|}\right) \qquad (9.11)$$

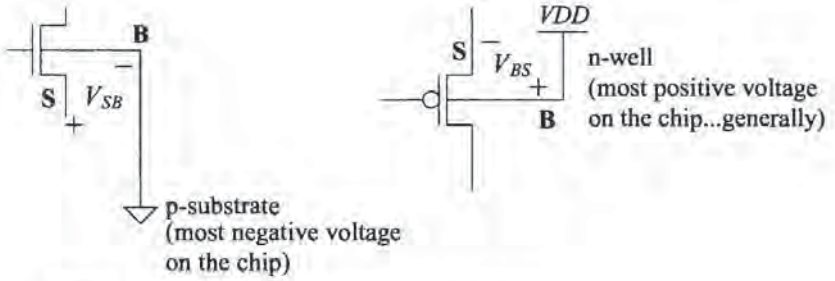



**Figure 9.11** The body connection of a MOSFET.

*Qualitative Discussion*

It's useful to have a feeling for how the threshold voltage changes with increasing source-to-body potential. Figure 9.12 shows a plot of the threshold variation with $V_{SB}$. For large values of $V_{SB}$, the threshold voltage change isn't very significant, while for small values it is significant. The circuit seen in Fig. 9.13 is an example where we might want small variations in the threshold voltage. Because the resistor is large, the gate-source voltage will be close to the threshold voltage. If the threshold voltage didn't vary, then the drain current would be linearly related to the input voltage. This circuit is useful for voltage-to-current conversions.

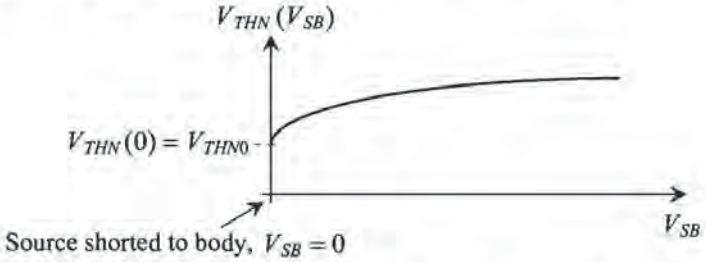



**Figure 9.12** Variation in threshold voltage with source to bulk potential.

IPR2023-00697
Theta EX2019

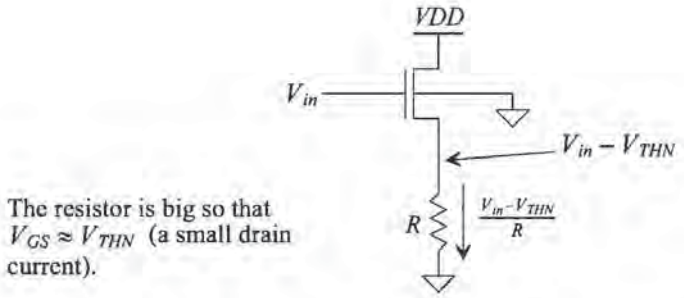



**Figure 9.13** An example of a circuit where body effect is important.

**Example 9.4**

Simulate the operation of the voltage-to-current converter in Fig. 9.13 if the size of the MOSFET is 10/1 and the resistor is 10MEG. Use long-channel devices.

Figure 9.14a shows how the current varies through the MOSFET with $V_{in}$ changing from 1 V to 5 V. We start at 1 V $V_{in}$ because the current will shut off when $V_{in} < V_{THN}$, resulting in a large nonlinearity. Looking at the linearity of the current, it looks really good (straight). However, in Fig. 9.14b we take the derivative of the line in (a) to see the slope. The variation in the slope is approximately 20%. For precision design of voltage-to-current converters, the variation in the linearity of the threshold voltage can be a limiting factor. We should note that channel length modulation also contributes to nonlinearities (so increasing the length, to increase $r_o$, of the device can help improve the linearity).

Question: what is the ideal slope for Fig. 9.14; that is, what is the ideal value of $dI/dV_{in}$? Answer: $1/R$ or, in this example, $100 \times 10^{-9}\ A/V$. ■

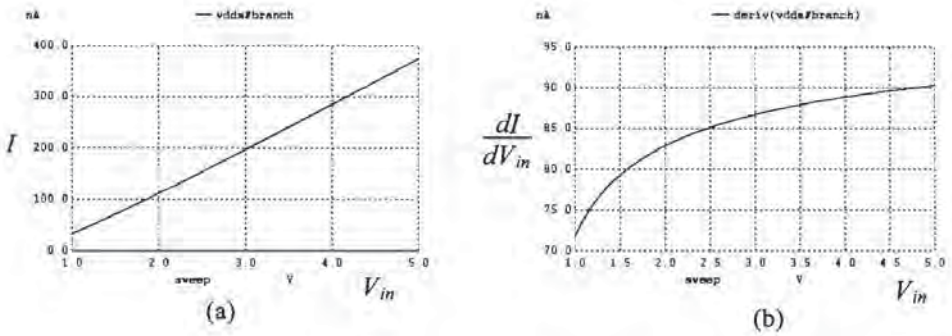



**Figure 9.14** (a) Current flowing in the circuit of Fig. 9.13 and (b) its linearity.

IPR2023-00697
Theta EX2019

There isn't any way to eliminate the body-effect in an n-well process where the NMOS device body is the substrate (though, if a triple well process is used, the NMOS devices can sit in their own wells). However, the PMOS devices can sit in their own wells and have separate bodies (and so we can design without body effect).

*The Triode Region*

The relationship between $I_D$, $V_{GS}$, and $V_{DS}$ for an NMOS device operating in the triode region is

$$I_D = KP_n \frac{W}{L} \cdot \left( (V_{GS} - V_{THN}) V_{DS} - \frac{V_{DS}^2}{2} \right) \tag{9.12}$$

where $V_{GS} \geq V_{THN}$ and $V_{DS} \leq V_{DS,sat}$ $(= V_{GS} - V_{THN})$. The equivalent expression for the PMOS device is

$$I_D = KP_p \frac{W}{L} \cdot \left( (V_{SG} - V_{THP}) V_{SD} - \frac{V_{SD}^2}{2} \right) \tag{9.13}$$

where $V_{SG} \geq V_{THP}$ and $V_{SD} \leq V_{SD,sat}$ $(= V_{SG} - V_{THP})$.

We said earlier, in Fig. 9.4 and the associated discussion, that the MOSFET looks like a resistor when it is operating in the triode region. To estimate the value of the resistance, we can use

$$R_{ch}^{-1} = \frac{\partial I_D}{\partial V_{DS}} = KP_n \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN}) - KP_n \cdot \frac{W}{L} \cdot V_{DS} \tag{9.14}$$

or

$$R_{ch} = \frac{1}{KP_n \cdot \frac{W}{L} \cdot (V_{DS,sat} - V_{DS})} \tag{9.15}$$

If $V_{DS,sat} >> V_{DS}$, this equation can be written as

$$R_{ch} \approx \frac{1}{KP_n \cdot \frac{W}{L} (V_{GS} - V_{THN})} \tag{9.16}$$

*The Cutoff and Subthreshold Regions*

For general design, we normally assume that the device is off (meaning zero drain current) when $V_{GS} < V_{THN}$ or $V_{SG} < V_{THP}$. However, it would be more correct to say that the device is operating in the subthreshold region (instead of the strong inversion region we've discussed so far in this chapter). The subthreshold current of a MOSFET operating in the active region (the amplifying region, which is similar to the saturation region for a MOSFET operating in strong inversion) is modeled using

$$I_D = I_{D0} \cdot \frac{W}{L} \cdot e^{(V_{GS} - V_{THN})/nV_T} \tag{9.17}$$

$V_{GS} \leq V_{THN}$ and $V_{DS} > 4V_T$ (remembering the thermal voltage, $V_T$, is $kT/q$ or 26 mV at room temperature). The current $I_{D0}$ is the (scaled) current that flows when $V_{GS} - V_{THN} = 0$ (the gate-source voltage is equal to the threshold voltage). Note that Eq. (9.17) shows no dependence on $V_{DS}$. This indicates that the MOSFET has infinite output resistance when operating in the active region (which isn't the case, and so this equation has limitations).

IPR2023-00697
Theta EX2019

### 9.1.2 Small Signal Models

A quick note concerning symbols: throughout the book we'll represent a signal containing both AC and DC components by a lowercase letter with uppercase subscripts. AC signals are represented with both lowercase letters and subscripts, while DC signals are represented with both uppercase letters and subscripts, see Fig. 9.15. (Note that if the maximum value of $v_{GS}$ is *VDD* in Fig. 9.15, then the MOSFET is either in the saturation region or off [$V_{GS} < V_{THN}$].)

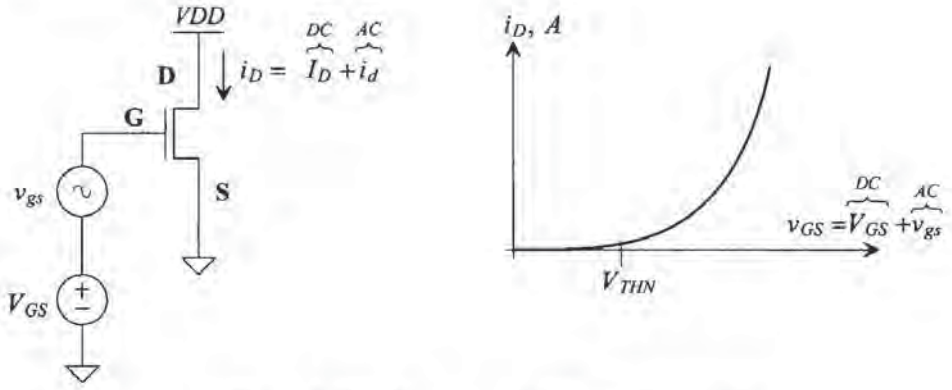



**Figure 9.15** IV curves of a MOSFET in saturation.

Small-signal models are used to calculate AC gains. Figure 9.16 shows the basic idea. We adjust the DC gate-source voltage, $V_{GS}$, to a value that corresponds to a DC drain current $I_D$. At this bias point, we apply a small AC signal where $|v_{gs}| << V_{GS}$ and $|i_d| << I_D$. Because the signals are small, the change in drain current, $i_d$, with gate voltage, $v_{gs}$ is essentially linear (as seen in the blown-up view in Fig. 9.16). If our AC signal amplitudes get comparable to the DC operating (or bias) points, we get high nonlinearity (which makes feedback necessary for any highly linear amplifier).

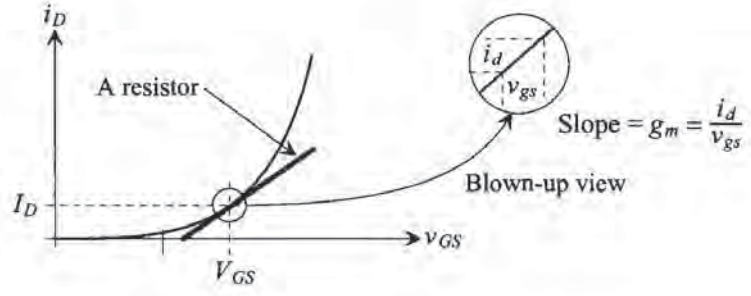



**Figure 9.16** How small-signal parameters are calculated.

IPR2023-00697
Theta EX2019

We often treat DC voltage sources as shorts when doing AC analysis (and current sources as opens). Looking at Fig. 9.15, we see that it's impossible for the AC signal to generate a voltage across the DC source. The voltage across the DC source is fixed with zero AC component so that the DC voltage source is an AC short.

Finally note that performing a small-signal analysis consists of the following steps:

(1) Calculating the bias point of the circuit using the DC equations from Sec. 9.1.1.

(2) Using the DC values from (1) to calculate the small-signal parameters. Small-signal AC parameters are always a function of the DC operating point.

(3) Replacing the active elements (e.g., MOSFETs) with their small-signal models. At the same time, the DC sources are removed (that is, short out all DC voltage sources and open up all DC current sources).

An AC analysis doesn't include any DC voltages or currents. For example, suppose we have an (AC) $v_{gs}$ = 1 mV. It *doesn't make sense* to say that the MOSFET is off because $v_{gs} < V_{THN}$. To perform a small-signal analysis (to calculate small-signal parameters), the MOSFETs are in saturation or triode (meaning $V_{GS} > V_{THN}$).

*Transconductance*

An extremely important parameter in analog design is a device's transconductance, $g_m$. The $g_m$ of a device is an AC small-signal parameter that relates the AC gate voltage to the AC drain current, that is,

$$i_d = g_m \cdot v_{gs} \tag{9.18}$$

From Figs. 9.15 and 9.16, $g_m$ is simply the slope of the line at the intersection of the DC operating values $V_{GS}$ and $I_D$. Using Eq. (9.1), without concerning ourselves with channel-length modulation, we can write

$$i_D = i_d + I_D = \frac{KP_n}{2} \cdot \frac{W}{L} \cdot \left( \overbrace{v_{gs} + V_{GS}}^{v_{GS}} - V_{THN} \right)^2 \tag{9.19}$$

To find the slope ($g_m$) of the $i_D$-$v_{GS}$ curve at the fixed bias points $V_{GS}$ and $I_D$ (Fig. 9.16), we take the derivative of this equation with respect to the x-axis variable ($v_{GS}$)

$$g_m = \left[ \frac{\delta i_D}{\delta v_{GS}} \right]_{V_{GS} = \text{constant}}^{I_D = \text{constant}} = KP_n \cdot \frac{W}{L} \cdot (v_{gs} + V_{GS} - V_{THN}) \tag{9.20}$$

If we remember

$$\beta_n = KP_n \cdot \frac{W}{L} \text{ and } |v_{gs}| << V_{GS} \tag{9.21}$$

then we can write

$$g_m = \beta_n \left( \overbrace{V_{GS} - V_{THN}}^{V_{DS,sat}} \right) = \sqrt{2\beta_n I_D} \tag{9.22}$$

The key points are that $g_m$ goes up as the root of the drain current and linearly with $V_{DS,sat}$.

IPR2023-00697
Theta EX2019

**Example 9.5**
Calculate the DC and AC voltages and currents for the circuit seen in Fig. 9.17. Use the long-channel MOSFET parameters from Ch. 6.

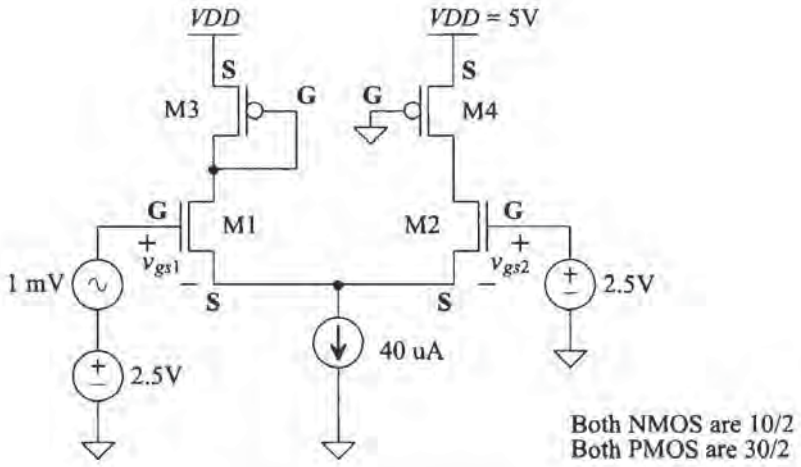



**Figure 9.17** Circuit discussed in Ex. 9.5.

We begin by looking at the DC operating point. The gates of both M1 and M2 are at 2.5V. This is necessary to keep them turned on. Since the sources of M1 and M2 are physically tied together, $V_{GS1} = V_{GS2}$ and $I_{D1} = I_{D2} = 20\ \mu A$. Rewriting Eq. (9.1), neglecting channel-length modulation, results in

$$V_{GS} = \overbrace{\sqrt{\frac{2I_D}{KP_n} \cdot \frac{L}{W}}}^{V_{DS,sat}} + V_{THN} \qquad (9.23)$$

Assuming both M1 and M2 are operating in the saturation region (we'll verify this in a moment), we get

$$V_{GS1} = V_{GS2} = \sqrt{\frac{40}{120} \cdot \frac{2}{10}} + 0.8 = 1.058\ V \rightarrow V_{DS,sat} \approx 250\ mV$$

The source-to-gate voltage for M3 (knowing its drain current is 20 μA) is

$$V_{SG3} = \sqrt{\frac{2 \cdot 20}{40} \cdot \frac{2}{30}} + 0.9 = 1.158\ V \rightarrow V_{SD,sat} \approx 250\ mV$$

The drain potential of M1 and M3 is

$$V_{D1} = V_{D3} = VDD - V_{SG3} = 3.842\ V$$

From Fig. 9.5 and the associated discussion, we know that M3 is in saturation. To see if M1 is in saturation, we use Eq. (9.7)

IPR2023-00697
Theta EX2019

$$V_{D1} \overset{?}{\geq} V_{G1} - V_{THN} \rightarrow 3.842 \geq 1.7\ V \text{ (yes, M1 is in saturation)}$$

Next we look at M4. M4's source-to-gate voltage is 5 V. It's very likely that it is in triode. We know, for a PMOS to be operating in the saturation region, we must have

$$\overbrace{V_{SD}}^{V_S - V_D} \geq \overbrace{V_{SG}}^{V_S - V_G} - V_{THP} \rightarrow V_D \leq V_G + V_{THP} \tag{9.24}$$

M4's gate is grounded ($V_G = 0$), so for M4 to be in saturation, $V_D \leq 0.9\ V$. Since the gate of M2 is at 2.5 V and $V_{GS2} = 1.058$, then its source is 1.442 V, which makes it impossible for M4 to be saturated. To estimate the drain-to-source voltage of M4, we use Eq. (9.13), knowing that M4's drain current is 20 μA and its gate-source voltage is 5 V

$$20 = 40 \cdot \frac{30}{2} \cdot \left( (5 - 0.9) V_{SD} - \frac{V_{SD}^2}{2} \right)$$

which results in $V_{SD}$ = 8.13 mV. The drains of M4 and M2 are then 4.992 V or essentially at *VDD*. Clearly M2 is operating in the saturation region. M4 can be thought of as a resistor with a value of (Eq. [9.16] noting $V_{SD,sat} >> V_{SD}$, that is, 4.1 V >> 8.13 mV)

$$R_{chM4} = \frac{1}{(40\ \mu A/V) \cdot \frac{30}{2} \cdot 4.1} = 407\ \Omega$$

Note that we could have estimated the resistance using $V_{SD}/I_D = 8.13m/20\mu = 407\ \Omega$. Before moving on, let's do an operating point analysis for this circuit (an .op analysis). The SPICE netlist is seen below.

```
*** Example 9.5 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
** for the operating point analysis
print all
*
** for the AC analysis
*plot mag(vd13) mag(vs12) mag(vg1) mag(vd34)
*
** for the transient analysis
*plot vd13
*plot vs12
*plot vg1
.endc

.option scale=1u
.op
**.ac dec 100 1 10k
**.tran 1u 300u

VDD   VDD   0   DC   5
VG1   VG1   0   DC   2.5   AC   1m   SIN 2.5 1m 10k
VG2   VG2   0   Dc   2.5
Ibias VS12  0   DC   40u
```

IPR2023-00697
Theta EX2019

```
M1    VD13  VG1   VS12  0     NMOS L=2 W=10
M2    VD24  VG2   VS12  0     NMOS L=2 W=10
M3    VD13  VD13  VDD   VDD   PMOS L=2 W=30
M4    VD24  0     VDD   VDD   PMOS L=2 W=30

.MODEL NMOS NMOS LEVEL = 3
+ TOX   = 200E-10          NSUB  = 1E17       GAMMA = 0.5
+ PHI   = 0.7              VTO   = 0.8        DELTA = 3.0
+ UO    = 650              ETA   = 3.0E-6     THETA = 0.1
+ KP    = 120E-6           VMAX  = 1E5        KAPPA = 0.3
+ RSH   = 0                NFS   = 1E12       TPG   = 1
+ XJ    = 500E-9           LD    = 100E-9
+ CGDO  = 200E-12          CGSO  = 200E-12    CGBO  = 1E-10
+ CJ    = 400E-6           PB    = 1          MJ    = 0.5
+ CJSW  = 300E-12          MJSW  = 0.5
*
.MODEL PMOS PMOS LEVEL = 3
+ TOX   = 200E-10          NSUB  = 1E17       GAMMA = 0.6
+ PHI   = 0.7              VTO   = -0.9       DELTA = 0.1
+ UO    = 250              ETA   = 0          THETA = 0.1
+ KP    = 40E-6            VMAX  = 5E4        KAPPA = 1
+ RSH   = 0                NFS   = 1E12       TPG   = -1
+ XJ    = 500E-9           LD    = 100E-9
+ CGDO  = 200E-12          CGSO  = 200E-12    CGBO  = 1E-10
+ CJ    = 400E-6           PB    = 1          MJ    = 0.5
+ CJSW  = 300E-12          MJSW  = 0.5

.end
```

A portion of the operating point analysis output is

```
vd13 = 3.854977e+00   (we hand calculated 3.842)
vd24 = 4.989641e+00   (we hand calculated 4.992)
vs12 = 1.171189e+00   (we hand calculated 1.442)
```

The only significant difference between our hand calculations and the simulation results is the potential calculated for the sources of M1/M2. The smaller value in the simulation is because we didn't include the body effect in our calculations. We used $V_{THN0} = 0.8\ V$ when the actual threshold voltage was closer to $1.1\ V$.

Let's now turn our attention towards the AC analysis. The transconductance of M1 and M2 is

$$g_{m1} = g_{m2} = \sqrt{2 \cdot KP_n \frac{W}{L} \cdot I_D} = \sqrt{2 \cdot 120\mu \cdot \frac{10}{2} \cdot 20\mu} \approx 150\ \mu A/V$$

The transconductance of M3 is

$$g_{m3} = \sqrt{2 \cdot KP_p \frac{W}{L} \cdot I_D} = \sqrt{2 \cdot 40\mu \cdot \frac{30}{2} \cdot 20\mu} \approx 150\ \mu A/V$$

M4 is operating in the triode region and so we think of it as a resistor (407 Ω). Figure 9.18 shows the simplified AC schematic of Fig. 9.17. Notice how we can replace M3 with a resistor of $1/g_{m3}$. This is because the AC voltage across M3 is $v_{sg3} = v_{sd3}$ and the AC current through it is $i_d$ or

IPR2023-00697
Theta EX2019

$$\frac{1}{g_m} = \frac{v_{sd}}{i_d} = \frac{v_{sg}}{i_d} \tag{9.25}$$

A gate-drain connected MOSFET with a current flowing through it is always in saturation, as discussed earlier, and can be thought of as a small-signal resistance of $1/g_m$ (again, remember this).

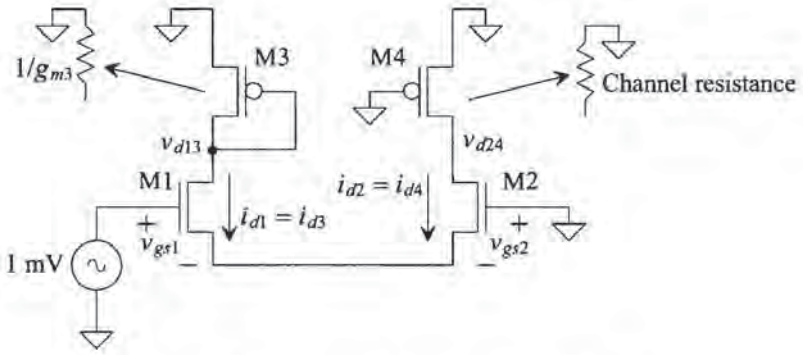



**Figure 9.18** AC schematic for the circuit in Fig. 9.17.

When we talk about negative currents or voltages in an AC small-signal analysis, we are indicating that the overall AC + DC signal is decreasing. For example, we'll see in a moment that the AC small-signal drain current in M2, Fig. 9.18, is negative. This means that the current may be going from 20 μA (its DC operating point) to, say, 19.995 μA with an AC input. We would then say that the AC drain current is, $i_{d2}$, − 5 nA.

To analyze the circuit, we can begin by writing

$$1\ mV = v_{gs1} - v_{gs2} = \frac{i_{d1}}{g_{m1}} - \frac{i_{d2}}{g_{m2}}$$

We know that $g_{m1} = g_{m2}$ and, from Fig. 9.18,

$$i_{d1} = -i_{d2}$$

so

$$v_{gs1} = -v_{gs2} = 0.5\ mV$$

The sources of M1 and M2 are at an AC voltage of 0.5 mV. The AC drain currents are

$$i_{d1} = i_{d3} = -i_{d2} = -i_{d4} = g_{m1} \cdot v_{gs1} = (150\ \mu A/V) \cdot 0.5\ mV = 75\ nA$$

This means the overall (AC + DC) drain current of M1/M3 is

$$i_{D1} = 20\ \mu A + 0.075 \sin 2\pi f t$$

where $f$ is the frequency of our 1 mV input signal and

$$i_{D2} = 20\ \mu A - 0.075 \sin 2\pi f t$$

IPR2023-00697
Theta EX2019

noting that $i_{D1}$ and $i_{D2}$ must sum to the DC bias of 40 μA.

The (AC) drain voltage of M1 and M3 is

$$v_{d1} = v_{d3} = -i_{d1} \cdot \frac{1}{g_{m3}} = -\frac{75\ nA}{150\ \mu A/V} = -0.5\ mV$$

The voltages on the drains of M2 and M4 are

$$v_{d2} = v_{d4} = -i_{d4} \cdot R_{chM4} = (75\ nA) \cdot 407 = 0.03\ mV$$

We'll verify these voltages with SPICE in a moment. This lengthy example illustrates several key concepts that will be used throughout the analysis and design of analog CMOS circuits in this book. Before moving on, make sure that the concepts are clear and well understood (SPICE can be very helpful for this).

Question: How would we look at the SPICE simulated currents (AC and DC) flowing in the MOSFETs in Fig. 9.17? Answer: add zero-volt voltage sources as seen in Fig. 9.9 and the associated discussion. ■

*AC Analysis*

We can perform a small-signal analysis using a SPICE ".ac" statement. Just like in our hand calculations, SPICE first calculates the operating point using the DC sources (no AC sources are present in the circuit). Then SPICE replaces the active devices with their small-signal equivalent circuits (no DC sources in the circuit, now just the AC sources). The key point here is that SPICE assumes the user knows that the AC components should be much smaller than the DC components. If we were to simulate, using an AC analysis, the operation of the circuit in Fig. 9.17, but used a 1,000 V AC input instead of a 1 mV AC input, SPICE would simply scale all of the outputs (the simulation won't tell the user that the AC input is too big).

The possible syntax for an .ac analysis statement is

```
.AC DEC ND FSTART FSTOP
.AC OCT NO FSTART FSTOP
.AC LIN NP FSTART FSTOP
```

where DEC stands for decade (our x-axis in an .ac analysis is frequency) and ND is the number of points per decade (OCT stands for octave and LIN stands for linear). FSTART and FSTOP are the starting and ending frequencies for the small-signal AC analysis.

We can do a SPICE AC analysis for the circuit in Fig. 9.17 by changing the .op analysis used in the netlist to an AC analysis control statement

```
.ac dec 100 1 10k
```

where we've picked a frequency range of 1 Hz to 10k Hz with 100 points calculated per decade. (The first decade is 1 to 10 Hz, the second, 10 Hz to 100 Hz, etc. If we had used a start frequency of 2 Hz, then the first decade would be 2 Hz to 20 Hz, the second would be 20 Hz to 200 Hz, noting that increasing by a decade is multiplying by 10 and decreasing by a decade is dividing by 10.) The input signal (the gate of M1) is

```
VG1   VG1   0     DC    2.5    AC    1m    SIN 2.5 1m 10k
```

IPR2023-00697
Theta EX2019

The operating point analysis ignores the AC voltage of 1 mV and the sinusoid specification (a sine wave with a DC offset of 2.5 V, a peak amplitude of 1 mV, and a frequency of 10 kHz). The AC analysis uses the DC value to calculate the operating point and the AC value to calculate the small-signal AC voltages and currents. We'll discuss the SIN portion in a moment when we talk about transient analysis. The simulation results are seen in Fig. 9.19. The values should be compared to the values calculated in Ex. 9.5. The only (small) difference is the voltage calculated at the sources of M1 and M2. We calculated 0.5 mV, while SPICE gives 0.42 mV. This is due to the body effect transconductance, $g_{mb}$, which is discussed on the next page.

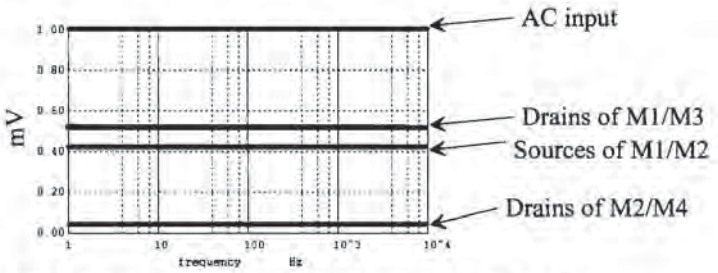



**Figure 9.19** AC simulation results for the circuit in Fig. 9.17.

*Transient Analysis*

The AC analysis is frequently used to look at the frequency response of a circuit. However, as we have just discussed, it will not show large signal nonlinearity. A transient analysis, however, will show large signal nonlinearity (the x-axis is time just like the x-axis on an oscilloscope display). The control statement for a transient analysis is

```
.TRAN TSTEP TSTOP <TSTART <TMAX> <UIC>
```

where TSTEP is the step size (say 1/1,000) of the stop time TSTOP. TSTART is an optional parameter to specify a later starting time. The simulation always starts at 0 seconds. However, in some simulations there are start-up transient signals that we are not interested in. To avoid saving this uninteresting data in our output file (to reduce the output file size), we can specify a start-up time later than 0. In this book we won't use this option. The TMAX parameter specifies a maximum step size for the simulation. If, for example, a sine wave is plotted and it looks *jagged* (meaning that our step size is too coarse), we would use a smaller step size (TMAX) in the simulation to smooth it out. UIC indicates, "Use Initial Conditions." If UIC isn't present in a simulation, SPICE ignores all initial conditions. A sample transient control statement is

```
.TRAN  1n 1000n 0 1n UIC
```

To simulate the circuit in Fig. 9.17 using a transient analysis, we use the input signal source of

```
VG1   VG1   0     DC    2.5    AC    1m    SIN 2.5 1m 10k
```

In a transient analysis, SPICE always ignores the AC component. It ignores the DC component as well if pulse, sinusoid, or some other signal is specified. Note that the SIN specification **has nothing to do with AC analysis**. For the transient simulation of the

IPR2023-00697
Theta EX2019

circuit in Fig. 9.17, we use a sinusoid with a DC offset of 2.5 V, a peak amplitude of 1mV, and a frequency of 10 kHz (picked arbitrarily for this simulation). The simulation results showing the drain potential of M1 and M3 are seen in Fig. 9.20.

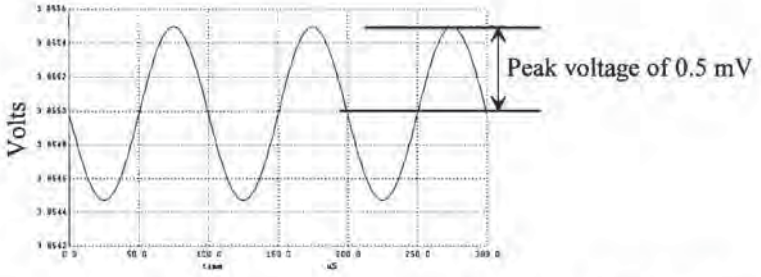



**Figure 9.20** Transient simulation showing the drain voltages of M1 and M3 from Fig. 9.17

*Body Effect Transconductance, $g_{mb}$*

Figure 9.21 shows the setup to determine how the drain current varies with source-to-bulk potential $V_{SB}$. If we raise the potential of the source, we eventually run into the point where the source potential is less than a $V_{THN}$ below the gate potential, and the MOSFET shuts off.

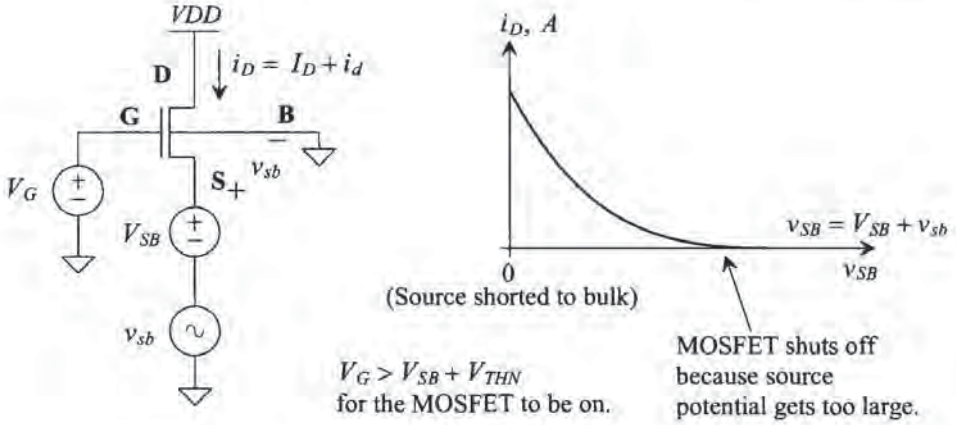



**Figure 9.21** How the drain current changes with body-effect.

Remembering that the body-effect is the variation of the threshold voltage with $V_{SB}$, we can write

$$g_{mb} = \left[\frac{\partial i_D}{\partial v_{SB}}\right]_{V_{SB}\,=\,\text{constant}}^{I_D\,=\,\text{constant}} = \frac{\partial}{\partial v_{SB}}\left[\frac{KP_n}{2}\cdot\frac{W}{L}(V_{GS}-V_{THN})^2\right]_{V_{SB}\,=\,\text{constan}}^{I_D\,=\,\text{constant}} \qquad (9.26)$$

or

IPR2023-00697
Theta EX2019

$$g_{mb} = \overbrace{KP_n \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN})}^{g_m} \cdot \left(-\frac{\partial V_{THN}}{\partial v_{SB}}\right) \tag{9.27}$$

or (noting there is a variation in $v_{GS}$ with $v_{SB}$ but this is simply the forward $g_m$).

$$g_{mb} = g_m \cdot \eta \tag{9.28}$$

The factor $\eta$ describes how the threshold voltage changes with $V_{SB}$ and generally ranges from 0 (no body effect) to 0.5. The minus sign in Eq. (9.27) simply indicates that the AC drain current contributions from changes in the threshold voltage with $V_{SB}$ (body effect) flow in the opposite direction of the contributions from the forward transconductance, $g_m$. Figure 9.22 shows the AC small-signal model of the MOSFET with both small-signal transconductances included.

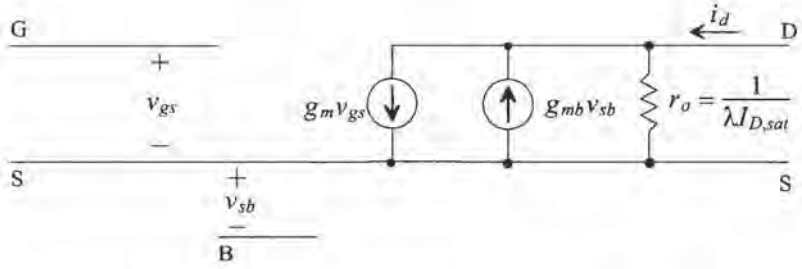



**Figure 9.22** Small-signal MOSFET with both transconductances.

*Output Resistance*

We've already calculated the output resistance of a MOSFET operating in the saturation region in Eq. (9.6) (added to Fig. 9.22). Let's calculate this value again using the circuit seen in Fig. 9.23. We can write

$$r_o^{-1} = \left[\frac{\partial i_D}{\partial v_{DS}}\right]_{V_{DS}\,=\,\text{constant}}^{I_D\,=\,\text{constant}} = \frac{\partial}{\partial v_{DS}}\left(\frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - V_{THN})^2\left(1 + \lambda\left(\overbrace{v_{ds} + V_{DS}}^{v_{DS}} - V_{DS,sat}\right)\right)\right) \tag{9.29}$$

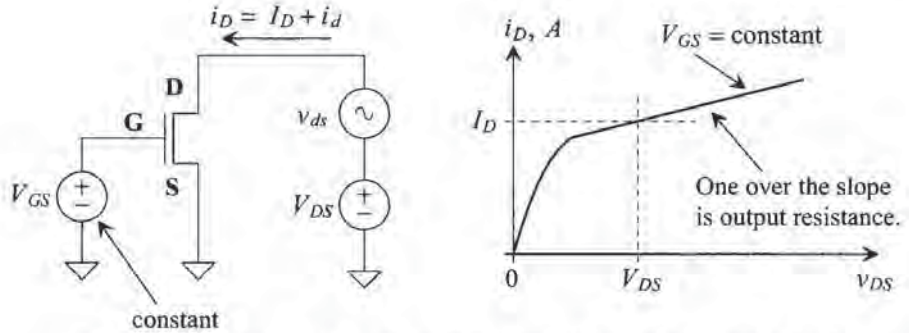



**Figure 9.23** How the drain current changes with drain-to-source voltage.

IPR2023-00697
Theta EX2019

or, once again,

$$r_o = \frac{1}{\lambda I_{D,sat}} \tag{9.30}$$

**Example 9.6**

Determine λ, using simulations, for the MOSFETs in Ex. 9.5.

The $V_{GS}$ (NMOS) and $V_{SG}$ (PMOS) of the MOSFETs in Ex. 9.5 are 1.05 V and 1.15 V, respectively (roughly for 20 μA of bias current). Figure 9.24 shows the IV plots for the MOSFETs and the reciprocal of the derivative of the drain current (which gives us the output resistance). Because the NMOS threshold voltage is 0.8 V and the PMOS threshold voltage is 0.9 V, both MOSFETs have a $V_{DS,sat}$ of 250 mV and a channel length of 2. The channel-length modulation parameter is calculated using

$$\lambda_n = \frac{1}{I_{D,sat} \cdot r_o} = \frac{1}{20\mu \cdot 5MEG} = 0.01\ V^{-1}$$

and

$$\lambda_p = \frac{1}{20\mu \cdot 4MEG} = 0.0125\ V^{-1}$$

noting that the values are approximations. Also note that the output resistance is very dependent on drain-to-source voltage. ■

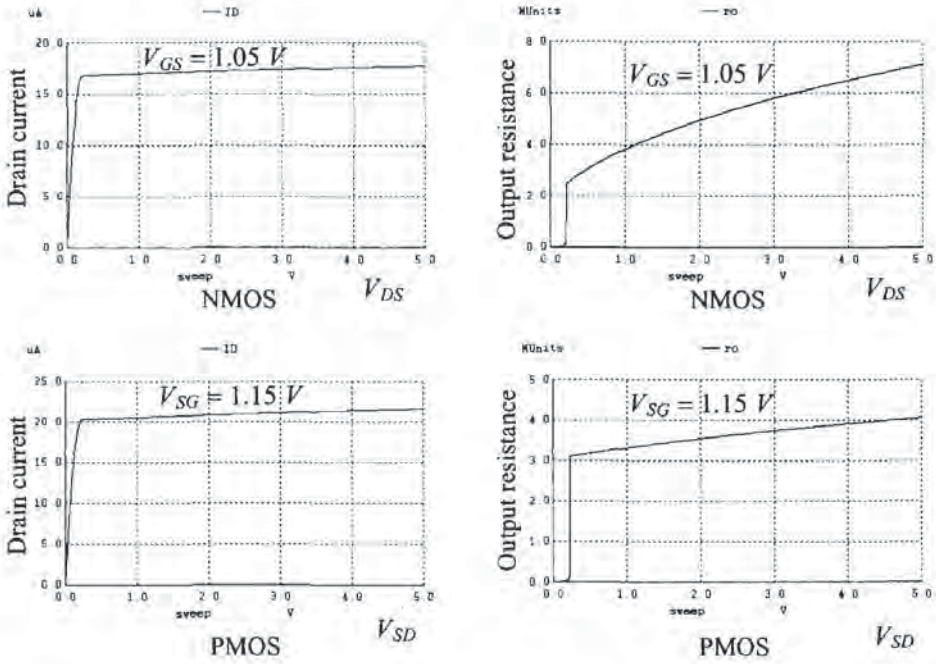



**Figure 9.24** Using simulations to determine lambda.

IPR2023-00697
Theta EX2019

It's of interest to determine how output resistance changes with channel length and $V_{DS,sat}$. Remember

$$I_{D,sat} = \frac{KP_N}{2} \cdot \frac{W}{L} \cdot V_{DS,sat}^2 \tag{9.31}$$

and, from Ch. 6,

$$\lambda \propto \frac{1}{L} \tag{9.32}$$

so we can write

$$r_o \propto \frac{L^2}{V_{DS,sat}^2} \tag{9.33}$$

If the length of the MOSFET is increased, for fixed $V_{GS}$ (equivalent to saying for fixed $V_{DS,sat}$, since $V_{DS,sat} = V_{GS} - V_{THN}$), then the drain current decreases and the output resistance increases. If the length is held constant, then decreasing $V_{DS,sat}$ causes the drain current to decrease and the output resistance to increase. We might think that, to get large output resistance, all we have to do is use a very long-length device. However, as we'll show next, this causes the inherent speed of the MOSFETs to decrease.

*MOSFET Transition Frequency, $f_T$*

Examine the circuit in Fig. 9.25. The drain of the MOSFET is at AC ground (shorted through the DC drain-source voltage). This causes, from the gate terminal, $C_{gs}$ and $C_{gd}$, to appear as though they are in parallel. We can then write

$$v_{gs} = \frac{i_g}{j\omega \cdot (C_{gs} + C_{gd})} \tag{9.34}$$

Knowing $i_d = g_m \cdot v_{gs}$, we can write the current gain of the MOSFET as

$$\left| \frac{i_d}{i_g} \right| = \frac{g_m}{2\pi f \cdot (C_{gs} + C_{gd})} \tag{9.35}$$

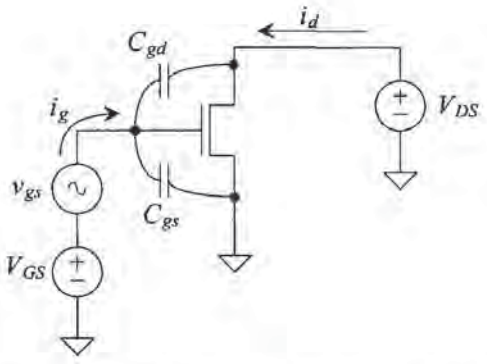



**Figure 9.25** Determining the transition frequency of an NMOS transistor.

IPR2023-00697
Theta EX2019

If we call the frequency where the current gain of the MOSFET is one, Fig. 9.26, the transition frequency, $f_T$ (the transistor transitions from an amplifier to an attenuator) and we remember that $C_{gs}\left(=\frac{2}{3}WLC'_{ox}\right) >> C_{gd}$, then we can write (see Eq. [9.22])

$$f_T \approx \frac{g_m}{2\pi C_{gs}} = \frac{3KP_n \cdot (V_{GS} - V_{THN})}{4\pi \cdot L^2 C'_{ox}} = \frac{3\mu_n}{4\pi} \cdot \frac{V_{DS,sat}}{L^2} \qquad (9.36)$$

This equation is *fundamentally important*. To get high speed, we need to use minimum channel lengths and design with a large $V_{DS,sat}$. However, as seen in Eq. (9.33), using minimum lengths results in lower output resistances (and, as we'll see later, lower gain). What this indicates is a constant gain-bandwidth product (higher speed resulting in lower gain). In using a large $V_{DS,sat}$, the MOSFETs enter the triode region earlier (resulting in reduced output swing in amplifiers or mirrors). Note that the $f_T$ for the PMOS device is inherently smaller than that of the NMOS device due to the lower value of hole mobility.

For short-channel devices, the mobility is no longer constant but starts to decrease (velocity saturation as discussed in Ch. 6) with decreasing length (increasing electric field between the drain and channel). Because of this, we treat the term $\mu_n/L$ as a relatively constant value and rewrite Eq. (9.36) as

$$f_T \propto \frac{V_{DS,sat}}{L} \text{ (Short-channel devices)} \qquad (9.37)$$

Again, for high-speed, we need to use the smallest possible channel length and large $V_{DS,sat}$.

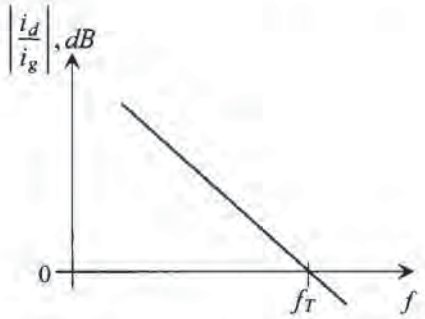



**Figure 9.26** The transition frequency of a MOSFET.

*General Device Sizes for Analog Design*

Based on Eqs. (9.33) and (9.36), we can make some general statements about selecting device $L$, $W$, and $V_{DS,sat}$. For general analog design, use an $L$ of 2–5 times minimum (in this book $L_{min}$ = 1, since we scale the MOSFET sizes by either 1 μm [long-channel MOSFETs] or 50 nm [short-channel MOSFETs] in the SPICE netlists or layout). We'll use an $L$, for analog design, of 2 as a good trade-off between speed and gain.

For general design, use a $V_{DS,sat}$ of 5% of *VDD*. For our long-channel MOSFETs (*VDD* = 5 V), we'll use a $V_{DS,sat}$ of 250 mV, while for the short-channel process (*VDD* = 1V), we'll use 50 mV. Table 9.1 shows the parameters for general analog design using the long-channel process discussed in this chapter.

IPR2023-00697
Theta EX2019

**Table 9.1** Typical parameters for analog design using the *long-channel* CMOS process discussed in this book. Note that the parameters may change with temperature or drain-to-source voltage (e.g., Fig. 9.24).

| *Long*-channel MOSFET parameters for general analog design<br>*VDD* = 5 V and a scale factor of **1 μm** (*scale* = 1e–6) | | | |
|---|---|---|---|
| **Parameter** | **NMOS** | **PMOS** | **Comments** |
| Bias current, $I_D$ | 20 μA | 20 μA | Approximate |
| $W/L$ | 10/2 | 30/2 | Selected based on $I_D$ and $V_{DS,sat}$ |
| $V_{DS,sat}$ and $V_{SD,sat}$ | 250 mV | 250 mV | For sizes listed |
| $V_{GS}$ and $V_{SG}$ | 1.05 V | 1.15 V | No body effect |
| $V_{THN}$ and $V_{THP}$ | 800 mV | 900 mV | Typical |
| $\partial V_{THN,P}/\partial T$ | −1 mV/C° | −1.4 mV/C° | Change with temperature |
| $KP_n$ and $KP_p$ | 120 μA/V² | 40 μA/V² | $t_{ox}$ = 200 Å |
| $C'_{ox} = \varepsilon_{ox}/t_{ox}$ | 1.75 *f*F/μm² | 1.75 *f*F/μm² | $C_{ox} = C'_{ox}WL \cdot (scale)^2$ |
| $C_{oxn}$ and $C_{oxp}$ | 35 *f*F | 105 *f*F | PMOS is three times wider |
| $C_{gsn}$ and $C_{sgp}$ | 23.3 *f*F | 70 *f*F | $C_{gs} = \frac{2}{3}C_{ox}$ |
| $C_{gdn}$ and $C_{dgp}$ | 2 *f*F | 6 *f*F | $C_{gd} = CGDO \cdot W \cdot scale$ |
| $g_{mn}$ and $g_{mp}$ | 150 μA/V | 150 μA/V | At $I_D$ = 20 μA |
| $r_{on}$ and $r_{op}$ | 5 MΩ | 4 MΩ | Approximate at $I_D$ = 20 μA |
| $g_{mn}r_{on}$ and $g_{mp}r_{op}$ | 750 V/V | 600 V/V | Open circuit gain |
| $\lambda_n$ and $\lambda_p$ | 0.01 V⁻¹ | 0.0125 V⁻¹ | At $L$ = 2 |
| $f_{Tn}$ and $f_{Tp}$ | 900 MHz | 300 MHz | For $L$ =2, $f_T$ goes up if $L$ = 1 |

*Subthreshold $g_m$ and $V_{THN}$*

Before leaving the topic of small-signal models, let's derive the forward transconductance of the MOSFET operating in the subthreshold region. Using Eqs. (9.17) and (9.20), we can write

$$g_m = \left[\frac{\delta i_D}{\delta v_{GS}}\right]_{\substack{I_D = \text{constant} \\ V_{GS} = \text{constant}}} = I_{D0} \cdot \frac{W}{L} \cdot e^{\left(\frac{v_{GS}}{V_{GS} + v_{gs}} - V_{THN}\right)/nV_T} \cdot \left(\frac{1}{nV_T}\right) \quad (9.38)$$

If, (as always for a small-signal analysis) $v_{gs} << V_{GS}$, then

$$g_m = \frac{I_D}{nV_T} \quad (9.39)$$

The transconductance increases linearly with bias current when operating in the subthreshold region (compare to Eq. (9.22)). Unfortunately, the speed ($f_T$) of MOSFETs operating in this region is considerably slower. The small currents charging the device's own capacitances limit the speeds to, in general, < MHz. As CMOS technology scales downwards, the inherent speeds increase. This, together with the need for lower power, increases the number of designs operating in, or near, the subthreshold region.

IPR2023-00697
Theta EX2019

As $I_D$ ($V_{GS}$) increases, the MOSFET moves from operating in the subthreshold region, $g_m = I_D/nV_T$ ($g_m$ is exponentially dependent on $V_{GS}$), to moderate inversion, and then to the strong inversion region, $g_m = \sqrt{2I_D\beta_n} = \beta_n(V_{GS} - V_{THN})$ ($g_m$ is linearly dependent on $V_{GS}$). As seen in Eq. (9.16), using small $V_{DS}$ results in a channel resistance, $R_{ch}$ , of $1/g_m$ (assuming that the MOSFET is operating in the *triode* region). Fig. 9.27a shows a plot of $I_D$ against $V_{GS}$ for a MOSFET in the short-channel process discussed later in the chapter. The threshold voltage in (a) is estimated by linearly extrapolating back to the x-axis (again from Eq. [9.16] the slope is $R_{ch}^{-1}$). In (b) we take the derivative of (a) to get the $g_m$ of the device. Knowing $g_m = \beta_n(V_{GS} - V_{THN})$ (for a MOSFET operating in the *saturation* region), we can linearly extrapolate back to the x-axis to estimate when $g_m$ goes to zero (to get $V_{THN}$). The two methods give different results (there are other methods as well, such as taking the derivative of $g_m$ with respect to $V_{GS}$ and looking at the $V_{GS}$ [= $V_{THN}$] when this second derivative of $I_D$ becomes constant). Because, even for small $V_{DS}$ (> a couple of $kT/q$), the MOSFET will be operating in the saturation region when $V_{GS} \approx V_{THN}$ we'll use method (b).

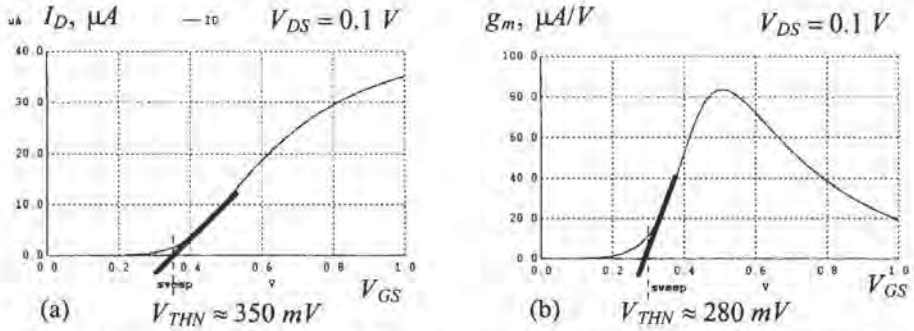



**Figure 9.27** (a) Drain current plotted against gate-source voltage with small Vds and (b) transconductance plotted against gate-source voltage.

### 9.1.3 Temperature Effects

In this section we look at how the drain current of a MOSFET operating in the saturation region changes with temperature. Looking at the long-channel expression for the drain current

$$I_{DS,sat} = \frac{\mu_n \cdot \varepsilon_{ox}}{2t_{ox}} \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN})^2 \tag{9.40}$$

(where we've written $KP_n$ as $\mu_n\varepsilon_{ox}/t_{ox}$), the variables that change with temperature are the mobility and the threshold voltage.

*Threshold Variation with Temperature*

From Ch. 6, Eq. (6.17), we can write

$$V_{THN} = -V_{ms} - 2V_{fp} + \frac{Q'_{b0} - Q'_{ss}}{C'_{ox}} \tag{9.41}$$

noting $Q'_{ss}$ is a constant and $Q'_{b0} = \sqrt{2qN_A\varepsilon_{si}|-2V_{fp}|}$ . Knowing

$$V_{fp} = -\frac{kT}{q}\ln\frac{N_A}{n_i} \text{ and } V_{ms} = V_G - V_{fp} = \frac{kT}{q}\ln\frac{N_{D,poly}}{n_i} - V_{fp} \tag{9.42}$$

IPR2023-00697
Theta EX2019

the change in the threshold voltage with temperature is

$$\frac{\partial V_{THN}}{\partial T} = -\frac{k}{q} \cdot \ln \frac{N_{D,poly}}{N_A} + \frac{Q'_{b0}}{C'_{ox} \cdot 2T} \approx -\frac{k}{q} \cdot \ln \frac{N_{D,poly}}{N_A} \tag{9.43}$$

The term $k/q$ is the change in the thermal voltage with temperature, that is,

$$\frac{\partial V_T}{\partial T} = \frac{\partial}{\partial T}\left(\frac{kT}{q}\right) = \frac{k}{q} = 0.085\ mV/C^\circ \tag{9.44}$$

(keeping in mind that we don't confuse the temperature behavior of the thermal voltage, $V_T$, with the behavior of the threshold voltage, $V_{THN}$). If $N_{D,poly} = 10^{20}$ and $N_A = 10^{15}$ (a ballpark value for a long-channel MOSFET), then

$$\frac{\partial V_{THN}}{\partial T} \approx -1\ mV/C^\circ \tag{9.45}$$

If $N_A = 10^{17}$ ($N_A$ scales upwards as the channel length of the CMOS technology decreases, see Table 6.3 in Ch. 6), then

$$\frac{\partial V_{THN}}{\partial T} \approx -0.6\ mV/C^\circ \tag{9.46}$$

noting that the threshold voltage decreases with increasing temperature, while the thermal voltage increases with increasing temperature. The temperature coefficient of the threshold voltage is defined as

$$TCV_{THN} = \frac{1}{V_{THN}} \cdot \frac{\partial V_{THN}}{\partial T} \tag{9.47}$$

so that the threshold voltage can be written as a function of temperature as

$$V_{THN}(T) = V_{THN}(T_0) \cdot (1 + TCV_{THN} \cdot (T - T_0)) \tag{9.48}$$

where the threshold voltage is measured at the temperature $T_0$, Fig. 9.28. The units for temperature can be Kelvin or Celsius because of the difference used in this equation. For our long-channel process with $V_{THN}$ = 0.8 V, the temperature coefficient is –1,250 ppm/$C^\circ$. For a short-channel process with $V_{THN}$ = 0.28 V, we get a temperature coefficient of –2,143 ppm/$C^\circ$. For a 100-degree increase in temperature, our short-channel threshold voltage decreases by 60 mV, resulting in a larger $I_{off}$ (a factor of 10 larger if the subthreshold slope is the ideal 60 mV/decade at room temperature).

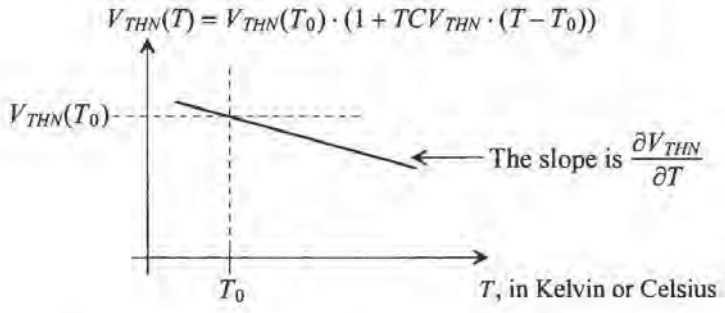



**Figure 9.28** Temperature dependence of the threshold voltage.

IPR2023-00697
Theta EX2019

*Mobility Variation with Temperature*

The reduction in mobility with increasing temperature is modeled using

$$\mu(T) = \mu(T_0) \cdot \left(\frac{T_0}{T}\right)^{1.5} \tag{9.49}$$

and so it follows that the reduction in the transconductance parameter with increasing temperature is

$$KP(T) = KP(T_0) \cdot \left(\frac{T_0}{T}\right)^{1.5} \tag{9.50}$$

where, once again, the mobility is measured at temperature $T_0$. Note that both $T$ and $T_0$ have units of Kelvin. The change in the MOSFET transconductance parameter, $KP$, with temperature around $T_0$ is

$$\left[\frac{\partial KP(T)}{\partial T}\right]_{T_0=\text{ constant}} = KP(T_0) \cdot (-1.5)\left(\frac{T_0}{T}\right)^{2.5} \cdot \frac{1}{T_0} \tag{9.51}$$

The temperature coefficient of the transconductance parameter around the temperature $T_0$ is then

$$\frac{1}{KP(T)} \cdot \frac{\partial KP(T)}{\partial T} = -\frac{1.5}{T} \tag{9.52}$$

The transconductance parameter at a particular temperature close to $T_0$ is given by

$$KP(T) = KP(T_0) \cdot \left(1 - 1.5 \cdot \frac{T - T_0}{T}\right) \tag{9.53}$$

As temperature increases, the mobility and $KP$ decrease. Note that our derivation used the change in the temperature (the slope of the line) around the measured temperature $T_0$ (just like we used for small-signal analysis). Equations (9.52) and (9.53) work well for hand calculations. However, for wide temperature changes, we'll need to use simulations (which can include the nonlinear variations in the mobility). Note, again, that unlike Eq. (9.48) where a difference in temperatures is present, and so either Kelvin or Celsius can be used, we must use Kelvin when using Eq. (9.53).

*Drain Current Change with Temperature*

We now know that as temperature goes up, the threshold voltage and mobility go down. A decrease in mobility, see Eq. (9.40), causes the drain current to go down. At the same time, a decrease in threshold voltage causes the drain current to go up. At low $V_{GS}$, the changes in $V_{THN}$ dominate and the drain current increases with increasing temperature. At higher $V_{GS}$, the mobility dominates and the drain current decreases with increasing temperature. When the effects cancel, the drain current doesn't change with temperature, Fig. 9.29. For a long-channel device like the one seen in Fig. 9.29, again, $\partial V_{THN}/\partial T = -1\ mV/C°$.

Figure 9.30a shows how the drain current changes with temperature in a short-channel CMOS technology. Again, we see where the effects of the mobility changing with temperature cancel the effects of the threshold voltage changing with temperature. Looking at Fig. 9.30b, we see that, for a constant current of 10 μA, $V_{GS}$ changes at a rate of $\approx -0.6\ mV/C°$ ($= \partial V_{THN}/\partial T$ because the $V_{GS}$ is small).

IPR2023-00697
Theta EX2019

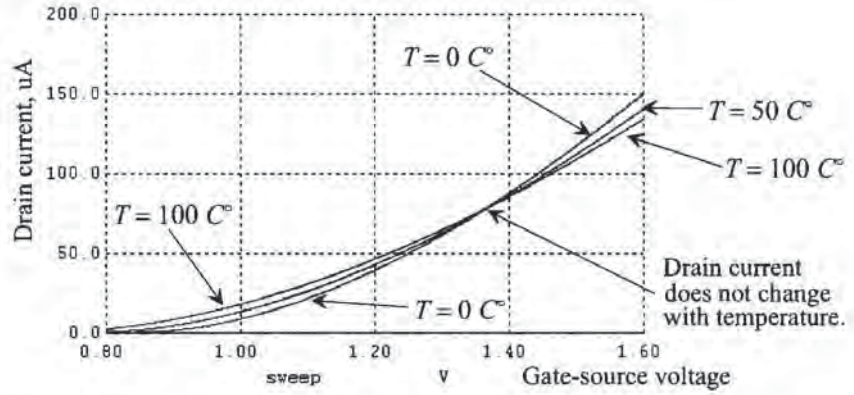
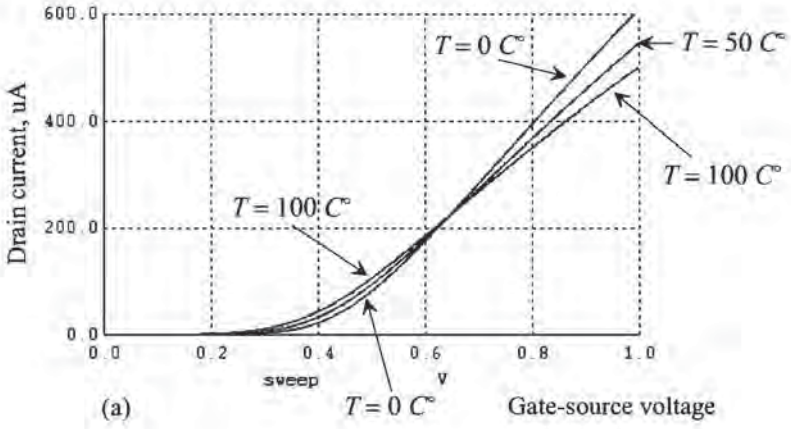
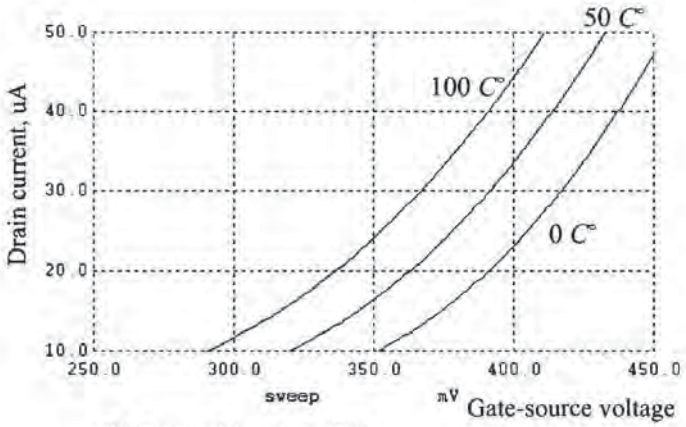



**Figure 9.29** Long-channel drain current change with temperature.




(a)




(b) Zoomed-in view of (a).

**Figure 9.30** Short-channel drain current change with temperature.

IPR2023-00697
Theta EX2019

## 9.2 Short-Channel MOSFETs

The last section covered the fundamental models used for MOSFETs in analog design. In this section we use these results to help with our selection of device sizes and biasing currents to meet certain design requirements: speed, power, output or input swing, etc. Our approach is to develop plots of device parameters based on simulations that can be used when designing with short-channel MOSFETs. The CMOS process technology that we use in this section has a minimum length of 50 nm and a *VDD* of 1 V.

### 9.2.1 General Design (A Starting Point)

We labeled the $V_{DS}$ where the MOSFET enters the saturation region, for a fixed $V_{GS}$ and for either long- or short-channel MOSFETs, $V_{DS,sat}$ (Fig. 9.4). For long-channel MOSFETs this voltage was determined using Eq. (9.2), that is, $V_{GS}-V_{THN}$. However, for short-channel devices this relationship ($V_{DS,sat}=V_{GS}-V_{THN}$) isn't meaningful. For these devices, we'll call the difference between $V_{GS}$ and $V_{THN}$ the *gate overdrive voltage*

$$V_{ovn}=V_{GS}-V_{THN}\neq V_{DS,sat} \tag{9.54}$$

As we saw in Fig. 9.27 the threshold voltage for this 50 nm process is 280 mV. We said earlier that for general analog design we set the overdrive voltage to roughly 5% of *VDD*. We might use, as a starting point,

$$V_{ovn}=70\ mV\rightarrow V_{GS}=350\ mV$$

For higher speed we increase $V_{ovn}$ at the price of reduced output swing (a higher $V_{DS}$ is needed to move the MOSFET into saturation). Using the overdrive voltage, we can rewrite, perhaps more correctly now, Eq. (9.37) as

$$f_T=\frac{g_m}{2\pi C_{gs}}\propto\frac{V_{ovn}}{L} \tag{9.55}$$

Now we must select the biasing current and the width/length of the MOSFETs. As mentioned earlier, we use 2–5 times minimum length for general design (we use minimum length for high-speed design). As we did with the long-channel process, let's use twice minimum length ($L$ = 2 or, with the scale factor, $L$ = 100 nm) to get started as a good trade-off between speed and gain. To select the bias current and width, let's remember the on current for a short-channel MOSFET, from Ch. 6, is

$$i_D=v_{sat}\cdot C'_{ox}\cdot W\cdot(v_{GS}-V_{THN}-V_{DS,sat}) \tag{9.56}$$

The transconductance, following the same procedure used earlier, for a short-channel MOSFET is then

$$g_m=\left[\frac{\partial i_D}{\partial v_{GS}}\right]_{V_{GS}=\text{constant}}^{I_D=\text{constant}}=v_{sat}\cdot C'_{ox}\cdot W \tag{9.57}$$

The $g_m$ of a short-channel device depends only on the MOSFET's width if the velocity of the carriers saturates (is a constant). Fortunately, effects such as velocity overshoot **cause $g_m$ to increase with increases in $V_{GS}$ or $V_{DS}$**. The width of the MOSFET is selected to ensure the MOSFET has enough current drive for a particular load (for a more detailed discussion see Sec. 26.1). Here we'll select a bias current of 10 μA and a $g_m$ of 150 μA/V. As a first cut, let's use $V_{ovn}$ = 70 mV, $W$ = 50, and $L$ = 2 (selected based on simulations).

IPR2023-00697
Theta EX2019

Figure 9.31 shows the IV curves for a 50/2 (actual size of 2.5μm/100nm device) with a $V_{GS}$ of 350 mV. The current shows significant variation as $V_{DS}$ changes. Knowing $I_D$ does change with $V_{DS}$, we'll still simply say the drain current is 10 μA (as an approximation).

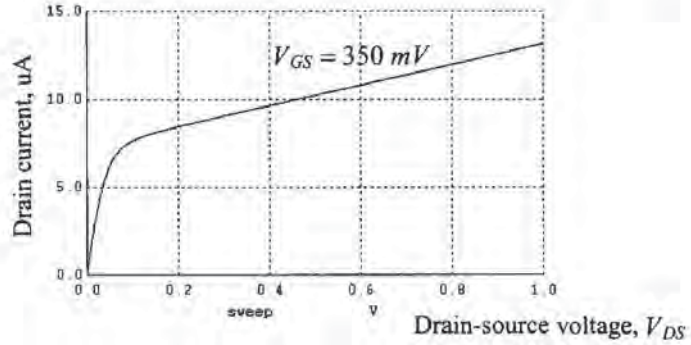



**Figure 9.31** IV curves for a 50/2 NMOS with VGS = 350 mV.

*Output Resistance*

Figure 9.32 shows the output resistance of this 50/2 NMOS device. We get $r_o$ by taking the reciprocal of the drain current's derivative in Fig. 9.31. To determine $V_{DS,sat}$, we can look at the point where the output resistance starts to increase. Here this is approximately 50 mV (= $V_{DS,sat}$). However, notice that if we use larger $V_{DS}$, we get considerably higher output resistances. This is an **important point** when we design current mirrors in Ch. 20. The lambda can be estimated using Figs. 9.31 and 9.32 as 0.6 $V^{-1}$.

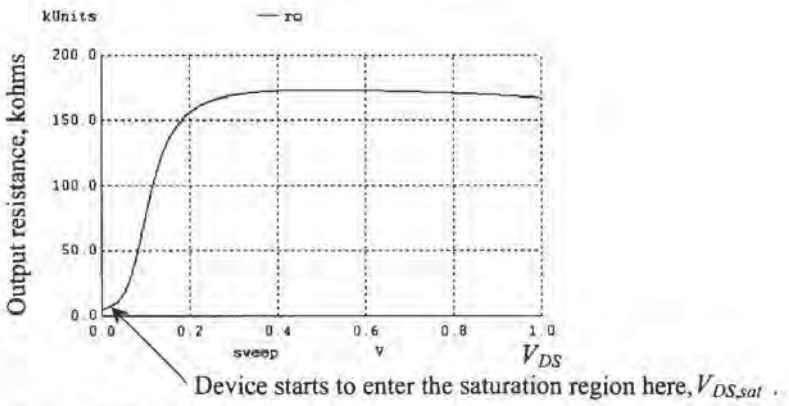



**Figure 9.32** Output resistance of the MOSFET in Fig. 9.31 against $V_{DS}$.

*Forward Transconductance*

The forward transconductance, $g_m$, is plotted against $V_{GS}$ in Fig. 9.33. At our $V_{GS}$ of 350 mV (gate overdrive, $V_{ovn}$, of 70 mV), we get, as designed for, a $g_m$ of 150 μA/*V*. The open circuit gain is $g_m r_o$ and is only 25. This is *considerably lower* than the open circuit gains seen in Table 9.1 for the long-channel devices and results in design "challenges" when designing with such small devices. Also notice that the $g_m$ does change with $V_{GS}$, unlike

IPR2023-00697
Theta EX2019

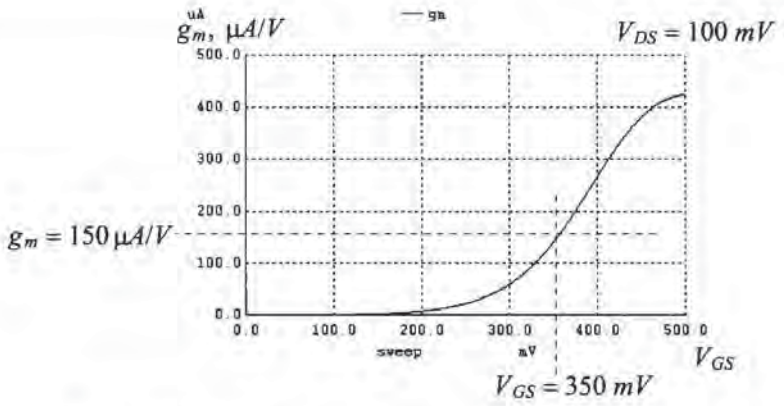



**Figure 9.33** How transconductance changes with gate-source bias.

what was indicated in Eq. (9.57). This is because the saturation velocity isn't exactly constant and depends on both $V_{GS}$ and $V_{DS}$.

*Transition Frequency*

From Fig. 9.34 we see that the transition frequency of the 50/2 NMOS is approximately 6 GHz. While the equations show that increasing the gate overdrive increases $f_T$, it can be educational to change the $V_{GS}$ in the netlist and look at how the speed ($f_T$) changes.

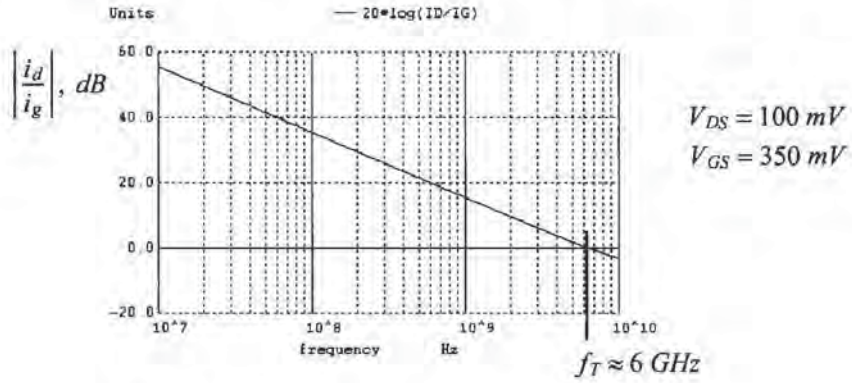



**Figure 9.34** MOSFET transition frequency.

Table 9.2 shows the typical parameters for the sizes and biasing current we've used in this section. These values are a good starting point for general design. However, if the design must be either high-speed or low-power, the biasing current, device sizes, and thus overdrive voltages **will be** different from what's listed. They'll also be different with shifts in process, voltage, and temperature (PVT).

Finally, we generated the PMOS data from simulation netlists that are available at cmosedu.com. These simulation netlists are invaluable for quickly regenerating the data in Table 9.2 for different device sizes and overdrive voltages.

IPR2023-00697
Theta EX2019

**Table 9.2** Typical parameters for analog design using the *short-channel* CMOS process discussed in this book. These parameters are valid only for the device sizes and currents listed.

| *Short*-channel MOSFET parameters for general analog design<br>*VDD* = 1 V and a scale factor of **50 nm** (*scale* = 50e–9) | | | |
|---|---|---|---|
| **Parameter** | **NMOS** | **PMOS** | **Comments** |
| Bias current, $I_D$ | 10 μA | 10 μA | Approximate, see Fig. 9.31 |
| $W/L$ | 50/2 | 100/2 | Selected based on $I_D$ and $V_{ov}$ |
| Actual $W/L$ | 2.5μm/100nm | 5μm/100nm | $L_{min}$ is 50 nm |
| $V_{DS,sat}$ and $V_{SD,sat}$<br>$V_{ovn}$ and $V_{ovp}$ | 50 mV<br>70 mV | 50 mV<br>70 mV | However, see Fig. 9.32 and the associated discussion |
| $V_{GS}$ and $V_{SG}$ | 350 mV | 350 mV | No body effect |
| $V_{THN}$ and $V_{THP}$ | 280 mV | 280 mV | Typical |
| $\partial V_{THN,P}/\partial T$ | – 0.6 mV/C° | – 0.6 mV/C° | Change with temperature |
| $v_{satn}$ and $v_{satp}$ | 110 x $10^3$ m/s | 90 x $10^3$ m/s | From the BSIM4 model |
| $t_{ox}$ | 14 Å | 14 Å | Tunnel gate current, 5 A/cm$^2$ |
| $C'_{ox} = \varepsilon_{ox}/t_{ox}$ | 25 $f$F/μm$^2$ | 25 $f$F/μm$^2$ | $C_{ox} = C'_{ox}WL \cdot (scale)^2$ |
| $C_{oxn}$ and $C_{oxp}$ | 6.25 $f$F | 12.5 $f$F | PMOS is two times wider |
| $C_{gsn}$ and $C_{sgp}$ | 4.17 $f$F | 8.34 $f$F | $C_{gs} = \frac{2}{3}C_{ox}$ |
| $C_{gdn}$ and $C_{dgp}$ | 1.56 $f$F | 3.7 $f$F | $C_{gd} = CGDO \cdot W \cdot scale$ |
| $g_{mn}$ and $g_{mp}$ | 150 μA/V | 150 μA/V | At $I_D$ = 10 μA |
| $r_{on}$ and $r_{op}$ | 167 kΩ | 333 kΩ | Approximate at $I_D$ = 10 μA |
| $g_{mn}r_{on}$ and $g_{mp}r_{op}$ | 25 V/V | 50 V/V | !!Open circuit gain!! |
| $\lambda_n$ and $\lambda_p$ | 0.6 V$^{-1}$ | 0.3 V$^{-1}$ | $L = 2$ |
| $f_{Tn}$ and $f_{Tp}$ | 6000 MHz | 3000 MHz | Approximate at $L = 2$ |

### 9.2.2 Specific Design (A Discussion)

A figure-of-merit (FOM) that is useful when comparing designs with different $W/L$s and bias currents is the gain-$f_T$ product (GFT) of an amplifier. The GFT of a MOSFET can be written as the product of the open-circuit gain and $f_T$, that is,

$$\text{GFT} = g_m r_o \cdot f_T \tag{9.58}$$

For a *long-channel* process we can write this equation (see Eqs. [9.22], [9.30], [9.32], and [9.36]) knowing $C_{gs} = \frac{2}{3}C'_{ox}WL$ as

$$\text{GFT} = g_m r_o \cdot f_T = \frac{g_m^2}{2\pi C_{gs}} \cdot \frac{1}{\lambda I_D} = \frac{3\mu_n}{2\pi \cdot L^2 \lambda} \propto \frac{\mu_n}{L} \tag{9.59}$$

Notice that this expression is *independent of drain current.* It is based entirely on the channel length and mobility of the MOSFET. This result is important because it shows that if the gain goes up, the speed ( $f_T$ ) goes down. Not understanding this equation will

IPR2023-00697
Theta EX2019

result in *wasted time* when trying to increase both the gain and bandwidth of an amplifier. Note that for a *short-channel* process the GFT is *dependent on mobility alone* since the $f_T$ of the MOSFET is proportional to $L$, Eq. (9.55), not $L^2$, Eq. (9.36) (discussion below).

We know from Eq. (9.36) that increasing the drain current (or increasing $V_{GS}$) results in an increase in speed. But the cost for the higher speed is a reduction in the open-circuit gain. This gain can be written, for a device operating in strong inversion, as

$$g_m r_o = \frac{\sqrt{2KP_n \frac{W}{L} I_D}}{\lambda I_D} = \frac{\sqrt{2KP_n \frac{W}{L}}}{\lambda} \cdot \frac{1}{\sqrt{I_D}} \tag{9.60}$$

showing that gain decreases with increasing drain current, Fig. 9.35. We can also write, with the help of Eq. (9.32),

$$g_m r_o = \frac{KP_n \frac{W}{L}(V_{GS} - V_{THN})}{\lambda \cdot \frac{KP_n}{2} \frac{W}{L}(V_{GS} - V_{THN})^2} = \frac{2}{\lambda (V_{GS} - V_{THN})} \propto \frac{L}{V_{ovn}} \tag{9.61}$$

showing that open-circuit gain increases with increasing $L$ or decreasing $V_{GS}$. When operating in the *subthreshold region* (weak inversion), we can write

$$g_m r_o = \frac{I_D}{nV_T} \cdot \frac{1}{\lambda I_D} = \frac{1}{nV_T \lambda} \tag{9.62}$$

showing that the gain is independent of drain current when the long-channel MOSFET is operating in the subthreshold region, again see Fig. 9.35.

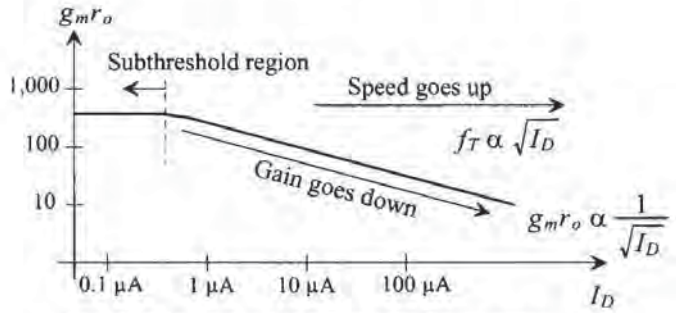



**Figure 9.35** Gain falling off with bias current.

While these results were derived using long-channel models, we can use the outcomes for design in a short-channel process. For high-speed we want to use higher biasing currents and thus overdrive voltages. As the speed of the design increases, the gain falls (but, as we'll see, not at the same rate). As an example, if we decrease the width of the NMOS device used to generate Table 9.2 from 50 to 20 (using the same gate-source voltage of 350 mV), we would expect $I_D$ to fall, $f_T$ to remain unchanged (the device is narrower so its capacitance decreases but so does the $g_m$), $r_o$ to increase, and $g_m$ to decrease. The GFT shouldn't change much by simply reducing the width of the device. First, from Table 9.2 the GFT is

$$\overbrace{g_m r_o}^{\text{open circuit gain = 25}} \cdot f_T = 150 \frac{\mu A}{V} \cdot 167\ k\Omega \cdot 6\ GHz = 150\ GHz$$

IPR2023-00697
Theta EX2019

Next, with the decrease in the device's width to 20, we get (with the help of the simulations that generated Figs. 9.31–9.34): $I_D = 4$ μA ($V_{ovn}$ is still 70 mV), $r_o = 435$ kΩ, $g_m = 55$ μA/V, and $f_T = 6$ GHz. The GFT with the device width of 20 is

$$g_m r_o \cdot f_T = \overbrace{55\tfrac{\mu A}{V} \cdot 435\ k\Omega}^{\text{open circuit gain} = 24} \cdot 6\ GHz = 144\ GHz$$

practically no difference. If we now increase the gate-source voltage to 450 mV (almost half of *VDD*) so that the overdrive goes from 70 mV to 170 mV while using a 50/2 NMOS, we would expect the $f_T$ to increase, $g_m$ to increase, $I_D$ to increase, and $r_o$ to decrease. Again, with the help of the simulations that generated Figs. 9.31–9.34, $I_D = 45$ μA ($V_{ovn}$ is now 170 mV), $r_o = 55$ kΩ, $g_m = 390$ μA/V, and $f_T = 10$ GHz. The GFT with the increased $V_{ovn}$ is

$$g_m r_o \cdot f_T = \overbrace{390\tfrac{\mu A}{V} \cdot 55\ k\Omega}^{\text{open circuit gain} = 21.5} \cdot 10\ GHz = 215\ GHz$$

Using the long-channel equations, Eq. (9.59), we would expect little change (again) since the GFT doesn't depend on the overdrive voltage. However, the mobility in a short channel process does change with $V_{ovn}$ (see $g_m$ and velocity overshoot discussions on pages 151 and 297). By using higher gate overdrives, in a short-channel process, we can improve the GFT (and speed) of our circuits at the cost of power and overhead (the devices will triode at a higher $V_{DS}$) with small degradation to the gain.

Based on large GFT it might appear that it's better to design using a long-channel (older) CMOS technology rather than a short-channel (nanometer) process. The GFT, for example, of the NMOS device in Table 9.1 is 675 GHz while it's only 150 GHz for the NMOS device in Table 9.2. In addition, the open circuit gains of short-channel MOSFETs are considerably lower than long-channel MOSFETs. The smaller device, however, is better for high-speed design because its $f_T$ is higher (it still behaves like an amplifier at high frequencies). In summary, for high-speed designs use a small process while for low-frequency designs use older CMOS (if there is a choice).

## 9.3 MOSFET Noise Modeling

In this section we cover modeling MOSFET noise using SPICE. It is assumed that the reader is familiar with the material presented in Ch. 8, that is, the spectral characteristics and origins of thermal and flicker noise.

*Drain Current Noise Model*

The noise generators in MOSFETs are due to thermal and flicker noise. The thermal noise due to the channel resistance, modeled in the saturation region using a resistor of $\frac{3}{2} \cdot \frac{1}{g_m}$ (subsitute Eq. [6.46], without the area dependence, into [6.27] and integrate) and $R_{CH}$ in the triode region, is given, in the saturation region, by a power spectral density, PSD, of

$$I_R^2(f) = \frac{4kT}{\frac{3}{2} \cdot \frac{1}{g_m}} = \frac{8kT}{3} \cdot g_m \tag{9.63}$$

This noise current source is placed across the drain and source of the MOSFET (so the output current is the sum of the desired and undesired [noise] components).

IPR2023-00697
Theta EX2019

Flicker noise (one-over-f, that is, $1/f$) results from the trapping of charges at the oxide/semiconductor interface. As indicated in Ch. 8, flicker noise is present whenever a direct current flows in a discontinuous material. The electrons jump from one location to the next while sometimes being randomly trapped and released. This trapping gives rise to a flickering in the drain current. Since the carrier lifetime in silicon is on the order of tens of microseconds (relatively long), the resulting current fluctuations are concentrated at lower frequencies. Flicker noise can be modeled in SPICE (with NLEV = 0, the default) by a PSD of

$$I_{1/f}^2(f) = \frac{KF \cdot I_D^{AF}}{f \cdot (C'_{ox} \cdot L)^2} \tag{9.64}$$

where $KF$ is the flicker noise coefficient, a typical value is $10^{-28}$ $A^{2-AF}(F/m)^2$, $I_D$ is the DC drain current, $AF$ is the flicker noise exponent, a value ranging from 0.5 to 2 (generally very close to 1), and $f$ is the frequency variable we integrate over. Setting NLEV = 1 in SPICE changes the $L^2$ term in the denominator to $LW$ (the area of the MOSFET), that is,

$$I_{1/f}^2(f) = \frac{KF \cdot I_D^{AF}}{f \cdot (C'_{ox})^2 LW} \tag{9.65}$$

Figure 9.36 shows measured data comparing a straight-line fit to the PSD generated using this noise model. The device used to collect this data is 10 μm wide and 2.8 μm long. Note that the noise PSD was averaged 20 times and still looks jagged. The 1/f noise spectrums we drew in Ch. 8, e.g., Fig. 8.31, show smooth PSDs. We only get smooth spectrums after much averaging (it is, after all, noise).

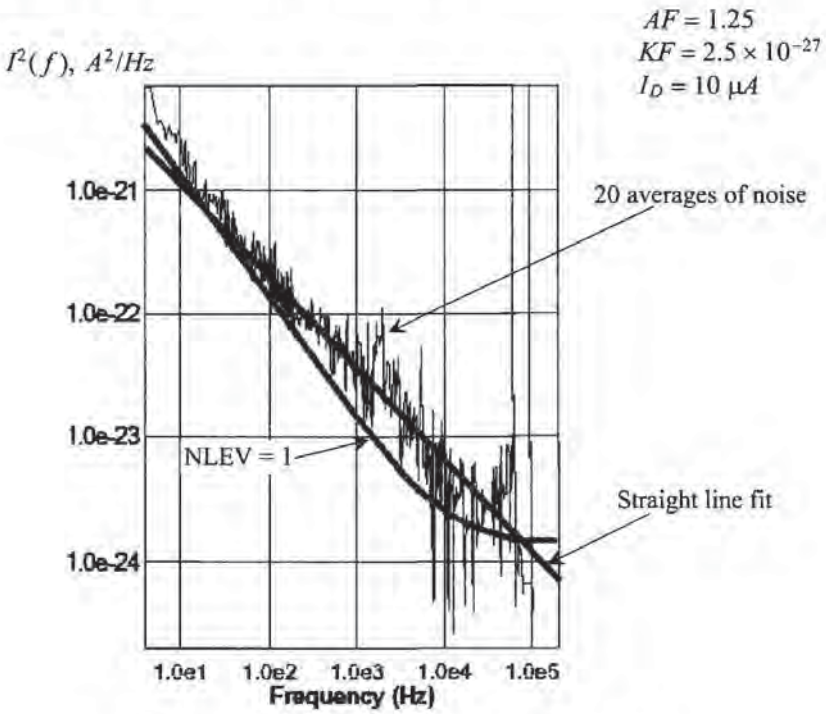



**Figure 9.36** Measured MOSFET flicker noise spectrum.

IPR2023-00697
Theta EX2019

The total PSD of MOSFET drain current noise (the sum of the flicker and thermal noise contributions) is given by

$$I_M^2(f) = I_{1/f}^2(f) + I_R^2(f) = \frac{KF \cdot I_D^{AF}}{f \cdot (C'_{ox})^2 LW} + \frac{8kT}{3} \cdot g_m \qquad (9.66)$$

Remembering noise is always measured on the output of a circuit and referred back to the input of a circuit so that it can be compared to the input signal, we can write (knowing $i_d = g_m v_{gs}$) the MOSFET's input-referred noise PSD as

$$V_{inoise}^2(f) = \frac{KF \cdot I_D^{AF}}{f \cdot (C'_{ox})^2 LW \cdot g_m^2} + \frac{8kT}{3g_m} \qquad (9.67)$$

Figure 9.37 shows how the noise models are added to the schematic representation of the MOSFET. Notice that increasing the MOSFET's forward transconductance, $g_m$, reduces the input-referred thermal noise. If the $g_m$ is increased by making the device wider (that is, it is not increased by simply raising the drain current), then the $1/f$ noise decreases as well. Note, as discussed in Ch. 8, that looking at the (size of the) output noise alone gives no indication of the noise performance of the amplifier or circuit. The input-referred noise should be compared to the input signal for an indication of noise performance. (A notable exception to this is a circuit that doesn't have an input signal like a current mirror. In these types of circuits we *do* care about reducing the size of the output noise.)

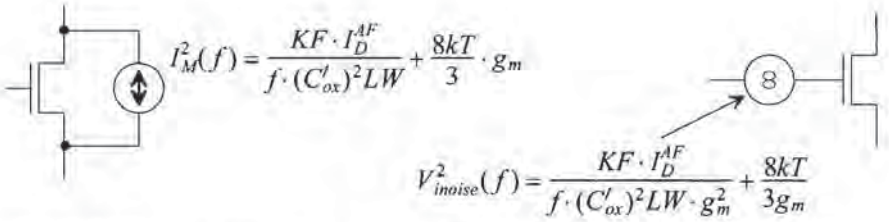



**Figure 9.37** (a) Modeling MOSFET noise current and (b) input-referred noise voltage.

Finally, we might ask if it is better to use a PMOS or NMOS device for low-noise design? In the older CMOS technologies, say 350 nm and earlier technology nodes, n-type polysilicon was used to form the gates of both NMOS and PMOS devices (single workfunction gates, see page 191). This keeps the PMOS device's channel from forming at the surface directly under the gate oxide. Instead a *buried channel* is formed in the device. By avoiding conduction at the oxide/semiconductor surface we also avoid the trapping and random release of carriers that is a major component of the MOSFET's drain current flicker noise. The result is the PMOS devices in the older CMOS processes have considerably better noise performance than the NMOS devices. In modern CMOS n-type polysilicon is used to form the gate of the NMOS device and p-type polysilicon is used for the gate of the PMOS (dual workfunction gates and why the polysilicon must be silicided, that is, so the connection between the p-type and n-type polysilicon isn't rectifying). The result is that both devices utilize a surface channel so the benefit of using PMOS devices is gone. The larger $g_m$ of the NMOS makes it the preferable device for low noise design (however, this is process dependent).

IPR2023-00697
Theta EX2019

## ADDITIONAL READING

*Textbooks Covering Small-Signal Analysis*

[1] R. C. Jaeger and T. N. Blalock, *Microelectronic Circuit Design*, 4th ed., McGraw-Hill Publishers, 2011. ISBN 978-0-07-338045-2.

[2] R. Spencer and M. Ghausi, *Introduction to Electronic Circuit Design*, Prentice-Hall Publishers, 2003. ISBN 0-201-36183-3.

[3] D. J. Comer and D. T. Comer, *Fundamentals of Electronic Circuit Design*, John Wiley and Sons, 2002. ISBN 0-471-41016-0.

[4] D. A. Neamen, *Electronic Circuit Analysis and Design*, McGraw-Hill Publishers, 2001. ISBN 0-072-45194-7.

[5] A. R. Hambley, *Electronics*, Prentice-Hall Publishers, 2nd ed, 2000. ISBN 0-136-91982-0.

[6] M. H. Rashid, *Microelectronic Circuits: Analysis and Design*, Brookes-Cole Publishing, 1998. ISBN 0-534-95174-0.

[7] A. Sedra and K. Smith, *Microelectronic Circuits*, Oxford University Press, 4th ed., 1998. ISBN 0-195-11663-1.

[8] R. T. Howe and C. G. Sodini, *Microelectronics: An Integrated Approach*, Prentice-Hall Publishers, 1997. ISBN 0-135-88518-3.

[9] N. Malik, *Electronic Circuits: Analysis, Simulation, and Design*, Prentice-Hall Publishers, 1995. ISBN 0-023-74910-5.

[10] M. N. Horenstein, *Microelectronic Circuits and Devices*, Prentice-Hall Publishers, 2nd ed., 1990. ISBN 0-135-83170-9.

[11] M. S. Roden, G. L. Carpenter, and C. J. Savant, *Electronic Design: Circuits and Systems*, Pearson Benjamin Cummings Publishers, 2nd ed, 1990.

[12] J. Millman and A. Grabel, *Microelectronics*, McGraw-Hill Publishers, 2nd ed. 1987. ISBN 0-070-42330-X.

*Textbooks Covering Noise Analysis*

[13] F. Bonani and G. Ghione, *Noise in Semiconductor Devices: Modeling and Simulation*, Springer-Verlag Publishers, 2002. ISBN 3-540-66583-8.

[14] C. D. Motchenbacher and J. A. Connelly, *Low-Noise Electronic System Design*, John Wiley and Sons, 1993. ISBN 0-471-57742-1.

[15] H. L. Krauss, C. W. Bostian, and F. H. Raab, *Solid State Radio Engineering*, John Wiley and Sons, 1980. ISBN 0-471-03018-X.

*A Good Paper on the Origins of Flicker Noise*

[16] Reimbold, G., "Modified 1/f Trapping Noise Theory and Experiments in MOS Transistors Biased from Weak to Strong Inversion - Influence of Interface States," *IEEE Transactions on Electron Devices*, vol. ED-31, no. 9, pp. 1190–1198, September, 1984.

IPR2023-00697
Theta EX2019

## PROBLEMS

For the following problems use the long-channel process information given in Table 9.1 for $KP$, $V_{TH}$, and $C'_{ox}$, unless otherwise indicated.

**9.1** Calculate and simulate the values of $I_D$ and $V_{GS}$ in the following circuit.

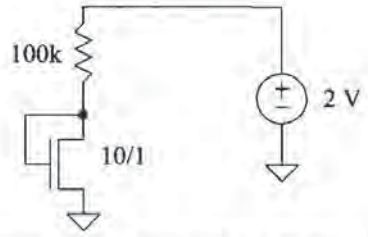



**Figure 9.38** Circuit used in Problem 9.1

**9.2** Repeat Problem 9.1 if the MOSFET size is changed to 10/10.

**9.3** Calculate and simulate the values of $I_D$ and $V_{SG}$ in the following circuit.

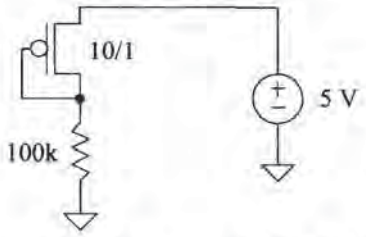



**Figure 9.39** Circuit used in Problem 9.3

**9.4** Repeat Problem 9.3 if the MOSFET size is changed to 10/10.

**9.5** Calculate $I_D$, $V_{DS}$, and estimate the small-signal resistance looking into the drain of the MOSFET in the following circuit.

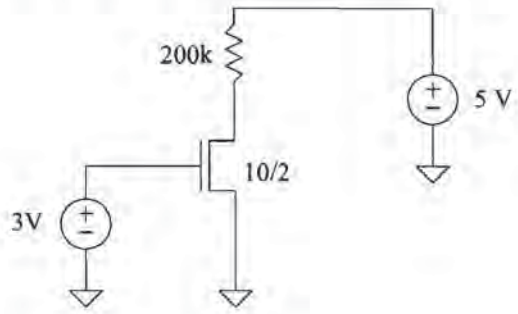



**Figure 9.40** Circuit used in Problem 9.5

IPR2023-00697
Theta EX2019

**9.6** To determine the value of a small-signal resistance in a simulation, the circuit seen in Fig. 9.41 is used. The ratio of the test voltage, $v_T$, to the current that flows in the source, $i_T$, that is, $v_T/i_T$, is the small signal resistance. Using this circuit determine, with SPICE, the value of the resistance looking into the drain of the circuit in Fig. 9.40. How does the 200k resistor affect $i_T$ in the simulation?

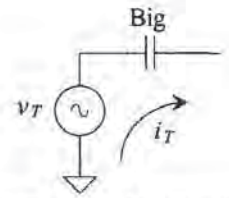



**Figure 9.41** Using a test voltage to determine a resistance in a simulation.

**9.7** Explain qualitatively what happens to $V_{GS4}$ and $V_{DS4}$ in Fig. 9.42 as the bias current is increased.

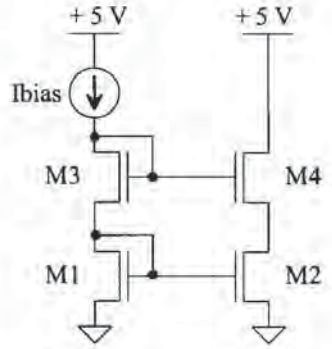



**Figure 9.42** Schematic for Problem 9.7.

**9.8** Describe qualitatively what happens if we steal or inject a current at the point indicated in Fig. 9.43. How does this affect the operation of M1 and M2? Verify your answer with SPICE.

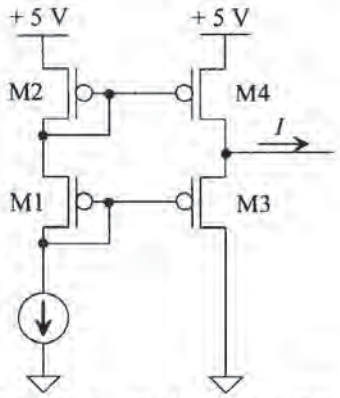



**Figure 9.43** Schematic for Problem 9.8.

IPR2023-00697
Theta EX2019

**9.9** Using simulations, generate the plot seen in Fig. 9.12 for both NMOS and PMOS devices.

**9.10** Design a circuit that will linearly convert an input voltage that ranges from 0 to 4V into a current that ranges from roughly 50 μA to 0. Simulate the operation of the design showing the linearity of the voltage-to-current conversion. How does the MOSFET's length affect the linearity?

**9.11** Using a PMOS device, discuss and show with simulations how it can be used to implement a 10k resistor. Are there any limitations to the voltage across the PMOS resistor? Explain.

**9.12** Using SPICE (and ensuring the MOSFET is operating in the saturation region with sufficient $V_{DS}$), generate the $i_D$ versus $v_{GS}$ curve seen in Fig. 9.15. Using SPICE take the derivative of $i_D$ (e.g., "plot deriv(ID)") to get the device's $g_m$ (versus $V_{GS}$). How does the result compare to Eq. (9.22)? Does the level 3 model used in the simulation show a continuous change from subthreshold to strong inversion?

**9.13** Estimate the AC, $i_d$ , drain current that flows in the circuit seen in Fig. 9.44. Verify your answer with SPICE in using both transient and AC simulations.

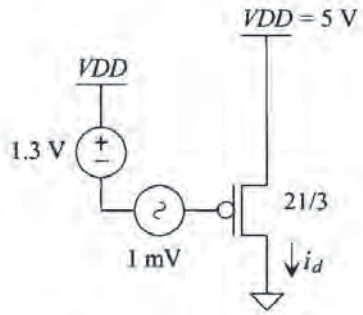



**Figure 9.44** Schematic for Problem 9.13.

**9.14** Calculate the DC and AC voltages and currents for the circuit seen in Fig. 9.45. Verify your answers with a SPICE AC analysis simulation.

**9.15** Repeat Problem 9.14 if the bias current is reduced to 20 μA.

**9.16** Using SPICE (and zero volt sources), verify the drain currents calculated in Ex. 9.5. Show both your AC analysis and transient analysis simulation results.

**9.17** Using SPICE, generate the $i_D$ versus $v_{SB}$ curve seen in Fig. 9.21.

**9.18** Repeat Ex. 9.6 if the widths and lengths of the PMOS and NMOS devices are multiplied by 10 (that is the NMOS is now 100/20 and the PMOS is now 300/20). How does the drain current change?

**9.19** Compare the $f_T$ of a 10/2 NMOS to a 100/20 NMOS. Verify your hand calculations using simulations.

IPR2023-00697
Theta EX2019

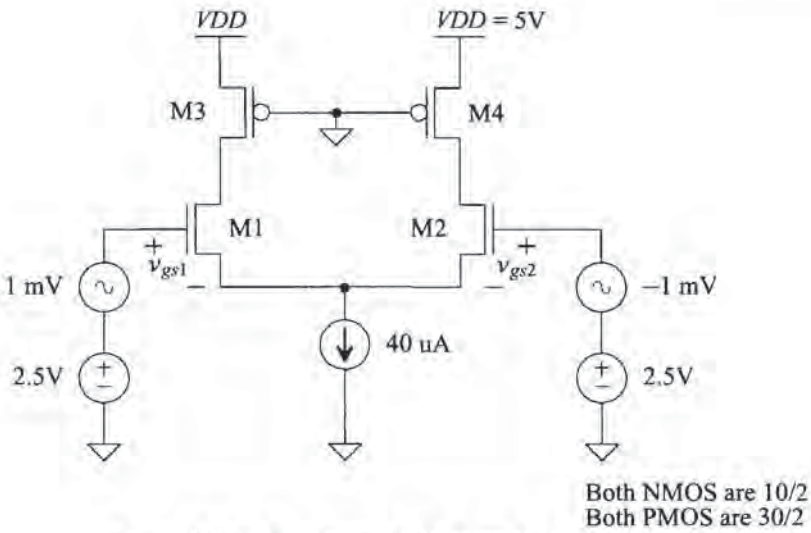



**Figure 9.45** Circuit used in Problem 9.14.

**9.20** Equation (9.39) is used to calculate the forward transconductance of a MOSFET operating in the subthreshold region. Is it possible to have subthreshold operation when $I_D$ is 100 μA? If so, how?

**9.21** Using the simulations that generated Fig. 9.27, estimate the threshold voltage for 50/5 PMOS and NMOS devices (both in the short-channel CMOS process).

**9.22** Show the details leading to Eq. (9.43). Show, as an example, that the approximation is valid if $Q'_{b0}/C'_{ox} = 30\,mV$. (Remember: temperature is in Kelvin.)

**9.23** For the circuits seen in Fig. 9.46, estimate both the output voltage at room temperature and how it will change with temperature. Verify your answer with SPICE. Use the short-channel CMOS process.

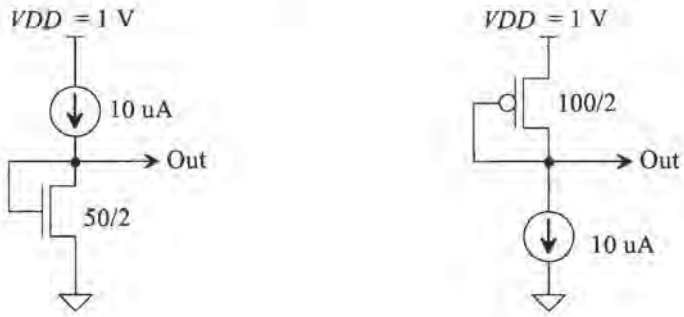



**Figure 9.46** Circuits for Problem 9.23.

IPR2023-00697
Theta EX2019

**9.24** Verify, using simulations, the result given in Eq. (9.59), that is, that the GFT of a long-channel process is independent of biasing conditions. Make sure that the MOSFET is biased away from the subthreshold and triode regions.

**9.25** When Eq. (9.36) was derived it was assumed that $C_{gs} >> C_{gd}$. Looking at the data in Table 9.2, is this a good assumption? Compare the hand-calculated value of $f_T$ to the simulation results in Table 9.2.

**9.26** Using the short-channel parameters in Table 9.2, calculate the drain voltage of M1 and its drain current (both AC and DC components) for the circuit seen in Fig. 9.47.

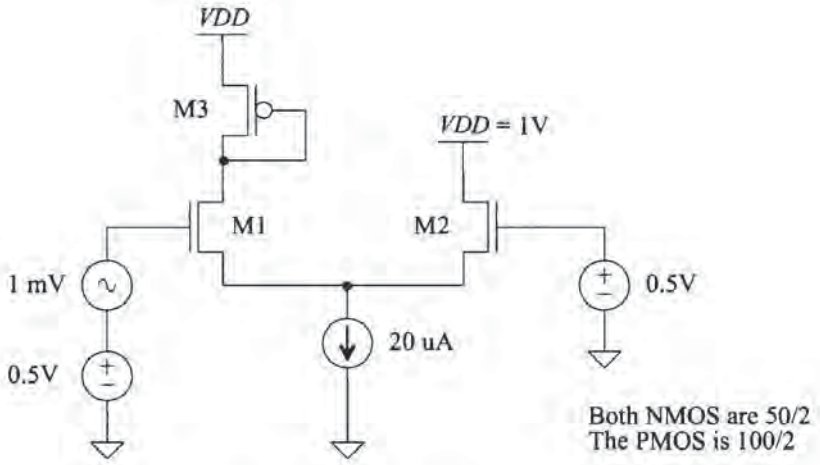



**Figure 9.47** Circuit used in Problem 9.26.

**9.27** What is the PSD of a MOSFET's thermal noise when it is operating in the triode region?

**9.28** Looking at Eq. (9.66), we see that decreasing the MOSFET's $g_m$ reduces the MOSFET's drain current noise PSD. If the MOSFET is to be used as an amplifier where the input signal is on the MOSFET's gate, is reducing $g_m$ a good idea? Why or why not?

**9.29** Show how the thermal noise resistance of the channel seen in Eq. (9.63) is derived for the MOSFET operating in the saturation region.

IPR2023-00697
Theta EX2019

Chapter
10

# Models for Digital Design

In this chapter we develop a digital model for the MOSFET. For a simple logical analysis of a digital circuit we think, Fig. 10.1, of the MOSFETs as simple switches. When the gate of an NMOS device is a logic 1 (*VDD*), the NMOS device is on. When the gate is a logic 0 (ground), the NMOS device is off. The PMOS device operates in a complementary way to the NMOS device. When its gate is a logic zero, it is on (hence why we draw a bubble at its gate). It's important, for a quick logical analysis of a complex digital circuit, to think of the MOSFETs as simple logic-controlled switches.




**Figure 10.1** Logical operation of the switches.

*Miller Capacitance*

One of the important concepts we'll use in this chapter has to do with the "Miller effect" discussed in detail in Ch. 21 (see Fig. 21.5 and the associated discussion). To understand this effect here in our development of digital models, consider the simple schematic in Fig. 10.2. If, initially, the input node is at 0 V and the output node is at *VDD*, the charge on the capacitor is

$$Q_{init} = C \cdot (0 - VDD) = -C \cdot VDD \qquad (10.1)$$

If the input node changes to *VDD* and the output node transitions to ground, the charge on the capacitor is

$$Q_{final} = C \cdot (VDD - 0) = C \cdot VDD \qquad (10.2)$$

IPR2023-00697
Theta EX2019

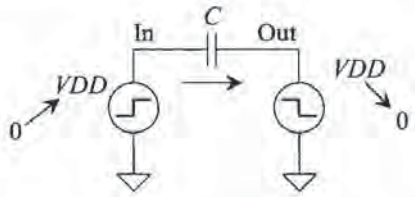



**Figure 10.2** Determining the charge through a capacitor.

The total charge supplied by the input or output voltage source to the capacitor after the transition is then

$$Q_{tot} = Q_{final} - Q_{init} = 2C \cdot VDD \tag{10.3}$$

Remembering, in this discussion, that the input source transitions from 0 to *VDD* while, at the same time, the output source transitions from *VDD* to 0, what is the effective capacitance that each source sees? Towards answering this question, consider the models seen in Fig. 10.3. In (a) the input source supplies a charge of

$$Q_{tot} = 2C \cdot VDD \tag{10.4}$$

In (b) the output sinks the same amount of charge. The point here is that the input or output capacitance of the circuits in 10.3a or b is twice the capacitance value connecting the input to the output in Fig. 10.2. We'll use this result in a moment as we develop a digital model for the MOSFET.




(a) Input circuit (b) Output circuit

**Figure 10.3** Splitting the capacitor in Fig. 10.2 up into two equivalent capacitors for developing a model.

## 10.1 The Digital MOSFET Model

*Effective Switching Resistance*

Consider the MOSFET circuit shown in Fig. 10.4. Initially, the MOSFET is off, $V_{GS} = 0$, and the drain of the MOSFET is at *VDD*. If the gate of the MOSFET is taken instantaneously from 0 to *VDD*, a current given by

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L} \cdot (VDD - V_{THN})^2 = \frac{\beta}{2} \cdot (VDD - V_{THN})^2 \tag{10.5}$$

IPR2023-00697
Theta EX2019

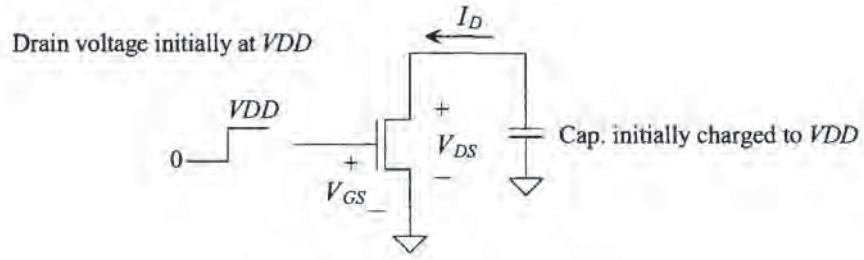



**Figure 10.4** MOSFET switching circuit.

initially flows through the MOSFET. Point A in Fig. 10.5 shows the operating point of the MOSFET prior to switching for $VDD$ = 5 V. After switching takes place, the operating point moves to point B and follows the curve $V_{GS} = VDD$ down to $I_D \approx 0$ and $V_{DS} = 0$, point C. At this point, the NMOS switch is on.

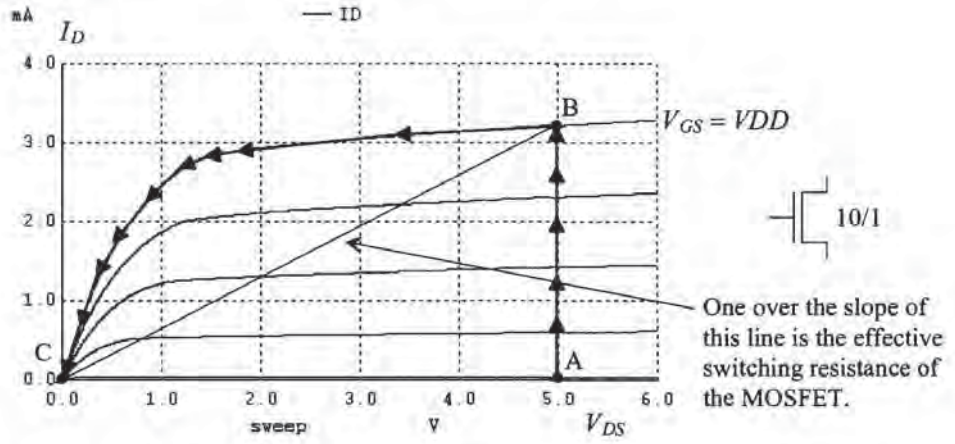



**Figure 10.5** IV plot for a 10/1 NMOS device to estimate average switching resistance.

An estimate for the resistance between the drain and source of the MOSFET is given by the reciprocal slope of the line BC in Fig. 10.5, or

$$R_n = \frac{VDD}{\frac{KP_n}{2}\frac{W}{L} \cdot (VDD - V_{THN})^2} = R'_n \cdot \frac{L}{W} \qquad (10.6)$$

The MOSFET is modeled by the circuit shown in Fig. 10.6. When $V_{GS} > VDD/2$, the switch is closed; when $V_{GS} < VDD/2$ the switch is open. In the derivation of this model, we assumed that the input step transition occurred in zero time; that is, that the rise time was zero, so that the point at which the switch was opened or closed was well defined. In practice, we will not encounter a zero rise time pulse; therefore, the model has limitations. Nevertheless, the model works remarkably well in designing and analyzing digital circuits by hand, giving results that are usually within a factor of two of simulation or measurement in general applications.

IPR2023-00697
Theta EX2019

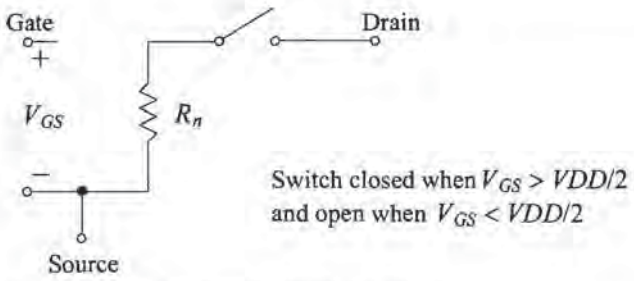



**Figure 10.6** Simple digital MOSFET model.

Because the square-law equation used in Eq. (10.6) doesn't model, for example, the reduction in mobility, Sec. 6.5, present in submicron devices it's better to use measured data, via an experimentally-generated SPICE MOSFET model or bench data, to estimate the switching resistances. From Fig. 10.5 we can write

$$R_n = R'_n \cdot \frac{L}{W} = \frac{VDD}{I_{D,sat}} = \frac{5}{3.3\ mA} \rightarrow R'_n \approx 15\ k\Omega \tag{10.7}$$

For an NMOS device with a specific width, $W$, or length, $L$, we can estimate the effective switching resistance of the device in the *long-channel CMOS* process in this book (scale factor of 1 μm) as

$$R_n \approx 15k \cdot \frac{L}{W} \tag{10.8}$$

For the PMOS device, the transconductance parameter, $KP$, is three times smaller than the NMOS's $KP$ (see Table 9.1), so we can write

$$R_p \approx 45k \cdot \frac{L}{W} \tag{10.9}$$

The effective resistance of the PMOS device is three times as large as the NMOS's due to the mobility of the electrons being larger than the mobility of the holes.

*Short-Channel MOSFET Effective Switching Resistance*

As discussed in Chs. 6 and 9, the short-channel MOSFET doesn't follow the square-law, long-channel, MOSFET models. We can't use Eq. (10.6) to estimate the effective switching resistance. However, we can use the *on current* for the devices (see Sec. 6.5.2). For the *short-channel CMOS* process used in this book (scale factor of 50 nm and a $VDD$ of 1 V), we can write, see Eqs. (6.59), (6.60), and (6.61),

$$R_n = \frac{VDD}{I_{D,n}} = \frac{VDD}{I_{on,n} \cdot W \cdot scale} = \frac{1\ V}{600\ \frac{\mu A}{\mu m} \cdot W \cdot scale} \rightarrow R_n \approx \frac{1.7k \cdot \mu m}{W \cdot scale} \tag{10.10}$$

and

$$R_p = \frac{VDD}{I_{D,p}} = \frac{VDD}{I_{on,p} \cdot W \cdot scale} = \frac{1\ V}{300\ \frac{\mu A}{\mu m} \cdot W \cdot scale} \rightarrow R_p \approx \frac{3.4k \cdot \mu m}{W \cdot scale} \tag{10.11}$$

noting that the effective switching resistance of a short-channel MOSFET is independent of the MOSFET's length. In practice this is true to a point. Increasing the channel length above a certain value, for example $L = 2$, causes the MOSFET's resistance to grow.

IPR2023-00697
Theta EX2019

Note that in the long-channel equations for switching resistance, Eqs. (10.8) and (10.9), the scale factor doesn't affect the resistance. However, using the short-channel equations the scale factor is important and does affect the switching resistance calculation when using $I_{on}$. Knowing, for the short-channel process used in this book, that the scale factor is 50 nm, we can rewrite Eqs. (10.10) and (10.11) as

$$R_n = \frac{34k}{W} \text{ and } R_p = \frac{68k}{W} \tag{10.12}$$

where $W$ is the drawn width of the devices. Practically, to model the effects of increasing switching resistance when $L$ is greater than 1 (minimum), we can re-write Eq. (10.12) as

$$R_n = 34k \cdot \frac{L}{W} \text{ and } R_p = 68k \cdot \frac{L}{W} \tag{10.13}$$

noting that now $L$ and $W$ can both be either drawn or actual sizes.

### 10.1.1 Capacitive Effects

At this point, we need to add the capacitances of the switching MOSFET to our model of Fig. 10.6. Consider the MOSFET circuit shown in Fig. 10.7 with capacitance $\frac{C_{ox}}{2}$ between the gate-drain and the gate-source electrodes. This is the capacitance when the MOSFET is in the triode region and is an *overestimate* for the capacitances of the MOSFET. For example, the capacitance between the gate and drain of the MOSFET is the overlap capacitance from the lateral diffusion when the MOSFET is operating in the saturation region (see Table 6.1). Because of this overestimate, we will neglect the depletion capacitances of the source and drain implants to substrate when doing hand calculations. Simulations using SPICE are required for better estimates of switching behavior (e.g., showing the nonlinearities of a depletion capacitance, see Fig. 2.15 or Eq. [5.17]).

Returning to Fig. 10.7, when the input pulse transitions from 0 to *VDD*, the output transitions from *VDD* to 0 (review Figs. 10.2 and 10.3). The current through $C_{gd}$ (= $C_{ox}$/2) is given by

$$I = C_{gd} \cdot \frac{dV_{gd}}{dt} = \frac{C_{ox}}{2} \cdot \frac{VDD-(-VDD)}{\Delta t} = C_{ox} \cdot \frac{VDD}{\Delta t} = C_{ox} \cdot \frac{dV_{DS}}{dt} \tag{10.14}$$

The voltage across $C_{gd}$ changes by $2 \cdot VDD$. The current that flows through this capacitance is the drain current of the MOSFET in Fig. 10.7. As seen in Fig. 10.3, we can break $C_{gd}$ into a component from the gate to ground and from the drain to ground of value $2C_{gd}$ or $C_{ox}$. The complete model of a switching MOSFET is shown in Fig. 10.8.

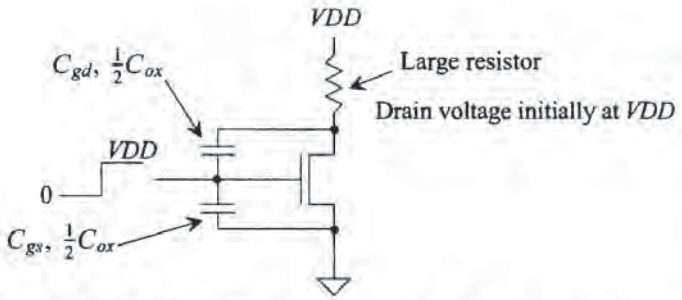



**Figure 10.7** MOSFET switching circuit with capacitances.

IPR2023-00697
Theta EX2019

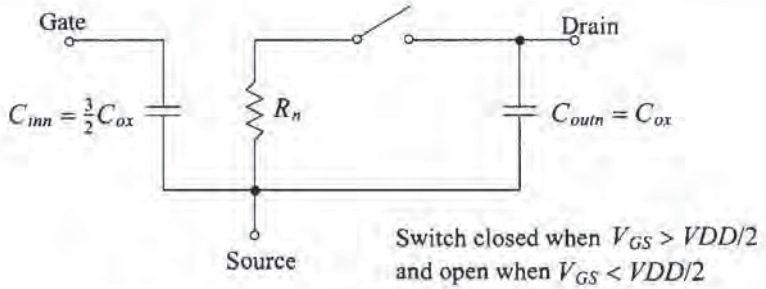



**Figure 10.8** Simple digital MOSFET model.

### 10.1.2 Process Characteristic Time Constant

An important question we can answer at this point is, "What is the intrinsic switching speed of a MOSFET?" Looking at Figs. 10.7 and 10.8, we can see an intrinsic *process characteristic time constant*, $\tau_n$, of $R_nC_{ox}$. That is, if the drain is charged to *VDD* as in Fig. 10.7 and the input switches from 0 to *VDD*, the output voltage will decay with a time constant of $R_nC_{ox}$. For an NMOS device in the long-channel process, this is given by

$$\tau_n = R_nC_{ox} = \frac{2L \cdot VDD}{KP_nW(VDD - V_{THN})^2} \cdot C'_{ox}WL \cdot (scale)^2 = \frac{2C'_{ox} \cdot VDD \cdot (L \cdot scale)^2}{KP_n \cdot (VDD - V_{THN})^2} \tag{10.15}$$

Notice that the "speed" of a process increases as the square of the channel length and that the speed is independent of the channel width, *W*. The process characteristic time constant is specified using the process minimum length. Also note that the larger the *VDD*, the faster the process. This is very similar to the unity current gain frequency, $f_T$, that we discussed in the last chapter. For a short-channel process, Eq. (10.15) can be rewritten as

$$\tau_n = R_nC_{ox} = \frac{VDD}{I_{on,n} \cdot W \cdot scale} \cdot C'_{ox}WL \cdot (scale)^2 = \frac{VDD \cdot C'_{ox} \cdot L \cdot scale}{I_{on,n}} \tag{10.16}$$

For the short-channel process, the time constant decreases linearly with decreasing channel length.

---

**Example 10.1**

Estimate the process characteristic time constants for the long- and short-channel CMOS processes used in this book.

For the NMOS device in the long-channel CMOS process, using the data in Table 5.1 and Eq. (10.8),

$$\tau_n = R_nC'_{ox}WL \cdot (scale)^2 = 15k \cdot \frac{L}{W} \cdot 1.75\,\frac{fF}{\mu m^2} \cdot LW \cdot (scale)^2 \approx 25\,ps$$

Knowing from Eqs. (10.8) and (10.9) that the effective resistance of the PMOS device is three times as large as the resistance of the NMOS device, we can write

$$\tau_p \approx 75\,ps$$

IPR2023-00697
Theta EX2019

For the short-channel process, using the data in Table 5.1 and Eq. (10.10), we can write

$$\tau_n = R_n C'_{ox} \cdot WL \cdot (scale)^2 = \frac{1.7k \cdot \mu m}{W \cdot scale} \cdot 25\frac{fF}{\mu m^2} \cdot WL \cdot (scale)^2 = 2.1\ ps$$

and for the PMOS device

$$\tau_p = 4.2\ ps$$

Table 10.1 summarizes the effective switching resistances and oxide capacitances for the long- and short-channel CMOS processes used in this book. ■

**Table 10.1** Digital model parameters used for hand calculations in the long- and short-channel CMOS processes used in this book. Note that the widths, $W$, and lengths, $L$, seen in this table are drawn lengths (minimum length is 1 while minimum width is 10).

| Technology | $R_n$ | $R_p$ | Scale factor | $C_{ox} = C'_{ox} WL \cdot (scale)^2$ |
|---|---|---|---|---|
| 1 μm (long-channel) | $15k\frac{L}{W}$ | $45k\frac{L}{W}$ | 1 μm | $(1.75\,fF) \cdot WL$ |
| 50 nm (short-channel) | $\frac{34k}{W}$ | $\frac{68k}{W}$ | 50 nm | $(62.5\,aF) \cdot WL$ |

### 10.1.3 Delay and Transition Times

Before we go any further in the discussion of the digital models, let's define delays and transition times in logic circuits. Consider Fig. 10.9. The top trace represents the input to a logic gate, while the bottom trace represents the output. Note that there is no logic inversion between the input and output; however, the following definitions apply equally well to the case when there is an inversion. The input rise and fall times are labeled $t_r$ and $t_f$ respectively. The output rise and fall times are labeled $t_{LH}$ and $t_{HL}$, respectively. The

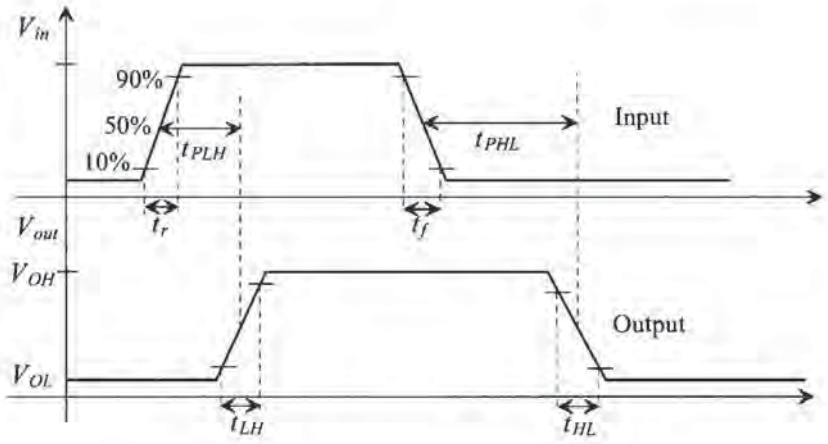



**Figure 10.9** Definition of delays and transition times.

IPR2023-00697
Theta EX2019

delay time between the 50% points of the input and output are labeled $t_{PLH}$ and $t_{PHL}$, depending on whether the output is changing from a high to a low or from a low to a high. These definitions are extremely important in characterizing the time-domain characteristics of digital circuits.

For the simple *RC* circuit shown back in Fig. 2.21, the delay time is given by

$$t_{delay} = 0.7RC \tag{10.17}$$

and the rise or fall time is given by

$$t_{rise} = 2.2RC \tag{10.18}$$

For our simple digital model of Fig. 10.8, we will assume that the propagation delay time, whether high-to-low or low-to-high, is given by

$$t_{PHL} \approx 0.7 \cdot R_n \cdot C_{tot} \text{ and } t_{PLH} \approx 0.7 \cdot R_p \cdot C_{tot} \tag{10.19}$$

and the output rise and fall times are given by

$$t_{HL} \approx 2.2 \cdot R_n \cdot C_{tot} \text{ and } t_{LH} \approx 2.2 \cdot R_p \cdot C_{tot} \tag{10.20}$$

where $C_{tot}$ is the total capacitance from the drain of the MOSFET to ground. These models for hand calculations **do not give exact results**. The models are useful for determining approximate delay and transition times, usually within a factor of two. They can be used to reveal the location of a speed limitation in a circuit.

**Example 10.2**

Using hand calculations, estimate the rise, fall, and delay times of the following circuits (Fig. 10.10) in both the long- and short-channel CMOS processes. Compare your results to SPICE simulations.

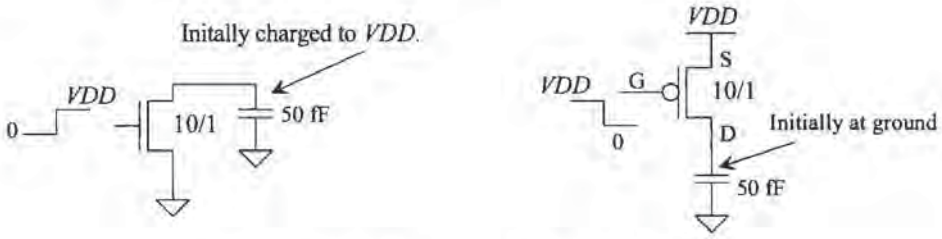



**Figure 10.10** Circuits used in Example 10.2.

The models used for calculating delays are seen in Fig. 10.11. The time constant associated with discharging a load capacitor (the NMOS switch) is

$$t_{PHL} = 0.7 \cdot R_n C_{tot} = 0.7 \cdot R_n \cdot (C_{ox} + C_L) \tag{10.21}$$

The time constant associated with charging a load capacitance (the PMOS switch) is

$$t_{PLH} = 0.7 \cdot R_p C_{tot} = 0.7 \cdot R_p \cdot (C_{ox} + C_L) \tag{10.22}$$

IPR2023-00697
Theta EX2019

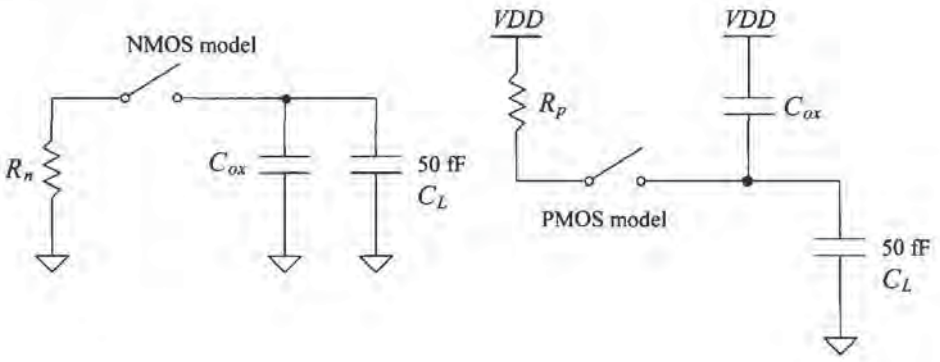



**Figure 10.11** Models used to determine switching times in Example 10.2.

In many practical situations the load capacitance is much larger than the output capacitances of the switches. These equations can then be reduced to

$$t_{PHL} = 0.7 \cdot R_n \cdot C_L \text{ and } t_{PLH} = 0.7 \cdot R_p \cdot C_L \text{ for } C_L >> C_{ox} \qquad (10.23)$$

Using Eqs. (10.21) and (10.22) with the parameters in Table 10.1, we can estimate the delays using the long-channel process as

$$t_{PHL} = 0.7 \cdot 15k \cdot \frac{1}{10} \cdot (1.75\,fF \cdot 10 + 50\,fF) = 70\,ps$$

and

$$t_{PLH} = 0.7 \cdot 45k \cdot \frac{1}{10} \cdot (1.75\,fF \cdot 10 + 50\,fF) = 210\,ps$$

For the short-channel process, we get

$$t_{PHL} = 0.7 \cdot \frac{34k}{10} \cdot (62.5\,aF \cdot 10 + 50\,fF) = 120\,ps$$

and

$$t_{PLH} = 0.7 \cdot \frac{68k}{10} \cdot (62.5\,aF \cdot 10 + 50\,fF) = 240\,ps$$

The simulation results are seen in Fig. 10.12. We can also estimate the output rise and fall times using Eqs. (10.19) and (10.20). For the long-channel devices

$$t_{HL} = \frac{2.2}{0.7} \cdot t_{PHL} = 220\,ps$$

and

$$t_{LH} = \frac{2.2}{0.7} \cdot t_{PLH} = 660\,ps$$

For the short-channel devices

$$t_{HL} = 377\,ps \text{ and } t_{LH} = 754\,ps \blacksquare$$

IPR2023-00697
Theta EX2019

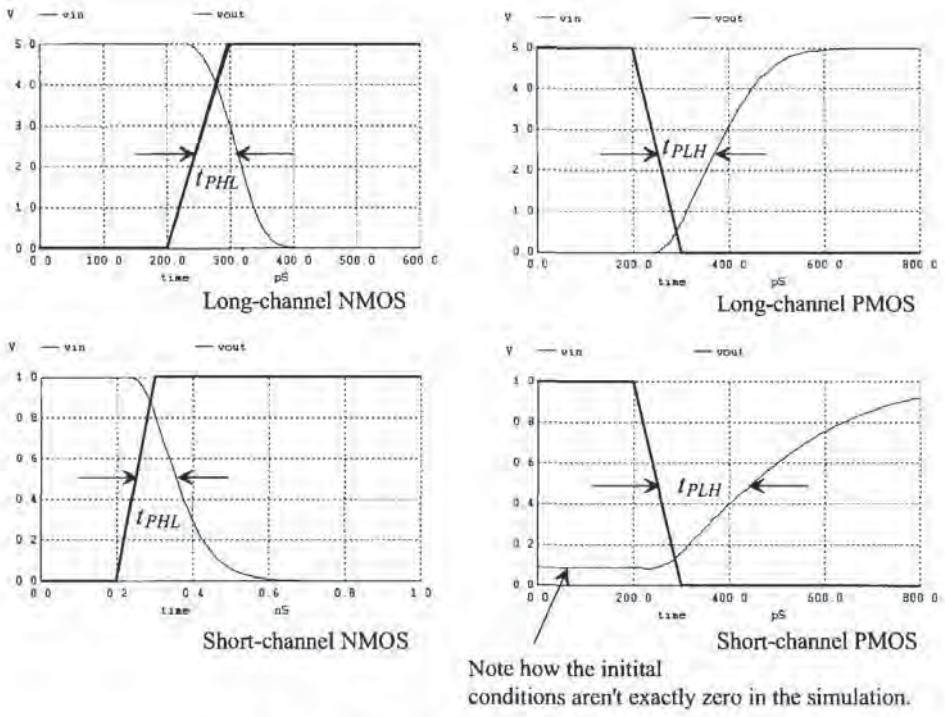



**Figure 10.12** Simulating the circuits in Fig. 10.10.

### 10.1.4 General Digital Design

For general digital design, using the long or short-channel CMOS processes discussed in this book, we can set the drawn size of the NMOS devices to 10/1. To match the effective resistance of the PMOS device to the NMOS device in the long-channel process, Table 10.1, we set the PMOS size to 30/1 (the width of the PMOS is three times the width of the NMOS). For the short-channel process, we use a PMOS device of 20/1 to match the effective switching resistances. Table 10.2 summarizes the parameters for general digital design in the short- and long-channel processes used in this book. For specific digital design solutions we may use a longer device if a higher resistance is needed or a wider device if more drive current through the MOSFET is required.

**Table 10.2** Parameters for general digital design using the long-channel (scale factor is 1 μm) or short-channel (scale factor of 50 nm) CMOS process **used in this book.**

| Technology | Drawn | Actual size | $R_{n,p}$ | $C_{ox,n,p}$ |
|---|---|---|---|---|
| NMOS (long-channel) | 10/1 | 10 μm by 1 μm | 1.5k | 17.5 *fF* |
| PMOS (long-channel) | 30/1 | 30 μm by 1 μm | 1.5k | 52.5 *fF* |
| | | | | |
| NMOS (short-channel) | 10/1 | 0.5 μm by 50 nm | 3.4k | 625 *aF* |
| PMOS (short-channel) | 20/1 | 1 μm by 50 nm | 3.4k | 1.25 *fF* |

IPR2023-00697
Theta EX2019

## 10.2 The MOSFET Pass Gate

Consider the NMOS device seen in Fig. 10.13a. This is the configuration we are accustomed to looking at, that is, where the NMOS device pulls the output to ground. In (b) we flip the MOSFET in (a) on its side and change the labels from ground to "logic 0" and from *VDD* to "logic 1." Note the locations of the source and drain in these configurations. Also note that in (b) the output is pulled all the way to ground (to a good "0"). It can be said that *an NMOS device is good at passing a* "0."

Next look at the configuration in (c). The configurations in (b) or (c) are sometimes called the *pass gate (PG) configuration*. The MOSFET passes the logic level on its input to its output when its gate is driven to *VDD* (when the PG is enabled). Note that when the PG is disabled (its control gate is driven to ground), the outputs are in a high-impedance state (a *Hi-Z* state). The PG can be useful when sharing a bus or a logic circuit. Also note that the inputs and outputs of the PG can be swapped. Logic flow through the PG can be *bi-directional*.

Returning to (c), we see that if the input to the PG is a "1" we can no longer think of the input as the source of the MOSFET like we did in (a) and (b). If we were to do so, then the $V_{GS}$ of the MOSFET would be zero and the MOSFET would be off. If the NMOS device were off while its gate was driven to *VDD*, then this would contradict our comments at the beginning of the chapter associated with Fig. 10.1. To keep the MOSFET on, we need at least a $V_{GS}$ of $V_{THN}$. As seen in (d) of the figure, this means that the NMOS PG passes a "1" with a threshold voltage drop. It can be said that *an NMOS device is not good at passing a* "1." Figure 10.14 shows how the output of a PG varies as its input transitions between ground and *VDD* using the 50 nm CMOS process.

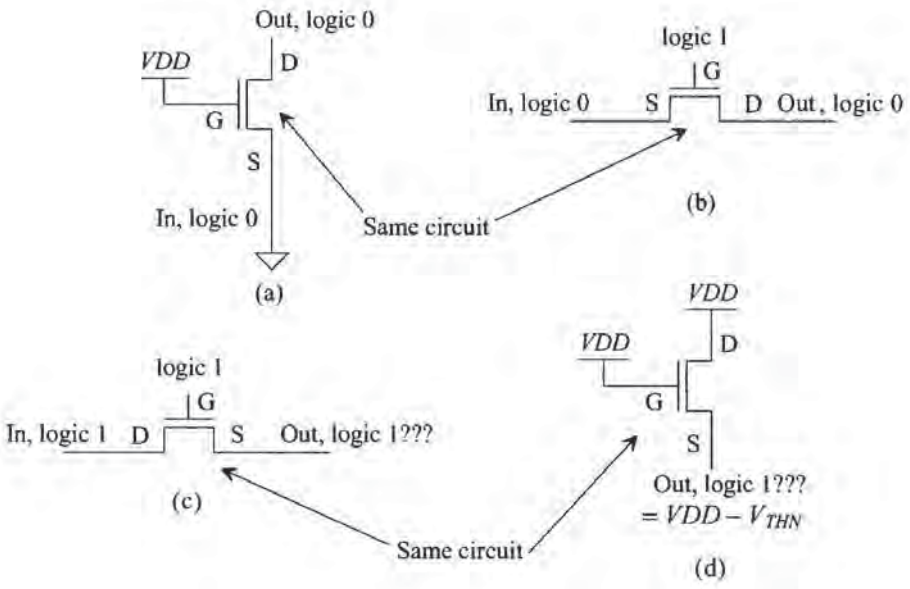



**Figure 10.13** Using the NMOS switch as a pass gate.

IPR2023-00697
Theta EX2019

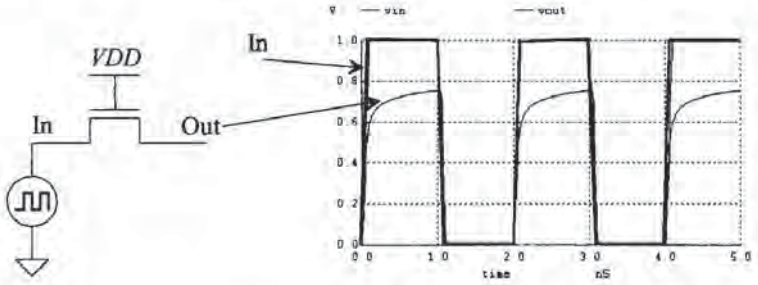



**Figure 10.14** The input and output of an NMOS pass gate.

*The PMOS Pass Gate*

Figure 10.15 shows the operation of the PMOS PG. As expected the operation of the PMOS device is complementary to the NMOS's operation. The PMOS device turns on when its gate is driven to ground. If its gate is pulled to *VDD*, the device is off (and the output is in the Hi-Z state). In Fig. 10.15a the PG is passing a "1" to the output (the $V_{SG}$ is *VDD*). In (b) a "0" is passed to the output. However, noting that the terminals we label drain and source are swapped from (a), the output only gets pulled down to $V_{THP}$. In (b) the $V_{SG}$ of the MOSFET is $V_{THP}$. It can be said that a *PMOS PG is good at passing a 1 and bad at passing a 0.*




**Figure 10.15** How the PMOS device does not pass a logic 0 well.

**Example 10.3**
Estimate the output voltages in the circuits seen in Fig. 10.16.

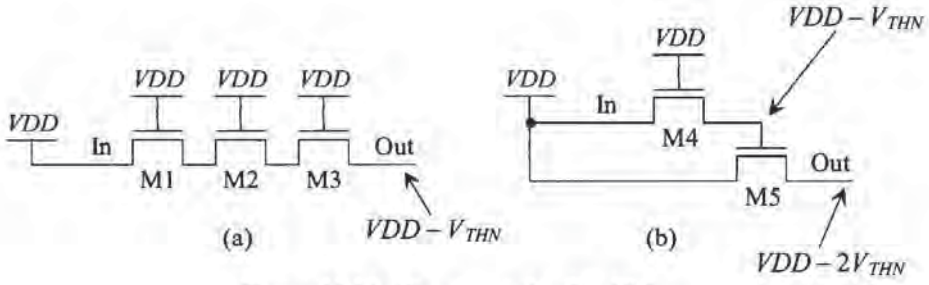



**Figure 10.16** Circuits used in Ex. 10.3.

IPR2023-00697
Theta EX2019

In (a) the output of M1 is $VDD - V_{THN}$. To keep M2 and M3 on, each MOSFET must have a $V_{GS}$ of at least $V_{THN}$. Because the gates of M2 and M3 are already at $VDD$, the output of M1 gets passed through M2 and M3 to the final output of the circuit. As seen in the figure this means the overall output is also $VDD - V_{THN}$. We only take one threshold voltage hit.

In (b) the output of the M4 is $VDD - V_{THN}$. This is the gate voltage of M5. For M5 to be on its gate-source voltage must be greater than $V_{THN}$. The final output is then $VDD - 2V_{THN}$ (again as seen in the figure). ■

### 10.2.1 Delay through a Pass Gate

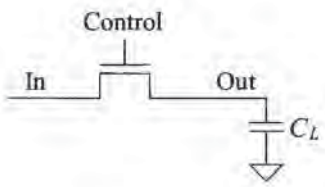
Consider the PG configuration seen in Fig. 10.17a. Let's estimate the delay between the input and the output of the PG. Note that if the input to the circuit is a "0" the PG behaves like the configuration seen in Fig. 10.4 and the output gets pulled all the way down to ground. Let's consider the configuration seen in Fig. 10.17b where the input is transitioning from a "0" to a "1" ($VDD$). The capacitance that the input sees is $C_{ox}/2$. The total load capacitance is

$$C_{tot} = C_L + \frac{C_{ox}}{2} \tag{10.24}$$

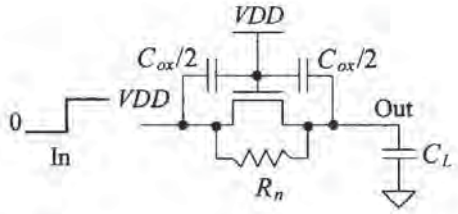
The delay through the PG can be estimated as

$$t_{delay} = 0.7 \cdot R_n C_{tot} = 0.7 \cdot R_n \cdot \left( C_L + \frac{C_{ox}}{2} \right) \tag{10.25}$$

Let's use this result in an example.




(a) Delay from the input to the output through a pass gate.




(b) The capacitances and effective resistance when turning the MOSFET on with an input 1.

**Figure 10.17** Estimating the delay through a pass transistor.

**Example 10.4**

Estimate the delays through the PGs shown in Fig. 10.18. Verify your estimates with simulations. Use the 50 nm (short-channel) CMOS process.

The MOSFETs used in this example have the effective resistances and oxide capacitances listed in Table 10.2. Because the oxide capacitance (the MOSFET's capacitance) is much less than the load capacitance, we can write

$$t_{delay} \approx 0.7 \cdot R_{n,p} C_L \tag{10.26}$$

IPR2023-00697
Theta EX2019

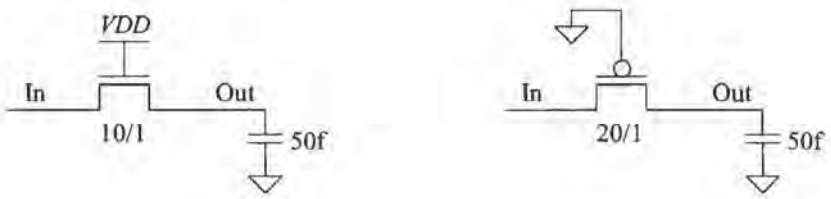



**Figure 10.18** Circuits used to calculate delays in Ex. 10.4.

For the NMOS or PMOS PGs in Fig. 10.18 the delays are estimated as

$$t_{delay} \approx 0.7 \cdot 3.4k \cdot 50fF = 120\,ps$$

The simulation results are seen in Fig. 10.19. If we measure the output delays in Fig. 10.19 at 50% of *VDD* (500 mV), as defined in Fig. 10.9, then we get considerably different values from our hand calculations. The fact that the outputs of the PGs don't swing all the way to the power supply rails changes the points where we would measure the delays (to, say, 50% of the output swing). Again, it's important to note that our hand calculations give approximate delays and, more importantly, can indicate the location of a speed limitation in a digital circuit. The hand calculations won't provide exact delay values. Even if they could, the results would be subjective, dependent on the locations (voltage levels) on the input and output waveforms where we measure the delay. ■

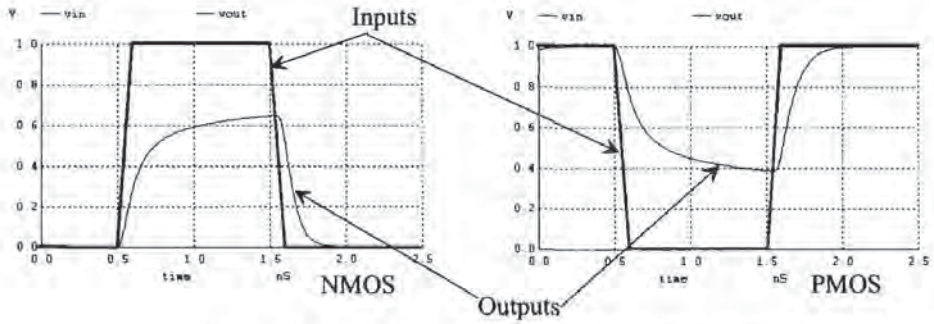



**Figure 10.19** The delay through the PGs in Fig. 10.18.

*The Transmission Gate (The TG)*

The observant reader may be wondering: "If an NMOS PG passes a 0 well and a PMOS PG passes a 1 well, can't we put the two together and pass full logic levels?" The resulting circuit is called a *transmission gate*, *TG*, and is seen in Fig. 10.20. When the select control signal, *S*, is high, the TG is on and the input is passed to the output. The drawbacks of the TG over the PG are increased layout area and the need for two control signals (*S* and its complement). The benefit of using the TG is its rail-to-rail output swing. We'll discuss the TG in more detail in Ch. 13.

IPR2023-00697
Theta EX2019

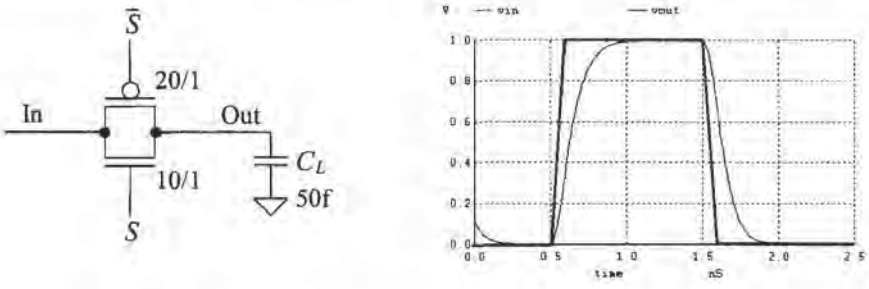



**Figure 10.20** Simulating the operation of a transmission gate.

Also seen in Fig. 10.20 are simulation results showing the delay through the TG. For the values seen, the load capacitance is, again, much larger than the oxide capacitances of the MOSFETs, so we can calculate the delay as

$$t_{delay} = 0.7 \cdot (R_n || R_p) \cdot C_L \tag{10.27}$$

Using the results from Ex. 10.4, the delay is estimated as 60 ps (very close to the simulation results).

### 10.2.2 Delay through Series-Connected PGs

Consider the series connection of (identically sized) NMOS PGs seen in Fig. 10.21. Reviewing Fig. 10.17, we see that capacitance on the internal nodes, in between MOSFETs, is $C_{ox}$ (a contribution of half of $C_{ox}$ from each MOSFET). To approximate the delay through the MOSFETs, we can use Eq. (2.32) or

$$t_{delay} \approx 0.35 \cdot R_n \cdot C_{ox} \cdot l^2 \tag{10.28}$$

Noting $R_n C_{ox}$ is the process characteristic time constant, $\tau_n$, we can quickly estimate delays through series-connected PGs without doing much of a calculation. For example,

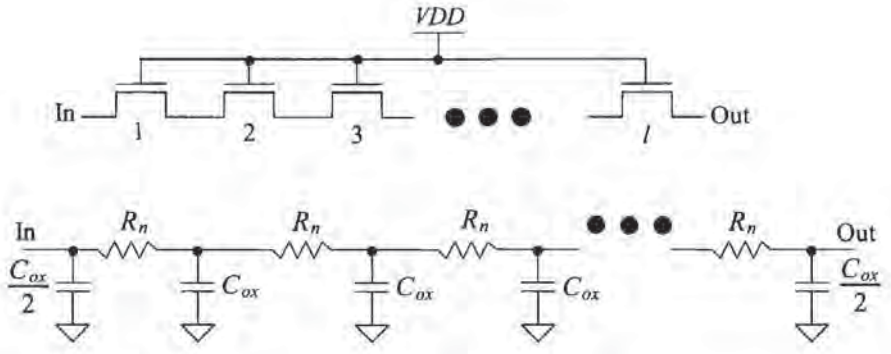



**Figure 10.21** How a series connection of MOSFETs behaves like an RC transmission line (see Fig. 2.22).

IPR2023-00697
Theta EX2019

we know that $\tau_n$, for the 50 nm process, is 2.1 ps from Ex. 10.1. If we have 10 NMOS PGs in a row, the delay through the string is estimated as 73.5 ps.

**Example 10.5**

Estimate the delay through the circuit in Fig. 10.22. Verify the estimate with a SPICE simulation. Use the 50 nm process with 10/1 NMOS devices.

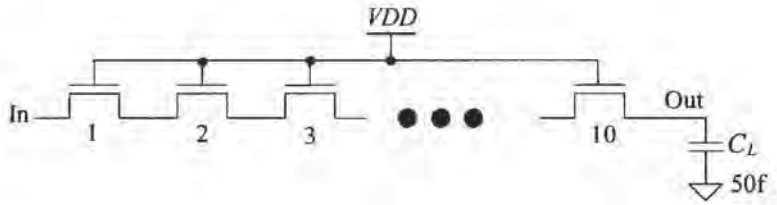



**Figure 10.22** Circuit used in Ex. 10.5.

The total delay is the sum of the RC transmission line delay with the delay in charging the load capacitance through the 10 PGs. This delay can be written for the general case as

$$t_{delay} \approx 0.35 \cdot R_n \cdot C_{ox} \cdot l^2 + 0.7 \cdot l \cdot R_n \cdot C_L \qquad (10.29)$$

For the present example, the delay is

$$t_{delay} \approx \overbrace{0.35 \cdot 2.1\,ps \cdot (10)^2}^{73.5\,ps} + 0.7 \cdot 10 \cdot 3.4k \cdot 50f \approx 1.2\,ns \qquad (10.30)$$

The simulation results are seen in Fig. 10.23. ■

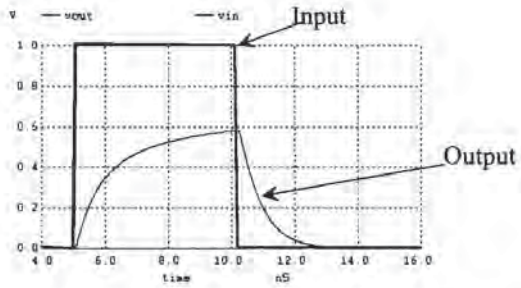



**Figure 10.23** Simulating the operation of the circuit in Fig. 10.22.

## 10.3 A Final Comment Concerning Measurements

Notice that the capacitances we are discussing in this chapter are relatively small. When making measurements it is extremely easy to add a capacitance to the circuit that significantly increases the delays (and may cause circuit failure). Towards understanding this comment in more detail consider the compensated scope probe shown in Fig. 10.24. In (a) we see the input impedance of the oscilloscope (o-scope) is a 1 MΩ resistor in

IPR2023-00697
Theta EX2019

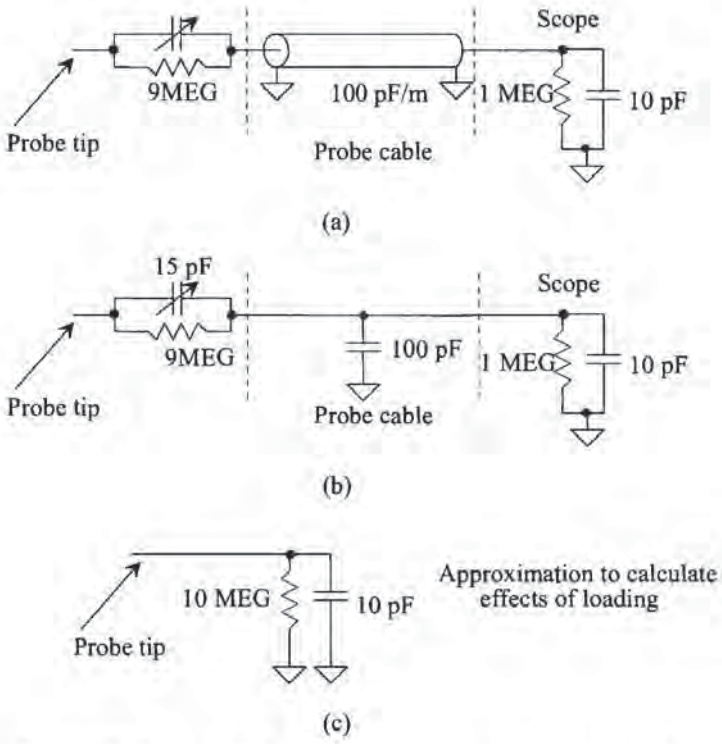



**Figure 10.24** Showing how a common scope probe is assembled.

parallel with a 10 pF capacitor. (The actual values are indicated next to the connector on the front of the particular o-scope.) If we connect a piece of cable (co-axial cable or simply coax) from the circuit under test to the input of the scope, we introduce significant capacitance into the circuit. As seen in (a) the cable's capacitance may be as much as 100 pF/meter. Using a piece of coax to probe the circuit (alone) would then add a 110 pF capacitor and a 1 MΩ resistor to ground at each point we probe! **Understanding this is important.** It's common to see new engineering students, in a digital logic lab for example, probing with a cable (that is, without an o-scope probe). They may wonder why their circuits don't work or only work at slow speeds.

To compensate for the cable capacitance, the probe tip has a series resistor and capacitor added in between the cable and the probe tip, (b). This combination of a cable and probe tip RC is called a *compensated scope probe.* The RC in the probe tip is adjusted to have nine times the impedance of the RC from the scope's input to ground (the cable capacitance in parallel with the scope's input impedance) over all frequencies of interest. In other words, a 10:1 voltage divider exists between the probe tip and the input of the scope (and so the minimum signal we can measure increases when using a compensated scope probe). The big benefit, as seen in the approximation for the loading in (c), is that the size of the capacitance introduced into the circuit is reduced. For probing on-chip (or on-wafer), special probes are used with active devices in the probe tips (to reduce the probe's loading on the circuit it is measuring). Active probe tips, called femtoprobes, can be purchased that only have femtofarads amounts of loading.

IPR2023-00697
Theta EX2019

## ADDITIONAL READING

[1] J. P. Uyemura, *Introduction to VLSI Circuits and Systems*, John Wiley and Sons Publishers, 2002. ISBN 0-471-12704-3.

[2] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI-Design Techniques for Analog and Digital Circuits,* McGraw-Hill Publishing Company, 1990. ISBN 0-07-023253-9.

## PROBLEMS

**10.1** Using the parameters in Table 6.2, compare the hand-calculated effective digital switching resistance from Eq. (10.6) to the empircially derived values given in Table 10.1.

**10.2** Regenerate Fig. 10.14 for the PMOS device.

**10.3** Using SPICE verify the results of Ex. 10.3.

**10.4** Replacing the NMOS PGs in Fig. 10.16 with PMOS PGs and changing the *VDD*-connected nodes to ground-connected nodes, show, and verify with simulations, the outputs of the two modified circuits.

**10.5** For the following circuits estimate the delay between the input and the output. Use the 50 nm (short-channel CMOS) process. Verify the estimates with SPICE.

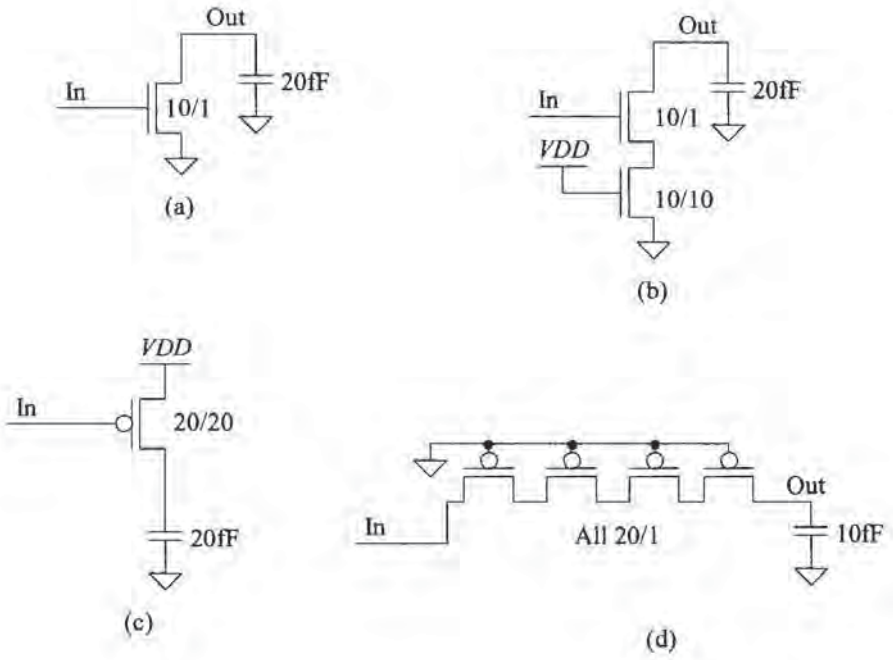



**Figure 10.25** Circuits used in Problem 10.5.

IPR2023-00697
Theta EX2019

## Possible Student Projects

This section lists some possible student projects for fabrication through the MOSIS service (see Ch. 1). Generally, two to four student projects should be implemented on one chip. MOSIS will return to the MOSIS liaison (generally, the course instructor) several copies of each chip design submitted.

The design rule-checked designs should be turned in along with (1) one sheet of paper showing the logic level diagrams *and* pin connections so that whoever is evaluating the chip can quickly determine functionality, and (2) final reports that consist of a block diagram, schematic diagram, layout information, hand calculations, SPICE simulations, and clear explanations (and trade-offs) of the operation of the circuit.

1. Quad 2-input MUX
2. Clock-doubling circuit using exclusive OR gate
3. Buffer with tristate outputs
4. SR flipflop with tristate outputs
5. Edge-triggered T flipflop
6. Edge-triggered D flipflop
7. Schmitt trigger
8. 1-of-16 decoder
9. Up counter with asynchrounous reset
10. 4-bit static shift register
11. 4-bit dynamic shift register
12. 2-bit adder with carryout
13. Current-starved VCO with center frequency of 20 MHz
14. 2-bit bidirectional transceiver
15. PE gate to implement $X = \overline{A + BCD + EF}$
16. One-shot whose output pulse width is determined by external RC
17. Buffer for driving a 20 pF load with minimum delay
18. Buffer for driving a 20 pF load with smaller layout area

Advanced projects

19. A 64-bit static RAM including a storage cell, addressing and decoding circuitry, buffers, a write/read enable, and a chip select.
20. Charge pump (voltage generator). The input to the charge pump is *VDD* (= 5 V) and the output is –3 V. The circuit should be fully simulated. The reference, oscillator, and feedback should be fully simulated and discussed in the final report.

IPR2023-00697
Theta EX2019

21. A 64-bit DRAM, including a storage cell, addressing and decoding circuitry, buffers, a write/read enable, and a chip select.

22. A DPLL which will take a 1 MHz input and generate a 4 MHz output. The output should follow the input for frequency changes from 900 kHz to 1.1 MHz. You should discuss the transient properties of the DPLL, as well as present a detailed design of the phase detector, VCO, and loop filter. The entire design should be monolithic; that is, no external components should be used.

IPR2023-00697
Theta EX2019

Chapter

11

# The Inverter

The CMOS inverter is a basic building block for digital circuit design. As Fig. 11.1 shows, the inverter performs the logic operation of $A$ to $\overline{A}$. When the input to the inverter is connected to ground, the output is pulled to *VDD* through the PMOS device M2 (and M1 shuts off). When the input terminal is connected to *VDD*, the output is pulled to ground through the NMOS device M1 (and M2 shuts off). The CMOS inverter has several important characteristics that are addressed in this chapter: for example, its output voltage swings from *VDD* to ground unlike other logic families that never quite reach the supply levels. Also, the static power dissipation of the CMOS inverter is practically zero, the inverter can be sized to give equal sourcing and sinking capabilities, and the logic switching threshold can be set by changing the size of the device.

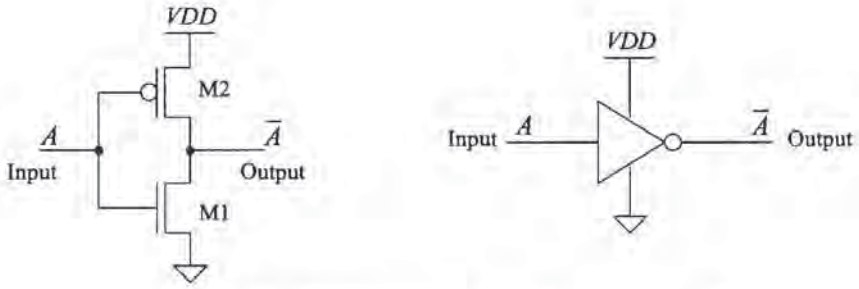



**Figure 11.1** The CMOS inverter, schematic, and logic symbol.

## 11.1 DC Characteristics

Consider the inverter shown in Fig. 11.2 and the associated transfer characteristic plot. In region 1 of the transfer characteristics, the input voltage is sufficiently low (typically less than the threshold voltage of M1), so that M1 is off and M2 is on ($V_{SG} >> V_{THP}$). As $V_{in}$ is increased, both M2 and M1 turn on (region 2). Increasing $V_{in}$ further causes M2 to turn off and M1 to fully turn on, as shown in region 3.

IPR2023-00697
Theta EX2019

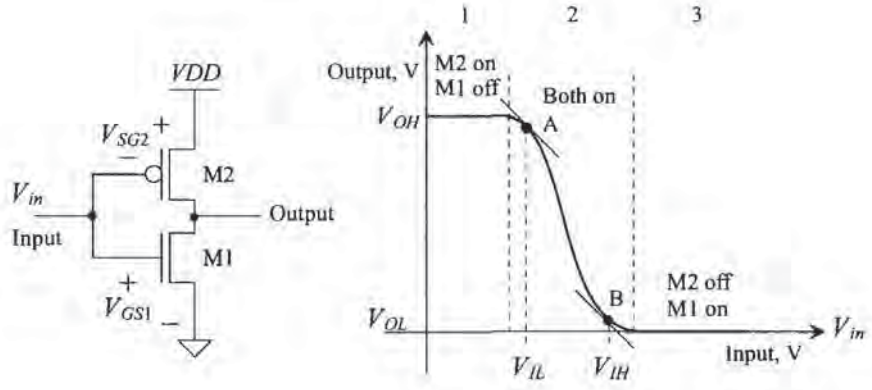



**Figure 11.2** The CMOS inverter transfer characteristics.

The maximum output "high" voltage is labeled $V_{OH}$ and the minimum output "low" voltage, $V_{OL}$. Points A and B on this curve are defined by the slope of the transfer curves equaling –1. Input voltages less than or equal to the voltage $V_{IL}$, defined by point A, are considered a logic low on the input of the inverter. Input voltages greater than or equal to the voltage $V_{IH}$, defined by point B, are considered a logic high on the input of the inverter. Input voltages between $V_{IL}$ and $V_{IH}$ do not define a valid logic voltage level. Ideally, the difference in $V_{IL}$ and $V_{IH}$ is zero; however, this is never the case in real logic circuits.

**Example 11.1**

Using SPICE, plot the transfer characteristics for the inverter seen in Fig. 11.3 in both the long- and short-channel CMOS processes used in this book. From the plot, determine $V_{IH}$, $V_{IL}$, $V_{OH}$, and $V_{OL}$.

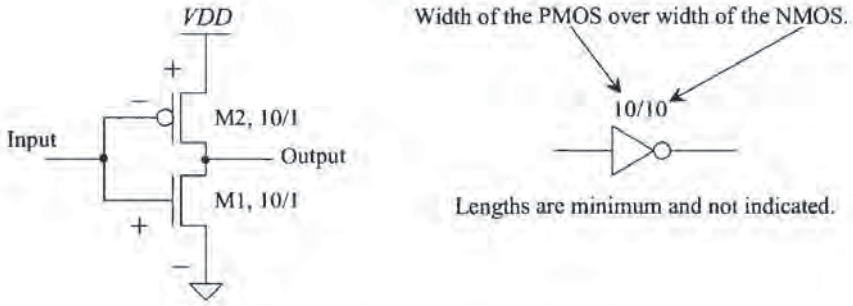



**Figure 11.3** Inverter used in Ex. 11.1.

The inverter *voltage transfer curves* (VTCs) are shown in Fig. 11.4. Notice how the *VDD* used for the long-channel process is 5 V, while the *VDD* used in the short-channel, process is 1 V. The output high voltage, $V_{OH}$, is *VDD* and the output low voltage, $V_{OL}$, is ground (for both inverters). For the inverter using the

IPR2023-00697
Theta EX2019

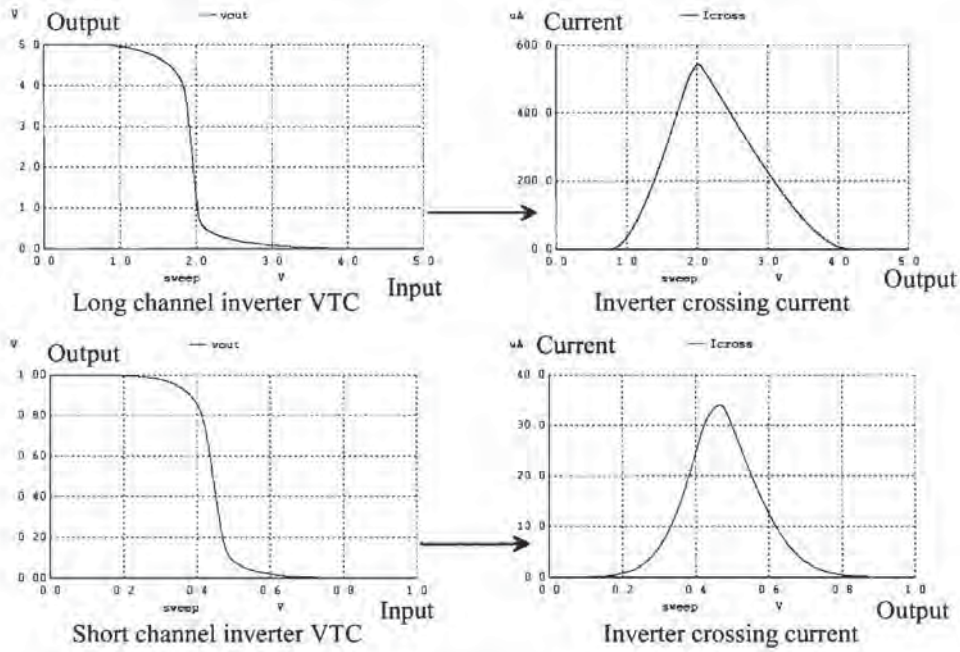



**Figure 11.4** Voltage transfer curves (VTCs) for a long and a short channel inverter.

long-channel process, $V_{IL}$ is approximately 1.8 V while, when using the short-channel process, $V_{IL}$ is approximately 400 mV. For $V_{IH}$ we get 2.1 V and 500 mV for the long- and short-channel processes, respectively.

Note that in Fig. 11.4 we also plotted the crossing current (the inverter's output voltage crossing between a logic 1 and a logic 0). This is the current that flows when the inverter is operating in region 2 in Fig. 11.2 (the inverter's input transitioning from a high to a low or from a low to a high). If the inverter's input transitions quickly, the amount of charge pulled from *VDD* (the amount of time the inverter is operating in region 2) is small. However, if the inverter's input logic signal transitions slowly or the logic levels don't swing all the way to the power supply rails (like what we get with the pass gates discussed in the last chapter, see Fig. 10.19), it's possible for a significant current to flow through the inverter (important!) ■

*Noise Margins*

The noise margins of a digital gate or circuit indicate how well the gate will perform under noisy conditions. The noise margin for the high logic levels is given by

$$NM_H = V_{OH} - V_{IH} \tag{11.1}$$

and the noise margin for the low logic levels is given by

$$NM_L = V_{IL} - V_{OL} \tag{11.2}$$

For $VDD = 1$ V, the ideal noise margins are 500 mV; that is, $NM_L = NM_H = VDD/2$.

IPR2023-00697
Theta EX2019

*Inverter Switching Point*

Consider the transfer characteristics of the basic inverter as shown in Fig. 11.5. Point C corresponds to the point on the curve when the input voltage is equal to the output voltage. At this point, the input (or output) voltage is called the *inverter switching point voltage*, $V_{SP}$, and both MOSFETs in the inverter are in the saturation region. Since the drain current in each MOSFET must be equal, the following is true:

$$\frac{\beta_n}{2}(V_{SP} - V_{THN})^2 = \frac{\beta_p}{2}(VDD - V_{SP} - V_{THP})^2 \tag{11.3}$$

Solving for $V_{SP}$ gives

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \tag{11.4}$$

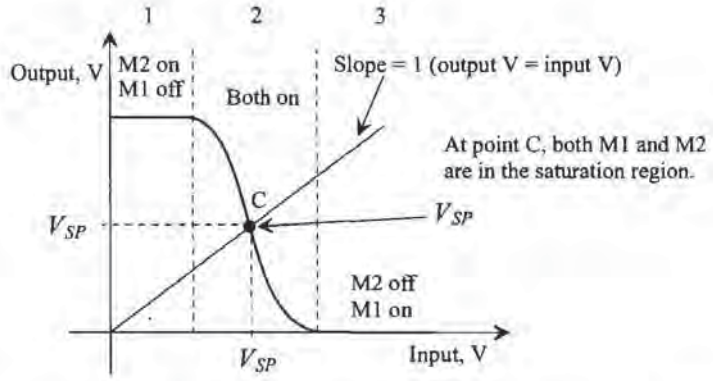



**Figure 11.5** Transfer characteristics of the inverter showing the switching point.

*Ideal Inverter VTC and Noise Margins*

The ideal voltage transfer curves for an inverter are seen in Fig. 11.6. The ideal switching point voltage, $V_{SP}$, is $VDD/2$. As seen in Eqs. (11.1) and (11.2), this makes the noise margins equal to ensure the best performance (a noise margin, say the logic low level, isn't improved at the cost of the other margin). When looking at Fig. 11.6, notice that, unlike what is seen in Fig. 11.5, the inverter never operates where both MOSFETs are on. The input to the inverter is recognized as either a 1 or a 0.

**Example 11.2**

Estimate $\beta_n$ and $\beta_p$ so that the switching point voltage of a CMOS inverter designed in the long-channel CMOS process is 2.5 V (= $VDD/2$).

Solving Eq. (11.4) with $V_{SP}$ = 2.5 V for the ratio $\beta_n/\beta_p$ gives a value of approximately unity. That is,

IPR2023-00697
Theta EX2019

$$\beta_n = \beta_p = KP_n \frac{W_1}{L_1} = KP_p \frac{W_2}{L_2}$$

Since $KP_n = 3KP_p$, the width of the PMOS device must be three times the width of the NMOS, assuming equal-length MOSFETs. For $V_{SP} = 2.5$ V, this requires

$$W_2 = 3W_1$$

which is also the requirement for making $R_n = R_p$. This is an important practical result. It shows how the electron and hole mobilities are related to both the effective switching resistances and the switching point voltage. As seen in Table 10.2, we often increase the widths of the PMOS devices to try to center $V_{SP}$ and equate the propagation delays (the pull-up resistance, $R_p$ , is the same as the pull-down resistance $R_n$). ■

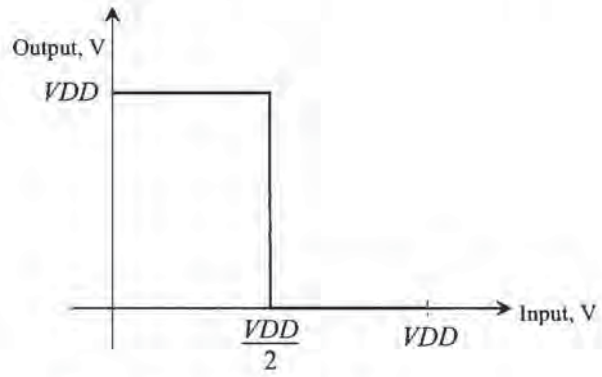



**Figure 11.6** Ideal VTCs for the inverter.

**Example 11.3**

Show, using SPICE, that the inverter seen in Fig. 11.7 and implemented in the short-channel CMOS process has a switching point voltage close to the ideal value of *VDD*/2. Comment on the sizes of the devices.

The simulation results are seen (also) in Fig. 11.7. Because we've sized the width of the PMOS to twice the width of the NMOS (see Table 10.2), the switching point is close to the ideal value of *VDD*/2. We know that short-channel devices don't follow the square-law models and so we can't use Eq. [11.4] to calculate the $V_{SP}$ in our 50 nm process. We can, however, get a good estimate using the effective switching resistances as seen in the figure (a voltage divider). When $R_p = R_n$ , the switching point voltage is close to ideal. By changing the widths of the devices, we can adjust the switching point voltage. In other words, for a short-channel CMOS process we can use

$$V_{SP} = VDD \cdot \frac{R_n}{R_n + R_p} \tag{11.5}$$

to estimate $V_{SP}$. This equation does have limitations. For example, if $R_n >> R_p$ , then this equation indicates $V_{SP}$ is *VDD*. However, we know that $V_{SP}$ has to be

IPR2023-00697
Theta EX2019

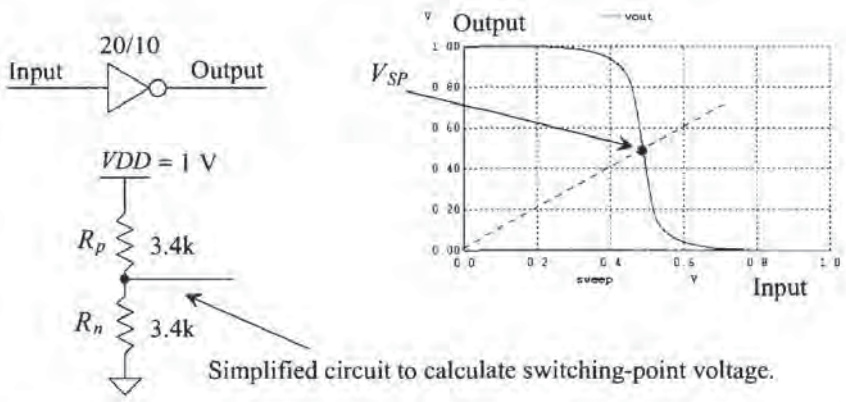



**Figure 11.7** Switching point voltage for an inverter in the short-channel process.

greater than $V_{THN}$ and less than $VDD - V_{THP}$ (make sure that this is understood). ■

---

**Example 11.4**

Show, using SPICE and the long-channel process, the transfer curves for the CMOS inverter with transconductance ratios $\beta_n/\beta_p$ of 3, 1, and 1/3. Explain what changing the inverter ratio does to the transfer characteristics.

The simulation results are seen in Fig. 11.8. For all three DC sweeps the MOSFET's lengths are 1. For the case when $\beta_n/\beta_p = 1$, the $W_n = 10$ and $W_p = 30$. For the case when $\beta_n/\beta_p = 3$, the $W_n = 10$ and $W_p = 10$, etc. Increasing the strength of the NMOS (increasing the NMOS's width, which decreases $R_n$) causes the switching point voltage to decrease. We also get a decrease in $V_{SP}$ by decreasing the strength of the PMOS device (which increases $R_p$). ■

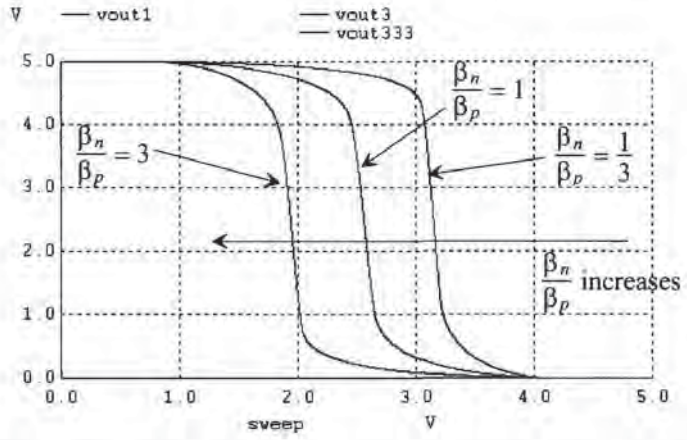



**Figure 11.8** Sizing the inverter changes the switching point voltages.

IPR2023-00697
Theta EX2019

## 11.2 Switching Characteristics

The switching behavior of the inverter can be generalized by examining the parasitic capacitances and resistances associated with the inverter. Consider the inverter shown in Fig. 11.9 with its equivalent digital model. Although the model is shown with both switches open, in practice one of the switches is closed, keeping the output connected to *VDD* or ground. The effective input capacitance of the inverter is

$$C_{in} = \frac{3}{2}(C_{ox1} + C_{ox2}) = C_{inn} + C_{inp} \tag{11.6}$$

The effective output capacitance of the inverter is simply

$$C_{out} = C_{ox1} + C_{ox2} = C_{outn} + C_{outp} \tag{11.7}$$

The intrinsic propagation delays of the inverter are

$$t_{PLH} = 0.7 \cdot R_{p2} \cdot C_{out} \text{ and } t_{PHL} = 0.7 \cdot R_{n1} \cdot C_{out} \tag{11.8}$$

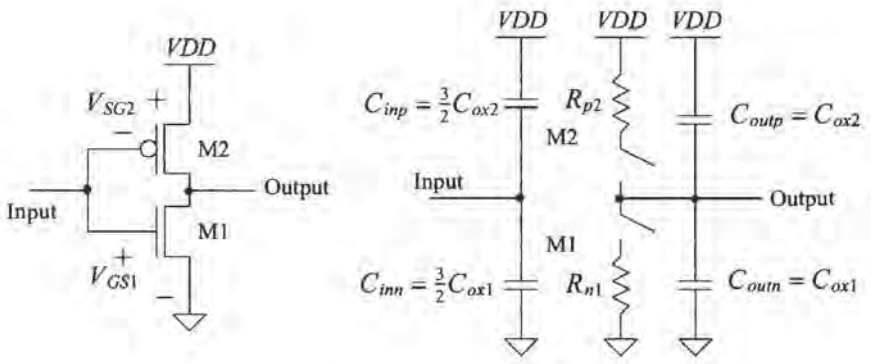



**Figure 11.9** The CMOS inverter switching characteristics using the digital model.

**Example 11.5**

Estimate and simulate the intrinsic propagation delays of the inverter seen in Fig. 11.7. Estimate the inverter's input capacitance.

From the data in Table 10.2 and Eqs. (11.6) to (11.8), we can write

$$t_{PHL} = t_{PLH} = 0.7 \cdot 3.4k \cdot (0.625 + 1.25)\,fF = 4.5\,ps$$

The simulation results are seen in Fig. 11.10. The intrinisic delays are considerably larger than this calculation (around 20 ps).

Using Eq. (11.6), the inverter's input capacitance is

$$C_{in} = \frac{3}{2}(0.625 + 1.25)\,fF = 2.8\,fF$$

Sizing up the width of the PMOS so that its effective resistance is equal to $R_n$ has the unwanted effect of increasing the inverter's input capacitance. ■

IPR2023-00697
Theta EX2019

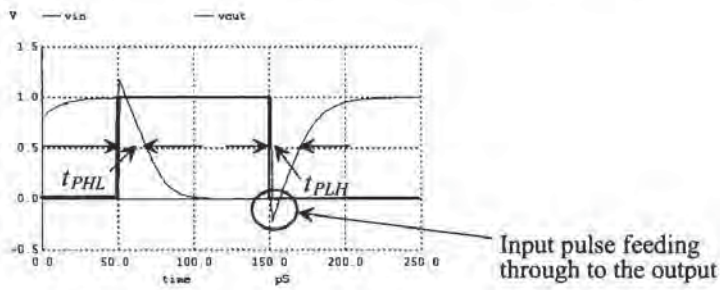



**Figure 11.10** The intrinsic propagation delays of an inverter.

The propagation delays for an inverter driving a capacitive load are

$$t_{PLH} = 0.7 \cdot R_{p2} \cdot C_{tot} = 0.7 \cdot R_{p2} \cdot (C_{out} + C_{load}) \tag{11.9}$$

and

$$t_{PHL} = 0.7 \cdot R_{n1} \cdot C_{tot} = 0.7 \cdot R_{n1} \cdot (C_{out} + C_{load}) \tag{11.10}$$

where $C_{tot}$ is the total capacitance on the output of the inverter, that is, the sum of the output capacitance of the inverter, any capacitance of interconnecting lines, and the input capacitance of the following gate(s).

**Example 11.6**

Estimate and simulate the propagation delays for the circuit seen in Fig. 11.11. Use the 50 nm CMOS process.

Because the load capacitance is much larger than the output capacitance of the inverter, we can rewrite

$$t_{PLH} = 0.7 \cdot R_{p2} \cdot C_{tot} \approx 0.7 \cdot R_{p2} \cdot C_{load} = 120\ ps$$

and

$$t_{PHL} = 0.7 \cdot R_{n1} \cdot C_{tot} \approx 0.7 \cdot R_{n1} \cdot C_{load} = 120\ ps$$

The simulation results are seen in Fig. 11.11. Notice that we are treating the MOSFETs used in the digital circuits as resistors and calculating the delays with a simple product of these resistors with the load capacitance. ■

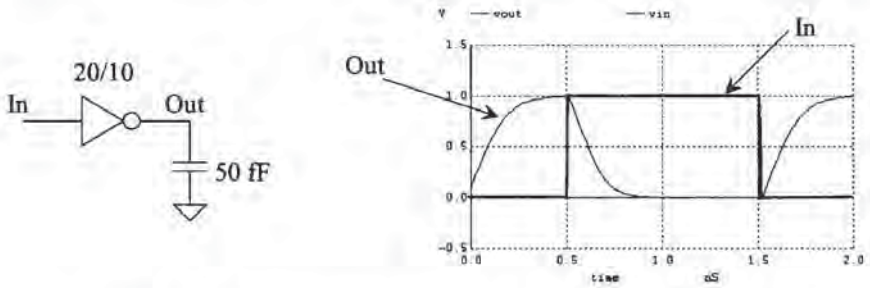



**Figure 11.11** The delay associated with an inverter driving a 50 fF load.

IPR2023-00697
Theta EX2019

*The Ring Oscillator*

The odd number of inverters in the circuit shown in Fig. 11.12 forms a closed loop with positive feedback and is called a ring oscillator. The oscillation frequency is given by

$$f_{osc} = \frac{1}{n \cdot (t_{PHL} + t_{PLH})} \tag{11.11}$$

assuming that the inverters are identical and $n$ is the number (odd) of inverters in the ring oscillator. Since the ring oscillator is self-starting, it is often added to a test portion of a wafer to indicate the speed of a particular process run. The sum of the high-to-low and low-to-high delays is used to calculate the period of the oscillation because each inverter switches twice during a single oscillation period.

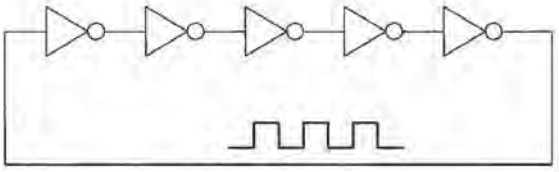


**Figure 11.12** A five-stage ring oscillator.

When identical inverters are used the capacitance on the inverter's input/output is the sum of an inverter's input capacitance with the inverter's output capacitance, see Fig. 11.9, or

$$C_{tot} = \overbrace{C_{oxp} + C_{oxn}}^{C_{out}} + \overbrace{\frac{3}{2} \cdot (C_{oxp} + C_{oxn})}^{C_{in}} = \frac{5}{2} \cdot (C_{oxp} + C_{oxn}) \tag{11.12}$$

where, again, $C_{oxp} = C'_{ox} \cdot W_p \cdot L_p \cdot (scale)^2$ and $C_{oxn} = C'_{ox} \cdot W_n \cdot L_n \cdot (scale)^2$. The delay is then calculated using

$$t_{PHL} + t_{PLH} = 0.7 \cdot (R_n + R_p) \cdot C_{tot} \tag{11.13}$$

*Dynamic Power Dissipation*

Consider the CMOS inverter driving a capacitive load shown in Fig. 11.13. Each time the inverter changes states, it must either supply a charge to $C_{tot}$ or sink the charge stored on $C_{tot}$ to ground. If a square pulse is applied to the input of the inverter with a period T and frequency, $f_{clk}$, the average amount of current that the inverter must pull from *VDD*, recalling that current is being supplied from *VDD* only when the PMOS device is on, is

$$I_{avg} = \frac{Q_{Ctot}}{T} = \frac{VDD \cdot C_{tot}}{T} \tag{11.14}$$

The average dynamic power dissipated by the inverter is

$$P_{avg} = VDD \cdot I_{avg} = \frac{C_{tot} \cdot VDD^2}{T} = C_{tot} \cdot VDD^2 \cdot f_{clk} \tag{11.15}$$

IPR2023-00697
Theta EX2019

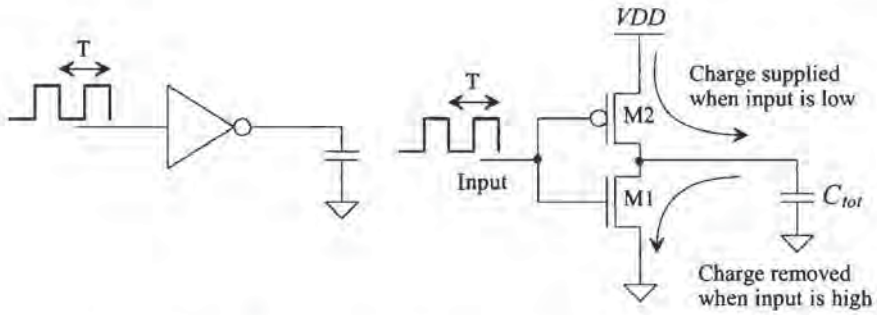



**Figure 11.13** Dynamic power dissipation of the CMOS inverter.

Notice that the power dissipation is a function of the clock frequency, power supply voltage, and load capacitance. A great deal of effort is put into reducing the power dissipation in CMOS circuits. One of the major advantages of dynamic logic (Ch. 14) is its lower power dissipation.

To characterize the speed of a digital process, a term called the power delay product (*PDP*) is often used. The *PDP*, measured in Joules, is defined by

$$PDP = P_{avg} \cdot (t_{PHL} + t_{PLH}) \tag{11.16}$$

These terms can be determined from a ring oscillator. The *PDP* is frequently used to compare different technologies or device sizes; for example, a GaAs process can be compared with a 50 nm CMOS process. Although the GaAs process may have a lower propagation delay, the power dissipation may be larger and result in a larger *PDP*.

---

**Example 11.7**

Estimate the oscillation frequency of an 11-stage ring oscillator in the 50 nm process using the inverter seen in Fig. 11.7 (see Table 10.2). Verify the estimate with simulations.

Using the data in Table 10.2 and Eqs. (11.11) – (11.13), we can write

$$C_{tot} = \frac{5}{2} \cdot (1.25 + 0.625)\, fF = 4.7\, fF$$

and

$$t_{PHL} + t_{PLH} = 0.7 \cdot (3.4k + 3.4k) \cdot 4.7\, fF = 22\, ps$$

For an 11-stage ring oscillator, we get an oscillation frequency of

$$f_{osc} = \frac{1}{11 \cdot (22\, ps)} = 4.1\ GHz$$

The simulation results are seen in Fig. 11.14. The simulated oscillation frequency is close to 1.25 GHz or considerably different from our hand calculations (as will be the case when delays are close to the intrinsic values). The average power, $P_{avg}$, dissipated by a single inverter in this ring oscillator, using Eq. (11.15), is estimated as 19.6 μW. The PDP for the 50 nm process is then $431 \times 10^{-18}$ *J*. ■

IPR2023-00697
Theta EX2019

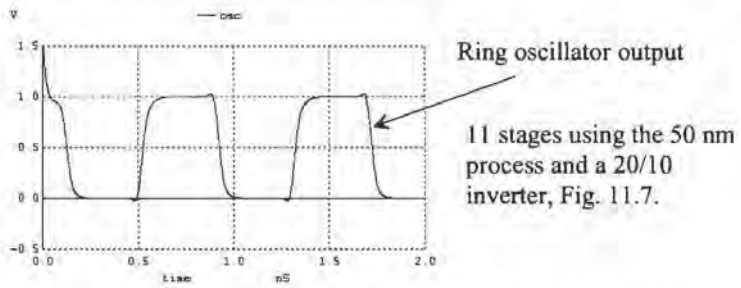



**Figure 11.14** Oscillation frequency for the ring oscillator described in Ex. 11.7.

## 11.3 Layout of the Inverter

If care is not taken when laying out CMOS circuits, the parasitic devices present can cause a condition known as latch-up. Once latch-up occurs, the inverter output will not change with the input; that is, the output may be stuck in a logic state. To correct this problem, the power must be removed. Latch-up is especially troubling in output driver circuits.

*Latch-Up*

Figure 11.15 illustrates two methods of laying out an inverter. Notice how the cell's inputs and outputs are on metal2, while the power and ground conductors are routed on metal1 in the standard cell frame (see Fig. 4.15 and the associated discussion). The cross-sectional view in Fig. 11.16 shows both the NMOS and PMOS devices that make up an inverter (and associated parasistics). Notice that in Fig. 11.10, the input pulse feeds through the gate-drain capacitance of the MOSFETs to the output of the inverter. This causes the output to change in the same direction as the input before the inverter starts to switch. This feedthrough and the parasitic bipolar transistors can cause the latch-up.

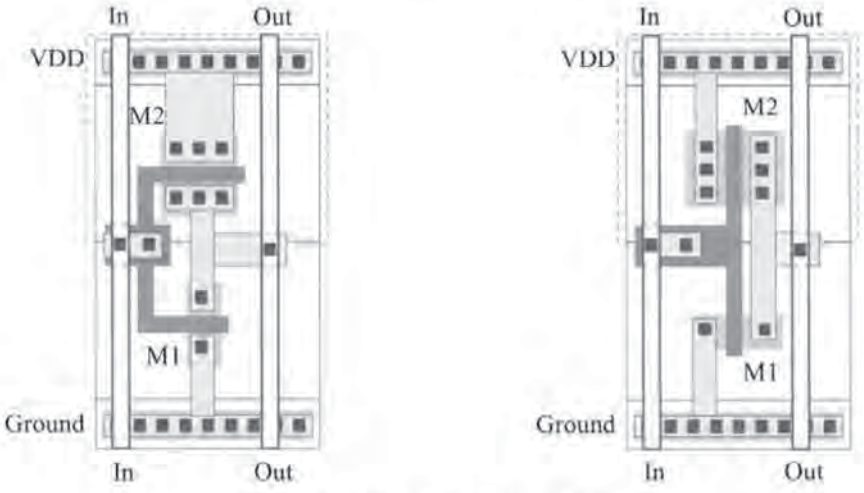



**Figure 11.15** Layout styles for inverters.

IPR2023-00697
Theta EX2019

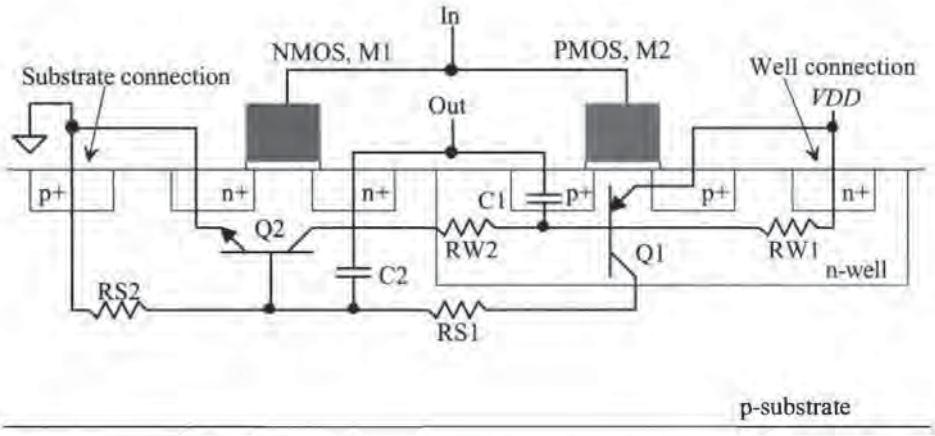



**Figure 11.16** Cross-sectional view of an inverter showing parasitic bipolar transistors and resistors.

In Fig. 11.16, the emitter, base, and collector of transistor Q1 are the source of the PMOS, n-well, and substrate, respectively. Transistor Q2's collector, base, and emitter are the n-well, substrate, and source of the NMOS transistor. Resistors RW1 and RW2 represent the effects of the resistance of the n-well, and resistors RS1 and RS2 represent the resistance of the substrate. The capacitors C1 and C2 represent the drain implant depletion capacitance, that is, the capacitance between the drains of the transistors and the n-well (for C1) and substrate (for C2). The parasitic circuit resulting from the inverter layout is shown in Fig. 11.17.

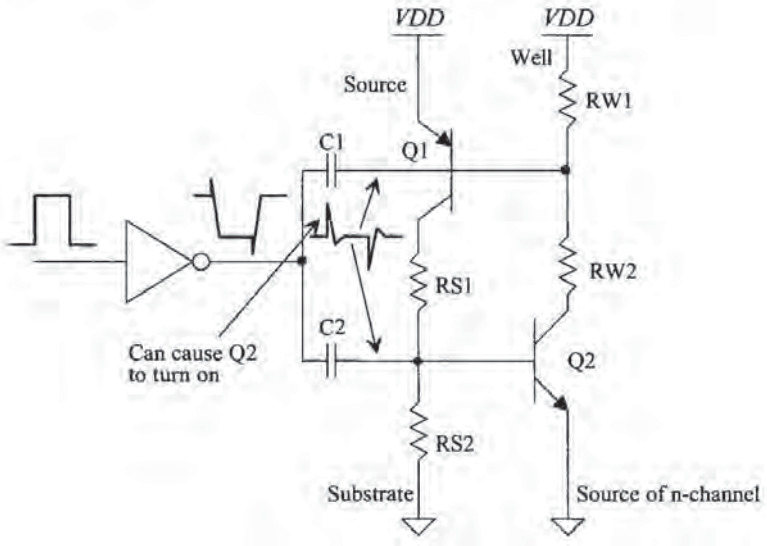



**Figure 11.17** Schematic used to describe latch-up.

IPR2023-00697
Theta EX2019

If the output of the inverter switches fast enough, the pulse fed through C2 (for positive-going inputs) can cause the base-emitter junction of Q2 to become forward biased. This then causes the current through RW2 and RW1 to increase, turning on Q1. When Q1 is turned on, the current through RS1 and RS2 increases, causing Q2 to turn on harder. This positive feedback will eventually cause Q2 and Q1 to turn on fully and remain that way until the power is removed and reapplied. A similar argument can be given for negative-going inputs feeding through C1, *VDD* bouncing upwards, or ground bouncing downwards.

Several techniques reduce the latch-up problem. One technique is to slow the rise and fall times of the logic gates, reducing the amount of signal fed through C1 and C2. Reducing the areas of M1 and M2's drains lowers the size of the depletion capacitance and the amount of signal fed through. Probably the best method of reducing latch-up effects is to reduce the parasitic resistances RW1 and RS2. If these resistances are zero, Q1 and Q2 never turn on. The value of these resistances, as seen in Fig. 11.16, is a strong function of the distance between the well and substrate contacts. Simply put, the closer these contacts are to the MOSFETs used in the inverter, the less likely it is that the inverter will latch up. These contacts should be plentiful as well as close. Placing substrate and well contacts between the PMOS and NMOS devices provides a low-resistance connection to *VDD* and ground, significantly helping to reduce latchup (see Fig. 11.18 for a simple layout example). Placing n+ and p+ areas between or around circuits reduces the amount of signal reaching a given circuit from another circuit. These implants are sometimes called guard rings (see Fig. 5.5). Notice that poly cannot be used to connect the gates of the MOSFETs, since poly over the n+ or p+ will be interpreted as a MOSFET. Therefore, metal2 is used to connect the MOSFETs together. The cost of reducing the possibility of latch-up is a more complicated layout in a larger area.

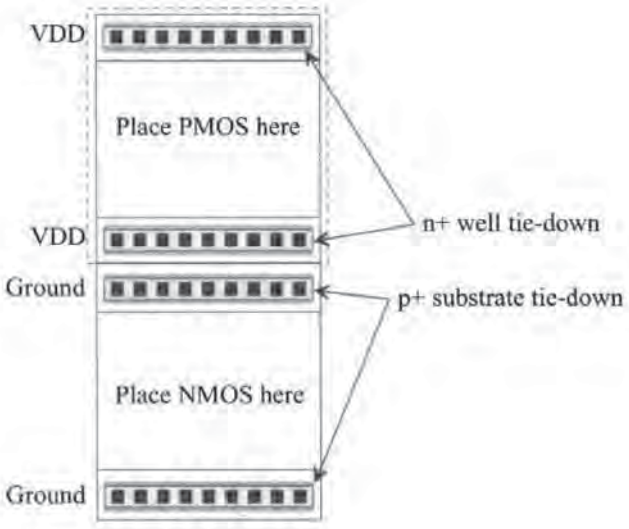



**Figure 11.18** Adding an extra implant between NMOS and PMOS to reduce latch-up.

IPR2023-00697
Theta EX2019

## 11.4 Sizing for Large Capacitive Loads

Designing a circuit to drive large capacitive loads with minimum delay is important when driving off-chip loads. As we saw in Ex. 11.6, a large load capacitance can drastically affect the delay through an inverter. Remembering our discussion in Sec. 10.3, we see that using a standard scope probe to measure an output signal can result in delays that are microseconds in length. In order to avoid this situation, we add a buffer circuit (a string of inverters) between the on-chip logic and the bonding pads. In this section we discuss how to design (select the widths of the MOSFETs) for low delays.

*Buffer Topology*

Consider the inverter string (a buffer) driving a load capacitance, labeled $C_{load}$ and shown in Fig. 11.19. Moving towards the load in a cascade of the $N$ inverters, each inverter larger than the previous by a factor $A$ (that is, the width of each MOSFET is multiplied by $A$), a minimum delay can be obtained as long as $A$ and $N$ are picked correctly. Each inverter's input capacitance is larger than the previous inverter's input capacitance by a factor of $A$,

$$C_{in2} = A \cdot C_{in1} \text{ and } C_{in3} = A \cdot C_{in2} = A^2 \cdot C_{in1}, \text{ etc.} \tag{11.17}$$

The effective switching resistances are also divided by a factor of $A$, resulting in the same delay for each stage of the buffer,

$$R_{n,p2} = \frac{R_{n,p1}}{A} \text{ and } R_{n,p3} = \frac{R_{n,p2}}{A} = \frac{R_{n,p1}}{A^2} \tag{11.18}$$

In other words, the effective switching resistance of the NMOS in the third inverter is $1/A^2$ smaller than the effective switching resistance of the NMOS in the first inverter.

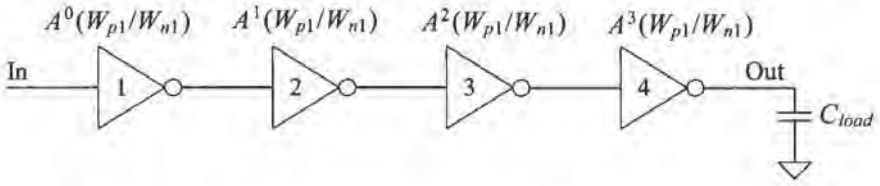



**Figure 11.19** Cascade of inverters used to drive a large load capacitance.

If the load capacitance equals the input capacitance of the last inverter multiplied by $A$ (so that the load capacitance has a value equal to the input capacitance of the next inverter if an additional inverter were used), then

$$\text{Input C of the final inverter} = C_{in1} \cdot A^N = C_{load} \tag{11.19}$$

or

$$A = \left[\frac{C_{load}}{C_{in1}}\right]^{\frac{1}{N}} \tag{11.20}$$

Again, the delays of each stage in the buffer are equal. The total delay of the inverter string is given by

IPR2023-00697
Theta EX2019

$$(t_{PHL}+t_{PLH})_{total}=0.7\cdot\underbrace{(R_{n1}+R_{p1})(C_{out1}+AC_{in1})}_{\text{First-stage delay}}+\underbrace{0.7\cdot\frac{(R_{n1}+R_{p1})}{A}\cdot(AC_{out1}+A^2C_{in1})\ldots}_{\text{Second-stage delay}} \tag{11.21}$$

Because as the inverters are increased in size by $A$, their capacitances, both input and output, increase by $A$, while their resistances decrease by a factor $A$. This equation can be rewritten as

$$(t_{PHL}+t_{PLH})_{total}=0.7\cdot\sum_{k=1}^{N}(R_{n1}+R_{p1})(C_{out1}+AC_{in1})=0.7\cdot N(R_{n1}+R_{p1})(C_{out1}+AC_{in1}) \tag{11.22}$$

or with the help of Eq. (11.20):

$$(t_{PHL}+t_{PLH})_{total}=0.7\cdot N(R_{n1}+R_{p1})\cdot\left(C_{out1}+\left(\frac{C_{load}}{C_{in1}}\right)^{\frac{1}{N}}\cdot C_{in1}\right) \tag{11.23}$$

The minimum delay can be found by taking the derivative of this equation with respect to $N$, setting the result equal to zero, and solving for $N$. Taking the derivative of Eq. (11.23) with respect to $N$ gives

$$0.7\cdot\left((R_{n1}+R_{p1})C_{out1}+(R_{n1}+R_{p1})C_{in1}\left(\left(\frac{C_{load}}{C_{in1}}\right)^{\frac{1}{N}}+N\cdot\left(\frac{C_{load}}{C_{in1}}\right)^{\frac{1}{N}}\frac{\ln(C_{load}/C_{in1})}{-N^2}\right)\right)=0 \tag{11.24}$$

The first term in this equation is the intrinsic delay of the first inverter in our cascade of inverters. *This first stage represents, generally, the on-chip logic gate and* ***is not part*** *of the buffer.* The first inverter is included in the calculations to represent the limited drive of the on-chip logic. If we assume that this delay is small, solving for $N$ gives

$$N=\ln\frac{C_{load}}{C_{in1}} \tag{11.25}$$

Equations (11.25) and (11.20) are used in the buffer design to drive a large capacitance. Note that the larger the first inverter, the fewer the number of inverters needed to drive a given capacitive load.

**Example 11.8**

Estimate $t_{PHL}+t_{PLH}$ for the inverter shown in Fig. 11.11 driving a load capacitance of 20 pF. Design a buffer to drive the load capacitance with a minimum delay.

The total propagation delay of the unbuffered inverter, Fig. 11.11, is given by

$$t_{PHL}+t_{PLH}=0.7\cdot(3.4k+3.4k)\cdot(20\,pF)=95\,ns\,!$$

Designing a buffer begins with determining $C_{in1}$. For our 20/10 inverter in the 50 nm process, the $C_{in1}$ (see Table 10.2) is $\frac{3}{2}(1.25+0.625)\,fF=2.81\,fF$ and $C_{out1}$ is $1.875\,fF$. To determine the number of inverters, we use

$$N=\ln\left(\frac{20\,pF}{2.81\,fF}\right)=8.87\rightarrow 9\text{ stages}$$

IPR2023-00697
Theta EX2019

To maintain the same logic, that is, an inversion of the input signal, we use seven inverters. In practice, the difference in delay between eight and nine inverters is negligible. If we did not want a logic inversion, we would use eight stages. The area factor is then

$$A = \left[\frac{20\,pF}{2.81\,fF}\right]^{\frac{1}{8.87}} = 2.718 = e$$

noting that for the *minimum delay* in all cases the widths are increased by $e$.

The total delay, using Eq. (11.23), is then

$$(t_{PHL} + t_{PLH})_{total} = 0.7 \cdot 9 \cdot (3.4k + 3.4k)(1.875\,fF + 2.718 \cdot 2.81\,fF) = 407\,ps$$

or over 200 times faster. Since the PMOS width is twice the width of the NMOS, the propagation delay times, $t_{PHL}$ and $t_{PLH}$, are equal, or

$$t_{PHL} = t_{PLH} = \frac{(t_{PHL} + t_{PLH})_{total}}{2N} = 22.5\,ps$$

A schematic of the design is shown in Fig. 11.20. ■

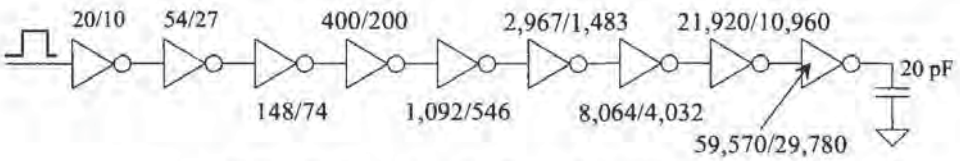



**Figure 11.20** Buffer designed in Ex. 11.8. Attains the minimum delay but the buffer is not practical.

It should be clear that, although this technique results in the least delay in driving the 20 pF load, the MOSFETs needed are very large. In many applications, the very minimum delay through a buffer is not required. The value of $A$ (ideally $e$) can be considerably larger and have little impact on the delay of the buffer (reducing the number of stages and their widths). Consider the following.

**Example 11.9**

Redesign the buffer of Ex. 11.8 with an $A$ of 8. Compare the delay of the modified (practical) buffer to the delay of the ideal buffer calculated in Ex. 11.8.

We can rewrite Eq. (11.20) as

$$N \cdot \ln A = \ln \frac{C_{load}}{C_{in1}}$$

The natural logarithm of 8 is roughly 2 so we can solve for the number of stages using

$$N = \frac{1}{2} \cdot \ln \frac{C_{load}}{C_{in1}} = 4.43$$

IPR2023-00697
Theta EX2019

To maintain the logic inversion, we'll use five stages. The delay is (roughly, because we are using 5 instead of 4.43) calculated as

$$(t_{PHL} + t_{PLH})_{total} = 0.7 \cdot 5 \cdot (3.4k + 3.4k)(1.875\,fF + 8 \cdot 2.81\,fF) = 580\,ps$$

(not too much larger than the 407 ps calculated in Ex. 11.7). The resulting buffer is shown in Fig. 11.21. ■

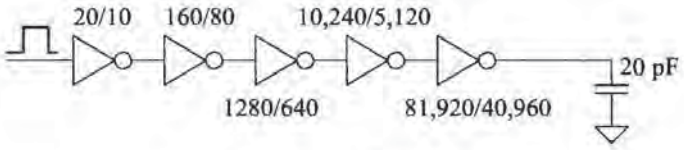



**Figure 11.21** Buffer designed in Ex. 11.9.

*Distributed Drivers*

Consider the driver circuit shown in Fig. 11.22a containing 11 inverters. If all of the inverters shown in the figure are the same size, the delay from the input to the output is

$$t_{PHL} + t_{PLH} = 0.7 \cdot (R_n + R_p)(C_{out} + 10C_{in}) \qquad (11.26)$$

Now consider the circuit shown in Fig. 11.22b with 13 inverters. Again, assuming all of the inverters are the same size, the delay from the input to the output is

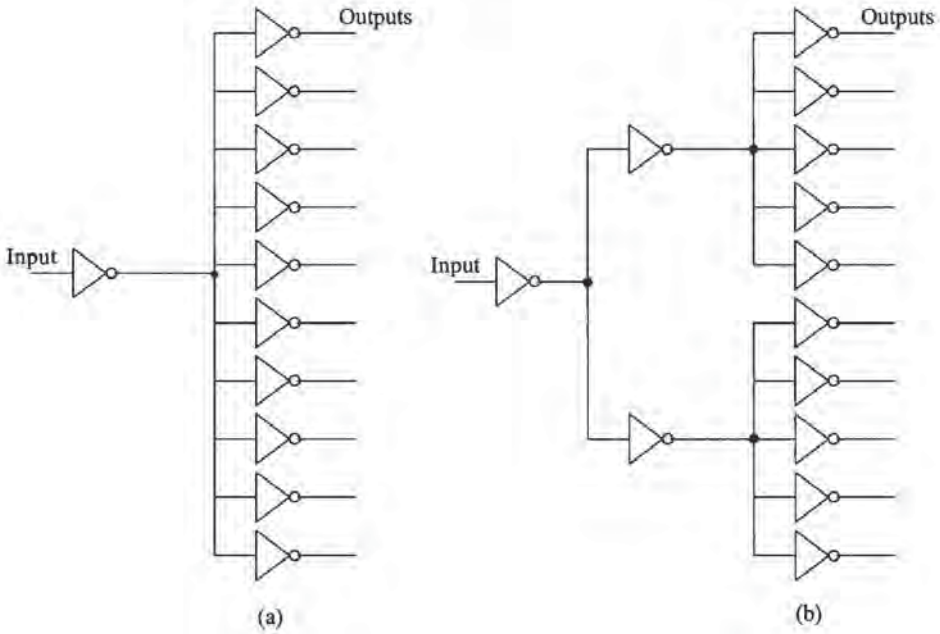



**Figure 11.22** Distributed drivers.

IPR2023-00697
Theta EX2019

$$t_{PHL} + t_{PLH} = 0.7 \cdot (R_n + R_p)[(C_{out} + 2C_{in}) + (C_{out} + 5C_{in})] = 0.7 \cdot (R_n + R_p)[2C_{out} + 7C_{in}] \tag{11.27}$$

which is less delay than the circuit with 11 inverters. Distributing the signal into different paths can reduce the propagation delays. Using the results from the last section, we can get minimum overall delay when the delays through each layer of logic are equal. This occurs when $A = e$ (each inverter drives 2.718 other inverters) and Eq. (11.25) is used to select the (number of) layers of logic. The load capacitance is equal to the number of outputs multiplied by the capacitance on each output. In practice, again as we saw in the last section, the change in delays isn't too significant, as long as one logic gate (one path) isn't loaded too much (compare actual numbers in Eqs. [11.26] and [11.27]).

At this point we can ask the question, "Why not make the first inverter in the circuits of Fig. 11.22 really large (small $R_n$ and $R_p$) so that it has small effective resistances for driving the ten inverters quickly?" The answer is simply that as we increase the size of an inverter, we also increase its input capacitance. In SPICE simulations, we use ideal voltage sources to drive the first gate in our circuit. In practice, this inverter is driven from another gate somewhere on the chip. Increasing the size will slow the propagation delay-time of the gate driving this inverter.

*Driving Long Lines*

Often when designing large systems, a signal may need to be driven across the chip. In some cases, for example, in dynamic random-access memory (DRAM), the signal must be transmitted over a line that has a large parasitic resistance and capacitance. We need to develop a method of determining the delay through this line using hand calculations. This will lend insight to the design and help to determine exactly how to design the driver (or drivers).

Consider the driver circuit shown in Fig. 11.23. The inverter is driving an RC transmission line with resistance/unit length, $r$, capacitance/unit length, $c$, and unit length, $l$. We can estimate the delay from the input to voltage across the capacitor by adding the delays. This is given by

$$t_{PHL} + t_{PLH} = 0.7 \cdot [(R_n + R_p)(C_{out} + c \cdot l + C_{load}) + 2 \cdot (r \cdot l)(C_{load})] + 2 \cdot 0.35 \cdot rcl^2 \tag{11.28}$$

where the first term in this equation is the delay associated with the inverter driving the total capacitance at its output to ground. The second term is the delay associated with driving a capacitive load through the line's resistance, while the last term is an estimate of

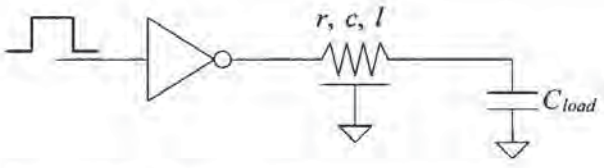



**Figure 11.23** Driving an RC transmission line.

IPR2023-00697
Theta EX2019

the delay through the RC line. The most common method of reducing the delay through the line is to place buffer stages at different locations along the line. This effectively breaks the line up and can lower the overall delay. If $C_{load}$ is a major contributor to the delay, a buffer can be inserted between the *RC* line and $C_{load}$ to reduce the delay.

## 11.5 Other Inverter Configurations

Three other inverter configurations are shown in Fig. 11.24. The inverter shown in Fig. 11.24a is an NMOS-only inverter, useful in avoiding latch-up. There's no PMOS device so there's no parasitic bipolar junction transistors. The inverters shown in Fig. 11.24b and c use a PMOS load, which is, in general, most useful in logic gates with a large number of inputs (more on this in the next chapter). In general, the selection of the MOSFET sizes in (a) and (b) follows the 4-to-1 rule; that is, the resistance ($R_n$ or $R_p$) of the load is made four times larger than the resistance of M1. The resistance of the PMOS in (c) can be made eight times the resistance of the NMOS. Because $R_p > R_n$, the $t_{PLH}$ will always be greater than the $t_{PHL}$. In other words, the switching times will be asymmetric.

For all inverter configurations in Fig. 11.24, a logic 1 input signal results in a DC current flowing in both MOSFETs. Making sure that this DC current isn't too large is an important design concern when sizing the MOSFETs. The output logic low will never reach 0 V in these inverters ($V_{OL}$ doesn't reach ground), and thus the noise margins are poorer than the basic CMOS inverter of Fig. 11.1. The output high level of the inverter of Fig. 11.24c will reach *VDD*, while the other inverter's output high level will be a threshold voltage drop below *VDD*. It might be concluded that the power dissipation of the inverters shown in Fig. 11.24 is greater than the basic CMOS inverter. However, since the input capacitance of these inverters is less than the basic CMOS inverter and the output voltage swing is reduced, the inverter with the greatest power dissipation is determined by the operating frequency. At high operating frequencies, the basic CMOS inverter dissipates the most power. At DC or low frequencies, the inverter configurations seen in Fig. 11.24 dissipate more power.

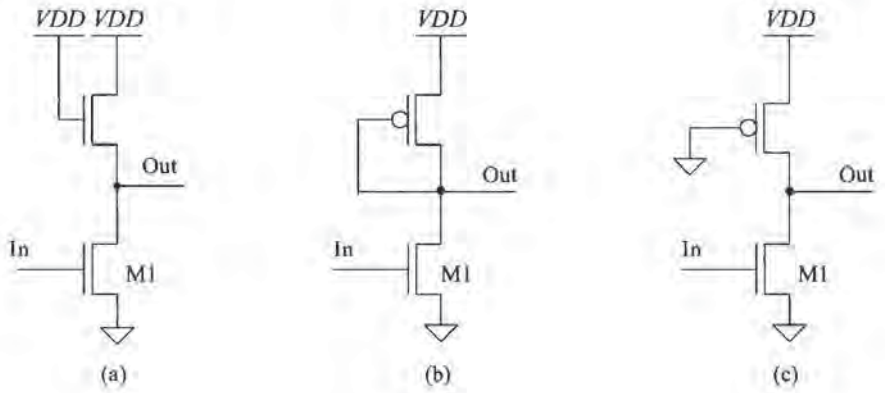



**Figure 11.24** Other inverter configurations.

IPR2023-00697
Theta EX2019

*NMOS-Only Output Drivers*

Because of the susceptibility of the basic CMOS inverter to latch-up, output drivers consisting of only NMOS devices are used. Figure 11.25 shows the basic "NMOS super buffer." When the input signal is low, M1 and M4 are off while M2 and M3 are on. The output is pulled to ground through M2. A high on the input to the buffer causes M1 and M4 to turn on pulling the output to $VDD - V_{THN}$, assuming that the input high-signal amplitude is *VDD*.

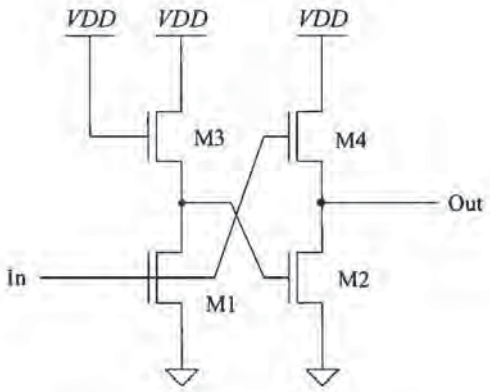



**Figure 11.25** NMOS super buffer.

The reduced output voltage of the NMOS-only output buffer can be improved using the circuit of Fig. 11.26. The inverter driving the gate of M2 uses an on-chip generated DC voltage of *VDD* + *Vadd* (where *Vadd* is large enough to ensure that M2 turns fully on and the output is driven to *VDD*). Thus, the output swings from 0 to *VDD* similar to the CMOS output buffer. Note that with the addition of an enabling logic gate, the gates of M1 and M2 can be held at ground, forcing the output into the high-impedance (Hi-Z) state. (Note: Figure 18.40 shows two dynamic NMOS-only output drivers.)

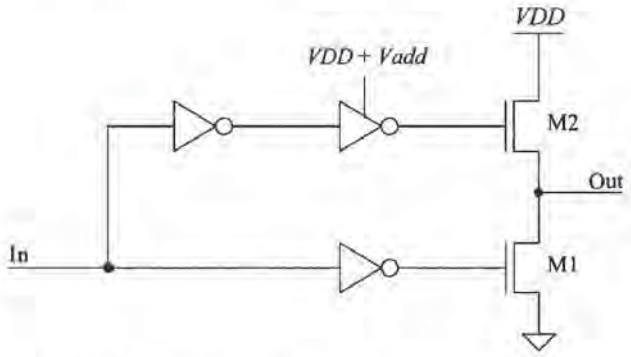



**Figure 11.26** Output buffer using a pumped voltage.

IPR2023-00697
Theta EX2019

*Inverters with Tri-State Outputs*

Two configurations used in the design of an inverter with tri-state outputs are shown in Fig. 11.27. A high on the $S$ input allows the circuit to operate normally, that is, as an inverter. A low on the S input forces the output into the Hi-Z, or high-impedance state. These circuits are useful when data is shared on a communication bus. The logic symbol for the tri-state inverter is shown in Fig. 11.27c. Note that the circuit in (a) dissipates power even when the the select signal, $S$ , is a low, while the circuit in (b) does not. However, the circuit in (a) has faster switching times because the effective resistance seen at the output to ground or *VDD* is lower (the NMOS and PMOS devices are in parallel).

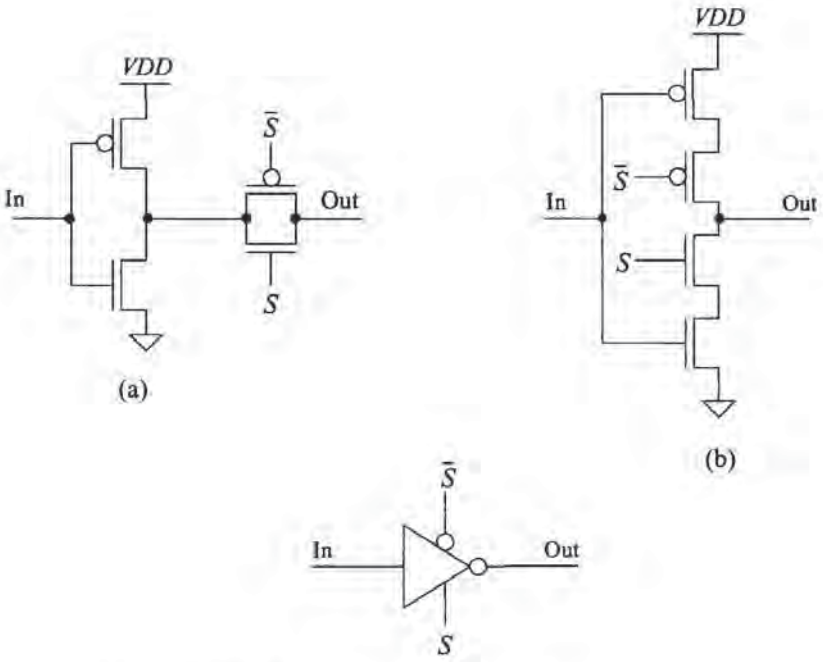



**Figure 11.27** Circuits and logic symbol for the tri-state inverter.

*Additional Examples*

Additional delay calculations using inverters (and other CMOS circuit building blocks) can be found in Sec. 13.4.

## ADDITIONAL READING

[1] K. Bernstein, K. M. Carrig, C. M. Durham, P. R. Hansen, D. Hogenmiller, E. J. Nowak, and N. J. Rohrer, *High Speed CMOS Design Styles*, Springer, 1999. ISBN 978-0792382201.

[2] J. P. Uyemura, *Introduction to VLSI Circuits and Systems*, John Wiley and Sons Publishers, 2002. ISBN 0-471-12704-3.

IPR2023-00697
Theta EX2019

[3] M. I. Elmasry, *Digital MOS Integrated Circuits II,* IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.

[4] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI-Design Techniques for Analog and Digital Circuits,* McGraw-Hill Publishing Co., 1990. ISBN 0-07-023253-9.

**PROBLEMS**

Use the 50 nm CMOS process for the following problems unless otherwise stated.

**11.1** Estimate the noise margins for the inverters used to generate Fig. 11.4.

**11.2** Design and simulate the DC characteristics of an inverter with $V_{SP}$ approximately equal to $V_{THN}$. Estimate the resulting noise margins for the design.

**11.3** Show that the switching point of three inverters in series is dominated by the $V_{SP}$ of the first inverter.

**11.4** Repeat Ex. 11.6 using a PMOS device with a width of 10.

**11.5** Repeat Ex. 11.6 using the long-channel process with a 30/10 inverter.

**11.6** Estimate the oscillation frequency of a 11-stage ring oscillator using inverters 30/10 inverters in the long-channel CMOS process. Compare your hand calculations to simulation results.

**11.7** Using the long-channel process, design a buffer with minimum delay ($A$ = 2.718) to insert between a 30/10 inverter and a 50 pF load capacitance. Simulate the operation of the design.

**11.8** Repeat Problem 11.7 using an area factor, $A$, of 8.

**11.9** Derive an equation for the switching point voltage, similar to the derivation of Eq. (11.4), for the NMOS inverter seen in Fig. 11.24a.

**11.10** Repeat Problem 11.9 for the inverter in Fig. 11.24c. Note that the PMOS transistor is operating in the triode region when the input/output are at $V_{SP}$.

IPR2023-00697
Theta EX2019

Chapter
12

# Static Logic Gates

In this chapter we discuss the DC characteristics, dynamic behavior, and layout of CMOS static logic gates. Static logic means that the output of the gate is always a logical function of the inputs and always available on the outputs of the gate regardless of time. We begin with the NAND and NOR gates.

## 12.1 DC Characteristics of the NAND and NOR Gates

The two basic input NAND and NOR gates are shown in Fig. 12.1. Before we get into the operation, notice that each input into the gate is connected to both a PMOS and an NMOS device similar to the inverter of the last chapter. We will make use of the results of Ch. 11 to explain the operation of these gates.

### 12.1.1 DC Characteristics of the NAND Gate

The NAND gate of Fig. 12.1a requires both inputs to be high before the output switches low. Let's begin our analysis by determining the voltage transfer curve (VTC) of a NAND gate with PMOS devices that have the same widths, $W_p$, and lengths, $L_p$, and NMOS devices with equal widths of $W_n$ and lengths of $L_n$. If both inputs of the gate are tied together, then the gate behaves like an inverter.

To determine the gate switching point voltage, $V_{SP}$, we must remember that two MOSFETs in parallel behave like a single MOSFET with a width equal to the sum of the individual widths. For the two parallel PMOS devices in Fig. 12.1a, we can write

$$W_3 + W_4 = 2W_p \tag{12.1}$$

again assuming that all PMOS devices are of the same size. The transconductance parameters can also be combined into the transconductance parameter of a single MOSFET, or

$$\beta_3 + \beta_4 = 2\beta_p \tag{12.2}$$

The two NMOS devices in series (with their gates tied together) behave like a single MOSFET with a channel length equal to the sum of the individual MOSFET lengths. We can write for the NMOS devices

IPR2023-00697
Theta EX2019

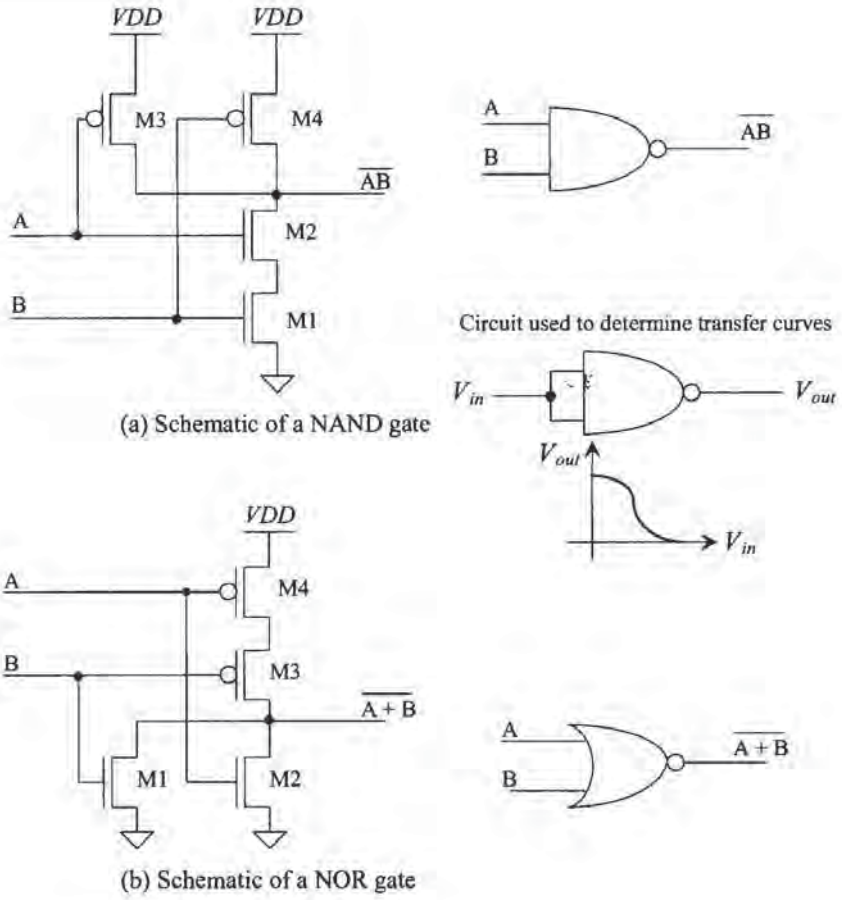



**Figure 12.1** NAND and NOR gate circuits and logic symbols.

$$L_1 + L_2 = 2L_n \tag{12.3}$$

and the transconductance of the single MOSFET is given by

$$\beta_1 + \beta_2 = \frac{\beta_n}{2} \tag{12.4}$$

If we model the NAND gate with both inputs tied together as an inverter with an NMOS device having a width of $W_n$ and length $2L_n$ and a PMOS device with a width of $2W_p$ and length $L_p$, then we can write the transconductance ratio as

$$\text{Transconductance ratio of NAND gate} = \frac{\beta_n}{4\beta_p} \tag{12.5}$$

The $V_{SP}$, with the help of Eq. (11.4), of the two-input NAND gate is then given by

IPR2023-00697
Theta EX2019

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{4\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{4\beta_p}}} \tag{12.6}$$

or in general for an n-input NAND gate (see Fig. 12.2), we get

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{N^2 \cdot \beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{N^2 \cdot \beta_p}}} \tag{12.7}$$

These equations are derived under the assumption that all inputs are tied together. If, for example, only one input is switching, the $V_{SP}$ will vary from what is calculated using Eq. (12.7) (assuming the single input switching does indeed cause the gate's output to switch). This equation is used to show why NAND gates are preferred in CMOS design. If equal- sized NMOS and PMOS devices are used, then, since the mobility of the hole is less than the mobility of the electron, $\beta_n > \beta_p$. Using NMOS devices in series and PMOS in parallel (as in the NAND gate) makes it easier to design a logic gate with the ideal switching point voltage of *VDD*/2.

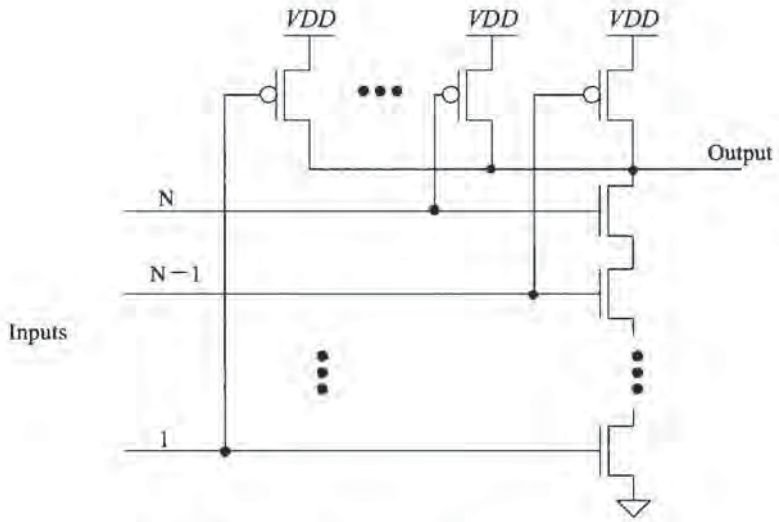



**Figure 12.2** Schematic of an n-input NAND gate.

**Example 12.1**
Determine $V_{SP}$ by hand calculations and compare to a SPICE simulation for a three-input NAND gate using 10/1 devices in the long-channel CMOS process used in this book, see Table 6.2. Compare the hand calculations to simulation results.

IPR2023-00697
Theta EX2019

The switching point voltage is determined by calculating the transconductance ratio of the gate, or

$$\sqrt{\frac{\beta_n}{N^2\beta_p}} = \sqrt{\frac{\frac{120\ \mu A/V^2 \cdot 10}{1}}{9 \cdot \frac{40\ \mu A/V^2 \cdot 10}{1}}} = 0.58$$

and then using Eq. (12.7),

$$V_{SP} = \frac{0.58 \cdot (0.8) + (5 - 0.9)}{1 + 0.58} = 2.9\ V$$

The SPICE simulation results are shown in Fig. 12.3. The simulation also gives a $V_{SP}$ of approximately 2.9 V. ■

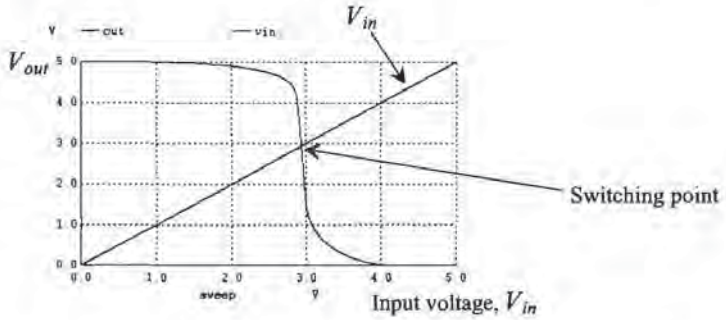



**Figure 12.3** VTCs of the three-input minimum-size (using 10/1 MOSFETs) NAND gate.

### 12.1.2 DC Characteristics of the NOR gate

Following a similar analysis for the n-input NOR gate (see Fig. 12.4) gives a switching point voltage of

$$V_{SP} = \frac{\sqrt{\frac{N^2 \cdot \beta_n}{\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{N^2 \cdot \beta_n}{\beta_p}}} \qquad (12.8)$$

**Example 12.2**

Compare the switching point voltage of a three-input NOR gate made from minimum-size MOSFETs to that of the three-input NAND gate of Ex. 12.1. Comment on which gate's $V_{SP}$ is closer to ideal, that is, $V_{SP} = VDD/2$.

The transconductance ratio is calculated as

$$\sqrt{N^2 \cdot \frac{\beta_n}{\beta_p}} = \sqrt{9 \cdot \frac{\frac{120\ \mu A/V^2 \cdot 10}{1}}{\frac{40\ \mu A/V^2 \cdot 10}{1}}} = 5.2$$

The $V_{SP}$ of the minimum-size three-input NOR gate is 1.33 V, while the $V_{SP}$ of the minimum-size three-input NAND gate was calculated to be 2.9 V. For an ideal

IPR2023-00697
Theta EX2019

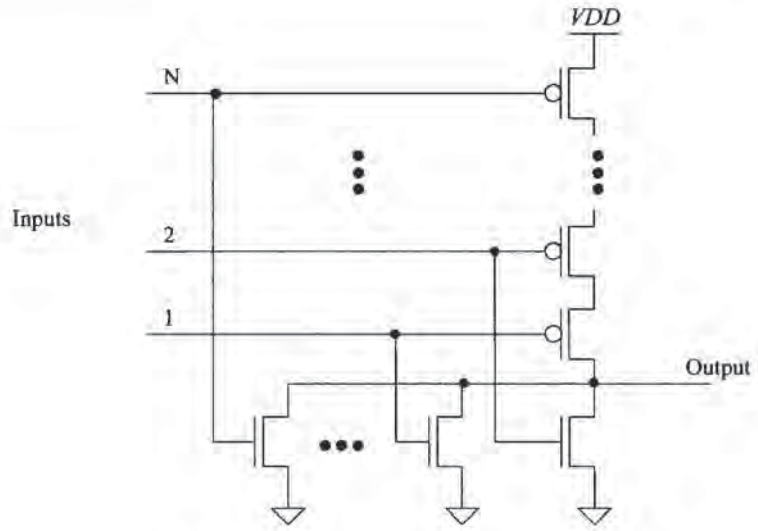



**Figure 12.4** Schematic of an n-input NOR gate.

gate, $V_{SP} = 2.5$ V, so that the NAND gate is closer to ideal than the NOR gate. In CMOS digital design, the NAND gate is used most often. This is due to the DC characteristics, better noise margins, and the dynamic characteristics. We will also see shortly that the NAND gate has better transient characteristics than the NOR gate. ■

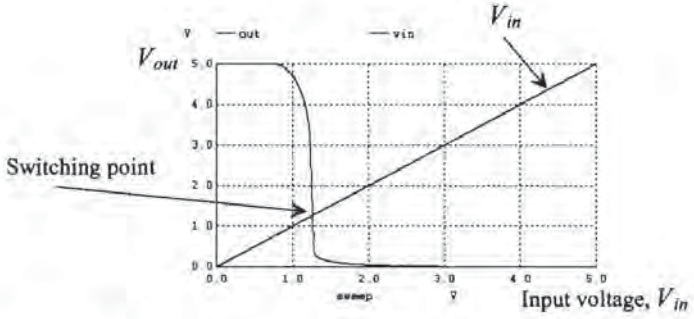



**Figure 12.5** VTCs of the three-input minimum-size (using 10/1 MOSFETs) NOR gate.

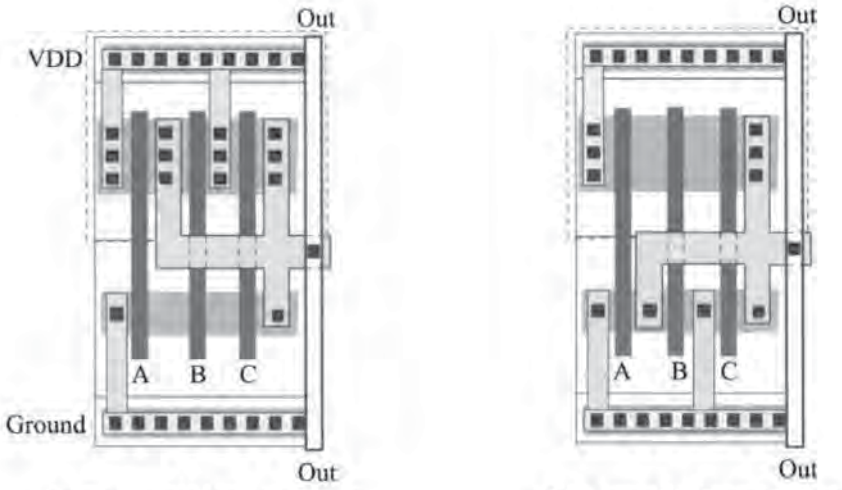
*A Practical Note Concerning $V_{SP}$ and Pass Gates*

Reviewing Fig. 10.19, we see that passing a logic signal through a pass gate (PG) can result in a reduction in the logic signal's amplitude. Using an NMOS PG, for example, results in an output signal swing from ground to $VDD - V_{THN}$. If this logic signal is connected to an inverter or a logic gate, we will want to set the $V_{SP}$ to maximize the noise margins. Using the NMOS PG, our inverter/gate would have a $V_{SP} = (VDD - V_{THN})/2$.

IPR2023-00697
Theta EX2019

## 12.2 Layout of the NAND and NOR Gates

Layout of the three-input minimum-size NOR and NAND gates is shown in Fig. 12.6, using the standard-cell frame. MOSFETs in series, for example, the NMOS devices in the NAND gate, are laid out using a single-drain and a single-source implant area. The active area between the gate poly is shared between two devices. This has the effect of reducing the parasitic drain/source implant capacitances. MOSFETs in parallel, for example, the NMOS devices in the NOR gate, can share a drain area or a source area. The inputs of the gates are shown on the poly layer while the outputs of the two logic gates are on metal2. To make the inputs easy to connect to we would route them up to metal2 like the outputs. Note how metal1 is used inside the cell and horizontally to connect power and ground to the cells, while metal2 is used for vertical running wires (inputs and outputs).




(a) 3-input NAND gate layout (b) 3-input NOR gate layout

**Figure 12.6** Layouts of NAND (a) and NOR (b) gates.

## 12.3 Switching Characteristics

In this section we discuss the switching characteristics of static logic gates.

*Parallel Connection of MOSFETs*

Consider the parallel connection of identical MOSFETs shown in Fig. 12.7 *with their* gates tied together. From the equivalent digital models, also shown, we can determine the propagation delay associated with this parallel connection of N MOSFETs as

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N} \cdot (N \cdot C_{oxp}) = 0.7 \cdot R_p C_{oxp} \qquad (12.9)$$

where $C_{oxp} = C'_{ox} \cdot W \cdot L \cdot (scale)^2$. With an external load capacitance, the low-to-high delay-time becomes

IPR2023-00697
Theta EX2019

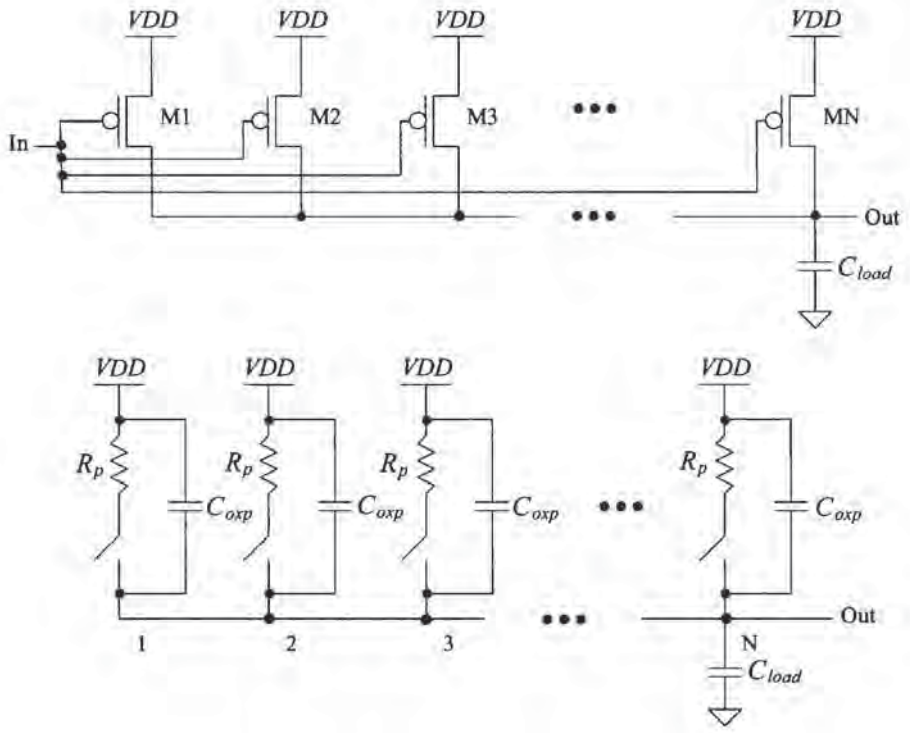



**Figure 12.7** Parallel connection of MOSFETs and equivalent digital model.

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N} \cdot (N \cdot C_{oxp} + C_{load}) \tag{12.10}$$

This again assumes that the MOSFET's gates are tied together (all are switching at the same time). For NMOS devices in parallel, a similar analysis yields

$$t_{PHL} = 0.7 \cdot \frac{R_n}{N} \cdot (N \cdot C_{oxn} + C_{load}) \tag{12.11}$$

The load capacitance, $C_{load}$, consists of all capacitances on the output node except the output capacitances of the MOSFETs in parallel.

*Series Connection of MOSFETs*

Consider the series connection of identical NMOS devices shown in Fig. 12.8. We can *estimate* the intrinsic switching time of series-connected MOSFETs by

$$t_{PHL} = 0.35 \cdot R_n C_{oxn} \cdot N^2 \tag{12.12}$$

as discussed back in Sec. 10.2.2. With an external load capacitance, the high-to-low delay-time becomes

$$t_{PHL} = 0.35 \cdot R_n C_{oxn} \cdot N^2 + 0.7 \cdot N \cdot R_n \cdot C_{load} \tag{12.13}$$

IPR2023-00697
Theta EX2019

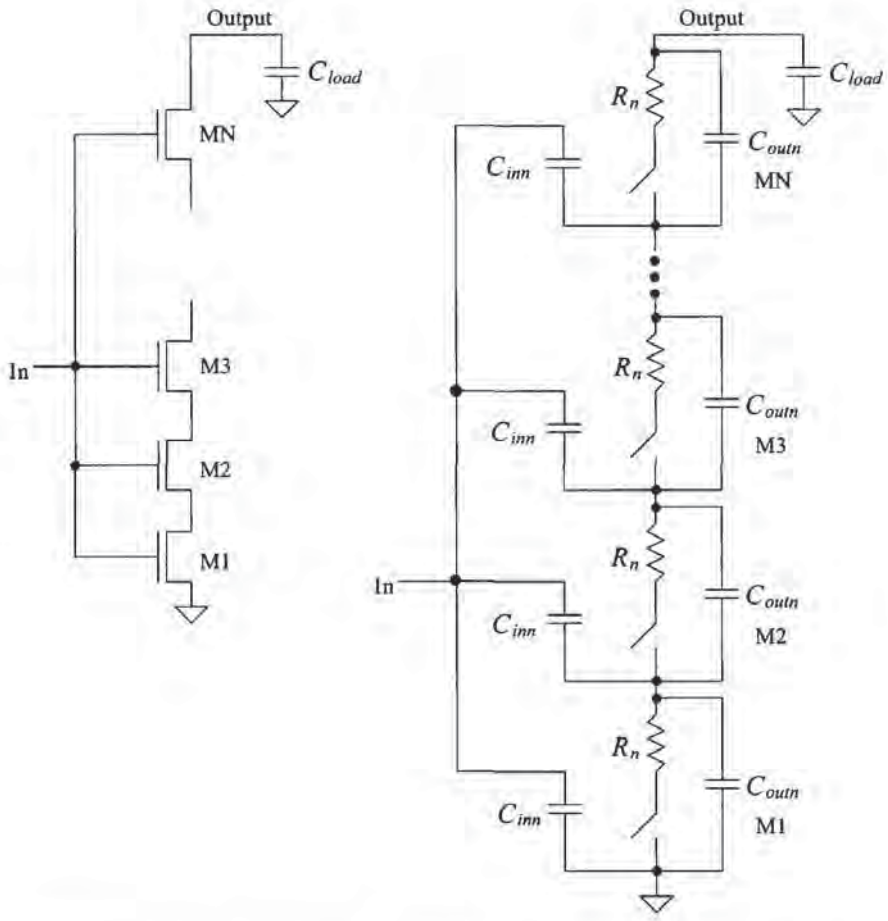



**Figure 12.8** Series connection of MOSFETs and equivalent digital model.

For PMOS devices in series, a similar analysis yields

$$t_{PLH} = 0.35 \cdot R_p C_{oxp} \cdot N^2 + 0.7 \cdot N \cdot R_p \cdot C_{load} \tag{12.14}$$

These equations are approximations for the propagation delays which give results usually to within a factor of two of the measurements.

### 12.3.1 NAND Gate

Consider the n-input NAND gate of Fig. 12.9 driving a capacitive load $C_{load}$. The low-to-high propagation time, using Eq. (12.10), is

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N}\left(N \cdot C_{outp} + \frac{C_{outn}}{N} + C_{load}\right) \tag{12.15}$$

where here $C_{load}$ represents the capacitance external to the gate, whereas in Eq. (12.10) $C_{load}$ represented the capacitance external to the parallel PMOS devices. If the load

IPR2023-00697
Theta EX2019

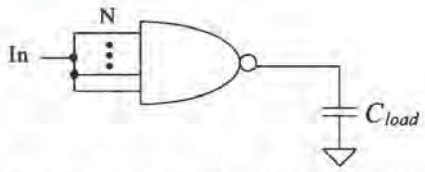



**Figure 12.9** An n-input NAND gate driving a load capacitance.

capacitance is much greater than the output capacitance of the NAND gate, the low-to-high propagation time can be estimated by

$$t_{PLH} \approx 0.7 \cdot \frac{R_p}{N} \cdot C_{load} \tag{12.16}$$

The high-to-low propagation time, using Eq. (12.13), is given by

$$t_{PHL} = 0.7 \cdot N \cdot R_n \left[ N \cdot C_{outp} + \frac{C_{outn}}{N} + C_{load} \right] + 0.35 \cdot R_n C_{oxn} \cdot N^2 \tag{12.17}$$

If $C_{load}$ is much larger than the output capacitance of the NAND gate, then

$$t_{PHL} \approx 0.7 \cdot N \cdot R_n \cdot C_{load} \tag{12.18}$$

**Example 12.3**

Estimate the intrinsic propagation delays, $t_{PHL} + t_{PLH}$, of a three-input NAND gate made using 10/1 NMOS and 20/1 PMOS in the short-channel process. Estimate and simulate the delay when the gate is driving a load capacitance of 50 fF. Assume that the inputs are tied together.

Using the data in Table 10.2 and Eqs. (12.15) and (12.17),

$$t_{PLH} = 0.7 \cdot \frac{3.4k}{3} \cdot \left( 3 \cdot 1.25\,fF + \frac{0.625\,fF}{3} + 50\,fF \right) = 43\,ps$$

and

$$t_{PHL} = 0.7 \cdot 3 \cdot 3.4k \cdot \left[ 3 \cdot 1.25\,fF + \frac{0.625\,fF}{3} + 50\,fF \right] + 0.35 \cdot 3.4k \cdot 0.625\,fF \cdot 9 = 393\,ps$$

The simulation results are seen in Fig. 12.10. ■

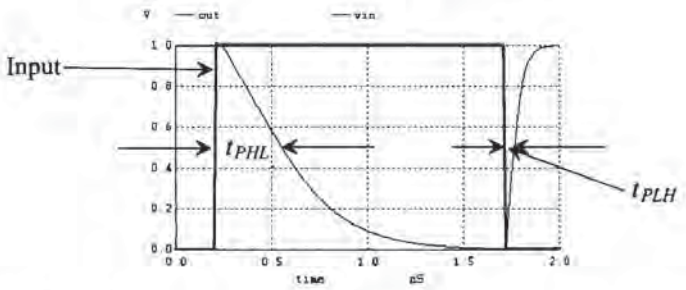



**Figure 12.10** Simulating the operation of a 3-input NAND gate in 50 nm CMOS driving a 50 fF load capacitance.

IPR2023-00697
Theta EX2019

*Quick Estimate of Delays*

The delay equations derived in this section are useful in understanding the limitations on the number of MOSFETs used in a NAND gate for high-speed design. Notice, in Ex. 12.3, how the load capacitance term dominates the gate's delay. A more useful, though not as precise, method of determining delays can be found by considering the fact that whenever the output changes from *VDD* to ground the discharge path is through $N$ resistors of value $R_n$. This is true if all or only one of the inputs to the NAND gate changes, causing the output to change. Under these circumstances, Eq. (12.18) predicts the high-to-low delay-time, or for *series* connection of $N$ NMOS devices as,

$$t_{PHL} \approx 0.7 \cdot N \cdot R_n \cdot C_{load} \tag{12.19}$$

The case when the output of the NAND gate changes from a low to a high is somewhat different than the high-to-low case. Referring to Fig. 12.7, we see that if one of the MOSFETs turns on, it can pull the output to *VDD* independent of the number of MOSFETs in parallel. Under these circumstances Eq. (12.16) can be used with $N = 1$ to predict the low-to-high delay-time

$$t_{PLH} \approx 0.7 \cdot R_p \cdot C_{load} \tag{12.20}$$

We will try to use Eqs. (12.19) and (12.20) as much as possible because of their simplicity. The further simplified digital models of MOSFETs are shown in Fig. 12.11. (Input capacitance is not shown.)

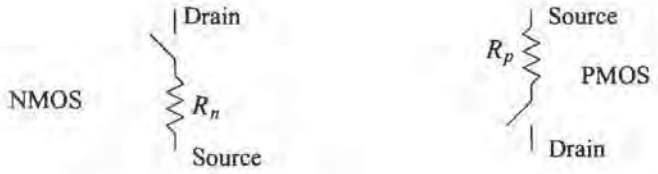



**Figure 12.11** Further simplification of digital models not showing input capacitance.

**Example 12.4**

Repeat Ex. 12.3 using Eqs. (12.19) and (12.20) with only one input switching. Compare the results to simulations.

Using Eq. (12.19), we get

$$t_{PHL} = 0.7 \cdot 3 \cdot 3.4k \cdot 50\,fF = 357\,ps$$

and

$$t_{PLH} = 0.7 \cdot 3.4k \cdot 50\,fF = 119\,ps$$

Figure 12.12 shows the simulation results. The delay through the NMOS devices doesn't change much from what was calculated in Ex. 12.3 since the load capacitance dominates the delay. However, for the PMOS turning on, the delay is considerably longer since only a single PMOS device is switching pulling the output high. This situation (one input changing) is the more practical case. ■

IPR2023-00697
Theta EX2019

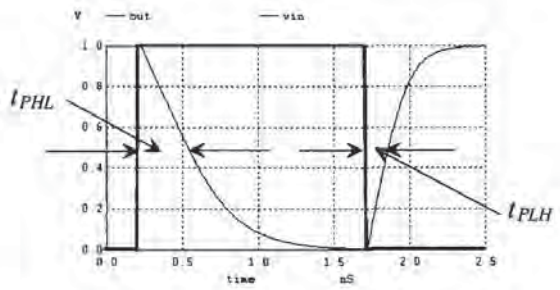



**Figure 12.12** Switching delays in a 3-input NAND gate with only one changing states and driving a 50 fF load capacitance.

### 12.3.2 Number of Inputs

As the number of inputs, $N$, to a static NAND (or NOR) gate increases, the scheme shown in Fig. 12.2 (Fig. 12.4) becomes difficult to realize. Consider a NOR gate with 100 inputs. This gate requires PMOS devices in series and a total of 200 MOSFETs ($2N$ MOSFETs). The delay associated with the series PMOS devices charging a load capacitance is too long for most practical situations.

Now consider the schematic of an $N$ input NOR gate shown in Fig. 12.13, which uses $N + 1$ MOSFETs. If any input to the NOR gate is high, the output is pulled low through the corresponding NMOS device to a voltage, when designed properly, well below $V_{THN}$. If all inputs are low, then all NMOS are off and the PMOS pulls the output high (to $VDD$). A simple (long-channel) analysis of the output low voltage, $V_{OL}$, with *one* input at $VDD$ yields

$$\frac{\beta_p}{2}(VDD - V_{THP})^2 = \beta_n\left[(VDD - V_{THN})V_{OL} - \frac{V_{OL}^2}{2}\right] \qquad (12.21)$$

The drawback of using this topology is the long pull-up time ($t_{PLH}$) resulting from the large output capacitance of the parallel NMOS (and the load capacitance) together with the large resistance of the long length pull-up device.

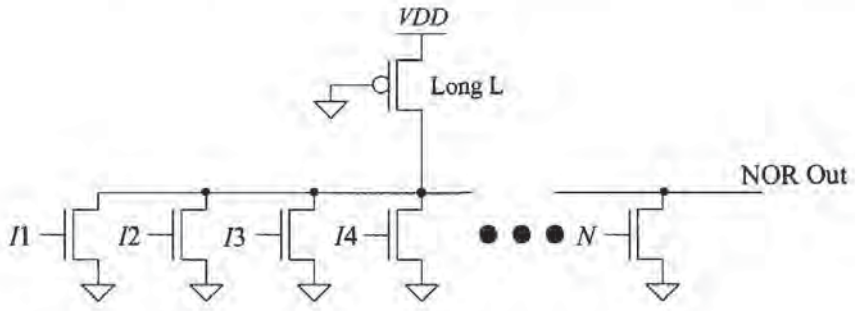



**Figure 12.13** NOR configuration used for a large number of inputs.

IPR2023-00697
Theta EX2019

## 12.4 Complex CMOS Logic Gates

Implementing complex logic functions in CMOS requires the basic building blocks shown in Fig. 12.14. We have already used the circuits to implement NAND and NOR gates. In general, any And-Or-Invert (AOI) logic function can be implemented using these techniques. A major benefit of AOI logic is that for a relatively complex logic function the delay can be significantly lower than a logic gate implementation. Consider the following example.

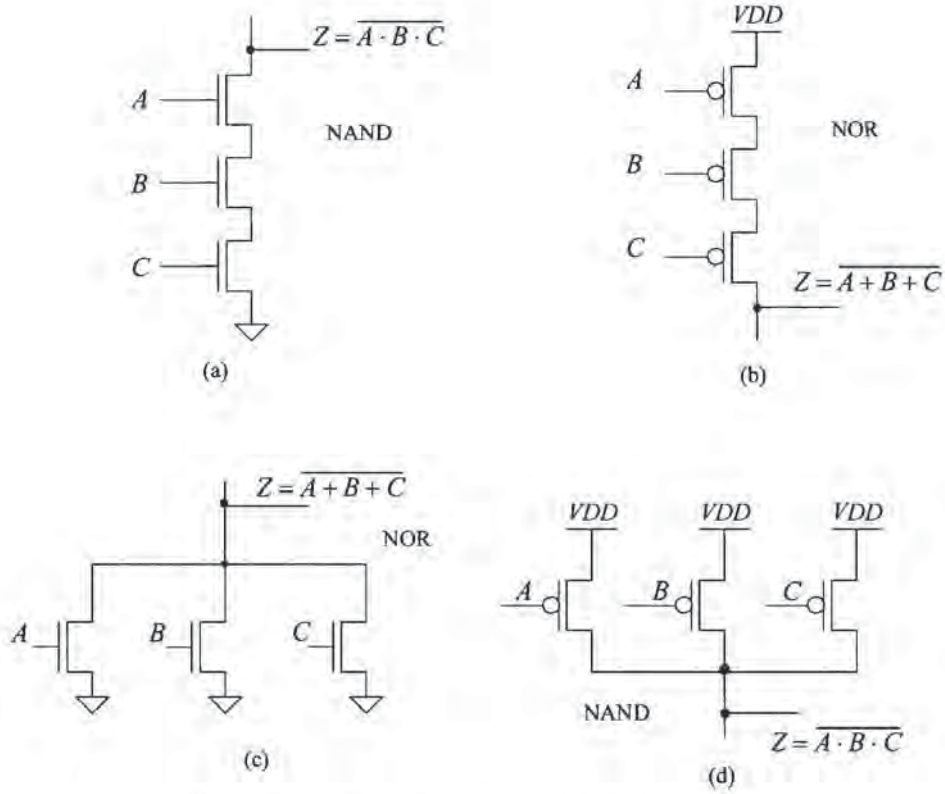



**Figure 12.14** Logic implementation in CMOS.

**Example 12.5**

Using AOI logic, implement the following logic functions:

$$Z = \overline{A} + BC \quad \text{and} \quad Z = A + \overline{B}C + CD$$

The implementation of the first function is shown in Fig. 12.15a. Notice that the PMOS configuration is complementary with the NMOS configuration. The function we obtain is the complement of the desired function, and, therefore, an inverter is used to obtain $Z$. Using an inverter is, in general, undesirable if both

IPR2023-00697
Theta EX2019

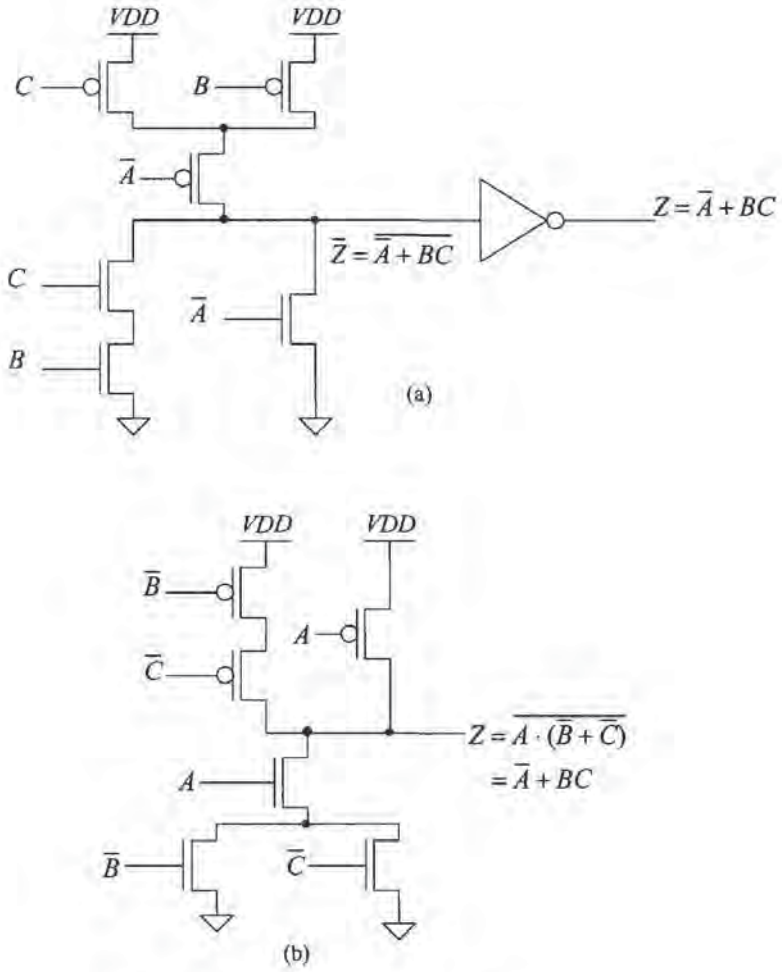



**Figure 12.15** First logic gate of Ex. 12.5.

true and complements of the input variables are available. Applying Boolean algebra to the logic function, we obtain

$$Z = \overline{A} + BC \Rightarrow \overline{Z} = \overline{\overline{A} + BC} = A \cdot (\overline{B} + \overline{C}) \Rightarrow Z = \overline{A \cdot (\overline{B} + \overline{C})}$$

The AOI implementation of the result is shown in Fig. 12.15b. Logically, the circuits of Figs. 12.15a and b are equivalent. However, the circuit of Fig. 12.15b is simpler and thus more desirable. **Note** that to reduce the output capacitance and thus decrease the switching times, the parallel combination of NMOS devices is placed at the bottom of the logic block.

The second logic function is given by

$$Z = A + \overline{B}C + CD = A + C(\overline{B} + D) \Rightarrow \overline{Z} = \overline{A + C(\overline{B} + D)} = \overline{A} \cdot (\overline{C} + B\overline{D})$$

IPR2023-00697
Theta EX2019

or

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

The logic implementation is given in Fig. 12.16. ■

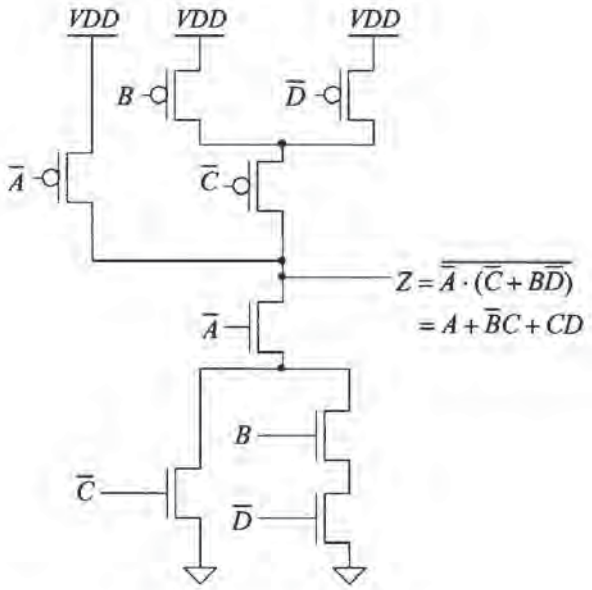
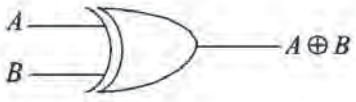



**Figure 12.16** Second logic gate of Ex. 12.5.

**Example 12.6**
Using AOI logic, implement an exclusive OR gate (XOR).

The logic symbol and truth table for an XOR gate are shown in Fig. 12.17. From the truth table, the logic function for the XOR gate is given by

$$Z = A \oplus B = (A + B) \cdot (\overline{A} + \overline{B}) \qquad (12.22)$$




| $A$ | $B$ | $A \oplus B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Figure 12.17** Exclusive OR gate.

IPR2023-00697
Theta EX2019

or

$$\overline{Z} = \overline{A \oplus B} = \overline{(A+B)\cdot(\overline{A}+\overline{B})} = \overline{A}\cdot\overline{B} + A\cdot B$$

and finally

$$Z = \overline{\overline{A}\cdot\overline{B} + A\cdot B} = A \oplus B \tag{12.23}$$

The CMOS AOI implementation of an XOR gate is shown in Fig. 12.18. ■

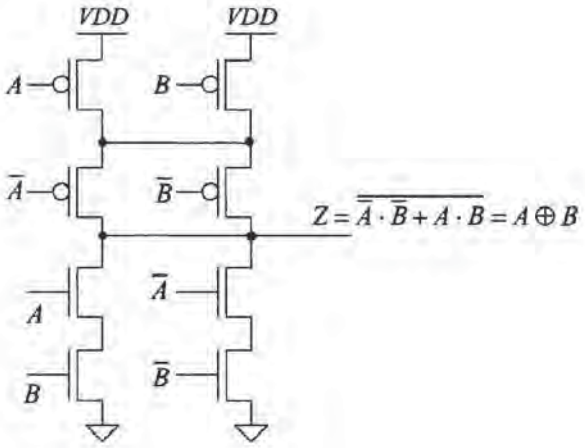
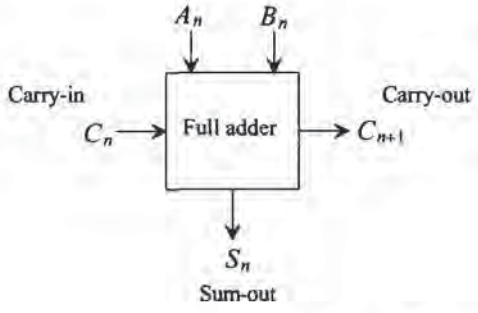



**Figure 12.18** CMOS AOI XOR gate.

**Example 12.7**
Design a CMOS full adder using CMOS AOI logic.

The logic symbol and truth table for a full adder circuit are shown in Fig. 12.19. The logic functions for the sum and carry outputs can be written as

$$S_n = A_n \oplus B_n \oplus C_n$$




| $A_n$ | $B_n$ | $C_n$ | $S_n$ | $C_{n+1}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Figure 12.19** Full adder.

IPR2023-00697
Theta EX2019

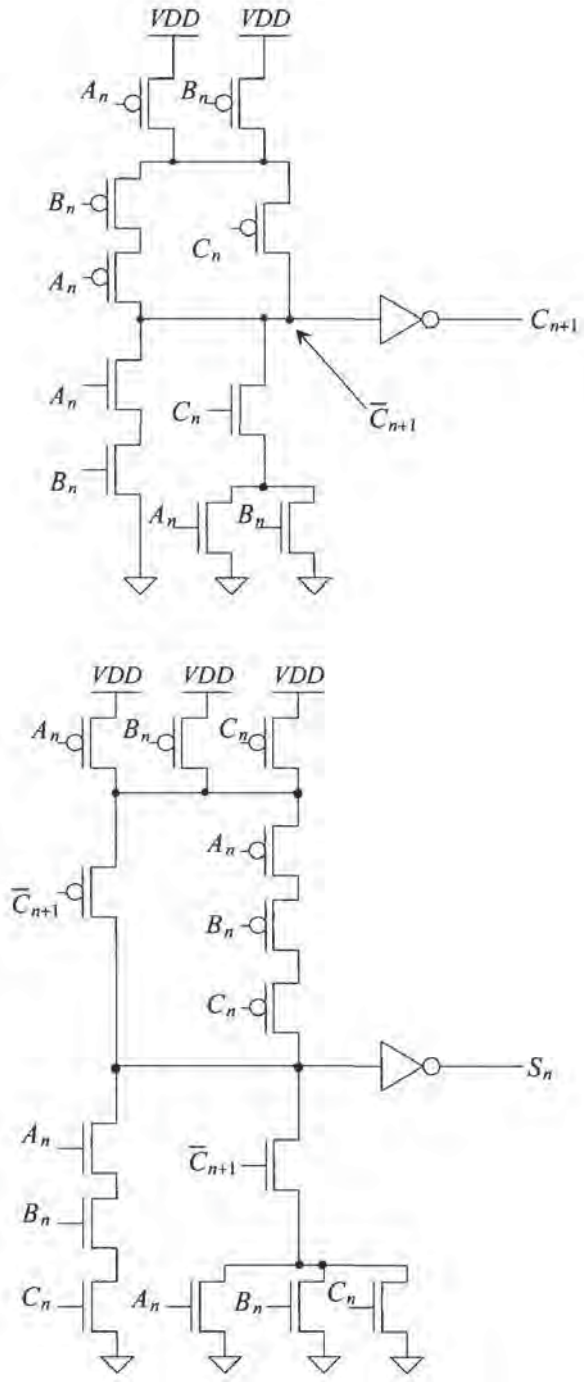



**Figure 12.20** AOI implementation of a full adder.

IPR2023-00697
Theta EX2019

and

$$C_{n+1} = A_n \cdot B_n + C_n(A_n + B_n)$$

The logic expression for the sum can be rewritten as a sum of products

$$S_n = \overline{A}_n\overline{B}_nC_n + \overline{A}_nB_n\overline{C}_n + A_n\overline{B}_n\overline{C}_n + A_nB_nC_n$$

or since

$$\overline{C}_{n+1} = \left(\overline{A}_n + \overline{B}_n\right) \cdot \left(\overline{C}_n + \overline{A}_n \cdot \overline{B}_n\right)$$

the sum of products can be rewritten as

$$S_n = (A_n + B_n + C_n)\overline{C}_{n+1} + A_nB_nC_n$$

The AOI implementation of the full adder is shown in Fig. 12.20. ■

*Cascode Voltage Switch Logic*

Cascode voltage switch logic (CVSL) or differential cascode voltage switch logic (DVSL) is a differential output logic that uses positive feedback in the load of the logic gate to speed up the switching times (in some cases). Figure 12.21 shows the basic idea. A PMOS gate cross-connected load is used instead of PMOS switches, as in the AOI logic, to pull the output high. Consider the implementation of $Z = \overline{A} + BC$. (This logic function was implemented in AOI in Fig. 12.15.) Figure 12.22 shows how NMOS devices can be used to implement $Z$ and $\overline{Z}$. The concern with this implementation is the contention current. When one branch of the NMOS starts to turn on, a significant current can flow through the "on" PMOS device. The amount of current flowing in the conducting PMOS load device(s) can be reduced by increasing their lengths. However, this has the unwanted side effects of lengthening the delay times.

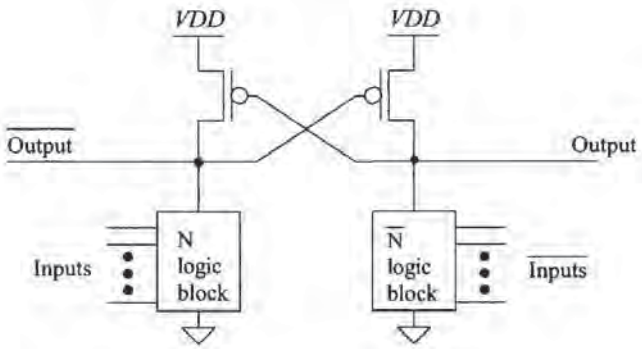



**Figure 12.21** CVSL block diagram.

As another example, Fig. 12.23a shows the implementation of a CVSL two-input XOR/XNOR gate, while Fig. 12.23b shows a CVSL three-input XOR/XNOR gate useful in adder design.

IPR2023-00697
Theta EX2019

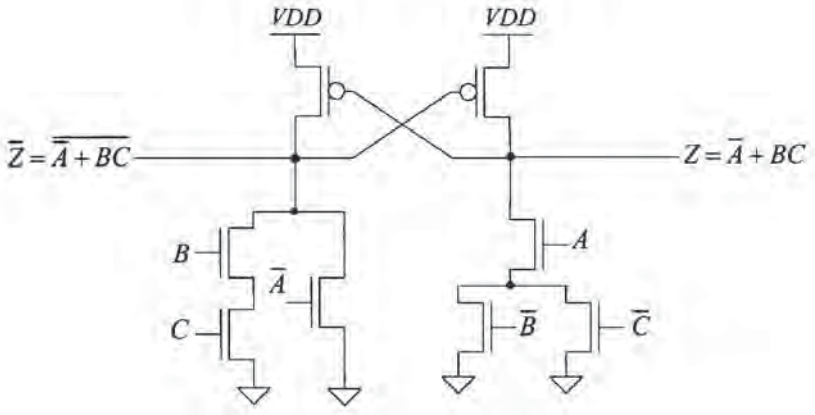



**Figure 12.22** CVSL logic gate.

*Differential Split-Level Logic*

Differential split-level logic (DSL logic) is a scheme wherein the load is used to reduce output voltage swing and thus lower gate delays (at the cost of smaller noise margins). The basic idea is shown in Fig. 12.24. The reference voltage $V_{ref}$ on the gates of M1 and M2 is set to $VDD/2 + V_{THN}$. The sources of M1 and M2 are then at a maximum voltage of $VDD/2$. This has the effect of limiting the output voltage swing to a maximum of $VDD$ and a minimum of $VDD/2$. The main drawback of this logic implementation is the increased power dissipation resulting from the continuous power draw through the output leg at a voltage of $VDD/2$. The output leg at $VDD$ draws no DC power.

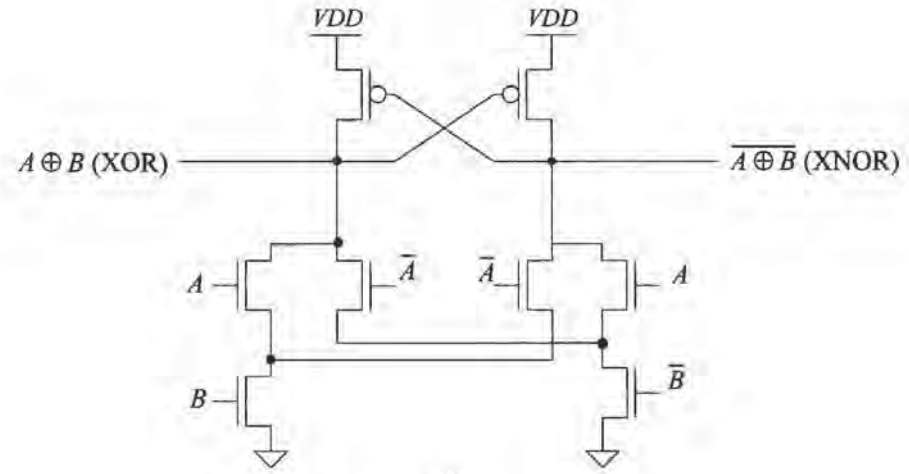
*Tri-State Outputs*

A final example of a static logic gate, a tri-state buffer, is shown in Fig. 12.25. When the *Enable* input is high, the NAND and NOR gates invert and pass $A$ ($VDD$ or ground) to the gates of M1 and M2. Under these circumstances, M1 and M2 behave as an inverter. The combination of M1 and M2 with the inversion NAND/NOR gate causes the output to be the same polarity as $A$. When *Enable* is low, the gate of M1 is held at ground and the gate of M2 is held at $VDD$. This turns both M1 and M2 off. Under these circumstances, the output is said to be in the high-impedance or Hi-Z state. This circuit is preferable to the inverter circuits of Fig. 11.27 because only one switch is in series with the output to $VDD$ or ground. An inverting buffer configuration is shown in Fig. 12.26.

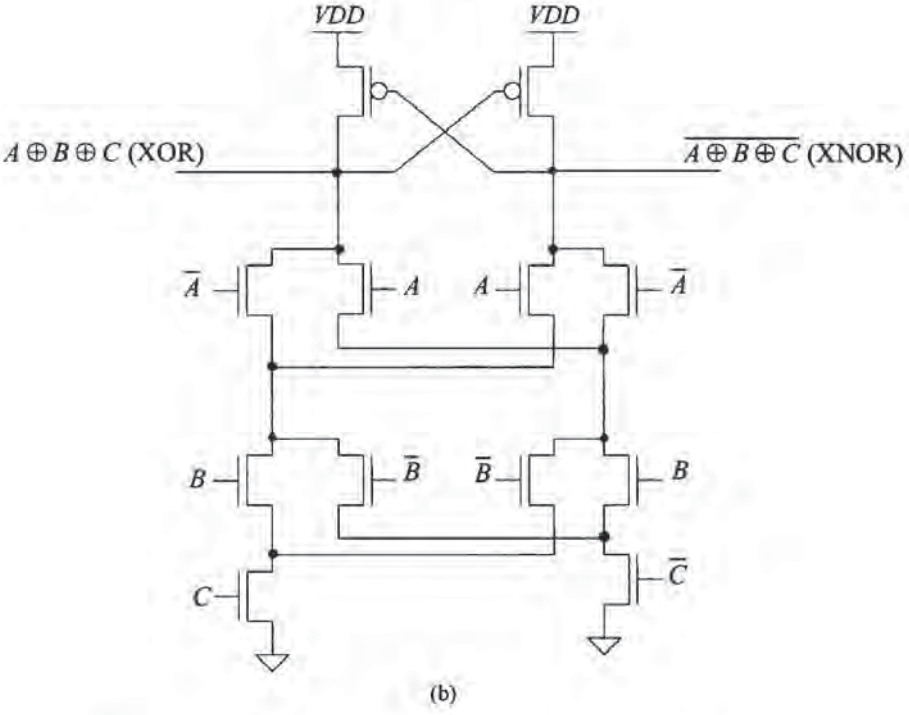
*Additional Examples*

Additional delay calculations using static logic gates (and other CMOS circuit building blocks) can be found in Sec. 13.4.

IPR2023-00697
Theta EX2019




(a)




(b)

**Figure 12.23** (a) Two-input and (b) three-input XOR/XNOR gates.

IPR2023-00697
Theta EX2019

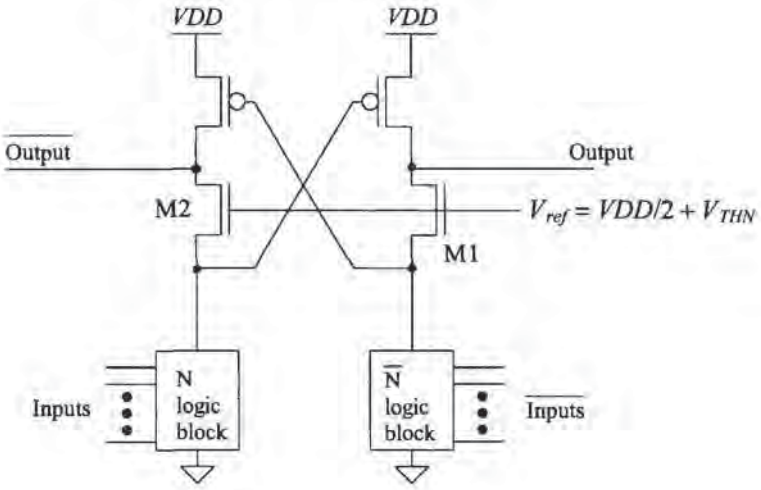



**Figure 12.24** DSL block diagram.

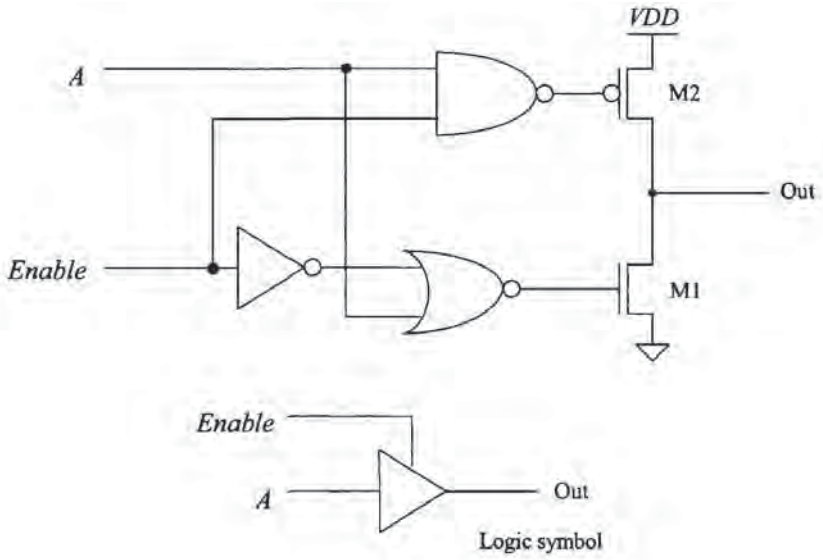



**Figure 12.25** Tri-state buffer.

IPR2023-00697
Theta EX2019

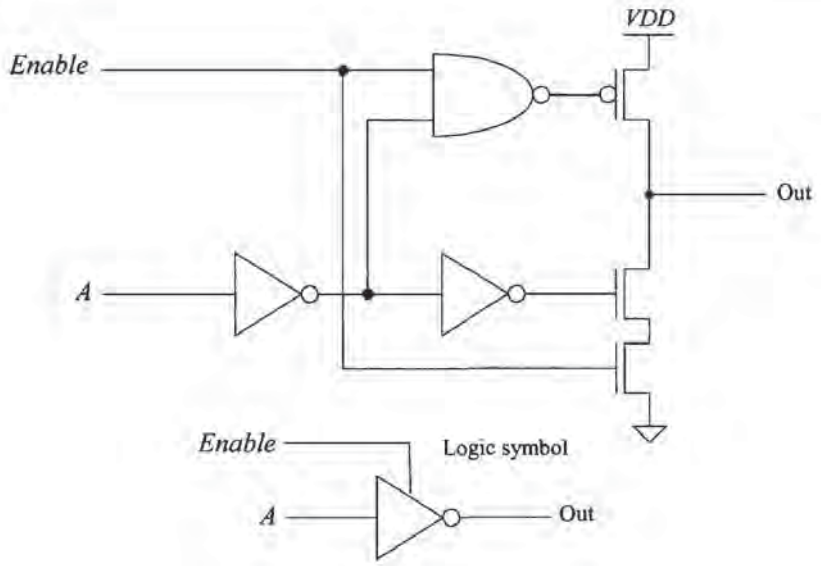



**Figure 12.26** Tri-state inverting buffer.

## ADDITIONAL READING

[1] I. Sutherland, R. F. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann, 1999. ISBN 978-1558605572

[2] M. I. Elmasry, *Digital MOS Integrated Circuits II,* IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.

[3] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI,* Kluwer Academic Publishers, 1992.

[4] M. Shoji, *CMOS Digital Circuit Technology,* Prentice-Hall, 1988. ISBN 0-13-138850-9.

## PROBLEMS

Use the 50 nm, short-channel process unless otherwise indicated.

**12.1** Design, lay out, and simulate the operation of a CMOS AND gate with a $V_{SP}$ of approximately 500 mV. Use the standard-cell frame discussed in Ch. 4 for the layout.

**12.2** Design and simulate the operation of a CMOS AOI half adder circuit using static logic gates.

**12.3** Repeat Ex. 12.3 for a three-input NOR gate. (Use the effective resistances to estimate the $V_{SP}$.)

IPR2023-00697
Theta EX2019

**12.4** Repeat Ex. 12.4 for a three-input NOR gate. (Use the effective resistances to estimate the $V_{SP}$.)

**12.5** Sketch the schematic of an OR gate with 20 inputs. Comment on your design.

**12.6** Sketch the schematic of a static logic gate that implements $\left(A+B\cdot\overline{C}\right)\cdot D$. Estimate the worst-case delay through the gate when driving a 50 fF load capacitance.

**12.7** Design and simulate the operation of a CSVL OR gate made with minimum-size devices.

**12.8** Design and simulate the operation of a tri-state buffer that has propagation delays under 5 ns when driving a 1 pF load. Assume that the maximum input capacitance of the buffer is 100 fF.

**12.9** Sketch the schematic of a three-input XOR gate implemented in AOI logic.

**12.10** The circuit shown in Fig. 12.27 is an edge detector. Discuss, and simulate, the operation of the circuit.

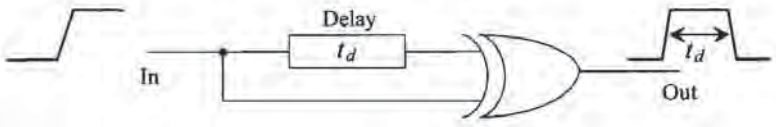



**Figure 12.27** An edge detector circuit.

IPR2023-00697
Theta EX2019

Chapter
13

# Clocked Circuits

The transmission gate (TG) is used in digital CMOS circuit design to pass or not pass a signal, see Sec. 10.2.1. The schematic and logic symbol of the transmission gate (TG) are shown in Fig. 13.1. The gate is made up of the parallel connection of an NMOS and a PMOS device. Referring to the figure when $S$ (for select) is high, we observe that the transmission gate passes the signal on the input to the output (noting the nodes we define as the input or output are interchangeable). The resistance between the input and the output can be estimated as $R_n \| R_p$. We begin this chapter with a description of the CMOS TG.




**Figure 13.1** The transmission gate.

## 13.1 The CMOS TG

Since the NMOS pass gate (PG) passes logic lows well and the PMOS PG passes logic highs well, putting the two complementary MOSFETs in parallel, as seen in Fig. 13.1, produces a TG that passes both logic levels well. The propagation delay-times of the CMOS TG in the configuration seen in Fig. 13.2 (with a large load capacitance) are estimated as

$$t_{PHL} = t_{PLH} = 0.7 \cdot (R_n \| R_p) \cdot C_{load} \qquad (13.1)$$

The capacitance on the $S$ input of the TG is the input capacitance of the NMOS device, or $C_{inn}$ (= $1.5C_{oxn}$). The capacitance on the $\bar{S}$ input of the TG is the input capacitance of the PMOS device, or $C_{inp}$.

IPR2023-00697
Theta EX2019

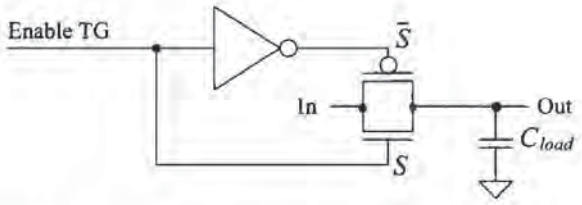



**Figure 13.2** The transmission gate with control signals shown.

Increasing the widths of the MOSFETs used in the TG reduces the propagation delay-times from the input to the output of the TG when driving a specific load capacitance. However, the delay-times in turning the TG on, the select lines going high, increase because of the increase in input capacitance. This should be remembered when simulating. Using a voltage source in SPICE for the select lines, which can supply infinite current to charge the input capacitance of the TG, gives the designer a false sense that the delay through the TG is limited by $R_n$ and $R_p$. Often, when simulating logic of any kind, the SPICE-generated control signals are sent through a chain of inverters so that the control signals more closely match what will actually control the logic on die (and the control signals have a finite source driving resistance).

**Example 13.1**

Estimate and simulate the delays through the TG circuit shown in Fig. 13.3 using the short-channel CMOS process and a load capacitance of 50 fF.

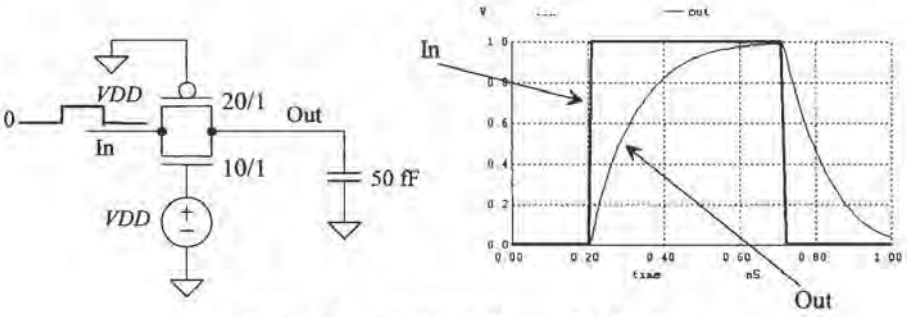



**Figure 13.3** TG circuit discussed in Ex. 13.1.

From Table 10.2 $R_n \| R_p = 1.7$ kΩ so that using Eq. (13.1) we get $t_{PLH} = t_{PHL} = 60$ ps. The SPICE simulation results are also seen in Fig. 13.3. ■

**Example 13.2**

Repeat Ex. 13.1 for the circuit seen in Fig. 13.4

The output load is initially at *VDD* and then discharged to ground when the TG turns on. The delay time calculation is exactly the same as the one used in Ex. 13.1. Also seen in the figure are the simulation results. ■

IPR2023-00697
Theta EX2019

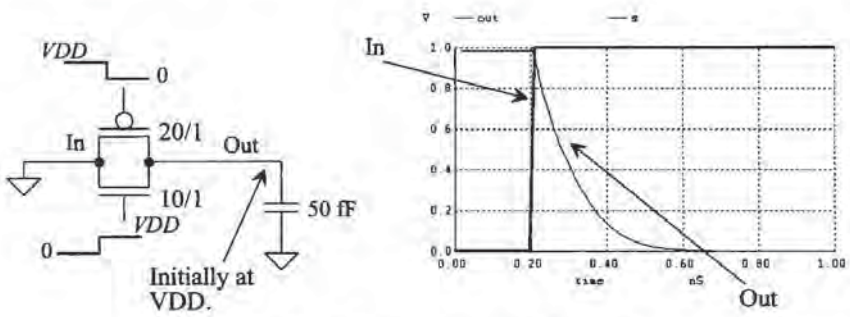



**Figure 13.4** TG circuit discussed in Ex. 13.2.

*Series Connection of TGs*

Consider the series connection of the CMOS transmission gates shown in Fig. 13.5. The equivalent digital model is also depicted in this figure. As seen in Figs. 10.17, 10.21, and the associated discussion, the capacitance on each MOSFET's source/drain (assuming triode operation) is $C_{ox}/2$. The total capacitance on each internal node in a series connection of TGs is then the sum of the oxide capacitances from each adjacent PG, that is, $C_{oxn} + C_{oxp}$. The delay through the series connection of TGs can be estimated using

$$t_{PHL} = t_{PLH} = 0.7 \cdot N \cdot (R_n || R_p)(C_{load}) + 0.35 \cdot (R_n || R_p)(C_{oxn} + C_{oxp})(N)^2 \tag{13.2}$$

The first term in this equation is simply the time needed to charge $C_{load}$ through the sum of the TG effective resistances, while the second term in the equation describes the RC transmission line effects.

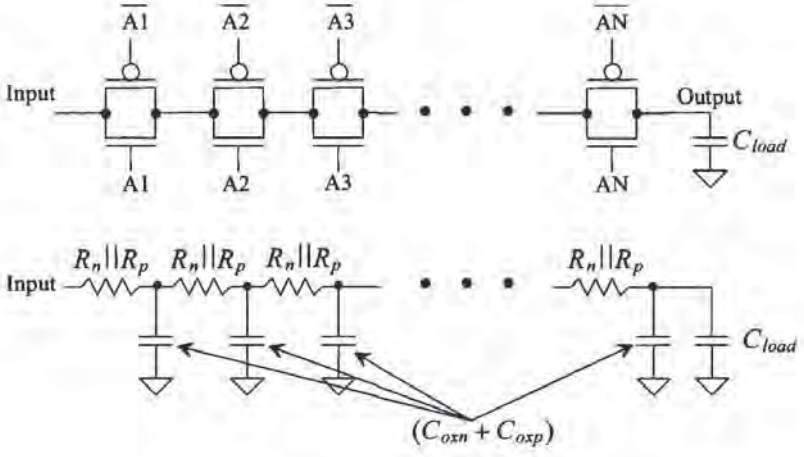



**Figure 13.5** Series connection of TGs with digital model.

IPR2023-00697
Theta EX2019

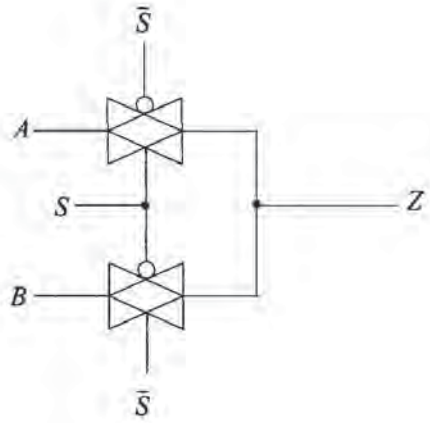



**Figure 13.6** Path selector.

## 13.2 Applications of the Transmission Gate

In this section we present some of the applications of the TG.

*Path Selector*

The circuit shown in Fig. 13.6 is a two-input path selector. Logically, the output of the circuit can be written as

$$Z = AS + B\bar{S} \tag{13.3}$$

When the selector signal $S$ is high, $A$ is passed to the output while a low on $S$ passes $B$ to the output.

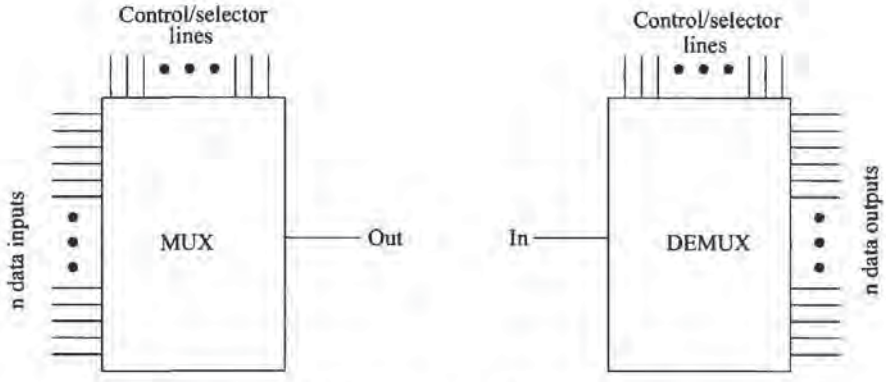



**Figure 13.7** Block diagram of MUX/DEMUX.

This same idea can be used to implement multiplexers/demultiplexers (MUX/DEMUX). Consider the block diagrams of a MUX and DEMUX shown in Fig. 13.7. The number of control lines is related to the number of input lines by

IPR2023-00697
Theta EX2019

$$2^m = n \tag{13.4}$$

where $n$ is the number of inputs (outputs) to the MUX (DEMUX) and $m$ is the number of control lines. A 4-to-1 MUX/DEMUX is shown in Fig. 13.8. Note that the MUX is bi-directional; that is, it can be used as a MUX or a DEMUX. The logic equation describing the operation of the MUX is given by

$$Z = A(S1 \cdot S2) + B(S1 \cdot \overline{S2}) + C(\overline{S1} \cdot S2) + D(\overline{S1} \cdot \overline{S2}) \tag{13.5}$$

Figure 13.9 shows the transistor-level implementation of the circuit in Fig. 13.8. Notice how the PMOS are grouped together (and laid out in the same n-well) while the NMOS are grouped together.

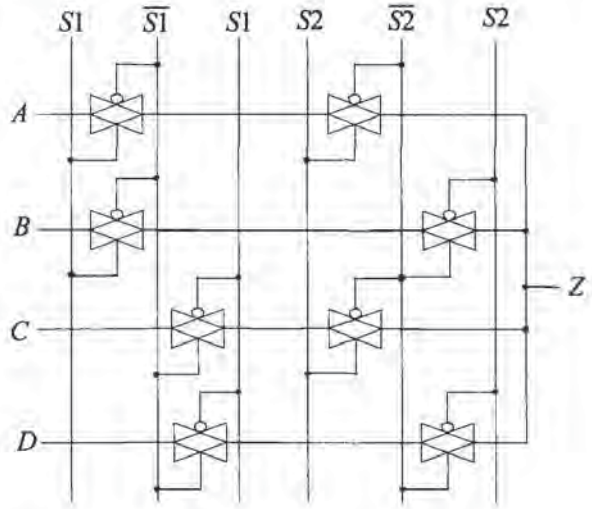



**Figure 13.8** Circuit implementation of a 4-to-1 MUX.

Figure 13.10a shows an NMOS PG implementation of the 4-to-1 MUX. The pass transistor implementation is simpler, using fewer transistors, at the price of a threshold voltage drop from input to output when the input is a high (*VDD*). A simplified version of the circuit of Fig. 13.10a is shown in Fig. 13.10b. Here the MOSFETs connected to *S2* and $\overline{S2}$ are combined to reduce the total number of MOSFETs used. The reduction of the total number of MOSFETs used can be extended to an n-input (output) MUX (DEMUX) (see Fig. 16.43). Again, it should be remembered that a DEMUX can be formed using the circuits of Figs. 13.8 or 13.10 by switching the inputs with the outputs.

*Static Gates*

The TG can be used to form static logic gates. Consider the OR gate shown in Fig. 13.11. To understand the operation of the gate, consider the case when both $A$ and $B$ are low. Under these circumstances the pass transistor, M1 is off, and the TG is on. Since the input $B$ is low, a low is passed to the output. If $A$ is high, M1 is on and $A$ is passed to the output. If $B$ is high and $A$ is low, $B$ is passed to the output through the TG. If both $A$ and $B$ are high, the TG is off and M1 is on passing $A$, a high, to the output.

IPR2023-00697
Theta EX2019

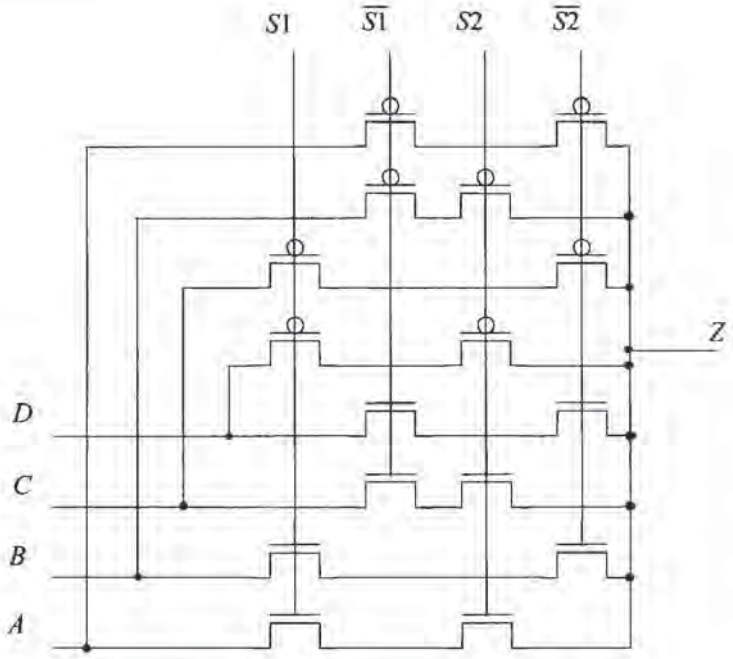



**Figure 13.9** Transistor implementation of Fig. 13.8.

Figure 13.12 shows an XOR and an XNOR gate made using TGs. Consider the XOR gate with both $A$ and $B$ low. Under these circumstances, the top TG is on and its output is connected to $A$, a low. If both inputs are high, the bottom TG connects the output to $\overline{A}$, again a low. If $A$ is high and $B$ is low, the top TG is on and the output is connected to $A$, a high. Similarly, if $A$ is low and $B$ is high, the bottom TG is on and connects the output to $\overline{A}$, a high.

## 13.3 Latches and Flip-Flops

*Basic Latches*

Consider the set-reset latch (SR latch) shown in Fig. 13.13 made using NAND gates. The logic symbol and truth table are also shown in this figure. Consider the case when $S$ is high and $R$ is low. Forcing $R$ low causes $Q$ to go high. Since $S$ is high and $Q$is high, the $\overline{Q}$ output is low. Now consider the case when both $S$ and $R$ are low. Under these circumstances, the latch's outputs are both high. This latch can easily be designed and laid out with the techniques of Ch. 12 (see also Fig. 15.6).

An alternative implementation of the SR latch is shown in Fig. 13.14 using NOR gates. Consider the case when $S$ is high and $R$ is low. For the NOR gate, a high input forces the output of the gate low. Therefore, the $\overline{Q}$ output is low whenever the $S$ input is high. Similarly, whenever the $R$ input is high, the $Q$ output must be low. The case of both inputs being high causes both $Q$ and $\overline{Q}$ to go low, or in other words the outputs of the latch are no longer complements.

IPR2023-00697
Theta EX2019

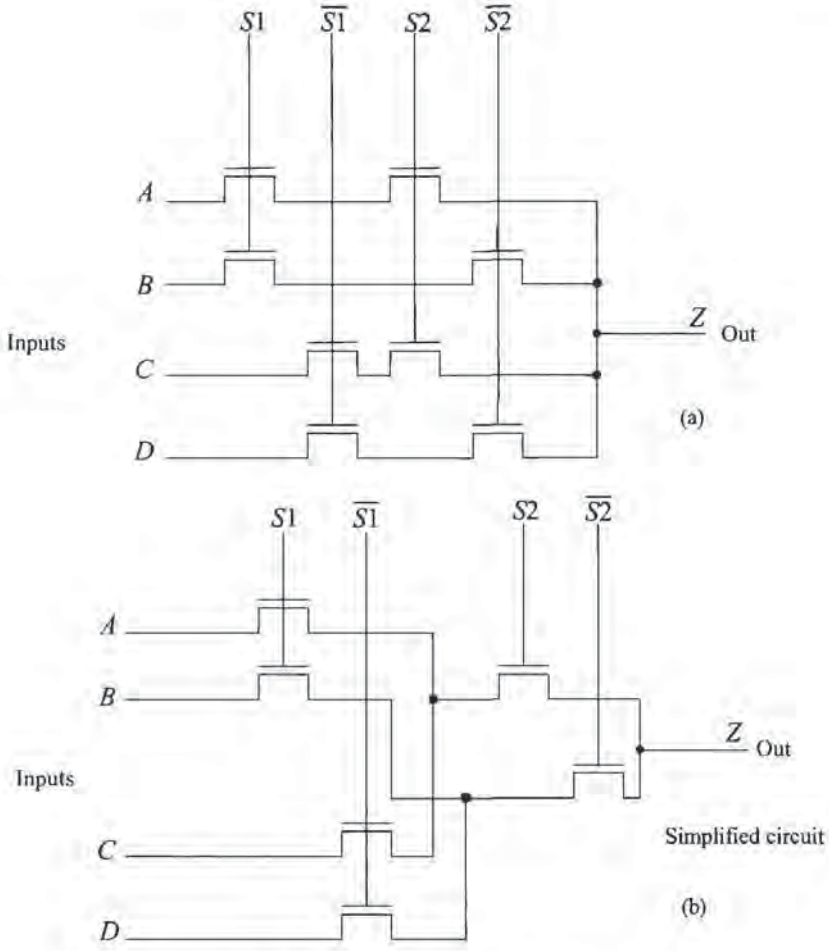



**Figure 13.10** MUX/DEMUX using pass transistors.

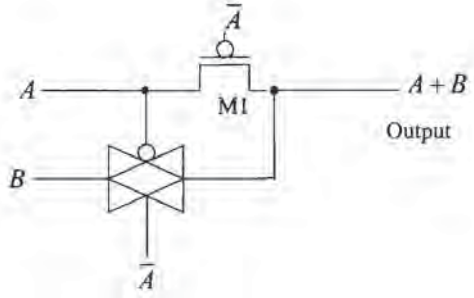



**Figure 13.11** TG-based OR gate.

IPR2023-00697
Theta EX2019

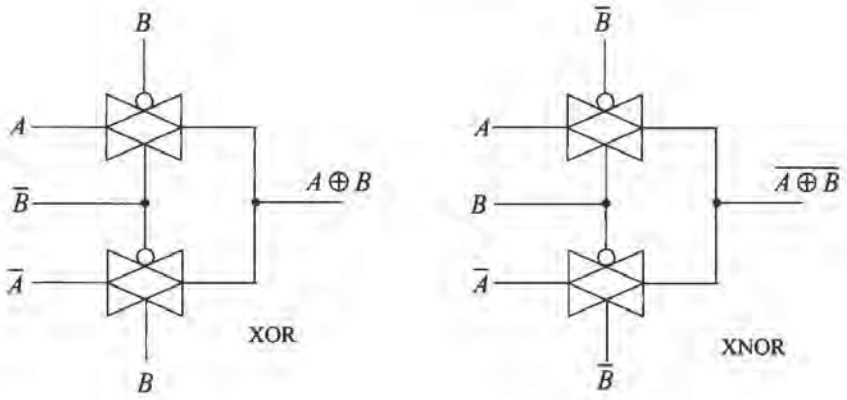
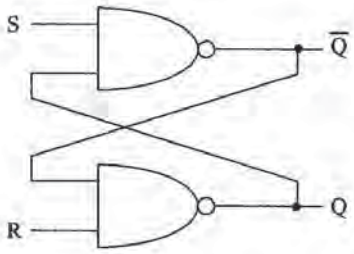


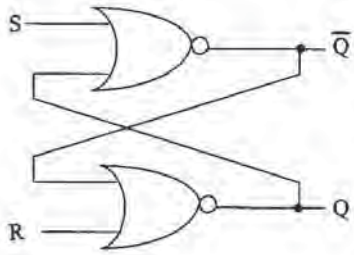

**Figure 13.12** TG implementation of XOR/XNOR gate.




Truth table

| $S$ | $R$ | $Q$ | $\overline{Q}$ |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | $Q$ | $\overline{Q}$ |

**Figure 13.13** Set-reset latch made using NAND gates.




Truth table

| $S$ | $R$ | $Q$ | $\overline{Q}$ |
|---|---|---|---|
| 0 | 0 | $Q$ | $\overline{Q}$ |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |

**Figure 13.14** Set-reset latch made using NOR gates.

IPR2023-00697
Theta EX2019

*An Arbiter*

One of the uses of the NAND latch is in an arbitration circuit (called an arbiter, Fig. 13.15). This circuit can be very useful in asynchronous circuit design or in clock synchronization circuits (see Ch. 19). To understand the operation of the arbiter in Fig. 13.15, let's consider the case when both In1 and In2 are low. Both NAND gates' outputs are high and the two inverters' outputs, that is Out1 and Out2, are low. When In1 goes high, the output of X1 goes low, while the output of X2 remains high. This causes Out1 to go high and Out2 to remain low. Notice that with the output of X1 going low the power supplied to the PMOS in the inverter connected to Out2 is removed (making it impossible for Out2 to go high). When In2 goes high, while In1 is already high, the fact that the output of X1 is a low keeps the output of X2 high and Out2 low. When In1 goes low, with In2 high, Out1 goes low and Out2 goes high. The usefulness of this circuit occurs when In1 and In2 transition high at nearly the same times. The way the inverters are powered by the NAND gates makes it impossible for both Out1 and Out2 to go high at the same time. *Thus an arbiter is useful for determining which input arrives first.*

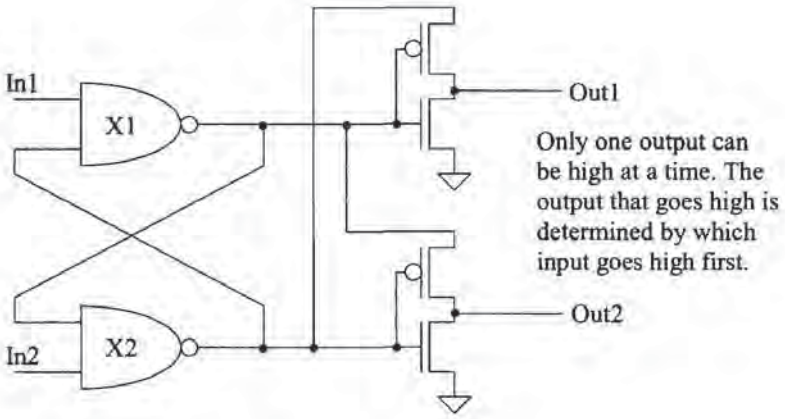



**Figure 13.15** An arbiter made using NAND gates.

To ensure that the arbiter makes the decision quickly (there isn't a long delay or *metastability* when the two inputs are arriving at the same time), two inverters can be added in series with the NAND gate outputs. This addition of inverters simply increases the gain of the NAND gates (their VTCs get steeper at the switching point, see Fig. 12.3 in the last chapter).

*Flip-Flops and Flow-through Latches*

A flow-through latch is a storage circuit whose output changes with a clock signal level (so a flow-through latch is often called a level-sensitive latch). A flip-flop (FF) is a storage circuit that changes states on the rising or falling edge of a clock signal.

Most FFs in CMOS IC design are based on the cross-coupled inverters seen in Fig. 13.16. Also seen in this figure are the VTCs for the two inverters. I1's characteristics are drawn with its input on the x-axis and its output on the y-axis (the normal curves as

IPR2023-00697
Theta EX2019

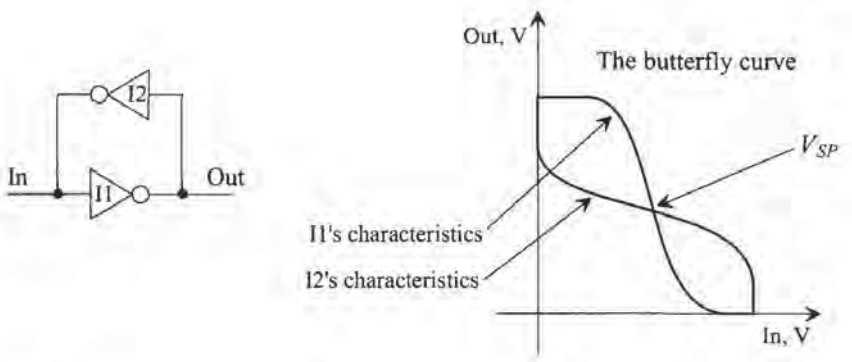



**Figure 13.16** A cross-coupled inverter (a latch) and the input output characteristics.

seen in Fig. 11.2 back in Ch. 11). I2's input is labeled Out in the figure while its output is labeled In (and so its input is drawn as the y-axis and its output is drawn as the x-axis). The big concern when designing a FF with cross-coupled inverters is *metastability* (meaning changing stability). When the input signal moves to the switching-point of the inverters, both In and Out will be at $V_{SP}$ (a stable state). This isn't where we want the circuit to operate because $V_{SP}$ isn't a valid logic level voltage. Driving the input higher or lower enables the positive feedback present in the cross-coupled inverters to drive the In/Out nodes to valid logic levels (the other stable states). To avoid metastability, we can 1) use longer length inverters (which increases the gains of the inverters at the cost of longer delays), 2) use two inverters in series with the outputs of each inverter in Fig. 13.16 (again, to boost the gain) so that three inverters are used between In and Out, 3) ensure the In signal is driven with good logic levels (and a low impedance source to overdrive the output of I2), or 4) add a switch on the output of I2 to disconnect it when the In signal is changing to ensure that the input signal doesn't have to "fight" with the output of I2 to drive the signal to a valid logic level (the input signal range must still swing to valid logic levels). Figure 13.17 shows how metastability can result in a long delay before the outputs move to valid logic states. In this simulation the two capacitors are charged to 500 mV. It takes approximately 2.5 ns for the In/Out signals to change states.

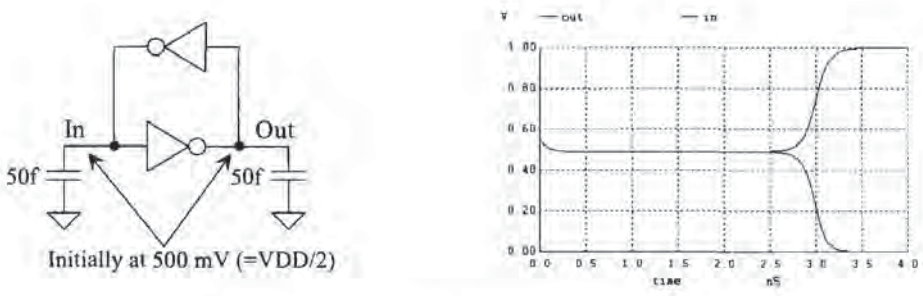



**Figure 13.17** The delay associated with metastability.

IPR2023-00697
Theta EX2019

Note that trying to move the switching point voltages of each inverter to differing levels makes the problem of metastability worse. When each inverter's $V_{SP}$ is the same the gain around the loop is the highest (and so it's possible noise alone can cause the inverters to snap to a valid logic state).

Figure 13.18 shows a level-sensitive latch. When the clock signal is high, the data or D input passes through the NMOS PG and drives the cross-coupled inverters. The feedback inverter's lengths are made long to ensure that the source driving the D input can "overpower" the feedback inverter and allow the output to switch states. This circuit, while simple, has several fundamental issues. To begin, we know that an NMOS pass gate's output will swing from 0 to $VDD - V_{THN}$. To optimize the noise margins, we lower the $V_{SP}$ of I1. Here we reduce the width of the PMOS from 20 to 10. Another concern is the contention current that flows when the D input is fighting with the output of I2 for control of I1's input. By increasing the lengths in I2 to 10, we lower this wasted current.

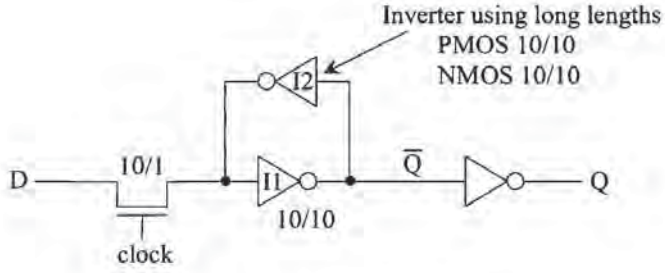



**Figure 13.18** A level-sensitive latch.

Figure 13.19 shows the input and output simulation results for the latch in Fig. 13.18. When the clock signal is high, the input flows to the output (with a delay). When the clock signal goes low, the value on the D input is captured and remains on the Q output until the clock signal goes back high again. Notice that at 8 ns, when the clock signal goes high, the delay between the D input going high and the Q output going high is considerable.

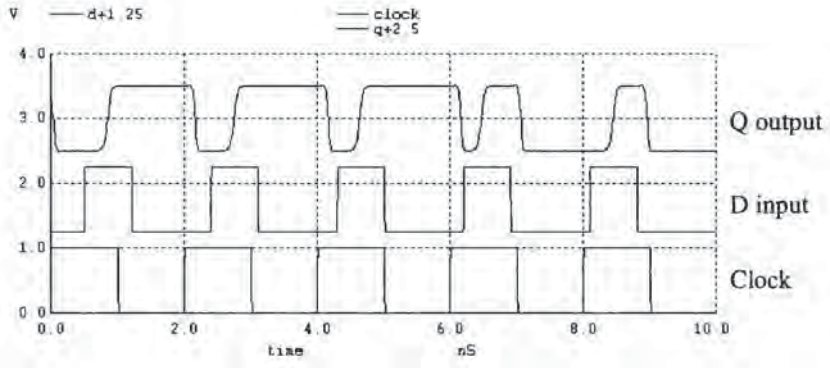



**Figure 13.19** Simulating the level-sensitive latch in Fig. 13.18.

IPR2023-00697
Theta EX2019

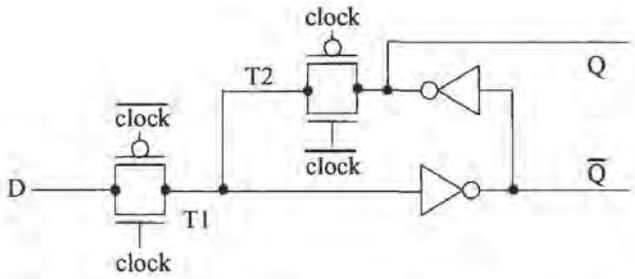



**Figure 13.20** A higher performance level-sensitive latch.

Figure 13.20 shows a modification of the latch in Fig. 13.18. A TG is used on the input of the latch instead of a PG. This improves the noise performance of the circuit (at the cost of an additional clock signal, that is, the complement of clock). We've also added a TG in series with the feedback inverter. When T1 is on, the Q output follows the D input. When T1 is on, T2 is off so that the input doesn't have to fight with the feedback inverter. When T1 turns off, T2 turns on and the value of D at that instance is stored in the inverters. Figure 13.21 shows the simulation results using this latch with the input signals used to generate Fig. 13.19. Notice, for example, the improvement in performance over what's seen in Fig. 13.19 when the clock transitions high at 8 ns.

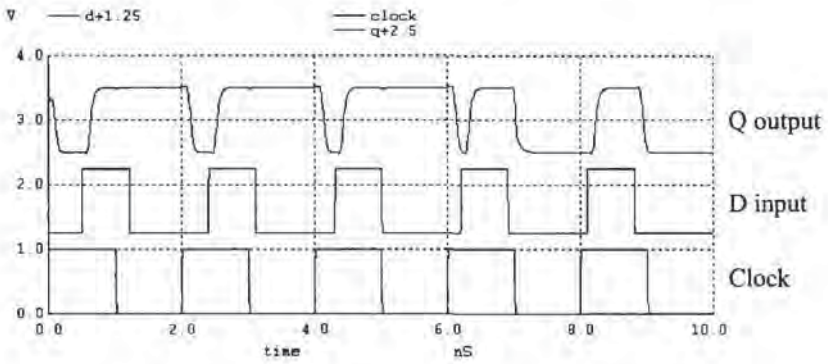



**Figure 13.21** Simulating the level-sensitive latch in Fig. 13.20.

*An Edge-Triggered D-FF*

Notice that the value of the D input in the circuit seen in Fig. 13.20 is stored or captured when clock goes low. To implement an edge-triggered FF, we can use two of these level-sensitive latches in cascade. When the clock is low, the first stage tracks the D input and the second stage holds the previous output. When clock goes high, the first stage captures the input and transfers it to the second stage. The first stage is often called the "master" latch, while the second stage is the "slave" latch. Figure 13.22 shows an implementation of an edge-triggered D-FF. When clock is low, T1 and T4 are on. T2 and T3 are off. The D input flows through to point A and its complement to point B. When clock goes high, T1 and T4 shut off while T2 and T3 turn on. This causes the value of the D input, when

IPR2023-00697
Theta EX2019

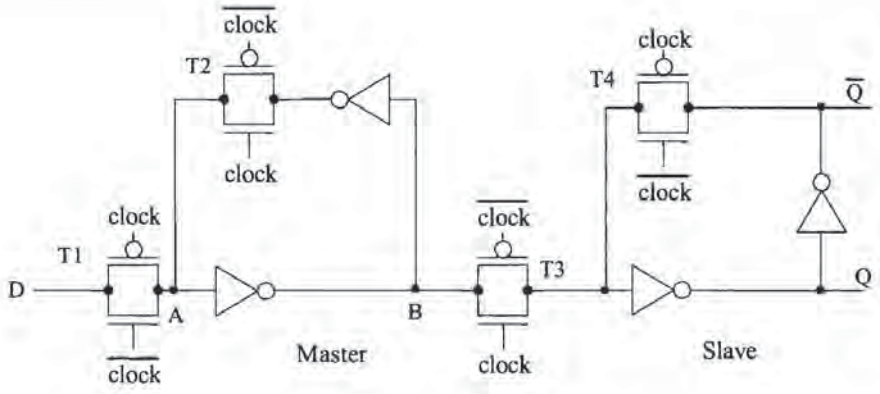



**Figure 13.22** An edge-triggered D-FF.

clock transitioned high, to be captured and passed to the Q output. When clock goes back low, T1 and T4 turn on. The value of D on the Q output is then circulated around the two inverters. The Q output can change states again when clock goes back high. Figure 13.23 shows results of simulating this FF with SPICE. Note the output only changes on the rising edge of the clock signal.

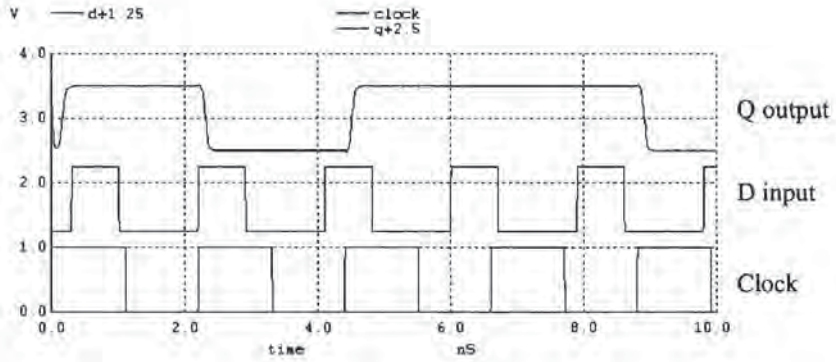



**Figure 13.23** Simulating the operation of the edge-triggered FF in Fig. 13.22.

Figure 13.24 shows the addition of NAND gates to the circuit of Fig. 13.22 for clear and set inputs. When clear goes low, the outputs of N1 and N4 go high. If T3 is on or T4 is on, the outputs of the FF are forced to Q = 0 (and thus its complement is set to a 1). If the set input goes low, then Q is forced to a 1. If both clear and set are pulled low, then both the Q output and its complement are driven high. This feature can useful in some situations.

Note that it is **very important** for the clock signals we use to have quick rise times. If they don't, then the FF may not function correctly. Often the clock signals are applied to a string of inverters both to sharpen their transition times and to generate the complement clock required for the TGs.

IPR2023-00697
Theta EX2019

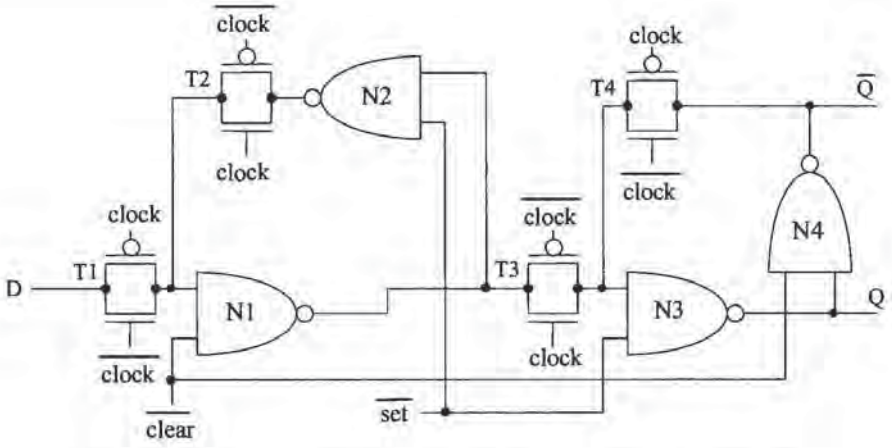



**Figure 13.24** An edge-triggered FF with asynchrounous set and clear.

*Flip-Flop Timing*

The data must be *set up* or present on the D input of the FF (see Fig. 13.22) a certain time before we apply the clock signal. This time is defined as the setup time of the FF. To understand the origin of this time, consider the time it takes the signal at D to propagate through T1 and the inverter to node B. Before the clock pulse can be applied, the logic level D must be settled on node B. Consider the waveforms of Fig. 13.25. The time between D going high (or low) and the clock rising edge is termed the setup time of the FF and is labeled, $t_s$.

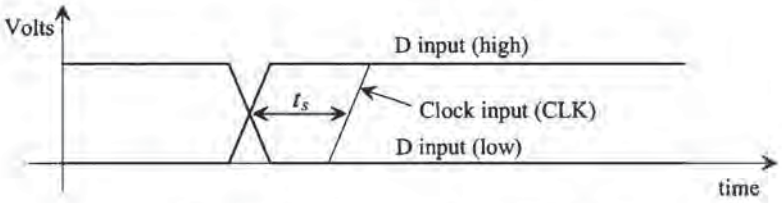



**Figure 13.25** Illustrating D FF setup time.

The wanted D input must be applied $t_s$ before the clock pulse is applied. Now the question becomes "How long does the wanted D input have to remain applied to, or *held* on, the input of the FF after the clock pulse is applied?" This time is called the hold time, $t_h$, and is illustrated in Fig. 13.26. Shown in this figure, $t_h$ is a positive number. However, inspection of Fig. 13.22 shows that if the D input is removed slightly before the clock pulse is applied, node B will remain unchanged because of the propagation delay from the D input to node B. Analysis of this FF would yield a negative hold time. In other words, for the point B to charge to D, a time labeled $t_s$ is needed. Once point B is charged, the D input can be changed as long as the clock signal occurs within $t_h$.

IPR2023-00697
Theta EX2019

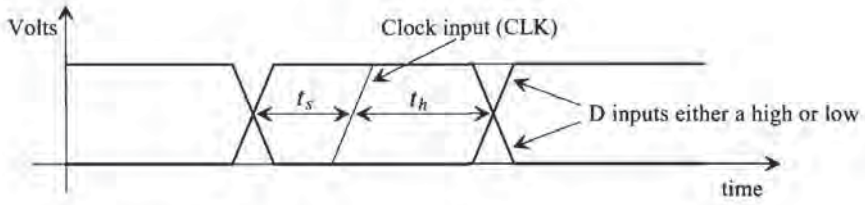



**Figure 13.26** Illustrating D FF hold time.

One final important comment regarding the clock input of an FF is (again) in order. If the clock input risetime is slow, the FF will not function properly. There will not be an abrupt transition between the sets of transmission gates turning on and off. The result will be logic levels at indeterminate states. What is usually done to eliminate this problem is to buffer the clock input through several inverters. This has the effect of speeding up the leading and trailing edges of a slow input pulse and presenting a lower input capacitance on the clock input to whatever is driving the FF. The main disadvantage is the increase in delay times, $t_{PHL}$ and $t_{PLH}$ (defined by clock to output), of the FF. In general, the FF of Fig 13.22 should not be laid out without buffering the clock inputs.

The minimum pulse width of the clock, set, or clear inputs in Figs. 13.22 and 13.24 is labeled $t_w$. The minimum width is determined by the delay through (referring to Fig. 13.20) two NAND gates and a TG. The last timing definition we will comment on here is the recovery time, that is, the time between removing the set or clear inputs and a valid clock input. This variable is labeled $t_{rec}$.

## 13.4 Examples

In this section we give examples of delay calculations and comments on how to decrease the delays and the associated performance costs. Throughout this section we use the 50 nm short-channel process.

---

**Example 13.3**

Estimate the input capacitance and the delay through the circuit seen in Fig. 13.27. Compare the hand calculation estimates to simulation results.

Using the information seen in Table 10.2, we can calculate the first inverter's input capacitance, and thus the input capacitance of the circuit,

$$C_{in1} = \frac{3}{2} \cdot (C_{oxn1} + C_{oxp1}) = 0.938\,fF + 1.875\,fF = 2.81\,fF$$

Since the widths of the second inverter are 10 times larger than the widths of the first inverter, the input capacitance of the second inverter is 28.1 fF.

The delay through the first inverter, that is, the delay from the node labeled "In" to the node labeled "N1" in Fig. 13.27 is calculated as

$$t_{PHL1} + t_{PLH1} = 0.7 \cdot (3.4k + 3.4k) \cdot (0.625 + 1.25 + 9.38 + 18.75)\,fF = 143\,ps$$

noting, because $t_{PHL} = t_{PLH}$, that $t_{PHL1} = t_{PLH1} = 71.5\,ps$.

IPR2023-00697
Theta EX2019

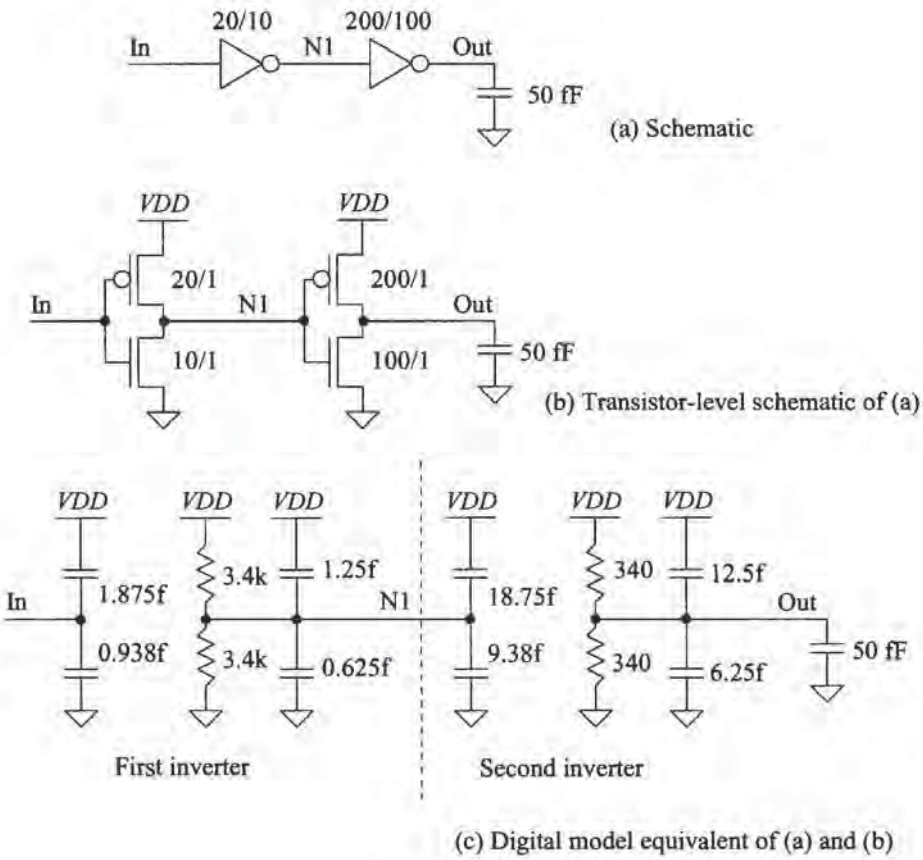



**Figure 13.27** Circuit used in Ex. 13.3.

The delay through the second stage, that is, from N1 to the output is calculated as

$$t_{PHL2} + t_{PLH2} = 0.7 \cdot (340 + 340) \cdot 68.75\, fF = 33\, ps$$

Again as seen in the first inverter, the effective switching resistance of the PMOS is equal to the NMOS device's effective resistance so $t_{PHL2} = t_{PLH2} = 16.5\, ps$.

The overall delay through both inverters is

$$t_{PHL} = t_{PLH} = 88\, ps$$

Figure 13.28 shows the SPICE simulation results. The delays, from the simulations, are approximately 120 ps.

To decrease the delay through the circuit, the simplest solution is to increase the widths of the first inverter. Of course, this increases the circuit's input capacitance. Note that the *analysis* of this circuit has nothing to do with the *design* equations used for implementing a buffer, Sec. 11.4. ■

IPR2023-00697
Theta EX2019

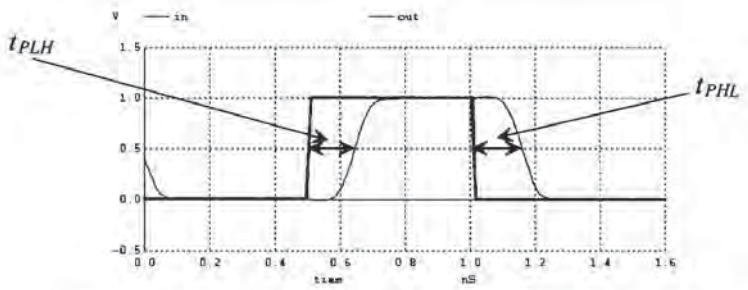



**Figure 13.28** Simulating the circuit in Ex. 13.3 and Fig. 13.27.

**Example 13.4**

In Ex. 13.3 the devices were sized to give (ideally) equal propagation delays. Repeat Ex. 13.3 for the circuit seen in Fig. 13.29.

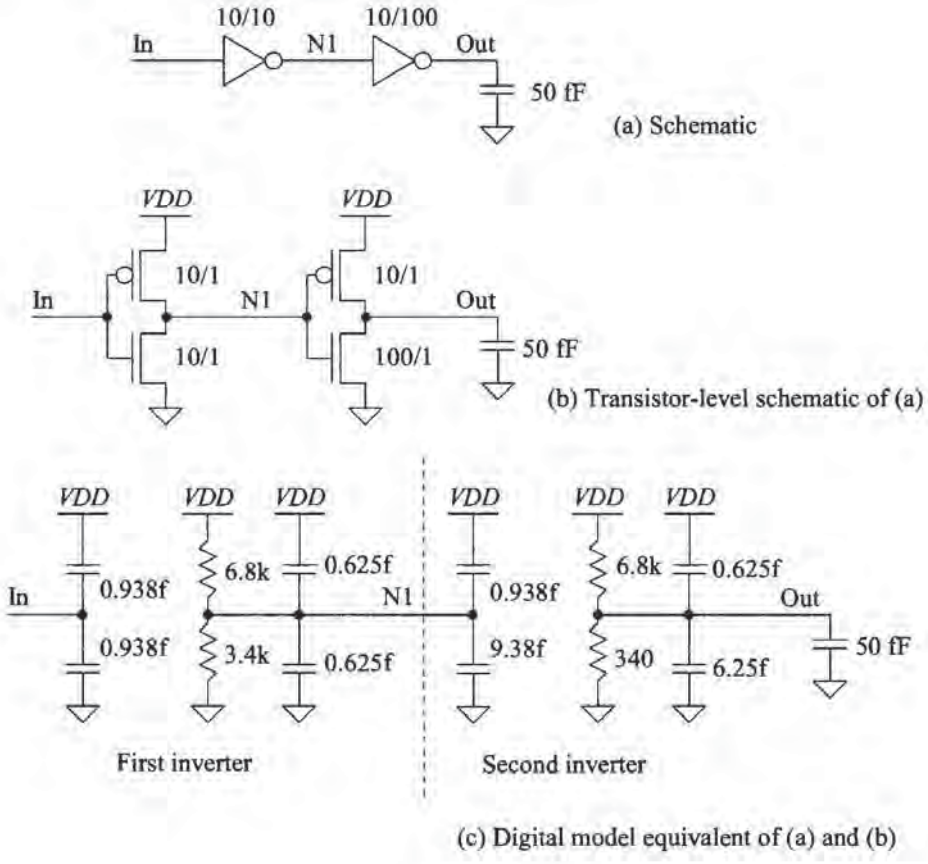



**Figure 13.29** Circuit used in Ex. 13.4.

IPR2023-00697
Theta EX2019

The PMOS device has a width of 10 which is half of the width used for the data in Table 10.2. We can write the effective switching resistance and oxide capacitance for this device as

$$R_p = 6.8k \text{ and } C_{oxp} = 0.625\,fF$$

As seen in Fig. 13.29, the input capacitance of the circuit is 1.876 fF.

Calculating $t_{PLH}$ begins by noting that when the input goes high the NMOS device in the first inverter turns on. This turns the PMOS device in the second inverter on. The overall delay is calculated as

$$t_{PLH} = \overbrace{0.7 \cdot 3.4k \cdot 11.57\,fF}^{\text{First inverter's delay}} + \overbrace{0.7 \cdot 6.8k \cdot 56.88\,fF}^{\text{Second inverter's delay}} \approx 300\,ps$$

Similarly, when the input goes low, the PMOS in the first inverter turns on and the NMOS in the second inverter turns on, pulling the output low

$$t_{PHL} = \overbrace{0.7 \cdot 6.8k \cdot 11.57\,fF}^{\text{First inverter's delay}} + \overbrace{0.7 \cdot 340 \cdot 56.88\,fF}^{\text{Second inverter's delay}} \approx 70\,ps$$

The simulation results are seen in Fig. 13.30. ■

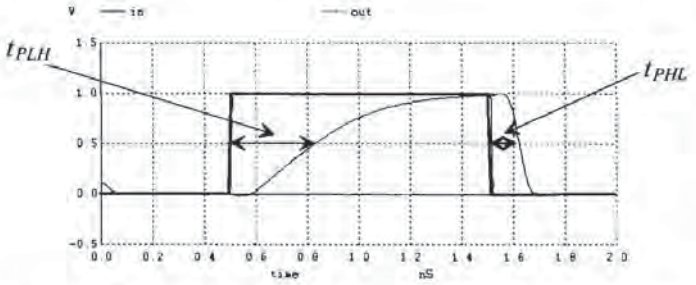



**Figure 13.30** Simulating the operation of the circuit in Fig. 13.29.

**Example 13.5**

Estimate the delay through the circuit seen in Fig. 13.31 using 20/1 PMOS and 10/1 NMOS devices.

Since the load capacitance, 50 fF, is much larger than the output capacitance of the NAND gate, we don't include the NAND gate's capacitance contributions in the digital model seen in Fig. 13.31c. When the input goes high, the NMOS device in the inverter turns on and the PMOS device in the NAND gate turns on (causing the output to go high). The low-to-high delay time is calculated as

$$t_{PLH} = 0.7 \cdot 3.4k \cdot 4.7f + 0.7 \cdot 3.4k \cdot 50f = 130\,ps$$

The high-to-low delay time is calculated, noting the two NMOS in series when pulling the output low, as

$$t_{PHL} = 0.7 \cdot 3.4k \cdot 4.7f + 0.7 \cdot 6.8k \cdot 50f = 250\,ps$$

Figure 13.32 shows the simulation results. ■

IPR2023-00697
Theta EX2019

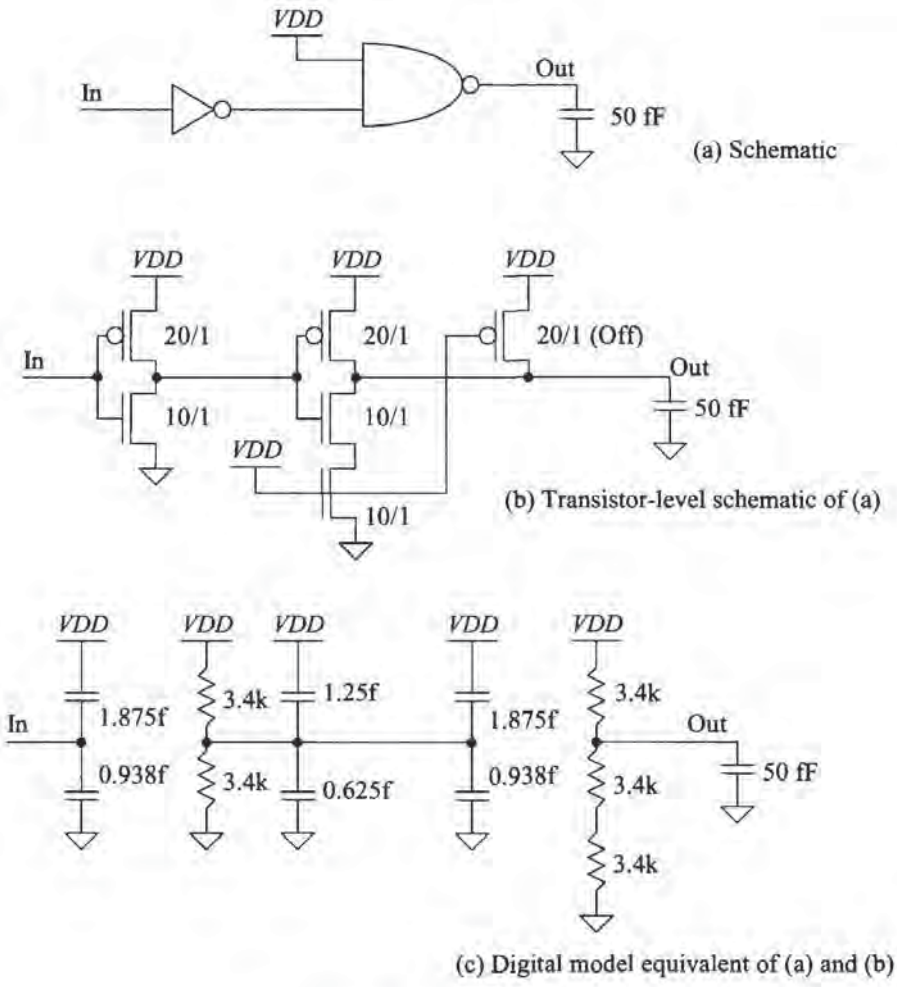



**Figure 13.31** Circuit used in Ex. 13.5.

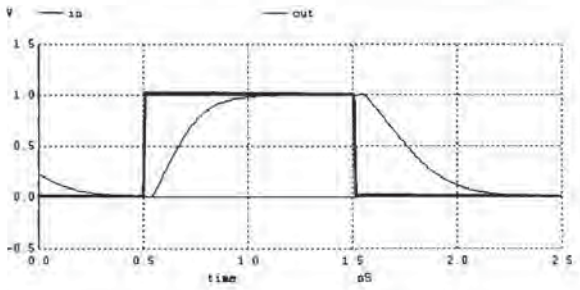



**Figure 13.32** Simulating the circuit in Fig. 13.31.

IPR2023-00697
Theta EX2019

**Example 13.6**

The circuit seen in Fig. 13.33 is the input portion of the edge triggered D-FF seen in Fig. 13.22. Estimate the delay from the D input to points A and B. Compare the estimate to SPICE simulations. Note that this delay is important because it directly determines the setup time of the FF.

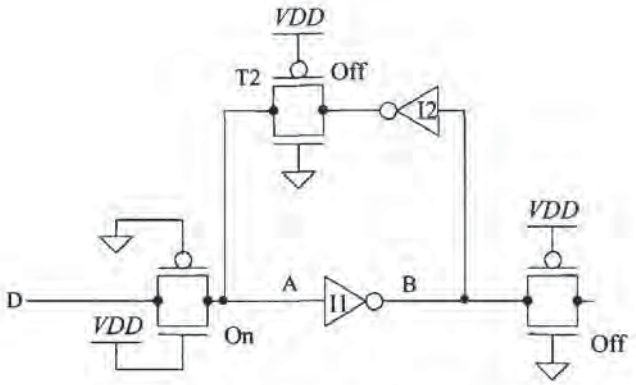



**Figure 13.33** Section of a D-FF used in Ex. 13.5.

Node A in Fig. 13.33 is driven through the on TG. The resistance of the TG is $R_n||R_p$. The capacitance on node A is

$$C_A = \overbrace{\frac{C_{oxn}}{2}+\frac{C_{oxp}}{2}}^{\text{Capacitance from the on TG}} + \overbrace{\frac{3}{2}\cdot(C_{oxn}+C_{oxp})}^{\text{Input capacitance of the inverter}} = 3.75\,fF$$

The first term is the capacitance on node A due to the on TG (see Fig. 10.7). The second term is the input capacitance of the inverter. Note that we have not included the capacitive loading from the off TG. This causes the actual delays to be longer than what we calculate here. (A good exercise at this point is to estimate the capacitive loading of the TGs and include them in the calculations here.) The capacitance on node B is made up of an inverter input capacitance and an inverter output capacitance or

$$C_B = \overbrace{\frac{3}{2}\cdot(C_{oxn}+C_{oxp})}^{\text{Inverter input capacitance}} + \overbrace{(C_{oxn}+C_{oxp})}^{\text{Inverter output capacitance}} = 4.7\,fF$$

The delay from the D input to point A is estimated as

$$t_{PLHA} = t_{PHLA} = 0.7\cdot R_n||R_p\cdot C_A = 4.5\,ps$$

The delay from point A to point B through the inverter is

$$t_{PLHB} + t_{PHLB} = 0.7\cdot(R_n+R_p)\cdot C_B = 22.5\,ps$$

The propagation delay from D to B is then

IPR2023-00697
Theta EX2019

$$t_{PLH} = t_{PHL} = 15.75\ ps$$

The simulation results are seen in Fig. 13.34. The delay to point A is approximately 30 ps, while the delay to point B is about 90 ps. It's important to reiterate the importance of simulations when determining delays. Hand calculations, as this example shows, have limitations. Hand calculations are still important, however, because they reveal the location of the dominant delays in a digital circuit. ■

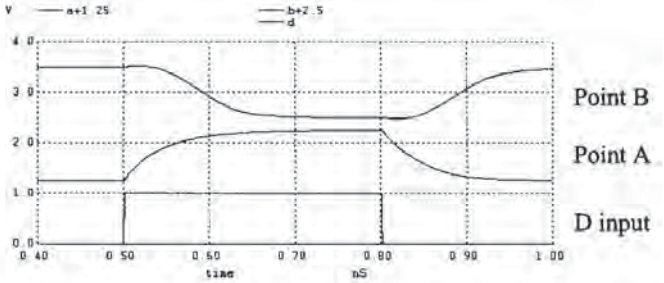



**Figure 13.34** Simulating the delays through the latch seen in Fig. 13.33.

## ADDITIONAL READING

[1] M. I. Elmasry, *Digital MOS Integrated Circuits II,* IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.

[2] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI,* Kluwer Academic Publishers, 1992.

[3] M. Shoji, *CMOS Digital Circuit Technology,* Prentice-Hall, 1988. ISBN 0-13-138850-9.

## PROBLEMS

Unless otherwise stated, use the 50 nm, short-channel CMOS process with the parameters seen in Table 10.2.

**13.1** Estimate and simulate the delay through 10 TGs connected a 50 fF load capacitance.

**13.2** Design and simulate the operation of a half adder circuit using TGs.

**13.3** Sketch the schematic of an 8-to-1 DEMUX using NMOS PGs. Estimate the delay through the DEMUX when the output is connected to a 50 fF load capacitance.

**13.4** Verify, using SPICE, that the circuit seen in Fig. 13.12 operates as an XOR gate.

**13.5** Simulate the operation of an SR latch made with NAND gates. Show all four possible input logic combinations.

**13.6** Simulate the operation of the arbiter seen in Fig. 13.15. Show how two inputs arriving at nearly the same time results in only one output going high.

IPR2023-00697
Theta EX2019

**13.7** Show, using simulations, how making the feedback inverter, I2, in Fig. 13.18 stronger (decrease the lengths) can result in the output having either a long delay or not fully switching.

**13.8** Redesign the FF in Fig. 13.22 without T2 and T4 present. Simulate the operation of your design. Show, by using a limited amplitude on the D input, how point B can have a metastability problems.

**13.9** In your own words describe setup and hold times. Use the D-FF in Fig. 13.22 and simulations to help support your clear descriptions.

IPR2023-00697
Theta EX2019

Chapter
14

# Dynamic Logic Gates

Dynamic or clocked logic gates are used to decrease complexity, increase speed, and lower power dissipation. The basic idea behind dynamic logic is to use the capacitive input of the MOSFET to store a charge and thus remember a logic level for use later. Before we start looking into the design of dynamic logic gates, let's discuss leakage current and the design of clock circuits.

## 14.1 Fundamentals of Dynamic Logic

Consider the NMOS pass gate (PG) driving an inverter, as shown in Fig. 14.1. If we clock the gate of the PG high, the logic level on the input, point A, will be passed to the input of the inverter, point B. If this logic level is a "0," the input of the inverter will be forced to ground, while a logic "1" will force the input of the inverter to $VDD - V_{THN}$. When the clock signal goes low, the PG shuts off and the input to the inverter "remembers" the logic level. In other words, when the PG turns on, the input capacitance of the inverter is either charged to $VDD - V_{THN}$ or discharged to ground through the PG. As long as this charge is present on the parasitic input capacitance of the inverter, the logic value is remembered. What we are concerned with at this point are the leakage mechanisms which can leak the stored charge off the node. A node, such as the one labeled B in Fig. 14.1, is called a dynamic node or a storage node. Note that this node is a high-impedance node and is easily susceptible to noise (see Ex. 3.5).

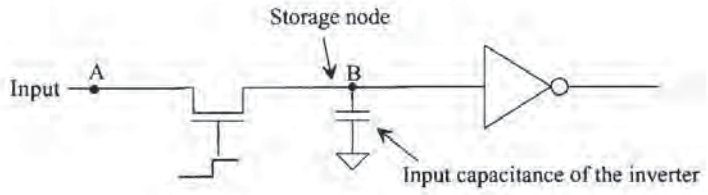



**Figure 14.1** Example of a dynamic circuit and associated storage capacitance.

IPR2023-00697
Theta EX2019

### 14.1.1 Charge Leakage

One of the important concerns with designing dynamic circuits is the MOSFET's off current, see Sec. 6.4.2. In Fig. 14.2 we show the simulated drain current of a 10/1 MOSFET in the 50 nm process plotted against gate-source voltage. The off current, with $V_{GS} = 0$, is taken from the plot as

$$\log I_D = -8.45 \rightarrow I_D = 3.55\ nA = I_{off} \cdot W \cdot scale \tag{14.1}$$

noting that the worst case off current occurs when the minimum channel length is used ($L = 1$). The off current can be estimated as

$$I_{off} = 7.1\ nA/\mu m \tag{14.2}$$

The off current is made up of leakage through the source/drain implant to substrate (the formed diode and the associated saturation currents), leakage around the edges of the poly at the edge of the field oxide, and by the source-to-drain leakage currents.

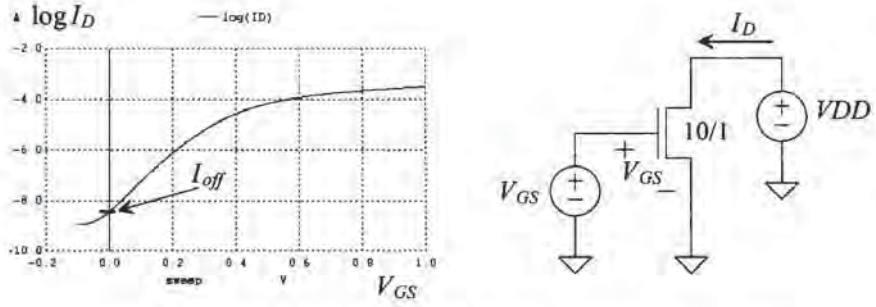



**Figure 14.2** Drain current of an NMOS device plotted from weak to strong inversion. See Fig. 6.16. PMOS netlist is found at cmosedu.com.

The rate at which the storage node seen in Fig. 14.1 discharges is given by

$$\frac{dV}{dt} = \frac{I_{off} \cdot W \cdot scale}{C_{node}} \tag{14.3}$$

The node capacitance, $C_{node}$, is the sum of the input capacitance of the inverter, the capacitance to ground of the metal or poly line connecting the inverter to the pass transistor, and the capacitance of the drain implant to substrate (the depletion capacitance). For practical applications, we assume that

$$C_{node} \approx C_{in} \text{ of the inverter} \tag{14.4}$$

**Example 14.1**

Estimate the discharge rate of the 50 fF capacitor shown Fig. 14.3.

Using Eqs. (14.1) and (14.3), we can write the capacitor's discharge rate as

$$\frac{dV}{dt} = \frac{3.55\ nA}{50\ fF} = 71\ mV/\mu s$$

IPR2023-00697
Theta EX2019

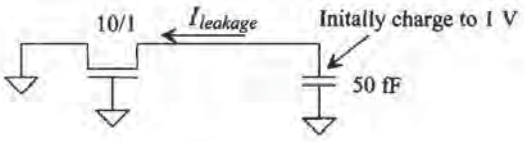



**Figure 14.3** Circuit used in Ex. 14.1.

For the capacitor to discharge to ground (from an initial voltage of 1 V) takes, roughly, $(1\ V)/(71\ mV/\mu s) = 14\ \mu s$. Figure 14.4 shows the simulation results. Note how the time it takes the output voltage to discharge to ground is twice as long as what we calculated or roughly 30 μs. This is due to the fact that as the voltage between the drain and source of the MOSFET decreases so does the off current. The reduction in the off current causes the capacitor to discharge slower. To keep the voltage across the capacitor from discharging too much we might try holding one side of the MOSFET switch at *VDD*/2. Consider the following. ■

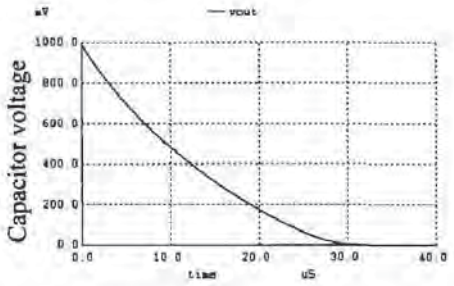



**Figure 14.4** Time it takes the capacitor to discharge due to the off current of the MOSFET in Fig. 14.3.

**Example 14.2**

Repeat Ex. 14.1 with the input tied to *VDD*/2 instead of ground.

The schematic and simulation results are seen in Fig. 14.5. The leakage current is minimized by lowering the $V_{DS}$ of the MOSFET. This is a common technique to

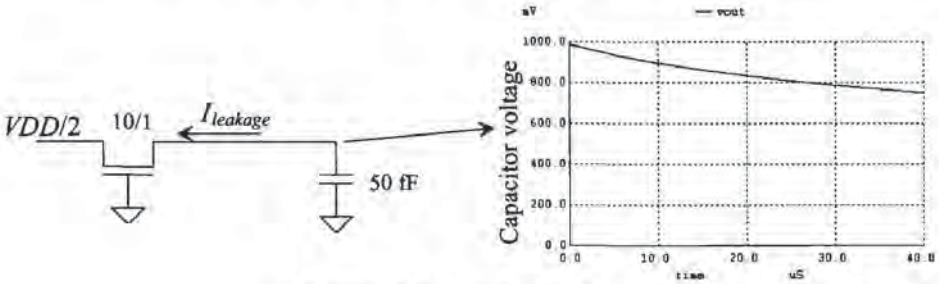



**Figure 14.5** Circuit used in Ex. 14.2.

IPR2023-00697
Theta EX2019

reduce leakage currents in DRAM (see Fig. 16.7 where the bitlines are equilibrated to *VDD*/2). Note that when the voltage across the capacitor (the storage node) is zero and the PG is off the leakage through the MOSFET causes the storage node to charge upwards (see Problem 14.2). ■

**Example 14.3**

Figure 14.6 shows a dynamic level-sensitive latch. Estimate the maximum time PG can be off before data is lost on the charge storage node. Compare the estimate to SPICE. Use the 50 nm process with 20/1 PMOS devices and 10/1 NMOS devices.

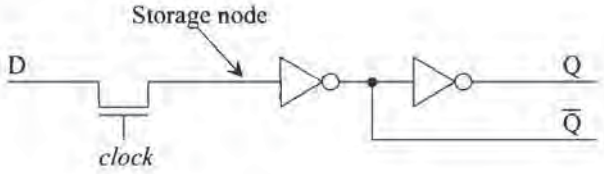



**Figure 14.6** A dynamic level-sensitive latch.

The input capacitance of the inverter, using Table 10.2, is estimated as

$$C_{in} = \frac{3}{2} \cdot (1.25 + 0.625)\, fF = 2.8\, fF$$

Using Eq. (14.3), we can estimate the rate the storage node decays as

$$\frac{dV}{dt} = \frac{3.55\, nA}{2.8\, fF} = 1.27\ V/\mu s$$

If we want, at most, 100 mV of droop in the storage node's voltage, then we would want to ensure that the PG is clocked, at the minimum, every 100 ns. Figure 14.7 shows the simulation results. Compared to our hand calculations, the discharge rate is considerably faster. This is due to our calculation of the input capacitance. The values used for $C_{in}$ were the switching input capacitance (see Figs. 10.2, 10.7, and 10.8), not the static input capacitance (see Fig. 6.4). ■

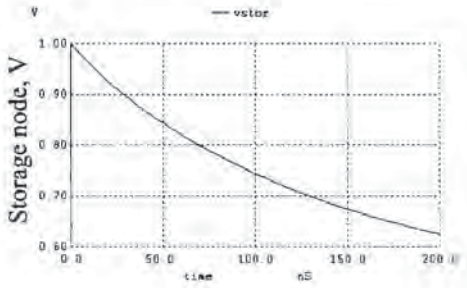



**Figure 14.7** How the voltage on the storage node varies in the dynamic latch circuit seen in Fig. 14.6.

IPR2023-00697
Theta EX2019

### 14.1.2 Simulating Dynamic Circuits

Because of the extremely small leakage currents involved, simulating dynamic circuits can be challenging. To begin, when SPICE simulates any circuit, it puts a resistor with a conductance value given by the parameter GMIN across every pn junction and MOSFET drain-to-source. The default value of GMIN is $10^{-12}$ mhos or a 1 TΩ resistor. A charge storage node at a potential of 1 V has a leakage current, due to GMIN, of 1 pA. The value of GMIN can be set, and increased, using the .OPTIONS command, at the cost of a longer or more difficult convergence time, to a smaller value, say $10^{-15}$. Some SPICE simulators are also capable of adding a resistor at every node to ground called RSHUNT (to help with convergence). If RSHUNT is $10^9$ and a node voltage is 1 V, then a current of 1 nA flows through the added RSHUNT resistor.

In practice, we can use the default values of SPICE when simulating dynamic circuits. The SPICE leakage paths (using the default values of GMIN and RSHUNT) have little effect on the estimates for the discharge times of storage nodes when simulating dynamic circuits. For example, the leakage current, for $VDD = 1$ V, due to the default value of GMIN is given by

$$I_{leakage} \approx 1\ pA = VDD \cdot GMIN \tag{14.5}$$

and

$$\frac{dV}{dt} = \frac{1\ pA}{C_{node}} = \frac{VDD \cdot GMIN}{C_{node}} \tag{14.6}$$

If $C_{node} = 5$ fF, then $dV/dt = 200 = 200\ \mu V/\mu s$. It takes approximately 5 ms for the voltage on the charge storage node to drop to ground. In all practical simulations, the leakage through a modern nanometer MOSFET is much more significant than the leakage through GMIN.

### 14.1.3 Nonoverlapping Clock Generation

Consider the string of PGs/inverters shown in Fig. 14.8. This circuit is called a dynamic shift register. When $\phi_1$ goes high, the first and third stages of the register are enabled. Data is passed from the input to point A0 and from point A1 to A2. If $\phi_2$ is low while $\phi_1$ is high, the data cannot pass from A0 to A1 and from A2 to A3. If $\phi_1$ goes low and $\phi_2$ goes high, data is passed from A0 to A1 and from A2 to A3. If both $\phi_1$ and $\phi_2$ are high at the same time, the input of the shift register and the output are connected together, which is not desirable in a shift register application. The purpose of the inverter between PGs is to restore logic levels, since the NMOS PG passes a high with a threshold voltage drop. Two inverters would be used to eliminate the logic inversion between stages. *The clocks used in this dynamic circuit must be nonoverlapping, or logically*

$$\phi_1 \cdot \phi_2 = 0 \tag{14.7}$$

There should be a period of dead time between transitions of the clock signals, labeled Δ in Fig. 14.8. The rise and fall times of the clock signals should not occur at the same time.

Since the design and layout of the dynamic shift register is straightforward, let's concentrate on the generation of clock signals, $\phi_1$ and $\phi_2$. Note that a simple logic inversion will not generate nonoverlapping clock signals. Consider the schematic of the nonoverlapping clock generator shown in Fig. 14.9. This circuit takes a clock signal and

IPR2023-00697
Theta EX2019

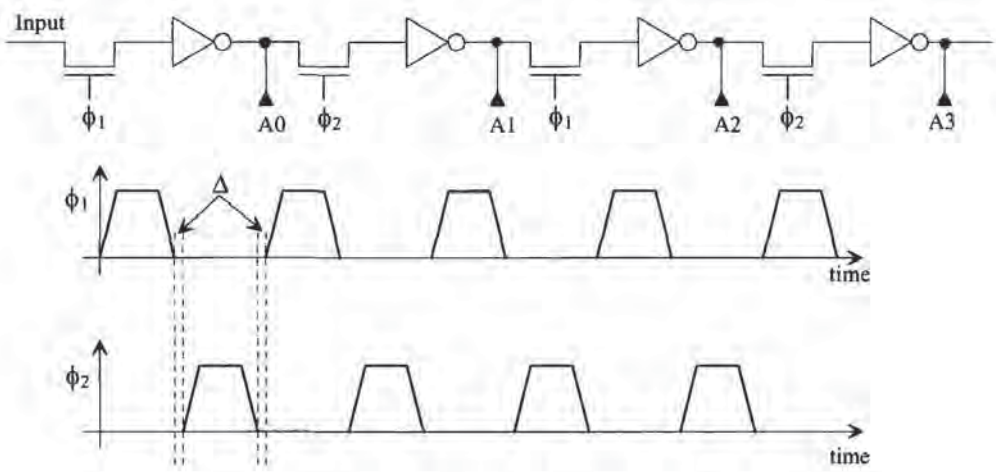



**Figure 14.8** Dynamic shift register with associated nonoverlapping clock signals.

generates a two-phase nonoverlapping clock. The amount of separation is set by the delay through the NAND gate and the two inverters on the NAND gate output. Consider the input clock going high. This forces $\phi_1$ high and $\phi_2$ low. When the input clock goes low, $\phi_1$ goes low. After $\phi_1$ goes low, $\phi_2$ can go high. When driving long transmission lines, like a wire implemented using polysilicon, where the rise time of the signals can be significant, a large number of inverters may be needed. Line drivers, a string of inverters used to drive a large capacitance, can be used as part of the delay in the nonoverlapping clock generation circuit.

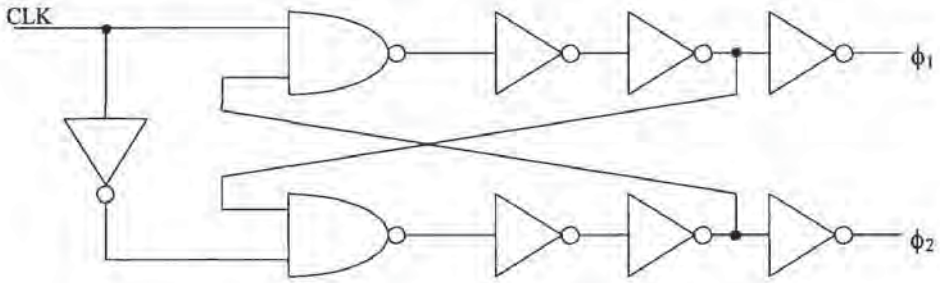



**Figure 14.9** Nonoverlapping clock generation circuit.

### 14.1.4 CMOS TG in Dynamic Circuits

The CMOS TG used as a switch to charge or discharge the node capacitance of the charge storage node is shown in Fig. 14.10. Because understanding the charging and discharging of the input capacitance of the inverter follows many of the same analysis and discussions of Ch. 13, we concentrate here on the charge leakage from the TG.

IPR2023-00697
Theta EX2019

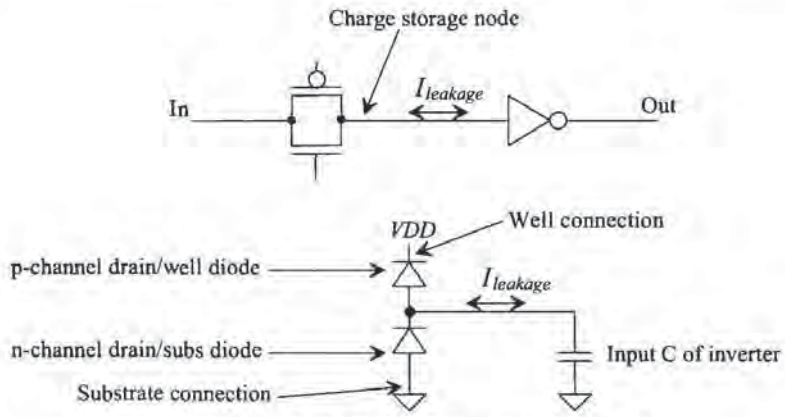



**Figure 14.10** CMOS TG used in dynamic logic.

The leakage of charge off of or onto the input capacitance of the inverter in Fig. 14.10 can be attributed to the drain-well diode of the PMOS device and the drain-substrate diode of the NMOS device used in the TG. If these leakage currents were equal, then the leakage of charge off of the storage node would be zero. Notice that unlike the NMOS PG, using a TG can result in the charge storage node leaking to *VDD* or ground, depending on the size of the drain areas and the leakage currents in each device.

## 14.2 Clocked CMOS Logic

In this section we provide a brief overview of dynamic, or clocked, CMOS logic design.

*Clocked CMOS Latch*

Figure 14.11 shows the schematic of a clocked CMOS latch. When $\phi_1$ is a low, M2 and M3 are on. The master stage simply behaves like an inverter. The D input drives, through the enabled master stage, the node N1. During this time, both M6 and M7 are off. The latch's output, Q, is a charge storage node. When $\phi_1$ goes high, M2 and M3 shut off and N1 is the charge storage node. M6 and M7 are on, and the Q output is actively driven either high or low.

*An Important Note*

Notice, in Fig. 14.11, that if the node N1 starts to move away from either *VDD* or ground, when $\phi_1$ is a high, that the slave stage can move towards its switching point. This can cause a significant current to flow from *VDD* to ground in the slave stage. Another example of where this problem can occur is seen in Fig. 14.6. If the storage node starts to wander, the inverter's input can move towards its switching point voltage. As seen in Fig. 11.4, again, a significant current can flow.

Note that the storage node's voltage can wander because of the MOSFET's off current (Fig. 14.2), a gate tunnel current leaking into the node (Fig. 16.67), or because of capacitive coupling from a noisy node (as discussed in Ex. 3.5). The issues of gate tunneling current and the reduction in parasitic capacitances present challenges when implementing dynamic logic in nanometer CMOS technologies.

IPR2023-00697
Theta EX2019

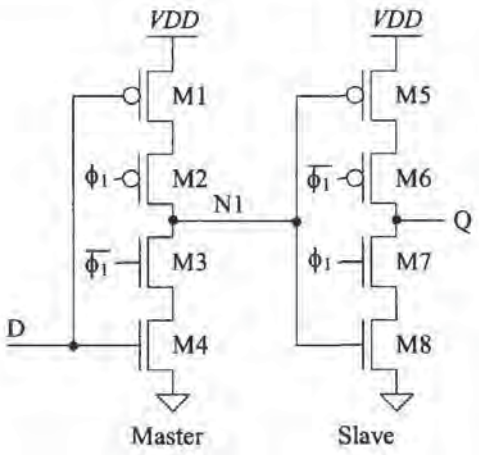



**Figure 14.11** A clocked CMOS latch. The clock signals can be generated with an RS latch so that the edges occur essentially at the same moment in time.

*PE Logic*

This section presents precharge-evaluate logic, or PE logic. Consider the three-input NAND gate shown in Fig. 14.12. The operation of this gate relies on a single clock input. When $\phi_1$ is low, the output node capacitance is charged to *VDD* through M5. During the evaluate phase, $\phi_1$ is high, M1 is on, and if A0, A1, and A2 are high, the output is pulled low. The logic output is available only when $\phi_1$ is high. The output is a logic one when $\phi_1$ is low. One disadvantage of PE logic is that the gate logic output is available part of the time and not all of the time as in the static gates.

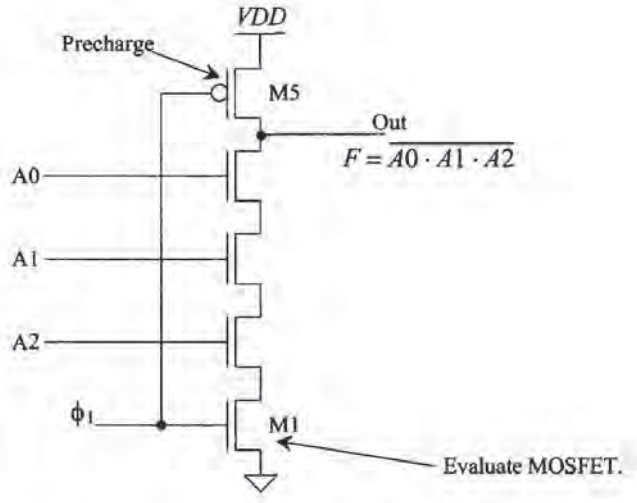



**Figure 14.12** Precharge-evaluate three-input NAND gate.

IPR2023-00697
Theta EX2019

Several important characteristics of the PE gate should be pointed out. The input capacitance of the PE gate is less than that of the static gate. Each input is connected to a single MOSFET where the static gate inputs are tied to two MOSFETs. Potentially the PE gate operates faster and dissipates less power.

The sizes of the MOSFETs used in a PE gate does not need adjusting for symmetrical switching point voltage. The absence of complementary devices and the fact that the output is pulled high during each half cycle makes the gate $V_{SP}$ meaningless. However, we may need to size the devices to attain a certain speed for a given load capacitance. If the sizes of all NMOS transistors used in Fig. 14.12 are equal, then the $t_{PHL}$ is approximately $4R_nC_{node}$ and the $t_{PLH}$ is $R_pC_{node}$, where $C_{node}$ is the total capacitance on the output node. This may include the interconnecting capacitance and the input capacitance of the next stage. Here we have neglected both the transmission line effects through a series connection of MOSFETs and the intrinsic switching speeds. A more complex logic function, $F = \overline{A0 + A1 \cdot A2 + A3 \cdot A4}$, implemented in PE logic is shown in Fig. 14.13.

*Domino Logic*

Consider the cascade of PE gates shown in Fig. 14.14. During the precharge phase of the clock, the output of each PE gate is a logic high. This high-level output is connected to the input of the next PE gate. Suppose that the logic out of the first PE gate during the evaluate phase is a low. This output will turn off any MOSFETs in the second PE gate. However, during the precharge phase, those same MOSFETs in the second PE gate will be turned on. The delay between the clock pulse going high and the valid output of the first gate will cause the second gates, output to glitch or show an invalid logic output. If we can hold the output voltage of the PE gate low instead of high, we can eliminate this race condition. Upon adding an inverter to the PE gate (Fig. 14.15), the condition for glitch-free operation is met. The PE gate with the addition of an inverter is called Domino logic. The name *Domino* comes from the fact that a gate in a series of Domino logic gates cannot change output states until the previous gate changes states. The change in output of the gates occurs similar to a series of falling dominoes. The inverter used in the Domino gate has the added advantage that it can be sized to drive large capacitive loads.

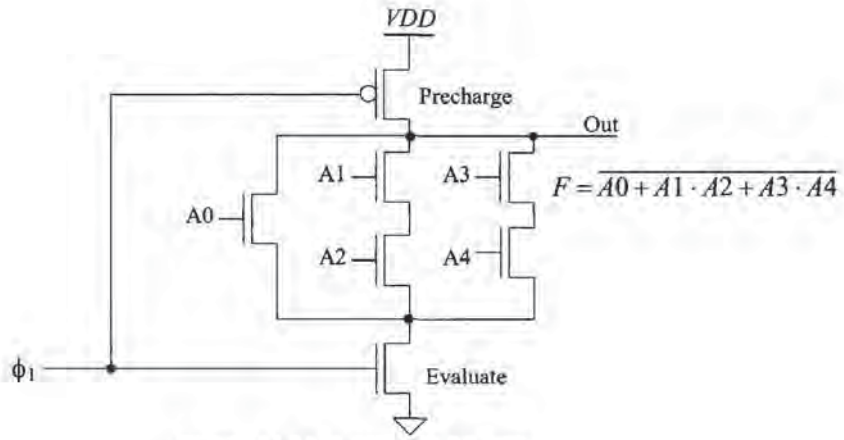



**Figure 14.13** A complex PE gate.

IPR2023-00697
Theta EX2019

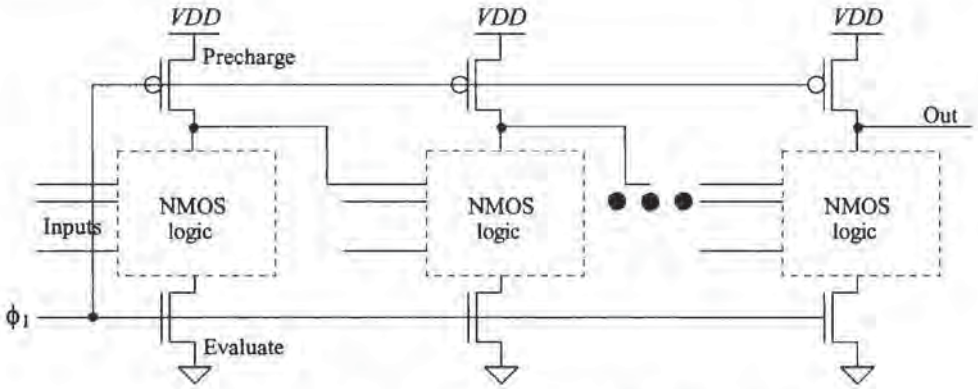



**Figure 14.14** Problems with a cascade of PE gates.

One problem does exist with this scheme, however. Referring to Fig. 14.15, note that during the precharge phase, node A is charged to *VDD*. If the NMOS logic results in a logic high on node A during the evaluate phase, then that node is at a high impedance with no direct path to *VDD* or ground. The result is charge leakage off of node A when the PE output is a logic high. The circuit of Fig. 14.16a eliminates this problem. A "keeper" PMOS device is added to help keep node A at *VDD* when the NMOS logic is off. The *W/L* of this MOSFET is small (long length and minimum width), so that it provides enough current to compensate for the leakage but not so much that the NMOS logic can't drive node A down to ground. The long-length keeper MOSFET is said to be weak. Sometimes, to implement the keeper MOSFET, two MOSFETs are used in series, as seen in Fig. 14.16b. Connecting a long-length MOSFET, and thus large area MOSFET, to the inverter's output can add unneeded load capacitance. By using a regular switch in series with a long-length PMOS device, the loading on the output of the inverter can be eliminated.

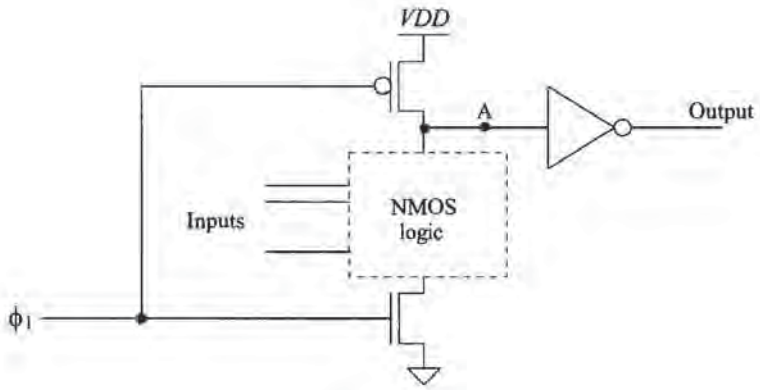



**Figure 14.15** Domino logic gate.

IPR2023-00697
Theta EX2019

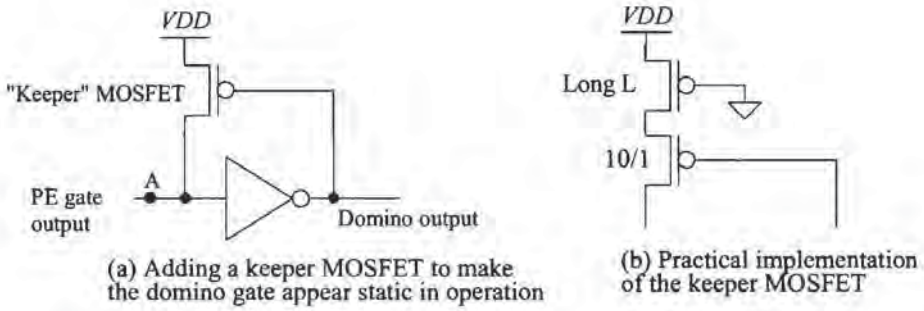



**Figure 14.16** Keeper MOSFET used to hold node A in Fig. 14.15 at VDD when PE gate output is high.

*NP Logic (Zipper Logic)*

The idea behind implementing a logic function using NP logic is shown in Fig. 14.17. Staggering NMOS and PMOS stages eliminates the need for, and delay associated with, the inverter used in Domino logic, making higher speed operation possible. A circuit that can easily be implemented in NP logic is the full adder circuit of Fig. 12.20. The NMOS section of the carry circuit is implemented in the first section of the NP logic, while the PMOS section of the sum circuit is implemented in the PMOS section of the NP logic gate.

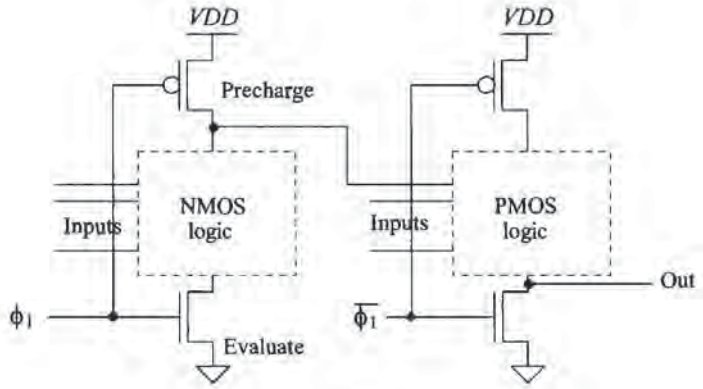



**Figure 14.17** NP logic.

*Pipelining*

The NP logic adder just described adds two 1-bit words with carry during each clock cycle. Adding two 4-bit words can use pipelining, see Fig. 14.18. The bits of the word are delayed, both on the input and output of the adder, so that all bits of the sum reach the output of the adder at the same time. Note, however, that two new 4-bit words can be input to the adder at the beginning of each clock cycle and that it takes four clock cycles

IPR2023-00697
Theta EX2019

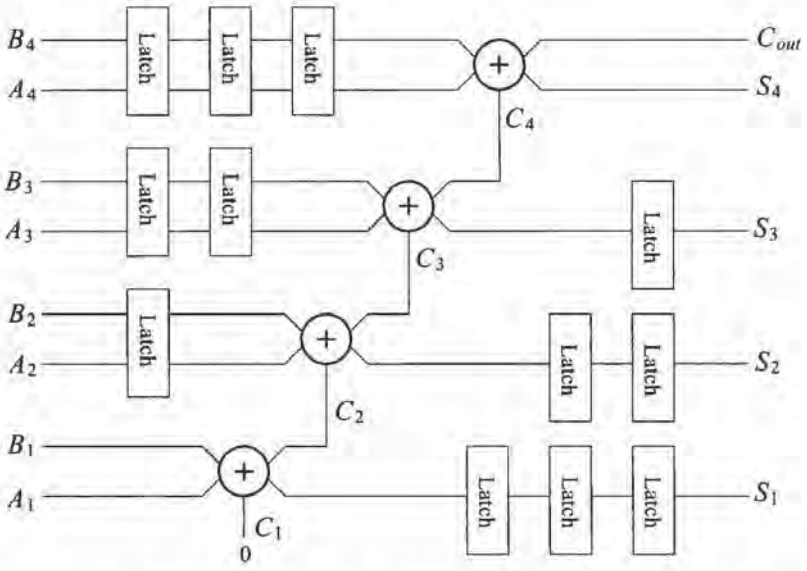



**Figure 14.18** A pipelined adder. The latches (clocked) behave as delay elements.

to finish the addition of the two words. If this circuit were dedicated to continually performing the addition of two words, we could input the words at a very fast rate. However, since performing a single addition requires four clock cycles, applications of pipelining where two numbers are not added continuously can result in longer delay-times.

## ADDITIONAL READING

[1] K. Bernstein, K. M. Carrig, C. M. Durham, P. R. Hansen, D. Hogenmiller, E. J. Nowak, and N. J. Rohrer, *High Speed CMOS Design Styles*, Springer, 1999. ISBN 978-0792382201.

[2] J. Yuen and C. Svensson, "New Single-Clock CMOS Latches and Flipflops with Improved Speed and Power Savings," *IEEE Journal of Solid-State Circuits,* vol. 32, no. 1, pp. 62–69, 1997.

[3] M. I. Elmasry, *Digital MOS Integrated Circuits II,* IEEE Press, 1992. ISBN 0-87942-275-0, IEEE order number: PC0269-1.

[4] J. P. Uyemura, *Circuit Design for Digital CMOS VLSI,* Kluwer Academic Publishers, 1992.

[5] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI-Design Techniques for Analog and Digital Circuits,* McGraw-Hill Publishing Co., 1990. ISBN 0-07-023253-9.

## PROBLEMS

Unless otherwise stated, use the 50 nm, short-channel process.

**14.1** Regenerate Fig. 14.2 for the PMOS device. From the results, determine the PMOS's $I_{off}$.

IPR2023-00697
Theta EX2019

**14.2** Repeat Ex. 14.2 if the storage node is a logic 0 (ground). Explain why the charge storage node charges up. What would happen if the PG's input were held at *VDD* instead of *VDD*/2.

**14.3** Comment on the usefulness of dynamic logic in our 50 nm CMOS process-based on the results given in Ex. 14.3 with a clock frequency of 10 MHz.

**14.4** Using the circuit in Fig. 14.6 and the SPICE simulation, show how the current drawn from *VDD*, by the inverters, changes with time. Do you see any concerns? If so, what?

**14.5** Simulate the operation of the nonoverlapping clock generator circuit in Fig. 14.9. Assume that the input clock signal is running at 100 MHz. Show how both $\phi_1$ and $\phi_2$ are nonoverlapping.

**14.6** Simulate the operation of the clocked CMOS latch shown in Fig. 14.11.

**14.7** Design and simulate the operation of a PE gate that will implement the logical function $F = \overline{ABCD + E}$.

**14.8** If the PE gate shown in Fig. 14.13 drives a 50 fF capacitor, estimate the worst-case $t_{PHL}$. Use a 20/1 PMOS and a 10/1 NMOS.

**14.9** Implement an XOR gate using Domino logic. Simulate the operation of the resulting implementation.

**14.10** The circuit shown in Fig. 14.19 results from the implementation of a high-speed adder cell (1-bit). What type of logic was used to implement this circuit? Using timing diagrams, describe the operation of the circuit.

**14.11** Discuss the design of a 2-bit adder using the adder cell of Fig. 14.19 If a clock, running at 200 MHz, is used with the 2-bit adder, how long will it take to add two words? How long will it take if the word size is increased to 32 bits?

**14.12** Sketch the implementation of an NP logic half adder cell.

**14.13** Design (sketch the schematic of ) a full adder circuit using PE logic.

**14.14** Simulate the operation of the circuit designed in Problem 14.10.

**14.15** Show that the dynamic circuit shown in Fig. 14.20 is an edge-triggered flip-flop [2]. Note that a single-phase clock signal is used.

IPR2023-00697
Theta EX2019

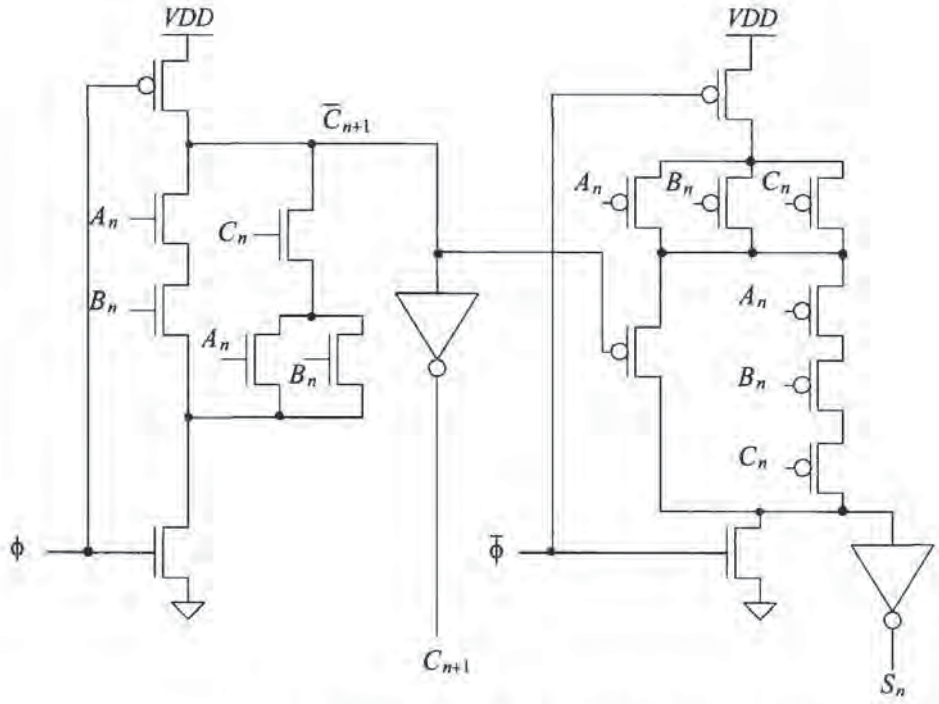



**Figure 14.19** A high speed adder cell. See Problem 14.10.

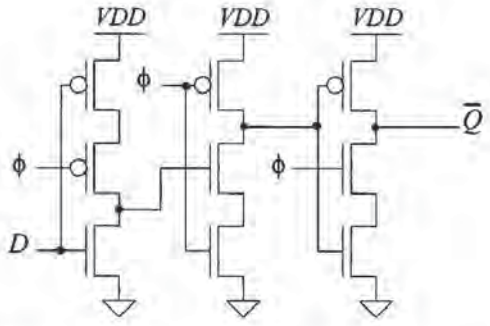



**Figure 14.20** A true-single phase clocked FF, see Problem 14.15.

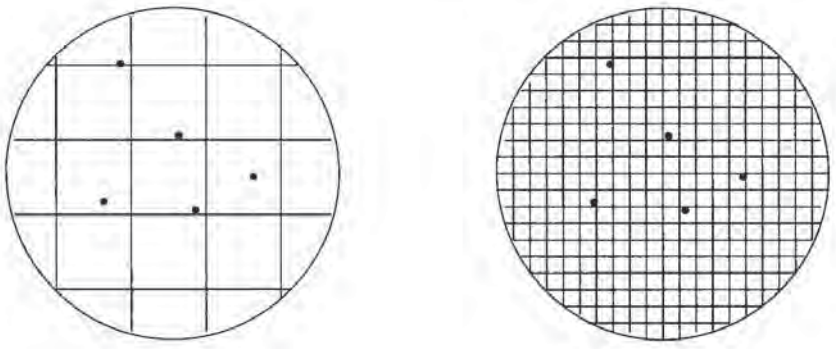
IPR2023-00697
Theta EX2019

Chapter
15

# VLSI Layout Examples

In the past chapters we have concentrated on basic logic-gate design and layout. In this chapter we discuss the implementation of logic functions on a chip where the size and organization of the layouts are important. The number of MOSFETs on a chip, depending on the application, can range from tens (an op-amp) to more than hundreds of millions (a 256 Mbit DRAM). Designs where thousands of MOSFETs or more are integrated on a single die are termed *very-large-scale-integration* (VLSI) designs.

To help us understand why chip size is important, examine Fig. 15.1. The dark dots indicate defects and thus bad chips. Figure 15.1a shows a wafer with nine full die. The partial die around the edge of the wafer are wasted. Five of the nine die do not contain a defect and thus can be packaged and sold. Next consider a reduction in the die size (Fig. 15.1b). We are assuming each die, whether discussing the die of Fig. 15.1a or b, performs the same function. This reduction can be the result of having better layout (resulting in a smaller layout area) or fabricating the chips in a process with smaller



(a) (b)

**Figure 15.1** Defect density effects on yield.

IPR2023-00697
Theta EX2019

device dimensions (e.g., going from a 130 nm process to a 50 nm process). The total number of die lost (see Fig. 15.1b), due to defects is five; however, the number of good die is significantly larger than the five good die of Fig. 15.1a. The yield (number of good die/total number of die on the wafer) is increased with smaller die size. The result is more die/wafer available for sale. Another benefit of reducing die size comes from the realization that processing costs per wafer are relatively constant. Increasing the number of die on a wafer decreases the cost per die.

## 15.1 Chip Layout

VLSI designs can be implemented using many different techniques including gate-arrays, standard-cells, and full-custom design. Because designs based on gate-arrays are used, in general, where low volume and fast turnaround time are required and the chip designers need know little to nothing[1] about the actual implementation of the CMOS circuits, we will concentrate on full-custom design and design using standard cells

*Regularity*

An important consideration when implementing a VLSI chip design is regularity. The layout should be an orderly arrangement of cells. Toward this goal, the first step in designing a chip is drawing up a chip (or section of the chip) floor plan. Figure 15.2 shows a simple floor plan for an adder data-path. This floor plan can be added to the floor plan of an overall chip, which includes output buffers, control logic, and memory. At this point, we may ask the question, "How do we determine the size of the blocks in Fig. 15.2?" The answer to this question leads us into the design and layout of the cells used to implement each of the logic blocks in Fig. 15.2.

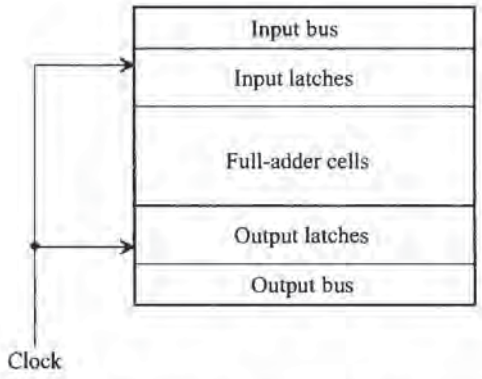



**Figure 15.2** Floor plan for an adder.

[1] At many universities, design using a hardware description language (HDL) with field-programmable-gate arrays (FPGAs) is discussed in the first (or perhaps second) course on digital systems design.

IPR2023-00697
Theta EX2019

*Standard-Cell Examples*

Standard cells are layouts of logic elements including gates, flip-flops, and ALU functions that are available in a cell library for use in the design of a chip. *Custom design* refers to the design of cells or standard cells using MOSFETs at the lowest level. *Standard-cell design* refers to design using standard cells; that is, the designer connects wires between standard cells to create a circuit or system. The difference between the two types of design can be illustrated using a printed circuit board-level analogy. A standard-cell design is analogous to designing with packaged parts. The design is accomplished by connecting wires between the pins of the packaged parts. Custom design is analogous to designing the "insides" of the packaged parts themselves

Figure 15.3 shows an example of an inverter. In addition to keeping the layout size as small as possible, an important consideration when laying out a standard cell is the routing of signals. Keeping this in mind, we can state the following general guidelines for standard-cell design:

1. Cell inputs and outputs should be available, at the same relative horizontal distance, on the top and bottom of the cell.
2. Horizontal runs of metal are used to supply power and ground to the cell, a.k.a., power and ground buses. Also, well and substrate tie downs should be under these buses.

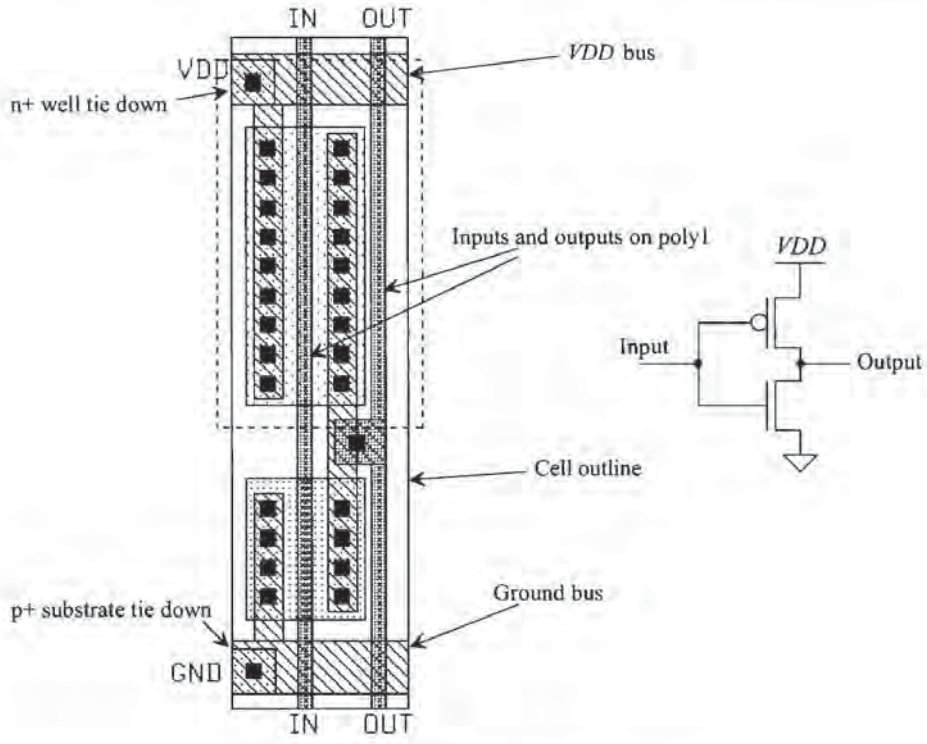



**Figure 15.3** Standard cell layout of an inverter.

IPR2023-00697
Theta EX2019

3. The height of the cells should be a constant, so that when the standard cells are placed end to end the power and ground buses line up. The width of the cell should be as narrow as the layout will allow. However, the absolute width is not important and can be increased as needed.

4. The layout should be labeled to indicate power, ground, and input and output connections. Also, an outline of the cell, useful in alignment, should be added to the cell layout.

Figure 15.4 illustrates the connection of standard cells to a bus. Note that poly, which runs vertically, can cross the metal1 lines, which run horizontally without making contact. This fact is used to route signals and interconnect standard cells in a VLSI design. Also, in this figure, note how the two inverter standard cells are placed end to end. The result is that power and ground are automatically routed to each cell.

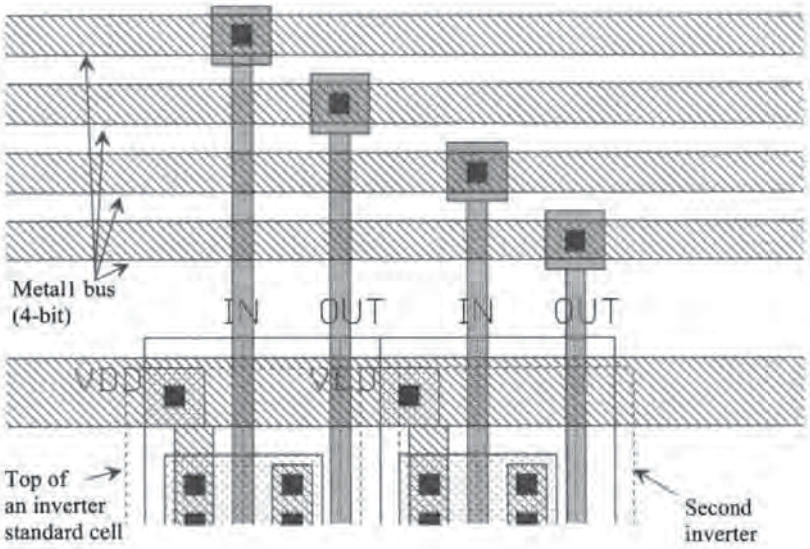



**Figure 15.4** Connection of two inverter standard cells to a bus.

Other examples of static standard cells are shown in Fig. 15.5. A double inverter standard cell is shown in Fig. 15.5a, while NAND, NOR, and transmission gate standard cells are shown in Figs. 15.5b, c, and d.

Figure 15.6 shows the layout of a NAND-based SR latch. This layout differs from the others we have discussed. In all layouts discussed so far metal1 and contacts are adjacent to the gate poly. Also, the gate poly has been laid down without bends. The expanded view of a PMOS device used in the SR latch is shown in Fig. 15.7. Keeping in mind that whenever poly crosses active (n+ or p+), a MOSFET is formed, we see that the source of the MOSFET is connected to metal through two contacts, while the p+ implant forms a resistive connection to metal1 along the remainder of the device. The layout size,

IPR2023-00697
Theta EX2019

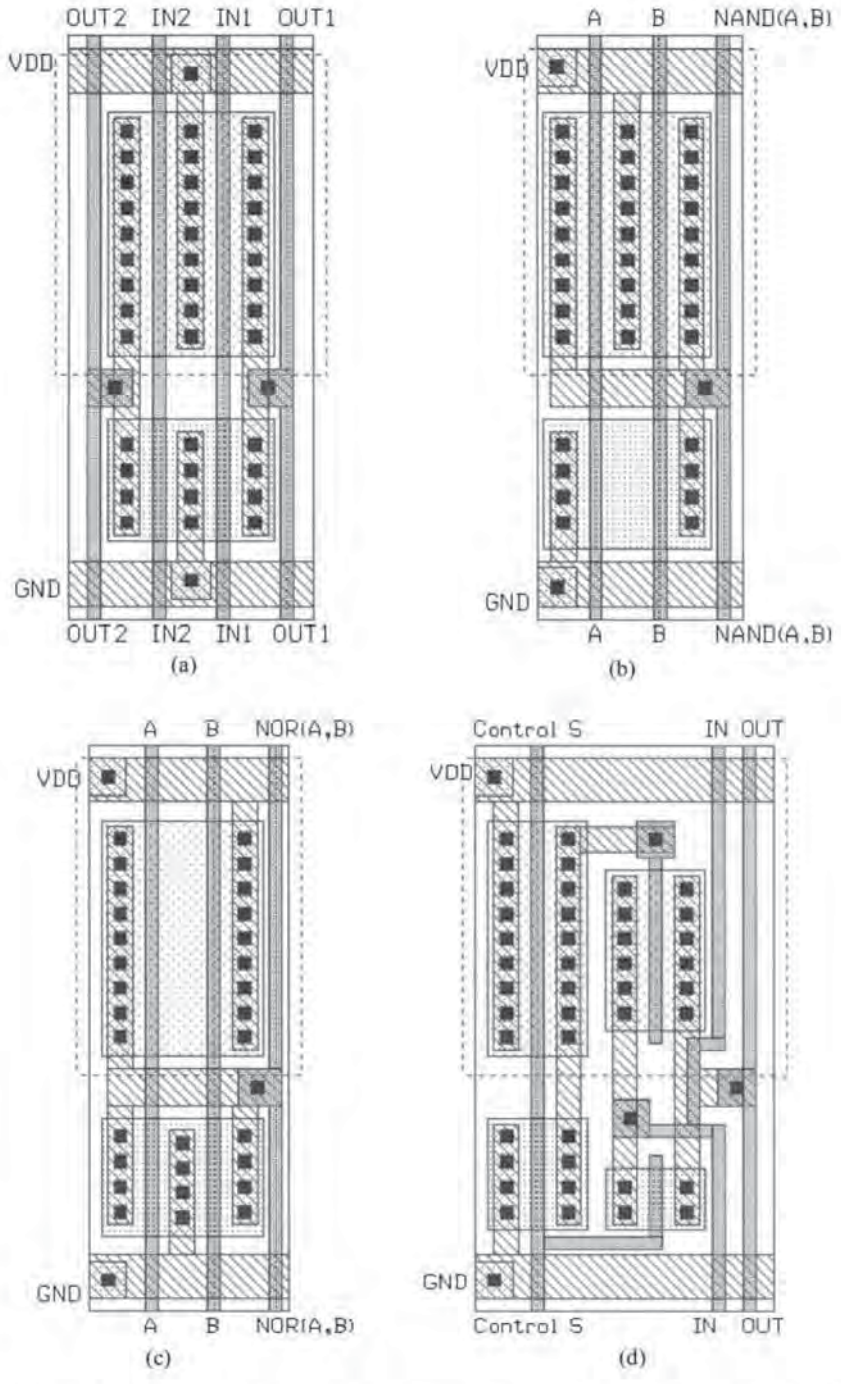



**Figure 15.5** (a) Double inverter, (b) two-input NAND, (c) two-input NOR, and (d) transmission gate.

IPR2023-00697
Theta EX2019

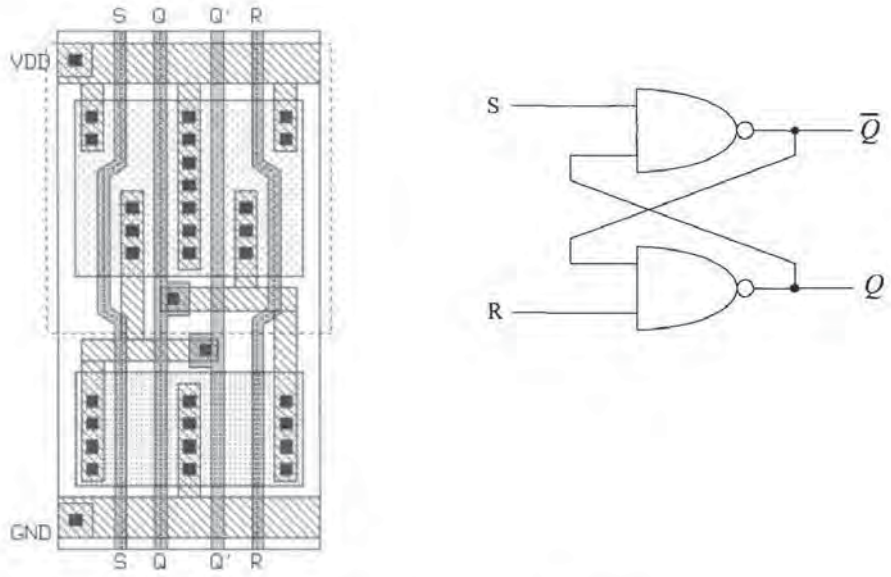



**Figure 15.6** SR latch using NAND gates.

in this case the width of the standard cell, can be reduced using this technique. Because of the bend in the gate, the width of this MOSFET is longer than the adjacent MOSFET. This additional width is of little importance and has little effect on the DC and transient properties of the gate. Figure 15.8 shows the NOR implementation of an SR latch.

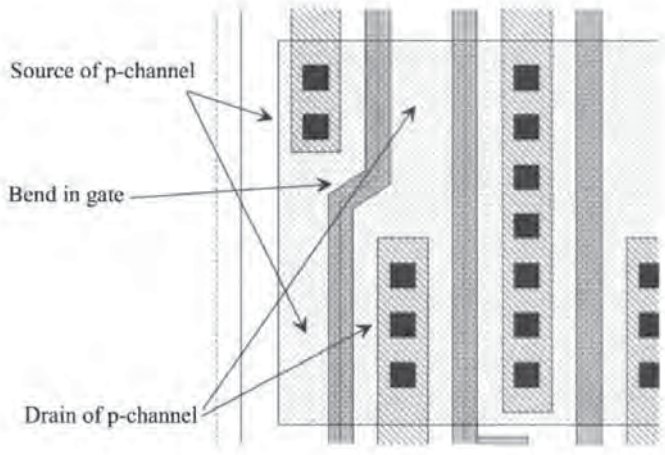



**Figure 15.7** Section of the layout shown in Fig. 15.6.

IPR2023-00697
Theta EX2019

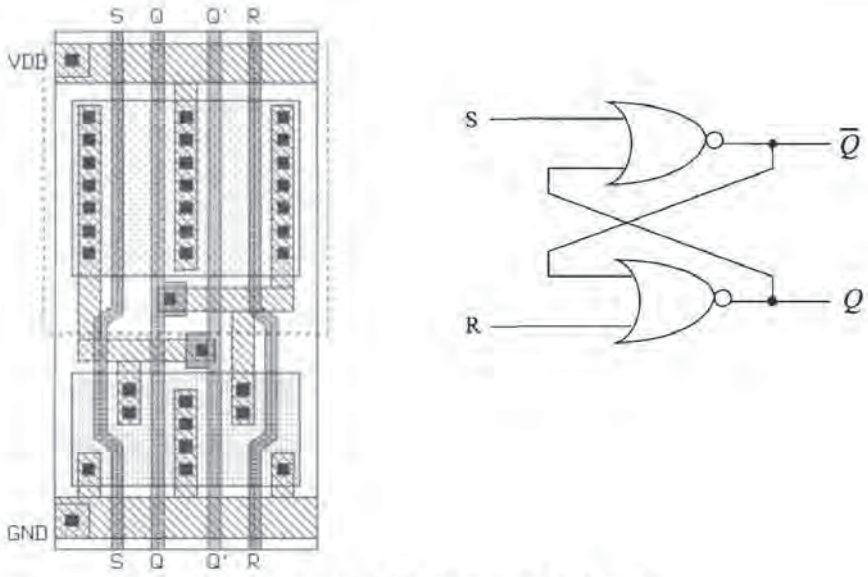



**Figure 15.8** SR latch using NOR gates.

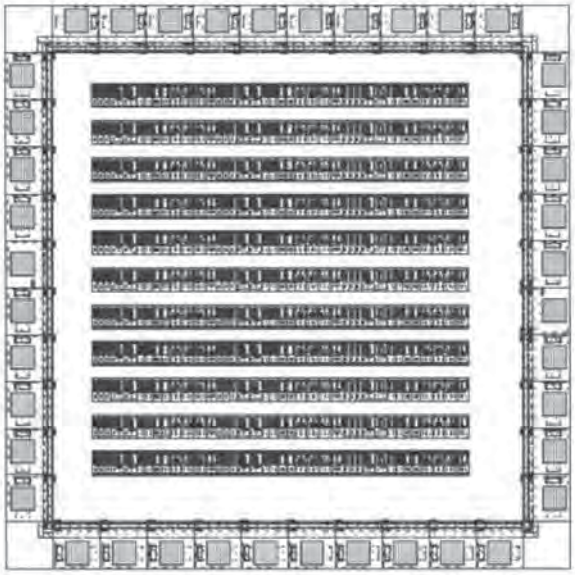
*Power and Ground Connections*

Many of the problems encountered when designing a chip can be related to the distribution of power and ground. When power and ground are not distributed properly, noise can be coupled from one circuit onto the power and ground conductors and injected into some other circuit.

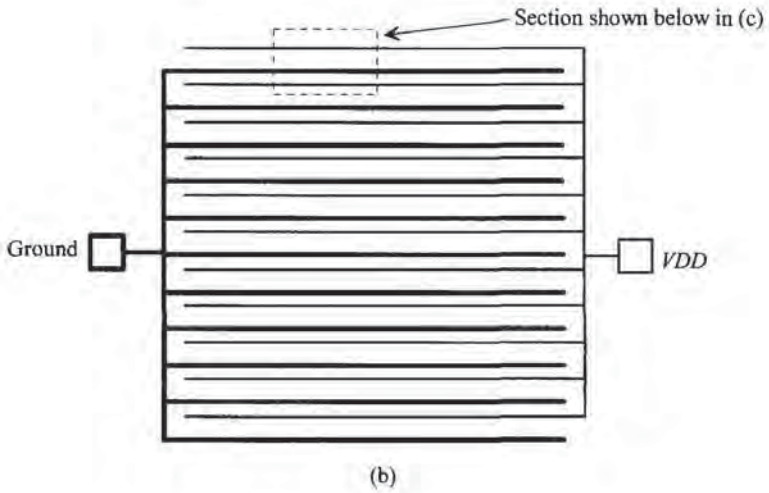
Consider the placement of standard cells in a padframe shown in Fig. 15.9a, without connections to power and ground. Approximately 600 standard cells are shown in this figure. The space between the rows of standard cells is used for the routing of signals. A line drawing of a possible power and ground busing architecture is shown in Fig. 15.9b. Consider the section of bus shown in Fig. 15.9c. Wire A connects the standard cells in the top row to *VDD*, while wire B is used for the connection to ground. Ideally, the current supplied on A (*VDD*) is returned on B (ground). In practice, there is coupling between conductors B and C, which gives rise to an unwanted signal (noise) on either conductor. This coupling can be reduced by increasing the space between B and C, which reduces the inductive and capacitive coupling between the conductors. Another solution is to increase the capacitance between A and B. A standard cell decoupling capacitor (Fig. 15.10) can be used toward this goal. The capacitor is placed in the middle of a standard-cell row. Also, the AC resistive drop effects discussed in Ch. 3 (see Fig. 3.17 and the associated discussion) are greatly reduced by including this capacitor.

Coupling is a problem on signal buses as well. Figure 15.11 shows a simple scheme to reduce coupling. The length of a section, where two wires are adjacent, is reduced by routing the wire to other locations at varying distances along the bus. The inductive or capacitive coupling between two conductors is directly related to the length of the wire runs.

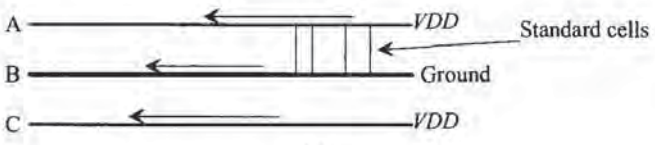
IPR2023-00697
Theta EX2019



(a)




(b)




(c)

**Figure 15.9** Connection of power and ground to standard cells.

IPR2023-00697
Theta EX2019

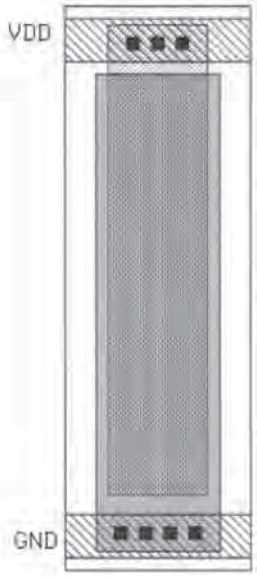



**Figure 15.10** Decoupling capacitor.

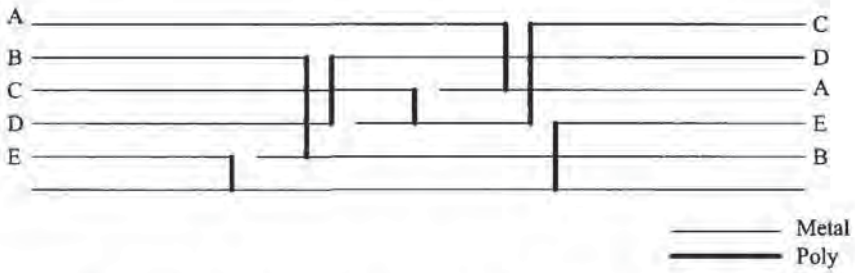



**Figure 15.11** Busing structure used to decrease signal coupling.

*An Adder Example*

As another example, let's consider the implementation of a 4-bit adder. (The floorplan for this adder was shown in Fig. 15.2.) The first components that must be designed are the input and output latches. Figure 15.12a shows the schematic of the latches. This latch is the level-sensitive type discussed in Ch. 13. When CLK is high, the output, $Q$, changes states with the input, $D$. The inverter, I4, provides positive feedback and is sized with a small $W/L$ ratio so that I1 does not need to supply a large amount of DC current to force the latch to change states. The layout of the latch is shown in Fig. 15.12b. The layout size and the size of the MOSFETs in these examples may be larger than normal to make it easier to understand and view the layouts.

IPR2023-00697
Theta EX2019

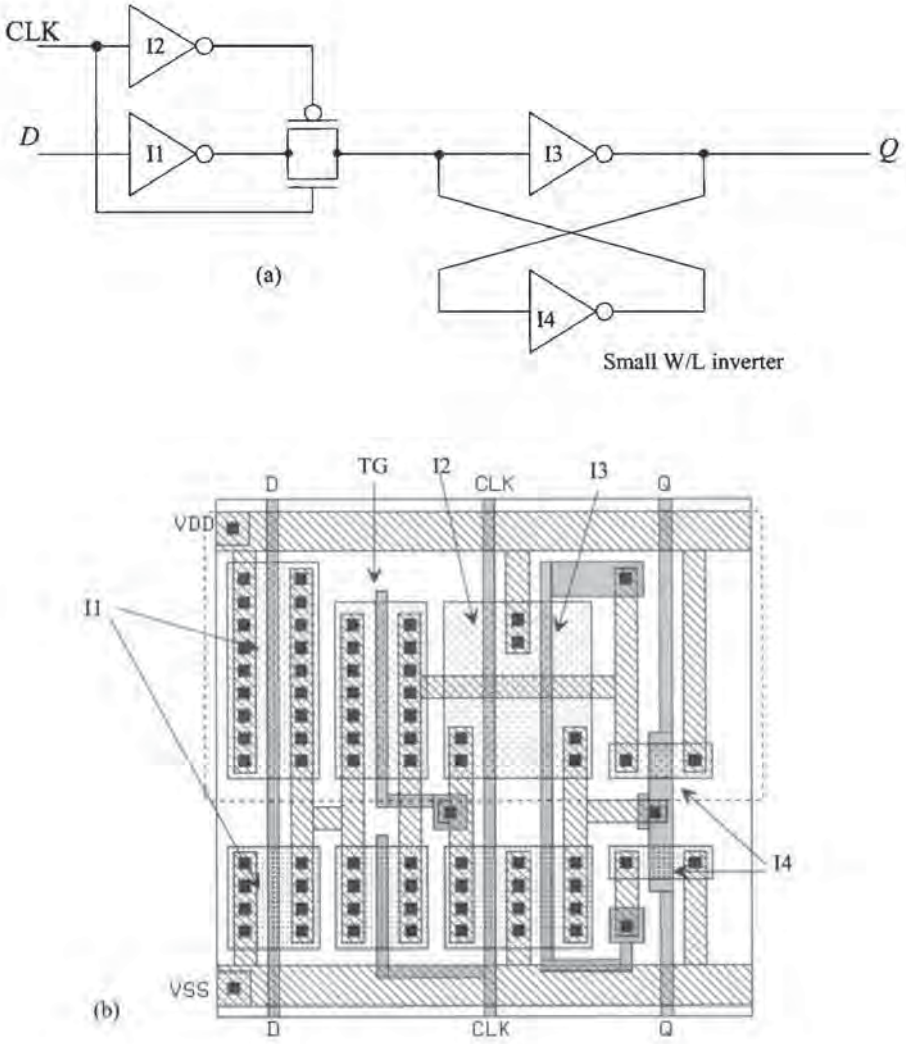



**Figure 15.12** Schematic and layout of a latch.

The layout of the static adder is shown in Fig. 15.13. This is the implementation, using near minimum-size MOSFETs, of an AOI (and-or-inverter) static adder. Both the carry-out and sum-out logic functions are implemented in this cell.

The complete layout of the adder is shown in Fig. 15.14. The two 4-bit words, Word-A and Word-B, are input to the adder on the input bus. These data are clocked into the input latch when CLK is high, while the results of the addition are clocked into the output latch when CLK is low. The inverter standard cell of Fig. 15.3 is placed at the end of the output latches and generates $\overline{\text{CLK}}$ for use in the output latches. The inputs and outputs of the adder cells are run on poly because of the short distances involved. The carry-in of the adders is connected to ground, as shown in the figure.

IPR2023-00697
Theta EX2019

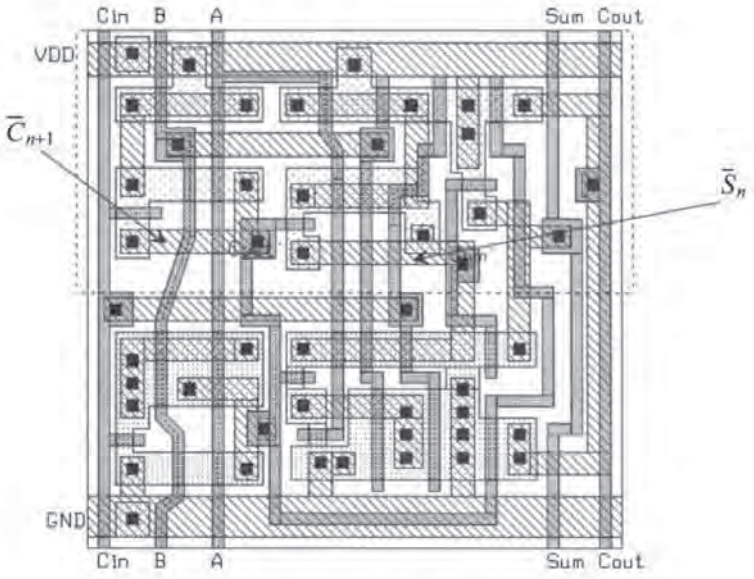



**Figure 15.13** Layout of an AOI static adder.

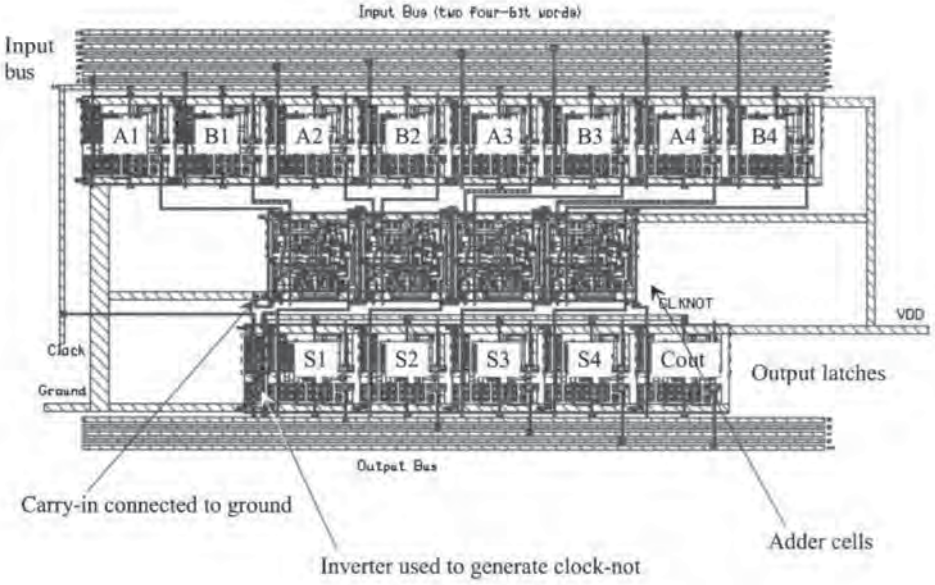



**Figure 15.14** Layout of the complete adder.

IPR2023-00697
Theta EX2019

*A 4-to-1 MUX/DEMUX*

The layout of a 4-to-1 MUX/DEMUX is shown in Fig. 15.15 (based on the use of NMOS pass gates). Notice that the (required) p+ substrate connections are not shown. This layout is different from the layouts discussed so far as the circuit does not require power and ground connections and the input/output signals are connected on n+. The select signals are supplied to the circuit on metal1 at the top of the layout. For A to be connected to the output, the signals S1 and S2 should be high. For a large MUX, the propagation delay through the n+ should be considered.

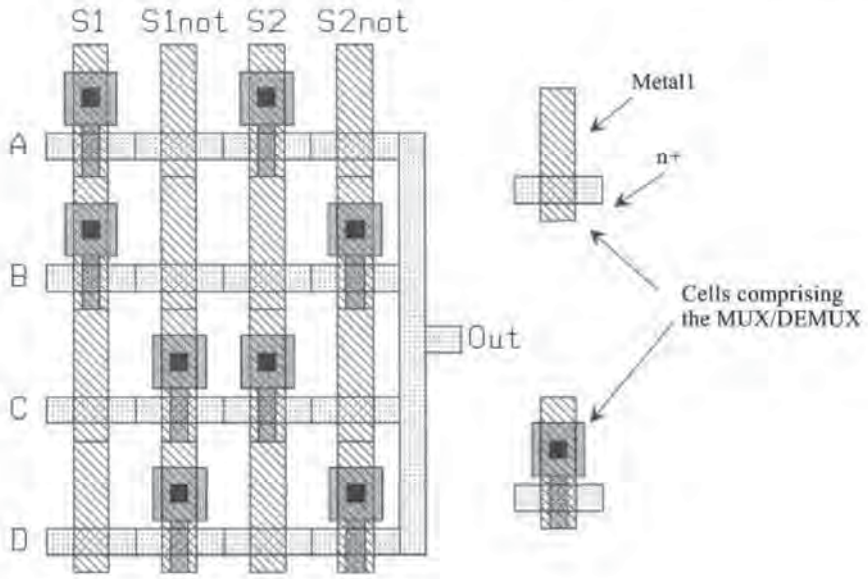



**Figure 15.15** Layout of a 4-to-1 MUX/DEMUX.

## 15.2 Layout Steps *by Dean Moriarty*

The steps involved in rendering a schematic diagram into its physical layout are plan, place, connect, polish, and verify. Let's illustrate each of these steps in some detail.

*Planning and Stick Diagrams*

The planning steps start with paper and pencil. Colored pencils are useful for distinguishing one object from another. You can use graph paper to help achieve a sense of proportion in the cell plan but don't get too bogged down in the details of design rules or line widths at this point; we just want to come up with a general plan. A "stick diagram" is a paper and pencil tool that you can use to plan the layout of a cell. The stick diagram resembles the actual layout but uses "sticks" or lines to represent the devices and conductors. When used thoughtfully, it can reveal any special hook-up problems early in the layout, and you can then resolve them without wasting any time.

IPR2023-00697
Theta EX2019

Figure 15.16a shows the schematic of an inverter. To realize the layout of this circuit, it is first necessary to define the direction and metallization of the power supply, ground, input, and output. Since the standard-cell template was discussed earlier (see also Fig. 4.15 and the associated discussion), we'll use it. Power and ground run horizontally and provide the substrate and well connections. The input and output are accessible from the top or bottom of the cell and will be in metal2 running vertically. Figure 15.16b shows the completed stick diagram. Note the use of "X" and "O" to denote contacts and vias, respectively. The stick diagram should be compared to the resulting layout of Fig. 15.17.

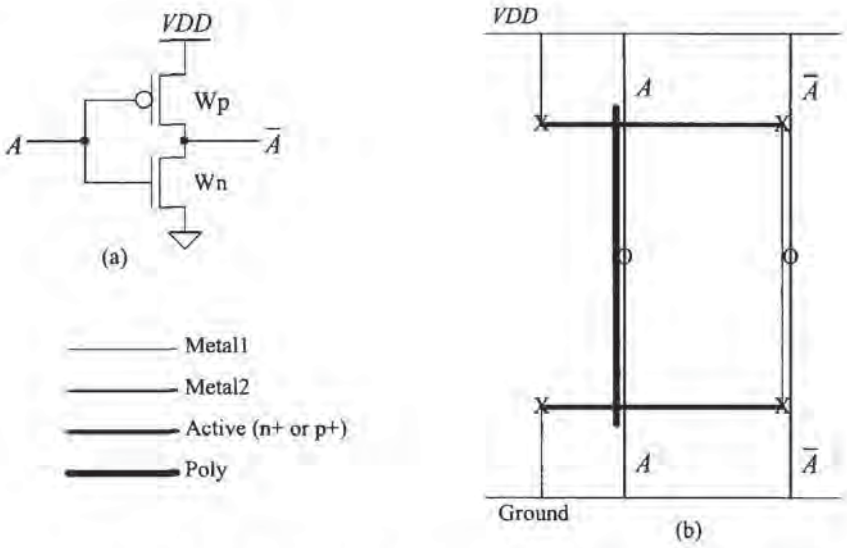



**Figure 15.16** (a) Inverter and (b) stick diagram used for layout.

Suppose the device widths of the inverter circuit in Figure 15.16a were quadrupled. Furthermore, let's assume that the maximum recommended poly1 gate width is 20 (due to the sheet resistance of the poly) and that exceeding that maximum could introduce significant unwanted RC delays. Let's also suppose that we are to optimize the layout for size and speed (as most digital circuits are). To meet these criteria, it will be necessary to split transistors M1 and M2 in half and lay them out as two parallel "stripes." Figures 15.18a–d show the schematics, stick diagram, and layout for this scenario. The output node (drain of M1 and M2) is shared between the stripes so as to minimize the output capacitance. Taking the output in metal2 also helps in this regard. Notice that the stick diagram for this circuit looks like the previous inverter plus its mirror image along the output node. Also observe that the layout of this inverter is mirrored, as shown in the stick diagram. This is a common layout technique.

Figure 15.19 shows stick diagrams and layouts for two more common circuits: the two-input NAND and the two-input NOR. Compare the stick diagrams of Figs. 15.19a and c to the layouts of Figs. 15.19b and d. Observe that the output nodes share the active area just as in the previous example. Also note that the spacing between the gates of the series-connected devices is minimal.

IPR2023-00697
Theta EX2019

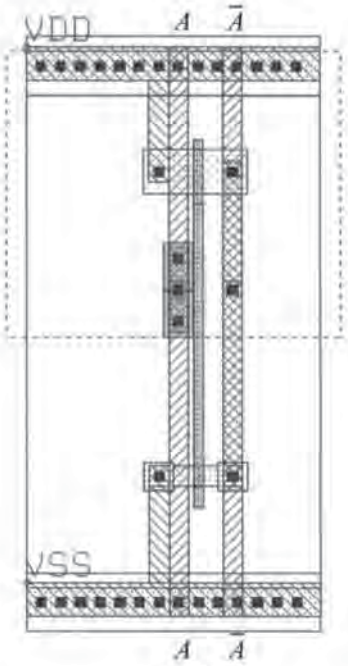


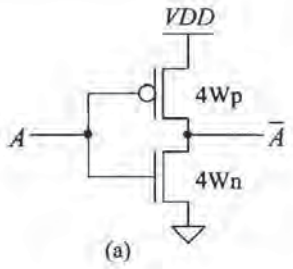

**Figure 15.17** Layout of the inverter shown in Fig. 15.16.

Take another look at the two circuits from a geometrical rather than an electrical viewpoint. Compare the NAND gate layout to the NOR gate layout. Each can be created from the other by simply "flipping" the metal and poly connections about the x-axis.

*Device Placement*

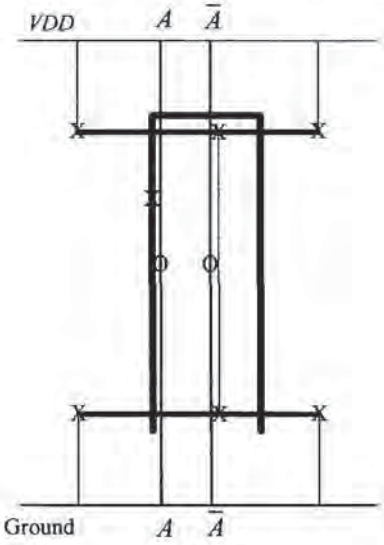
Figure 15.20 shows the schematic of a dynamic register cell, while Figs. 15.21 a–c show the stick diagrams and layout for a dynamic register. Compare the schematic of Fig. 15.20 to the stick diagram of Fig. 15.21a. We have labeled this stick diagram "preliminary" for reasons that will soon become apparent. Notice that there is a break or gap in the active area, which will form our NMOS devices. Also note that the clock signals CLK and $\overline{\text{CLK}}$ must be "cross-connected" from one side of the layout to the other. We don't have to think this through very far to notice that, with this placement of devices, hooking up the clock signals is going to be very difficult. Now look at the stick diagram shown in Fig 15.21b. Notice that we have rearranged the devices so that the active area is a continuous unbroken line. Normally, this "unbroken line" approach to device placement is preferred. It usually results in the most workable device placement. We say "usually" because at times your layout has to fit in an area defined by other blocks around it and you have no control over it. Also observe from Fig. 15.21b that the clock signal hook-up is more straightforward. Compare this stick diagram to the layout of Fig. 15.22c. Obviously, the device sizes used for this circuit are not practical; its purpose is merely to illustrate a layout concept. We can also see that the stick diagram is a useful tool throughout the layout process.

IPR2023-00697
Theta EX2019

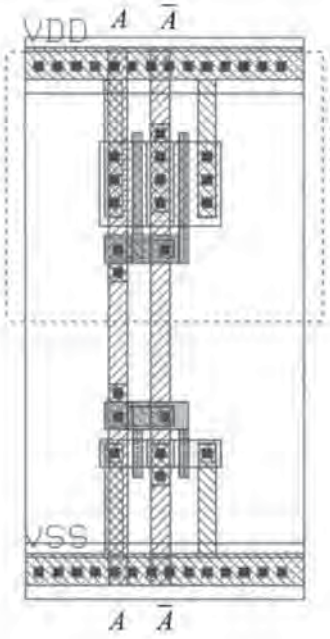



(a)

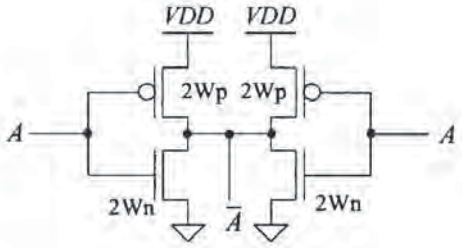



(b)




(c)




(d) Doubling the widths of the MOSFETs used in an inverter.

**Figure 15.18** (a) Inverter, (b) stick diagram used for layout, (c) layout, and (d) equivalent schematic.

IPR2023-00697
Theta EX2019

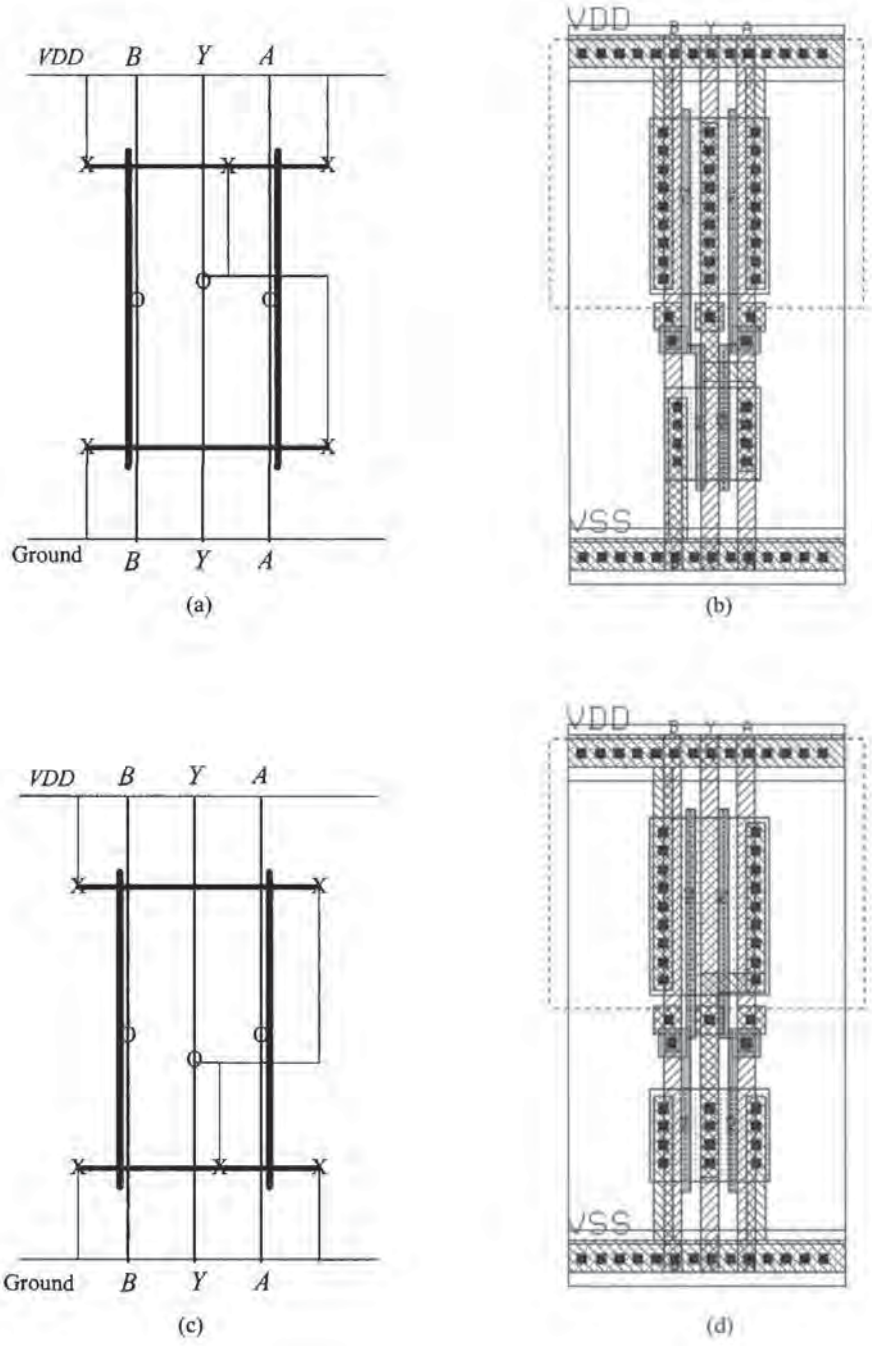



**Figure 15.19** (a) NAND stick diagram, (b) layout, (c) NOR stick diagram, and (d) layout.

IPR2023-00697
Theta EX2019

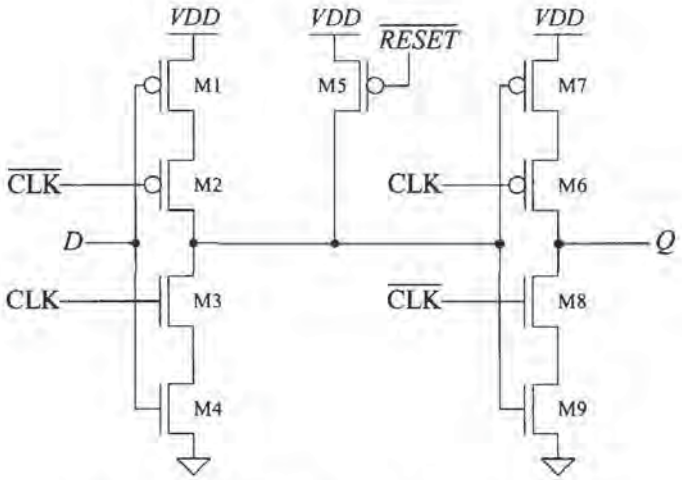



**Figure 15.20** Schematic of a dynamic register cell.

*Polish*

After your layout is basically finished, it is time to step back and take a look at it from a purely aesthetic point of view. Is it pleasing to the eye? Is the hook-up as straightforward as possible, or is it "busy" and hard to follow? Are the spaces between poly gates and contacts minimum? What about the space between diffusions? Are there enough contacts? Did you share all of the source and drain implants that can be shared? Are there sufficient well and substrate ties? If you have planned well and followed the plan described here, you shouldn't run into too many problems.

*Standard Cells Versus Full-Custom Layout*

The standard cell approach to physical design usually dictates a fixed cell height and variable width when implementing the circuit. Furthermore, standard cells are designed to abut on two sides, usually left and right, and that abutment scheme must be quite regular so that any cell can reside next to any other cell without creating a design rule violation. The standard cell approach to layout is very useful and is always an excellent place to start. However, in the real world, area on a wafer translates directly into profit and loss (money). Wafer costs are relatively fixed whether they're blank or as tightly packed with circuitry as possible. Therefore, it follows that we want a layout that is as small as possible so that there can be as many die per wafer as possible. These are the economics of the situation. There are also technical advantages to be gained from having as small a layout as possible: interconnecting wires can be as short as possible, thereby reducing parasitic loading and crosstalk effects.

Figure 15.22 shows a typical standard-cell block that has been placed and routed by an automatic tool. Most of the individual cells have been omitted for clarity. Notice the interconnect channels between the rows of standard cells. Power, ground, and clock signal trunks run vertically to both sides of the block by means of a special cell called an "end cap." Cell rows are connected to power and ground through horizontal buses that are

IPR2023-00697
Theta EX2019

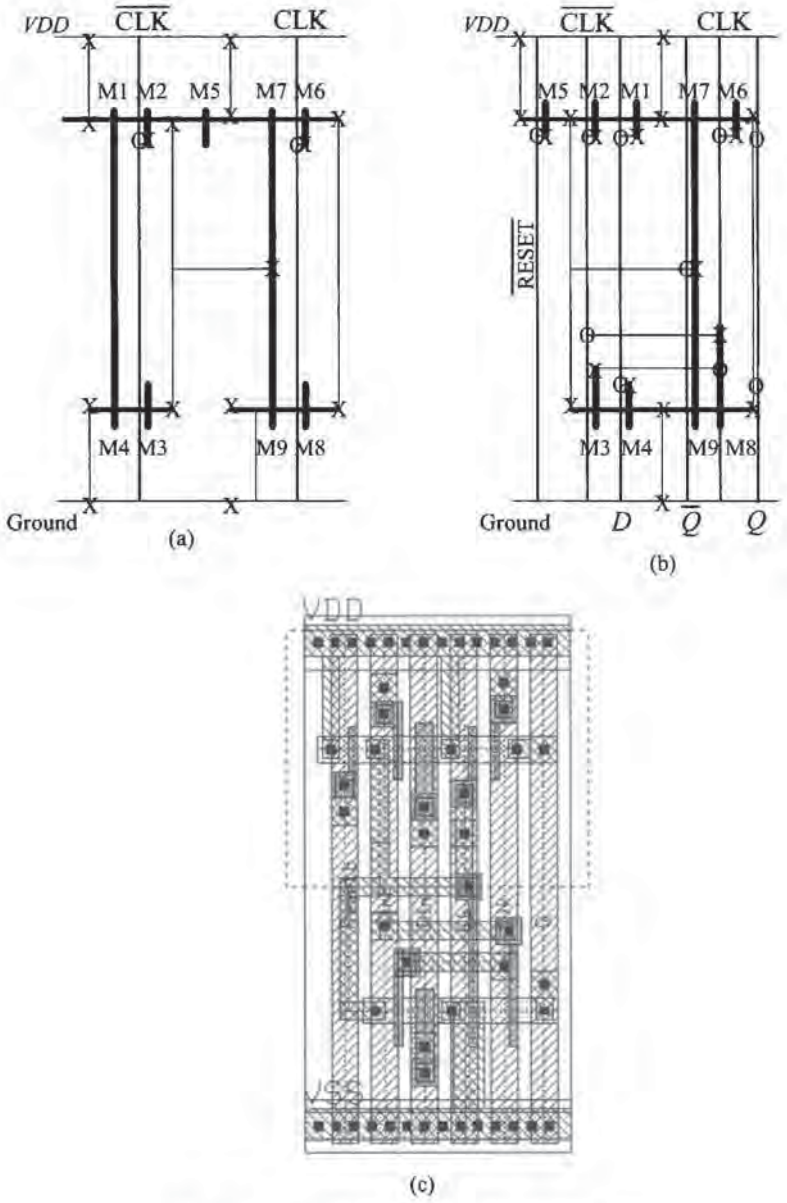



**Figure 15.21** Layout of a dynamic register cell.

part of the standard cells themselves. All remaining connections are made via the routing channels. The standard-cell layouts are designed to accommodate metal2 feedthroughs that run vertically through each cell. The autorouter makes use of this space and adds the feedthroughs as needed to connect or pass signals from one routing channel to another. The routing channels and their associated interconnecting wires are the limiting factors for both the density and circuit performance of this type of layout.

IPR2023-00697
Theta EX2019

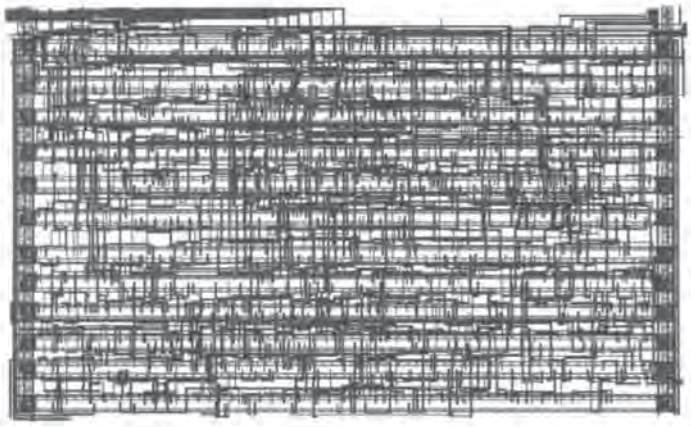


**Figure 15.22** Layout based on standard cells.

Before we continue our discussion of relative layout density, we need to define a metric with which to quantify the matter. It is customary to use the number of transistors per square millimeter of area for this purpose. Because it is a raw number and the common denominator of all circuit layouts, we can use it even when comparing different types of circuitry or even unlike processes.

The density of the standard-cell route shown in Fig. 15.22 is approximately 5,000 transistors per square millimeter. This is fairly representative of the possible density for the channel-based routing approach and the process used (0.8 μm). Figure 15.23 shows a full-custom layout for a digital filter. The circuit area is approximately 2.1 square millimeters. The density is approximately 17,500 transistors per square millimeter, representing a 3.5-fold increase. This circuit, too, is fairly representative of the attainable density of full-custom layout using this particular 0.8 μm process. Both of these circuits were laid out using the same process, and in fact they are from the same die. The device sizes within each block would probably average out to minimum or close to minimum. The main difference affecting density is the interconnect wiring. This overhead associated with interconnect wiring is commonly referred to as the "interconnect burden." The designer must bear this burden in terms of both physical (wasted area) and electrical parameters (parasitic loading). Let us examine one method of creating a high-density custom layout that will minimize interconnect burden and circuit area.

Figures 15.24a–c show a small section of the interpolation filter from Fig. 15.23. In Fig. 15.24a we see an exploded view of four cells that form part of a data-path: an input data register, a t-gate, a full adder, and an output data register. These are instantiated (placed as a cell) twice, creating a view of eight cells. The two adder cells are slightly different: the carry inputs and outputs are on opposite sides, so that the carry-out can cascade to the carry-in of the next adder by abutting (placing next to one another) the cells. Unlike standard cells, the height and width constraints placed on custom layouts are contextual. In other words, a cell's aspect ratio depends on that of its neighbors. In this

IPR2023-00697
Theta EX2019

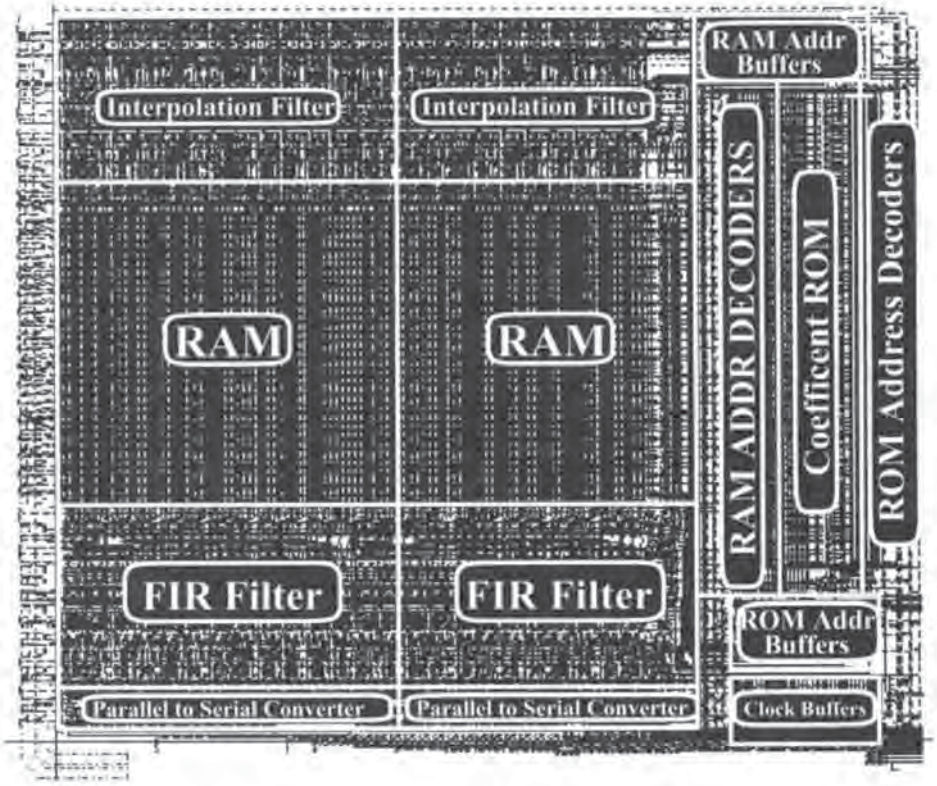



**Figure 15.23** Full custom layout of a digital filter.

case, the width of each cell depended on the maximum allowable width of the widest cell in the group: the data register. Notice the top, bottom, left, and right boundaries of each cell in Fig. 15.24a. Data enter the register cell from the top and are output at the bottom. Clocks, power, ground, and control signals route across all the cells. The adder receives its A and B inputs from the top and outputs their SUM at the bottom. As already mentioned, carry-out and carry-in are available on the left and right edges of the adder, respectively. Figure 15.24b shows a 2-bit slice of this data-path with all of the connections made by cell abutment. Figure 15.24c illustrates how all four edges of each cell join together to complete the hook-up.

We have seen how circuits can be implemented by means of standard cells or custom layout. The time needed to produce a standard-cell route is far less than that of a full-custom implementation. The trade-offs are area and performance. Automatic place and route tools based on routing area rather than routing channels are now coming into use. These promise a compromise solution between the two extremes. The density of their results rivals that of full custom layout. Perhaps the hand-rendered, full custom layout will someday become a thing of the past. Nevertheless, process technology continues to advance, circuit designers continue to design circuits that test the outermost

IPR2023-00697
Theta EX2019

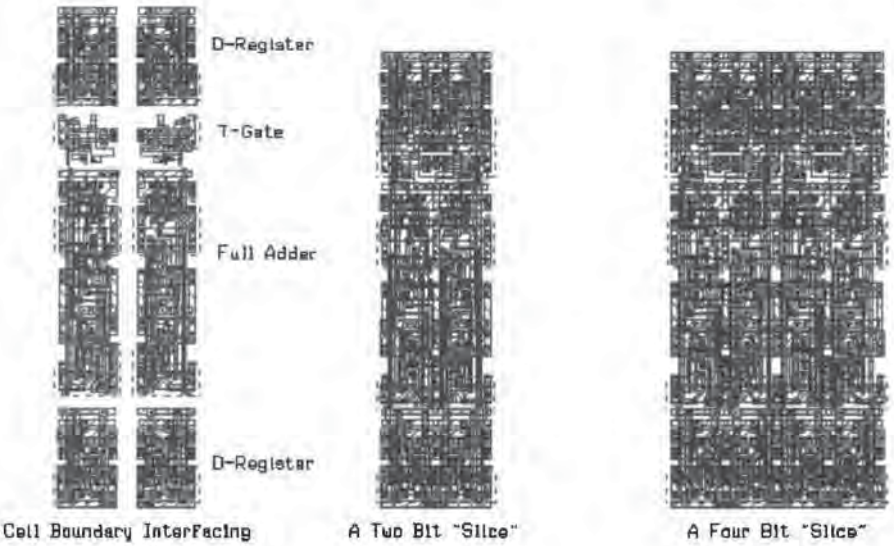



**Figure 15.24** Sections of the digital interpolation filter.

limits of this technology, and the marketplace will still be there demanding ever cheaper, more powerful, and faster products. It is likely then that we will all still have the opportunity to "push a polygon" or two for the foreseeable future. There remains no doubt that the future will bring us ever more powerful software tools that will take over the tedious aspects of placing and connecting layouts, leaving to us the more creative aspects of planning and polishing them.

## ADDITIONAL READING

[1] N. H. E. Weste and D. Harris, *Principles of CMOS VLSI Design*, 4th ed., Addison-Wesley, 2010. ISBN 978-0321547743.

[2] J. P. Uyemura, *Introduction to VLSI Circuits and Systems*, John Wiley and Sons Publishers, 2002. ISBN 0-471-12704-3.

[3] C. Saint and J. Saint, *IC Mask Design: Essential Layout Techniques*, McGraw-Hill, 2002. ISBN 0-071-38996-2. Companion book for reference [4].

[4] C. Saint and J. Saint, *IC Layout Basics: A Practical Guide*, McGraw-Hill, 2001. ISBN 0-071-38625-4. Very good introductory book on IC Layout.

[5] D. Clein, *CMOS IC Layout: Concepts, Methodologies, and Tools*, Newnes Publishers, 2000. ISBN 0-750-67194-7. Excellent introductory book for CMOS IC layout. Covers the entire layout process.

[6] J. Uyemura, *Physical Design of CMOS Integrated Circuits Using L-EDIT*, PWS Publishing Co., 1995. ISBN 0-534-94326-8. Covers the use of the L-EDIT layout software.

IPR2023-00697
Theta EX2019

[7] Kerth, Donald A. *"Floorplanning-Lecture Notes,"* Crystal Semiconductor, Inc.

[8] Kerth, Donald A. *"Analog Tricks of the Trade-Lecture Notes."* Crystal Semiconductor, Inc.

[9] D. V. Heinbuch, *CMOS3 Cell Library*, Addison-Wesley, 1988. ISBN 0-201-11257-4. Excellent source of layout examples.

IPR2023-00697
Theta EX2019

Chapter
16

# Memory Circuits

In this chapter we turn our attention towards the design of semiconductor memory circuits. This includes the array design, sensing, and, finally, the operation of the memory cells themselves. The memories we look at in this chapter are termed random access memories or RAM because any bit of data can be accessed at any time. A block diagram of a RAM is shown in Fig. 16.1. At the intersection of a row line (a.k.a., word line) and a column line (a.k.a., a digit or bit line) is a memory cell. External to the memory array are

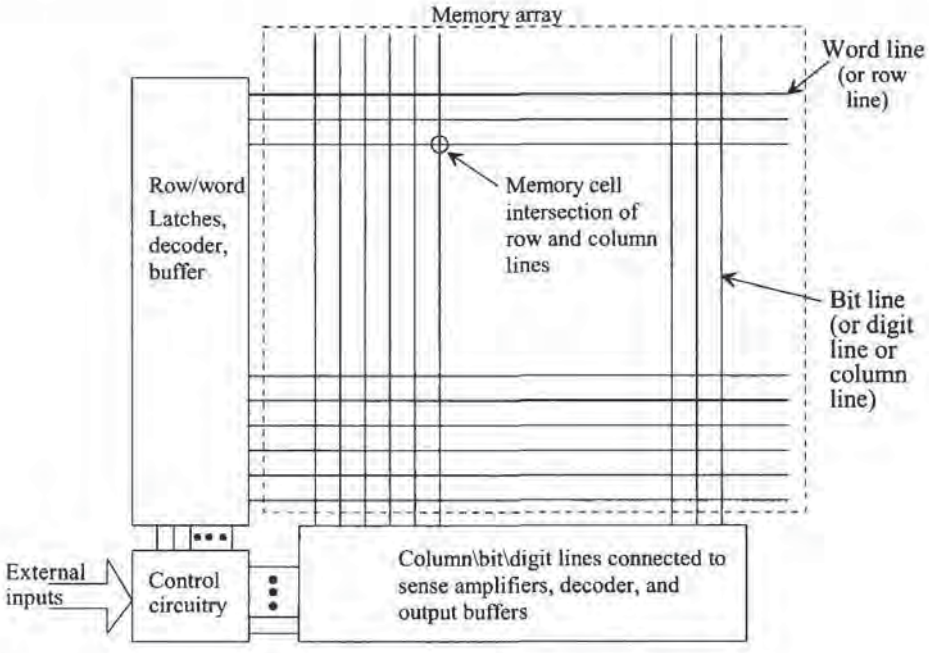



**Figure 16.1** Block diagram of random access memory.

IPR2023-00697
Theta EX2019

the row and column logic. Referring to the row lines, the row address is latched, decoded, and then buffered. A particular row line will, when selected, go high. This selects the entire row of the array. Since the row line may be long and loaded periodically with the capacitive memory cells, a buffer is needed to drive the line. The address is latched with signals from the control logic. After a particular row line is selected, the column address is used to decode which of the bits from the row are the addressed information. At this point, data can be read into or out of the array through the column decoder. The majority of this chapter presents the circuits used to implement a RAM.

## 16.1 Array Architectures

Examine the long length of metal shown in Fig. 16.2. Let's treat this metal line as one of the bit lines seen in Fig. 16.1. The parasitic capacitance of this bit line to ground (substrate) can be calculated using

$$C_{col1sub} = Area \cdot C_{1sub} \tag{16.1}$$

If the capacitance from the metal lines to substrate is 100 aF/μm$^2$, then

$$C_{col1sub} = (0.1)(100)(100\ aF) = 1\ fF \tag{16.2}$$

Not that significant of a capacitance. However, at the intersection of the bit line with every word line we have a memory cell. Let's say that we have a memory cell every 400 nm (250 total cells or word lines) and that each memory cell is connected through an NMOS device to the bit line. As seen in Fig. 16.3, this results in a periodic (depletion or junction) capacitance on the bit line from each MOSFET's source or drain implant. If this capacitance is 0.4 fF, then the total capacitance hanging on the bit line is

$$C_{col} = (\text{number of word lines}) \cdot (\text{capacitance of the MOSFET's source/drain}) + C_{col1sub} \tag{16.3}$$

or $C_{col} = 101\ fF \approx 100\ fF$ for this discussion. For the majority of the discussions in this chapter, we'll treat the column conductor as a capacitor, $C_{col}$. Note that increasing the number of word lines (the number of memory cells connected to a bit line) increases the bit line capacitance.

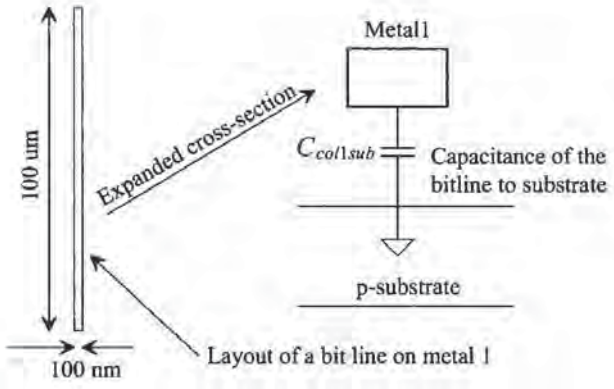



**Figure 16.2** The parasitic capacitance of a bitline to ground (substrate).

IPR2023-00697
Theta EX2019

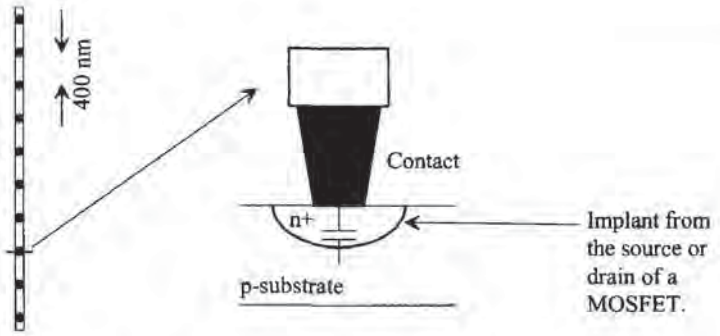



**Figure 16.3** How a bit line is loaded with implants (depeletion capacitance).

### 16.1.1 Sensing Basics

In this section we discuss sensing the datum from a memory cell. Examine Fig. 16.4. When a memory cell is accessed (the word [row] line goes high), the datum from the cell (a charge) is placed on the bit line. The bit line voltage changes. At this point we know the bit line looks like a capacitor and that only one word line can go high at a time in a memory array (so that we don't have two memory cells trying to dump their data onto the same bit line at the same time). The voltage movement on the bit line, $\Delta V_{bit}$, may be very small, e.g., 50 mV or less, and so determining if the voltage is moving upwards or downwards can be challenging. In addition, we would like our sense amplifier to drive the bit line to full, valid, logic levels (for speed reasons in some memories and to refresh the cell in a dynamic RAM, DRAM). Let's consider some sense amplifier topologies.

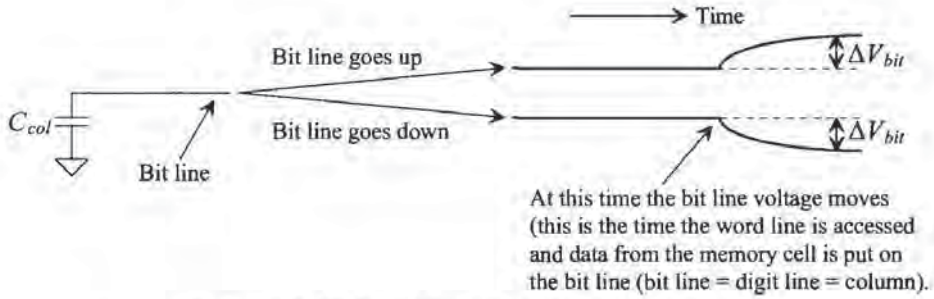



**Figure 16.4** Sensing a change on the bit line in an array.

*NMOS Sense Amplifier (NSA)*

Consider the schematic of an NMOS sense amplifier (NSA) seen in Fig. 16.5. M1 and M2 form the NMOS portion of an inverter-based latch. The idea is to develop an imbalance on the gates of M1/M2 so that one MOSFET turns on harder than the other. Before we start sensing, we set the drains of M1/M2 to the same potential (an equilibrium potential). When *sense_N* goes high (indicating that we are starting the sense operation), the signal *NLAT* (N-latch) goes to ground (the sources of M1/M2 move to ground). While this circuit clearly can't pull the bit line high (we'll add a PMOS sense amplifier later to do this), we should see a problem. How do we get good sensing if the transistor loads are not balanced?

IPR2023-00697
Theta EX2019

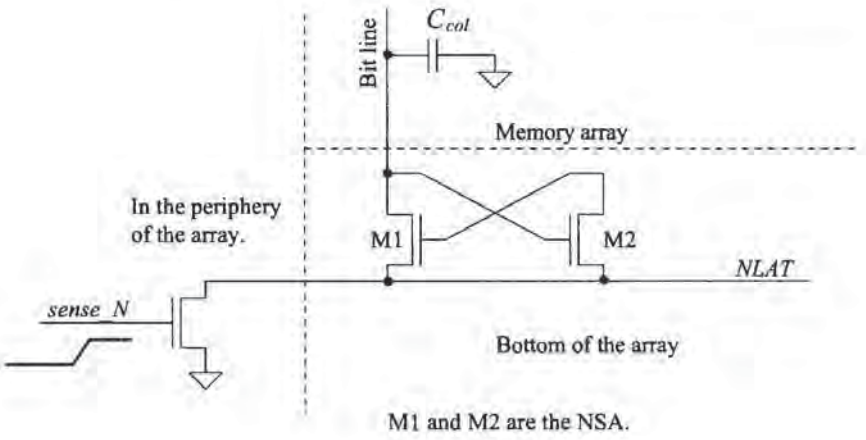



**Figure 16.5** Development of an NMOS sense amplifier (NSA).

*The Open Array Architecture*

To provide the same load capacitance for each input of the NSA, two arrays can be used, as seen in Fig. 16.6. The array architecture in Fig. 16.6, from the side, looks like an open book and so it is called an "open array architecture." Let's use some numbers and the partial schematic seen in Fig. 16.7 to illustrate the sense amplifier's operation.

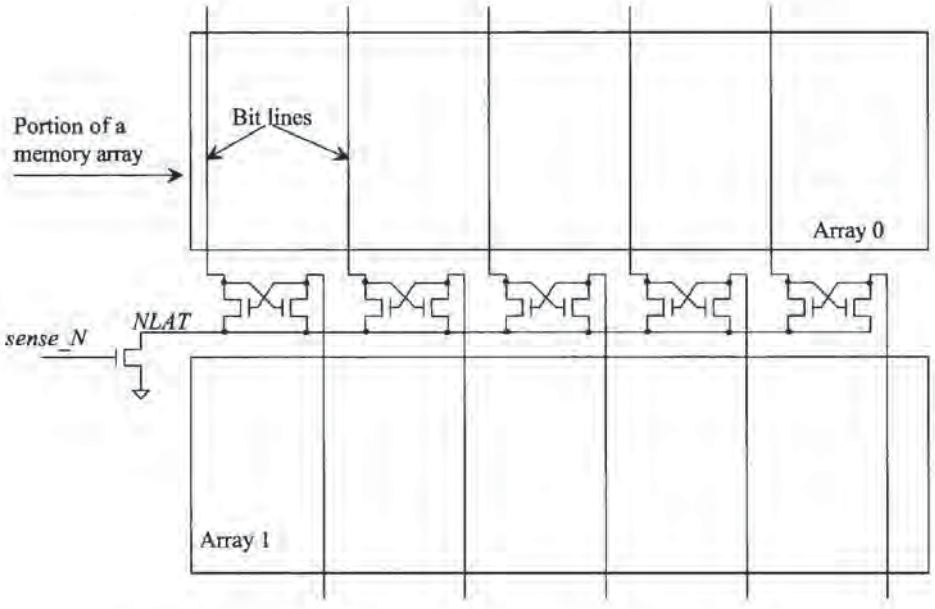



**Figure 16.6** How the NSA is placed between two memory arrays in the so-called open memory array architecture.

IPR2023-00697
Theta EX2019

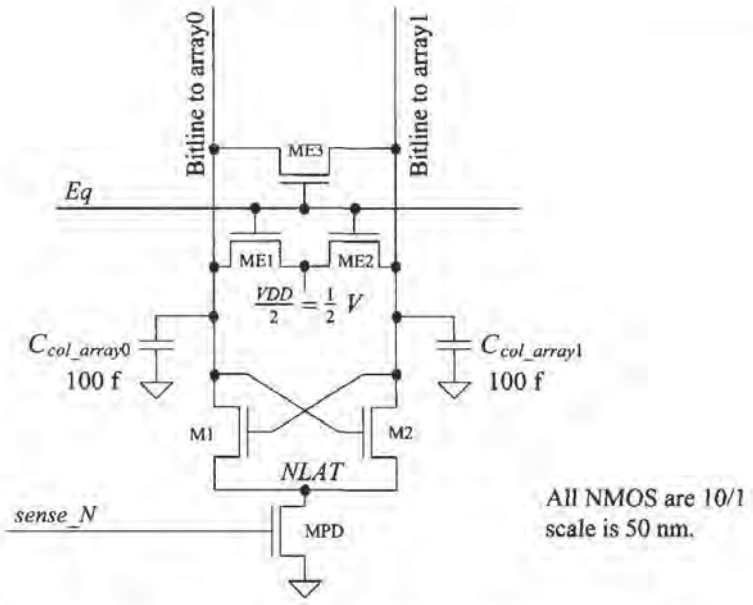



**Figure 16.7** An NSA with equilibration circuitry and connections to two bit lines.

In the following we assume $VDD = 1$ V and the column capacitance is 100 fF. We start any sense operation by equilibrating the bit lines (the inputs to the sense amplifier). In Fig. 16.7 ME1–ME3 short the bit lines together and to $VDD/2$ (= 0.5 V). When the equilibrate signal, *Eq*, goes high, all of the word lines are low. We are not accessing any data in the array but, rather, getting ready for the sense (read) operation. Figure 16.8 shows how the *Eq* signal is asserted for a short period of time and how, during this time, the bit lines are equilibrated together and to 0.5 V.

During a sense operation, one of the bit lines will be pulled from $VDD/2$ to $VDD$. The other line will be pulled from $VDD/2$ to ground. The amount of power used during a sense depends on frequency and given by

$$P_{avg} = (\text{number of sense amplifiers}) \cdot C_{col} \cdot (VDD/2)^2 \cdot f \qquad (16.4)$$

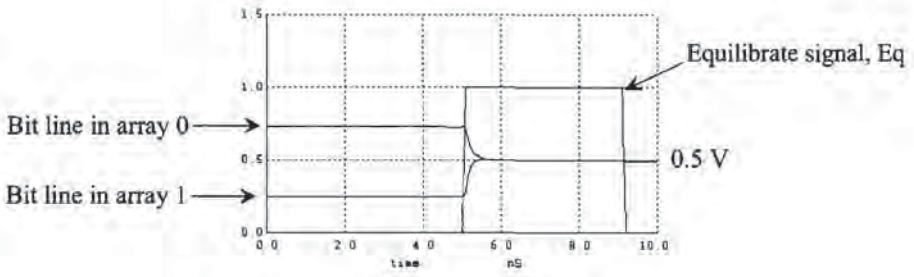



**Figure 16.8** How the equilibrate circuitry operates.

IPR2023-00697
Theta EX2019

Note, however, that when we equilibrate, no power is consumed. The two bit line capacitances simply share their charge (one going from ground to *VDD*/2 while the other goes from *VDD* to *VDD*/2).

To show a sensing operation, let's use the basic one-transistor, one-capacitor (1T1C) DRAM memory cell seen in Fig. 16.9. To fully turn on the access MOSFET, we need to drive the word line to $VDD + V_{THN}$ (with body effect). A typical value for the memory bit capacitance, $C_{mbit}$, is 20 fF. If the voltage on this capacitance is called $V_{mbit}$ and the bit line is precharged to *VDD*/2, then we can write the total charge on both capacitors *before* the access MOSFET turns on as

$$Q_{tot} = C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col\_array} \tag{16.5}$$

After the access MOSFET turns on, the voltage across each capacitor will be the same. We'll call this voltage, $V_{final}$. Since charge must be conserved

$$V_{final}(C_{mbit} + C_{col\_array}) = C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col\_array} \tag{16.6}$$

or

$$V_{final} = \frac{C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col\_array}}{C_{mbit} + C_{col\_array}} \tag{16.7}$$

If a logic one is stored on $C_{mbit}$ (which means $V_{mbit}$ is *VDD*, or here, 1 V) and $C_{mbit} = 20$ fF and $C_{col\_array} = 100$ fF, then $V_{final} = 0.583$ V. The change in the bit line voltage is

$$\Delta V_{bit} = V_{final} - VDD/2 \tag{16.8}$$

or here $\Delta V_{bit} = 83\ mV$.

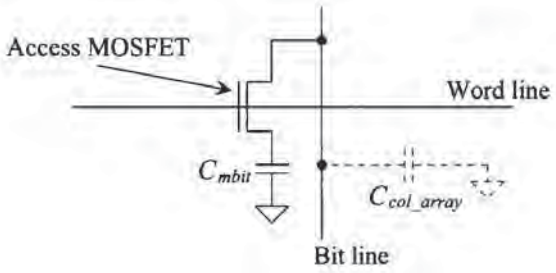



**Figure 16.9** The one-transistor, one-capacitor (1T1C) DRAM memory cell.

Figure 16.11 shows the signals resulting from the operation of the sense amplifier in Fig. 16.10. The 1T1C DRAM (mbit) cell in array1 (the DRAM cell at the bottom of the schematic) has its word line held at ground when we sense the data in the cell in array0. The bit line from array1 is simply used as a reference. When we sense the data in array1, the bit line in array0 is used as the reference (and so the word line in array0 is held at ground).

For the simulation results in Fig. 16.11, we start out by equilibrating the bit lines (as in Fig. 16.8). Next our word line, in array0, goes to a voltage greater than $VDD + V_{THN}$. This causes charge sharing between $C_{mbit}$ and the digit line capacitance. For the simulation results seen in Fig. 16.11, we set the voltage on $C_{mbit}$ to zero so we are reading out zero.

IPR2023-00697
Theta EX2019

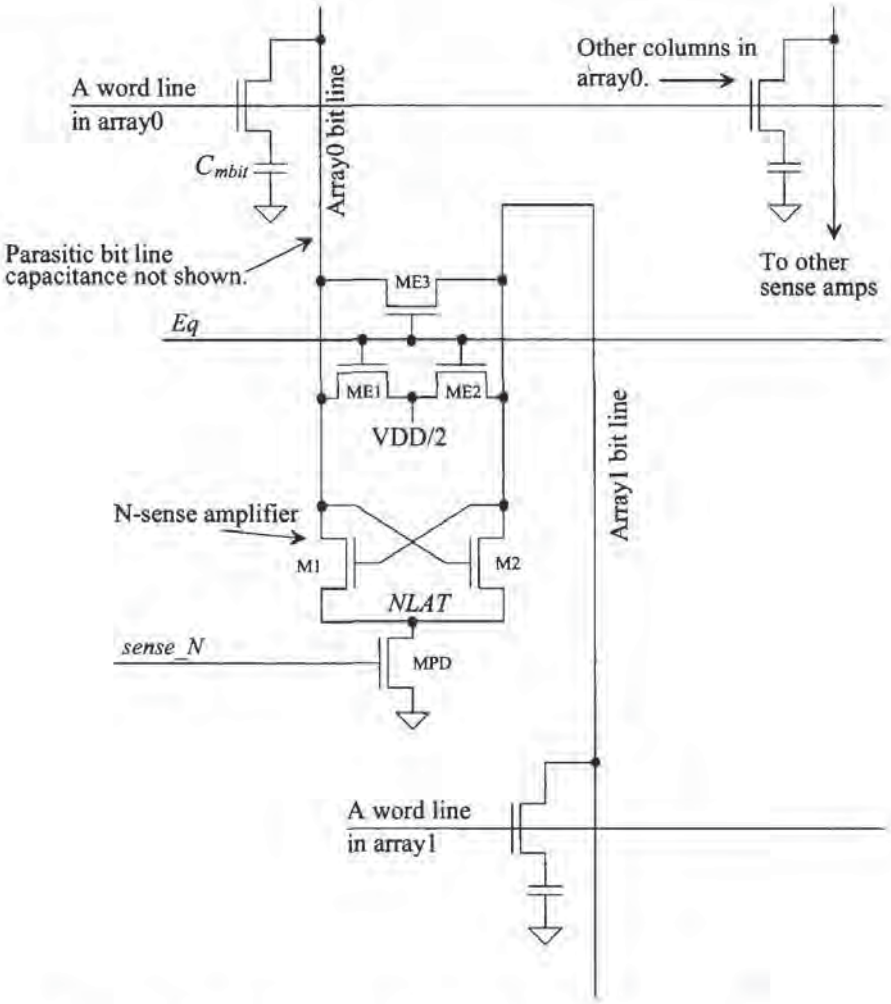



**Figure 16.10** The connection of the NSA to the memory arrays.

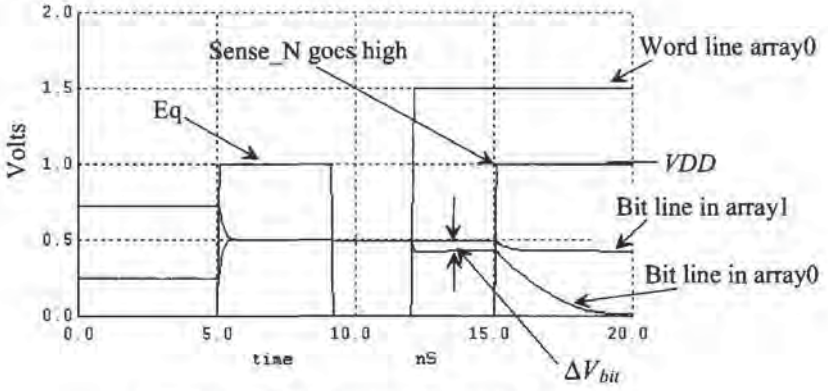



**Figure 16.11** N-sense amp's operation.

IPR2023-00697
Theta EX2019

When *sense_N* goes high, the NSA "fires" causing the bit line in array0 to move to ground. Note that our reference bit line (from array1) droops a little. Figure 16.12 shows the signals in a sensing operation if the mbit cell contains a "1." The bit line voltage in array0 now increases. The reference bit line in array1 is pulled to ground (this doesn't harm the data in array1). To pull the bit line in array0 high, we'll now add a PMOS sense amplifier (PSA).

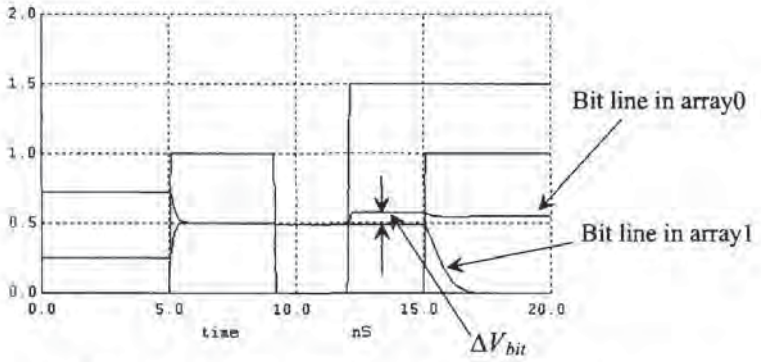



**Figure 16.12** Reading out a "1" from the cell in array0.

*PMOS Sense Amplifier (PSA)*

To pull the bit lines up to *VDD*, we can add a PSA to the periphery of the array. Figure 16.13 shows the schematic of the PSA. The signal *ACT* (active pullup) is common to all of the PSAs on the periphery of the array (as is *NLAT* for the NSAs seen in Fig. 16.6). The signal $\overline{sense\_P}$ is active low and indicates that the PSA is firing. The PSA is usually fired after the NSA because the matching of the NMOS devices is, generally, better than

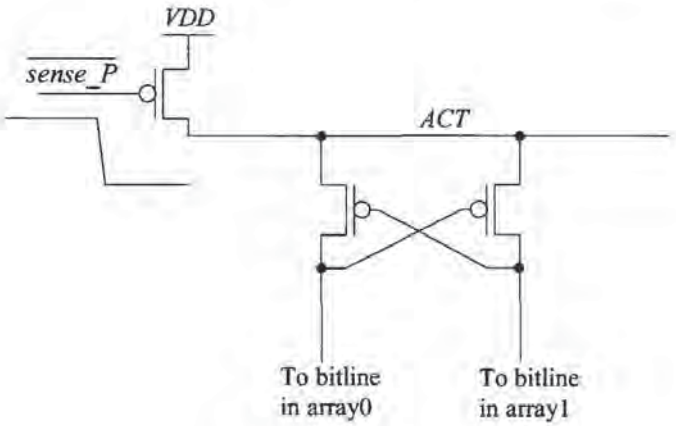



**Figure 16.13** Schematic diagram of a PMOS sense amplifier.

IPR2023-00697
Theta EX2019

the matching of the PMOS. It's not common to fire the sense amplifiers at the same time because of the potentially significant contention current that can flow. Figure 16.14 shows the full operation of a sense when the cell we are reading out has a one stored in it.

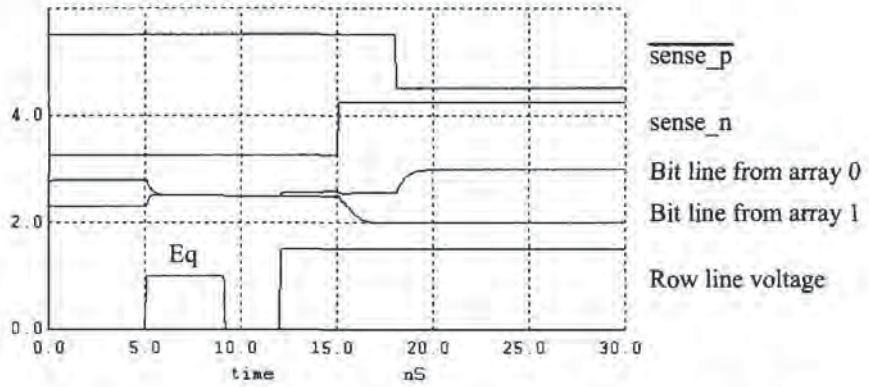



**Figure 16.14** How the PSA pulls the bit line from array 0 high.

*Refresh Operation*

Our sensing operation determined whether a 1 or a 0 was stored in the memory bit. For this particular sense amplifier topology, the inputs and outputs are the same terminals. This is fundamentally important for DRAM operation where the charge can leak off the capacitor because of the finite subthreshold slope of the access transistors or leakage through the MOSFET's source/drain implant to substrate (hence the name "dynamic"). To ensure long-lasting data retention in a DRAM, the cells must be periodically refreshed. When we fire the sense amplifiers, with the access device still conducting, the mbit capacitor is refreshed through the access MOSFET (the bit line is driven to ground or *VDD* by the sense amp).

### 16.1.2 The Folded Array

The open array architecture seen in Fig. 16.6 has a memory cell at the location of every intersection of a word line and a bit line. The open array architecture results in the most dense array topology. Unfortunately, because of the physical distance between the bit lines used by the sense amplifier, this architecture is sometimes not used. Figure 16.15 shows the basic problem. Coupled noise (e.g., from the substrate) feeds unequal amounts

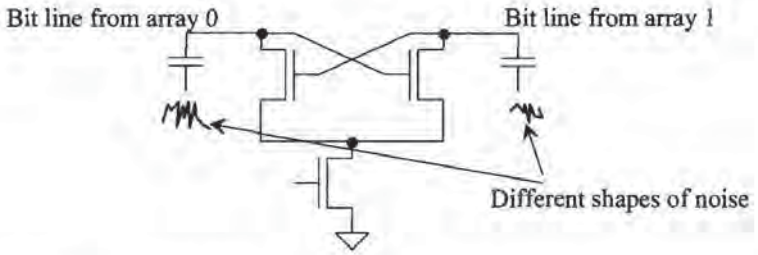



**Figure 16.15** Different amplitudes of coupled noise into physically separated bit line.

IPR2023-00697
Theta EX2019

of charge into the bit lines. This can cause the sense amplifier to make wrong decisions. To attempt to make each bit line see the same sources of noise, we can lay the bit lines out next to each other by folding array 1 on top of array 0 (see Fig. 16.6). The result, called a *folded array architecture*, is seen in Fig. 16.16. The bold lines in this figure are the bit lines from array 1.

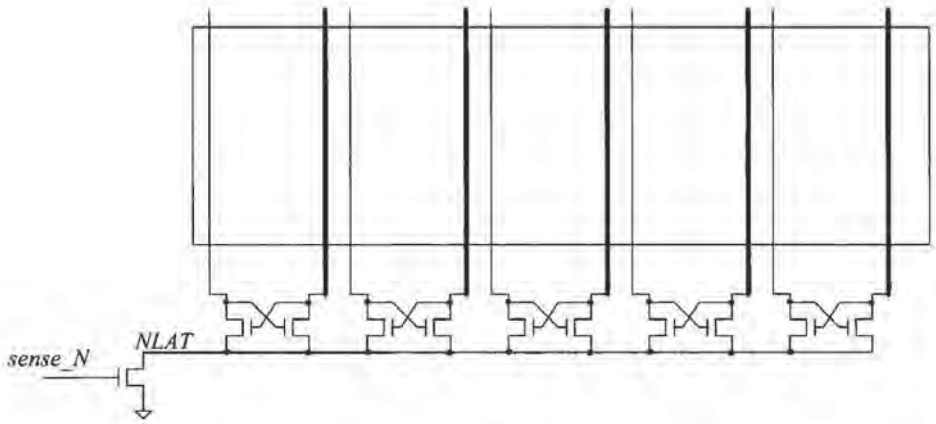



**Figure 16.16** The folded array is formed by taking the open array architecture (open book) topology seen in Fig. 16.6 and "closing the book," that is, folding array 1 on top of array 0. Note that the bold lines indicate the bitlines from array 1 in the newly formed array.

What is the cost for this improved noise performance? We know that when a sense amplifier is used, one input is varied by the memory cell we are sensing, while the other input is simply used as a reference. What this means is that instead of having a memory cell at the intersection of every row and column, as in the open array, now we have a memory cell at the location of every other row and column, Fig. 16.17.

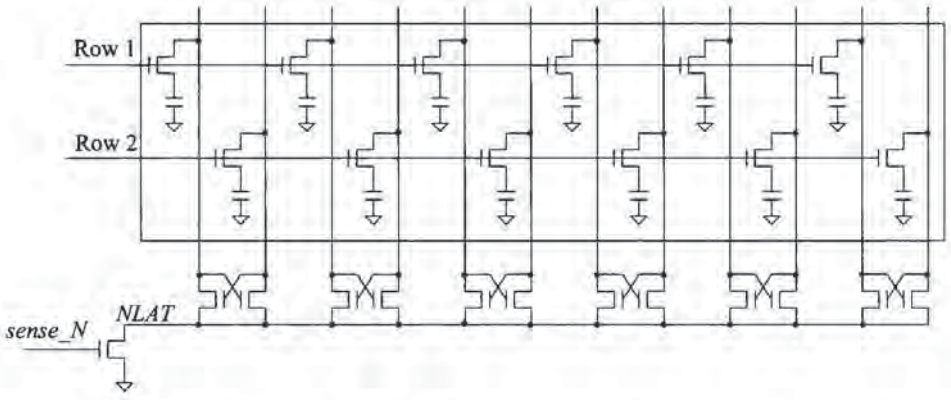



**Figure 16.17** How a memory cell is located at every other intersection of a row line and a column line in a folded-area architecture.

IPR2023-00697
Theta EX2019

*Layout of the DRAM Memory Bit (Mbit)*

Layout area, especially in a periodic array like a memory, is very important. Reducing the chip size increases the number of die on a wafer. Since the processing cost for a wafer is fixed, having more chips to sell per wafer increases the manufacturer's profit. To reduce the size of an mbit, often (always in DRAM) the contact to the bit line is shared between two mbit cells, Fig. 16.18. The word lines (row lines) are made using silicided polysilicon. Using polysilicon for the word lines can lead to significant delays. As seen in Fig. 16.17, for example, a signal applied to row 1 must propagate down a distributed $R$ (the resistance of the polysilicon) $C$ (the gate oxide capacitance of the MOSFET) line. The propagation delay through a word line can be estimated using

$$t_d = \overbrace{(\text{number of columns}) \cdot \left( WL\frac{\varepsilon_{ox}}{t_{ox}} + C_{parasitic} \right)}^{\text{total capacitance on the word line}} \cdot \overbrace{(\text{number of columns}) \cdot R_{gate}}^{\text{total resistance of the word line}} \qquad (16.9)$$

The number of columns is simply the number of MOSFETs in the row. The term $WL\frac{\varepsilon_{ox}}{t_{ox}}$ is $C_{ox}$, while $R_{gate}$ is the resistance from one end of the polysilicon gate to the other end in the memory cell layout seen in Fig. 16.18. The term $C_{parasitic}$ is the parasitic capacitance associated with the cell (such as the capacitance from the word line to the bit line). If $C_{ox}$ is 400 $aF$, $C_{parasitic}$ = 100 $aF$, $R_{gate}$ = 4 Ω, and there are 512 bit lines in the array then the delay time through the word line is estimated as 500 ps. To fully turn the word line on (not just to the 50% point where delay is measured), we would probably want 3 ns.

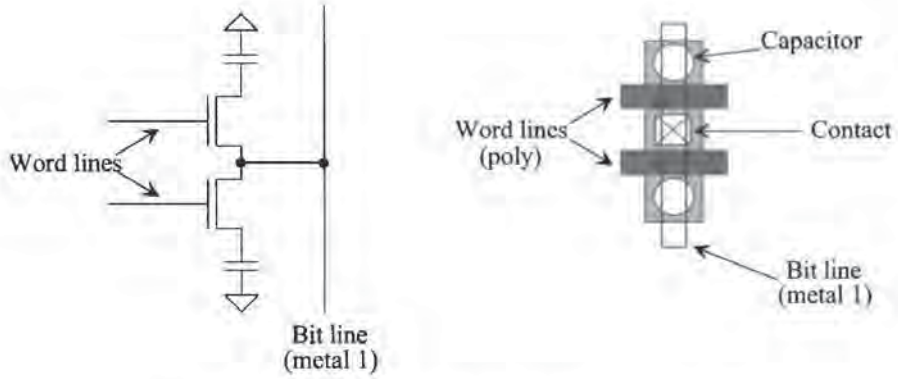



**Figure 16.18** Two mbits sharing a contact to the bit line.

A section of the area layout for an open architecture DRAM is seen in Fig. 16.19. Notice that at the intersection of every bit and word line is a memory cell. A common term used to describe the density of, or distance in, a periodic array is *pitch*. As seen in Fig. 16.19, the pitch is defined as the distance between like points in the array. We used the distance between the right side of the digit lines to show pitch in this figure. A figure of merit for memory cell layout is its area. The area of the mbit DRAM cell seen in this figure is $6F^2$ where $F$ = pitch/2. Question: How many MOSFETs are laid out in Fig. 16.19? Answer: Because polysilicon over active forms a MOSFET, we simply count the number of times poly crosses active (12 MOSFETs).

IPR2023-00697
Theta EX2019

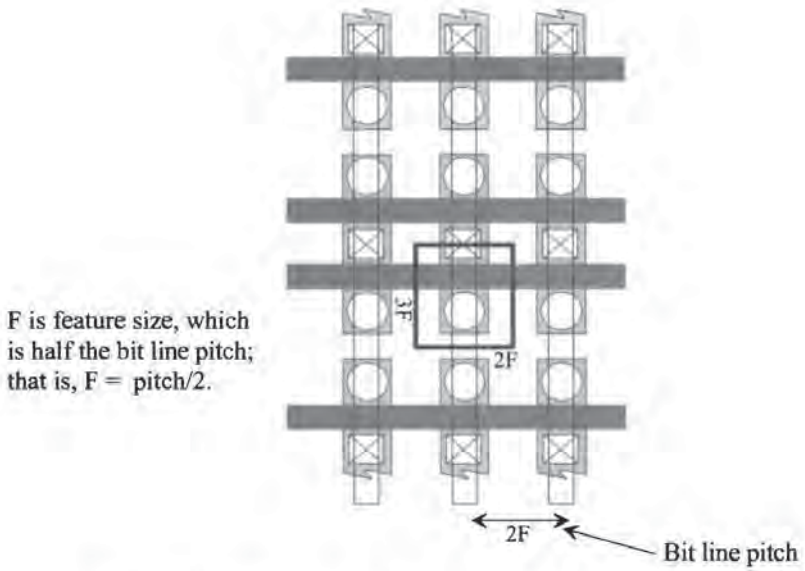



**Figure 16.19** Layout of mbit used in an open bit line configuration.

Figure 16.20 shows the layout of the cell used in the folded area architecture. The schematic seen in this figure is different from the one seen in Fig. 16.17. In Fig. 16.20 our mbits share the contact to the bit line. This means that memory cells are located at the intersection of every other mbit pair with the bit lines. Figure 16.21 shows a section of the array layout in the folded architecture. The cell size is $8F^2$. The increase in the cell size is due to the needed poly interconnect between adjacent cells (the poly that runs over the field oxide, FOX).

Figures 16.22 and 16.23 show the process cross sectional views for mbits using trench capacitors and buried capacitors, respectively. The trench capacitor is formed by etching a hole in the substrate. For high-density memory, the aspect ratio (the ratio of the

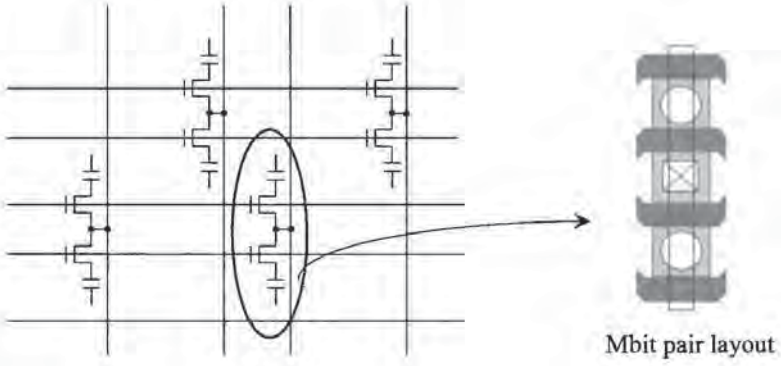



**Figure 16.20** The mbit pair used for a folded architecture.

IPR2023-00697
Theta EX2019

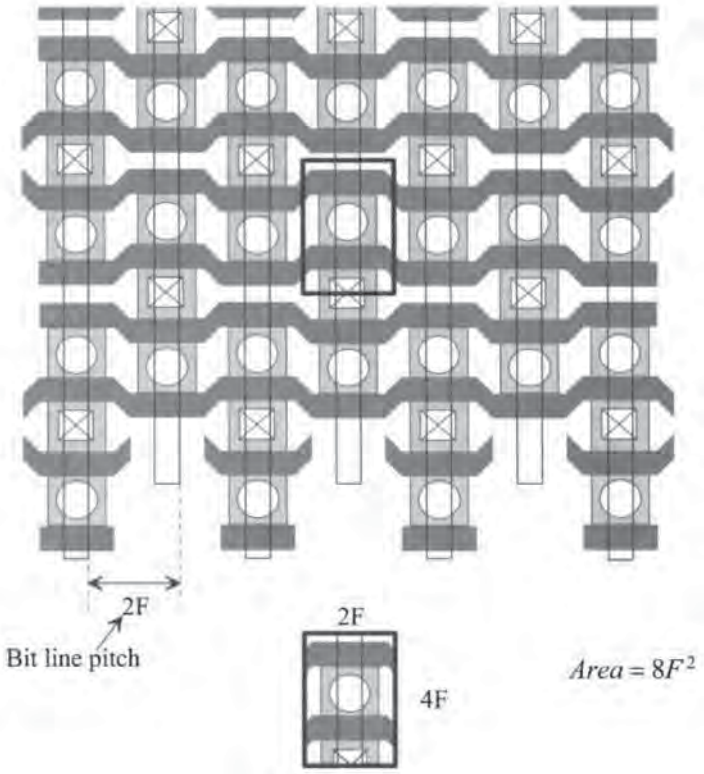



**Figure 16.21** Folded architecture layout and cell size.

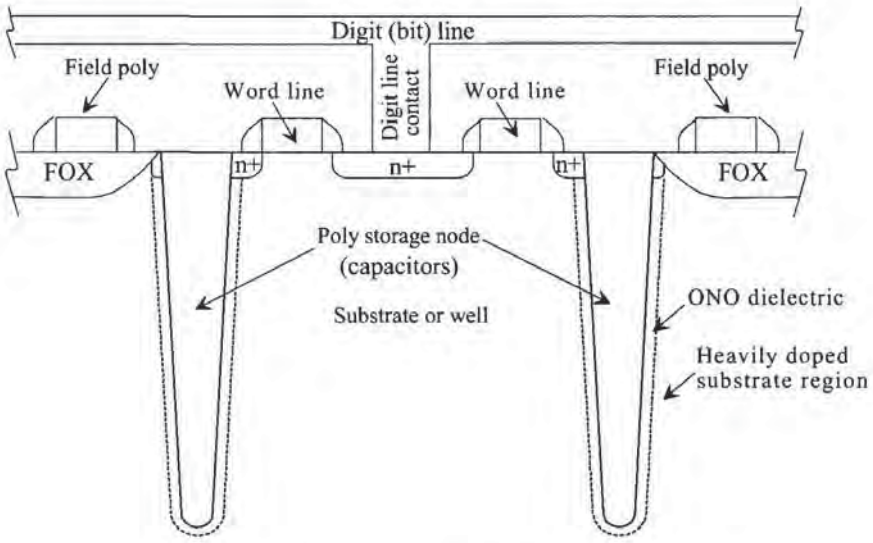



**Figure 16.22** Cross-sectional view of a trench capacitor cell.

IPR2023-00697
Theta EX2019

hole's depth to its diameter) can be quite high, which leads to processing concerns. In Fig. 16.23 the buried capacitor cell pair is seen. Unlike the trench capacitor-based cell that places the cell directly in the substrate, the capacitor is "buried" under the digit line but still above the substrate. The benefit of this cell is simpler processing steps. One problem with this cell, when compared to the trench-based cell, is that the parasitic capacitances are higher (such as the capacitance loading the (digit) bit line). Also, for a given bit capacitance, $C_{mbit}$, the area of the buried capacitor cell may need to increase, while the area of the cell using the trench capacitor remains constant. The depth of the trench capacitor is increased for more capacitance.

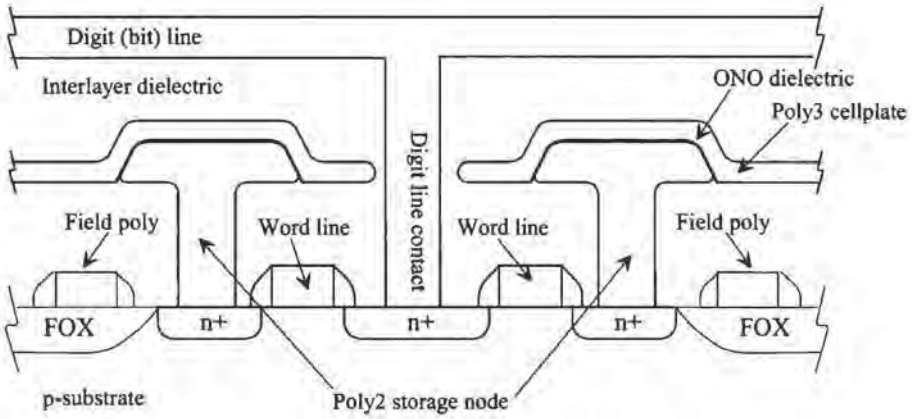



**Figure 16.23** Cross-sectional view of a buried capacitor cell.

As we saw in Eq. (16.7), the value of $C_{mbit}$ is very important. Thin dielectrics with high dielectric constants are used to implement $C_{mbit}$. Tricks like roughing up the dielectric to increase the surface area are often employed. To minimize the stress on the oxide, the common node of the capacitor is usually tied to $VDD/2$, as seen in Fig. 16.24. When a logic 1 ($VDD$) is written to the cell, the charge on the capacitor is $(VDD/2) \cdot C_{mbit}$. When a logic 0 (ground) is written, the charge on the capacitor is $-(VDD/2) \cdot C_{mbit}$. The difference in these charges is $VDD \cdot C_{mbit}$, which is the same difference we would have if the common node were connected to ground.

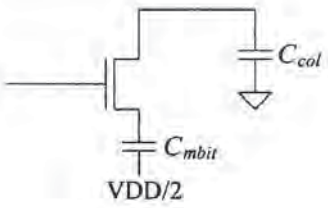



**Figure 16.24** Holding the capacitor's common plate at VDD/2 to minimize oxide stress.

IPR2023-00697
Theta EX2019

### 16.1.3 Chip Organization

The number of rows in a DRAM is limited because each additional row adds capacitance to the bit line. The power goes up, as seen in Eq. (16.4), with increasing $C_{col}$. The number of bit lines is limited by the delay through the row line (the length of the row line), as indicated in Eq. (16.9). For these reasons, the size of a DRAM array is limited. A typical array size is 512 word line pairs and 512 bit lines for a total memory size of 256 kbits. We use the term "pairs" to indicate that, as seen in Fig. 16.20, there are actually 1,024 row lines in a 256k array. We can arrange these 256k arrays to form a larger memory array. Figure 16.25 shows the basic idea for an 16-Mbit array block. A 1-Gbit DRAM uses 8,192 256k arrays. These arrays may be further subdivided into "banks" of memory. We might, for a 1-Gbit memory, have 8 banks of 128-Mbits. We subdivide the memory so that an operation (say a read) can be done in one bank while, at the same time, some other operation (say a refresh) can be performed in a different bank.

The *array efficiency* is defined as the ratio of the area of the memory blocks to the total chip area. A typical value ranges from 50 to 60%. The peripheral circuitry seen in Fig. 16.25 (the sense amplifier and row decoder blocks) take up a significant amount of chip real estate. Because of their importance, we'll devote the entire next section to the design of peripheral circuits including sense amplifiers, row drivers, and decoder circuits.

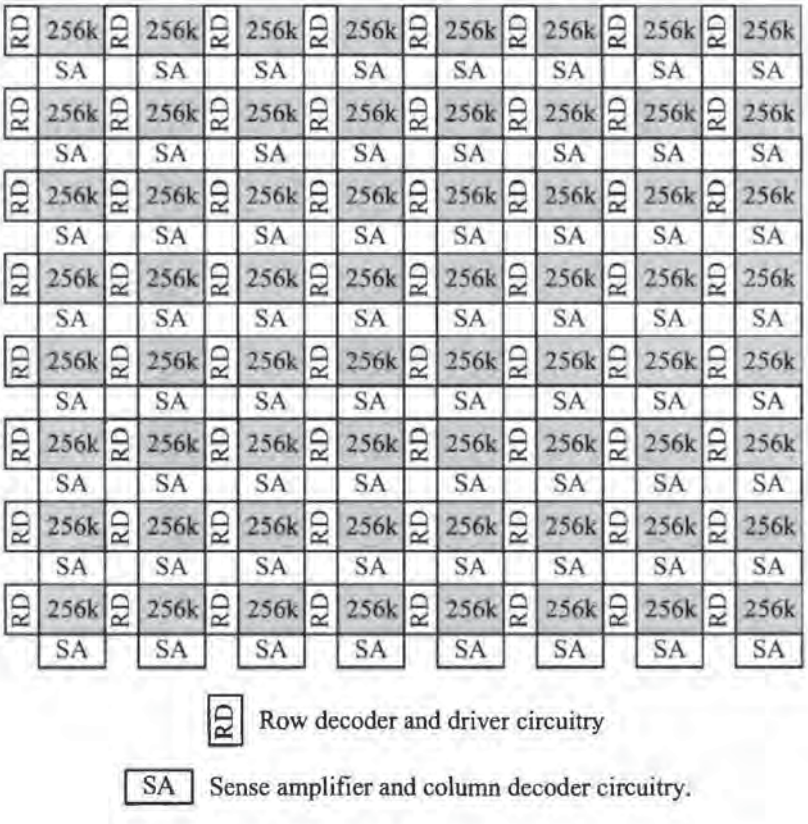



**Figure 16.25** A 16-Mbit array block.

IPR2023-00697
Theta EX2019

## 16.2 Peripheral Circuits

In this section we discuss the design of general sense amplifiers (for general use in memory design), row drivers (remembering, for a DRAM, the row voltage needs to be driven above *VDD*), and column and row decoder circuits.

### 16.2.1 Sense Amplifier Design

Examine the clocked sense amplifier seen in Fig. 16.26. When *clock* is low, MS3 is off while MS1 and MS2 are on. Our input signals can't go below $V_{THP}$ (with body effect) without shutting MS1 and MS2 off. Assuming that the inputs stay above $V_{THP}$, the drains of M1/M3 and M2/M4 are charged to In+ or In– (creating an imbalance). When *clock* goes high, the imbalance causes the circuit to latch high or low depending on the state of the inputs. Figure 16.27 is the simulation output showing the typical operation of the circuit. When *clock* is low, the circuit outputs are not valid logic signals but rather, ideally, track the input signals. This circuit is plagued with problems including: kickback noise, memory, and significant potential contention current.

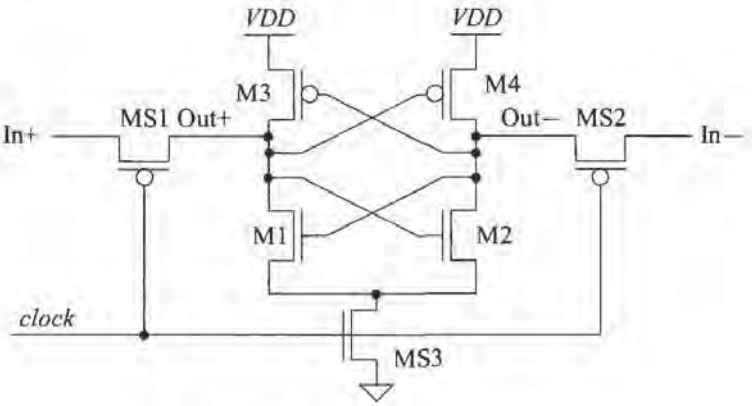



**Figure 16.26** Clocked sense amplifier.

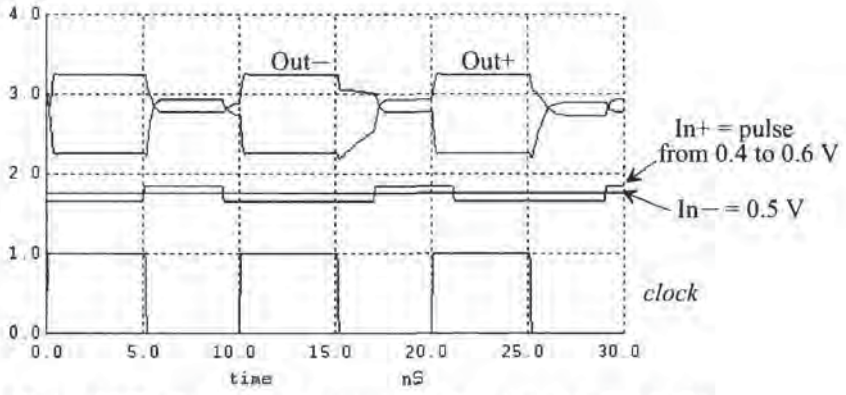



**Figure 16.27** Simulating the operation of the circuit in Fig. 16.26.

IPR2023-00697
Theta EX2019

*Kickback Noise and Clock Feedthrough*

In the simulation that generated Fig. 16.27 we used, for our inputs, voltage sources. In a real application, the inputs would come from some other logic or a bit line (a source with a finite driving impedance). Consider including logic for driving the sense amplifier seen in Fig. 16.28. We use long-length inverters to simulate a weak driver circuit (such as a decoder made with a series of transmission gates). Seen in the figure are simulation results at the time when the clock goes high (we get similar glitches when the clock goes low). The large glitch seen in these figures is called *clock feedthrough* noise. It is present when a clock signal has a capacitive path directly to the input of the sensing or comparator circuit. The clock feedthrough noise in Fig. 16.28 is close to 50 mV on both inputs.

Another type of noise, called *kickback noise*, is present and injected into the inputs of the circuit when the latch switches states. *It's important to simulate the operation of the sense amplifier with nonideal sources (with finite source resistance) to determine the significance of clock feedthrough and kickback noise.* Using voltage sources with 10k resistors is a reasonable source for general circuit simulations. This noise is an important specification for a sense amplifier. It can often be the *limiting factor* when making sensitive measurements. If the kickback noise, for example, is too large, it can interfere with sensing. A simple example of a potential problem occurs when 256 sense amps are all being clocked at the same time (e.g., on the bottom of a memory array) and they are directly adjacent to each other. We don't want one of the amplifiers interfering with any of the others by feeding noise into the others' inputs.

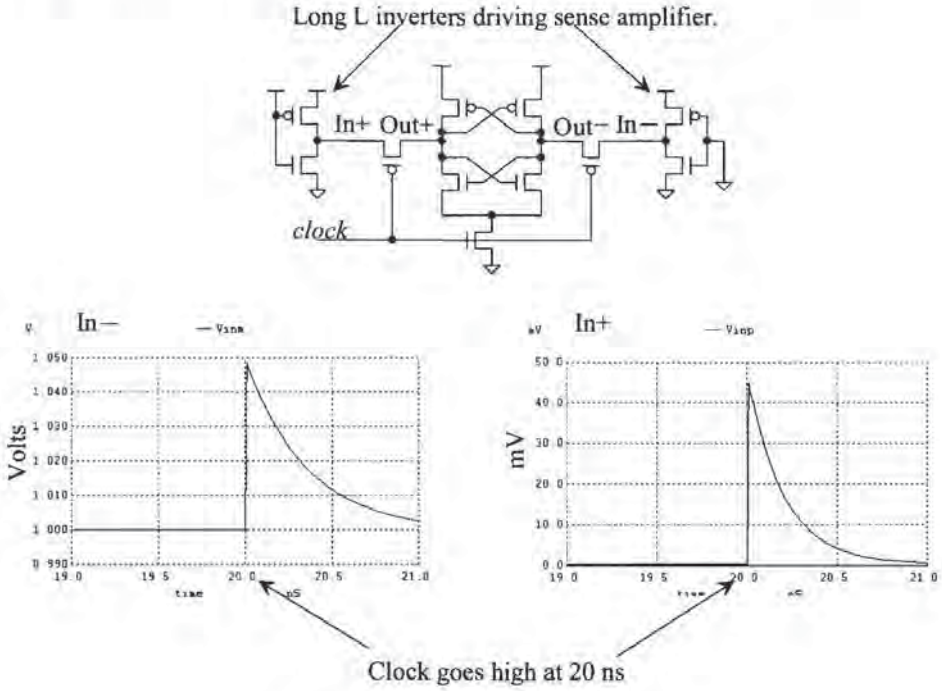



**Figure 16.28** Clock feedthrough noise.

IPR2023-00697
Theta EX2019

*Memory*

For a good comparison, the sense amplifier shouldn't have a memory of previous sensing operations. For the sense amplifier in Fig. 16.26, the outputs are actively driven by the input signals prior to clocking. However, the drain of MS3 (sources of M1/M2) is floating. This node is a dynamic node that floats to a voltage dependent on the previous decision and the input signals. *For a precision sense operation, all nodes in the sense amplifier must be driven or equilibrated, prior to clocking, to a known voltage.*

*Current Draw*

Because there may be thousands of sense amplifiers operating on a chip at the same time, it is extremely important to minimize the power used by these circuits. Let's take a look at the current supplied by *VDD* to the circuit in Fig. 16.26 (with the signals seen in Fig. 16.27). Because the inputs actively drive the gates of M3 and M4, it is possible M3/M4 source a significant current into the inputs. In this case, Fig. 16.27, the $V_{SG}$ voltages are roughly 0.5 V (and so both M3 and M4 are conducting a drain current). The current supplied by *VDD* is seen in Fig. 16.29. *Clock* is high during 0 to 5ns, 10ns to 15ns, etc. (when the current supplied by *VDD* is small). If the average current is 50 μA and there are 1,000 sense amplifiers operating on the chip, then *VDD* supplies 50 mA of current (just to the sense amplifiers). *For minimum power, it's important that there are no DC paths from VDD to ground except during switching times.*

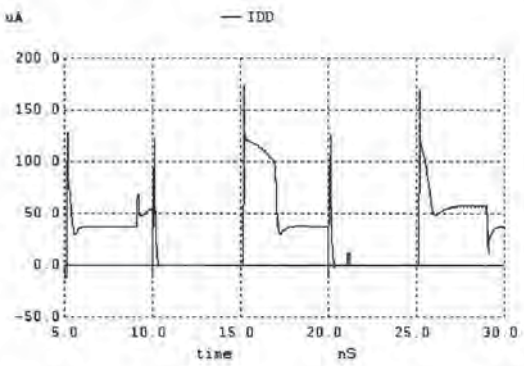



**Figure 16.29** The current that flows in the sense amplifier in Fig. 16.26.

*Contention Current (Switching Current)*

When switching takes place in the sense amplifier of Fig. 16.26, both M3 and M4 are conducting. When *clock* goes high, both M1 and M2 turn on and conduct current as well. In this situation there is a direct path from *VDD* to ground. If the inputs are at relatively the same voltages, the time that M1–M4 remain conducting can be very long (resulting in significant current being pulled from *VDD*). To eliminate this *metastable* condition and force the comparator to make a decision (switch to valid logic levels), the gain in series with the input signals can be increased. A simple example of increasing input signal gain is to place an amplifier on the inputs of the latch.

Note that if our input signals amplitudes are within a $V_{THP}$ of *VDD*, the PMOS devices are off and we don't have this problem. The inputs can generate an imbalance on

IPR2023-00697
Theta EX2019

the drains/gates of the NMOS transistors when clock goes high. Neither PMOS device will turn on until either M1 or M2's drain moves below $VDD - V_{THP}$. *For low power (low contention current), try to keep one side of the latch shut off until a significant imbalance is present in the circuit.*

*Removing Sense Amplifier Memory*

Figure 16.30 shows how the sense amplifier's memory can be erased. M1–M4 form a latch (Fig. 13.16). To remove the sense amplifier's memory, *all* nodes in the sense amplifier must be actively driven to a known voltage (no floating or dynamically charged nodes). When *clock* is low, the sense amplifier's outputs are pulled to *VDD* through MS3 and MS4. The MOSFETs MS1 and MS2 are off, breaking the connection between *VDD* and ground (so no current flows in the latch). The gates of M1 and M2 are at *VDD* so their drains are actively driven to ground. In other words, when *clock* is low, all nodes in the circuit are either pulled high to *VDD* or low to ground.

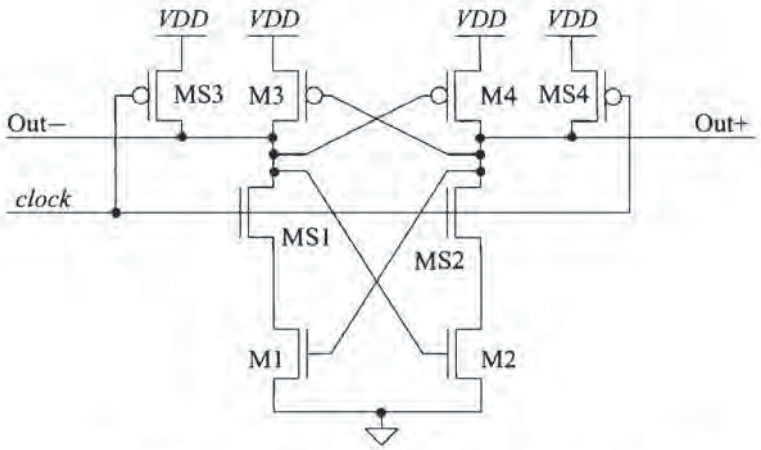



**Figure 16.30** Removing memory in a sense amplifier.

*Creating an Imbalance and Reducing Kickback Noise*

When *clock* goes high, the latch action can take place. For the comparison to function as desired, we need to generate an imbalance. This, alone, isn't too difficult. What is difficult is creating an imbalance for a wide range of input signals while not causing an excessive amount of current to flow in the circuit. Consider the additions to our sensing circuit seen in Fig. 16.31. MB1/MB2 can be connected to either the drain or the source of MS1/MS2, as seen in the figure. Let's first consider connecting them to the drains. When *clock* is low, the drains of MB1 and MB2 are pulled to *VDD*. If the input voltages are significant, a *large* current can be drawn from *VDD* (when *clock* is low). If the input voltages are relatively small, then the current pulled from *VDD* may be tolerable (keeping in mind that the sense amplifier doesn't function correctly when the input voltages are less than $V_{THN}$). The benefit of connecting to the drains is high gain. Very small voltage differences can cause quick sensing (the outputs swing quickly to valid logic voltages). Note that the possibility for large current flowing through either MB1/MB2 is present independent of the state of *clock*.

IPR2023-00697
Theta EX2019

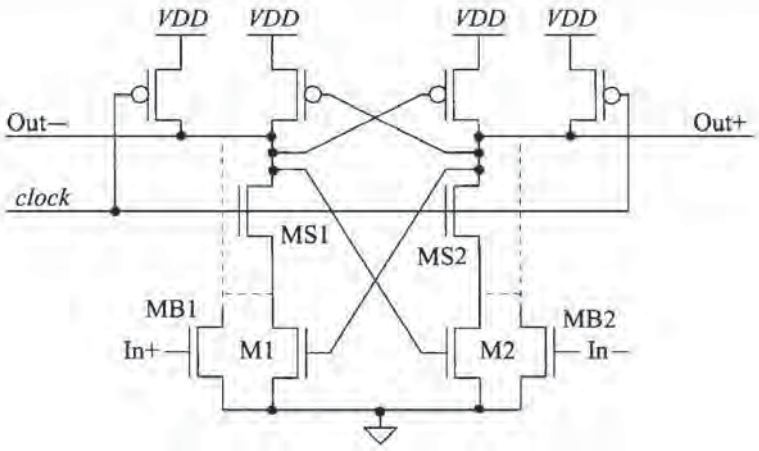



**Figure 16.31** Creating an imbalance in a sense amplifier.

Next let's consider connecting MB1/MB2 to the sources of MS1/MS2 (drains of M1/M2). When *clock* goes low, the drains of MB1/MB2 are pulled low through M1/M2. No current flows in the comparator when *clock* is low (unlike when the drains of MB1/MB2 are connected to the drains of MS1/MS2). The gain is much lower because now MB1/MB2 are operating in the triode region when *clock* goes high. The potential for large current flow is still present when *clock* is high but, now, the maximum value on the drain of MB1/MB2 is $VDD - V_{THN}$. And so the potential for MB1/MB2 to move into the triode region can result in a smaller output current.

We seem to have a problem. To reduce the possibility of metastability, we need to increase the gain in series with the signal path (increase the *W/L* ratio of MB1 and MB2; that is, reduce their switching resistance). However, this results in larger power dissipation. How can we keep power down while maximizing the sensitivity of our sense amplifier? While we can use long *L* devices for MB1 and MB2 so they never pull significant current (at the cost of sensitivity and speed) or have a separate pre-amplifier to generate the input signals, let's look at some other ideas (noting that no design is perfect but a trade-off between power, speed, and sensitivity).

Figure 16.32 shows one idea for creating an imbalance without the possibility of significant current flow. MB1 and MB2 are used to create an imbalance in the gate-source voltages of M1/M2. Since, prior to switching, the voltage on the gates of M1/M2 is *VDD*, we can still have good sensitivity even though MB1 and MB2 are operating in the triode region (look like resistors). The large voltage dropped across $V_{GS,M1}$ and $V_{DS,MB1}$ ensure that good gain. It is important, however, that the inputs signals are $> V_{THN}$ to ensure that the circuit operates properly. The sensing fails if this isn't the case because neither MB1 or MB2 can turn on and provide the needed path to ground for proper logic level signals. As long as the inputs are above the threshold voltage, there aren't any dynamic nodes (*no memory in the sense amplifier*). The sources of MS1 and MS2 are still pulled to ground when *clock* is low. *Kickback noise* is reduced using this topology because the inputs are isolated from the latch by MB1/MB2. Kickback noise is still present and must be considered when designing and simulating this sense amplifier.

IPR2023-00697
Theta EX2019

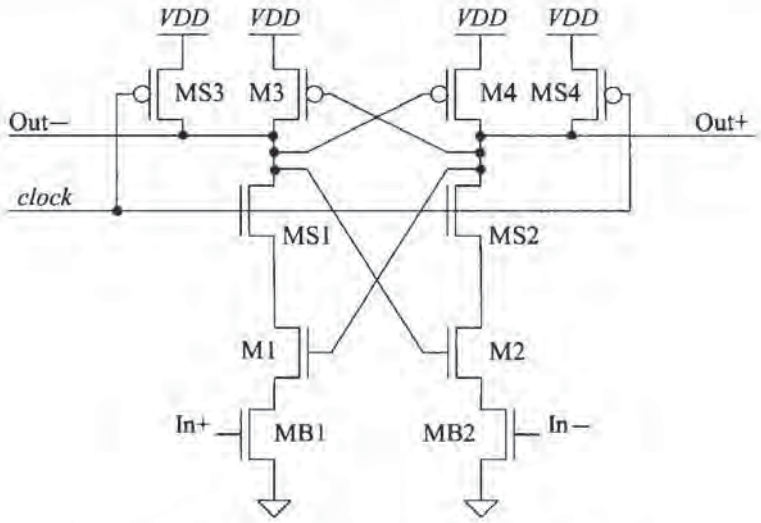



**Figure 16.32** Reducing power in a sense amplifier.

Note that we might try to design our comparator to operate over a wide range of input voltages by adding PMOS devices in series with the sources of M3/M4, as seen in Fig. 16.33. We know that MB1–MB4 must be capable of turning on when *clock* goes high. If either pair can't turn on, then the latch won't be able to generate full logic levels. For the design in Fig. 16.33, the input signals would have to fall within $V_{THN}$ and $VDD - V_{THP}$ (more restricted than the circuit in Fig. 16.32). Also, and perhaps more importantly, MB3/MB4 can't generate an imbalance in the latch. If the sources of M1/M2 are connected to ground so that MB1/MB2 don't generate the imbalance, then when *clock* goes high, MB3 and MB4 are off. They won't turn on until the NMOS pair M1/M2 turns on and drops the gates of M3/M4 below $VDD - V_{THP}$. For great differences in the input signals, the comparator may still function correctly. However, for the majority of the input signal differences, the resulting output logic levels is determined by the matching between M1/M2, e.g., the one with the lower "actual" $V_{THN}$ will turn on faster.

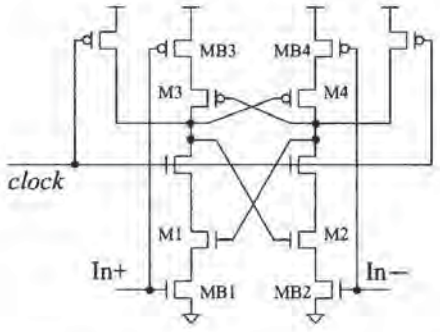



**Figure 16.33** A bad design: how not to get wide-swing operation.

IPR2023-00697
Theta EX2019

**Example 16.1**
Suppose that it is suggested that the gates of M3/M4 be tied to the drains of M1/M2 or MB1/MB2, respectively, in the sense amplifier (clocked comparator) seen in Fig. 16.32. Comment on the concerns.

If the gates of M3/M4 are tied to the drains of M2/M1, then, when *clock* goes high, a significant contention current flows in the latch. As discussed earlier, it is desirable that one side of the latch (either the NMOS or PMOS pairs) is off when the latch is clocked in order to reduce power. The same concerns are present when these gates are connected to the drains of MB1/MB2.

We might try to eliminate MS3/MS4 when we tie the gates of M3/M4 to the drains of MB1/MB2. Because the drains of MB1/MB2 always move to ground when *clock* is low (again assuming the inputs are > $V_{THN}$), M3/M4 will turn on and pull the outputs high. When *clock* goes high, though, it's impossible to shut M3/M4 off. Because the maximum voltage on the sources of MS1/MS2 is *VDD* – $V_{THN}$, the gates of M3/M4 can't go to *VDD* to shut one of the devices off. ■

*Increasing the Input Range*

Figure 16.34 shows adding, to our basic sense amplifier in Fig. 16.32, MOSFETs MB3–MB8 so that the circuit will function with input signals ranging from 0 to *VDD* (actually it will operate correctly with input signals beyond the power supply rails). We know current can only flow out of the sources of M1/M2, so our additional circuitry, for creating the imbalance, must be capable of sinking current from these sources (and so we'll add MB3 and MB4 for this reason). Next, we know that MB1/MB2 function fine for creating the imbalance if the input signals are above $V_{THN}$. However, if the input signals are below $V_{THN}$, they turn off. To level-shift the input, let's use the MB5–MB8. If the input signals are < $V_{THN}$, then MB7 and MB8 are on. A difference in the input voltages causes different currents to flow in MB5 and MB6. The different currents flowing in these transistors results in different voltages across each one. This voltage difference is then used to create an imbalance in MB3 and MB4. Unfortunately, the addition of MB5–MB6 contradicts our earlier statement that, "*For minimum power it's important that there are no DC paths from VDD to ground except during switching times.*" There is a DC path through MB8/MB6 and MB7/MB5. To lower the power, we can increase the length of MB5 and MB6 so that the continuous current flowing through the DC path is lessened.

*Simulation Examples*

Let's simulate the operation of the circuit in Fig. 16.32. We know that the outputs of the sense-amp go high every time *clock* goes low. To make the outputs change only on the rising edge of *clock*, we can use the SR latch discussed in Ch. 13 on the output of the circuit, Fig. 16.35. Now, with the addition of the NAND gates, when the sense amplifier's outputs are high, the outputs of the latch don't change from the previous decision (made on the rising edge of clock).

Figure 16.36 shows the current supplied by *VDD* for the circuit in Fig. 16.35. While the current seen in this figure includes the current supplied to the NAND gates, it still should be compared to Fig. 16.29. Notice in Fig. 16.36 that current flows only during the times the clock changes states (unlike the current flow in Fig. 16.29).

IPR2023-00697
Theta EX2019

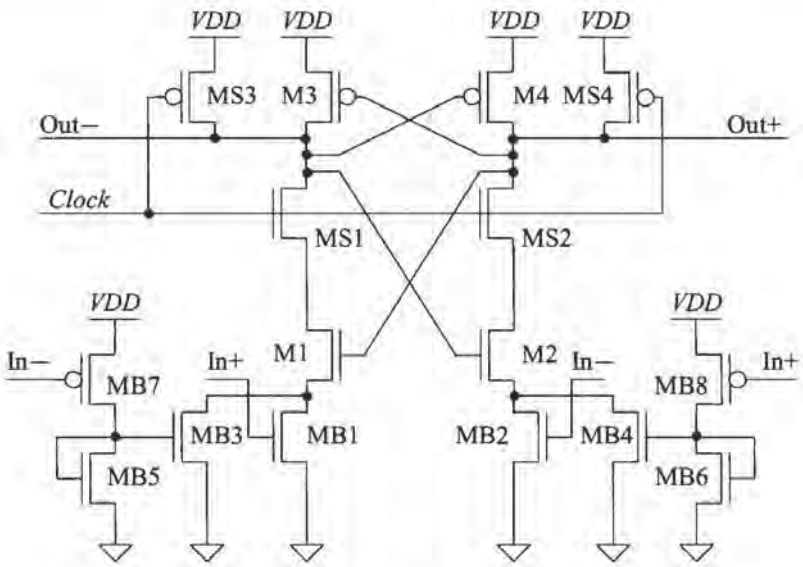



**Figure 16.34** Rail-to-rail input signal range. See also Fig. 27.16.

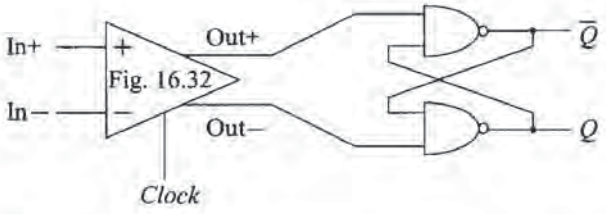



**Figure 16.35** Adding an SR latch so output only changes on rising edge of the clock.

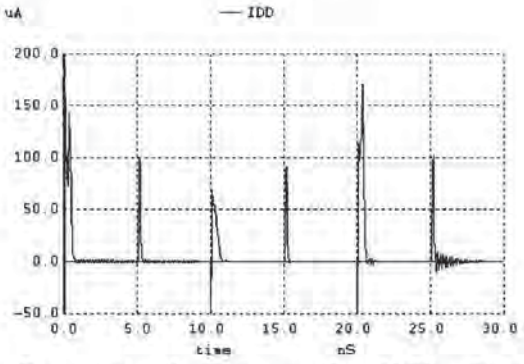



**Figure 16.36** Regenerating the plot seen in Fig. 16.29 with the sense amplifier from Fig. 16.35. The current includes the switching current from the NAND gates.

IPR2023-00697
Theta EX2019

Next let's look at the kickback noise. The clock feedthrough noise is small for this topology because the *clock* is isolated from the input by three transistors. Let's put the sense amplifier in the topology seen in Fig. 16.28 where the inputs are driven to the rails by long-length inverters. The simulation results are seen in Fig. 16.37. The kickback noise is approximately 5 mV.

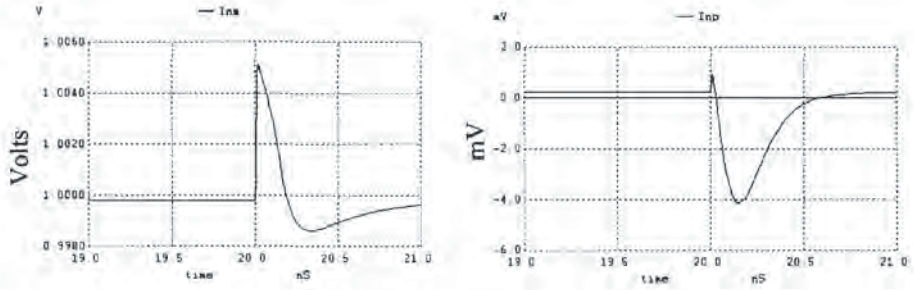



**Figure 16.37** Kickback noise.

Figure 16.38 shows the operation of the circuit in Fig. 16.35 where the outputs only change on the rising edge of *clock*. The input signals are the same ones that were used in generating Fig. 16.27 (although the simulation was 20 ns longer in Fig. 16.38). The sensitivity of this sense amplifier is better than 10 mV. The sensitivity can be improved by increasing the lengths of MB1/MB2 (so that they have a large resistance). The drawback of the improved sensitivity is the increased time it takes to pull one of the outputs to ground (the output transition time). The sensitivity can be increased without this penalty by adding a preamplifier (e.g., a differential amplifier) between the sense amplifier and the inputs, see Fig. 27.16.

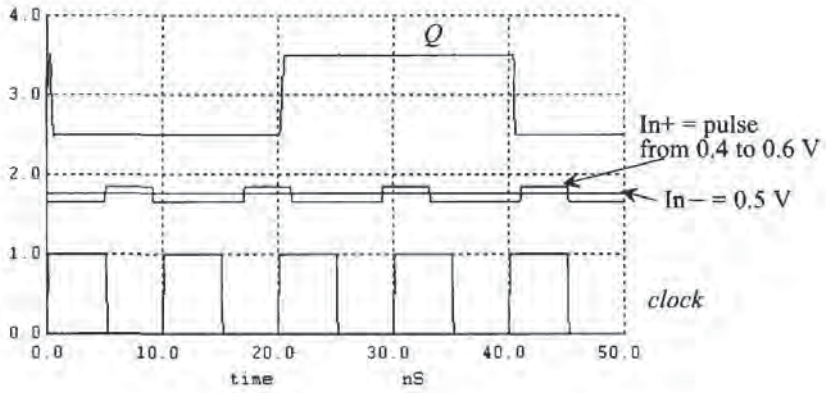



**Figure 16.38** The operation of the circuit in Fig. 16.35.

IPR2023-00697
Theta EX2019

### 16.2.2 Row/Column Decoders

Reviewing the memory block organization in Fig. 16.25, we may wonder how to address one or more of the 256kbit subarrays. When we say "address," for the row lines, what we mean is that we are driving one, and only one, of the row lines in one or more of the 256k subarrays to a voltage greater than $VDD + V_{THN}$ (to fully turn the access MOSFET on), while holding all other row (word) lines at ground. To accomplish this selection, a *row address decoder* is used. Once the word line has transitioned high and the sensing is complete, the data from the memory cells is present on the bit lines (as discussed in detail in Sec. 16.1). To select the addressed data from one or more of these bit lines (in one or more memory arrays), a *column decoder* is used.

In a large (capacity) memory the chip's address pins are multiplexed, that is, the same set of pins are used for both the row and column addresses. A separate clock signal is used to strobe in either the row or the column addresses, at different times, into some hold registers. In older DRAMs, for example, the falling edge of $\overline{RAS}$ (row address strobe) was used to clock in the row address and the falling edge of $\overline{CAS}$ was used to clock in the column address. The outputs of the row address hold register (column address hold register) are decoded and used to select a row (column) line, Fig. 16.39.

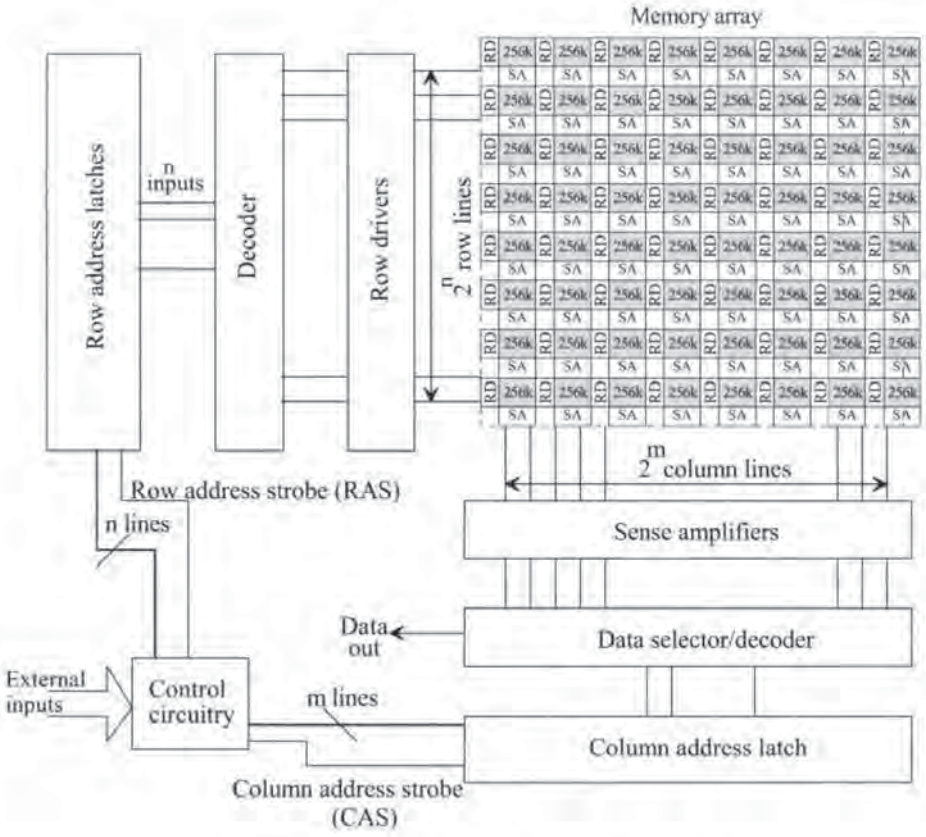



**Figure 16.39** Detailed block diagram of a RAM.

IPR2023-00697
Theta EX2019

The length of the column (m-bits in the Fig. 16.39) and row (n-bits in Fig. 16.39) address words depends on the size of the memory and the word size. For example, a 1-Gbit memory can be organized in x1 (by one), x2 (by two), x4 (by four), etc., configuration. The "by four" for example, simply signifies the word size used in the memory's input/output data path is 4-bits. A 1-Gbit memory in a x4 configuration simply indicates that 256-Mwords can be addressed where each word is 4-bits. In a x1 configuration 1-Gbits of data can be accessed ($2^{30}$ = 1-G = 1,073,741,824). We would then have 15 address pins for $2^{15}$ different row and column addresses (remembering that the address pins are shared). The next question then becomes how to decode the addresses and select the appropriate row(s) and column(s).

*Global and Local Decoders*

Looking at Fig. 16.39, we see that we decode the outputs of the row address latches and the column address latches and then feed the outputs to the subarrays. This is called a *global decode*. If the memory size is, once again, 1-Gbit (in a x1 configuration), then $2^{15}$ (= 32,768) row line wires and $2^{15}$ column line wires are fed to the memory arrays. If each memory array is 256-kbits (512 word lines and 512 column lines), then we need 4,096 arrays to get a 1-Gbit memory (64 by 64 memory arrays in Fig. 16.39). When the output of the row decoder goes high, it turns on a word line in the 64 memory arrays of the addressed row. A total of 32k columns then have data sitting on them. Because our memory is going to take only 1 bit of data and feed it to the output, we have, perhaps, wasted a lot of power. (In many topologies this large amount of data is called a "page." Once the row is opened it can be quickly read out of the memory by simply changing the column address.) The other issue with the global decoder is that a separate layer of metal is needed to route the decoded signals throughout the entire chip (adding process complexity). The big benefit of global decoding is reduced chip size.

A *local decoder* takes the input addresses and, as the name says, decodes them locally at the memory array. The 15 bits, for the example given above, are routed to each subarray so that they can be decoded locally. This takes up a considerable amount of space on the chip (having a decoder at each 256k array, for example) but doesn't call for increased process complexity. In most practical designs a combination of local and global techniques are used. Some of the address bits are decoded globally while others are decoded locally. The global decoder enables an array or (arrays), while the local decoder selects the array's row line and column line. A static decoder is seen in Fig. 16.40. Figure 16.41 shows how the 10-bit address (for 1,024 word or row lines) is connected to each decoder element.

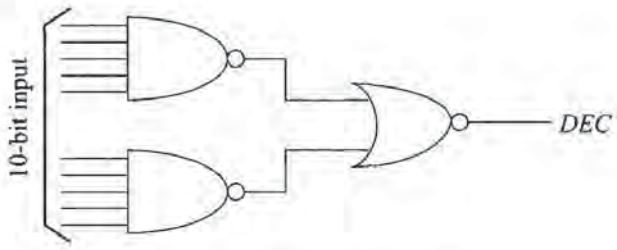



**Figure 16.40** A static decoder.

IPR2023-00697
Theta EX2019

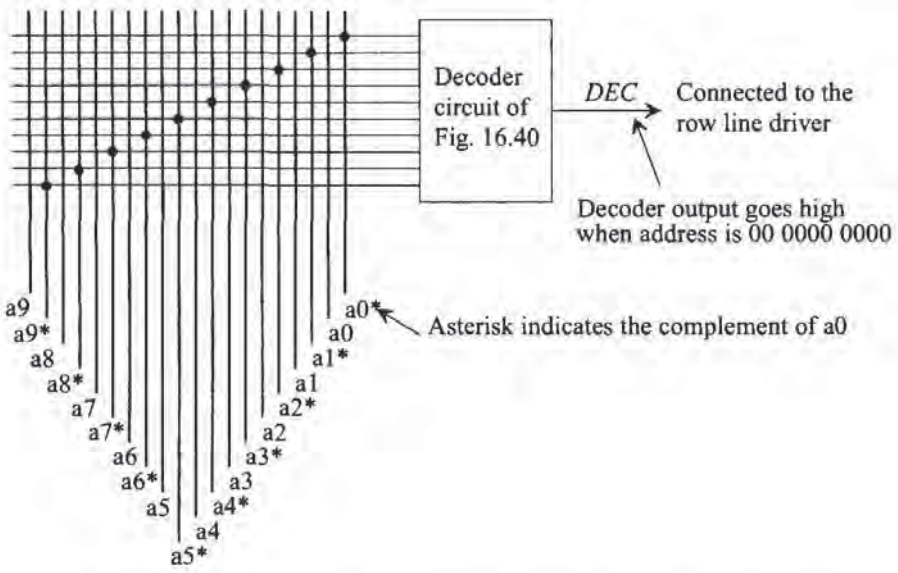



**Figure 16.41** How the address lines are connected to a decoder element.

For the column decoder, we know that we have to be able to write or read data to the memory cells (not just select a column). For this reason pass transistors are used on the output of the decoder, as seen in Fig. 16.42. The pass transistor doesn't pass a logic one to full levels. We lose a $V_{THN}$ with body effect when writing or reading a one. The sense amplifier may be used to pull the bit line up to a full logic one. Similarly, on the output of the column decoder a sense amplifier may be used to pull the output line to full logic levels. Finally, note that we drew the column decoder and its outputs going horizontally. In practice, the outputs of the column decoder run in the same direction as the columns themselves in order to save space.

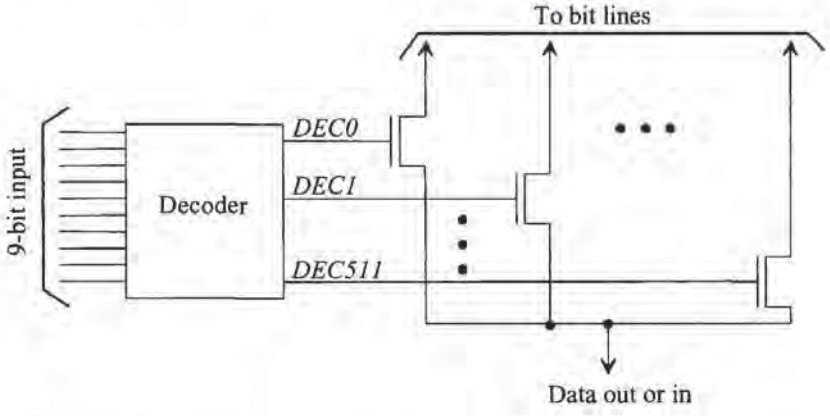



**Figure 16.42** Addition of pass transistors to implement a column decoder.

IPR2023-00697
Theta EX2019

*Reducing Decoder Layout Area*

To reduce the layout area of address decoders, a pass-transistor based decoder, see Fig. 16.43a, can be used. The concerns with using this decoder are that the unselected outputs are not actively driven high or low (they are floating) and, once again, as NMOS devices they don't pass a logic one to a full logic level (that is, *VDD*). This means that on the output of the decoder, as part of our row driver, we need a circuit that, when not selected, pulls the output of the decoder low. At the same time, to maximize the noise margins, the switching point of the driver needs to be reduced. For example, with $VDD = 1$ V and $V_{THN} = 0.35$ V (high because of body effect), the selected output of the decoder swings from 0 to 0.65 V. The switching point of the row driver should lie in the middle of this swing at 0.3125 (ideally), to maximize the noise margins, Fig. 16.43b.

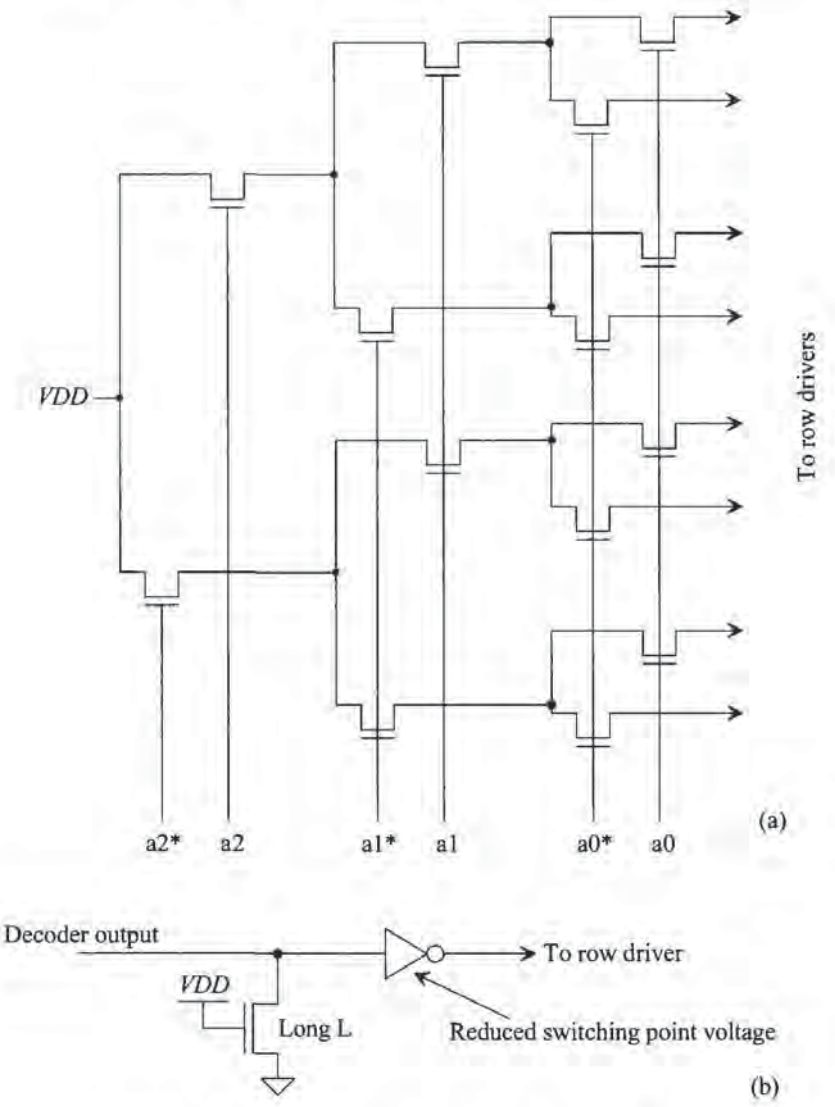



**Figure 16.43** A tree decoder used to reduce layout area.

IPR2023-00697
Theta EX2019

Another technique to reduce decoder layout size uses the precharge-evaluate (PE) logic, Fig. 16.44, discussed in Ch. 14. This figure shows a 3-bit input decoder. The output of the circuit goes high when the input address is 000. Using the long-length keeper MOSFET makes the dynamic circuit operate as a static circuit. Variations of this circuit are common in commercially available memories.

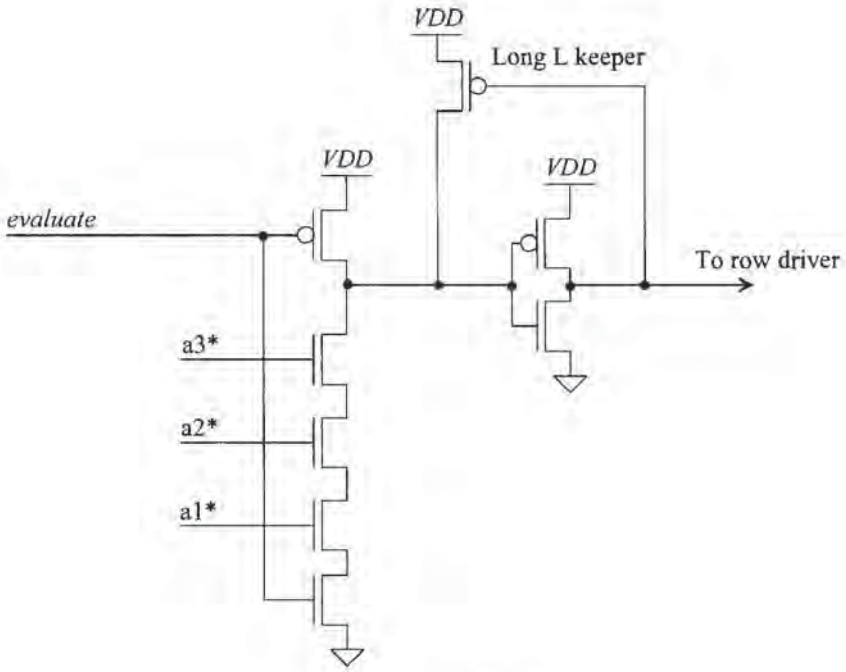



**Figure 16.44** 3-bit PE decoder.

### 16.2.3 Row Drivers

In most memories a single NMOS pass transistor switches data into or out of the memory cell. To fully turn this pass transistor on, the row line is driven to a pumped voltage (a voltage outside of the power supply rails that is generated using the on-chip charge pump circuit discussed in Ch. 18). We'll call this pumped voltage *VDDP*. For our purposes this voltage must be greater than $VDD + V_{THN}$ (with body effect). In the following discussion we'll assume a *VDDP* of 1.5 V (since our *VDD* is 1 V and $V_{THN}$ is 280 mV without body effect).

Examine the inverter circuit seen in Fig. 16.45. The input to the inverter connected to the word line swings from 0 to *VDD* (1 V). Note that the PMOS is sitting in its own well. Both its source and body are tied to *VDDP*. When the input to this inverter is low (0 V), the PMOS is on and the NMOS is off. The word line is driven to *VDDP* (which is what we want). Unfortunately when this inverter input is high (= *VDD*), the NMOS turns on but the PMOS can't shut off. The source gate voltage of the PMOS is 0.5 V (above the PMOS's threshold voltage). In this section we discuss word-line drivers, knowing that a simple inverter can't be used.

IPR2023-00697
Theta EX2019

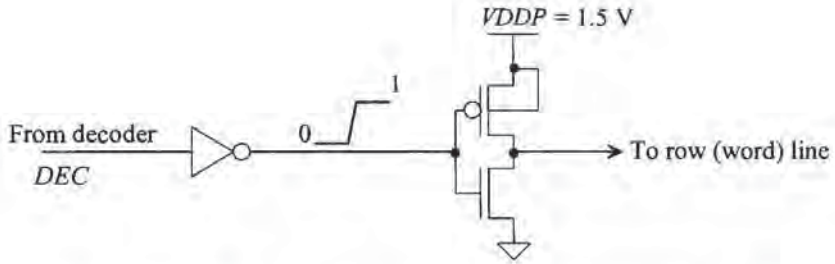



**Figure 16.45** Problems with using an inverter for a row driver.

A row driver based on the inverter is seen in Fig. 16.46. When the decoder output is low, M1 is off and M2 is on. This causes M3 to turn on, driving the gate of M4 to *VDDP* so that it shuts off. When the decoder output goes high, M1 turns on and M2 shuts off. The gate of M4 is pulled to ground turning it on. This causes the word line to move to *VDDP* and M3 to shut off. This circuit works well but there can be a significant contention current when M3/M4 are turning on. To avoid this, the pumped voltage can be a clocked signal that goes high *after* the output of the decoder has transitioned. The idea is seen in Fig. 16.47.

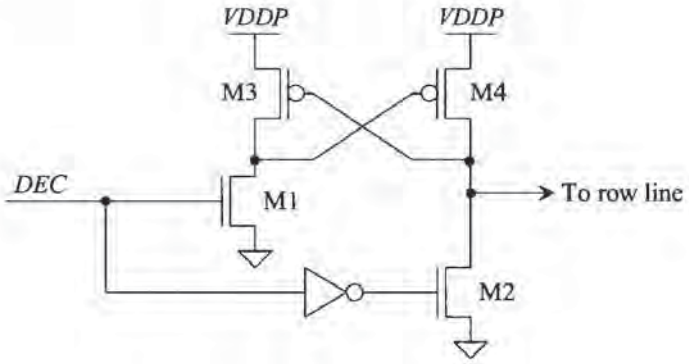



**Figure 16.46** A CMOS word line driver.

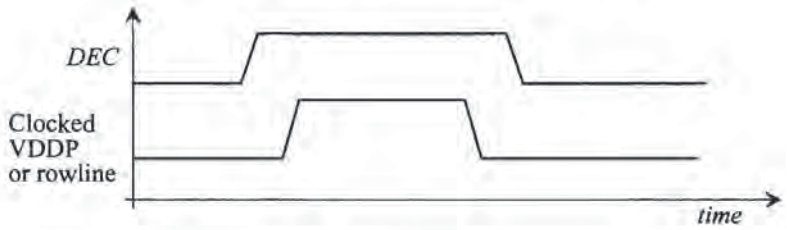



**Figure 16.47** Using a clocked VDDP to reduce contention current in a row line driver.

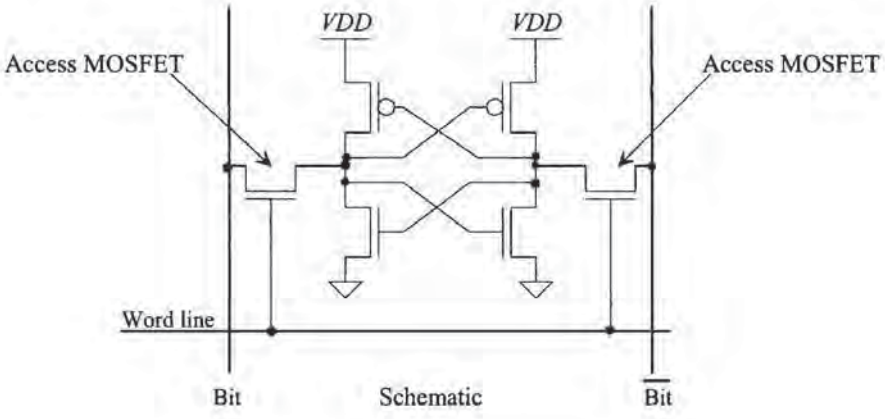
IPR2023-00697
Theta EX2019

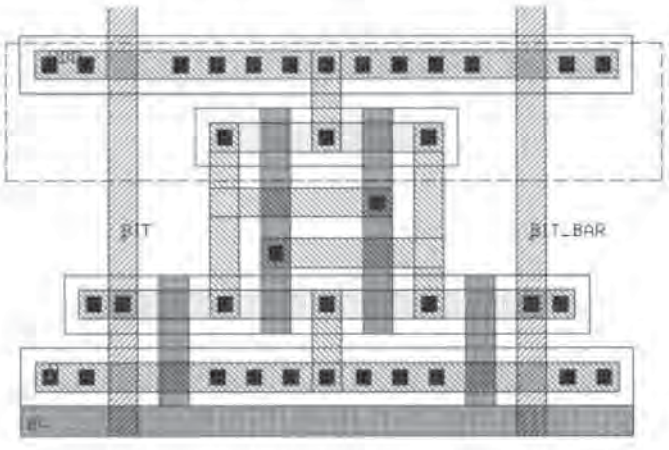
## 16.3 Memory Cells

We've already looked at the DRAM memory cell in detail. In this section we'll take a look at the static RAM cell (SRAM), the erasable programmable read-only memory (EPROM) cell, the electrically erasable programmable read-only memory (EEPROM) cell, and the Flash memory cell.

### 16.3.1 The SRAM Cell

The schematic and layout of a six-transistor SRAM memory cell is seen in Fig. 16.48. This is, as its name implies, static, meaning that as long as power is applied to the cell it will remember its contents (unlike the DRAM cell, which loses its memory after a short time). The basic cross-coupled inverter latch should be recognized in the topology. To access the cell, the word line goes high and turns on the access MOSFETs. The bit lines are driven in complementary directions with a "strong" driver for writing. When the word line goes low, the datum is latched in the cell.




Layout

**Figure 16.48** The six-transistor SRAM memory cell.

IPR2023-00697
Theta EX2019

The sizing of the access MOSFETs is important. If they are too weak, the bit lines won't be able to flip the state of the cell when writing. If they are too strong, the layout area can be large (noting that bit lines are often precharged high in an SRAM so that the access devices are initially off during sensing). To reduce the layout area, a cell that doesn't use PMOS devices can be used, Fig. 16.49. The cell size is decreased by eliminating the n-well and using high resistance poly n+/p+ resistors. The layout of the polysilicon resistor is also seen in Fig. 16.49. The resistor can be thought of as a leaky bipolar transistor. Typical resistance values approximately 10 MΩ (or more). The CMOS SRAM cell dissipates little static power, while the resistor/n-channel SRAM cell dissipates $VDD^2/R_{poly}$.

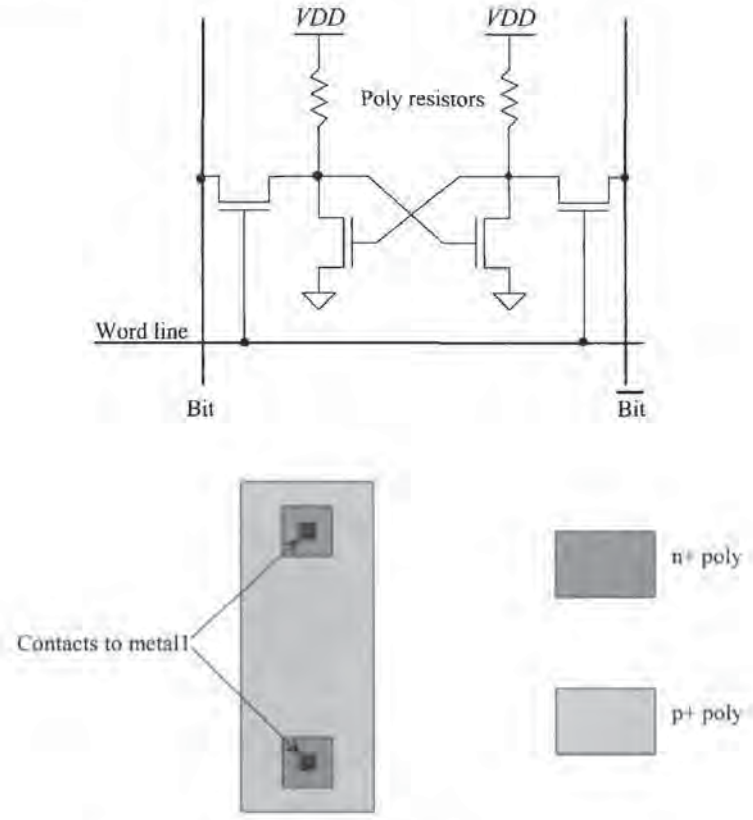



**Figure 16.49** SRAM memory cell with poly resistors.

### 16.3.2 Read-Only Memory (ROM)

ROM is the simplest semiconductor memory. It is used primarily to store instructions or constants in a digital system. The basic operation of a ROM can be explained with the ROM memory schematic shown in Fig. 16.50. Remembering that only one word line (row line) can be high at a time, we see that $R_1$ going high causes the column lines $C_1$, $C_2$, and $C_4$ to be pulled low. Column lines $C_3$ and $C_5$ are pulled high through the long L

IPR2023-00697
Theta EX2019

MOSFET loads at the top of the array. If the information that is to be stored in the ROM memory is not known prior to fabrication, the memory array is fabricated with an n-channel MOSFET at every intersection of a row and column line (Fig. 16.51a). The ROM is programmed (PROM) by cutting (or never fabricating) the connection between the drain of the MOSFET and the column line Fig. (16.51b). Because it is not easy to program ROM, it is limited to applications where it is mass produced.

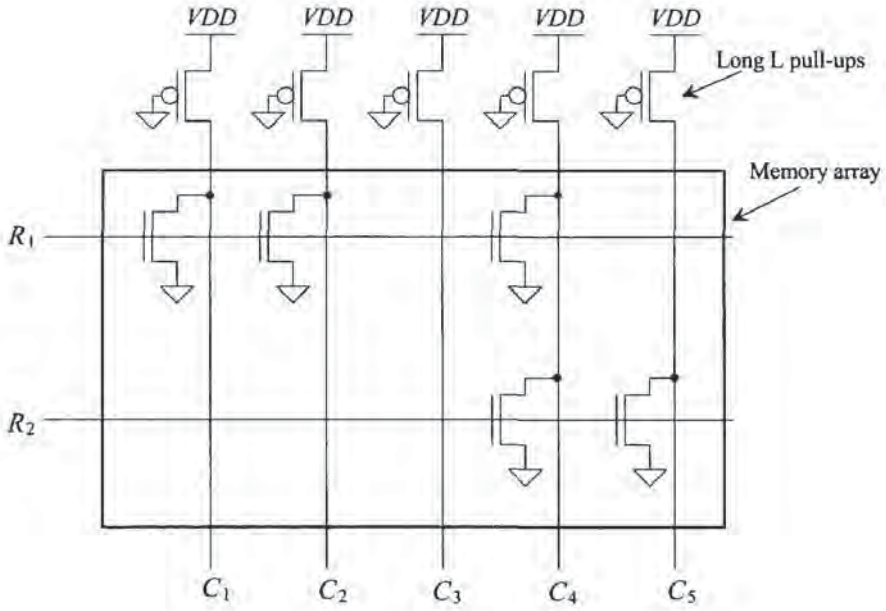



**Figure 16.50** A ROM memory array.

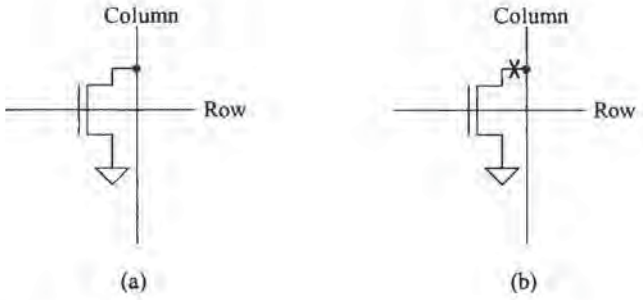



**Figure 16.51** (a) n-channel MOSFET at the intersection of every column and row line and (b) eliminating the connection between the drain and column line to program the ROM.

IPR2023-00697
Theta EX2019

### 16.3.3 Floating Gate Memory

A MOSFET made with two layers of poly is seen in Fig. 16.52 (along with its schematic symbol and a typical layout). As seen in the figure, the poly1 is floating, that is, not electrically connected to anything. A dielectric surrounds this floating island of poly1. With the gate oxide on the bottom, a thin oxide insulates the MOSFET from the poly2 (word/row line) above. Poly2 is the *controlling gate* of the transistor (the terminal we drive to turn the MOSFET on). We are going to make a memory element out of this cell by changing, adding or removing, the charge stored on the floating gate.

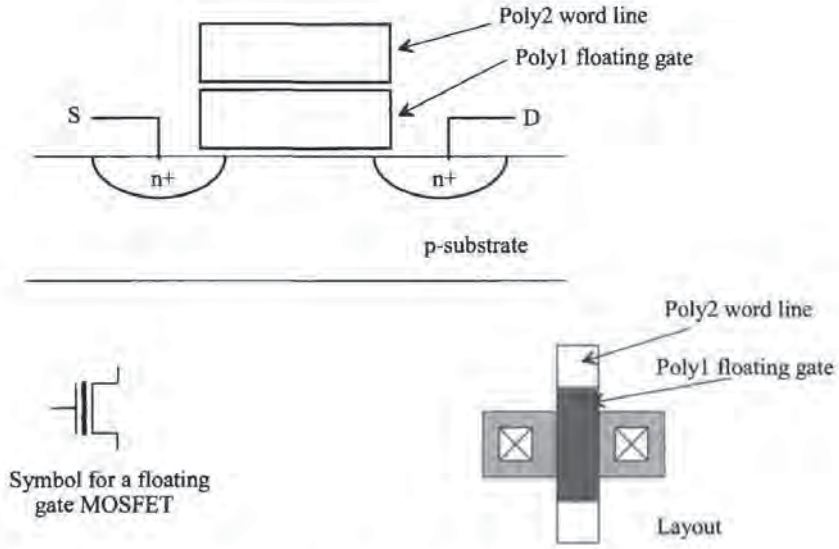



**Figure 16.52** A floating gate MOSFET, its symbol and layout.

Figure 16.53 shows the difference between the *erased* state and the *programmed* state in a floating gate memory. The erased state, the state of the memory when it is fabricated (that is, before we force charge onto the floating gate), shows normal MOSFET behavior. Above the threshold voltage, the device turns on and conducts a current. When the cell is programmed, we force a negative charge (electrons) onto the floating gate (how we force this charge will be discussed shortly). This negative charge attracts a positive charge beneath the gate oxide. The result is that a larger controlling gate voltage must be applied to turn the device on (the threshold voltage increases).

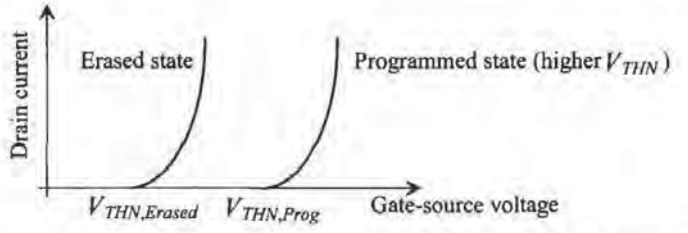



**Figure 16.53** Programmed and erased states of a floating gate memory.

IPR2023-00697
Theta EX2019

*The Threshold Voltage*

From Ch. 6 we can write the threshold voltage of a single-poly gate MOSFET as

$$V_{THN} = -V_{ms} - 2V_{fp} + \frac{Q'_{b0}}{C'_{ox}} \tag{16.10}$$

where we haven't included the shifts due to the threshold voltage implant, $Q'_c$, or any unwanted surface state charge, $Q'_{ss}$ (to keep the equations shorter). Looking at Fig. 16.54, we see that the effective oxide capacitance from the controlling gate to the channel has decreased from $C'_{ox}$ for a single-poly gate MOSFET to $C'_{ox}/2$ for a dual poly gate MOSFET. Our threshold voltage, for a floating gate MOSFET, can then be written as

$$V_{THN,Erased} = -V_{ms} - 2V_{fp} + 2 \cdot \frac{Q'_{b0}}{C'_{ox}} \tag{16.11}$$

If the term $Q'_{b0}/C'_{ox}$ is approximately 50 mV (a typical oxide thickness used in a floating gate memory is 100 Å), then the erased threshold voltage, $V_{THN,Erased}$ (see Fig. 16.53) is only 50 mV larger than the $V_{THN}$ of a normal (single poly gate) MOSFET.

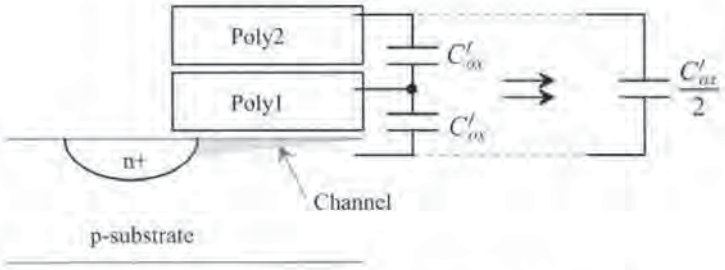



**Figure 16.54** Oxide capacitance estimation for calculating threshold voltage..

Next consider what happens if we trap a negative charge on the floating poly1 gate, Fig. 16.55. The threshold voltage with this trapped charge, $Q'_{poly1}$, is shifted to

$$V_{THN,Prog} = -V_{ms} - 2V_{fp} + 2 \cdot \left( \frac{Q'_{b0}}{C'_{ox}} + \frac{|Q'_{poly1}|}{C'_{ox}} \right) \tag{16.12}$$

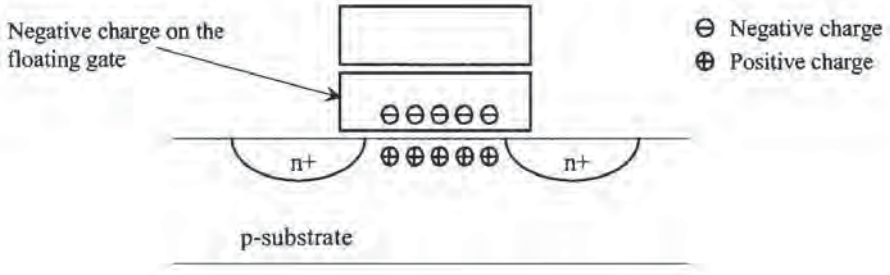



**Figure 16.55** Trapped negative charge on the floating gate.

IPR2023-00697
Theta EX2019

*Erasable Programmable Read-Only Memory*

Erasable programmable ROM (EPROM) was the first floating gate memory that could be programmed electrically. To erase (return the cell to its fabricated state, that is, leave no trapped charge on the floating gate), the cell is exposed to ultra-violet light through a quartz window in the top of the chip's package. The ultra-violet light increases the conductivity of the silicon-dioxide surrounding the floating gate and allows the trapped charge to leak off. The inability to erase the EPROM electrically has resulted in its being replaced by Flash memory (discussed later).

Programming the EPROM relies on *channel hot-electron* (CHE) injection. CHE is accomplished by driving the gate and the drain of the MOSFET to high voltages (say a pumped voltage of 25 V), Fig. 16.56. The high voltage on the drain of the device causes hot electrons (those with significant kinetic energy) to flow in the channel. A large positive potential applied to the gate attracts some of these electrons to the floating gate. The electrons can penetrate the potential barrier between the floating gate and channel because of their large energies.

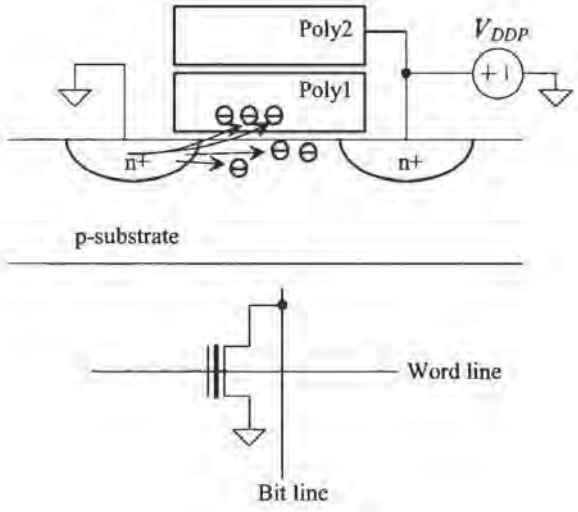



**Figure 16.56** How charge is trapped on the floating gate using channel hot electron (CHE) injection.

*Two Important Notes*

When we are programming the floating gate device, the accumulation of electrons on the floating gate causes an increase in the device's threshold voltage. The more electrons that are trapped the higher the threshold voltage. This increase in threshold voltage causes the drain current to decrease. The decrease in drain current then reduces the rate at which the electrons are trapped on the floating gate oxide. If we apply the programming voltages for a long period of time, the drain current drops to zero (or practically a small value). Because of this feedback mechanism, the programming is said to be *self-limiting*. We simply apply the high voltages for a long enough time to ensure that the selected devices are programmed.

IPR2023-00697
Theta EX2019

Next examine Fig. 16.57. When we are programming a row of cells, we drive the word line to a high voltage. If the cell is to remain erased, we simply leave the corresponding bit line at ground. If the cell is to be programmed, we drive the bit line to the high voltage. For CHE, both the drain and gate terminals of the floating device must be at a high voltage. This is important because it removes the need for a select transistor (something we'll have to use in other programming methods to keep from programming an unselected cell).

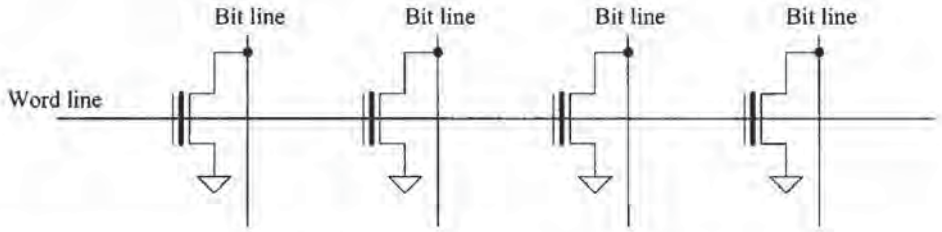



**Figure 16.57** Row of floating gate devices. When programming the word line is driven to a high voltage. To program a specific cell, its bit line is also driven to a high voltage. To leave the cell erased, the bit line is held at ground.

*Flash Memory*

By reducing the thickness of the oxide from, say, 300 Å (a typical value used in an EPROM) to 100 Å, Fowler-Nordheim tunneling (FNT) can be used to program or erase the memory cell. Floating gate memory that can be both electrically erased and programmed is called electrically erasable programmable ROM (EEPROM). Note that this name is an oxymoron. If we can electrically *write* to the memory, then it isn't a *read*-only memory. For this reason, and because the rows of memory are generally erased in a *flash* (that is, large amounts of memory, say a memory array, are erased simultaneously and, when compared to the EPROM method of removing the chips from the system and exposing to ultra-violet light to erase, very quickly), we call floating gate memory that can be electrically programmed and erased *Flash* memory.

While CHE and FNT can be used together (CHE for the programming and FNT for erasing) to implement a memory technology, we assume FNT is used for both programming and erasing in the remaining discussion in this chapter.

Figure 16.58 shows the basic idea of using FNT to *program* a device (recall that this means to trap electrons on the floating gate so that the devices threshold voltage increases). The control gate (poly2) is driven to a large positive voltage. For a 100 Å gate oxide this voltage is somewhere between 15 and 20 V. Note that we are assuming that our NMOS devices are sitting in a p-well that is sitting in an n-well (so we can adjust the p-well [body of the NMOS] potential). The electrons tunnel through the thin oxide via FNT and accumulate on the floating gate. Like programming in an EPROM device, this mechanism is self-limiting. As electrons accumulate on the floating gate, the amount of tunneling current falls. Note that if we didn't want to program the device when poly2 is at 20 V we could hold the drain at a higher voltage (not ground) to reduce the potential across the thin gate oxide (aka tunnel oxide).

IPR2023-00697
Theta EX2019

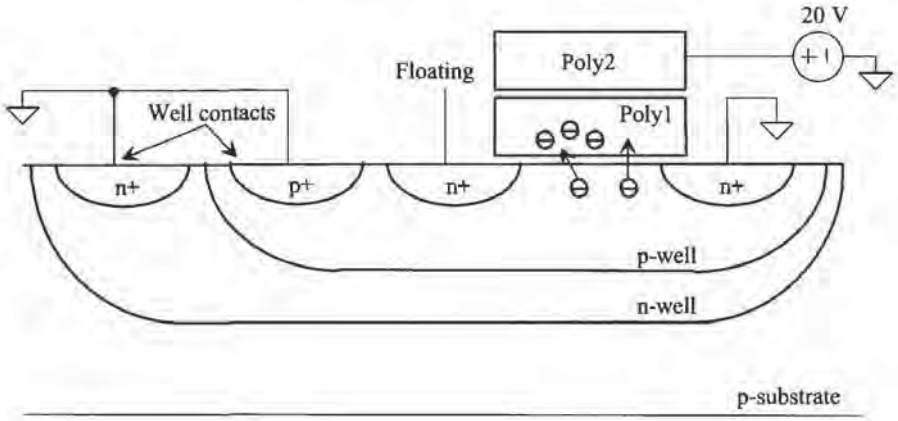



**Figure 16.58** FNT of electrons from the p-well to a floating gate to increase threshold voltage (showing programming).

To *erase* the device using FNT, examine Fig. 16.59. Both the p-well and the n-well are driven to 20 V while the control gate (poly2) is grounded. Electrons tunnel via FNT off of the floating gate (poly1) to the p-well. The source and drain contacts to the device are floating (to accomplish this we will float the bit line and the source n+ outside of the array). Again, the movement of charge is self-limiting (however, there are device issues that can result in over erasing). The tunnel current drops as positive charge accumulates on the floating gate. If the erasing time is long, a significant amount of positive charge can accumulate on the floating gate. This will *decrease* the threshold voltage of the MOSFET. Figure 16.60 shows the programmed and erase states for a Flash memory where a positive threshold voltage indicates that the device is programmed and a negative threshold voltage indicates that the device is erased (we show $\pm 3$ *V* as typical values).

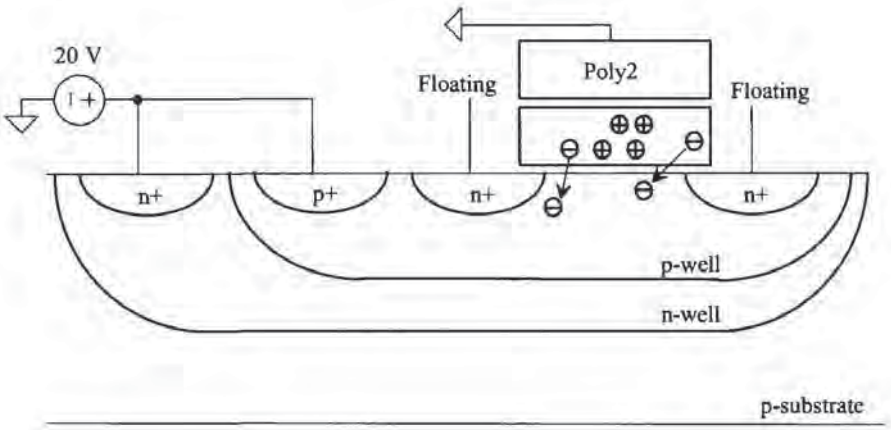



**Figure 16.59** FNT of electrons from the floating gate to p-well to decrease threshold voltage (showing erasing).

IPR2023-00697
Theta EX2019

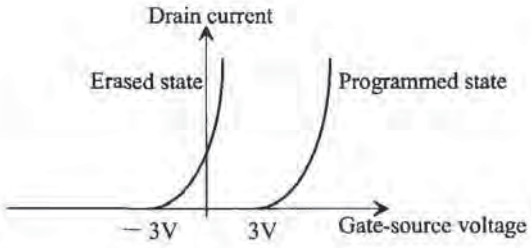



**Figure 16.60** Programmed and erased states of a flash memory.

The schematic and layout of a 4-bit NAND Flash memory cell is seen in Fig. 16.61. The select transistors are made using single poly (normal) MOSFETs. When the cell (all four bits) is erased, the p-well and the n-well are driven to 20 V external to the memory array via the p+ implant. The bit lines and the n+ source connection at the bottom of the layout are floated. The four control gates and the two select MOSFET gates are pulled to ground (so all six poly gates, aka, word lines, are at ground for an erase).

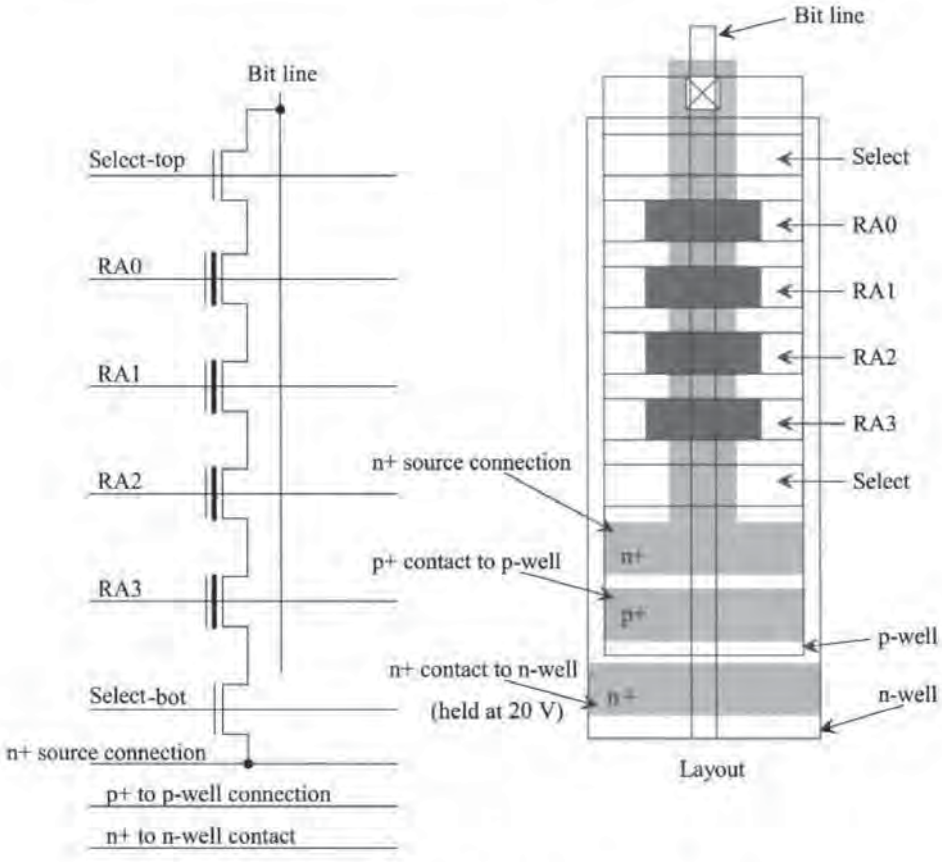



**Figure 16.61** A 4-bit NAND Flash memory cell.

IPR2023-00697
Theta EX2019

To illustrate programming the floating gate MOSFET connected to RA0 (row address 0), examine the connections to the NAND cell seen in Fig. 16.62. The bit line is driven to ground. A voltage, say 20 V, is applied to the gate of the top select gate. The gates of the floating gate MOSFETs connected to RA1–RA3 are driven to 5V. This 5 V signal turns on these devices but isn't so big that FNT will occur in them. The bottom select MOSFET remains off so that there is no DC path from the bit line to ground, Fig. 16.58. The p-well (which is common to all of the cells in the memory array, that is, not just the four in the memory cell) is pulled to ground external to the memory array via the p+ implant. Because the gate, RA0, is pulled to 20 V, as seen in Fig. 16.58, and the drain implant is pulled to ground through the bit line, the device will be programmed (electrons will tunnel through the oxide and accumulate on the floating gate).

The next thing we need to look at before talking about reading the cell is how we keep from programming the adjacent floating gate devices, those also connected to *RA*0, if they are to remain erased. What we need to do is ensure that no FNT occurs in these unselected devices. Figure 16.63 shows how we keep from programming a device. The bit line of the cell that is to remain erased is driven to a voltage that is large enough to keep FNT from occurring. The bottom select MOSFET is off, so there won't be a DC path from the bit line to ground (this is important because all other MOSFETs in the memory cell will be on).

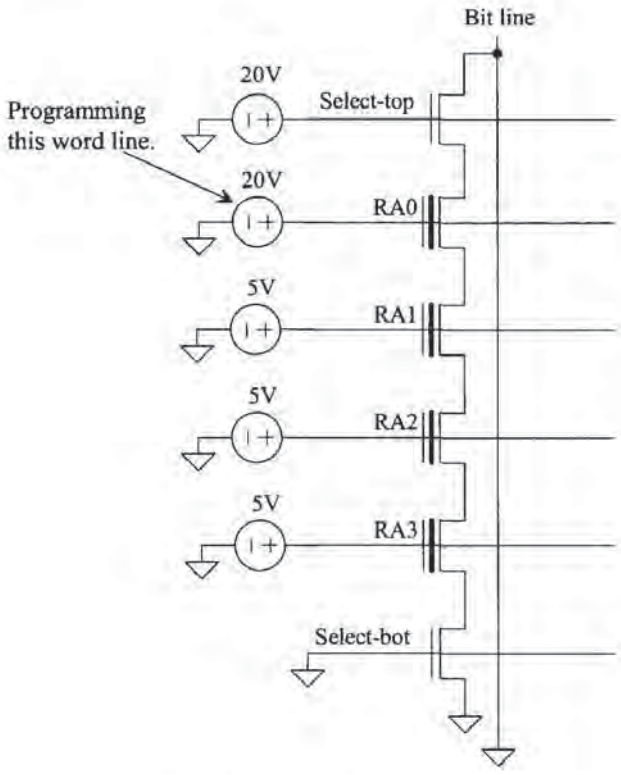



**Figure 16.62** Programming in a Flash NAND cell.

IPR2023-00697
Theta EX2019

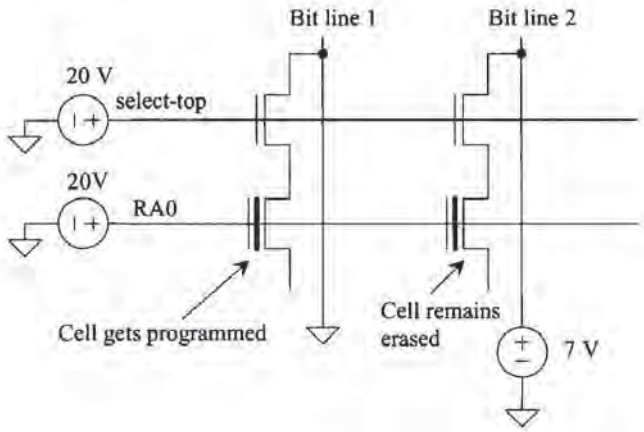



**Figure 16.63** How cells are programmed (or not) in a NAND Flash memory array.

To understand reading a NAND Flash cell, consider the expanded view of Fig. 16.60 shown in Fig. 16.64. If both select transistors are turned on (say 5 V on their gates), the unselected row lines are driven high (say, again, to 5 V), and the selected row line is held at zero volts, then the current difference between $I_{erased}$ and $I_{prog}$ can be used to determine if the (selected) floating gate MOSFET is erased or programmed. If an average current, that is, $(I_{erased} + I_{prog})/2$ is driven into the bit line, an erased cell will keep the bit line at a low voltage. The selected (erased) MOSFET will want to sink a current of $I_{erased}$ when its gate is zero volts. A programmed cell won't be able to sink this current (it will want to sink a current of $I_{prog}$) and so the bit line will go high.

Table 16.1 shows a summary of erasing, programming, and reading a NAND Flash memory cell. Notice that in Figs. 16.58 and 16.59 (during either erasing or programming the cell) the source of the MOSFET is floating. In Fig. 16.58 we can float the source by shutting off the bottom select MOSFET in the NAND stack. However, in Fig. 16.59 if we try to isolate the cell by shutting off the select transistors we see an issue, that is, when 20 V is applied to the p-well, the n+ source/drain implants can forward bias. This is why we must float both the bit line and the n+ source external to the array. Also, note that the top select MOSFET is used to provide better isolation between the memory cell and the bitline (it lowers the capacitive loading on the bit line). Going through the

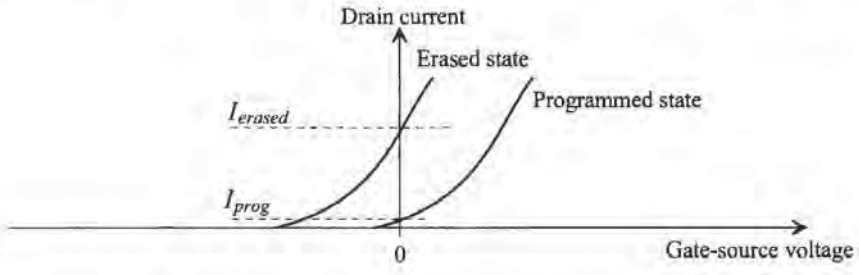



**Figure 16.64** Expanded view showing erased and programmed IV curves.

IPR2023-00697
Theta EX2019

operation of the NAND memory cells, we see that the lower select transistor, from a logical point of view, is all that is needed for basic cell operation.

**Table 16.1** Summarizing NAND Flash cell operation

| Inputs | Erase | Program | Read |
|---|---|---|---|
| Bit line | Floating | 0 V | High or low |
| select_top | 0 V | 20 V | 5 V |
| RA0 | 0 V | 20 V | 0 V |
| RA1 | 0 V | 5 V | 5 V |
| RA2 | 0 V | 5 V | 5 V |
| RA3 | 0 V | 5 V | 5 V |
| select_bot | 0 V | 0 V (so the source of the cell floats) | 5 V |
| n+ source | Floating | 0 V | 0 V |
| p+ well tie-down | 20 V | 0 V | 0 V |
| Comments | Erases entire array since well and word lines are common (Flash). The p-well at 20V will forward bias the n+ source and drain regions. This requires the bit line and n+ source float external to the array. | Programming the RA0 cell. Bit lines of the cells not to be programmed, but on RA0, are driven to 7 V to avoid FNT. Unused word lines driven to 5V. | Reading the contents of RA0. An average current is put into the bit line. |

Before leaving this topic, let's use our short-channel process, which, of course, has only a single poly layer to show how direct tunneling gate current varies with terminal voltages. For our 50 nm, short-channel process used in this book ($t_{ox}$ = 14 Å), the gate current density is (roughly) 5 A/cm$^2$. A 500 nm by 50 nm device will have a gate current of (typically) 50 pA. However, if the potentials of either the drain or source terminals of the MOSFET move further away from the gate's potential, the gate tunnel current will increase. Figure 16.65 shows how the gate current changes in a 10/1 NMOS device (actual width of 500 nm and a length of 50 nm) with the drain and bulk held at ground while the gate voltage is swept from 0 to 2 V. Notice that at a $V_{GS}$ above *VDD* (= 1 V), the gate current becomes much more significant than the 50 pA we calculated for a typical value a moment ago. Figure 16.66 shows that if we hold the drain at 1 V, the gate current is significantly less than the values seen in Fig. 16.65. This is important because we can keep from programming an erased cell (if we were using a floating gate in this technology with oxide thickness of 14 Å) simply by driving its drain (the bit line) to *VDD*. Finally, Fig. 16.67 shows the current when erasing the device. The problem with using direct tunneling for Flash memory is retention. While it is easy to program/erase the floating gate device, it is equally easy for trapped charge to tunnel back off of a floating gate over time.

IPR2023-00697
Theta EX2019

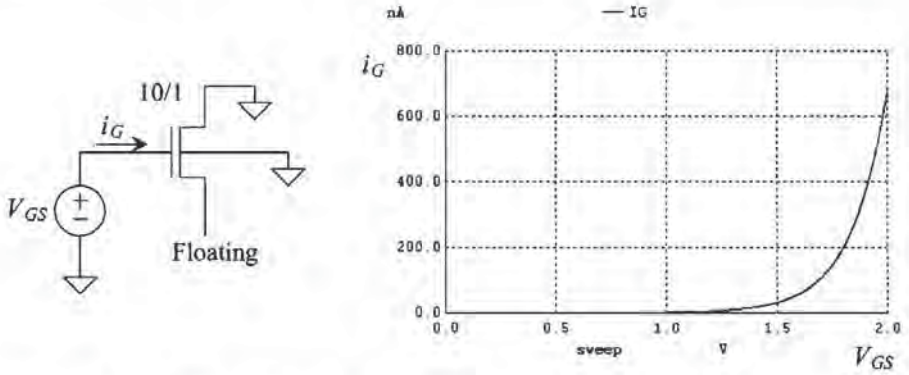



**Figure 16.65** How gate current changes with gate voltage with either the gate or source grounded.

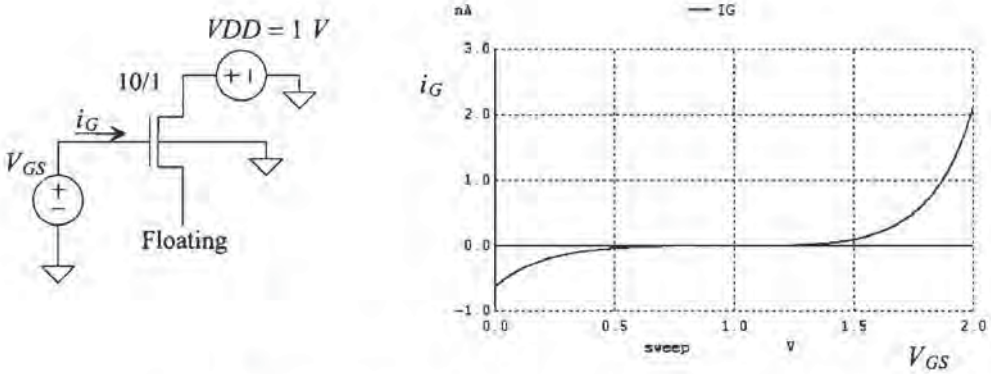



**Figure 16.66** How gate current changes with gate voltage when drain is floating and drain is at VDD.

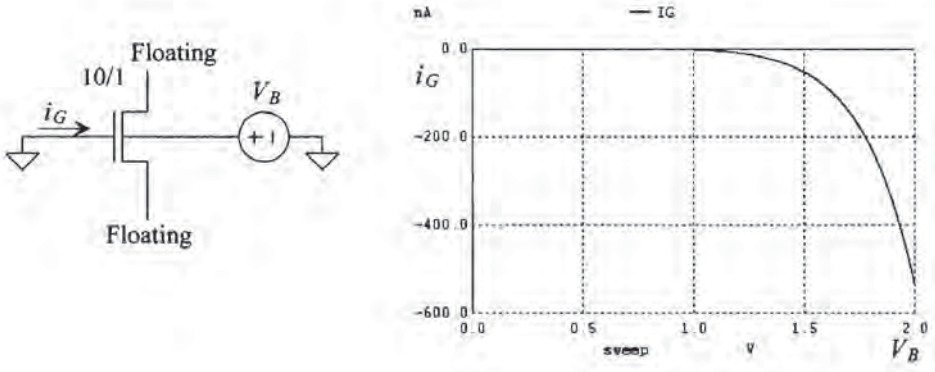



**Figure 16.67** Erasing. Since electrons are being removed from the gate (they are tunneling through the gate oxide), we know the gate current will flow out of the device.

IPR2023-00697
Theta EX2019

An interesting use for floating gate MOSFETs is in the design of analog circuits. Being able to adjust the threshold voltage of a MOSFET can be very useful for low-power or low-voltage design. For example, the input common-mode range of a differential amplifier can be widened by reducing the threshold voltage of the input diff-pair. Also, trimming (removing offsets) can be accomplished when floating gate MOSFETs are used. The interested reader is referred to the references on the following page for additional information.

**ADDITIONAL READING**

[1] B. Keeth, R. J. Baker, B. Johnson, and F. Lin, *DRAM Circuit Design: Fundamental and High-Speed Topics, Second Edition*, Wiley-IEEE, 2008. ISBN 978-0-470-18475-2

[2] B. Jacob, S. W. Ng, and D. T. Wang, *Memory Systems: Cache, DRAM, Disk*, Morgan Kaufmann, 2008. ISBN 978-0123797513

[3] J. E. Brewer and M. Gill, *Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices*, Wiley-IEEE, 2008. ISBN 978-0471770022

[4] K. Itoh, *VLSI Memory Chip Design*, Springer-Verlag Publishers, 2001. ISBN 3-5406-7820-4

[5] W. D. Brown and J. E. Brewer (eds.), *Nonvolatile Semiconductor Memory Technology: A Comprehensive Guide to Understanding and Using NVSM Devices*, John Wiley and Sons Publishers, 1998. ISBN 0-7803-1173-6

[6] A. K. Sharma, *Semiconductor Memories: Technology, Testing and Reliability*, John Wiley and Sons Publishers, 1997. ISBN 0-7803-1000-4

[7] T. Mohihara, et al., "Disk-Shaped Capacitor Cell for 256Mb Dynamic Random-Acess Memory," *Japan Journal of Applied Physics*, vol. 33, part 1, no. 8, pp. 4570–4575, August 1994.

[8] K. Sagara, et al., "Recessed Memory Array Technology for a Double Cylindrical Stacked Capacitor Cell of 256M DRAM," *IEICE Trans. Electron.*, vol. E75-C, No. 11, pp. 1313–1322, November 1992.

[9] T. Hamada, "A Split-Level Diagonal Bit-Line (SLDB) Stacked Capacitor Cell for 256Mb DRAMs," 1992 *IEDM Technical Digest*, pp. 799–802.

[10] J. H. Ahn et al., "Micro Villus Patterning (MVP) Technology for 256Mb DRAM Stack Cell," 1992 *Symposium on VLSI Technical Digest of Technical Papers*, pp. 12–13.

[11] M. I. Elmasry, *Digital MOS Integrated Circuits II*, IEEE Press, 1992. ISBN 0-87942-275-0

[12] B. Prince, *Semiconductor Memories: A Handbook of Design Manufacture, and Application*, 2nd ed., John Wiley and Sons Publishers, 1991. ISBN 0-471-92465-2

[13] R. D. Pashley and S. K. Lai, "Flash Memories: The Best of Two Worlds," *IEEE Spectrum*, 1989, pp. 30–33.

IPR2023-00697
Theta EX2019

[14] D. Frohman-Bentchkowsky, "FAMOS-A New Semiconductor Charge Storage Device," *Solid-State Electronics*, vol. 17, pp. 517–529, 1974.

[15] E. H. Snow, "Fowler-Nordheim Tunneling in $SiO_2$ Films," *Solid-State Communications*, vol. 5, pp. 813–815, 1967.

*Additional Readings Covering Analog Applications of Floating Gate Devices*

[16] C. T. Charles and R. R. Harrison, "A Floating Gate Common Mode Feedback Circuit for Low Noise Amplifiers," *Proceedings of the Southwest Symposium on Mixed-Signal Design*, Las Vegas, NV, pp. 180–185, February 23–25, 2003.

[17] F. Adil, G. Serrano, and P. Hasler, "Offset Removal Using Floating-Gate Circuits for Mixed-Signal Systems," *Proceedings of the Southwest Symposium on Mixed-Signal Design*, Las Vegas, NV, pp. 190–195, February 23-25, 2003.

[18] R. R. Harrison, J. A. Bragg, P. Hasler, B. A. Minch, and S. P. Deweerth, "A CMOS programmable analog memory-cell array using floating-gate circuits," *IEEE Transactions on Circuits and Systems-II*, vol. 48, no. 1, pp. 4–11, 2001.

[19] F. Munoz, A. Torralba, R. G. Carvajal, J. Tombs, and J. Ramírez Angulo, "Floating-Gate based tunable CMOS low-voltage linear transconductor and its application to HF $g_m$-$C$ filter design," *IEEE Transactions on Circuits and Systems*, vol. 48, no. 1, January 2001, pp. 106–110.

[20] E. Sánchez-Sinencio and A. G. Andreou, "Low-Voltage/Low-Power Integrated Circuits and Systems: Low-Voltage Mixed-Signal Circuits," *IEEE Press*, 1999. ISBN 0-7803-3446-9

[21] P. M. Furth and H. A. Ommani, "A 500-nW floating-gate amplifier with programmable gain," 41st Midwest Symp. Circuits and Systems, South Bend, IN, August 1998.

[22] J. Ramírez-Angulo, "Ultracompact Low-voltage Analog CMOS Multiplier Using Multiple Input Floating Gate Transistors," 1996 European Solid State Circuits Conference, pp. 99–103.

[23] J. Ramírez-Angulo, S.C. Choi, and G. Gonzalez-Altamirano, "Low-Voltage OTA Architectures Using Multiple Input Floating gate Transistors," *IEEE Transactions on Circuits and Systems*, vol. 42, no. 12, pp. 971–974, November 1995.

[24] C. G. Yu and R. L. Geiger, "Very Low Voltage Operational Amplifiers Using Floating Gate MOS Transistors," *IEEE Symp. Circuits Sys.*, vol. 2, pp. 1152–1155, 1993.

[25] L. R. Carley, "Trimming Analog Circuits Using Floating-Gate Analog MOS Memory," *IEEE Journal of Solid-State Circuits*, vol. SC-24, pp. 1569–1575, 1989.

[26] J. Sweeney and R. L. Geiger, "Very High Precision Analog Trimming Using Floating Gate MOSFETs," *Proceedings of the European Converence on Circuit Theory and Design*, Brighton, United Kingdom, September 1989, pp. 652–655.

IPR2023-00697
Theta EX2019

## PROBLEMS

Unless otherwise indicated use the short-channel CMOS BSIM4 models for all simulations.

**16.1** Estimate the bit line capacitance if there are 256 word lines and we include the gate-drain overlap capacitance from each MOSFET, as seen in Fig. 16.68.

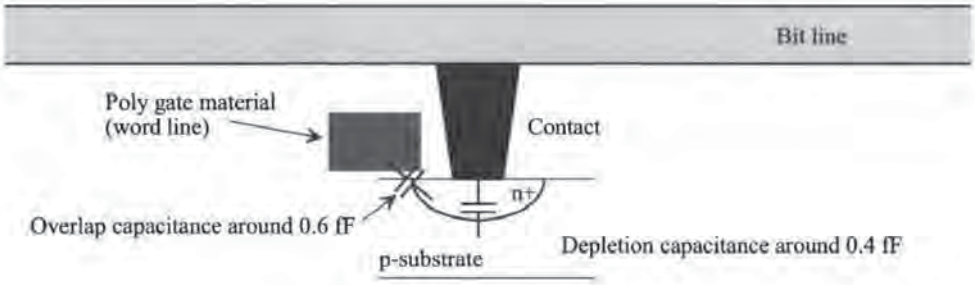



**Figure 16.68** How the gate-drain overlap capacitance loads the bit line. Note that this cross-sectional view is rotated 90 degrees from the one seen in Fig. 16.3.

**16.2** Consider the NSA seen in Fig. 16.69. Suppose that the load capacitance is mismatched by 20%, as seen in the figure. Assuming that both caps are equilibrated to 0.5 V prior to sensing, which capacitor will fully discharge to ground (which MOSFET will fully turn on)? What voltage difference on the capacitors is needed to cause metastability? Verify your answers with SPICE.

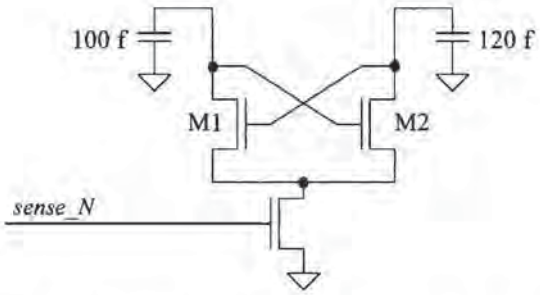



**Figure 16.69** Problem 16.2 showing how a mismatch in the load capacitance of an NSA can result in in sensing errors.

**16.3** Repeat Problem 16.2 with the circuit in Fig. 16.70. In this figure M1 and M2 experience a threshold voltage mismatch (modeled by the DC voltage source seen in the figure).

**16.4** Examining Fig. 16.17 we see that there will be a voltage drop along the metal line labeled *NLAT* when the sense amplifiers fire. Re-sketch this metal line as resistors between each NSA. If the voltage drop along the line is significant (the length of

IPR2023-00697
Theta EX2019

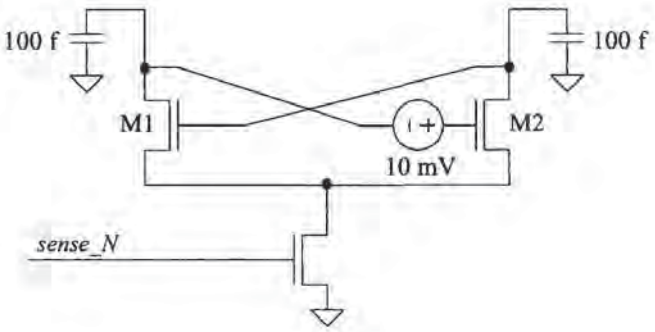



**Figure 16.70** Circuit for Problem 16.3.

the line is very long), errors can result. Would we make things better by using individual MOSFETs on the bottom of each NSA connected to *sense_N*? Why or why not?

**16.5** Suppose a memory array has 1024 columns. If the word line resistance of one cell is 2 Ω and the capacitance per cell (to ground) is 500 aF, estimate the delay to open a row line. Sketch the equivalent RC circuit of the word line.

**16.6** In Fig. 16.71, if the top plate of capacitor $C_1$ is initially charged to $V_1$ and the top plate of capacitor $C_2$ is initially charged to $V_2$, estimate the final voltage, $V_{final}$, on the top plates of the capacitors after the switch closes.

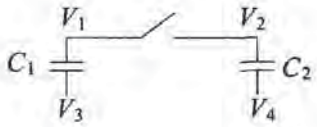



**Figure 16.71** Circuit for Problem 16.6.

**16.7** Figure 16.72 shows a clocked comparator topology based on the topology seen in Fig. 16.32. The location of the imbalance MOSFETs has been moved in this figure. Comment on the merits and disadvantages of this topology compared to the one in Fig. 16.32. Simulate the operation of this circuit.

**16.8** Figure 16.73 shows the addition of I/O (input/output) transistors to the memory array and NSA seen in Fig. 16.17. Sketch a column decoder design using static logic. Show how the 3-bit input address is connected to each stage.

**16.9** The I/O lines in the previous problem won't swing to full logic levels. To restore a full *VDD* level on these lines and to speed up the signals, a *helper flip-flop* is used. Sketch a possible implementation of the helper flip-flop. Why can it be used to speed up the signals?

IPR2023-00697
Theta EX2019

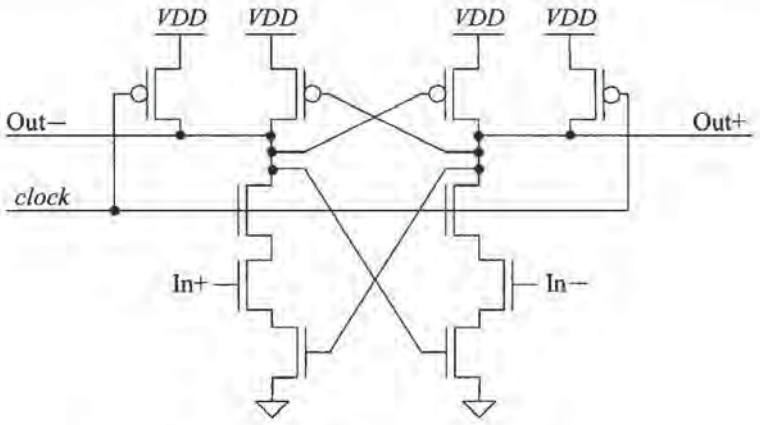



**Figure 16.72** An alternative clocked sense amplifier.

**16.10** Suppose a 2Mbit memory is to be designed as a x2 part (2-bit input/output words). Further suppose that 10 address pins are available to access the memory. Sketch a block diagram of how the row and column addresses are multiplexed together and stored in separate registers. Use $\overline{RAS}$ and $\overline{CAS}$, as discussed in the chapter to clock in the addresses. Comment on the validity of using 20-bits to access 2Mbits of data.

**16.11** Suppose that the memory in Problem 16.10 is made up of 8-256k memory arrays (assume 1024 row lines, a folded array, and 512 column lines). How many I/O lines are needed for each array? If three bits of the address are used to select one of the memory arrays (so that only a single row is open in a single memory array

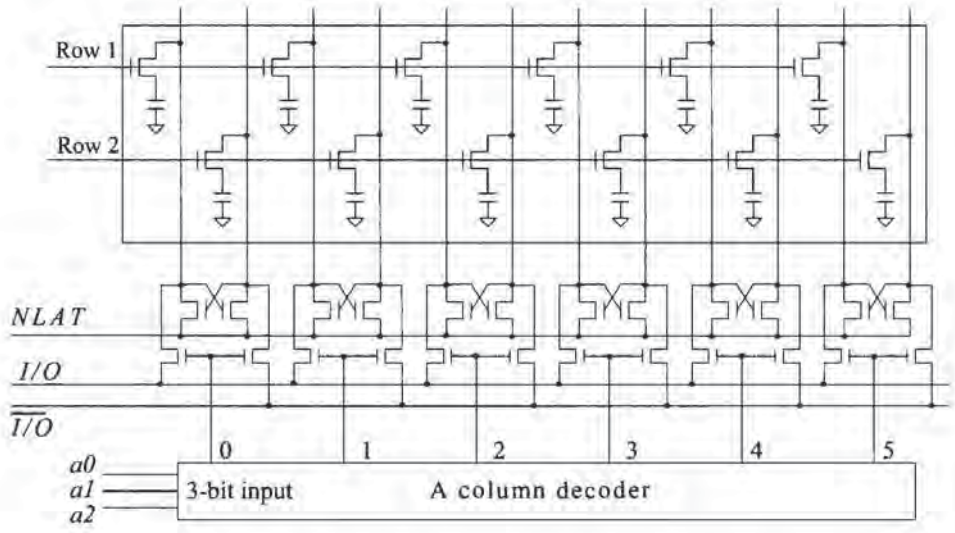



**Figure 16.73** Design of a column decoder for problem 16.8.

IPR2023-00697
Theta EX2019

at a time), how big is a page of data? Sketch a block diagram of a possible decoding scheme for the chip. Assume that only the three bits of the address are globally decoded and that the remaining 17 bits are locally decoded. How many of these bits must be routed to each array assuming an enable (one for each of the eight memory arrays) signal from the global decoder is routed to each memory array.

**16.12** Suppose that it is suggested that the word line driver in Fig. 16.46 be modified to simplify it as seen in Fig. 16.74. What is the problem with this design? Use SPICE to illustrate the driver not functioning properly.

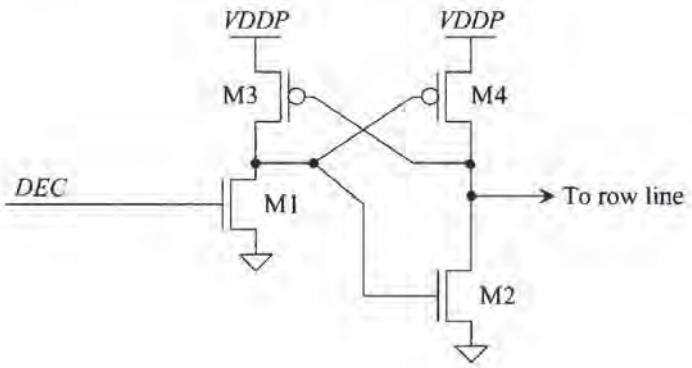



**Figure 16.74** A (bad) CMOS word line driver, Problem 16.12.

**16.13** Simulate the operation of the SRAM cell seen in Fig. 16.48. Use the 50 nm process with NMOS of 10/1 and PMOS of 20/1. Is it wise for the access MOSFETs to be the same size as the latch MOSFETs? Why or why not? Use simulations to verify your answers.

**16.14** Suppose an array of EPROM cells, see Fig. 16.57, consisting of two row lines and four bit lines is designed. Further assume $V_{THN,Erased} = 1\ V$ and $V_{THN,Prog} = 4\ V$. Will there be any problems reading the memory out of the array if an unused row line is grounded while the accessed row line is driven to 5 V ? Explain why or why not.

**16.15** Suppose that it is suggested that the n-well in a NAND Flash memory cell should be grounded at all times except when the array is being erased. Is this OK? What would be a potential benefit?

**16.16** Explain in your own words and with the help of pictures why a top select MOSFET is needed in a NAND Flash memory cell.

**16.17** Suppose that the transistor connected to RA3 in Fig. 16.61 is to be programmed. Further suppose that the n+ source implant seen in the layout is to remain grounded as indicated in Table 16.1. How do we float the source of the transistor connected to RA3 so that it can be programmed as seen in Fig. 16.58?

IPR2023-00697
Theta EX2019

**16.18** Reviewing Fig. 16.62, is it necessary that the gates of the floating gate MOSFETs connected to RA1 – RA3 be driven to 5 V? Can the gates of these MOSFETs remain grounded? Why or why not? If the RA1 MOSFET is to be programmed, in this figure, must the gate of RA0 be driven to 5 V?

**16.19** If the $I_{erased}$ of a NAND Flash memory cell is 20 μA and the $I_{prog}$ is 2 μA, explain what the bit line voltage will do when reading out the cell in the following configuration, Fig. 16.75. Will the bit line go all the way to *VDD*? All the way to ground? Explain. If the bit line capacitance is 200 fF, estimate the length of time it will take the data on the bit line to settle before it can be read out. Assume that the *VDD* is 5 V and the bit line is equilibrated to 2.5 V prior to sensing.

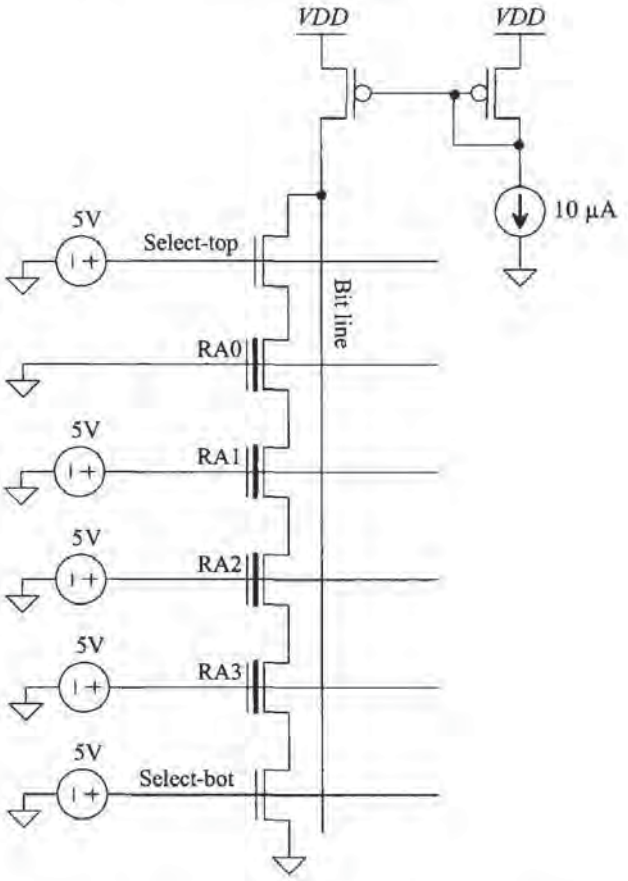



**Figure 16.75** Reading the contents of RA0 in a NAND Flash cell.

**16.20** Show, using a configuration similar to the one seen in Fig. 16.66, if it is possible to get negative gate tunnel gate current (used for erasing) by grounding the gate and raising the potential on the drain of the MOSFET. Explain why it works or why it doesn't work.

IPR2023-00697
Theta EX2019

Chapter

17

# Sensing Using ΔΣ Modulation

In the last chapter we performed sensing by clocking a sense amplifier that compared two inputs and determined which one had the greater value. To illustrate some concerns with this approach, as signals get smaller or noisier, consider the water analogy seen in Fig. 17.1. In this figure we show two buckets filled with water. Our sensing circuit should determine if the water is above or below the line indicated on the bucket. In (a) our sense-amp can clearly determine that the water is below the line. In (b), however, the water is sloshing around, so it's difficult to determine if the level is above or below the line. To make a more accurate determination in (b), we might try averaging, at different times, several sense amplifier outputs. For example, we might get, after strobing the sense-amp four times, outputs of: yes (it's above the line), no (it's not), yes, yes. Averaging these responses results in the answer "yes, it is above the line." The averaging can be thought of as reducing the noise in the signal (the variations in the water level because of sloshing).

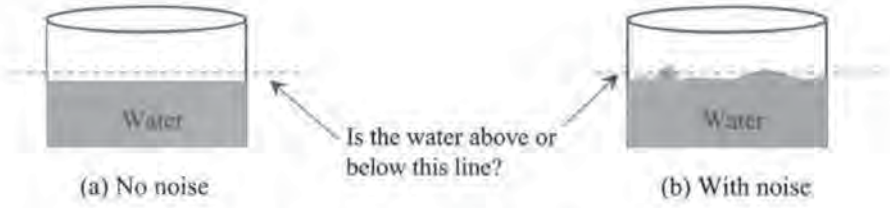



**Figure 17.1** Using a water analogy to illustrate the problems with sensing.

Let's take this a step further. The output in the previous analogy resulted in answering the question, "Is the water above or below the line?" In many applications (multi-level memory cells, imaging sensors, analog-to-digital conversion, etc.) this isn't enough information. We need to determine the actual height of the water in the bucket. This means we need more than just a "1" (yes) or "0" (no) output. The purpose of this chapter is to present a powerful and practical circuit technique called *delta-sigma modulation* (also known as *sigma-delta modulation*) for sensing applications (where the desired signal we are sensing is a constant but may be corrupted with noise).

IPR2023-00697
Theta EX2019

## 17.1 Qualitative Discussion

Figure 17.2 shows how delta-sigma modulation (DSM) works. The height of the water in the bucket that is "sensed" is changed into a rate of water flow. If the height of the water level increases, the float moves upwards causing the valve to open up and more water to fall into the "sigma" bucket. If the water level gets too high (above an arbitrary line in the sigma bucket), we remove a cup of water. The delta is the difference in how many times we remove a cup of water per time with the rate the water flows into the sigma bucket. By averaging the number of times we remove a cup of water over time, we can determine the rate the water flows out of the valve and thus the relative height of the water in the bucket we are sensing. The best way to understand the fundamentals of DSM is with examples.

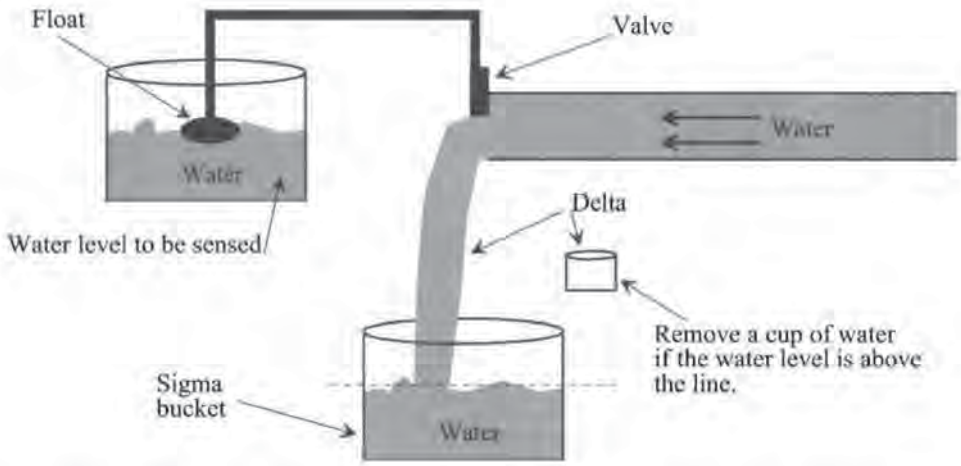



**Figure 17.2** Changing the water height (pressure or voltage) into a water flow (current) using the float and valve. The delta comes from the difference in the number of cups we remove from the bucket with the water we add to the bucket. The sigma (sum) is the storage of the difference in the bucket.

### 17.1.1 Examples of DSM

Let's say the rate the water is flowing into the sigma bucket is one cup every 40 seconds (0.25 cups per 10 seconds). Further, let's say we check the height of the water in this bucket every 10 seconds. If we start the sense off with the water height in the sigma bucket at (arbitrarily) 5 cups (our reference line), we can generate the data in Table 17.1. At a time of 10 seconds 0.25 cups have fallen into the sigma bucket and so the water level is 5.25 cups. Since this water level is > 5 cups, we remove a cup of water from the bucket (at 10 seconds) leaving 4.25 cups in the bucket. At 20 seconds, after another 0.25 cups of water have fallen into the bucket, we have 4.5 cups. Notice that the longer we average, the closer our output moves to 0.25 cups/10 seconds.

Another key point to notice is that if we make a mistake when determining if the water level in the sigma bucket is > 5 cups, it doesn't really matter. The error averages out over time. What does matter though is how carefully we fill up the cup when

IPR2023-00697
Theta EX2019

removing water. If we don't fill it up all the way or if the water spills out of the cup, the level of the water in the sigma bucket changes. The result limits the precision of the sense. One other limiting factor is the sigma bucket. If it is "leaky" and the water it holds leaks out, the quality of the sense will be affected.

**Table 17.1** An example of DSM.

| Time, seconds | Water level in the sigma bucket (cups) | Remove cup? (water level > 5?) | Running average |
|---|---|---|---|
| 0 | 5 | | 0 |
| 10 | 5.25 | Yes | 1 |
| 20 | 4.5 | No | 0.5 |
| 30 | 4.75 | No | 0.33 |
| 40 | 5 | No | 0.25 |
| 50 | 5.25 | Yes | 0.4 |
| 60 | 4.5 | No | 0.33 |
| 70 | 4.75 | No | 0.29 |
| 80 | 5 | No | 0.25 |
| 90 | 5.25 | Yes | 0.33 |
| 100 | 4.5 | No | 0.3 |
| 110 | 4.75 | No | 0.27 |
| 120 | 5 | No | 0.25 |
| 130 | 5.25 | Yes | 0.31 |
| 140 | 4.5 | No | 0.29 |
| 150 | 4.75 | No | 0.26 |
| 160 | 5 | No | 0.25 |
| 170 | 5.25 | Yes | 0.29 |

*The Counter*

We might wonder how, in a practical sensing circuit, we can get decimal numbers like the ones seen in Table 17.1? The answer to this question is seen in Fig. 17.3. If a "yes, remove a cup of water" is indicated by the output of the DSM sensing circuit going high, a counter can be used to generate the number. If $N$ is the total number of times we clock the DSM, then

$$\text{Output number} = \frac{\text{number of Yes outputs}}{N} \qquad (17.1)$$

In Table 17.1 $N$ is 17. The number of "yes" outputs is 5. The final output number is then 5/17 or 0.29. If a 10-bit counter is used (assuming it was reset to zeroes at the beginning of the sense), then its digital output is 00 0000 0101. To find the absolute value of the input, we multiply this number by the size of the feedback signal (here one cup) to get

IPR2023-00697
Theta EX2019

$$\text{Input signal} = \text{Ouput number} \times \text{fed back signal size} \qquad (17.2)$$

or here 0.29 cups (per 10 seconds). Note that to converge on the actual rate of 0.25 cups/10 seconds we would need to average more points.

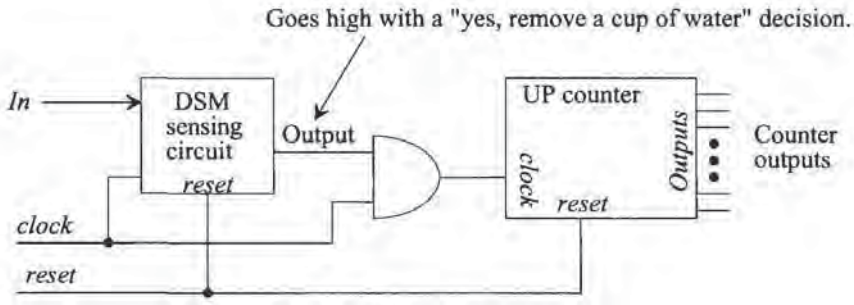



**Figure 17.3** How a counter is used to average the outputs of a DSM sensing circuit.

*Cup Size*

The size of our fed back signal, here the cup size, is very important. If we use a small cup, we can converge on the correct digital representation (the counter output) of our analog signal (the flow of water into the sigma bucket) quicker. However, if we use too small of a cup, then we can't remove the water fast enough from the bucket and it will overfill. For a large range, we need a big cup. Using a big cup means that we have to average longer (the sensing lasts for a longer period of time).

*Another Example*

Figure 17.4 shows another water analogy where DSM can be used for measuring an analog quantity (in this case the water flowing out of the sigma bucket). When the water level gets too low, we add a cup of water to the bucket. Again, averaging how often we add the cup of water to the bucket gives a digital representation of the rate at which water leaves the bucket.

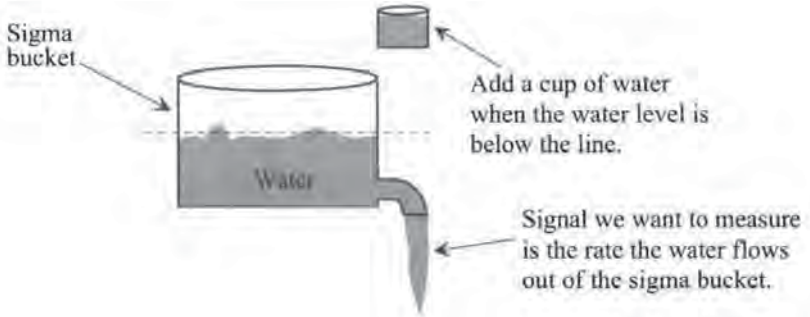



**Figure 17.4** Using DSM to measure the water leaving a bucket.

IPR2023-00697
Theta EX2019

### 17.1.2 Using DSM for Sensing in Flash Memory

Let's discuss sensing using DSM in a floating gate memory cell technology (Flash). Figure 17.5a shows the characteristics of the Flash memory cell discussed in the last chapter (see Fig. 16.64). When we are sensing the state of the Flash cell, that is, erased or programmed, we hold the row line at ground so that the cell's $V_{GS}$ is 0. If the bit line is held at a potential above ground, say 1 V, then a current of either $I_{erased}$ or $I_{prog}$ (ideally) flows in the cell. In reality variations in the production of the cell and the consistency of the erase from cell to cell affect these values. Using DSM we can more precisely determine the drain current in the cell. This allows us to make a more intelligent decision about the state of the cell. Further, DSM can also be used to program the cell to precisely set the programmed current flow. This allows us to make a memory cell out of a single transistor, which can be used to store several logic levels (values of programmed current). Because of the ability to precisely control the programming operation in a floating gate technology, programming using Fowler-Nordheim Tunneling (FNT) can be abandoned (gate oxides of around 80 Å) for direct tunneling (gate oxides < 20 Å). Of course, the issues with data retention (the charge leaking off the floating gate) are still a concern. In this section we qualitatively discuss how DSM can be used for sensing in Flash technology.

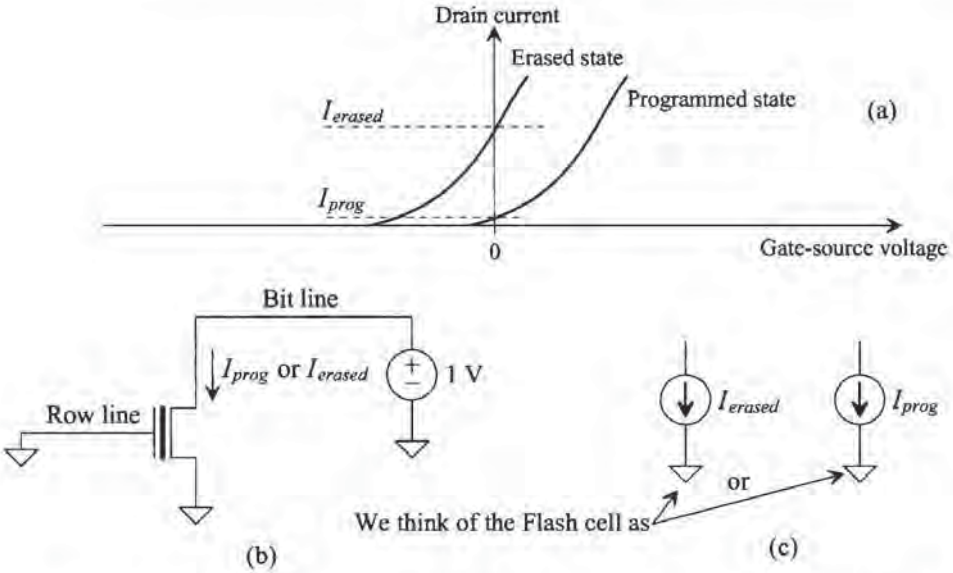



**Figure 17.5** The IV characteristics of a NAND memory cell and how we think of the cell when sensing.

*The Basic Idea*

Figure 17.6 shows the basic idea. The sigma bucket in this figure is the bit line capacitance, $C_{bit}$. As seen in Fig. 17.4, the signal we are measuring is the rate current flows out of this bucket, that is $I_{erased}$ or $I_{prog}$. The comparator is used to determine if the voltage on the bit line is below 1 V (again an arbitrary reference). The memory cell is continuously removing charge from the bit line. When the bit line voltage is below 1 V,

IPR2023-00697
Theta EX2019

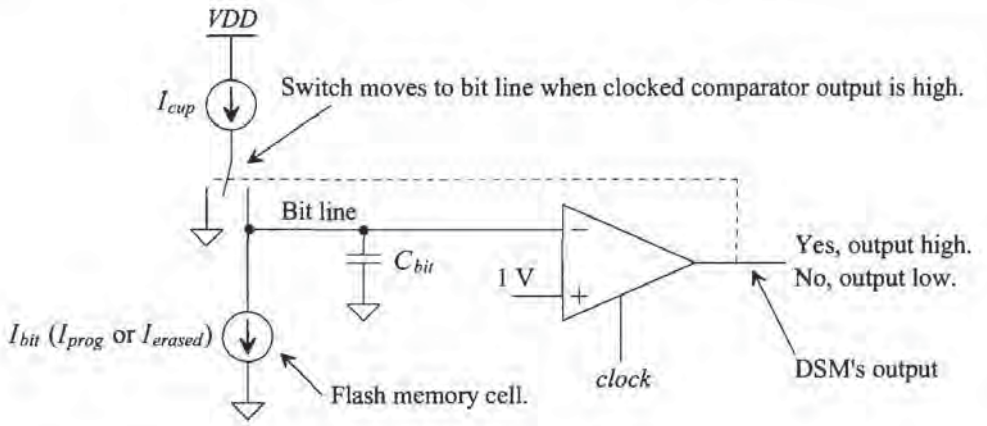



**Figure 17.6** Sensing a Flash memory cell using DSM.

the current source, $I_{cup}$, is connected to $C_{bit}$ to provide the water to the bucket. Note that *we cannot connect the bit line to a voltage source when the output of the comparator goes high.* A voltage source, in our water analogy, is a pressure with an infinite supply of water. The connection of the voltage source would fill the bucket up (the bit line capacitance) to the pressure of the source (the voltage of the source). Again, by looking at the number of times the output of the DSM sensing circuit goes high, we can determine, very precisely, the value of $I_{bit}$.

If we are clocking the comparator at a rate of $f$ at a period of $T$ (= $1/f$), then the rate that charge is removed from the bit line capacitance (the sigma bucket) is

$$\frac{I_{bit}}{C_{bit}} = \frac{\Delta V_{bit}}{T} \tag{17.3}$$

where $\Delta V_{bit}$ is the change in the bit line voltage. The amount of charge removed from the bit line in one clock cycle $T$ is

$$Q_{bit} = I_{bit}T = C_{bit} \cdot \Delta V_{bit} \tag{17.4}$$

These quantities are constant. However, the rate at which we add charge to the bit line isn't a constant (unless the output of the comparator is always the same). If $N$ is the total number of clock cycles (the total number of times we clock the comparator) and $M$ is the number of times the output of the comparator goes high (the number of "Yes, add charge to the capacitor" signals), then the rate we add charge to the bit line from $I_{cup}$ is

$$Q_{cup} = I_{cup} \cdot \frac{M}{N} \cdot T \tag{17.5}$$

In order for the voltage on the bit line (the water level in the bucket) to remain, on average, constant, we require that the amount of charge leaving the bucket ($Q_{bit}$) equal the amount of charge entering the bucket ($Q_{cup}$) or

$$Q_{bit} = I_{bit}T = Q_{cup} = I_{cup} \cdot \frac{M}{N} \cdot T \tag{17.6}$$

or

IPR2023-00697
Theta EX2019

$$\frac{I_{bit}}{I_{cup}} = \frac{M}{N} \tag{17.7}$$

This result is important. It relates the counter output code, $M$, and the total number of times the DSM sensing circuit is clocked, $N$, to the ratio of the Flash memory cells current, $I_{bit}$ and the fed-back signal $I_{cup}$. Note again that the charge leaving the bit line capacitance can't be greater than the charge entering the bit line capacitance, that is, we require $I_{cup} \geq I_{bit}$. As mentioned earlier, using a large cup (large $I_{cup}$) increases the sensing time for a required resolution (we're averaging a larger variable).

Notice that Eq. (17.7) doesn't include the clock frequency or period. We might think that these quantities aren't important. While, if the DSM is designed correctly, it doesn't directly affect the output of the sensing operation, we can have the situation where the bucket (bit line capacitance) empties or overflows (goes to ground or *VDD*). To avoid this situation, we may require that the maximum deviation on the bit line, $\Delta V_{bit,\max}$ be less than some value over a time $T$. We can write, assuming $I_{cup} \geq I_{bit}$,

$$\Delta V_{bit,\max} = \frac{I_{cup}T}{C_{bit}} = \frac{I_{cup}}{C_{bit} \cdot f_{clk}} = I_{cup} \cdot R_{sc} \tag{17.8}$$

For example, if the bit line capacitance is 500 *fF*, the clock frequency is 100 MHz ($T$ = 10 ns), and $I_{cup}$ is 10 μA, then $\Delta V_{bit,\max} = 0.2\ V$. If we clock the DSM slower, we must increase our bucket size (add capacitance in parallel with $C_{bit}$ on the input of our DSM).

**Example 17.1**

Suppose that the DSM sensing circuit in Fig. 17.6 is used to determine the current flowing in a programmed Flash memory cell. If the single transistor Flash memory cell is programmed to conduct 1, 3, 5, or 7 μA of current, estimate the counter output codes if the DSM is clocked 15 times (a 4-bit counter is used) and $I_{cup}$ is 10 μA. Estimate the maximum bit line voltage change if the DSM is clocked at 100 MHz and the bit line capacitance is 500 *fF*.

For the 1 μA program current, using Eq. (17.7), we get

$$\frac{1}{10} = \frac{M}{15} \rightarrow M = 1.5 \text{ so the counter output would be 2 (0010)}$$

For the 3 μA program current through the Flash memory cell,

$$\frac{3}{10} = \frac{M}{15} \rightarrow M = 4.5 \text{ so the counter output would be 5 (0101)}$$

For the 5 μA program current,

$$\frac{5}{10} = \frac{M}{15} \rightarrow M = 7.5 \text{ so the counter output would be 8 (1000)}$$

Finally, for the 7 μA program current,

$$\frac{7}{10} = \frac{M}{15} \rightarrow M = 10.5 \text{ so the counter output would be 11 (1011)}$$

The maximum deviation of the voltage on the bit line would be when the program current is 1 μA (leaving the bit line) and $I_{cup}$ is connected (10 μA charging the bit line). This would give a net current of 9 μA into the bit line so

IPR2023-00697
Theta EX2019

$$\Delta V_{bit,\max} = \frac{9\mu A \cdot 10\ ns}{500\ fF} = 180\ mV$$

Let's show the output decisions for the case when $I_{prog}$ is 1 μA, Table 17.2. Note that if 1 μA is removed from the 500 *fF* bit line capacitor for 10 ns, the bit line voltage drops 20 mV. If (net) 9 μA of current is put into the bit line for 10 ns, the bit line voltage increases by 180 mV. Note that as calculated, the maximum change in the bit line voltage is 180 mV. Also note that if the comparator had output a Yes at 100 ns instead of at 110 ns, it wouldn't have affected the final output. ■

**Table 17.2** See Ex. 17.1.

| Time, nanoseconds | Bit line voltage | Add current? ($V_{bit}$ < 1 V?) | Bit line current | Counter output |
|---|---|---|---|---|
| 0 | 1 | | −1 μA | 0 (0000) |
| 10 | 0.98 | Yes (1) | 9 μA | 1 (0001) |
| 20 | 1.16 | No (0) | −1 μA | 1 (0001) |
| 30 | 1.14 | No (0) | −1 μA | 1 (0001) |
| 40 | 1.12 | No (0) | −1 μA | 1 (0001) |
| 50 | 1.1 | No (0) | −1 μA | 1 (0001) |
| 60 | 1.08 | No (0) | −1 μA | 1 (0001) |
| 70 | 1.06 | No (0) | −1 μA | 1 (0001) |
| 80 | 1.04 | No (0) | −1 μA | 1 (0001) |
| 90 | 1.02 | No (0) | −1 μA | 1 (0001) |
| 100 | 1 | No (0) | −1 μA | 1 (0001) |
| 110 | 0.98 | Yes (1) | 9 μA | 2 (0010) |
| 120 | 1.16 | No (0) | −1 μA | 2 (0010) |
| 130 | 1.14 | No (0) | −1 μA | 2 (0010) |
| 140 | 1.12 | No (0) | −1 μA | 2 (0010) |
| 150 | 1.1 | No (0) | −1 μA | 2 (0010) |

Notice, in the previous example, that we can clock the DSM indefinitely. The only limitation is the size of the counter (after a time the counter will roll over). This is important because, to get better resolution, all we have to do is sense for a longer period of time. We can define the *dynamic range* of the sense operation, in dB, as the maximum counter output code ($N$) to the smallest output code (1)

$$\text{dynamic range (DR)} = 20 \cdot \log \frac{N}{1} = 20 \cdot \log N \qquad (17.9)$$

noting that no signal would give a counter output code of zero. Clocking the DSM 1,000 times results in, ideally, a DR of 60 dB. Clocking the DSM 15 times results in a DR of 23.5 dB.

IPR2023-00697
Theta EX2019

The maximum input signal, $I_{bit,max}$, equals the fed-back signal $I_{cup}$ ($I_{cup} \geq I_{bit}$). The minimum resolvable signal is then

$$\text{minimum resolvable signal} = \frac{\text{fed-back signal}}{N} \qquad (17.10)$$

Again, this illustrates that the larger the fed-back signal, the more averages, $N$, are needed for a given resolution.

**Example 17.2**

Determine the minimum resolvable programmed current, $I_{bit}$, in Ex. 17.1. Verify the answer with a table similar to Table 17.2.

Using Eq. (17.10), the minimum resolvable current is 10 μA/15 or 0.666 μA. If this current is removed from the bit line capacitance for 10 ns, the voltage drop is 13.33 mV. If the bit line capacitance is charged with 9.333 μA for 10 ns, its voltage increases to 186.66 mV. Table 17.3 tabulates the operation of the DSM sensing circuit when $I_{prog}$ is 666.6 nA. Note that in all situations where $I_{prog}$ is not zero the first decision is a Yes. However, after neglecting this decision, $N$ clock cycles later the counter increments to 2 when the cell is sinking the minimum resolvable signal (here 666.6 nA). ■

**Table 17.3** See Ex. 17.2.

| Time, nanoseconds | Bit line voltage (millivolts) | Add current? ($V_{bit} < 1$ V?) | Bit line current | Counter output |
|---|---|---|---|---|
| 0 | 1,000 | | –0.666 μA | 0 (0000) |
| 10 | 986.67 | Yes (1) | 9.333 μA | 1 (0001) |
| 20 | 1,173.33 | No (0) | –0.666 μA | 1 (0001) |
| 30 | 1,160 | No (0) | –0.666 μA | 1 (0001) |
| 40 | 1,146.67 | No (0) | –0.666 μA | 1 (0001) |
| 50 | 1,133.34 | No (0) | –0.666 μA | 1 (0001) |
| 60 | 1,120 | No (0) | –0.666 μA | 1 (0001) |
| 70 | 1,106.67 | No (0) | –0.666 μA | 1 (0001) |
| 80 | 1,093.33 | No (0) | –0.666 μA | 1 (0001) |
| 90 | 1,080 | No (0) | –0.666 μA | 1 (0001) |
| 100 | 1,066.67 | No (0) | –0.666 μA | 1 (0001) |
| 110 | 1,053.33 | No (0) | –0.666 μA | 1 (0001) |
| 120 | 1,040 | No (0) | –0.666 μA | 1 (0001) |
| 130 | 1,026.67 | No (0) | –0.666 μA | 1 (0001) |
| 140 | 1,013.33 | No (0) | –0.666 μA | 1 (0001) |
| 150 | 1,000 | No (0) | –0.666 μA | 1 (0001) |
| 160 | 986.67 | Yes (1) | 9.333 μA | 2 (0010) |

IPR2023-00697
Theta EX2019

*The Feedback Signal*

The precision of a sense operation in a DSM is limited by how precisely current can be guided into or out of the bit line capacitance. The bit line capacitance can be thought of as integrating (an integrator, *sigma*) the difference (*delta*) between the fed-back signal ($I_{cup}$ in Fig. 17.6) and the signal we are measuring ($I_{bit}$). If the amount of charge (current) steered into the bit line isn't consistent from one clock period to the next, errors will occur. To illustrate the possibility of errors, examine the circuit seen in Fig. 17.7. When the switch is connected to ground $C_{parasitic}$ (the parasitic capacitance on the output of the current source), is discharged. Now when the switch is connected to the bit line $I_{cup}$ must charge both $C_{parasitic}$ and $C_{bit}$. Because the bit line voltage will be moving around and the switch may stay connected to the bitline for two or more consecutive cycles, errors in the sense will occur that limit the resolution. For example, the charge supplied to $C_{parasitic}$ may be $1.05 \cdot C_{parasitic}$ (a bit line voltage of 1.05V) in one clock cycle, while in a different clock cycle it may be $1.02 \cdot C_{parasitic}$. In our water analogy in Fig. 17.4 the amount of water in the cup effectively changes due to this parasitic. If the size of the bit line capacitance is large (or if capacitance is added to the input of the DSM to purposely increase this capacitance) or only a course sense is used (say 128 or less clock cycles), then $C_{parasitic}$ will probably not affect the sense in any significant way. Note that if the switch were connected to 1V in Fig. 17.7 instead of ground, the errors from $C_{parasitic}$ would be reduced.

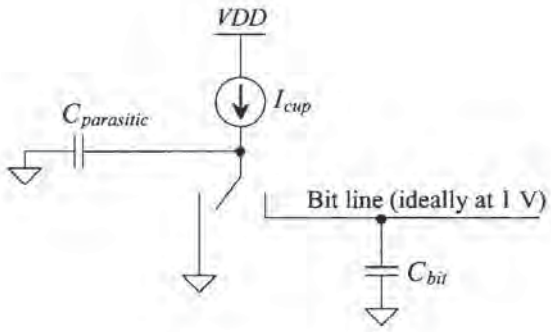



**Figure 17.7** How the parasitic capacitance on the output of the current source causes errors.

In an attempt to minimize the power (power is burned unnecessarily when $I_{cup}$ is connected to ground in Figs. 17.6 or 17.7) and the effects of parasitics consider the switched-capacitor (SC) circuit seen in Fig. 17.8. In the following we assume the bit line reference voltage (here 1 V) is greater than the threshold voltage of the PMOS transistors so that the PMOS switches used in the SC can turn fully on. The two clock signals, $\phi_1$ and $\phi_2$ form nonoverlapping clock signals (they are never low at the same time). This is important because we never want to connect the bit line directly to *VDD* (which occurs if the clock signals are all low at the same time). When $\phi_1$ is low, $C_{cup}$ charges to *VDD*. Next, $\phi_1$ goes high. Between $\phi_1$ going high and $\phi_2$ going low, the comparator is clocked (on the rising edge of *clock*). If charge needs to be added to the bit line, the comparator output goes high and the gate of M3 is driven low. Note that the inverter can be removed

IPR2023-00697
Theta EX2019

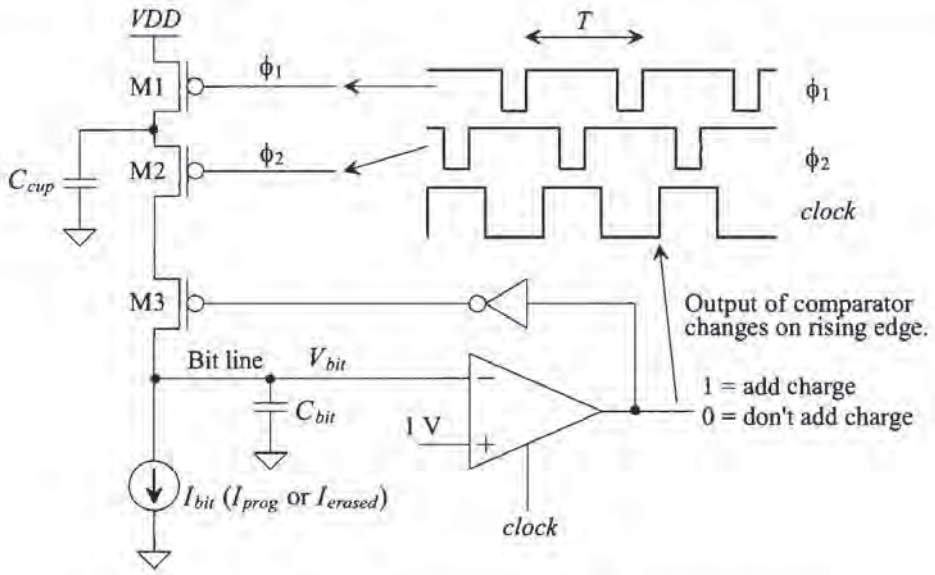



**Figure 17.8** Using a switched-capacitor to add charge to the bit line.

if we swap the inputs of the clocked comparator so that a 1 output indicates "don't add charge" and a 0 output indicates "add charge." When $\phi_2$ goes low, the charge on $C_{cup}$ is dumped into the bit line. The charge leaving the bit line is still given by Eq. (17.4). The charge dumped into the bit line from $C_{cup}$ is

$$Q_{cup} = C_{cup} \cdot (VDD - V_{bit}) \tag{17.11}$$

When we compare this amount of charge to Eq. (17.5), we see a benefit when $Q_{cup}$ is not a function of the clock period. However, the big drawback is that the charge we add to the bit line is a function of the bit line voltage. This limits the accuracy of the sense. The water analogy to this problem is that the cup is being filled up to a level dependent on the sigma bucket water level. The amount of water dumped into the bucket from the cup should be independent of the water level in the bucket.

To make the charge we add to the bit line independent of the bit line voltage, consider the addition of another PMOS device in Fig. 17.9. When both the signal from the comparator and $\phi_2$ are low, M2 and M3 are on. The charge from $C_{cup}$ is dumped into the source of M4. The result is an initial increase in M4's source potential (M4 turns on). However, after the charge is dumped into the bit line, M4 shuts off. Its $V_{SG}$ goes to $V_{THP}$ (M4's source potential goes to $V_{REF} + V_{THP}$). This keeps the potential across $C_{cup}$ constant. Equation (17.11) can be rewritten as

$$Q_{cup} = C_{cup} \cdot (VDD - V_{REF} - V_{THP}) \tag{17.12}$$

knowing, to keep M4 from being fully on, that is, $V_{SD,sat}$ not zero

$$V_{REF} + V_{THP} > V_{bit,\max} \tag{17.13}$$

where $V_{bit,\max}$ is the maximum voltage on the bit line (and, again, $V_{bit,\min} > V_{THP}$).

IPR2023-00697
Theta EX2019

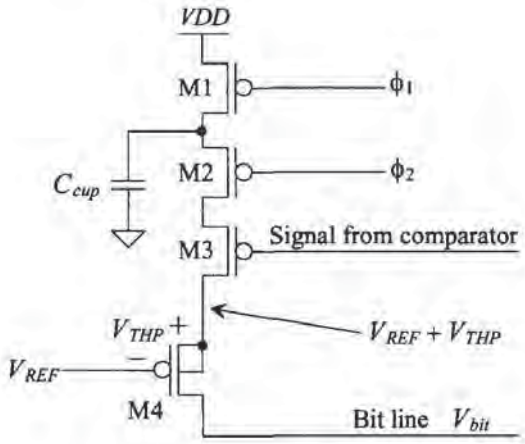



**Figure 17.9** Adding a MOSFET to set the swing across the capacitor.

**Example 17.3**
Using SPICE demonstrate the validity of Eqs. (17.11) and (17.12).

To demonstrate the validity of Eq. (17.11), examine the circuit in Fig. 17.10. We know that a PMOS device passes a voltage from *VDD* to $V_{THP}$, so we make sure that the bit line is always at a potential greater than $V_{THP}$ (= 280 mV for the short channel process used in this book). If it's not, the PMOS devices don't behave like switches. In the simulation we'll sweep the bit line voltage from 0.3 to 1V and look at the current through the bit line voltage source, $V_{bit}$. M3 is always on so we can look at the SC's operation alone (without the effects of the comparator). If Eq. (17.11) is valid, we should see a linear decrease in the current pulses (the charge) as $V_{bit}$ is increased. Figure 17.11a shows the nonoverlapping clocks used in the simulation. Note how neither clock ($\phi_1$ or $\phi_2$) is low at the same time (again, this is important). Figure 17.11b shows the current through $V_{bit}$. As expected it decreases as $V_{bit}$ increases. As Eq. (17.11) indicates. the current goes to zero when

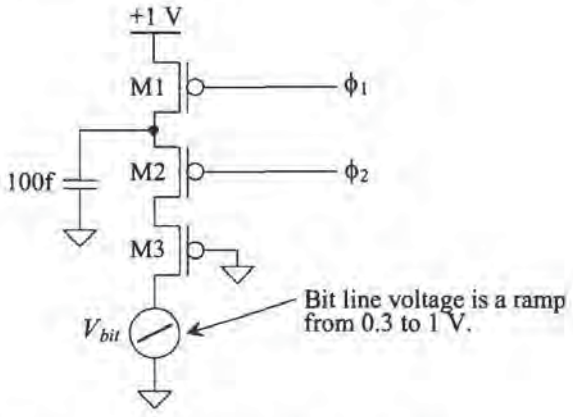



**Figure 17.10** Verifying Eq. (17.11), see Ex. 17.3.

IPR2023-00697
Theta EX2019

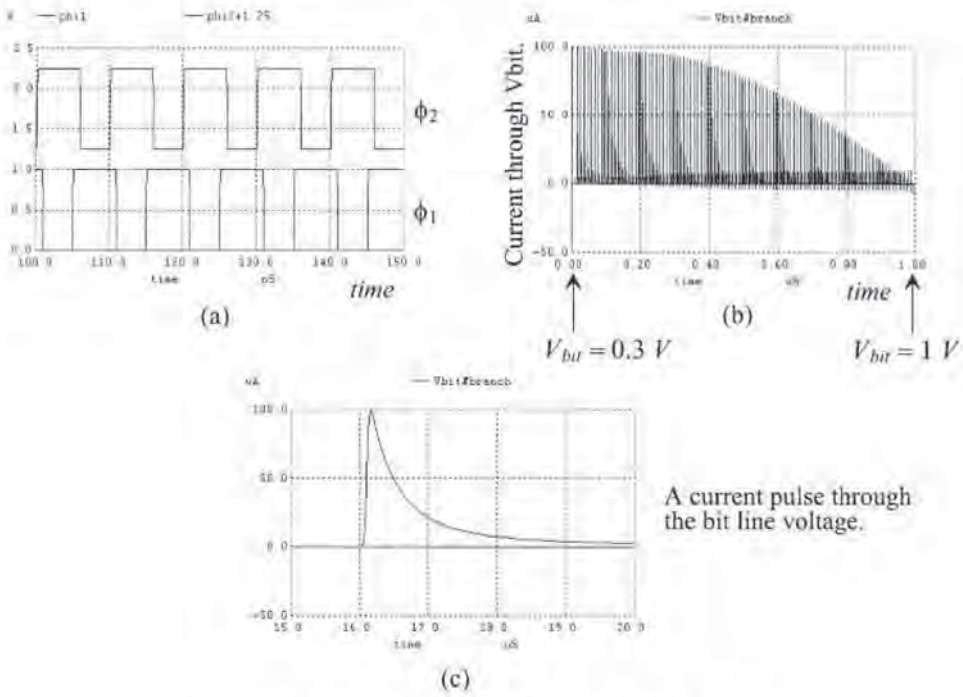



**Figure 17.11** The operation of the circuit in Fig. 17.10.

$V_{bit}$ is 1 V (= *VDD*). Let's use Eq. (17.11) to calculate the charge supplied by $C_{cup}$ when $V_{bit}$ is 300 mV

$$Q_{cup} = 100\,fF \cdot (1 - 0.3) = 70\,fC \tag{17.14}$$

Figure 17.11c shows the pulse of current that is dumped into $V_{bit}$ at the beginning of the simulation. To estimate the amount of charge in this pulse, we take the amplitude and multiply it by an estimate of the pulse's width. That is, $100\ \mu A \cdot 0.7\ ns = 70\,fC$ or the same result as seen in Eq. (17.14).

To validate Eq. (17.12), we can add M4 from Fig. 17.9 to the circuit seen in Fig. 17.10. The only parameter we need to calculate before simulating is the value of $V_{REF}$. Using Eq. (17.13) and knowing $V_{THP}$ = 280 mV, we see that if we set $V_{REF}$ to a large voltage, $Q_{cup}$ gets small. If, for example, $V_{REF} = VDD - V_{THP}$ (= 750 mV in this example), $Q_{cup}$ goes to zero. We can write

$$V_{REF} < VDD - V_{THP} \tag{17.15}$$

Let's use a $V_{REF}$ of *VDD*/2 or 500 mV. Figure 17.12 shows the simulation results. Note, in Fig. 17.9, that we place M4 in its own well. Since we care about how the threshold voltage varies, we've eliminated the body effect in this device. The simulation results are seen in Fig. 17.12. In this simulation we've swept the bit line voltage from 0.3 to 0.6 V. Comparing the plot to Fig. 17.11b, we see much better linearity (the charge added to the bit line doesn't change with bit line voltage). The cost for this improvement is more limited voltage swing on the bit line. ■

IPR2023-00697
Theta EX2019

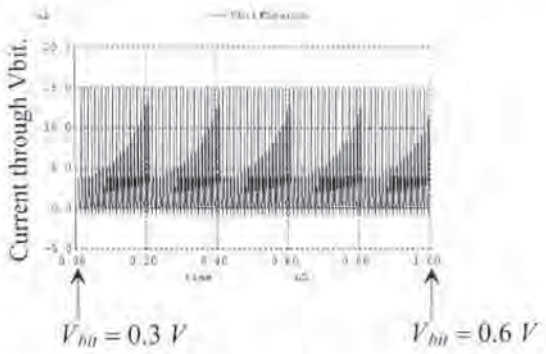



**Figure 17.12** How the charge supplied to Vbit becomes linear if we add M4, from Fig. 17.9, to the simulation in Fig. 17.10. Note the reduced bit line voltage swing and the reduction in the current pulse amplitudes.

*Incomplete Settling*

If we zoom in on the current pulses in Fig. 17.12, we get a view like the one seen in Fig. 17.13. The glitches at 30 and 40 ns are the result of the $\phi_2$ clock going high and M2 shutting off. Notice how the current through M4 (and $V_{bit}$) isn't zero when M2 shuts off. Using our water analogy, this is equivalent to stopping the pouring of the water out of the cup before the cup is empty. This incomplete emptying of the cup or capacitor is termed *incomplete settling*. The currents in the circuit haven't gone to zero before the clock transitions. The parasitic capacitances on the sources of M3/M4 continue to discharge through $V_{bit}$ after M2 shuts off, resulting in nonzero current. Incomplete settling has the effect of making the capacitor, $C_{cup}$ , appear smaller. It won't affect the linearity of the sense but it can affect the maximum value of the sensed current. To eliminate the incomplete settling behavior we can increase the width of M4. This results in the source of M4 being held closer to $V_{REF} + V_{THP}$ when M2 and M3 turns on. We could also slow the clock frequency down to ensure that the circuit settles. Because the amount of charge transferred from $C_{cup}$ is constant from one clock cycle to the next, the effects of incomplete settling on DSM are simply a limit on the maximum current that can be removed from the bit line and a modification to the digital output for the absolute value of the input signal, that is, Eq. (17.2) (a smaller value of fed-back signal size is used).

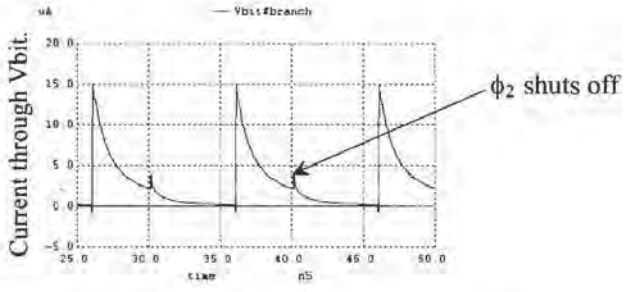



**Figure 17.13** The incomplete settling in Fig. 17.12.

IPR2023-00697
Theta EX2019

## 17.2 Sensing Resistive Memory

Let's apply the techniques we've just discussed to sensing a resistive memory cell. Figure 17.14a shows a schematic of one-transistor, one-resistor (1T1R) memory cell. Notice the similarity to the 1T1C DRAM memory cell seen in Fig. 16.9. *Ideally* the resistor is either zero ohms (which we'll call *programmed*) or infinite (which we'll call *erased*), Fig. 17.14b. To sense in this ideal case, we simply precharge the bit line to *VDD*. When the word line is driven high, the access device turns on, resulting in the bit line either moving to *VDD*/2 (if the cell is programmed) or not moving at all (if the cell is erased). In a practical circuit, however, the cell's resistance will be nonzero and not infinite (say 10kΩ to 100kΩ as seen in Fig. 17.14c).

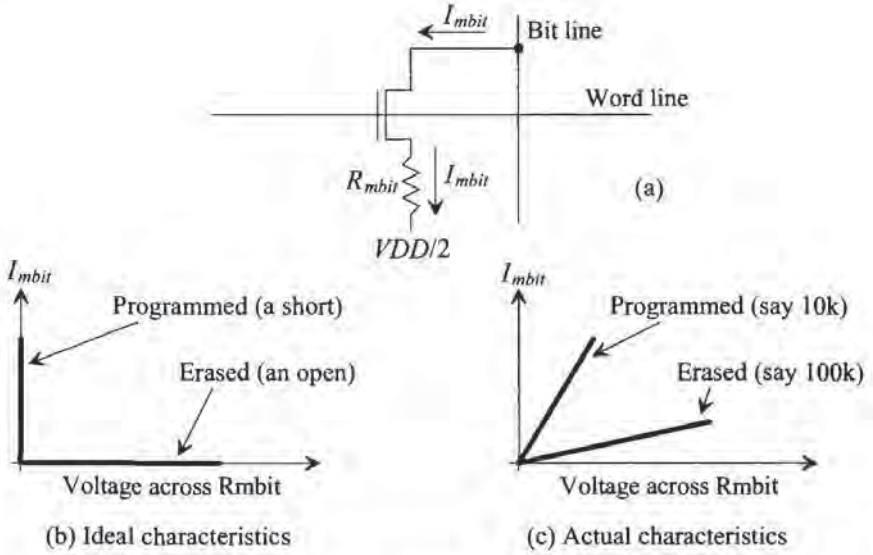



**Figure 17.14** The one-transistor, one-resistor (1T1R) RAM memory cell.

*The Bit Line Voltage*

Consider the block diagram of a DSM seen in Fig. 17.15 with a resistive memory cell. To keep the schematic simpler, we won't include the access MOSFET. We know from the previous discussions that the DSM tries to hold the bit line at precisely the reference voltage used by the comparator. In Fig. 17.15 if the reference voltage used by the comparator is *VDD*/2, then the current that flows through the memory resistor is zero. To avoid this, we might use a reference that is offset from *VDD*/2 by $V_{os}$. The current that flows in the resistor is then

$$I_{mbit} = \frac{V_{os}}{R_{mbit}} \tag{17.16}$$

In the ideal condition, the erased resistor results in zero $I_{mbit}$. The output of the DSM is then simply a string of zeroes (we never have to add current to the bit line because none is leaving). If the resistor is programmed, the bitline gets pulled down towards *VDD*/2. The current source can't supply enough charge to pull the bit line up and the DSM output is a constant string of ones (yes, add current).

IPR2023-00697
Theta EX2019

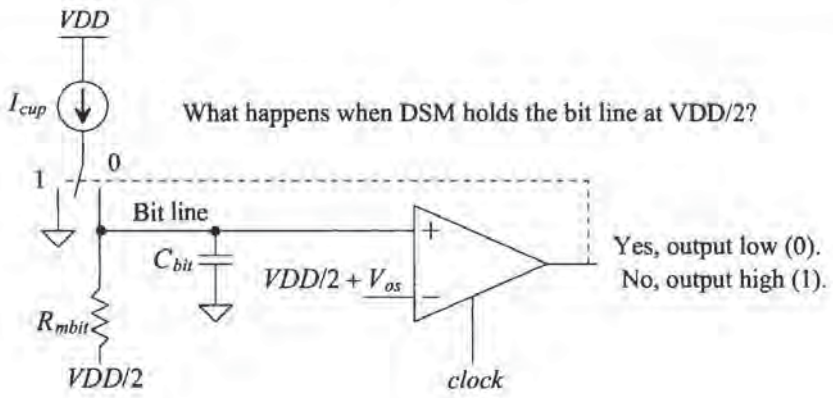



**Figure 17.15** Sensing a resitive memory cell using DSM. Notice that we've eliminated the inverter from Fig. 17.8 by switching the input terminals of the comparator.

*Adding an Offset to the Comparator*

It's important to minimize the number of reference voltages used in a memory sense scheme. We'll use *VDD* and *VDD*/2 and then design a comparator with a built-in offset voltage. The comparator design is seen in Fig. 17.16 (see Figs. 16.32 and 16.35 from the last chapter). Simulation results showing the offset are seen in Fig. 17.17. When In+ gets 50 mV above In−, the output of the comparator changes states. It's possible to design the comparator to simplify the sense circuitry, e.g., clock the comparator on the falling edge of $\phi_2$, remove the NAND gates on the output, and combine M2 and M3 in Fig. 17.9. In a

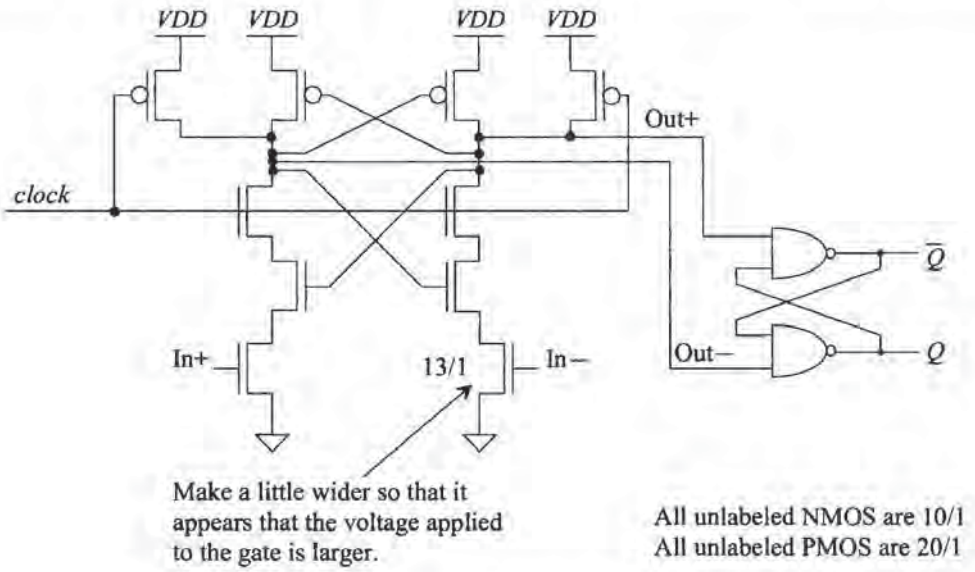



**Figure 17.16** Designing a clocked comparator with a built-in offset.

IPR2023-00697
Theta EX2019

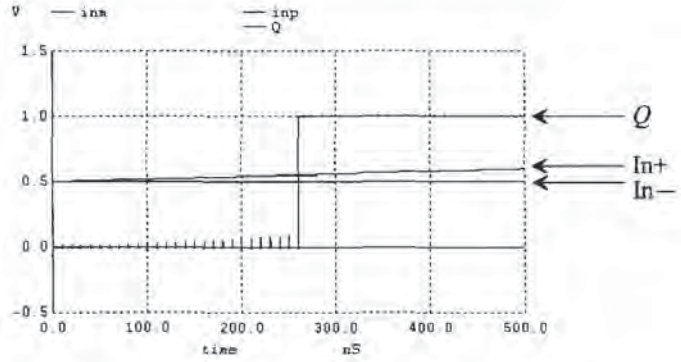



**Figure 17.17** The output of the comparator switching states when the + input is 50 mV above the - input.

production part we might want to do this. However, here, where we want to minimize the glitches that may turn some switches inadvertently partially on, we don't try to simplify the circuitry (see problems at the end of the chapter).

*Schematic and Design Values*

The schematic for a DSM sensing circuit for resistive memories is seen in Fig. 17.18. The capacitance associated with the bit line is shown explicitly (even though it's a parasitic associated with the line). When we are sensing the value of a resistive cell, a word line goes high and connects the resistance, $R_{mbit}$, to the digit line. The comparator, with its built-in offset, tries to hold the bit line, through the feedback loop, at $VDD/2 + V_{OS}$. The

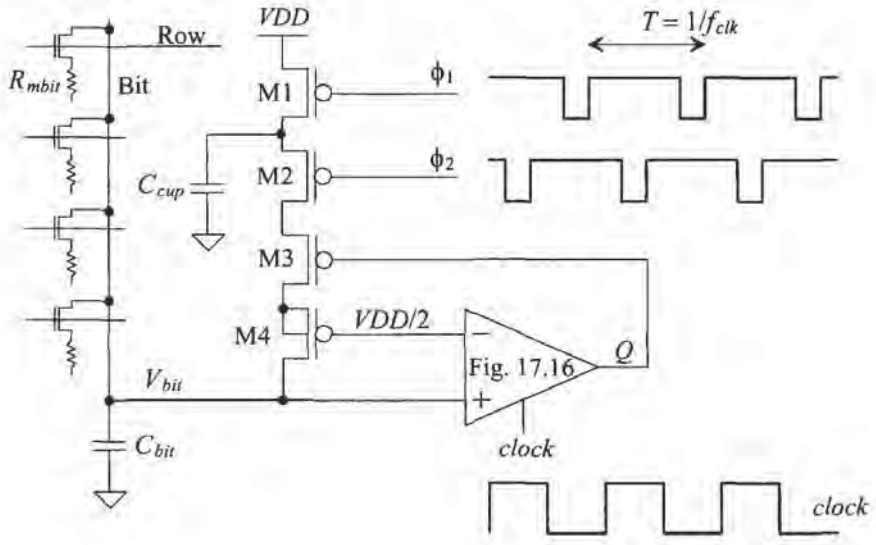



**Figure 17.18** Sensing circuit for resistive memory.

IPR2023-00697
Theta EX2019

current through the cell is then given by Eq. (17.16). Again, if $N$ is the total number of times the DSM is clocked and $M$ is the number of times M3 is enabled so that $C_{cup}$ can dump its charge into the bit line, then we can write (reviewing Eqs. [17.3] to [17.7], [17.12], and [17.16])

$$\frac{V_{OS}}{R_{mbit}} = Q_{cup} \cdot \frac{M}{N} \cdot \frac{1}{T} = (VDD - VDD/2 - V_{THP}) \cdot C_{cup} \cdot \frac{M}{N} \cdot \frac{1}{T} \qquad (17.17)$$

or

$$R_{mbit} = \frac{V_{OS} \cdot T}{(VDD/2 - V_{THP}) \cdot C_{cup} \cdot \frac{M}{N}} \qquad (17.18)$$

Notice that if $R_{mbit}$ goes to zero (programmed), the output of the DSM is always low ($M$ is always a 1). If $R_{mbit}$ is infinite, then the output of the DSM is always high ($M$ is always a 0). To design the DSM, we need to pick an $R_{mbit}$ value that separates what we define as a low resistance (a programmed state) from what we define as a high resistance (an erased state). Let's use 50k. When the resistance is 50k, then $M = N/2$ (half of the clock cycles the output of the DSM goes high). If $M > N/2$, then the cell is programmed. If $M < N/2$, the cell is erased. If $V_{OS}$ is 50 mV, $VDD = 1\ V$, $V_{THP} = 280$ mV, and $f_{clk} = 100$ MHz then

$$C_{cup} = \frac{0.05}{0.22 \cdot 50k \cdot \frac{1}{2}} \cdot 10\ ns = 90\ fF \qquad (17.19)$$

Since offset won't be precisely 50 mV, we'll round this up to 100 $fF$ (and it will vary with process runs). To estimate the value of the resistor given both $N$ and $M$, we use Eq. (17.18)

$$R_{mbit} \approx 25k \cdot \frac{N}{M} \qquad (17.20)$$

Again, this is an approximation since our value for the offset won't be exactly 50 mV. Note that the minimum value of resistor we can sense occurs when $N = M$ ($R_{mbit} = 25k$).

One more thing that we must calculate before simulating the design is the variation of the bit line voltage (see Eq. [17.8]). The minimum voltage on the bit line is $VDD/2$ (when $R_{mbit}$ is small). The maximum voltage will be $VDD/2 + V_{os} + \Delta V_{bit}$ (where the last term is the maximum change in the bit line voltage). To determine $\Delta V_{bit}$, we can write (knowing charge must be conserved)

$$C_{cup} \cdot (VDD - VDD/2 - V_{THP}) = \Delta V_{bit} \cdot C_{bit} \qquad (17.21)$$

or

$$\Delta V_{bit} = \frac{C_{cup}}{C_{bit}} \cdot (VDD/2 - V_{THP}) \qquad (17.22)$$

Using the numbers above with a $C_{bit}$ of 500 fF gives a $\Delta V_{bit} = 37\ mV$. Note that we were very concerned about how the bit line voltage change affected the amount of charge supplied by $C_{cup}$ (see Figs. 17.8, 17.9, and the associated discussion). Here (and Eq. [17.16]) we aren't concerned with how changes in the bit line voltage affect the current through $R_{mbit}$. The *average* current through $R_{mbit}$ is $V_{OS}/R_{mbit}$. The DSM holds the bit line, on average, at $VDD/2 + V_{OS}$. The charge supplied by $C_{cup}$, however, is not constant on average (as discussed earlier), and that is why M4 was added to the DSM.

IPR2023-00697
Theta EX2019

Figure 17.19 shows the DSM sensing circuit's output for various values of $R_{mbit}$. Let's compare the theoretical estimate, that is, Eq. (17.20) to the simulated results. In all of the simulations seen in Fig. 17.19 we clock the DSM 50 times (a 100 MHz clock for 500 ns). In (a), with $R_{mbit}$ = 25k, we get an output of 14 zeroes and 36 ones (= $M$). We can then write

$$R_{mbit} = 25k \cdot \frac{50}{36} = 35k \text{ (actual value 25k)}$$

For (b), we see $M$ = 17 so

$$R_{mbit} = 25k \cdot \frac{50}{17} = 73k \text{ (actual value 50k)}$$

and for (c) and (d),

$$R_{mbit} = 25k \cdot \frac{50}{10} = 125k \text{ (actual value 100k)}$$

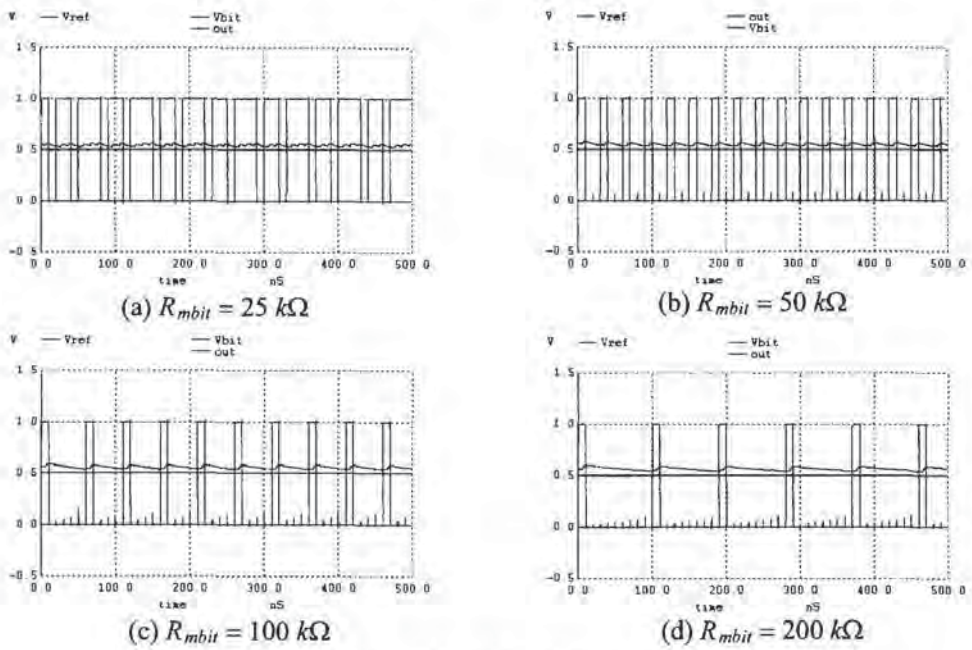
$$R_{mbit} = 25k \cdot \frac{50}{6} = 208k \text{ (actual value 200k)}$$

Notice that as we increase the value of $R_{mbit}$, we start to get nonlinear. The maximum finite value we can sense with 50 clock pulses is

$$R_{mbit,\max} = \frac{V_{OS} \cdot T \cdot N}{(VDD/2 - V_{THP}) \cdot C_{cup}} \text{ (= 1.25M here)} \qquad (17.23)$$

Again, the minimum resistance is

$$R_{mbit,\min} = \frac{V_{OS} \cdot T}{(VDD/2 - V_{THP}) \cdot C_{cup}} \text{ (= 25k here)} \qquad (17.24)$$




(a) $R_{mbit}$ = 25 kΩ (b) $R_{mbit}$ = 50 kΩ

(c) $R_{mbit}$ = 100 kΩ (d) $R_{mbit}$ = 200 kΩ

**Figure 17.19** Outputs of the DSM for various Rmbit values.

IPR2023-00697
Theta EX2019

Notice that we get good linearity with (actual) resistance values ranging from 25k to 75k. To sense larger values of resistances with good linearity, we must clock the DSM sensing circuit more times (increase $N$). However, if we are simply trying to separate a "big" resistor from a "small" resistor, we see that the scheme works very well.

*A Couple of Comments*

We might wonder why the simulated values are different from the actual values. The answers come from incomplete settling (making $C_{cup}$ appear smaller than it actually is) and from the fact that the offset voltage is not precisely 50 mV. For the incomplete settling problem we can clock the circuit slower or try using a larger width for M4. For the offset voltage problem we might try to generate a precise reference voltage ($VDD/2 + V_{OS}$) for the comparator. Practically, this could lead to problems. In a real memory system, the voltages are noisy. Reviewing the designs in Figs. 17.15 and 17.18, we see that if there is noise on $VDD/2$ it feeds evenly into the comparator circuit and doesn't affect the operation (the – input of the comparator is connected to $VDD/2$ and so is $R_{mbit}$). Further, even if we could generate noise-free reference voltages, the comparator will exhibit an offset because the MOSFETs won't be perfectly matched. In most sensing applications, the relative values are usually more important than the absolute values.

**Example 17.4**

Show that M1, M2, M3 and $C_{cup}$ in Fig. 17.18 can be thought of as a resistor. What is the resistor's value?

Figure 17.20 shows the circuit portion from Fig. 17.18. The current that flows in the resistor is

$$I_{avg} = \frac{VDD - VDD/2 - V_{THP}}{R_{SC}} \tag{17.25}$$

The amount of charge that flows during one clock cycle, if M3 is on, is

$$Q_{cup} = (VDD - VDD/2 - V_{THP}) \cdot C_{cup} \tag{17.26}$$

or, if this amount of charge is only allowed to flow $M$ times out of $N$

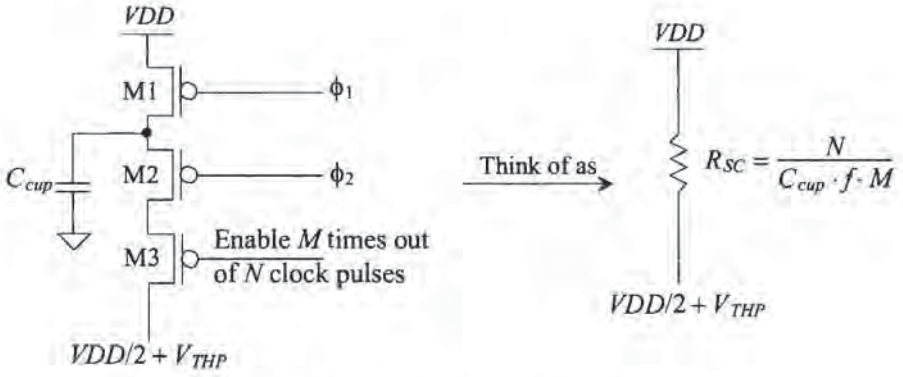



**Figure 17.20** How a switched-capacitor resistor is modeled.

IPR2023-00697
Theta EX2019

$$I_{avg} = \frac{Q_{cup}}{T} \cdot \frac{M}{N} = Q_{cup} \cdot f \cdot \frac{M}{N} \tag{17.27}$$

or

$$R_{SC} = \frac{N}{C_{cup} \cdot f \cdot M} \tag{17.28}$$

Noting that if M3 is always enabled (the circuit with only M1 and M2) we get a switched-capacitor resistance of

$$R_{SC} = \frac{1}{C_{cup} \cdot f} \tag{17.29}$$

which, for Fig. 17.18 and the associated discussion, is 25k. ■

**Example 17.5**

Show that M1, M2, and $C_{cup}$ can be replaced with a resistor in the DSM sensing circuit of Fig. 17.18. Show the DSM's output when $R_{mbit}$ is 50k and 200k.

The schematic for the DSM modulator is seen in Fig. 17.21. We might think that the errors due to incomplete settling would be eliminated using this scheme. However, the source of M4 will not be precisely at $VDD + V_{THP}$, so an error will still be present. Again, increasing the width of M4 will lessen the error's effects.

Using the same values as used to generate Fig. 17.19b and d, we get the simulation results seen in Fig. 17.22. The big benefit of this topology over the one in Fig. 17.18 using the switched capacitor resistor is simpler design (no nonoverlapping clocks are needed) and lower power (there are not as many parasitic capacitances to charge and discharge). The current pulled from *VDD* in Fig. 17.18 is 26 μA when clocked at 100 MHz, while the circuit in Fig. 17.21 uses 20 μA. The big drawback of using the simple resistor is the inability to adjust the resistance by changing the clock frequency. In a practical circuit the fabrication

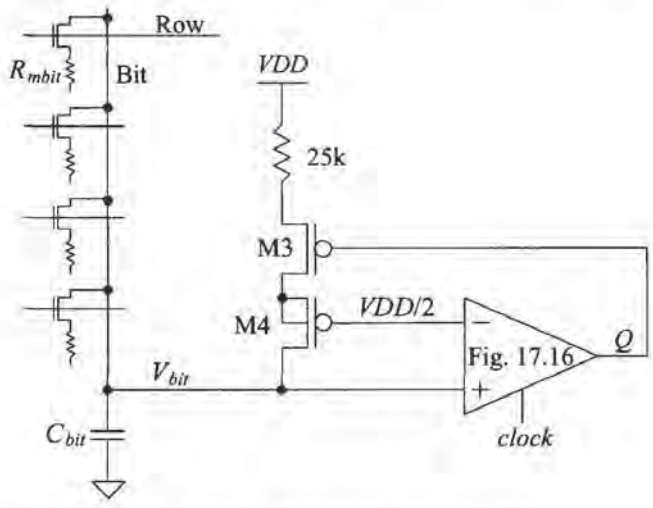
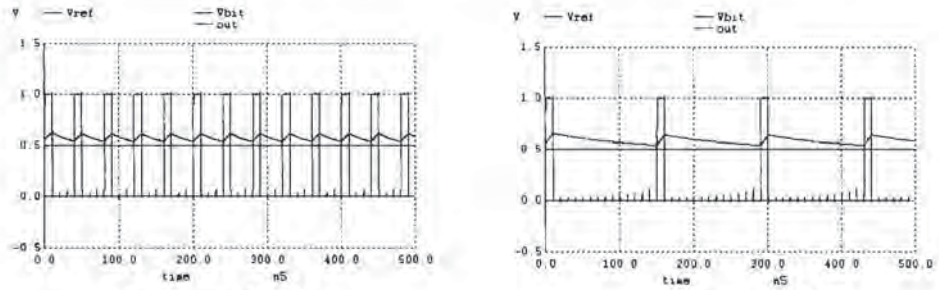



**Figure 17.21** Simpler DSM for sensing resistive memory.

IPR2023-00697
Theta EX2019




(a) $R_{mbit} = 50\ k\Omega$ (b) $R_{mbit} = 200\ k\Omega$

**Figure 17.22** Simulation results for the circuit in Fig. 17.21.

process characteristics and temperature vary. To center the value of sense to half way between a programmed and an erased resistor value, the clock frequency can be adjusted. Adjusting the clock frequency in Fig. 17.21 simply adjusts the variation in the bit line voltage. ■

## 17.3 Sensing in CMOS Imagers

Another area where DSM (delta-sigma modulation) can be used for sensing is in CMOS imaging chips that acquire images in cameras or video recording. A schematic of a CMOS active pixel sensor (APS) is seen in Fig. 17.23. Light is applied through a filter (so that only red, green, or blue light passes) to the photodiode. The photodiode changes the light intensity into a charge. The charge is converted into a voltage and passed to the column line. The brighter the light, the bigger voltage change we get on the column line.

*Resetting the Pixel*

Prior to acquiring an image each pixel in the imaging chip is reset. This is accomplished by driving the reset row line (*ResetN*) to a voltage, *VDDP*, greater than $VDD + V_{THN}$ (with body effect). This turns M1 on and sets the voltage across the photodiode to *VDD*. This condition is called the *dark or reference level* of the pixel. We can then turn M3 on by driving *RowN* to *VDDP* and, with M2 behaving as a source follower, driving the column line to the reference voltage level, $V_R$. This is important because each pixel in an imaging array will have slightly different characteristics. The differences in the pixels can result in speckles in the image. We can subtract this reference level from the actual measured signal level to get an accurate idea of the light intensity applied to the pixel (to eliminate the pixel gain differences).

*The point here is that our DSM circuit will have to sense, and store, a reference level, $V_R$, at the beginning of the sense.*

*The Intensity Level*

After we have stored the reference level voltage $V_R$ (on a capacitor), the *ResetN* signal goes low. This allows the light striking the reverse-biased photodiode (through the color filter) to generate electron-hole pairs that cause the voltage across the diode to decrease. After a time, the information stored on the gate of M2 (the decrease in the voltage across the photodiode) is sampled on the column line (assuming M3 is on). The time between

IPR2023-00697
Theta EX2019

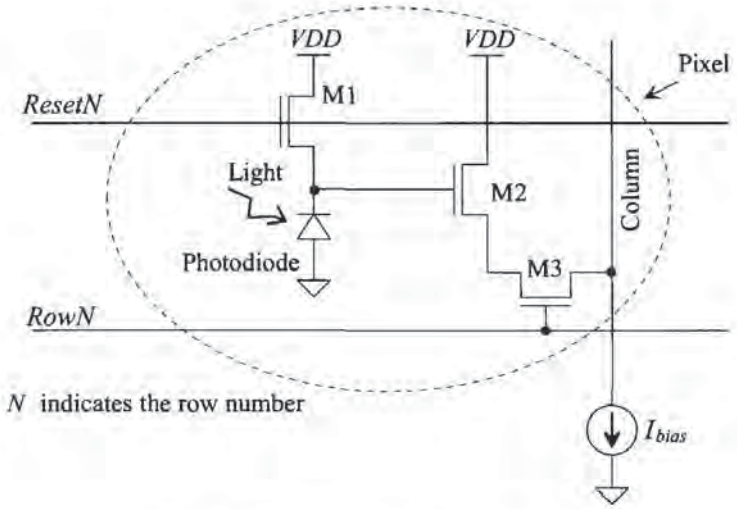



**Figure 17.23** A CMOS active pixel sensor (APS).

the *ResetN* going low and the data (the column voltage, $V_I$, corresponding to the intensity of the light striking the photodiode) being sampled on the column line is called the *aperture* time. Note that a dark signal corresponds to a large voltage on the column line ($VDD - V_{THN}$), while a bright signal corresponds to a lower voltage (less than $VDD - V_{THN}$).

*Sampling the Reference and Intensity Signals*

When *ResetN* and *RowN* are high, the pixel's reference (or dark) voltage, $V_R$, is placed on the column line, Fig. 17.24. At this time the sample and hold reference signal, *SHR*, goes high and $V_R$ is sampled onto a hold capacitor. Next, *ResetN* goes low and the photodiode changes light into charge. After the aperture time, the information from the photodiode (the intensity of the light) is on the column line, $V_I$. This voltage is then sampled and held on a hold capacitor when the signal *SHI* (sample and hold the intensity of the light) goes

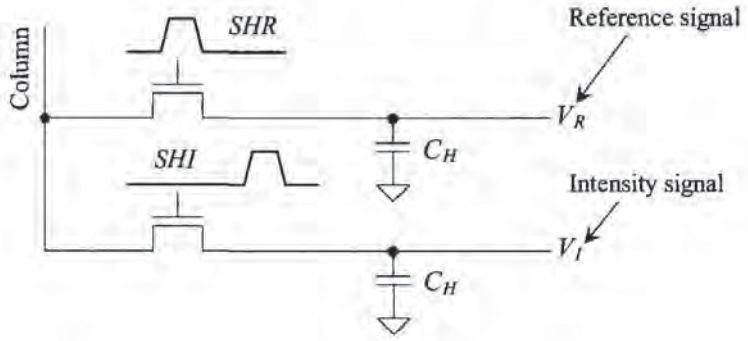



**Figure 17.24** Sampling the reference and intensity signals.

IPR2023-00697
Theta EX2019

high. What we want is to design a circuit that takes the difference in $V_R$ and $V_I$ and generates a digital number.

*Noise Issues*

Circuit noise limits the dynamic range of the sense, resulting in the blurring of the images or the inability to detect low light levels or distinguish between bright or high-intensity light levels. The major noise sources in CMOS, as discussed in Chs. 8 and 9, are flicker and thermal noises. The first design value we must select is the size of the hold capacitor in Fig. 17.24. The size of this capacitor limits the thermal noise in the sample. As seen in Table 8.1, using a hold capacitor of 1 pF results in an RMS noise voltage, just due to thermal noise, of 64 μV. This corresponds to a peak-to-peak voltage in the time domain of roughly 400 μV (six times the RMS value as seen in Fig. 8.33). Using a larger capacitor takes up more layout area and takes longer to charge but lowers the amount of thermal noise sampled onto $C_H$.

The output current of the pixel (when M3 is on and M2 is a source follower) also contains flicker noise. When this pixel is connected to the large hold capacitance and the capacitance of the bit line, the flicker noise current will be integrated. The result is a noise power spectral density with a $1/f^3$ spectral shape. As discussed in Ex. 8.14, the RMS value of the resulting noise signal will grow linearly with measurement time. What this means is that to *achieve a low noise* sample onto $C_H$ we want to minimize the amount of time *SHR* and *SHI* are high *and* the time difference between the two signals. *SHR* and *SHI* should only go high long enough to charge $C_H$. The amount of time between M1 turning on and *SHI* shutting off should be as small as possible.

The amount of noise in $V_R$ and $V_I$ cannot be reduced after they are captured on the hold capacitors. For example, ideally $V_I$ may be 0.5 V. However, because of noise $V_I$ may be 0.501 V or 0.495 V, etc. We *may* be able to reduce the noise by averaging successful samples though (see Eq. [8.48] and Ex. 8.14).

Ideally, the sensing circuitry (the circuitry used to change the analog voltages, $V_R$ and $V_I$, into a digital number) doesn't introduce any additional noise. When using the DSM, the counter can be thought of as a lowpass digital filter, Fig. 17.25. (See also the book entitled *CMOS Mixed-Signal Circuit Design* for much more detailed description of the frequency response of a counter.) As seen in this figure, if the sense time is the total number of times the DSM is clocked, $N$, multiplied by the clock's period $T$, that is $T_{meas} = NT$, then increasing the sense times lowers the bandwidth of the digital filter. This has the effect of lowering the noise bandwidth (so that the DSM sensing circuit does not contribute any further noise to the measured signal).

An example where a counter is used that doesn't result in filtering the measured signal is seen in Fig. 17.26. Here the column voltage corresponding to the intensity of the light is used as the – input to a comparator. For the + input, a constant current is used to charge a capacitor to generate a voltage ramp. When the ramp's voltage gets above $V_I$, the output of the comparator goes high and stops the counter. In this scenario the counter is simply used to give a digital representation of a time or ramp voltage. The counter doesn't provide any filtering (or more precisely averaging as used in DSM, see Tables 17.1 to 17.3). Notice also that this sensing topology very likely adds noise to the measured signal. For example, we know that using a constant current to charge a

IPR2023-00697
Theta EX2019

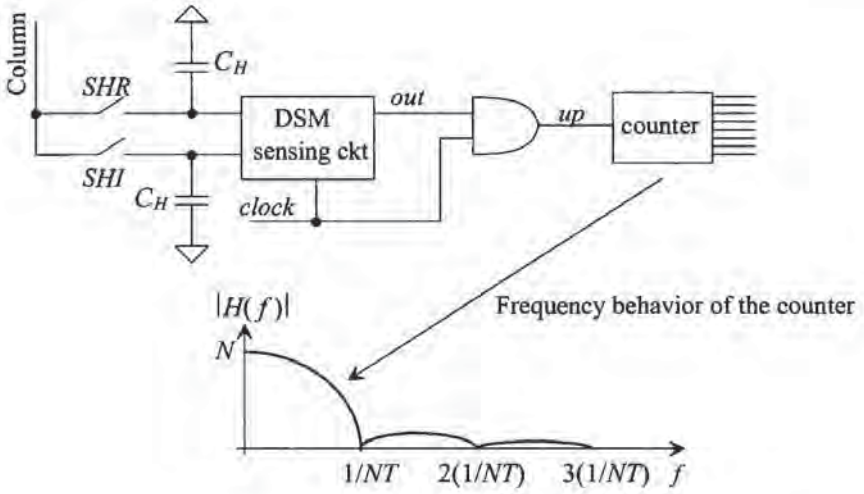



**Figure 17.25** Thinking of the counter as a digital low pass filter

capacitor results in a voltage signal containing $1/f^3$ noise. The ramp, with noise, voltage is connected directly to the comparator and causes noise to be added to the measured signal. Further, if the comparator makes an error and switches states too early or too late because of an offset or noise coupling from the adjacent column sensing circuits, again, the sense adds noise to the measured signal (the digital output code isn't constant but rather moves around even though the inputs to the sense amplifier may be constant).

Also note that using the DSM, we can run the sense operation indefinitely, while the scheme in Fig. 17.26 is limited by the ramp rate. Further increasing the clock frequency in Fig. 17.26 won't increase the resolution if the sense is noise limited. In the DSM sensing circuit, increasing the clock frequency, as seen in Fig. 17.25, lowers the bandwidth of the digital filter and increases the resolution of the sense.

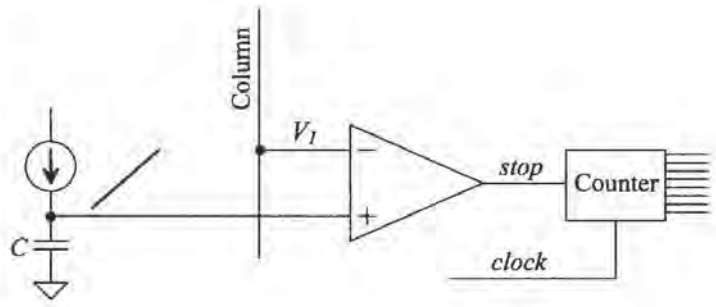



**Figure 17.26** A circuit where the counter doesn't behave like a lowpass filter.

IPR2023-00697
Theta EX2019

*Subtracting $V_R$ from $V_I$*

As we mentioned earlier, each pixel has slightly different characterisitics. For example, the threshold voltage of M2 in one pixel may be 10 mV different from the threshold voltage of M2 in a different pixel. To remove this error, we subtract the measured reference voltage, $V_R$, from the measured signal intensity, $V_I$. It's important, during the sense, not to change these voltages with our sensing circuit. Let's convert these voltages to currents and then subtract the currents to get the difference (and not try to subtract the voltages directly).

Examine the voltage-to-current converter seen in Fig. 17.27. This circuit is simply a source follower that is made wide so that its $V_{SG}$ is always approximately the threshold voltage. The relationship between the drain current and the column voltage $V_{col}$ ($V_I$ or $V_R$) is

$$I = \frac{VDD - V_{THP} - V_{col}}{R} \tag{17.30}$$

for

$$V_{col} < VDD - V_{THP} \tag{17.31}$$

Reviewing Fig. 17.23, we see that if the threshold voltage drop of M2 (with body effect) is more than $V_{THP}$ (without body effect because we placed the PMOS device in Fig. 17.27 in its own well), Eq. (17.31) will always be satisfied.

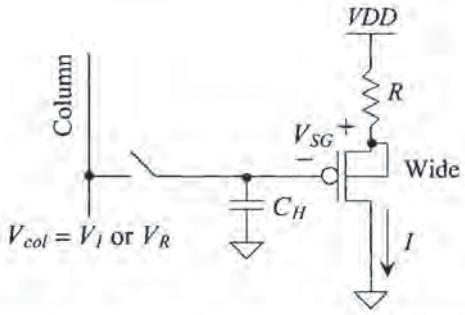



**Figure 17.27** Linear voltage-to-current conversion.

**Example 17.6**

Using SPICE determine the linearity of the current in the circuit seen in Fig. 17.28. Use the short-channel CMOS process and compare the simulation results to hand caculations.

If *VDD* is 1 V, $V_{THP}$ = 250 mV, and $R$ is 10MEG, then

$$I = 75 - 100 \cdot V_{col}\ nA$$

Noting that the ideal slope of the line is $-1/R$ ($= -100 \times 10^{-9}$). Figure 17.29 shows the simulation results. In (a) we see from the transfer curve how the output, $I$, changes with the input, $V_{col}$. In (b) we take the derivative of $I$ to see if the slope is perfectly flat. The linearity is pretty good (1%) for $V_I < 400$ mV. ■

IPR2023-00697
Theta EX2019

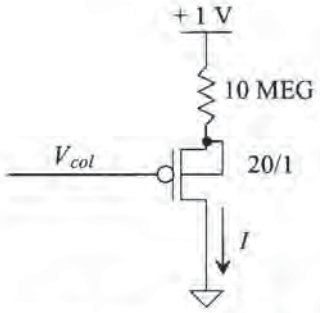
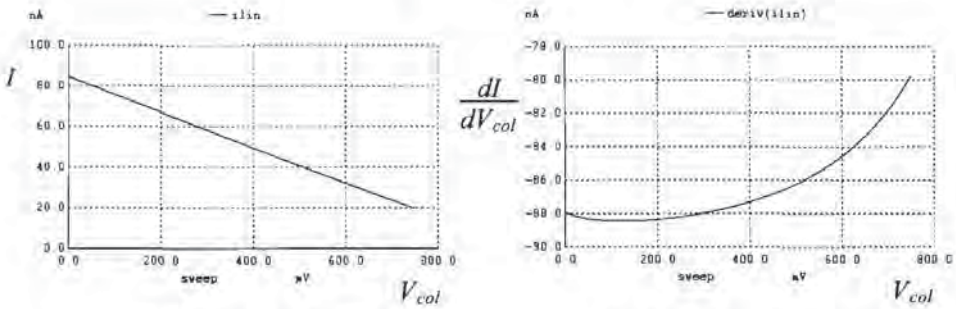



**Figure 17.28** Circuit used in Ex. 17.6.




(a) Showing transfer curve (b) Taking the derivative of (a)

**Figure 17.29** Simulating the operation and linearity of the voltage-to-current converter in Fig. 17.28.

We might wonder, from the last example, if it would be a good idea to try to make the linearity of our voltage-to-current converter even better. Before doing this, let's look at the linearity of the source follower in the pixel itself. Figure 17.30 shows a simplified schematic of the source follower (M2) used in the APS. We've modeled the finite output resistance of the current source with a 10MEG resistor. The simulation results in Fig. 17.31 show the transfer relation of this circuit and its linearity. For column voltages between 100 mV and 600 mV, the linearity is 0.5% (which is comparable so we won't concern ourselves any further with trying to better the linearity at this point).

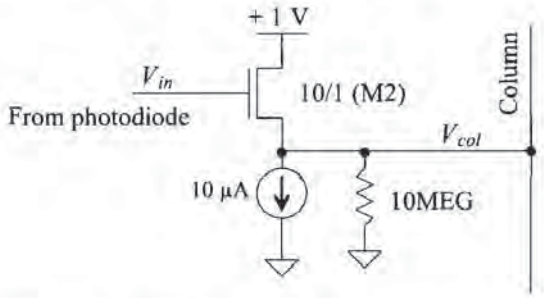



**Figure 17.30** Looking at the linearity of the source follower in the pixel.

IPR2023-00697
Theta EX2019

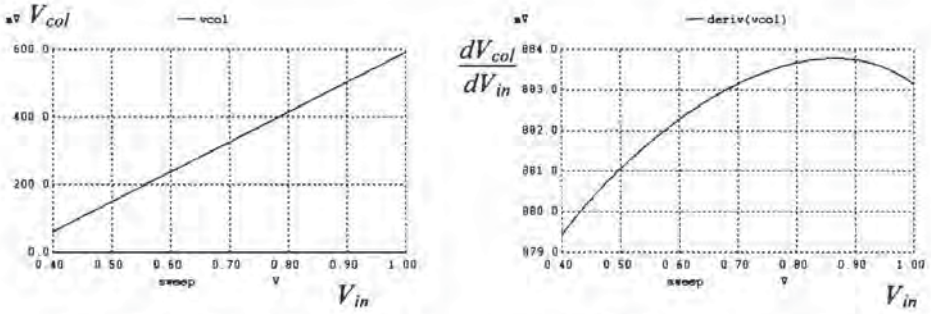



**Figure 17.31** Showing the transfer curve and linearity of the circuit in Fig. 17.30.

To take the difference in the currents, let's use a current mirror as seen in Fig. 17.32. The current corresponding to the reference voltage is

$$I_R = \frac{VDD - V_{THP} - V_R}{R_R} = \frac{V_{R,shift}}{R_R} \tag{17.32}$$

and the current corresponding to the intensity of light is

$$I_I = \frac{VDD - V_{THP} - V_I}{R_I} = \frac{V_{I,shift}}{R_I} \tag{17.33}$$

The difference in these currents is summed (sigma) in the bucket capacitor, as seen in Fig. 17.32. When we add our comparator and feedback to form a DSM, the charge on the capacitor, averaged over time, is a constant. This occurs when

$$I_R = I_I = \frac{V_{R,shift}}{R_R} = \frac{V_{I,shift}}{R_I} \tag{17.34}$$

or

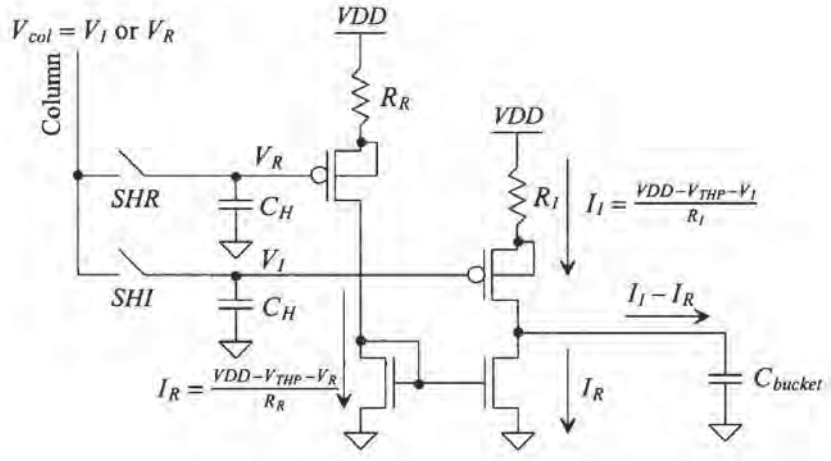



**Figure 17.32** Subtracting the currents.

IPR2023-00697
Theta EX2019

$$V_{I,shift} = \frac{R_I}{R_R} \cdot V_{R,shift} \tag{17.35}$$

The ratio of the resistances gives us the information we need to determine the relative (to the reference level, $V_R$) intensity of light on the pixel. To implement the resistors, let's use the switched-capacitor resistor as seen in Fig. 17.20. For the reference voltage we know

$$V_{I,shift} \geq V_{R,shift} \tag{17.36}$$

and so ($R_I \geq R_R$). Let's use (M3 always on in Fig. 17.20)

$$R_R = \frac{1}{f \cdot C_{cap}} \tag{17.37}$$

and for $R_I$ (which *will* be enabled via M3 when its gate goes low)

$$R_I = \frac{1}{f \cdot C_{cap} \cdot \frac{\overline{M}}{N}} \tag{17.38}$$

Rewriting Eq. (17.35), knowing $\overline{M} = N - M$, gives

$$V_{I,shift} = V_{R,shift} \cdot \frac{N}{N - M} \tag{17.39}$$

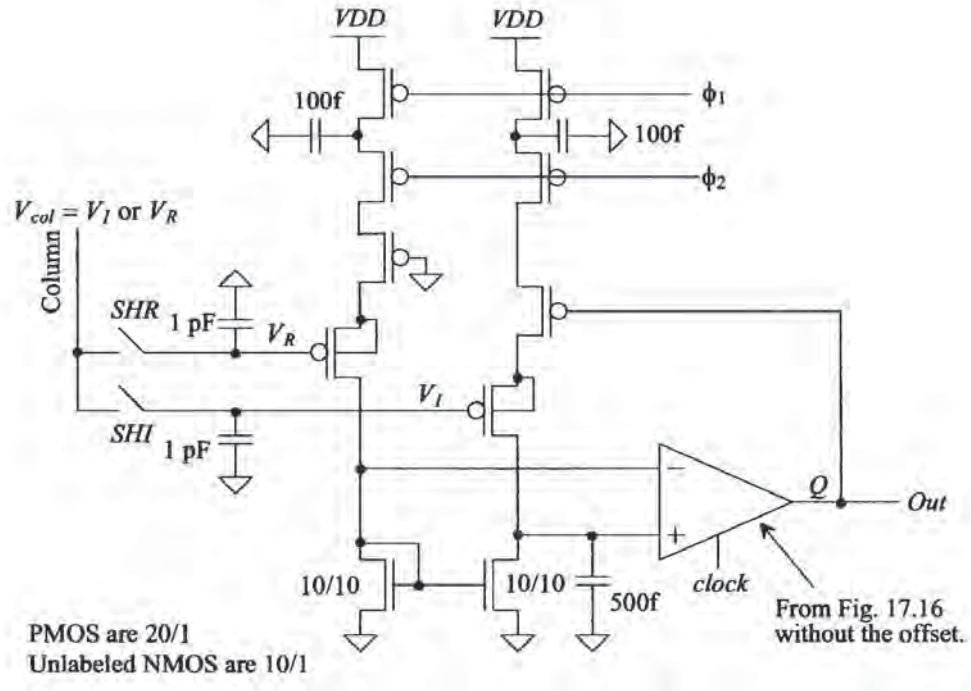


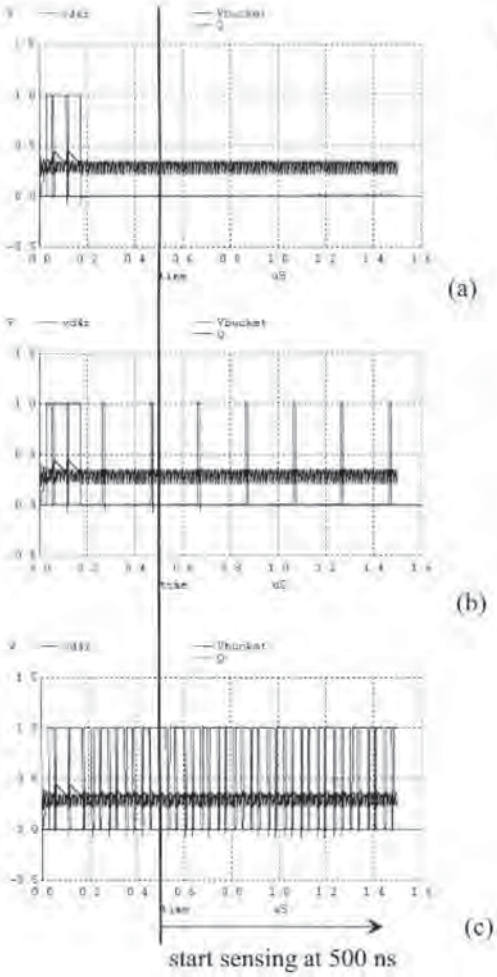

**Figure 17.33** Schematic of a DSM for sensing in CMOS imaging chips.

IPR2023-00697
Theta EX2019

If the DSM is clocked 1,000 times ($N$), then $M$ can range from 0 (the pixel is not illuminated, that is, the dark or reference level) to 1,000 (very bright). Figure 17.33 shows a schematic of the sensing circuit with some typical values.

The circuit in Fig. 17.33 was made as symmetrical as possible so that power supply or ground noise affected each signal path equally. The MOSFETs in the current mirror are made long (10 drawn) so that the voltages on the input of the comparator are greater than the NMOS threshold voltage. The size of the capacitors isn't that critical. We want the $C_{cup}$ capacitors to be less than the $C_{bucket}$ capacitor. We don't have to worry about overcharging $C_{bucket}$ in this scheme because the input signal contributions are limited by the switched capacitor resistors. (We only get one $C_{cup}$ every clock cycle.)

Figure 17.34 shows the simulation results for the DSM circuit of Fig. 17.33. We assume that the counter, Fig. 17.25, is enabled after 500 ns. Prior to this time, the




$$V_R = 650\ mV \qquad V_I = 650\ mV$$
$$V_{R,shift} = VDD - V_{THP} - V_R = 100\ mV$$
$$V_{I,shift} = VDD - V_{THP} - V_I = 100\ mV$$
$$V_{I,shift} = 100\ mV \cdot \frac{100}{100 - 0} = 100\ mV \quad \text{(sensed)}$$

$$V_R = 650\ mV \qquad V_I = 645\ mV$$
$$V_{R,shift} = VDD - V_{THP} - V_R = 100\ mV$$
$$V_{I,shift} = VDD - V_{THP} - V_I = 105\ mV$$
$$V_{I,shift} = 100\ mV \cdot \frac{100}{100 - 5} = 105\ mV \quad \text{(sensed)}$$

$$V_R = 650\ mV \qquad V_I = 400\ mV$$
$$V_{R,shift} = VDD - V_{THP} - V_R = 100\ mV$$
$$V_{I,shift} = VDD - V_{THP} - V_I = 350\ mV$$
$$V_{I,shift} = 100\ mV \cdot \frac{100}{100 - 72} = 357\ mV \quad \text{(sensed)}$$

**Figure 17.34** How the DSM sensing circuit in Fig. 17.33 operates.

IPR2023-00697
Theta EX2019

reference and intensity signals are sampled onto the hold capacitors. If $V_R$ = 650 mV, then

$$V_{R,shift} \approx 1 - 0.25 - 0.65 = 100\ mV$$

In Fig. 17.34a we apply the same signal to the DSM sensing circuit, that is 650 mV, for $V_I$. As expected, the output stays low for all times. In (b) we drop $V_I$ to 645 mV and see, during the 500 ns to 1,500 ns sensing time, 5 output ones. As seen in the figure, the sensed value indicates the intensity is 5 mV below the reference. In (c) we drop the intensity signal to 400 mV resulting in $V_{I,shift}$ = 350 mV. The sensed value with 72 of the 100 clock cycles being high is 357 mV. Let's look at what would happen if we sensed 71 ones

$$100\ mV \cdot \frac{100}{100-71} = 345\ mV$$

In either case (71 or 72 ones) the resolution was so coarse that we couldn't resolve the actual signal. If we think about this for a moment, we see that if the counter output code is small then the resolution of the measurement is better. For example, counter outputs of 1 and 2 result in

$$100\ mV \cdot \frac{100}{100-1} = 101\ mV \text{ and } 100\ mV \cdot \frac{100}{100-2} = 102\ mV$$

Looking at Eq. (17.39), we see that the dependence on $M$ (the number of times the output of the DSM goes high) is *not linearly related* to the light intensity, $V_I$. What we want is an equation like (17.7), that is,

$$V_{I,shift} = V_{R,shift} \cdot \frac{M}{N} \tag{17.40}$$

To get a relationship like this, we might try to control the value of $R_R$ too as seen in Fig. 17.35. The complementary output of the comparator is fed back so that

$$R_R = \frac{1}{f \cdot C \cdot \frac{M}{N}} \tag{17.41}$$

and thus

$$V_{I,shift} = V_{R,shift} \cdot \frac{M}{N-M} \tag{17.42}$$

Again, we do not have a linear relationship. Further, half or more of our outputs must be used to enable the switched-capacitor resistor in series with the reference signal. If, for example, $V_{R,shift} = V_{I,shift}$, then $M = N/2$. Since $V_R \geq V_I$, then $0 \leq M \leq N/2$ (not good design).

If we review the derivations leading up to Eqs. (17.7) and (17.20), the common theme is that the feedback signal controlled by the comparator is a constant addition to the capacitive bucket (or bit line). In Fig. 17.18, for example, we used M4 to ensure that the charge from $C_{cup}$ was a constant added to $C_{bit}$. In our current sense amplifier, Fig. 17.33, the signal we feedback is not a constant but rather a function of $V_I$. When sensing in a CMOS imager, the signal fed back should be a function of the reference level, $V_R$. The comparator's output should control a fed-back signal that is derived from $V_R$.

IPR2023-00697
Theta EX2019

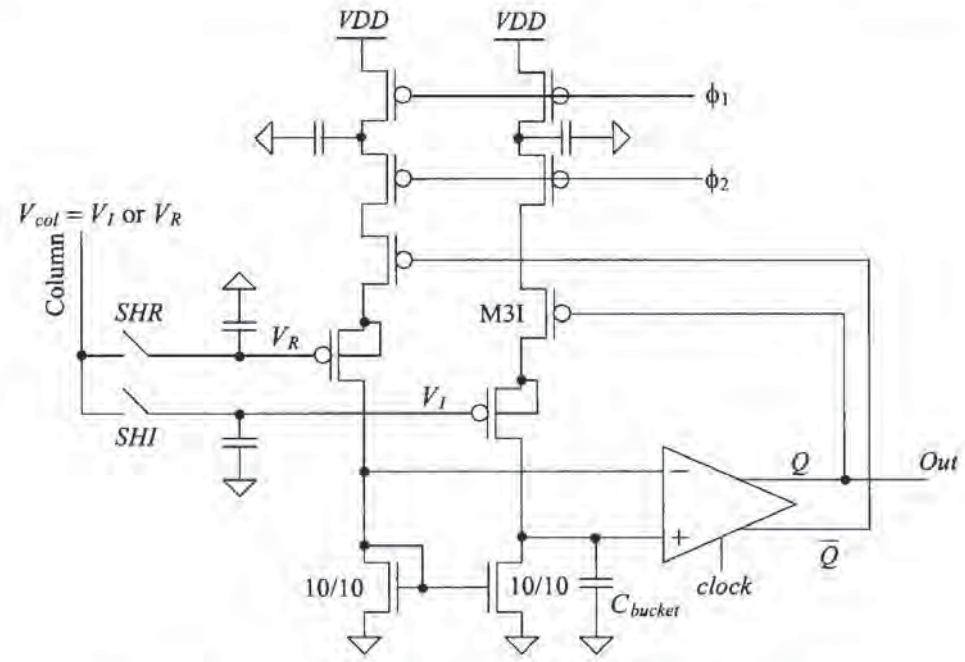



**Figure 17.35** Using both comparator outputs for a DSM sensing circuit.

Note that it's a bad idea to ground the gate of M3I in Fig. 17.35 and only have a single feedback path. Since $V_R > V_I$, we won't be able to supply enough current through the reference signal path. The charge supplied by the intensity path will always be greater than the charge supplied from the reference path (and so $C_{bucket}$ will overfill). A new topology is needed.

Figure 17.36 shows an NMOS version of Fig. 17.33. We've replaced the capacitors with MOSFETs to show that the DSM sensor can be implemented using a single-poly CMOS process. The PMOS devices are used for the "cup" capacitors. We use PMOS instead of NMOS because, for the topology seen, the PMOS devices always remain in strong inversion. The NMOS devices, for example, move towards accumulation mode when $\phi_2$ turns on and discharges the capacitors. The result is a nonlinear capacitance (the size of the cup varies). Similarly, we use NMOS for the bucket capacitors because, for the topology used, they will always remain in strong inversion (the capacitance won't vary with the changes in the voltage on their gates). Note that a second "bucket" capacitor was added across the 10/10 diode connected (gate and drain tied together) PMOS device. This addition serves two purposes. The first is to ensure that ground noise affects the comparator inputs equally (noise on ground will feed evenly through the bucket capacitors to the input of the comparator). The second reason is that it smoothes out the summation of the currents. Note that if the added capacitor is too big, stability can be an issue (the added capacitor adds a delay in series with the feedback path).

IPR2023-00697
Theta EX2019

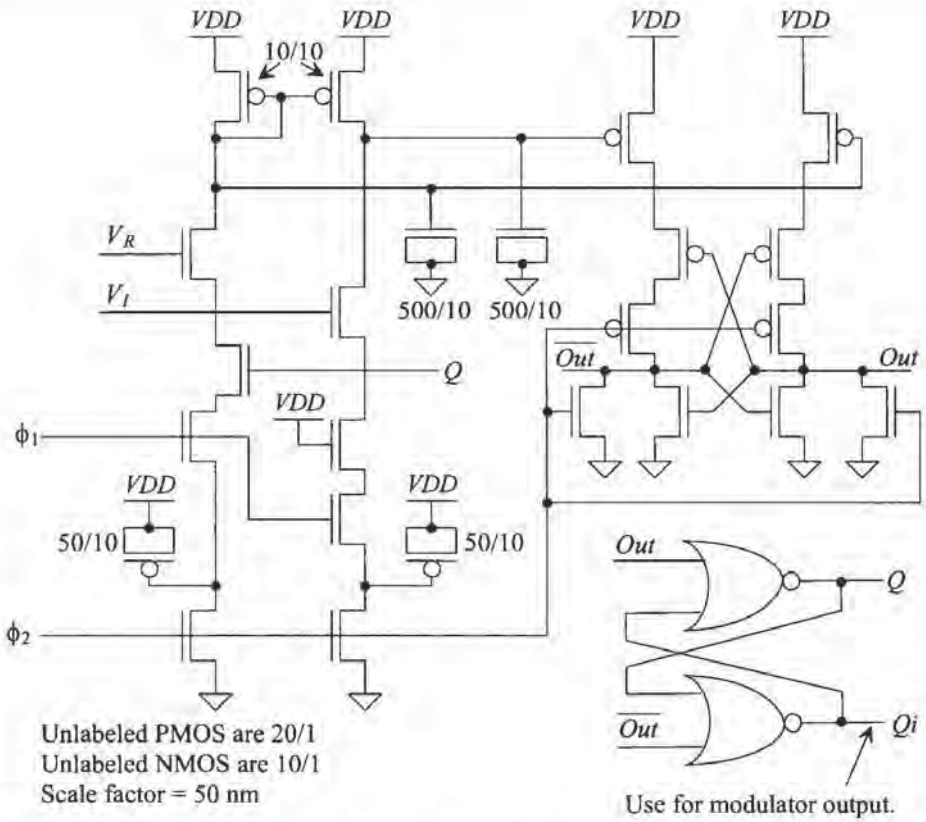


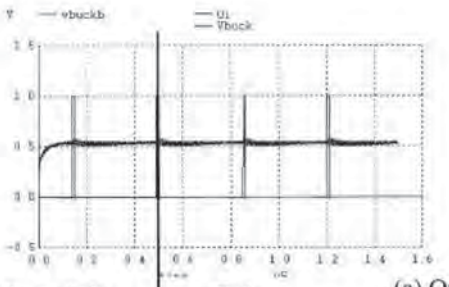

**Figure 17.36** DSM circuit for sensing in a CMOS imaging chip.

In Fig. 17.36 we show a comparator design using PMOS imbalance MOSFETs (as seen in Fig. 16.32 for the NMOS flavor). When $\phi_2$ goes high, the outputs of the comparator, *Out* and $\overline{Out}$ are driven low. When $\phi_2$ goes low, on the falling edge, the comparator makes a decision concerning which of the bucket capacitors has the higher potential across it. Based on this decision, the outputs of the comparator are latched with the NOR-based SR latch. The *Q* output is fed back to enable or disable the summation of charge via the reference path. We've used the *Qi* output as the DSM's output. When the intensity level is the same as the reference level, the output stays low for all times. It may be possible to eliminate this SR latch in a production part. However, we include it here because it makes the simulation results easier to look at (the glitches in the comparator's output are reduced).

Figure 17.37 shows some simulation results for the DSM circuit seen in Fig. 17.36. Notice, in this figure, that we've *assumed* the threshold voltage of the NMOS device is 250 mV. Since our reference level (the black level for the pixel) is 650 mV, our shifted reference level, $V_{R,shift}$, is 400 mV. For 100 samples then we can estimate the resolution as

$$V_{res} = \frac{V_{R,shift}}{N} \rightarrow 4\ mV \tag{17.43}$$

IPR2023-00697
Theta EX2019



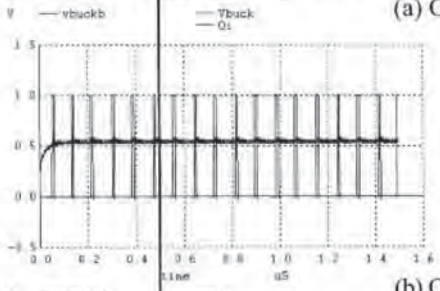
$V_R = 650\ mV \qquad V_I = 640\ mV$

$V_{R,shift} = V_R - V_{THN} = 400\ mV$

$V_{I,shift} = V_I - V_{THN} = 390\ mV$

$V_{I,shift} = 400\ mV \cdot \frac{98}{100} = 392\ mV$ (sensed)

(a) Output goes high 2 times out of 100 (M = 98).

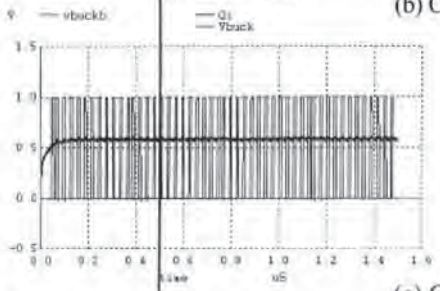


$V_R = 650\ mV \quad V_I = 600\ mV$

$V_{I,shift} = V_I - V_{THN} = 350\ mV$

$V_{I,shift} = 400\ mV \cdot \frac{88}{100} = 352\ mV$ (sensed)

(b) Output goes high 12 times out of 100 (M = 88).

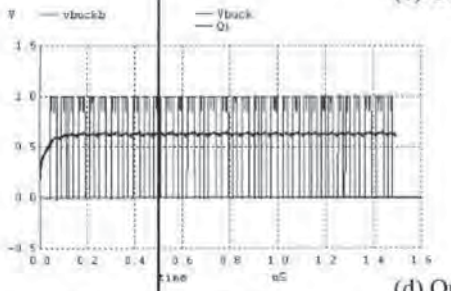


$V_R = 650\ mV \quad V_I = 500\ mV$

$V_{I,shift} = V_I - V_{THN} = 250\ mV$

$V_{I,shift} = 400\ mV \cdot \frac{65}{100} = 264\ mV$ (sensed)

(c) Output goes high 35 times out of 100 (M = 65).

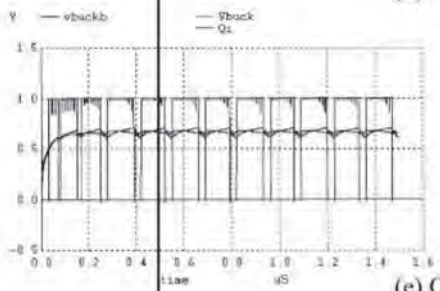


$V_R = 650\ mV \quad V_I = 400\ mV$

$V_{I,shift} = V_I - V_{THN} = 150\ mV$

$V_{I,shift} = 400\ mV \cdot \frac{44}{100} = 176\ mV$ (sensed)

(d) Output goes high 56 times out of 100 (M = 44).



$V_R = 650\ mV \quad V_I = 300\ mV$

$V_{I,shift} = V_I - V_{THN} = 50\ mV$

$V_{I,shift} = 400\ mV \cdot \frac{24}{100} = 96\ mV$ (sensed)

(e) Output goes high 76 times out of 100 (M = 24).

Start sensing at 500 ns (enable the counter)

**Figure 17.37** The operation of the DSM sensing circuit in Fig. 17.36.

IPR2023-00697
Theta EX2019

Looking at the figure, we may think that there is a large nonlinearity in the sense because, for lower input voltages, $V_{I,shift}$, the sensed value doesn't exactly match the shifted value. However, notice that the input changes from, say, 650 mV to 600 mV we get a code difference of 12 (24/100 mV), or from 600 mV to 500 mV (23/100 mV), from 400 mV to 300 mV (20/100 mV). To get a better estimate for the resolution, let's use an average change of 23 counts per 100 mV to estimate the resolution, that is,

$$V_{res} = \frac{100\ mV}{23} = 4.35\ mV \tag{17.44}$$

and so the shift in the reference voltage can be more accurately predicted as

$$V_{R,shift} = 435\ mV \text{ because } (= N \cdot V_{res}) \text{ so } V_{THN} = 215\ mV \tag{17.45}$$

Using this, the sensed outputs in Fig. 17.37 can be rewritten as:

(a), $V_{I,shift} = 640\ mV - 215\ mV = 425\ mV$ and the sensed value $435\ mV \cdot \frac{98}{100} = 426\ mV$

(b), $V_{I,shift} = 600\ mV - 215\ mV = 385\ mV$ and the sensed value $435\ mV \cdot \frac{88}{100} = 382\ mV$

(c), $V_{I,shift} = 500\ mV - 215\ mV = 285\ mV$ and the sensed value $435\ mV \cdot \frac{65}{100} = 283\ mV$

(d), $V_{I,shift} = 400\ mV - 215\ mV = 185\ mV$ and the sensed value $435\ mV \cdot \frac{44}{100} = 191\ mV$

(e), $V_{I,shift} = 300\ mV - 215\ mV = 85\ mV$ and the sensed value $435\ mV \cdot \frac{24}{100} = 104\ mV$

Indicating a linear sense that becomes nonlinear at the edges of operation (when $V_I$ becomes comparable to the $V_{THN}$).

It's important to understand the robustness of this sensing scheme. If the comparator makes a mistake, it is averaged out (comparator gain and offset aren't important). If noise is coupled into the sense amplifier, it will be averaged out. The sensing operation can be indefinite (noting that the hold capacitor voltages changing because of charge leaking off of the capacitors will ultimately limit the length of the sense). To increase the resolution of the sense, the clock frequency can be increased (noting the size of the counter used must be increased too). Finally, the topology requires little power. For the topology in Fig. 17.36 the current supplied by *VDD* is approximately 25 μA. If 1,000 of these sense amplifiers are used at the same time (say on the bottom of an imaging array with 1,000 columns), then the current required from *VDD* is 25 mA. The current can be further reduced by designing the DSM without the NOR latch seen in Fig. 17.36 or by using smaller capacitors.

The input-referred thermal noise is set by the sampling capacitors and is characterized using $kT/C$ (see Table 8.1) and the associated discussion. We might think that using smaller capacitors results in an increase in the thermal noise. However this noise is averaged by $N$, the number of clock cycles (the counter), so we can rewrite Eq. (8.24) to show the decrease in the thermal noise with averaging as

$$V_{onoise,RMS} = \sqrt{\frac{kT}{NC}} \tag{17.46}$$

IPR2023-00697
Theta EX2019

*Sensing Circuit Mismatches*

The point of sampling the reference or dark level and then subtracting the desired signal (the intensity of light on the pixel) was to subtract out mismatches in the pixel. For example, M2 in one pixel, may have a threshold voltage of 250 mV, while in a different pixel the threshold voltage may be 230 mV. The result, without the subtraction, would be two pixels with different output voltages even though the light applied to each is exactly the same. After thinking about this for a moment, we might realize that the DSM sensing circuit, having two separate paths for the reference and intensity signal paths, will also be subject to a mismatch. If, for example, one sensing circuit on the bottom of a column in

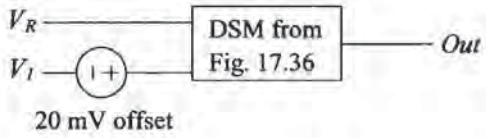



**Figure 17.38** Modeling the differences in the signal paths in the DSM of Fig. 17.36 with a voltage in series with the intensity signal path.

an imaging chip has different characteristics than the sensing circuits directly adjacent to it, then the image will show vertical streaks. We might try, using layout techniques, to reduce the mismatch. However, the human eye is very perceptive and will likely detect any differences in the sense (especially if the image is a single color). This is why the majority of imaging chips used a single pipeline ADC (see Ch. 29) operating at a high conversion rate at the time of this writing. Each pixel sees the exact same sensing circuit.

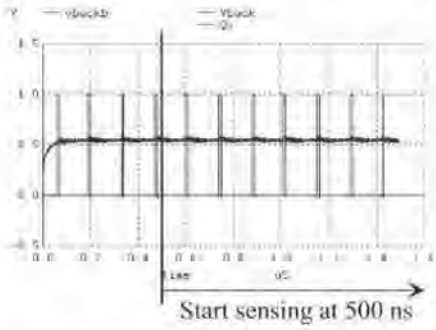



**Figure 17.39** How a 20 mV offset changes the outputs in Fig. 17.37a.

To illustrate this problem, let's resimulate the DSM in Fig. 17.36 with an offset, as seen in Fig. 17.38. This offset voltage, which is an unknown that may be positive or negative, simply models a random difference in the signal paths. Using the input values seen in Fig. 17.37a and a 20 mV offset, we get the simulation results seen in Fig. 17.39. Instead of getting two outputs going high, we now get seven.

To reduce the effects of mismatch on the sense, consider, halfway through the sense time, switching the inputs to the DSM, as seen in Fig. 17.40. By switching the inputs halfway through the sense, the effects of path mismatch average, ideally, to zero.

IPR2023-00697
Theta EX2019

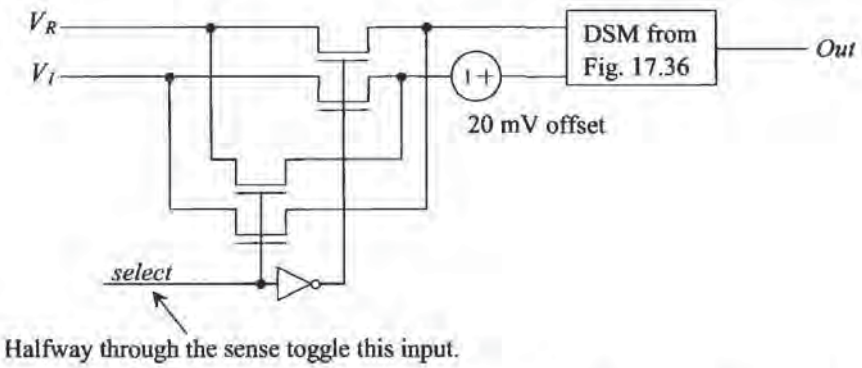



**Figure 17.40** Switching the inputs of a DSM to eliminate path mismatch.

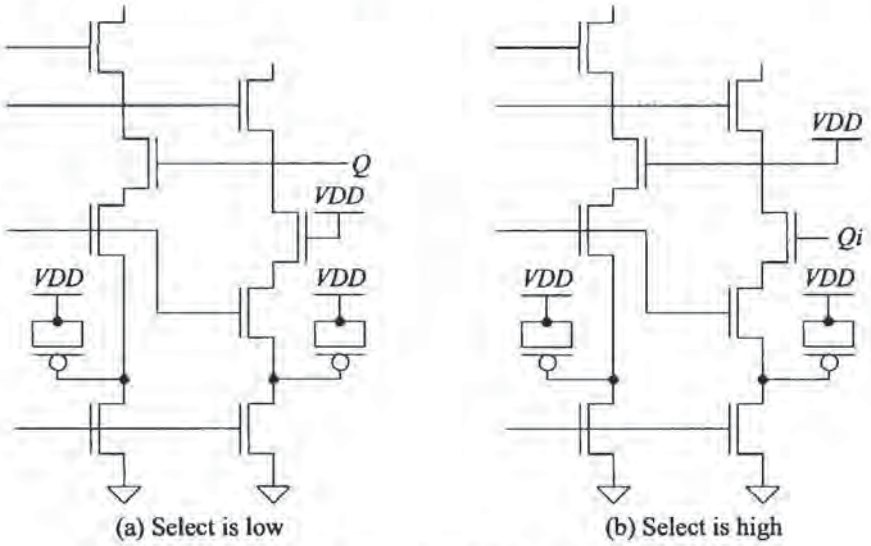



**Figure 17.41** The change in the feedback with change in select.

Note that by switching the inputs we also have to change the feedback internal to the DSM. As seen in Fig. 17.36, we must switch the gate connections of $Q$ and $VDD$ in series with the switched capacitor resistors, Fig. 17.41, (the outputs of the NOR latch are switched).

Note at signal intensities *close to the reference*, $V_R$, that the averaging will not work out with just an up counter (as seen in Fig. 17.3). The signal, $V_I$, must move away from the reference by more than $V_{OS}$. For example, if we swap the inputs to the DSM at one μs in the simulation seen in Fig. 17.39, then the output of the DSM after 1 μs is always low and the output code is 3 (the ideal output code from Fig. 17.37a is 2). This (the averaging won't work unless the $|V_R - V_I| < V_{OS}$) shouldn't be a problem since the output codes at these levels correspond to dark signals. Again note that feeding back both $Q$ and $Qi$, Fig. 17.35, results in a nonlinearity, as seen in Eq. (17.42).

IPR2023-00697
Theta EX2019

Finally note that whenever starting or switching the inputs during a sense operation there will be a start-up transient (see Fig. 17.34 for example). What this means is that the counter should be disabled at the beginning of the sense or in the middle (if the inputs to the DSM are swapped, as seen in Fig. 17.40, halfway through the sense operation). Not disabling the counter during these times can result in sensing errors.

## ADDITIONAL READING

[1] R. J. Baker, *CMOS: Mixed-Signal Circuit Design, Second Edition*, Wiley-IEEE Press, 2009.

[2] R. J. Baker, "Method and system for reducing mismatch between reference and intensity paths in analog to digital converters in CMOS active pixel sensors," US Patent 7,515,188, April 7, 2009.

[3] J. Taylor and R. J. Baker, "Method and apparatus for sensing flash memory using delta-sigma modulation," US Patent 7,366,021, April 29, 2008.

[4] R. J. Baker, "Per column one-bit ADC for image sensors," US Patent 7,456,885, November 25, 2008.

[5] R. J. Baker, "Resistive memory element sensing using averaging," US Patent 6,504,750, January 7, 2003.

## PROBLEMS

**17.1** Regenerate Table 17.1 in which the water level where a cup of water is removed is 4.7 instead of 5. How are the results affected? If the sensing time is increased, how are the final results affected (compare a water level of 5 against 4.7).

**17.2** Generate a table, similar to Table 17.1, for the situation seen in Fig. 17.4 if the amount of water leaving the bucket is 0.3 cups per 10 seconds.

**17.3** Rederive Eqs. (17.3) – (17.8) if $I_{cup}$ in Fig. 17.6 is replaced with a resistor. Assume that the clock frequency is large (why?) to simplify the equations. What is the requirement for the current through the resistor when it is connected to the bit line in terms of the maximum bit current, $I_{bit}$.

**17.4** Using SPICE simulations demonstrate that the error because of parasitics, as seen in Fig. 17.7, is reduced by connecting the switch to a 1 V source instead of ground. Illustrate, with drawings, what is happening.

**17.5** Show, using simulations, that if the output of the comparator swings from *VDD* to *VDD*/2 we can eliminate M4 in Fig. 17.9 and still have $Q_{cup} = C_{cup} \cdot (VDD - V_{REF} - V_{THP})$. Why? Does the amount of current supplied by *VDD*/2 increase? Could this be a problem?

**17.6** Show how the incomplete settling seen in Fig. 17.13 can be made more complete by reducing the clock frequency or increasing the width of M4.

**17.7** Demonstrate, using SPICE and discussions, that the circuit in Fig. 17.42 may be used in place of the comparator and M3 in Fig. 17.18. How do output glitches affect the sensing circuit's operation?

IPR2023-00697
Theta EX2019

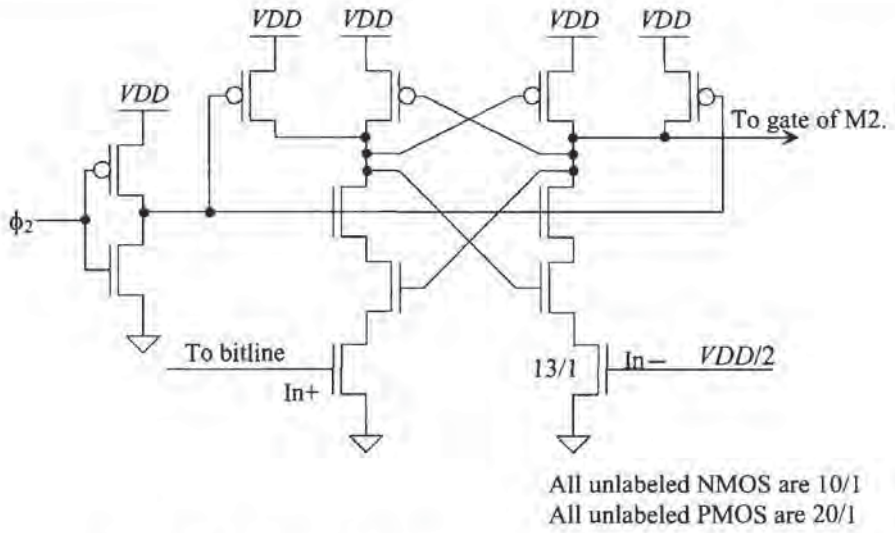



**Figure 17.42** Simplifying the comparator in Figs. 17.16 and 17.18.

**17.8** Design a DSM sensing circuit that will determine the value of a resistor that may range from 100k to 10 MΩ. Simulate your design with SPICE and comment on the design trade-offs concerning operating frequency and resistor (both the sensed and, if used in the DSM, the feedback resistance) changes with process variations or temperature.

**17.9** Suppose that it is desired to have a noise floor of 100 μV RMS in a CMOS imager. Further suppose that the sensing circuit doesn't contribute any noise to the sense (the transformation from the analog column voltage to a digital word). Estimate the size of the hold capacitors used to sample both the reference and the intensity signals.

**17.10** Using simulations, determine if the linearity of the voltage-to-current converter can be made better by adjusting the length and width of the PMOS device seen in Fig. 17.28. Why does, or doesn't, the performance get better?

**17.11** Using the short-channel CMOS devices determine the average current that flows in the circuit seen in Fig. 17.43. The clocks are nonoverlapping (never low at the same time) as used throughout the chapter and have a frequency of 100 MHz. Verify your answer using SPICE.

**17.12** What happens, to the simulation results seen in Fig. 17.34, if the time step used in the transient simulation is increased to 1 ns? How do the nonoverlapping clocks look with this time step?

**17.13** Simulate the operation of the circuit seen in Fig. 17.44. Do the comparator outputs make full logic transitions? Are glitches a concern? Why? How do the simulation results compare to the results seen in Fig. 17.37? Note that the outputs of the comparator go low each time $\phi_2$ goes high so that $\overline{Out}$ can be used to clock a counter directly.

IPR2023-00697
Theta EX2019

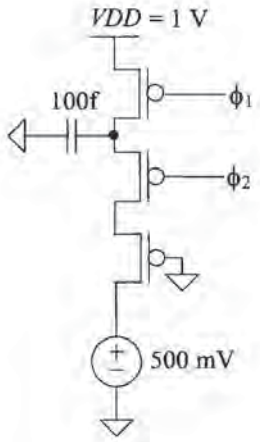



**Figure 17.43** Determining the average current that flows in a switched-capacitor resistor. See Problem 17.11.

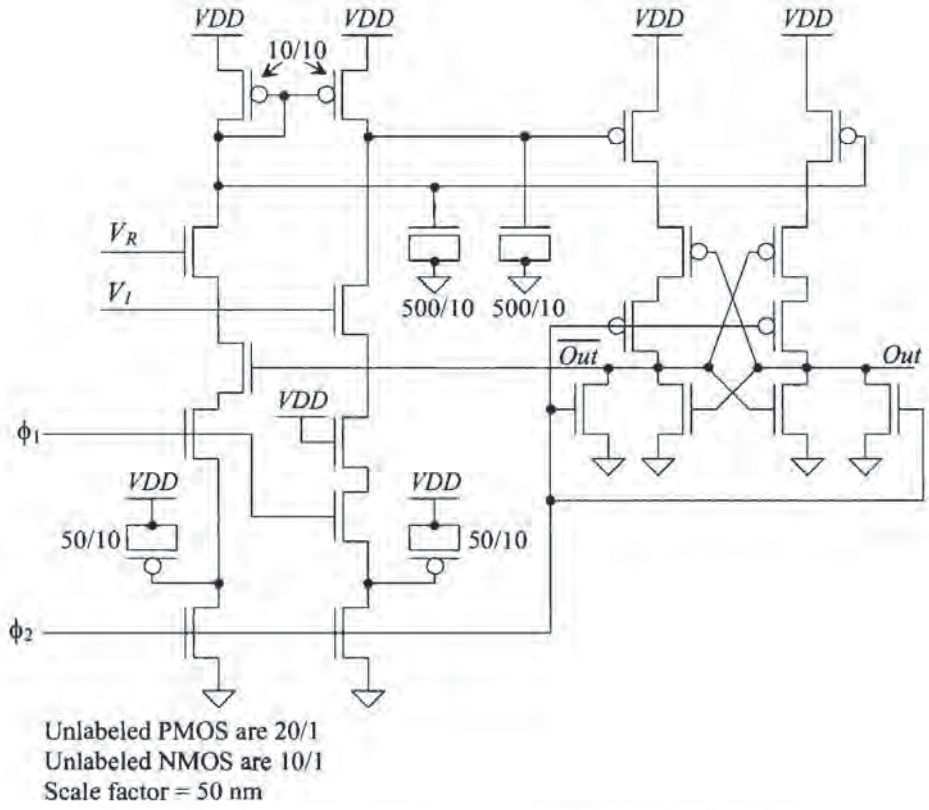



**Figure 17.44** Simplifying the DSM sensing circuit, see Problem 17.13.

IPR2023-00697
Theta EX2019

Chapter
# 18

# Special Purpose CMOS Circuits

In this chapter we discuss some special-purpose CMOS circuits. We begin with the Schmitt trigger, a circuit useful in generating clean pulses from a noisy input signal or in the design of oscillator circuits. Next, we discuss multivibrator circuits, both astable and monostable types. This is followed by a discussion concerning input buffer design. Good receiver circuits (input buffers) in CMOS chips are required in any high-speed, board-level design to change the distorted signals transmitted between chips (because of the imperfections in the interconnecting signal paths) into well-defined digital signals with the correct pulse widths and amplitudes. Finally, we end this chapter with a discussion of on-chip voltage generators.

## 18.1 The Schmitt Trigger

The schematic symbol of the Schmitt trigger is shown in Fig. 18.1 along with typical transfer curves. We should note the similarity to the inverter transfer characteristics with the exception of a steeper transition region (and hysteresis). Curve A in Fig. 18.1 corresponds to the output of the Schmitt trigger changing from a low to a high, while curve B corresponds to the output changing from a high to a low. The hysteresis present in the transfer curves is what sets the Schmitt trigger apart from the basic inverter.

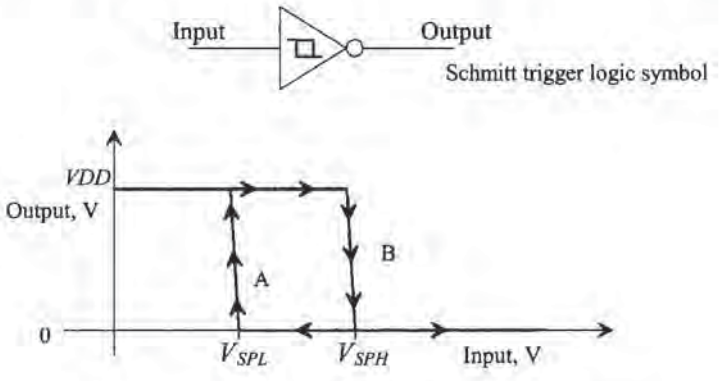



**Figure 18.1** Transfer characteristics of a Schmitt trigger.

IPR2023-00697
Theta EX2019

Figure 18.2 shows a possible input to a Schmitt trigger and the resulting output. When the output is high and the input exceeds $V_{SPH}$, the output switches low. However, the input voltage must go below $V_{SPL}$ before the output can switch high again. Note that we get normal inverter operation when $V_{SPH} = V_{SPL}$. The hysteresis of the Schmitt trigger is defined by

$$V_H = V_{SPH} - V_{SPL} \tag{18.1}$$

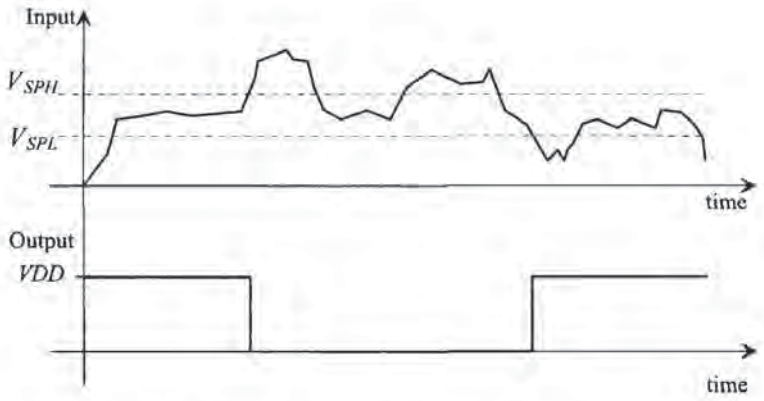



**Figure 18.2** Input, top trace, and output of a Schmitt trigger.

### 18.1.1 Design of the Schmitt Trigger

The basic schematic of the Schmitt trigger is shown in Fig. 18.3. We can divide the circuit into two parts, depending on whether the output is high or low. If the output is low, then M6 is on and M3 is off and we are concerned with the p-channel portion when calculating the switching point voltages, while if the output is high, M3 is on and M6 is off and we are concerned with the n-channel portion. Also, if the output is high, M4 and M5 are on, providing a DC path to *VDD*.

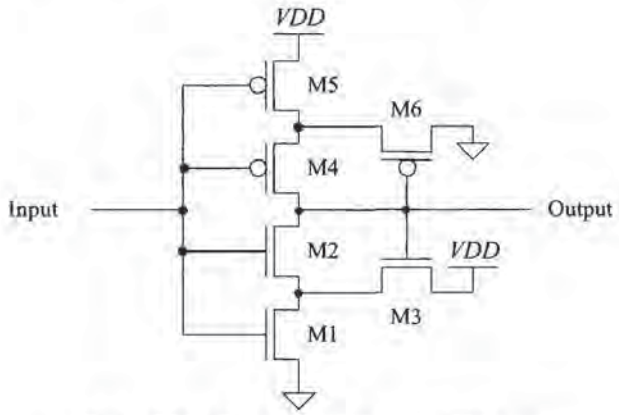



**Figure 18.3** Schematic of the Schmitt trigger.

IPR2023-00697
Theta EX2019

Let's begin our analysis of this circuit, assuming that the output is high (= *VDD*) and the input is low (= 0 V). Figure 18.4 shows the bottom portion of the Schmitt trigger used in calculating the upper switching point voltage, $V_{SPH}$. MOSFETs M1 and M2 are off, with $V_{in}$ = 0 V while M3 is on. The source of M3 floats to $VDD - V_{THN}$, or approximately 4 V for *VDD* = 5 V. We can label this potential $V_x$, as shown in the figure.

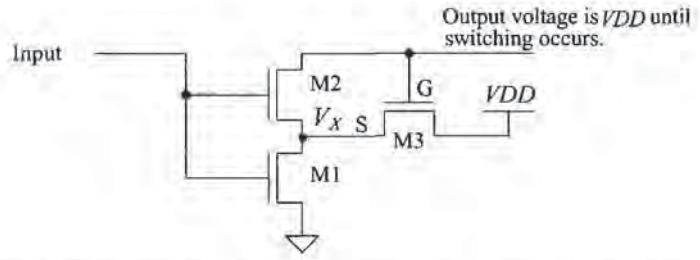



**Figure 18.4** Portion of the Schmitt trigger schematic used to calculate upper switching point voltage.

With $V_{in}$ less than the threshold voltage of M1, $V_X$ remains at $VDD - V_{THN3}$. As $V_{in}$ is increased further, M1 begins to turn on and the voltage, $V_X$, starts to fall toward ground. The high switching point voltage is defined when

$$V_{in} = V_{SPH} = V_{THN2} + V_X \tag{18.2}$$

or when M2 starts to turn on. As M2 starts to turn on, the output starts to move toward ground, causing M3 to start turning off. This in turn causes $V_X$ to fall further, turning M2 on even more. This continues until M3 is totally off and M2 and M1 are on. This positive feedback causes the switching point voltage to be very well defined.

When Eq. (18.2) is valid, the currents flowing in M1 and M3 are essentially the same. Equating these currents gives

$$\frac{\beta_1}{2}(V_{SPH} - V_{THN})^2 = \frac{\beta_3}{2}(VDD - V_X - V_{THN3})^2 \tag{18.3}$$

Since the sources of M2 and M3 are tied together, $V_{THN2} = V_{THN3}$, the increase in the threshold voltages from the body effect is the same for each MOSFET. The combination of Eqs. (18.2) and (18.3) yields

$$\frac{\beta_1}{\beta_3} = \frac{W_1 L_3}{L_1 W_3} = \left[\frac{VDD - V_{SPH}}{V_{SPH} - V_{THN}}\right]^2 \tag{18.4}$$

The threshold voltage of M1, given by $V_{THN}$ in this equation, is the zero body bias threshold voltage (= 0.8 V in our long-channel CMOS process and 0.25 V in the short-channel process). Given a specific upper switching point voltage, the ratio of the MOSFET transconductors is determined by solving this equation. A general design rule for selecting the size of M2, that is, $\beta_2$, is to require that

$$\beta_2 \geq \beta_1 \text{ or } \beta_3 \tag{18.5}$$

since M2 is used as a switch.

IPR2023-00697
Theta EX2019

A similar analysis can be used to determine the lower switching point voltage, $V_{SPL}$, resulting in the following design equation:

$$\frac{\beta_5}{\beta_6} = \frac{W_5 L_6}{L_5 W_6} = \left[\frac{V_{SPL}}{VDD - V_{SPL} - V_{THP}}\right]^2 \tag{18.6}$$

The following example illustrates the design procedure for a Schmitt trigger.

**Example 18.1**

Design and simulate a Schmitt trigger using the short-channel CMOS process with $V_{SPL} = 400$ mV and $V_{SPH} = 700$ mV.

We begin by solving Eqs. (18.4) and (18.6) for the transconductance ratios. For the upper switching point voltage,

$$\frac{W_1 L_3}{W_3 L_1} = \left[\frac{1 - 0.7}{0.7 - 0.25}\right]^2 = 0.444 \rightarrow L_1 = L_3 = 1 \text{ and } W_1 = 10, W_3 = 22.5$$

and for the lower switching point voltage,

$$\frac{W_5 L_6}{W_6 L_5} = \left[\frac{0.4}{1 - 0.4 - 0.25}\right]^2 = 1.3 \rightarrow L_5 = L_6 = 1 \text{ and } W_6 = 20, W_5 = 26$$

M2 is set to 10/1 and M4 is set to 20/1. Simulation results are seen in Fig. 18.5. This figure reveals the benefit of using a Schmitt trigger, namely, it allows slow moving inputs to be made into good solid logic high and low values. In a practical circuit, connecting the ramp-shaped input seen in Fig. 18.5 to an inverter would produce oscillations in the inverter's output (because of noise on the ramp). ■

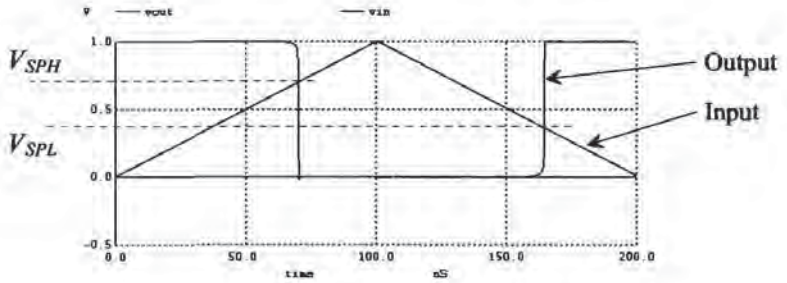



**Figure 18.5** The input and output of the Schmitt trigger designed in Ex. 18.1.

*Switching Characteristics*

The propagation delays of the Schmitt trigger can be calculated in much the same way as the inverter of Ch. 11. Defining equivalent digital resistances for M1, M2, M4, and M5 as $R_{n1}$, $R_{n2}$, $R_{p4}$, and $R_{p5}$, respectively, gives a high-to-low propagation delay-time, neglecting the Schmitt trigger output capacitance, of

$$t_{PHL} = 0.7 \cdot (R_{n1} + R_{n2}) \cdot C_{load} \tag{18.7}$$

and

$$t_{PLH} = 0.7 \cdot (R_{p4} + R_{p5}) \cdot C_{load} \tag{18.8}$$

IPR2023-00697
Theta EX2019

### 18.1.2 Applications of the Schmitt Trigger

Consider the waveform shown in Fig. 18.6. A pulse with ringing is a common voltage waveform encountered in buses or lines interconnecting systems. If this voltage is applied directly to a logic gate or inverter input with a $V_{SP}$ of 0.5 V, the output of the gate will vary with the period of the ringing on top of the pulse. Using a Schmitt trigger with properly designed switching points can eliminate this problem.

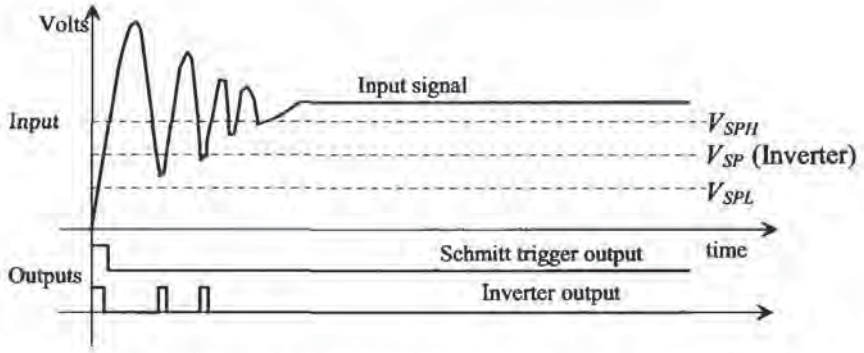



**Figure 18.6** Applying a Schmitt trigger to clean up an interconnecting signal.

The Schmitt trigger can also be used as an oscillator (Fig. 18.7). The delay-time in charging and discharging the capacitor sets the oscillation frequency. At the moment in time when the output of the Schmitt trigger switches low, the voltage across the capacitor is $V_{SPH}$. The capacitor will start to discharge toward ground. The voltage across the capacitor is given by

$$V_c(t) = V_{SPH} \cdot e^{-t/RC} \tag{18.9}$$

At the time when $V_c(t) = V_{SPL}$, the output of the Schmitt trigger changes state. This time is given by solving Eq. (18.9) by

$$t_1 = RC \cdot \ln \frac{V_{SPH}}{V_{SPL}} \tag{18.10}$$

A similar analysis for the case when the capacitor is charged from $V_{SPL}$ to $V_{SPH}$ gives

$$V_c(t) = V_{SPL} + (VDD - V_{SPL})\left(1 - e^{\frac{-t}{RC}}\right) \tag{18.11}$$

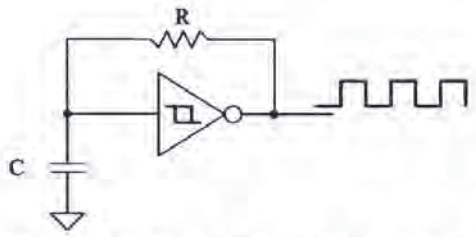



**Figure 18.7** Oscillator design using a Schmitt trigger.

IPR2023-00697
Theta EX2019

and

$$t_2 = RC \cdot \ln \frac{VDD - V_{SPL}}{VDD - V_{SPH}} \tag{18.12}$$

The oscillation frequency, neglecting the intrinsic delay of the Schmitt trigger, is given by

$$f_{osc} = \frac{1}{t_1 + t_2} \tag{18.13}$$

The capacitance used in these equations is the sum of the input capacitance of the Schmitt trigger and any external capacitance.

An alternative oscillator using the Schmitt trigger is shown in Fig. 18.8. Here the MOSFETs M1 and M4 behave as current sources (see Ch. 20) mirroring the current in M5 and M6. When the output of the oscillator is low, M3 is on and M2 is off. This allows the constant current from M4 to charge $C$. When the voltage across $C$ reaches $V_{SPH}$, the output of the Schmitt trigger swings low. This causes the output of the oscillator to go high and allows the constant current from M1 to discharge $C$. When $C$ is discharged down to $V_{SPL}$, the Schmitt trigger changes states. This series of events continues, generating the square wave output.

If we label the drain currents of M1 and M4 as $I_{D1}$ and $I_{D4}$, we can estimate the time it takes the capacitor to charge from $V_{SPL}$ to $V_{SPH}$ as

$$t_1 = C \cdot \frac{V_{SPH} - V_{SPL}}{I_{D4}} \tag{18.14}$$

and the time it takes to charge from $V_{SPH}$ to $V_{SPL}$ is

$$t_2 = C \cdot \frac{V_{SPH} - V_{SPL}}{I_{D1}} \tag{18.15}$$

The period of the oscillation frequency is, as before, the sum of $t_1$ and $t_2$.

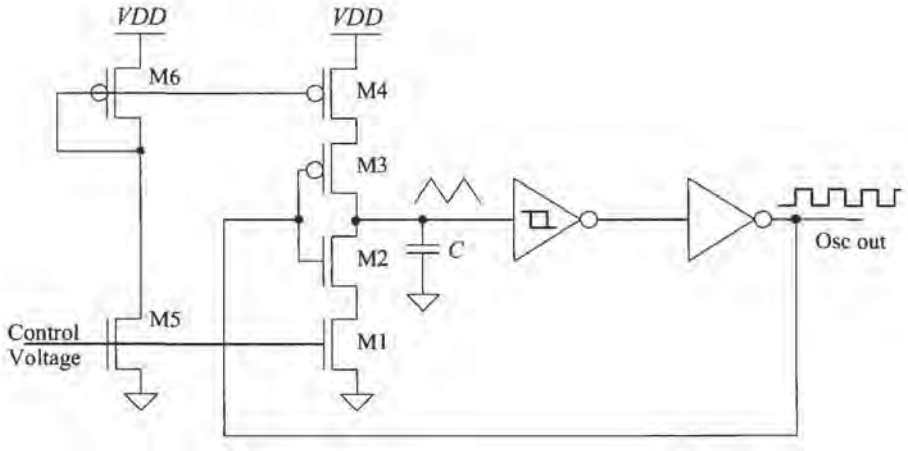



**Figure 18.8** Voltage-controlled oscillator using Schmitt trigger and current sources. MOSFETs M2 and M3 are used as switches.

IPR2023-00697
Theta EX2019

This type of oscillator is termed a voltage-controlled oscillator (VCO) since the output frequency can be controlled by an external voltage. The currents $I_{D1}$ and $I_{D4}$, (Fig. 18.8) are directly controlled by the control voltage. As we will see in Ch. 20, the current in M5 is mirrored in M1, M4, and M6, with an appropriate scaling factor dependent on the size of the transistors.

## 18.2 Multivibrator Circuits

Multivibrator circuits (Fig. 18.9) are circuits that employ positive feedback. The name "multivibrator" is a vestige from early-time electronics development (prior to the ubiquitous term "digital") where the circuits' outputs vibrate between two states. There are three types of multivibrators: astable, bistable, and monostable. Astable multivibrator circuits are unstable in either output (high or low) state. The oscillators that we have discussed are examples of astable multivibrators. The bistable multivibrator is stable in either the high or low state. Flip-flops and latches are examples of the bistable multivibrator. Monostable multivibrators are stable in a single state. Monostable multivibrators are also called one-shots. In this section we discuss the monostable and astable multivibrators. We distinguish the material in this section from the other material in the book by using resistor-capacitor time constants to set time intervals.

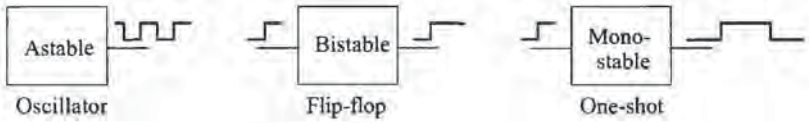



**Figure 18.9** Multivibrator circuits.

### 18.2.1 The Monostable Multivibrator

A CMOS implementation of the monostable multivibrator is shown in Fig. 18.10. Under normal conditions, $V_{in}$ is low and the output of the NOR gate, $V_1$, is high. The voltage $V_2$ is pulled high through the resistor, and the output of the inverter, $V_3$, is a low. Upon application of a trigger pulse, that is, $V_{in}$ going high, both $V_1$ and $V_2$ drop to zero volts and the output of the inverter, which is also the output of the monostable, goes high. This output is fed back to the input of the NOR gate holding $V_1$ at ground potential.

After triggering, the potential $V_2$ will start to increase because $C$ is charged through $R$. The potential across the capacitor after triggering takes place is given by

$$V_c(t) = V_2(t) - \overbrace{V_1(t)}^{=0} = VDD \cdot (1 - e^{\frac{-t}{RC}}) \qquad (18.16)$$

If we assume that the $V_{SP}$ of the inverter is $VDD/2$, then the time it takes for the capacitor to charge to $V_{SP}$ is given by

$$t = RC \cdot \ln \frac{VDD}{VDD - V_{SP}} = RC \cdot \ln(2) \approx 0.7RC \qquad (18.17)$$

IPR2023-00697
Theta EX2019

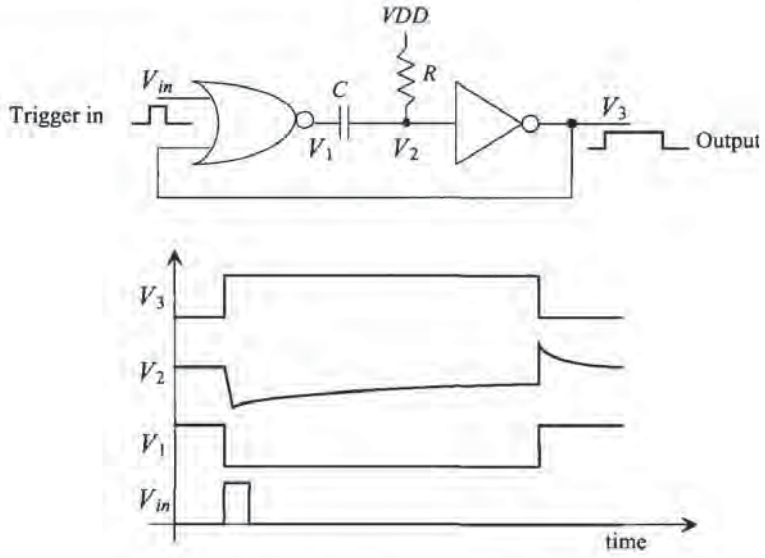



**Figure 18.10** Operation of the monostable multivibrator.

This time also defines the output pulse width, neglecting gate delays, since the inverter switches low. The inverter output, $V_3$, going low causes $V_1$ to go back to *VDD* and $V_2$ to go to *VDD* + *VDD*/2. If the resistor or capacitor is bonded out (connected to the output pads), the ESD diodes may keep $V_2$ from going much above *VDD* + 0.7. The time it takes $V_2$ to decay back down to *VDD* limits the rate at which the one-shot can be retriggered. Also note that the trigger input can be longer than the output pulse width. Longer output pulse widths may cause $V_2$ to go as high as 10 V as well as limit the maximum trigger rate.

### 18.2.2 The Astable Multivibrator

An example of an astable multivibrator is shown in Fig. 18.11. This circuit has no stable state and thus oscillates. To analyze the behavior of this multivibrator, let's begin by assuming that the output, $V_3$, has just switched high. The output going high causes $V_1$ to go high (to $VDD + V_{SP1}$), forcing $V_2$ low. The voltage across the capacitor after this switching takes place is given by

$$V_c(t) = V_1(t) - \overbrace{V_3(t)}^{=VDD} = (VDD + V_{SP1}) \cdot e^{\frac{-t}{RC}} - VDD \qquad (18.18)$$

The output of the astable will go low, $V_3 = 0$, when $V_1 = V_{SP1}$. Substituting this condition into the previous equation gives the time the output is high (or low) and is given by

$$t_1 = RC \cdot \ln \frac{VDD + V_{SP1}}{V_{SP1}} = t_2 \qquad (18.19)$$

If $V_{SP1} = VDD/2$, then

$$t_1 = t_2 = 1.1RC \qquad (18.20)$$

IPR2023-00697
Theta EX2019

and the frequency of oscillation is

$$f_{osc} = \frac{1}{t_1 + t_2} = \frac{1}{2.2RC} \tag{18.21}$$

Again, if the resistor and capacitor are bonded out, the ESD diodes on the pads will limit the voltage swing of $V_1$.

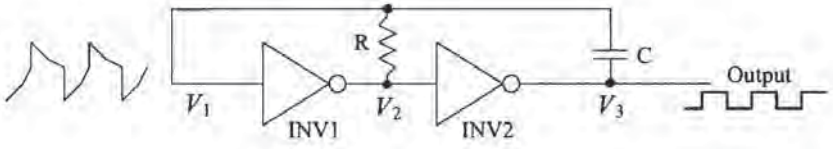



**Figure 18.11** An astable multivibrator.

## 18.3 Input Buffers

Input buffers are circuits that take a chip's input signal, with imperfections such as slow rise and fall times, and convert it into a clean digital signal for use on-chip. If the buffer doesn't "slice" the data in the correct position, timing errors can occur. For example, consider the waveforms seen in Fig. 18.12. If the input signal is sliced too high or too low, the output signal's width is incorrect. In high-speed systems this reduces the timing budget in the system and can result in errors.

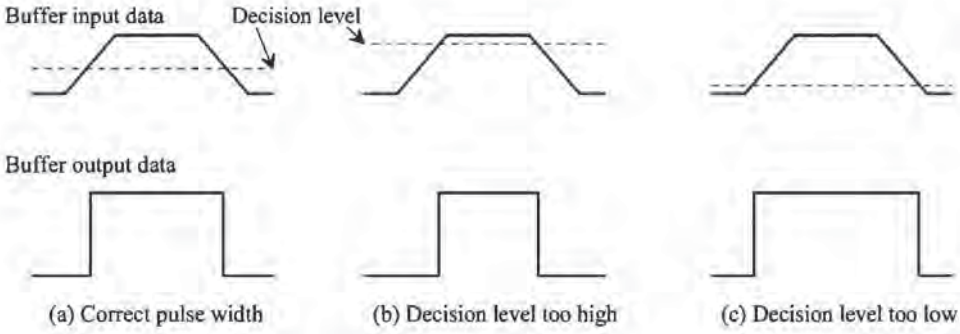



**Figure 18.12** Timing errors in regenerating digital data.

### 18.3.1 Basic Circuits

If not careful, simple circuits can introduce unwanted pulse width variations because of differences in rise and fall times. Consider the inverter circuit seen in Fig. 18.13. The switching point voltage of the inverter, because of process, *VDD* (voltage), or temperature variations (PVT) will move around. Further, differences in the PMOS and the NMOS resistance will affect the rise and fall times of the inverter's output signal. The output of the first inverter, in Fig. 18.13, sees a much larger load capacitance than the second inverter. The delay time from the input going high to the output going high is roughly

IPR2023-00697
Theta EX2019

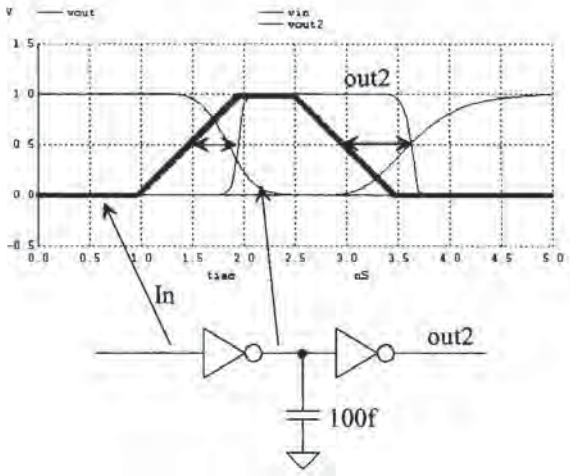



**Figure 18.13** How skew is introduced into a high-speed signal.

400-ps. However, the delay time between the input and the output going low is 600 ps. As the input signal travels through the digital system, the time skew added to the input signal can add up and result in timing errors (the data is received too early or too late at different points in the system). Note that had we matched the loading of each inverter, the propagation delays would be more similar, Fig. 18.14. When the input signal goes high, the output of the first inverter goes low ($t_{PHL}$) and the output of the second inverter goes high ($t_{PLH}$). The total delay is the sum of $t_{PHL} + t_{PLH}$. However, when the input signal goes low, the output of the first inverter goes high ($t_{PLH}$) and the output of the second inverter goes low ($t_{PHL}$), resulting in the same sum ($t_{PHL} + t_{PLH}$). *If possible, make the number of $t_{PHL}$ delays in series with a high-speed signal even and equal to the number of $t_{PLH}$ delays.*

Of course, to better equalize the delays, the second inverter in Fig. 18.14 would need an additional bit of loading (an inverter connected to its output). Also, the input signal should transition faster in an attempt to match the transition times of the inverters.

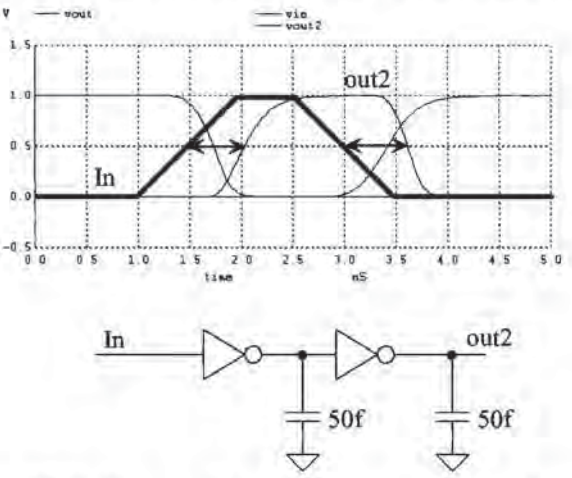



**Figure 18.14** How equalizing delays can be used to reduce skew.

IPR2023-00697
Theta EX2019

*Skew in Logic Gates*

Consider the NAND gate in Fig. 18.15. A different delay will be added to an input signal depending on which input of the NAND gate is used. As seen in the figure, the *A* input, with the *B* input high, propagates to the output slightly quicker than the *B* input (with the *A* input high). For the NAND gate, this difference in propagation delays is only a factor for the NMOS devices (because they are in series with the output of the gate).

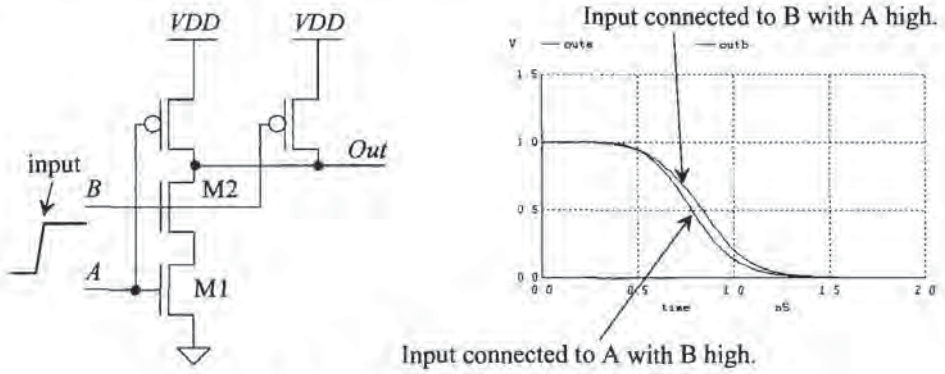



**Figure 18.15** The skew introduced by using different inputs.

Figure 18.16 shows how two NAND gates can be used in parallel to eliminate the differences in the propagation delay between inputs. The series NMOS are arranged so that no matter which input is toggled the output changes at the same rate. Note that two of the PMOS devices can be eliminated to simplify the circuit without affecting the propagation delays.

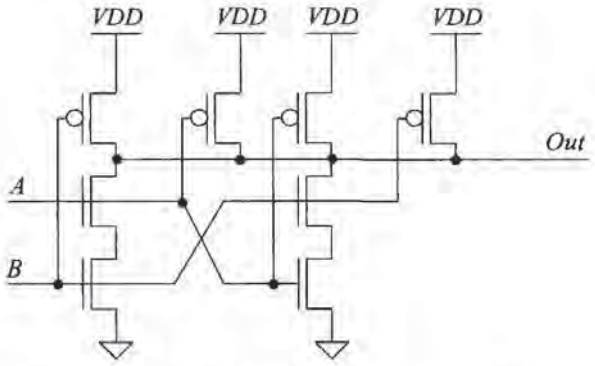



**Figure 18.16** Using two NAND gates in parallel to reduce input-dependent skew.

The previous discussion neglected the effects of the gate's switching point voltage. If the input signals are not transitioning to full logic levels (*VDD* and ground) or the rise and fall times are not fast (compared to the gate delay time), then the point where the input circuit switches is critical. This leads us to our next topic.

IPR2023-00697
Theta EX2019

### 18.3.2 Differential Circuits

In order to precisely slice input data, as seen in Fig. 18.12, a reference voltage may be transmitted, on a different signal path, along with the data. Alternatively, the data may be transmitted differentially (an input and its complement). In either case a differential amplifier is needed, Fig. 18.17. A differential amplifier input buffer (which we'll simply call an input buffer from this point on) amplifies the difference between the two inputs. In the simplest case one input to the input buffer is a DC voltage, say 0.5 V (*Vinm* in Fig. 18.17). When the other input (*Vinp*) goes above 0.5 V, the output of the buffer changes states (goes from a low to a high) or

$$Vinp > Vinm \rightarrow \text{out} = \text{"1"} \tag{18.22}$$

and

$$Vinp < Vinm \rightarrow \text{out} = \text{"0"} \tag{18.23}$$

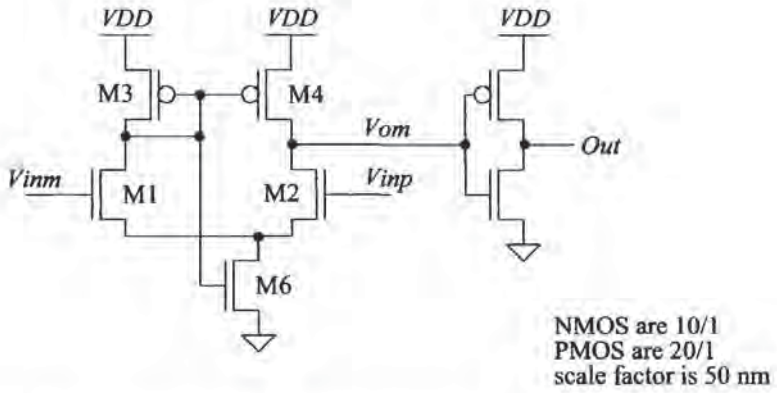



**Figure 18.17** An (n-flavor) input buffer for high-speed digital design.

The diff-amp in Fig. 18.17 is based on the topologies seen in Ch. 22. This circuit is *self-biased* because no external references are used to set the current in the circuit (the gate of M6 is tied up to the gate of M3). When *Vinp* is larger than *Vinm*, the current in M2 is larger than the current in M1 ($V_{GS2} > V_{GS1}$). The current in M1 flows through M3 and is mirrored by M4 (and so M4's current is less than M2's current). This causes the diff-amp's output, *Vom*, to go towards ground (until the current in M2 equals the current in M4) and the output of the inverter, *Out*, to go high. Note that the gain from the *Vinp* input to the output of the circuit is larger than the gain from *Vinm* to the output (M3, being diode-connected, is a lower resistance than M4). It is generally a good idea to connect the reference voltage to the *Vinm* input.

The DC simulation results for the buffer in Fig. 18.17 are shown in Fig. 18.18. The x-axis is the *Vinp* input swept from 0 to 1 V. The *Vinm* input is held from 0 to 1 V in 200 mV steps. When *Vinm* is held at 400 mV and *Vinp* goes above 400 mV, the output changes from 0 to 1 (although there is a small offset which necessitates that *Vinp* go to approximately 415 mV before the output transitions).

IPR2023-00697
Theta EX2019

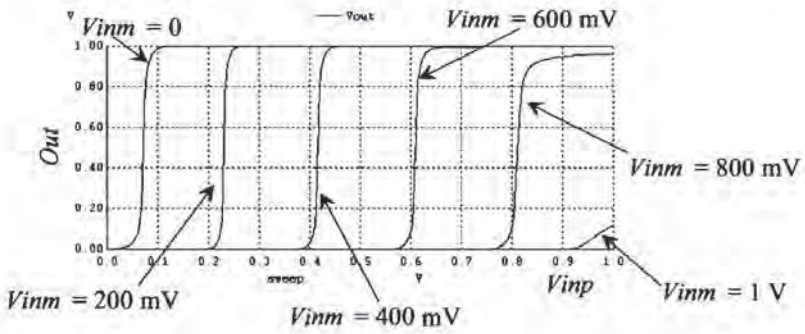



**Figure 18.18** Simulating the DC behavior of the buffer in Fig. 18.17.

*Transient Response*

An example transient response for the buffer in Fig. 18.17 is seen in Fig. 18.19. A very small increase in *Vinp* above *Vinm* is required to make the output of the buffer switch states. We have several practical questions that should be answered with variations of this simulation. For example, what are the minimum and maximum values of *Vinm* allowed. Next, why are the delays between *Vinp* going high and going low different? Is it because of the offset seen in Fig. 18.18 (the output doesn't precisely switch at a *Vinm* 400 mV)? Let's provide some discussion concerning these concerns.

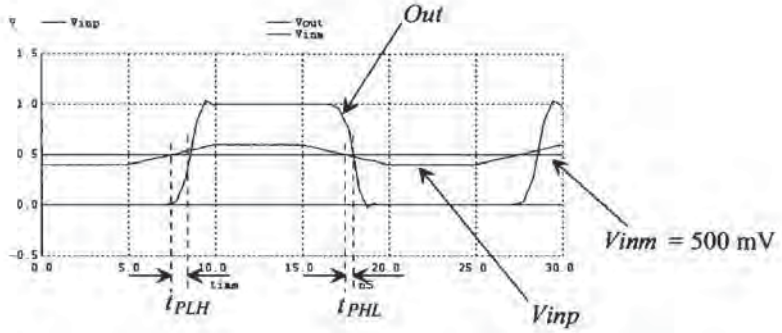



**Figure 18.19** Transient response of the input buffer in Fig. 18.17 driving a 50 fF load.

Looking at Fig. 18.17, we can see that if the inputs fall below $V_{THN}$ (= 250 mV here), then the circuit won't work very quickly (the MOSFETs move into the subthreshold region). So we would expect the propagation delays to increase. Indeed, as seen in Fig. 18.20, the delays go up. Ideally, the delay of the buffer is independent of power supply voltage, temperature, or input signal amplitudes (or pulse shape). To get better performance for lower input level signals, we might use the PMOS version of the buffer in Fig. 18.17, as seen in Fig. 18.21. Resimulating this buffer with the signals seen in Fig. 18.20 gives the results seen in Fig. 18.22. The delays are considerably better, however, there is an offset that appears rather large (because the output changes at the same time as *Vinp* going past *Vinm*, indicating that an offset is present). To avoid this

IPR2023-00697
Theta EX2019

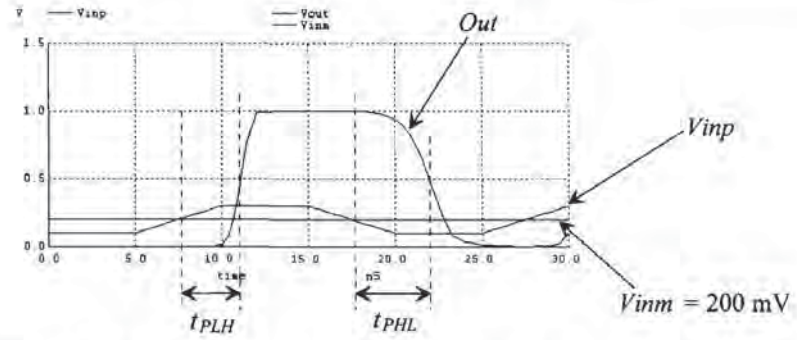



**Figure 18.20** Resimulating with lower input signal voltages.

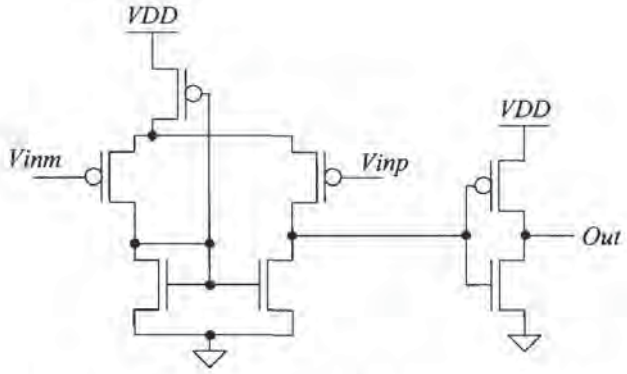



**Figure 18.21** A PMOS input buffer for high-speed digital design.

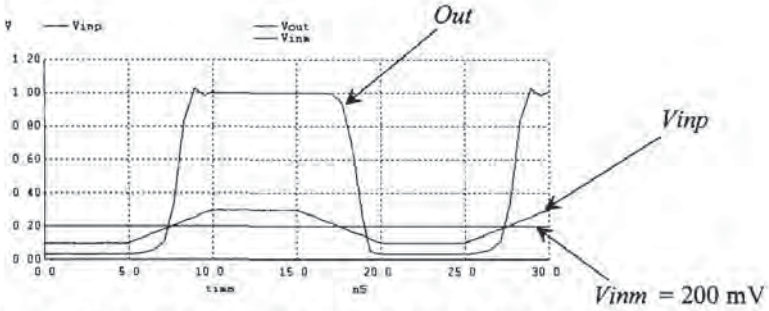



**Figure 18.22** Repeating the simulation in Fig. 18.20 with the buffer in Fig. 18.21.

offset, we might use the NMOS buffer in Fig. 18.17 with the PMOS buffer in Fig. 18.21 to form a buffer that operates well with input signals approaching ground or *VDD*. The result is seen in Fig. 18.23. By using the buffers in parallel, the complementary nature results in a buffer that is robust and works over a wide range of operating voltages. Figure 18.24 shows a DC sweep simulation for the buffer with the *Vinp* input swept and the *Vinm* changed in increments of 100 mV. Note the smaller offset.

IPR2023-00697
Theta EX2019

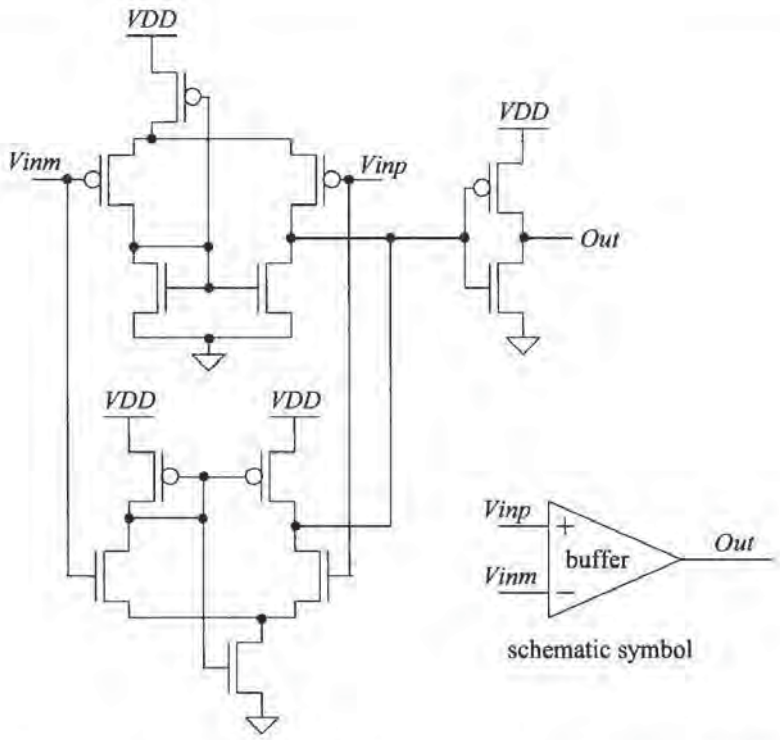



**Figure 18.23** A rail-to-rail input buffer based on the topologies in Figs. 18.17 and 18.21.

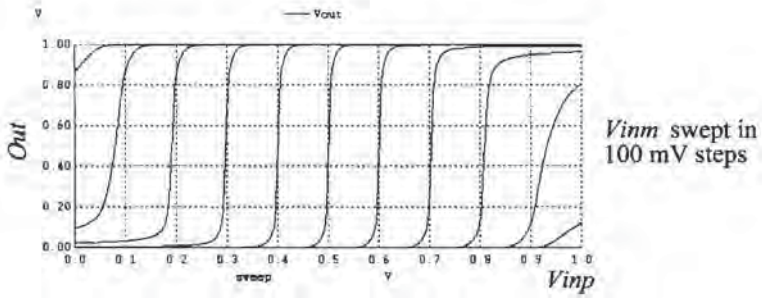



**Figure 18.24** DC characterisitics of the buffer in Fig. 18.23.

**Example 18.2**

Design a high-speed input buffer based on the topology in Fig. 18.17 with a small amount of hysteresis.

In any circuit that has hysteresis we have some positive feedback. We can introduce positive feedback into the buffer in Fig. 18.17 by adding a long L MOSFET across the output inverter, Fig. 18.25. When the output is low, this turns the long L MOSFET on and make the self-biased diff-amp work harder to pull the input of the inverter to ground. When the output is high, the long L MOSFET is off and doesn't affect the circuit. ■

IPR2023-00697
Theta EX2019

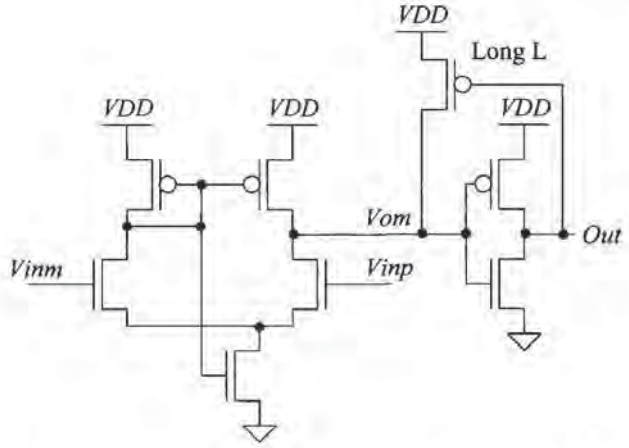



**Figure 18.25** An input buffer for high-speed digital design with hysteresis.

### 18.3.3 DC Reference

In the previous section we assumed we had a DC voltage, *Vinm* , at precisely the correct voltage to slice the data (in the middle). In a real system the data varies and the communication channel is bandlimited (behaves like a lowpass filter). The result is that the amplitudes of the data vary depending on the data and the channel frequency response. Consider the simple RC lowpass filter and data used to model a transmission system as seen in Fig. 18.26. The ideal point where we slice the output data changes with the input data. What we need is a circuit that determines the maximum and minimum of an input waveform and outputs the average of the two to slice the buffer's input data in the center.

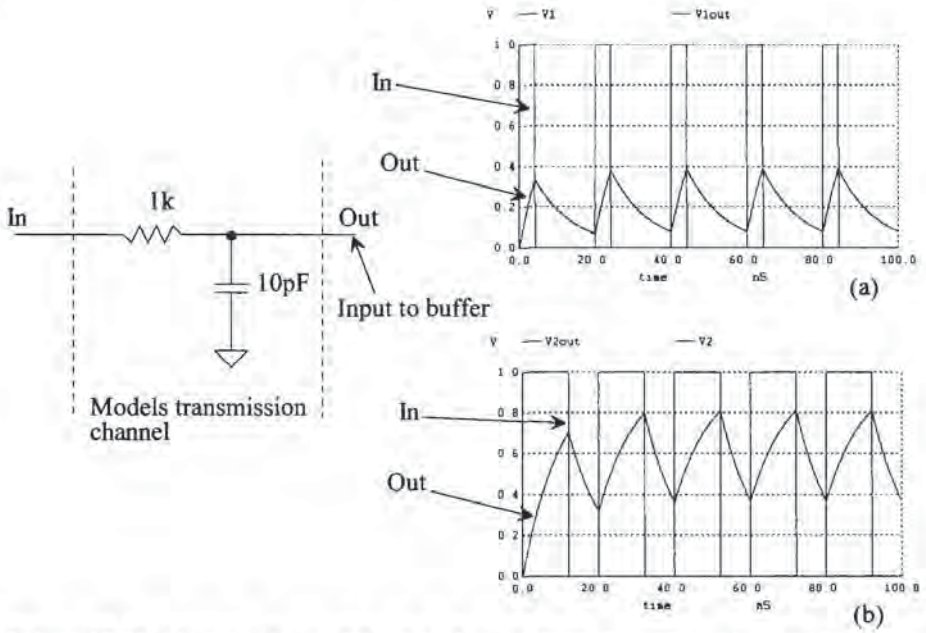



**Figure 18.26** How the shape of the data changes through a communication channel.

IPR2023-00697
Theta EX2019

Towards this goal consider the peak detector circuit seen in Fig. 18.27. The buffer from Fig. 18.23 can be used in this circuit. When the input, $v_{in}$, goes above the voltage stored on the capacitor, $v_{peak}$, the output of the buffer goes low, turning on the long-length MOSFET and pulling the output towards *VDD*. As $v_{peak}$ approaches $v_{in}$, the MOSFET starts to shut off. As a result the output voltage across the capacitor, $v_{peak}$, corresponds to the peak voltage of the input signal. This peak detector can be used to generate a reference voltage that falls within the middle of the input data, Fig. 18.28. The peak and valley detectors are used to find the minimum and maximum of the input signal. The two resistors average the minimum voltages and feed the result (the DC average of the input) to the bottom buffer circuit. The resistors also are used to leak charge off of the capacitors so that the averaging circuit can follow changes in the input data (the averaging is actually a *running average*). Figure 18.29 shows some simulation results. In the top plots the output signal from Fig. 18.26a is applied to the input of the DC generation circuit. As expected, the output of the DC generation circuit, after a start-up delay, goes right to the middle of the input data.

The speed (response time) of this generation circuit can be increased by reducing the resistors and capacitors in the circuit. However, if the values are reduced too much, a long string of zeroes or ones will cause the DC generation circuit's output (labeled "average" in Fig. 18.28) to go to ground or *VDD*. The bottom traces in Fig. 18.29 show the results of applying the output signal from Fig. 18.26b to the DC generation circuit. Again, the output moves quickly to the center of the data. What's more interesting in these figures is a comparison between the original input data and the regenerated output data (labeled *In* and *Out* in Fig. 18.29). The output data pulse widths are considerably different from the input pulse widths (remembering that there is a delay through both the channel RC seen in Fig. 18.26 and the buffer in Fig. 18.28). Ideally, the pulse widths (the data) of the input and output are the same. Looking at the responses in Fig. 18.26a, we see that the effect of the finite channel bandwidth is to distort the channel's output data. *The only solution to this problem is to make the channel transmission bandwidth effectively wider.* An equalizer (discussed in the next chapter) can be used for this purpose at the cost of reduced signal swing. We can also reduce the input impedance of the input buffer to lower the time constant associated with the channel. Let's discuss this second approach.

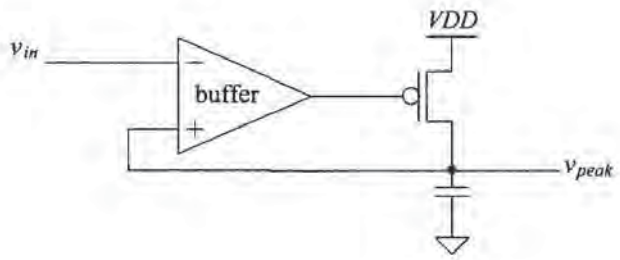



**Figure 18.27** CMOS peak detector.

IPR2023-00697
Theta EX2019

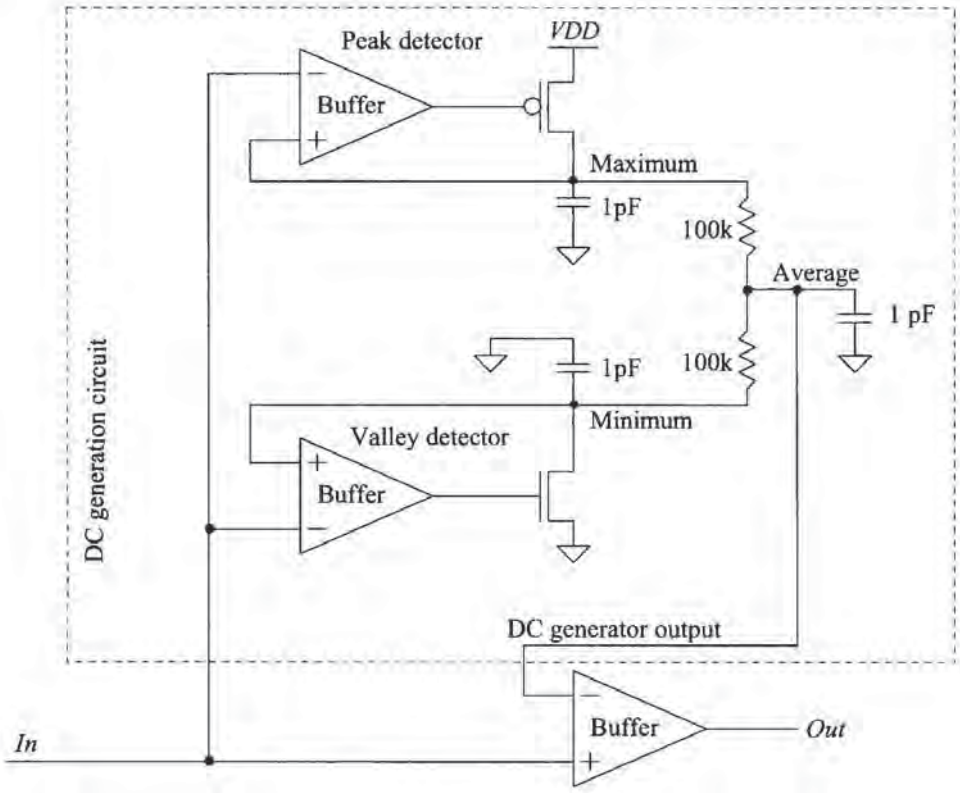



**Figure 18.28** A DC generation circuit.

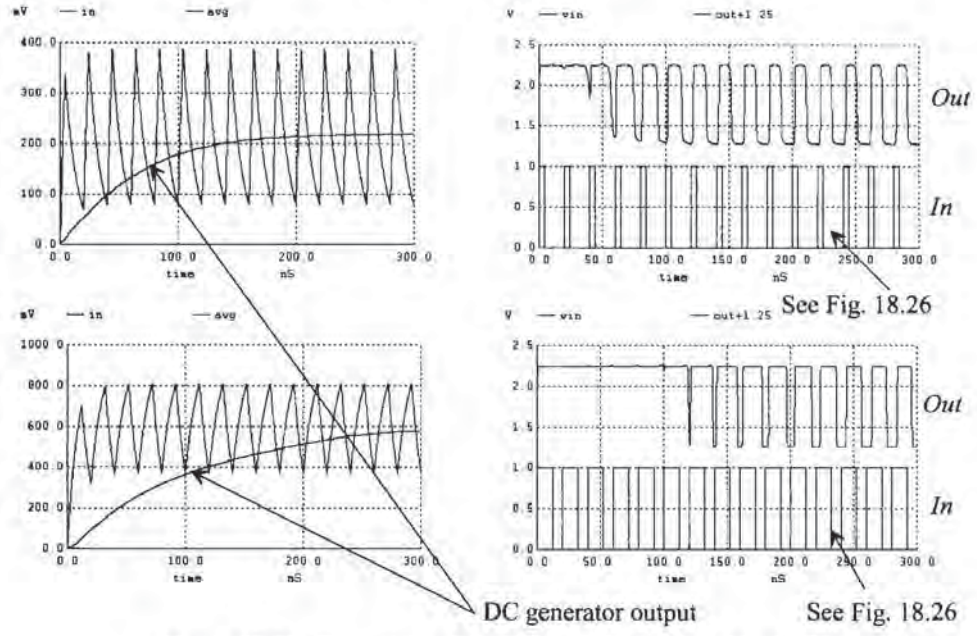



**Figure 18.29** The output of the DC restore circuit in Fig. 18.28.

IPR2023-00697
Theta EX2019

### 18.3.4 Reducing Buffer Input Resistance

A simple solution to increasing the effective bandwidth of the transmission channel is to reduce the input resistance of the input buffer. The time constant associated with the channel in Fig. 18.30 is 10 ns. When we connect the buffer to the transmission channel, the time constant drops to 3.33 ns (the three 1k resistors in parallel multiplied by the 10 pF capacitance). The drawbacks of this approach are increased power dissipation and reduced input signal amplitudes. Figure 18.31 shows how the buffer in Fig. 18.30 behaves with the input signals seen in Fig. 18.26a. The distortion (difference in the pulse widths of ones and zeroes when comparing the communication channel input, *In* , and the buffer output, *Out*) is much better.

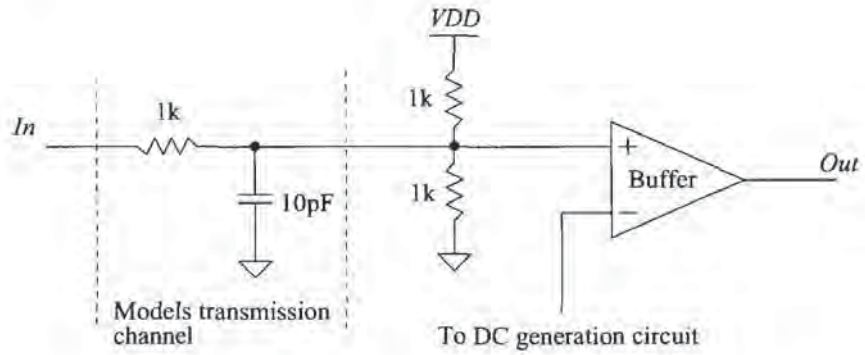



**Figure 18.30** Reducing the input resistance of the input buffer.

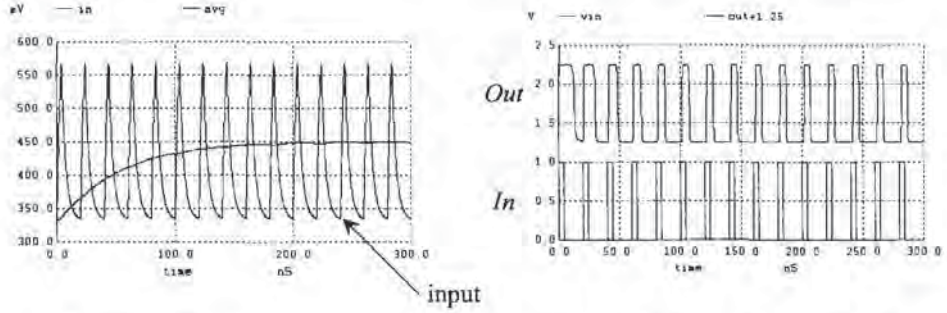



**Figure 18.31** How reducing the input resistance of the buffer reduces the input signal amplitude (bad) and distortion (good).

Figure 18.32 shows an alternative low input resistance input buffer. When the input goes high, M1 turns on and M4 shuts off. This causes M2 to turn on and M5 to shut off. The output then goes high. When the input goes low, M1 turns off and M4 turns on (with M2 shutting off and M5 turning on). The result: the output goes low. Again, the power burned by M1 and M4 can be high if the transistors aren't sized properly. Further, the input swing is limited to a threshold voltage away from the supply voltages. This reduces the input voltage swing and further enhances the speed.

IPR2023-00697
Theta EX2019

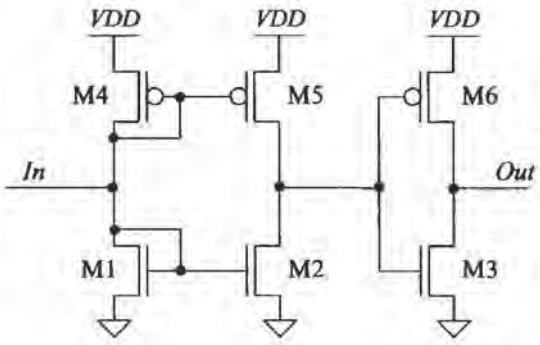



**Figure 18.32** A low-input resistance input buffer.

## 18.4 Charge Pumps (Voltage Generators)

Often, when designing CMOS circuits, positive and negative DC voltages are needed that do not lie between ground and *VDD*. An example of an application where a larger DC voltage source is needed was seen in Ch. 16 when we discussed Flash memory (see Fig. 16.59). A simple circuit, sometimes called a voltage pump (or more often, a *charge pump*), useful in generating a voltage greater than *VDD*, is seen in Fig. 18.33.

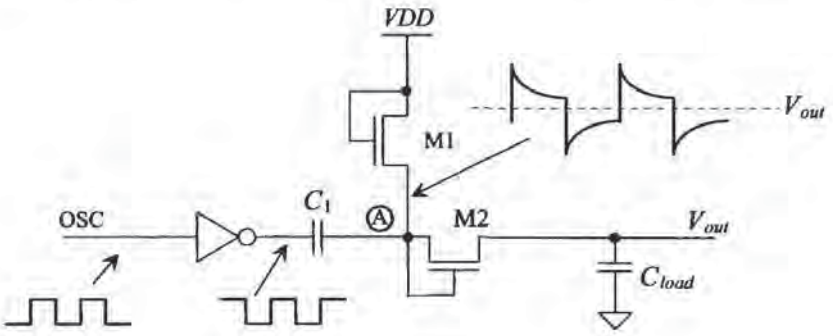



**Figure 18.33** Pump used to generate a voltage greater than *VDD*.

The operation of the voltage pump of Fig. 18.33 can be explained by first realizing that both M1 and M2 operate like a diode. M1 is simply used to pull point A to a voltage of $VDD - V_{THN}$. M2 allows the charge from $C_1$ to charge $C_{load}$ but not vice-versa. Let's begin the description of the circuit operation by assuming that the output of the inverter is low and point A is at a potential of $VDD - V_{THN}$. When the output of the inverter goes high, the potential at point A increases to $VDD + (VDD - V_{THN}) = 2 \cdot VDD - V_{THN}$. This turns on M2 and charges $C_{load}$ to $2 \cdot (VDD - V_{THN})$, [an extra $V_{THN}$ because of M2's gate-source voltage drop], provided $C_1 >> C_{load}$ and the oscillator frequency allows the capacitors to fully charge or discharge before changing states. In most practical situations, $C_{load}$ and $C_1$ are comparable in size, and the output of the pump is loaded with a

IPR2023-00697
Theta EX2019

DC load. The result is an output voltage with a startup time; that is, $V_{out}$ does not immediately rise to $2\cdot(VDD - V_{THN})$ but requires several oscillator cycles to reach steady state. Also, the output has a ripple dependent on the DC load. Figure 18.34 shows the simulation results for the circuit of Fig. 18.13 using 10/1 NMOS (scale factor of 50 nm) with $C_{load} = C_1 = 1$ pF and an oscillator frequency of 10 MHz. Using this simple pump with a DC load, such as a resistor, may require employing larger MOSFETs and capacitors to avoid excessive voltage droop.

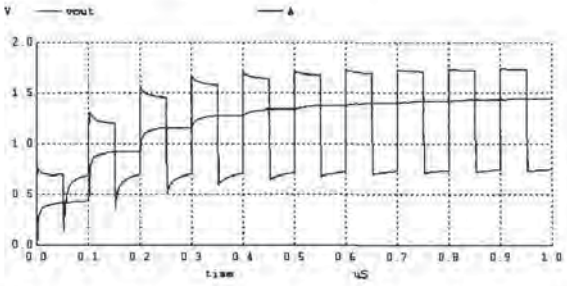



**Figure 18.34** Simulating the operation of the voltage pump in Fig. 18.33.

*Negative Voltages*

The positive voltage pump uses n-channel MOSFETs, while the negative voltage pump, Fig. 18.35, uses p-channel MOSFETs. The reason for this comes from the requirement that the diode formed with the n+ (p+) implant used in the drain/source of the MOSFET combined with the p-substrate (n-well) does not become forward-biased. Forward biasing this parasitic diode is an *important concern* and is often the reason why some more exotic pump topologies can't be implemented. For example, in Fig. 18.35, we connect the well of the PMOS devices to ground instead of to their respective sources. Consider what would have happened had we connected the well to the source of the MOSFET. When the output voltage goes negative, the (n-type) well goes negative too. If the substrate (p-type) is at ground, this forward biases the n-well to substrate diode acting to clamp the output of the pump at a negative diode drop. While we could have left the bodies of the PMOS (the n-wells) tied to *VDD*, here we chose to connect them to ground so that the body effect wasn't so severe (a lower threshold voltage).

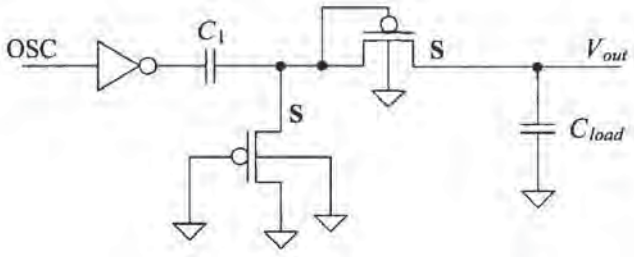



**Figure 18.35** Negative voltage pump.

IPR2023-00697
Theta EX2019

*Using MOSFETs for the Capacitors*

The capacitors used in Figs. 18.33 and 18.35 can be replaced with MOSFETs. An example of using n-channel MOSFETs in place of the capacitors in Fig. 18.33 is shown in Fig. 18.36. The main requirement on a MOSFET used as a capacitor is that its $V_{GS}$ remain greater than $V_{THN}$ at all possible operating conditions. In other words, the MOSFET must remain in the strong inversion region so that its capacitance is a constant $C'_{ox} \cdot W \cdot L$. The capacitor, $C_{load}$, in Fig. 18.36 remains in strong inversion because $V_{out} >> V_{THN}$. The capacitor $C_1$ remains in strong inversion because, when the inverter output is low, the voltage on the gate of $C_1$ is $VDD - V_{THN}$ ($= V_{GS}$). When the output of the inverter is high ($VDD$), the voltage on the gate of $C_1$ is $2 \cdot VDD - V_{THN}$. In both cases (the output of the inverter high and low), $V_{GS} = VDD - V_{THN}$, and the MOSFET is in the strong inversion region.

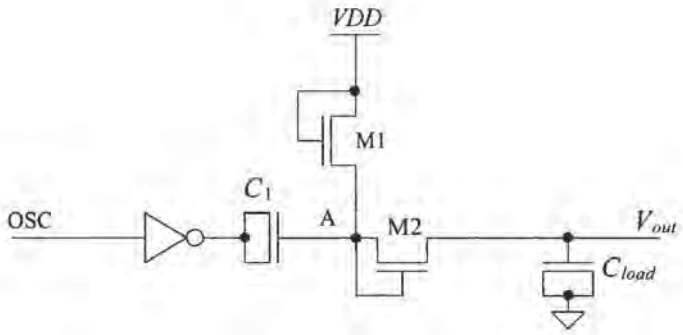



**Figure 18.36** Using MOSFETs as capacitors.

### 18.4.1 Increasing the Output Voltage

Figure 18.37 shows a voltage pump with a higher output voltage. The increase in the output voltage comes from eliminating the threshold voltage drop at the gate and drain of M7. This allows the output to swing up to $2 \cdot VDD - V_{THN}$.

To understand the operation of the circuit, let's assume that the voltage at point A is low and the voltage at point C is $VDD - V_{THN}$. When the output of INV1 goes high, point A is $VDD$ and the voltage at point C swings up to $2 \cdot VDD - V_{THN}$. This causes M4, M5, and M6 to turn on and pull points D and E to $VDD$. Now when point B goes high (point A goes back to zero), points D and E swing up to $2 \cdot VDD$ and the output goes to $2 \cdot VDD - V_{THN}$. Note that MOSFETs M2 and M3 are not needed; unless the pump drives a DC load , they never turn on. Also, separating points D and E is unnecessary unless the pump supplies a DC current.

### 18.4.2 Generating Higher Voltages: The Dickson Charge Pump

The voltage pumps of the last section are limited to voltages less than $2 \cdot VDD$ and greater than $-2 \cdot VDD$. Figure 18.38 shows a scheme for generating arbitrarily high voltages (limited by the breakdown voltage of the capacitors or the oxide breakdown of the MOSFETs). Again, the MOSFETs are used as diodes in this configuration.

IPR2023-00697
Theta EX2019

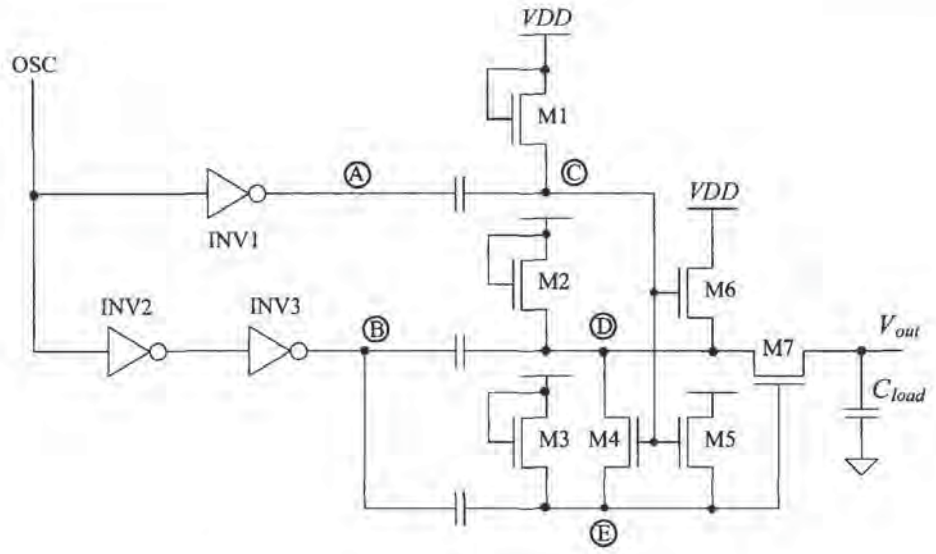



**Figure 18.37** Increased output voltage pump.

In the following description of circuit operation, we assume steady-state operation with no DC load present. When CLK is low, point A is pulled, by M1, to $VDD - V_{THN}$. When CLK goes high, point A swings up to $2 \cdot VDD - V_{THN}$. Diode M2 turns on, and point B charges to $2 \cdot VDD - 2 \cdot V_{THN}$. When CLK goes low, $\overline{\text{CLK}}$ goes high and point B swings up to $3 \cdot VDD - 2 \cdot V_{THN}$. When $\overline{\text{CLK}}$ goes low, point B swings back down to $2 \cdot VDD - 2 \cdot V_{THN}$. This operation proceeds through the circuit to the last stage of the multiplier. The output of the multiplier swings from $(N+1) \cdot VDD - N \cdot V_{THN}$ down to $N \cdot VDD - N \cdot V_{THN}$. $C_N$ can be made larger than the other capacitors to reduce this ripple.

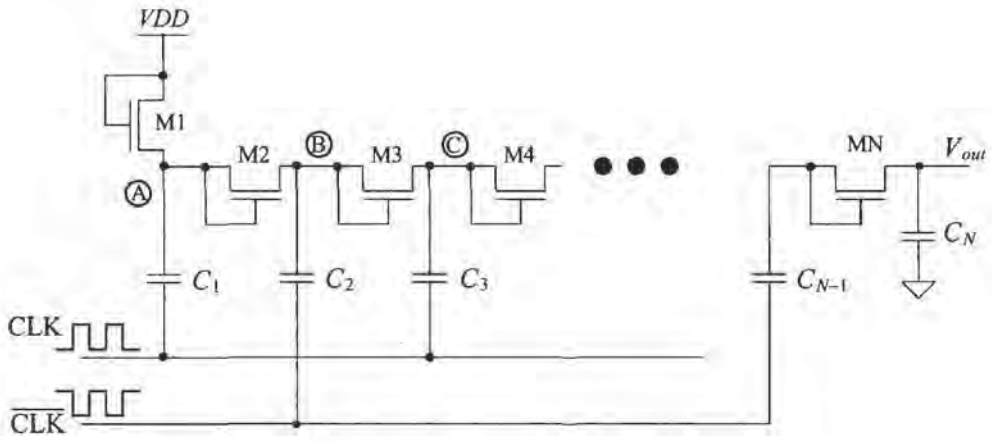



**Figure 18.38** N-stage voltage multiplier (the Dickson charge pump).

IPR2023-00697
Theta EX2019

*Clock Driver with a Pumped Output Voltage*

To fully turn on NMOS switches, a driver configuration is needed with a pulsed output voltage greater than $VDD + V_{THN}$ (with body effect). One such configuration is seen in Fig. 18.39. When the input to the circuit is a logic low, the output of INV1 is a high and the output of INV2 is a low. Node B in the figure is at roughly *VDD*, and node A is at roughly 2*VDD*. M1 is off and M2 is on. The output of the driver is a logic low (ground). When the input of the circuit goes high, the output of INV1 goes low. This causes node A to go to *VDD*. The output of INV2 goes high, and node B boots up to 2*VDD* (M1 on M2 off). The output of the driver circuit then goes to 2*VDD* as well. Notice that one side of the cross-coupled NMOS devices is sized smaller than the other side. This is to minimize layout area and power (the charging and discharging of the parasitic capacitors doesn't waste power). Node A doesn't supply any power to the output of the circuit. It is simply used to turn M2 on so that node B is precharged to *VDD* when the input to the circuit is low. A larger capacitor is used on node B because the charge that goes to load must be supplied by this capacitor. If the output load capacitance gets large, then the size of this capacitor must be increased. When designing the driver, it's a good idea to characterize the performance as a function of the load capacitance.

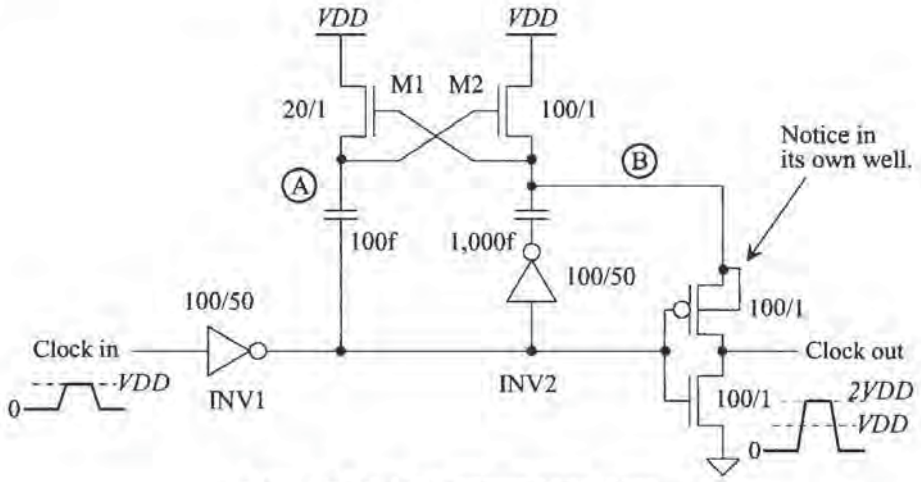



**Figure 18.39** Charge-pump clock driver.

*NMOS Clock Driver*

Figure 18.40a shows an NMOS bootstrapped inverter. This circuit is useful to avoid latch-up since an n-well isn't present. The pull-up device, Mpu, is made 4 times longer than the pull-down device, Mpd, so that the output voltage can swing closer to ground. The transistor, Mc, is used as a capacitor to "boot" the gate of Mpu above *VDD* so that it can turn fully on and drive the output to *VDD*. Md is used as a diode, as in the other charge pumps seen in this chapter, to ensure that the boot node doesn't drop below $VDD - V_{THN}$. Note that this isn't a static circuit, that is, the input must be active else the high output voltage is only $VDD - 2V_{THN}$. In order to avoid continuous current flow when both Mpd and Mpu are conducting the NMOS "clock" driver (a buffer) seen in Fig. 18.40b can be used. The 40/1 output devices are both actively driven.

IPR2023-00697
Theta EX2019

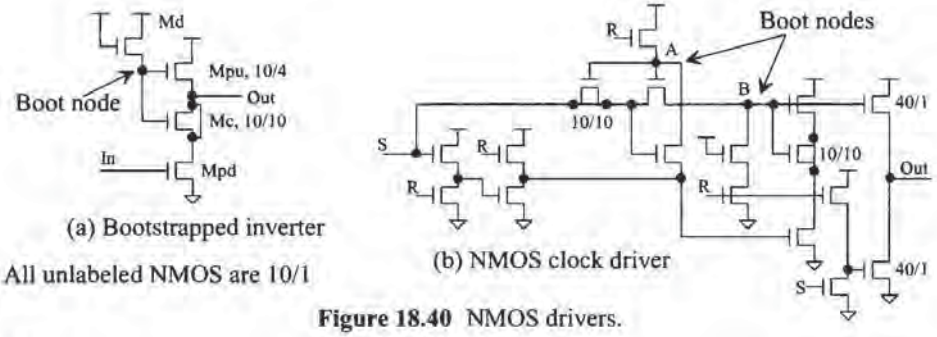



**Figure 18.40** NMOS drivers.

### 18.4.3 Example

A common application of a voltage generator in digital circuits is generating a negative substrate bias; that is, instead of tying the substrate to ground, the substrate is held at some negative voltage. Typically, this negative voltage is between –0.5 and –1 V. "Pumping" the substrate negative is found in some DRAMs. A negative substrate bias has several benefits. It (1) stabilizes n-channel threshold voltages, (2) increases latch-up immunity after power up, (3) prevents forward biasing n+ to p-substrate pn junction, (4) allows chip inputs to go negative without forward biasing a pn junction, (5) prevents substrate from going locally above ground, (6) reduces depletion capacitances associated with the n+ to p-substrate junction, and (7) reduces subthreshold leakage current.

A simple substrate pump is shown in Fig. 18.41. In this circuit we can use n-channel MOSFETs to generate a negative potential, unlike Fig. 18.35, since the negative voltage is connected to the substrate. In this situation, the drain/source implants of the n-channel MOSFETs cannot become forward biased. Note the absence of the load capacitance. It turns out that the capacitance of the substrate to everything else in the circuit presents a very large capacitance to the pump. In other words, the substrate itself is a very large capacitor.

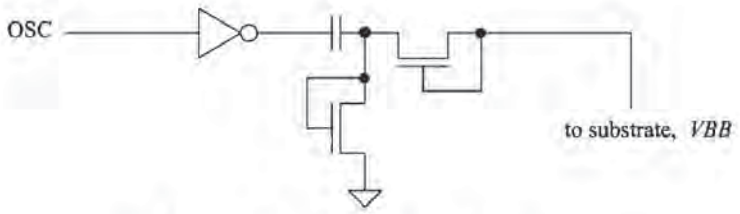



**Figure 18.41** Simple substrate pump.

An example oscillator circuit that is used to drive the substrate pump is shown in Fig. 18.42. This is a standard ring oscillator with a NAND gate to enable/disable the oscillator. Capacitors are added at a few points in the middle of the oscillator to increase the delay and lower the frequency of the oscillator so that the pump's capacitors fully charge.

The final component in a substrate pump is the regulator, a circuit that senses the voltage on the substrate and enables or disables the substrate pump (the oscillator). Using

IPR2023-00697
Theta EX2019

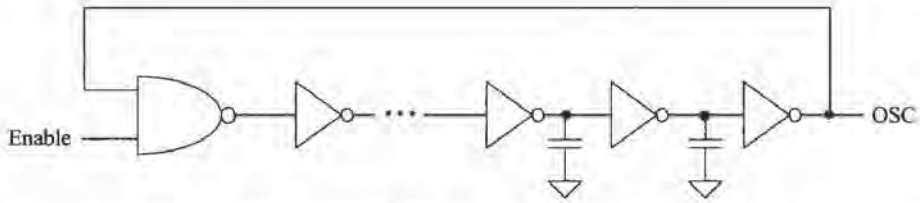



**Figure 18.42** Ring oscillator with enable.

a comparator with hysteresis, a precision voltage reference, and a level shifting circuit, the enable signal can be generated with the circuit of Fig. 18.43. The hysteresis of the comparator determines the amount of ripple on the substrate voltage. Forcing a constant current through the two MOSFETs, M1 and M2, compels their source-gate voltages to remain constant (note how the body effect is eliminated by using p-channel MOSFETs), causing the MOSFETs to behave as if they were batteries. The battery action of M1 and M2 shifts the substrate voltage up so that it lies in the input range of the comparator. The number of MOSFETs (in this case two) and the magnitude of the current $I$ determine the substrate voltage.

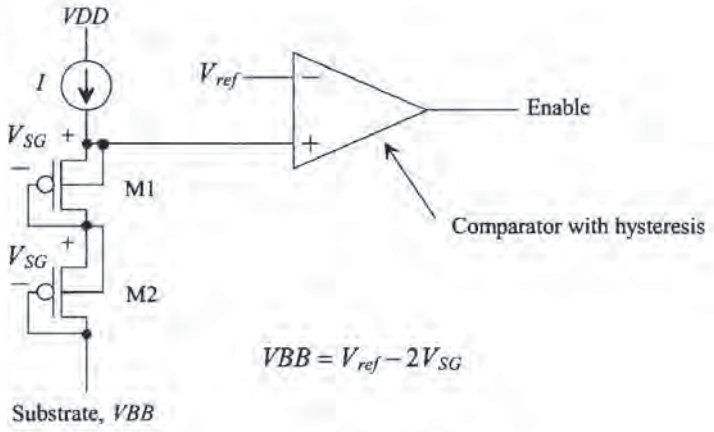



**Figure 18.43** Regulator circuit used in substrate pump.

A simpler and less accurate implementation of the regulator is shown in Fig. 18.44. The comparator and voltage reference are implemented with an inverter and M1–M3. MOSFET M3 causes the inverter to have hysteresis, that is, behave like a Schmitt trigger. MOSFETs M1 and M2 form an inverter with a switching point voltage of approximately $V_{THN}$. The current source of Fig. 18.43 is implemented with the long L MOSFET M4 in Fig. 18.44. The level shifting is performed with the n-channel MOSFETs M5 and M6. When point A gets above a potential of $V_{THN}$, the Enable output goes high, causing the substrate pump to turn on and drive the substrate voltage negative. When point A gets pulled, via the substrate voltage through M5 and M6, below $V_{THN}$, the Enable output goes low and the pump shuts off. The substrate voltage generated with this circuit is approximately $-V_{THN}$ with body effect.

IPR2023-00697
Theta EX2019

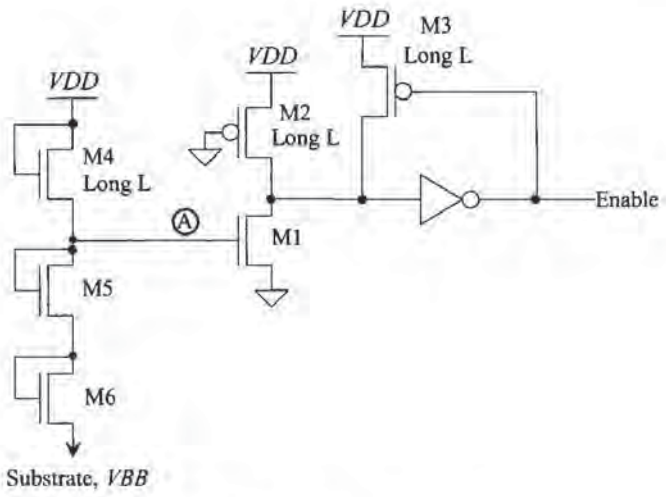



**Figure 18.44** Simpler substrate regulator.

## ADDITIONAL READING

[1] R. J. Baker, "Input and output buffers having symmetrical operating characteristics and immunity from voltage variations," US Patent 7,102,932, September 5, 2006.

[2] F. Pan and T. Samaddar, *Charge Pump Circuit Design*, McGraw-Hill, 2006. ISBN 978-0071470452.

[3] H. Lin, N.-H. Chen, and J. Lu, "Design of Modified Four-Phase CMOS Charge Pumps for Low-Voltage Flash Memories," *Journal of Circuits, Systems, and Computers*, Vol. 11, No. 4, pp. 393-403, 2002. Excellent paper.

[4] N. Otsuka and M. A. Horowitz, "Circuit techniques for 1.5-V power supply flash memory," *IEEE Journal of Solid-State Circuits,* Vol. 32, pp. 1217–1230, August 1997.

[5] M. Bazes, "Two Novel Full Complementary Self-Biased CMOS Differential Amplifiers," *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 2, pp. 165-168, February 1991.

[6] J. F. Dickson, "On-Chip High-Voltage Generation in MNOS (sic) Integrated Circuits Using an Improved Voltage Multiplier Technique," *IEEE Journal of Solid State Circuits*, Vol. SC-11, No. 3, June 1976.

[7] J. Millman and H. Taub, *Pulse, Digital, and Switching Waveforms*, McGraw-Hill Publishers, 1965. ISBN 07-042386-5. Information on multivibrators and oscillators.

[8] O. H. Schmitt, "A Thermionic Trigger," *Journal of Scientific Instruments*, Vol. XV, pp. 24–26, No. 1, Jan. 1938. Information on the "Schmitt trigger."

IPR2023-00697
Theta EX2019

## PROBLEMS

For the following problems use the short-channel CMOS process with a scale factor of 50 nm and a *VDD* of 1 V.

**18.1** Design a Schmitt trigger with $V_{SPL} = 0.35$ and $V_{SPH} = 0.55$ V. Use SPICE to verify your design.

**18.2** Estimate $t_{PHL}$ and $t_{PLH}$ for the Schmitt trigger of Ex. 18.1, driving a 100 fF load capacitance. Compare your hand calculations to simulation results.

**18.3** Design and simulate the operation of a Schmitt trigger-based oscillator with an output frequency of 10 MHz. Simulate the design using SPICE.

**18.4** Estimate the total input capacitance on the control voltage input for the VCO shown in Fig. 18.8.

**18.5** Design an astable multivibrator with an output oscillation frequency of 20 MHz.

**18.6** Design and simulate the operation of a one-shot that has an output pulse width of 100 ns. Comment on the maximum rate of retrigger and how ESD diodes connected to bonding pads will affect the circuit operation if the resistor and capacitor are bonded out.

**18.7** Design and simulate a 100 MHz oscillator using the astable multivibrator in Fig. 18.11. Comment how process shifts in the resistor and capacitor affects the oscillation frequency. If the resistor and capacitor are bonded out, will the ESD diodes affect the circuit's operation?

**18.8** Using the buffer seen in Fig. 18.23 to drive a load capacitance of 100 *fF*, plot the propagation delays against the *Vinp* input signal amplitude when *Vinm* is 250, 500, and 700 mV. (*Vinp* is centered around *Vinm*.)

**18.9** The DC restore circuit seen in Fig. 18.28 has limitations, such as it stops working if a long string of 1s or 0s is applied to the input buffer or the input data moves too quickly and the long RC time constants keep the DC restore output from following the center of the data. Comment, with the help of SPICE simulation results, on these limitations. Show how changing the values of the resistors and capacitors can compensate for these limitations. (For example, using longer time constants allows for longer strings of ones are zeroes before the DC restore signal is invalid.)

**18.10** Using the model for the transmission channel seen in Figs. 18.26 and 18.30, show that the circuit in Fig. 18.32 will work as an input buffer. Using SPICE show the limitations of the buffer (signal amplitude and speed). Also, comment on the power pulled by the buffer.

**18.11** Design a nominally 2 V voltage generator that can supply at most 1 μA of DC current (2 MEG resistor). Simulate the operation of your design with SPICE.

**18.12** Design and simulate the operation of a nominally –1 V substrate pump. Comment on the design trade-offs.

IPR2023-00697
Theta EX2019

Chapter

19

# Digital Phase-Locked Loops

The digital phase-locked loop, DPLL, is a circuit that is used frequently in modern integrated circuit design. Consider the waveform and block diagram of a communication system shown in Fig. 19.1. Digital data[1] is loaded into the shift register at the transmitting end. The data is shifted out sequentially to the transmitter output driver. At the receiving end, where the data may be analog (and, thus, without well-defined amplitudes) after passing through the communication channel, the receiver amplifies and changes the data back into digital logic levels. The next logical step in this sequence is to shift the data back into a shift register at the receiver and process the received data. However, the absence of a clock signal makes this difficult. The DPLL performs the function of generating a clock signal, which is locked or synchronized with the incoming signal. The generated clock signal of the receiver clocks the shift register and thus recovers the data. This application of a DPLL is often termed *a clock-recovery circuit* or *bit synchronization circuit*.

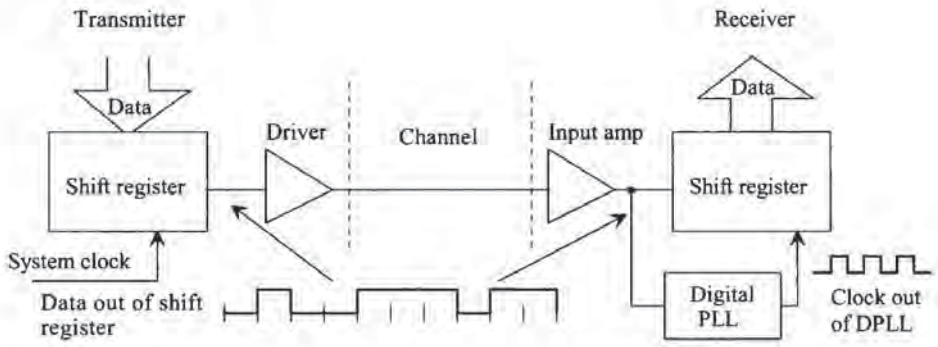



**Figure 19.1** Block diagram of a communication system using a DPLL for the generation of a clock signal.

[1] While "data" is plural and "datum" is singular, we will not use, in this chapter, the grammatically correct "data are" in favor of the colloquial "data is".

IPR2023-00697
Theta EX2019

A more detailed picture of the incoming data and possible clock signals out of the DPLL are shown in Fig. 19.2. The possible clock signals are labeled XOR *clock* and PFD (phase frequency detector) *clock*, corresponding to the type of phase[2] detector (PD) used. For the XOR PD, the rising edge of the clock occurs in the center of the data, while for the PFD, the rising edge occurs at the beginning of the data. The phase of the clock signal is determined by the PD used[3].

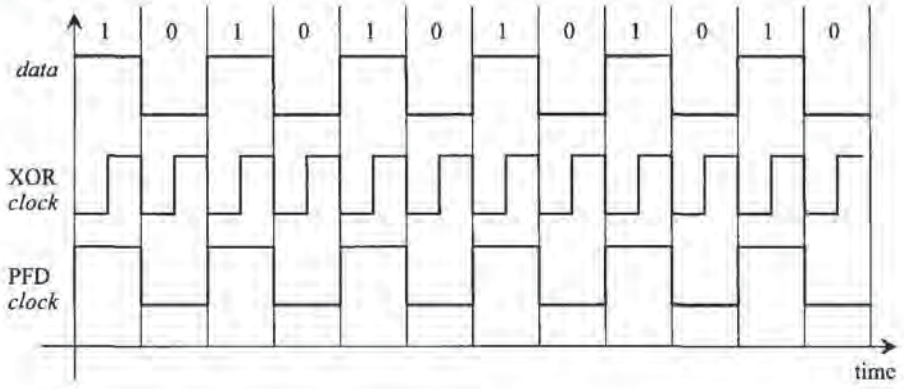



**Figure 19.2** Data input to DPLL in lock and possible clock outputs using the XOR phase detector and PFD.

A block diagram of a DPLL is shown in Fig. 19.3. The PD generates an output signal proportional to the time difference between the *data in* and the divided down clock, *dclock*. This signal is filtered by a loop filter. The filtered signal, $V_{inVCO}$, is connected to the input of a voltage-controlled oscillator (VCO). Each one of these blocks is discussed in detail in the following sections. Once each block is understood, we will put them together and discuss the operation of the DPLL.

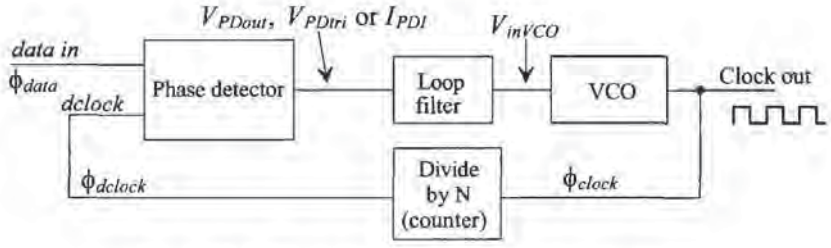



**Figure 19.3** Block diagram of a digital phase-locked loop.

[2] A more correct name for digital applications is time difference detector (TDD).

[3] The PFD is rarely used in clock recovery applications. Figure 19.2 simply illustrates the different phase relations resulting from an XOR PD or PFD in a locked DPLL.

IPR2023-00697
Theta EX2019

## 19.1 The Phase Detector

The first component in our DPLL is the phase detector. The two types of phase detectors, an XOR gate and a phase frequency detector (PFD), have significantly different characteristics. It is therefore very important to understand their performance capabilities and limitations. Selection of the type of phase detector is the first step in a DPLL design.

### 19.1.1 The XOR Phase Detector

The XOR PD is simply an exclusive OR gate. When the output of the XOR is a pulse train with a 50% duty cycle (square wave), the DPLL is said to be in lock; or in other words, the clock signal out of the DPLL is synchronized to the incoming data, provided the following conditions are met. Consider the XOR PD shown in Fig. 19.4. Let's begin by assuming that the incoming data is a string of zeros and that a divide by two is used in the feedback loop. The output of the phase detector is simply a replica of the *dclock* signal. Since the *dclock* signal has a 50% duty cycle, it would appear that the DPLL is in lock. If a logic "1" is suddenly applied, there is no way to know if the clock signal is synchronized (the clock rising edge coincides with the center of the data bit) to the data. This leads to the first characteristic of an XOR PD;

1. The incoming data must have a minimum number of transitions over a given time interval.

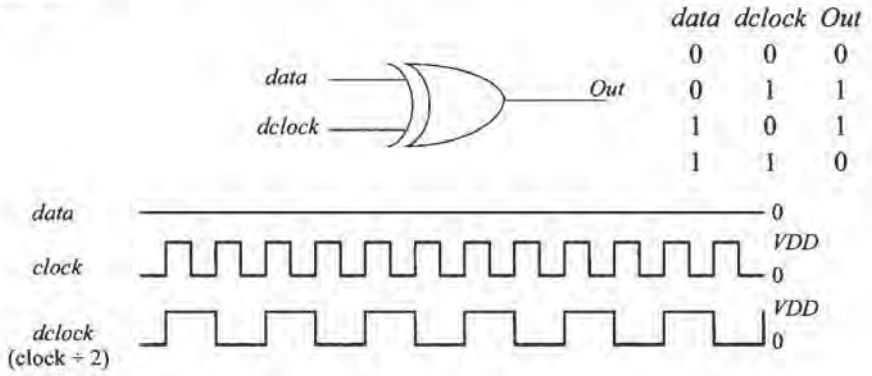



**Figure 19.4** Operation of the XOR phase detector.

Now consider the situation when the output of the phase detector, with a string of zeros as the *data* input, is applied to a simple RC low-pass filter (Fig. 19.5). If RC >> period of the clock signal, the output of the filter is simply *VDD*/2. This leads to the second characteristic of the XOR phase detector.

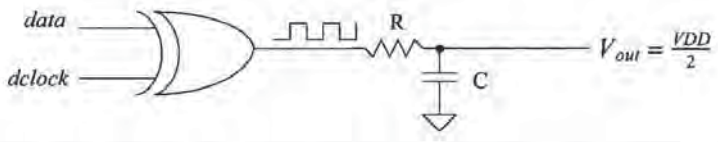



**Figure 19.5** How the filtered output of the phase detector becomes VDD/2.

IPR2023-00697
Theta EX2019

2. With no input data, the filtered output of the phase detector is *VDD*/2.

The voltage out of the loop filter is connected to the input of the VCO, as seen in Fig. 19.3. Consider the typical characteristics of a VCO shown in Fig. 19.6. The frequency of the square wave output of the VCO is $f_{center}$ when $V_{in}$ ( = $V_{center}$) is *VDD*/2 (typically). The other two frequencies of interest are the minimum and maximum oscillator frequencies, $f_{min}$ and $f_{max}$ possible, with input voltage $V_{min}$ and $V_{max}$, respectively. It is important that the VCO continues to oscillate with no input data. Normally, the VCO is designed so that the nominal data input rate and the VCO center frequency are the same. This minimizes the time it takes the DPLL to lock (and is critical for proper operation of the XOR PD).

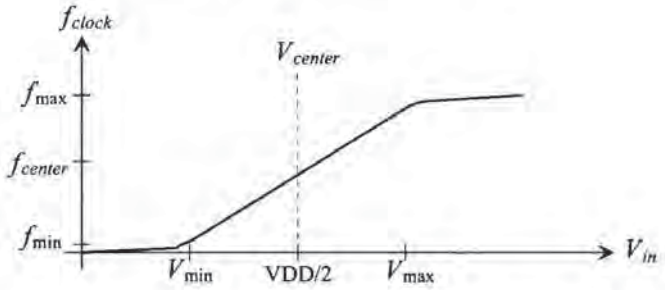



**Figure 19.6** Output frequency of VCO versus input control voltage.

Now let's consider an example input to the phase-detector and corresponding output. The data shown in Fig. 19.7 is leading the *dclock* signal. The corresponding output of the phase detector is also shown. If this output is applied to a low-pass filter, the result is an average voltage less than *VDD*/2. This causes the VCO frequency to decrease until the edge of the *dclock* is centered on the data.

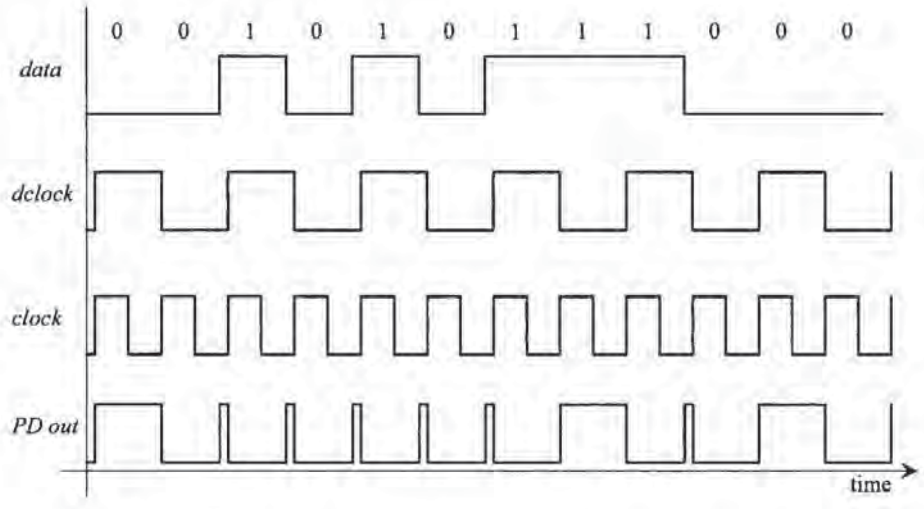



**Figure 19.7** Possible XOR phase-detector inputs and the resulting output.

IPR2023-00697
Theta EX2019

3. The time it takes the loop to lock depends on the data pattern input to the DPLL and the loop-filter characteristics.

Since the output of the PD is averaged, or more correctly integrated, noise injected into the data stream (a false bit) can be rejected. A fourth characteristic is:

4. The XOR DPLL has good noise rejection.

Another important characteristic of a DPLL is whether or not it will lock on a harmonic of the input data. The XOR DPLL will lock on harmonics of the data. To prove that this is indeed the case, consider replacing any of the clock signals of Figs. 19.2, 19.4, or 19.7 with a clock at twice or half the frequency. The average of the waveforms will remain essentially the same. A fifth characteristic of a DPLL using an XOR gate is:

5. The VCO operating frequency range should be limited to frequencies much less than $2f_{clock}$ and much greater than $0.5f_{clock}$, where $f_{clock}$ is the nominal clock frequency for proper lock with a XOR PD.

The loop filter used with this type of PD is a simple RC low-pass filter, as shown in Fig. 19.5. Since the output of the PD is oscillating, the output of the filter shows a ripple as well, even when the loop is locked. This modulates the clock frequency, an unwanted characteristic of a DPLL using the XOR PD. This characteristic can be added to our list:

6. A ripple on the output of the loop filter with a frequency equal to the clock frequency modulates the control voltage of the VCO.

To characterize the phase detector (see Fig. 19.8), we can define the time difference between the rising edge of the *dclock* and the beginning of the *data* as $\Delta t$. The phase difference between the *dclock* and *data*, $\phi_{data} - \phi_{dclock}$, is given by

$$\Delta\phi = \phi_{data} - \phi_{dclock} = \frac{\Delta t}{T_{dclock}} \cdot 2\pi \text{ (radians)} \tag{19.1}$$

or, in terms of the DPLL output clock frequency,

$$\Delta\phi = \frac{\Delta t}{2T_{clock}} \cdot 2\pi \tag{19.2}$$

$$f_{clock} = \frac{1}{T_{clock}} = 2f_{dclock} = \frac{2}{T_{dclock}} \tag{19.3}$$

When the loop is locked, the *clock* rising edge is centered on the data; the time difference, $\Delta t$, between the *dclock* rising edge and the beginning of the data is simply $T_{clock}/2$ or $T_{dclock}/4$ (see Fig. 19.8c). Therefore, the phase difference between *dclock* and the *data*, under locked conditions, may be written as

$$\Delta\phi = \frac{\pi}{2} \tag{19.4}$$

Note that the phase difference between *clock* and *data* when in lock is $\pi$. The average voltage out of the phase detector (Fig. 19.8) may be expressed by

$$V_{PDout} = VDD \cdot \frac{\Delta\phi}{\pi} = K_{PD} \cdot \overbrace{\Delta\phi}^{input} \tag{19.5}$$

where the gain of the PD may be written as

IPR2023-00697
Theta EX2019

$$K_{PD} = \frac{VDD}{\pi} \text{ (V/radians)} \tag{19.6}$$

As an aid in the understanding of these equations, consider the diagrams shown in Fig. 19.8. If the edges of the clock and data are coincident in time (Fig. 19.8a), the XOR output, $V_{PDout}$, is 0 V and the phase difference is 0. The loop filter averages the output of the PD and causes the VCO to lower its output frequency. This causes $\Delta\phi$ to increase, thus increasing $V_{PDout}$. Depending on the selection of the loop filter, this increase could cause the rising edges to increase beyond, or *overshoot*, the desirable center point, as shown in Fig. 19.8b. In this case, the phase difference is $-\frac{3}{4}\pi$, and $V_{PDout}$ is $\frac{3}{4}VDD$. Figure 19.8c shows the condition when the loop is in lock and the phase difference is $\pi/2$.

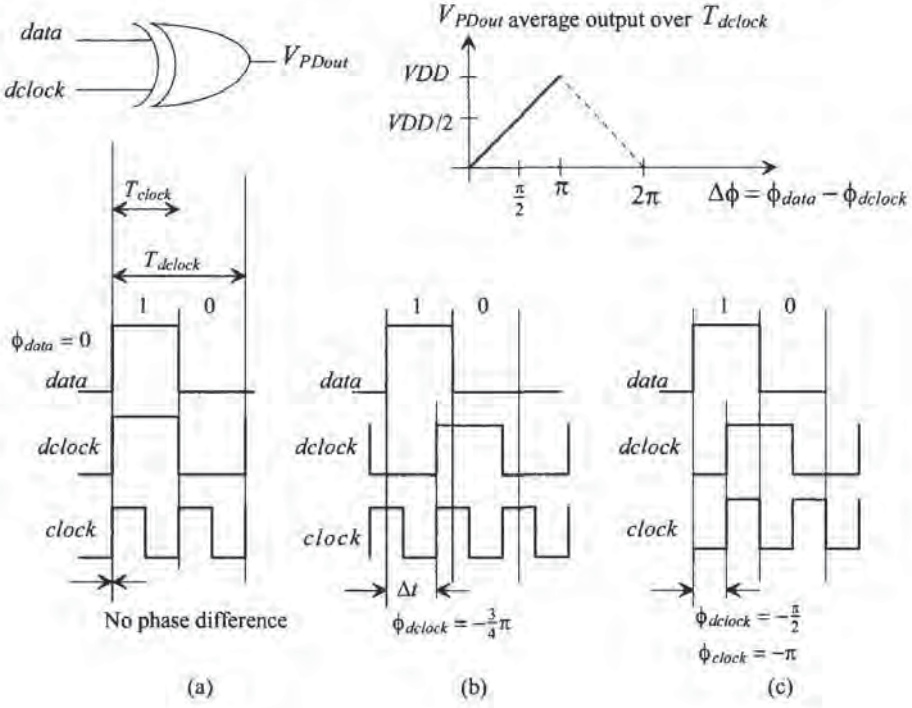



**Figure 19.8** XOR PD output for various inputs (assuming input data are a string of alternating ones and zeros).

Here we should point out the importance of the VCO center frequency, $f_{center}$, being equal to the desired clock frequency. If $f_{center} = f_{clock}$, the VCO control voltage, depending on the loop filter, will look similar to Fig. 19.9a during acquisition (the loop trying to lock). If these frequencies are not equal, the control voltage will oscillate, causing the clock to move, in time, around some other point than the center of the data bit (Fig. 19.9b). The actual control voltages will look different from those portrayed in Fig. 19.9 because the VCO control voltage depends on the input data pattern, as discussed earlier.

IPR2023-00697
Theta EX2019

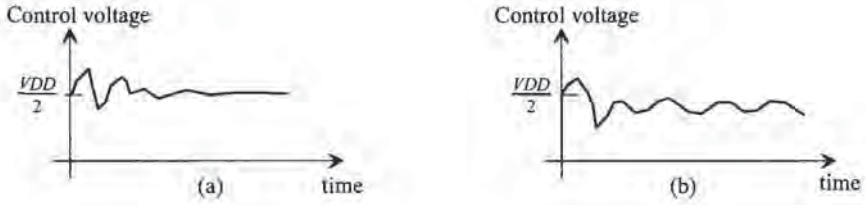



**Figure 19.9** Average output voltage of phase detector during acquisition.

To summarize the design criteria of the VCO with the XOR PD, we desire that:

1. The center frequency, $f_{center}$, should equal the clock frequency when the VCO control voltage is *VDD*/2.

2. The maximum and minimum oscillation frequencies, $f_{max}$ and $f_{min}$, of the VCO should be selected to avoid locking on harmonics of the input data.

3. The VCO duty cycle is 50%. If this is not the case, the DPLL will have problems locking, or once locked the clock will jitter (move around in time).

### 19.1.2 The Phase Frequency Detector

A schematic diagram of the phase frequency detector is shown in Fig. 19.10. The output of the PFD depends on both the phase and frequency of the inputs. This type of phase detector is also termed a sequential phase detector. It compares the leading edges of the *data* and *dclock*. A *dclock* rising edge cannot be present without a data rising edge. To aid in understanding the PFD, consider the examples shown in Fig. 19.11. The first thing we notice is that the *data* pulse width and the *dclock* pulse width do not matter. If the rising edge of the *data* leads the *dclock* rising edge (Fig. 19.11a), the "up" output of the phase detector goes high, while the *Down* output remains low. This causes the *dclock* frequency to increase, having the effect of moving the edges closer together. When the *dclock* signal leads the data (Fig. 19.11b), *Up* remains low, while the *Down* goes high a time equal to the phase difference between *dclock* and *data*. Figure 19.11c shows the condition of the locked loop. Notice that, unlike the XOR PD, the outputs remain low when the loop is locked. Again, several characteristics of the PFD can be described:

1. A rising edge from the *dclock* and *data* must be present when making a phase comparison.

2. The widths of the *dclock* and the *data* are irrelevant.

3. The PFD will not lock on a harmonic of the data.

4. The outputs (*Up* and *Down*) of the PFD are both logic low when the loop is in lock, eliminating ripple on the output of the loop filter.

5. This PFD has poor noise rejection; a false edge on either the *data* or the *dclock* inputs drastically affects the output of the PFD.

IPR2023-00697
Theta EX2019

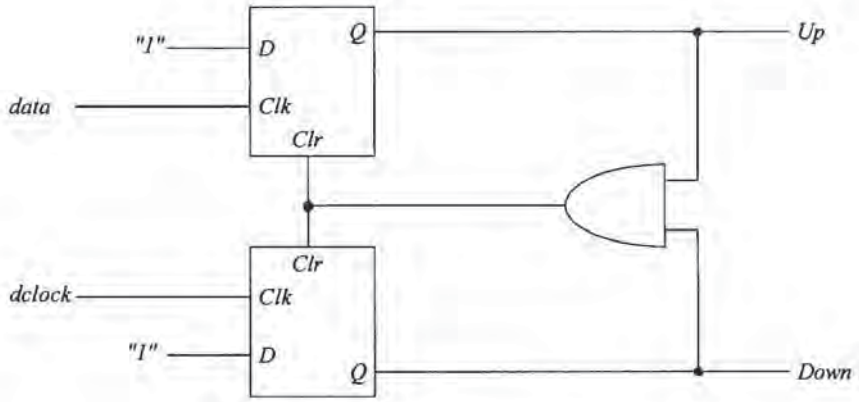



**Figure 19.10** Phase frequency detector (PFD).

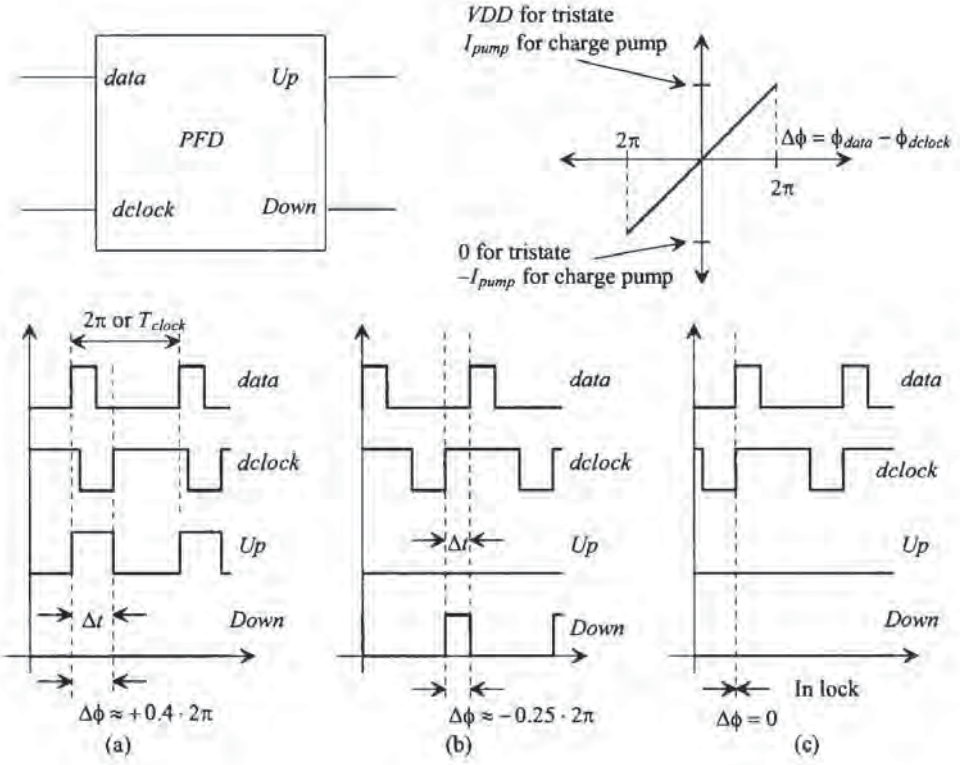



**Figure 19.11** PFD phase-detector inputs and outputs.

IPR2023-00697
Theta EX2019

The output of the PFD should be combined into a single output for driving the loop filter. There are two methods of doing this, both of which are shown in Fig. 19.12. The first method is called a *tri-state* output. When both signals, *Up* and *Down*, are low, both MOSFETs are off and the output is in a high-impedance state. If the *Up* signal goes high, M2 turns on and pulls the output up to *VDD*, while if the *Down* signal is high, the output is pulled low through M1. The main problem that exists with this configuration is that power supply variations can significantly affect the output voltage when M2 is on. The effect is to modulate the VCO control voltage. This wasn't as big a problem when the XOR PFD was used due to the averaging taking place.

The second configuration shown in this figure is the so-called *charge pump*. MOS current sources are placed in series with M1 and M2. When the PFD *Up* signal goes high, M2 turns on, connecting the current source to the loop filter. Because the current source can be made insensitive to supply variations, modulation of the VCO control voltage is absent.

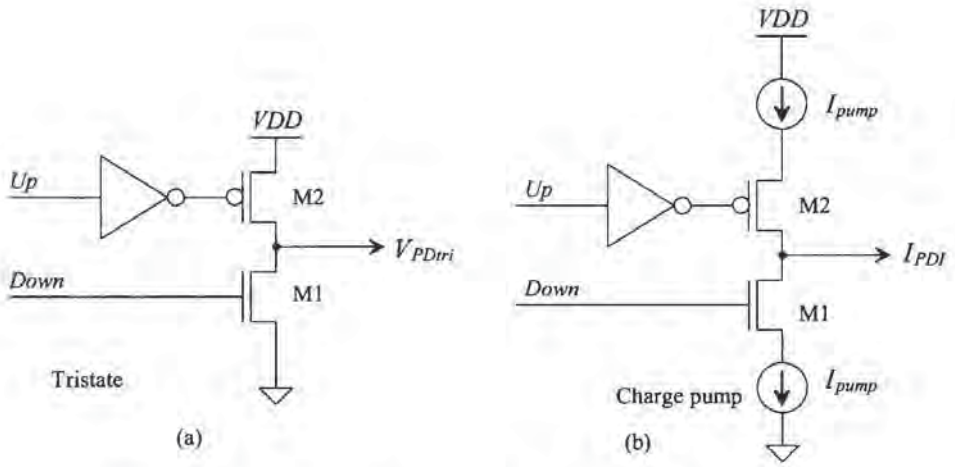



**Figure 19.12** (a) Tri-state and (b) charge pump outputs of the PFD.

We can characterize the PFD in much the same manner as we did the XOR PD. We will assume that $f_{clock}$ and $f_{dclock}$ are equal in this analysis. Therefore, the feedback loop, Fig. 19.3, divides by one ($N = 1$). If we again assume that the time difference between the rising edges of the *data* and *dclock* is labeled $\Delta t$ and the time between leading edges of the *clock* (or the time differences between the leading edges of the *data* since we must have both leading edges present to do a phase comparison) is labeled $T_{clock}$, then we can write the phase ($T_{clock} = T_{dclock}$) as

$$\Delta\phi = \frac{\Delta t}{T_{clock}} \cdot 2\pi \text{ (radians)} \tag{19.7}$$

The phase difference, $\Delta\phi$, is zero when the loop is in lock. The output voltage of the PFD using the tri-state output configuration (see Fig. 19.11) is

$$V_{PDtri} = \frac{VDD - 0}{4\pi} \cdot \Delta\phi = K_{PDtri} \cdot \Delta\phi \tag{19.8}$$

IPR2023-00697
Theta EX2019

where the gain is,

$$K_{PDtri} = \frac{VDD}{4\pi} \text{ (volts/radian)} \tag{19.9}$$

If the output of the PFD uses the charge-pump configuration, the output current can be written (again see Fig. 19.11) as

$$I_{PDI} = \frac{I_{pump} - (-I_{pump})}{4\pi} \cdot \Delta\phi = K_{PDI} \cdot \Delta\phi \tag{19.10}$$

where

$$K_{PDI} = \frac{I_{pump}}{2\pi} \text{ (amps/radian)} \tag{19.11}$$

The loop filters used with tri-state and the charge-pump outputs are shown in Fig. 19.13. The first loop filter has a transfer function given by

$$V_{inVCO} = \frac{1 + j\omega R_2 C}{1 + j\omega(R_1 + R_2)C} \cdot V_{PDtri} = K_F \cdot V_{PDtri} \tag{19.12}$$

The charge-pump loop-filter transfer function is given (noting the input variable is a current while the output is a voltage) by

$$V_{inVCO} = I_{PDI} \cdot \frac{1 + j\omega R C_1}{j\omega(C_1 + C_2) \cdot \left[1 + j\omega R \frac{C_1 C_2}{C_1 + C_2}\right]} = K_F \cdot I_{PDI} \tag{19.13}$$

To qualitatively understand how these loop filters work, let's begin by considering the loop filter for use with the tri-state output. For slow variations in the phase difference, the filter acts like an integrator averaging the output of the PD. For fast variations, however, the filter looks like a resistive divider without any integration. This allows the loop filter to track fast variations in the time difference between the rising edges. A similar discussion can be made for the loop filter used with the charge pump. For slow variations in the phase, the current, $I_{pump}$, linearly charges $C_1$ and $C_2$. This gives an averaging effect. For fast variations, the charge pump simply drives the resistor $R$ (assuming $C_2$ is small), eliminating the averaging and allowing the VCO to track quickly moving variations in the input data.

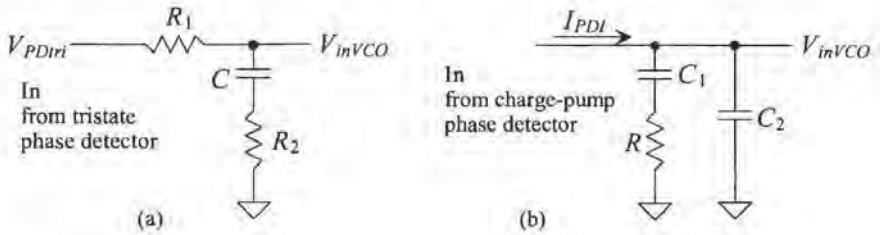



**Figure 19.13** Loop filters for (a) tristate output and (b) charge-pump output.

IPR2023-00697
Theta EX2019

The design requirements of the VCO used with the PFD can be much more relaxed than those for the XOR PD; therefore, it is preferred over the XOR PD. Because the output of the PD can be a voltage from 0 to *VDD*, the requirement that $f_{center} = f_{clock}$ is not present, although it is still a good idea to design the VCO so that this is true. The oscillation range, $f_{max} - f_{min}$, is not limited by the harmonics of $f_{clock}$ because the PFD will not lock on a harmonic of the clock frequency. The VCO clock duty cycle is irrelevant with the PFD because the PD looks only at rising edges. In high-speed or data communications applications, however, one may be forced to use the XOR PD.

## 19.2 The Voltage-Controlled Oscillator

The gain of the voltage-controlled oscillator is simply the slope of the curves given in Fig. 19.6. This gain can be written as

$$K_{VCO} = 2\pi \cdot \frac{f_{\max} - f_{\min}}{V_{\max} - V_{\min}} \text{ (radians/s} \cdot \text{V)} \tag{19.14}$$

The VCO output frequency, $f_{clock}$, is related to the VCO input voltage (see Fig. 19.6) by

$$\omega_{clock} = 2\pi \cdot f_{clock} = K_{VCO} \cdot V_{inVCO} + \omega_o \text{ (radians/s)} \tag{19.15}$$

where $\omega_o$ is a constant. However, the variable we are feeding back is not frequency but phase (hence the name of the circuit). The phase of the VCO clock output is related to $f_{clock}$ by

$$\phi_{clock} = \int \omega_{clock} \cdot dt = \frac{K_{VCO}}{j\omega} \cdot V_{inVCO} \text{ (radians)} \tag{19.16}$$

where this signal can be related to the $\phi_{dclock}$ by

$$\phi_{dclock} = \frac{1}{N} \cdot \phi_{clock} = \beta \cdot \phi_{clock} \tag{19.17}$$

where $N$ is the divide by count and $\beta$ is the feedback factor.

### 19.2.1 The Current-Starved VCO

The current-starved VCO is shown schematically in Fig. 19.14. Its operation is similar to the ring oscillator discussed earlier. MOSFETs M2 and M3 operate as an inverter, while MOSFETs M1 and M4 operate as current sources. The current sources, M1 and M4, limit the current available to the inverter, M2 and M3; in other words, the inverter is starved for current. The drain currents of MOSFETs M5 and M6 are the same and are set by the input control voltage. The currents in M5 and M6 are mirrored in each inverter/current source stage. An important property of the VCO used in any of the CMOS DPLLs discussed in this chapter is the input impedance. The filter configurations we have discussed rely on the fact that the input resistance of the VCO is practically infinite and the input capacitance is small compared to the capacitances present in the loop filter. Attaining infinite input resistance is usually an easy part of the design. For the charge-pump configuration, the input capacitance of the VCO can be added to $C_2$.

To determine the design equations for use with the current-starved VCO, consider the simplified schematic of one stage of the VCO shown in Fig. 19.15. The total capacitance on the drains of M2 and M3 is given by

IPR2023-00697
Theta EX2019

$$C_{tot} = C_{out} + C_{in} = \overbrace{C'_{ox}(W_pL_p + W_nL_n)}^{C_{out}} + \overbrace{\frac{3}{2}C'_{ox}(W_pL_p + W_nL_n)}^{C_{in}} \quad (19.18)$$

which is simply the output and input capacitances of the inverter. This equation can be written in a more useful form as

$$C_{tot} = \frac{5}{2}C'_{ox}(W_pL_p + W_nL_n) \quad (19.19)$$

The time it takes to charge $C_{tot}$ from zero to $V_{SP}$ with the constant-current $I_{D4}$ is given by

$$t_1 = C_{tot} \cdot \frac{V_{SP}}{I_{D4}} \quad (19.20)$$

while the time it takes to discharge $C_{tot}$ from $VDD$ to $V_{SP}$ is given by

$$t_2 = C_{tot} \cdot \frac{VDD - V_{SP}}{I_{D1}} \quad (19.21)$$

If we set $I_{D4} = I_{D1} = I_D$ (which we will label $I_{Dcenter}$ when $V_{inVCO} = VDD/2$), then the sum of $t_1$ and $t_2$ is simply

$$t_1 + t_2 = \frac{C_{tot} \cdot VDD}{I_D} \quad (19.22)$$

The oscillation frequency of the current-starved VCO for $N$ (an odd number $\geq 5$) of stages is

$$f_{osc} = \frac{1}{N(t_1 + t_2)} = \frac{I_D}{N \cdot C_{tot} \cdot VDD} \quad (19.23)$$

which is $= f_{center}$ (@$V_{inVCO} = VDD/2$ and $I_D = I_{Dcenter}$)

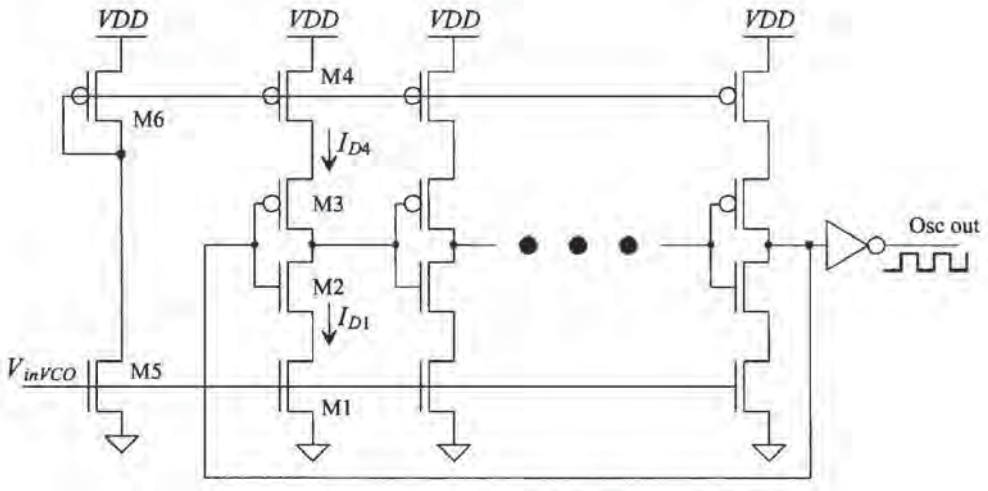



**Figure 19.14** Current-starved VCO.

IPR2023-00697
Theta EX2019

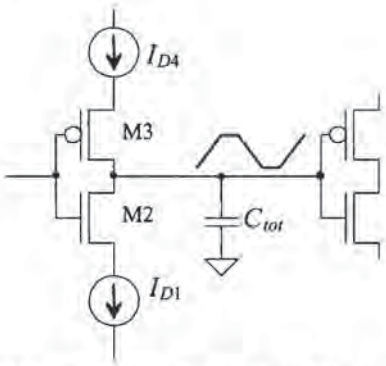



**Figure 19.15** Simplified view of a single stage of the current-starved VCO.

Equation (19.23) gives the center frequency of the VCO when $I_D = I_{Dcenter}$. The VCO stops oscillating, neglecting subthreshold currents, when $V_{inVCO} < V_{THN}$. Therefore, we can define

$$V_{\min} = V_{THN} \text{ and } f_{\min} = 0 \tag{19.24}$$

The maximum VCO oscillation frequency, $f_{max}$, is determined by finding $I_D$ when $V_{inVCO} = VDD$. At the maximum frequency then, $V_{max} = VDD$.

The output of the current-starved VCO shown in Fig. 19.14 is normally buffered through one or two inverters. Attaching a large load capacitance on the output of the VCO can significantly affect the oscillation frequency or lower the gain of the oscillator enough to kill oscillations altogether.

The average current drawn by the VCO is

$$I_{avg} = N \cdot \frac{VDD \cdot C_{tot}}{T} = N \cdot VDD \cdot C_{tot} \cdot f_{osc} \tag{19.25}$$

or

$$I_{avg} = I_D \tag{19.26}$$

The average power dissipated by the VCO is

$$P_{avg} = VDD \cdot I_{avg} = VDD \cdot I_D \tag{19.27}$$

If we include the power dissipated by the mirror MOSFETs, M5 and M6, the power is doubled from that given by Eq. (19.27), assuming that $I_D = I_{D5} = I_{D6}$. For low-power dissipation we must keep $I_D$ low, which is equivalent to stating that for low-power dissipation we must use a low-oscillation frequency.

---

**Example 19.1**

Design a current-starved VCO with $f_{center} = 100$ MHz in the short-channel process. Simulate the design using SPICE.

We begin by calculating the total capacitance, $C_{tot}$. Using Eq. (19.19) and assuming the inverters, M2 and M3, are sized for equal drive, that is, $L_n = L_p = 1$, $W_n = 10$ and $W_p = 20$, the capacitance is

IPR2023-00697
Theta EX2019

$$C_{tot} = \frac{5}{2} \cdot 25\frac{fF}{\mu m^2} \cdot (10 \cdot 1 + 20 \cdot 1) \cdot \overbrace{(0.050\ \mu m)^2}^{\text{scale factor}} = 4.7\ fF$$

Let's use a center drain current of 10 μA based on the $I_D$ - $V_{GS}$ characteristics of the MOSFETs. The selection of this current is important. We want, when $V_{inVCO}$ is *VDD*/2, the oscillation frequency to be 100 MHz. Here, just to show the design procedure, we will simply adjust the $V_{inVCO}$ to set the 100 MHz output frequency.

The number of stages, using Eq. (19.23), is given by

$$N = \frac{I_D}{f_{osc} \cdot C_{tot} \cdot VDD} = \frac{10\ \mu A}{100\ MHz \cdot 4.7\ fF \cdot 1\ V} \approx 21$$

The simulation results are shown in Fig. 19.16. For an output frequency of 100 MHz, the input voltage, $V_{inVCO}$ , was set to 450 mV. The big problem with this design is that the output oscillation frequency is not linearly related to the control voltage (easy to verify using the simulation that generated Fig. 19.16). Having a nonlinear VCO gain can greatly reduce the quality of the performance of the DPLL (*this is important*). The output of the phase-locked loop can jitter (move around) or may not lock at all when the VCO gain is nonlinear. ■

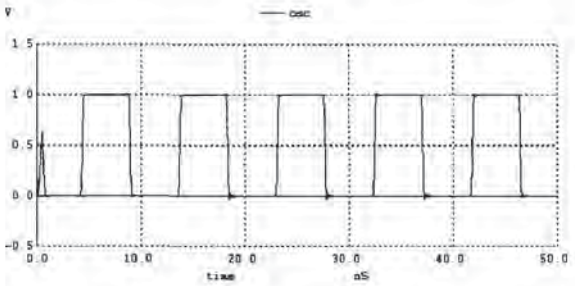



**Figure 19.16** Output of the oscillator designed in Ex. 19.1.

*Linearizing the VCO's Gain*

After studying Fig. 19.14, we see that applying $V_{inVCO}$ directly to the gate of M5 causes the current in M5 (and M6) to be nonlinear. In the long-channel case the drain current of a MOSFET is related to the square of the MOSFET's $V_{GS}$. To make the current in a MOSFET linearly related to the VCO's input voltage, consider the circuit seen in Fig. 19.17. The width of M5R is made wide so that its $V_{GS}$ is always (independent of $V_{inVCO}$) approximately $V_{THN}$. Note that the current in M6R is mirrored over to M6 and M5 to control the current used in the current-starved VCO. Figure 19.18 shows the output of the oscillator (and its transfer curves) in Fig. 19.16 if the linearizing scheme in Fig. 19.17 is used with *R* of 10k and a wide device (M5R) with a size of 100/1. The gain of the VCO is $K_{VCO} = 2\pi \cdot 25MHz/100mV = 1.57 \times 10^9\ radians/V \cdot s$. We'll use this VCO in examples later in this chapter. Note that the gain of this *VCO* is rather high. A 10 mV voltage variation in $V_{inVCO}$ gives a 2.5 MHz variation in the output frequency (e.g., the output frequency varies from 100 to 102.5 MHz). In the time domain, the variation in the period (jitter) is then $1/100\ MHz - 1/102.5\ MHz = 244\ ps$ (roughly 2.5% of the ideal 10 ns period). *For any low jitter PLL design, a VCO with low gain must be used.*

IPR2023-00697
Theta EX2019

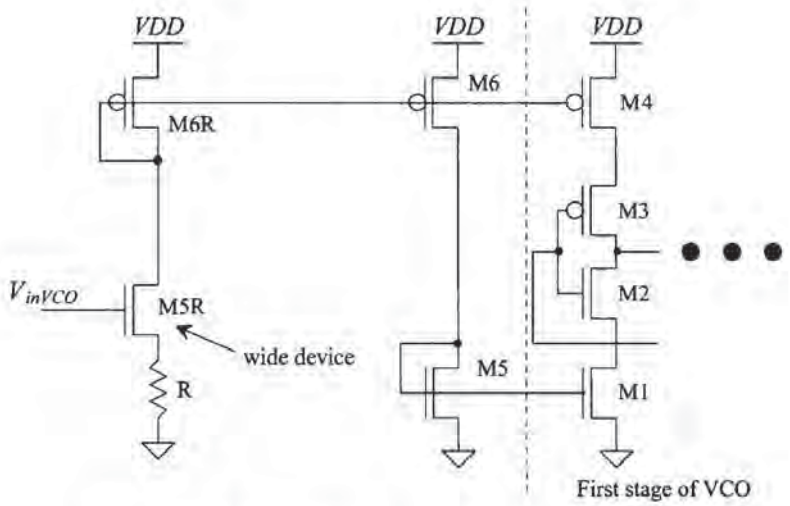



**Figure 19.17** Linearizing the current in a current-starved VCO.

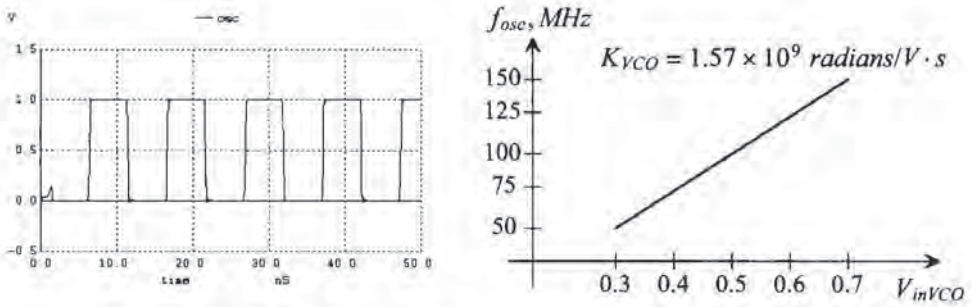



**Figure 19.18** Gain of the VCO in Fig. 19.16 after linearizing with Fig. 19.17.

### 19.2.2 Source-Coupled VCOs

Another variety of VCO, source-coupled VCOs, is shown in Fig. 19.19. These VCOs can be designed to dissipate less power than the current-starved VCO of the last section for a given frequency. The major disadvantage of these configurations is the need for a capacitor, something that may not be available in a single-poly pure digital process without using parasitics for example, a metal1 to metal2 capacitor, and a reduced output voltage swing. However, this configuration is useful when the VCO center frequency is set by an external capacitor; that is, the capacitor shown in the figure is bonded out.

To understand the operation, let's consider the NMOS source-coupled VCO of Fig. 19.19a. The operation of the CMOS source-coupled VCO of Fig. 19.19b is identical to (a) except for the fact that the load MOSFETs M3 and M4 pull the outputs to $VDD - V_{THN}$ (for the NMOS-load VCO) and $VDD$ (for the PMOS-load VCO). The buffer in Fig. 18.17 of the last chapter can be used to restore full logic levels.

IPR2023-00697
Theta EX2019

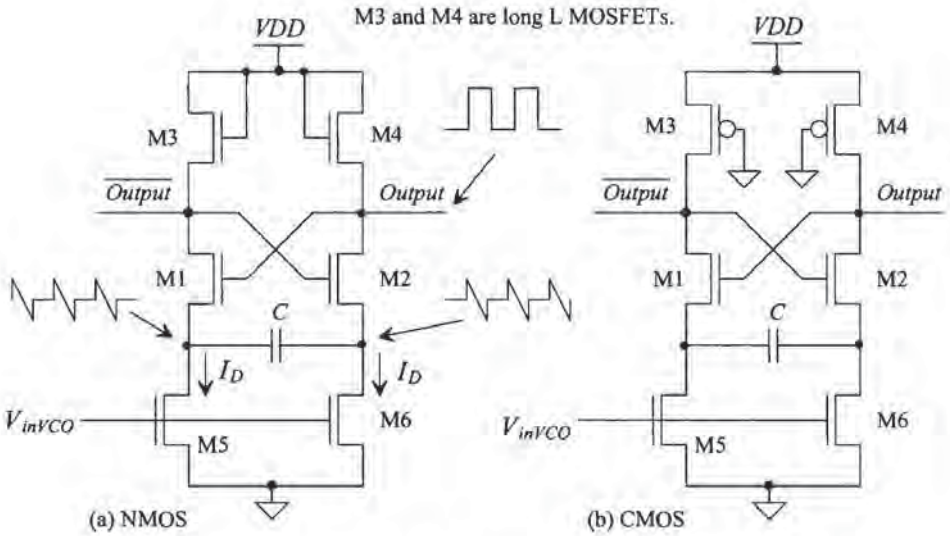



**Figure 19.19** Source coupled voltage-controlled oscillators (also known as source coupled multivibrators).

For the circuit in Fig. 19.19a, MOSFETs M5 and M6 behave as constant-current sources sinking a current $I_D$. MOSFETs M1 and M2 operate as switches. If M1 is off and M2 is on, the drain of M1 is pulled to $VDD - V_{THN}$ by M3. Since the gate of M2 is at $VDD - V_{THN}$, the source and drain (the *Output*) of M2 are approximately $VDD - 2 \cdot V_{THN}$. This is the minimum output voltage. The output voltage swing is limited to $V_{THN}$. A simplified schematic shown in Fig. 19.20 with M1 off and M2 on is helpful in determining the oscillator frequency. The *Output* gate of M1 is approximately $VDD - 2 \cdot V_{THN}$ and is held at this voltage through M2 until M1 turns on and M2 turns off. Initially, at the moment when M1 turns off and M2 turns on, point X is $VDD - V_{THN}$. The current through $C$, $I_D$, causes point X to discharge down toward ground. When point X gets down to $VDD - 3 \cdot V_{THN}$, M1 turns on and M2 turns off. In other words, the voltage at point X changed a

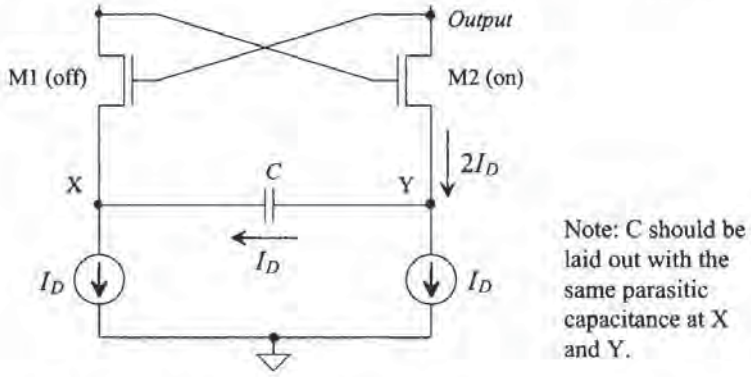



**Figure 19.20** Simplified schematic of source coupled oscillator, M1 is on and M2 is off.

IPR2023-00697
Theta EX2019

total of $2 \cdot V_{THN}$ before switching took place. The time it takes point X to change $2 \cdot V_{THN}$ is given by

$$\Delta t = C \cdot \frac{2 \cdot V_{THN}}{I_D} \tag{19.28}$$

Since the circuit is symmetrical, two of these discharge times are needed for each cycle of the oscillation. The frequency of oscillation is given by

$$f_{osc} = \frac{1}{2\Delta t} = \frac{I_D}{4 \cdot C \cdot V_{THN}} \tag{19.29}$$

The waveforms at the points X, Y, and *Output* are shown in Fig. 19.21 for continuous time operation.

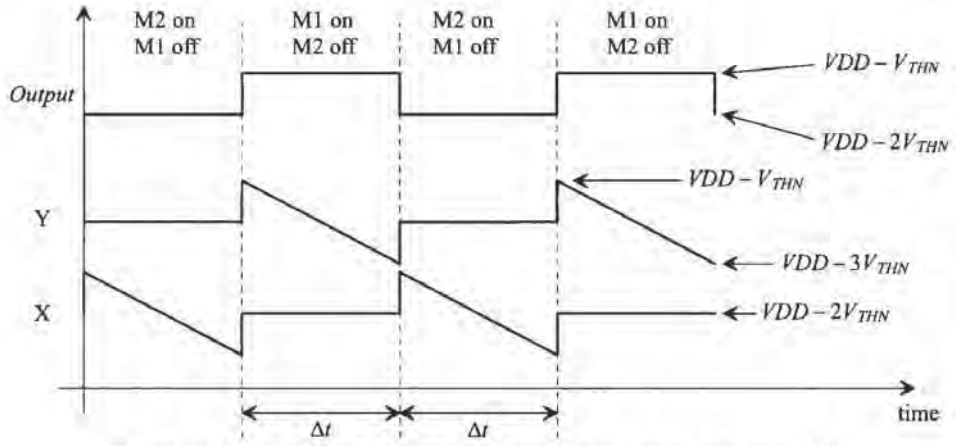



**Figure 19.21** Voltage waveforms for the NMOS source-coupled VCO.

## 19.3 The Loop Filter

The loop filter is the brain of the DPLL. In this section, we discuss how to select the loop-filter values in order to keep the DPLL from oscillating (i.e., keep the $V_{inVCO}$ voltage from oscillating, causing the frequency out of the VCO to wander). If the loop-filter values are not selected correctly, it may take the loop too long to lock, or once locked, small variations in the input data may cause the loop to unlock.

In the following discussion, we will be concerned with the *pull-in range* and the *lock range*. The pull-in range, $\pm\Delta\omega_P$, is defined as the range of input frequencies that the DPLL will lock to. The time it takes the loop to lock is labeled $T_p$ and can be a very long time. If the center frequency of the DPLL is 10 MHz and the pull-in range is 1 MHz, the DPLL will lock on an input frequency from 9 to 11 MHz in a time $T_p$ (assuming $N = 1$). The lock range, $\pm\omega_L$, is the range of frequencies in which the DPLL locks within one single beat note between the divided down output (*dclock*) and input (*data*) of the DPLL. *The operating frequency of the DPLL should be limited to the lock range for normal operation.* Once the DPLL is locked, it will remain locked as long as abrupt frequency changes, $\Delta\omega$, in the input frequency (input frequency steps) over a time interval $t$ are much smaller than the natural frequency of the system squared, that is, $\Delta\omega/t < \omega_n^2$.

IPR2023-00697
Theta EX2019

### 19.3.1 XOR DPLL

Consider the block diagram of the DPLL using the XOR DPLL shown in Fig. 19.22. The phase transfer function (neglecting the static or DC behavior) is given by

$$H(s) = \frac{\phi_{clock}}{\phi_{data}} = \frac{K_{PD}K_F K_{VCO}}{s + \beta \cdot K_{PD}K_F K_{VCO}} \tag{19.30}$$

with $s = j\omega$ and the feedback factor, $\beta$, is

$$\beta = \frac{1}{N} \tag{19.31}$$

The transfer function of the loop filter is given by

$$K_F = \frac{1}{1 + j\omega RC} = \frac{1}{1 + sRC} \tag{19.32}$$

Substituting Eqs. (19.31) and (19.32) into Eq. 19.30 yields

$$H(s) = \frac{\phi_{clock}}{\phi_{data}} = \frac{K_{PD}K_{VCO} \cdot \frac{1}{1+sRC}}{s + \frac{1}{N} \cdot K_{PD}K_{VCO} \cdot \frac{1}{1+sRC}} \tag{19.33}$$

This is a second-order system. $H(s)$ can be rewritten as

$$H(s) = \frac{\frac{K_{PD}K_{VCO}}{RC}}{s^2 + \frac{s}{RC} + \frac{1}{N} \cdot \frac{K_{PD}K_{VCO}}{RC}} = \frac{f_{clock}}{f_{data}} = \frac{(K_{VCO}V_{inVCO})/2\pi + f_o}{f_{data}} \tag{19.34}$$

since $j\omega \cdot \phi = f$ . The natural frequency, $\omega_n$, of this system is given by

$$\omega_n = \sqrt{\frac{K_{PD}K_{VCO}}{N \cdot RC}} \tag{19.35}$$

and the damping ratio, $\zeta$, is given by

$$\zeta = \frac{1}{2RC\omega_n} = \frac{1}{2} \cdot \sqrt{\frac{N}{K_{PD}K_{VCO} \cdot RC}} \tag{19.36}$$

The pull-in range is given by

$$\Delta\omega_P = \frac{\pi}{2} \cdot \sqrt{2\zeta\omega_n K_{VCO}K_{PD} - \omega_n^2} \tag{19.37}$$

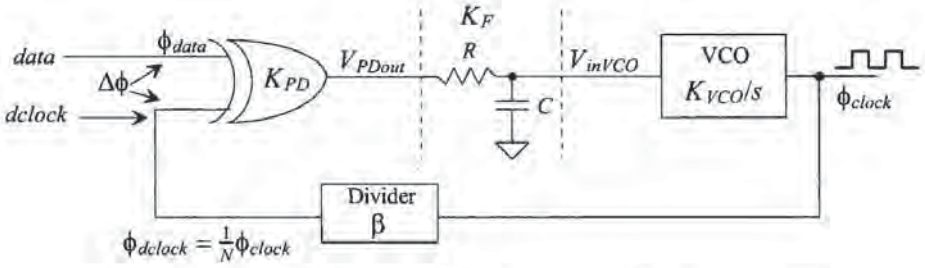



**Figure 19.22** Block diagram of a DPLL using a XOR phase detector

IPR2023-00697
Theta EX2019

while the pull-in time is given by

$$T_P = \frac{4}{\pi^2} \cdot \frac{\Delta\omega_{center}^2}{\zeta\omega_n^3} \tag{19.38}$$

The lock range of the loop is given by

$$\Delta\omega_L = \pi\zeta\omega_n = \frac{\pi}{2} \cdot \frac{1}{RC} \tag{19.39}$$

while the lock time is

$$T_L = \frac{2\pi}{\omega_n} \tag{19.40}$$

Probably the best way to understand how these equations affect the performance of a DPLL is through an example.

**Example 19.2**

Design and simulate the operation of a DPLL with the topology seen in Fig. 19.22 using the VCO from Fig. 19.18. Assume that the input data rate is 100 Mbits/s and has a format as in Fig. 19.7 (so the DPLL's output is a 100 MHz square wave).

Let's start off by drawing the schematic of the XOR phase detector, Fig. 19.23, and the divide by two circuit, Fig. 19.24. Notice that the XOR gate won't have zero output resistance, so the value of $R$ used in the loop filter may need to be modified accordingly. The divide by two circuit is implemented using true single phase logic.

To calculate the values for the loop filter, let's write the gain of the VCO in Fig. 19.18 as

$$K_{VCO} = 1.57 \times 10^9 \; radians/V \cdot s$$

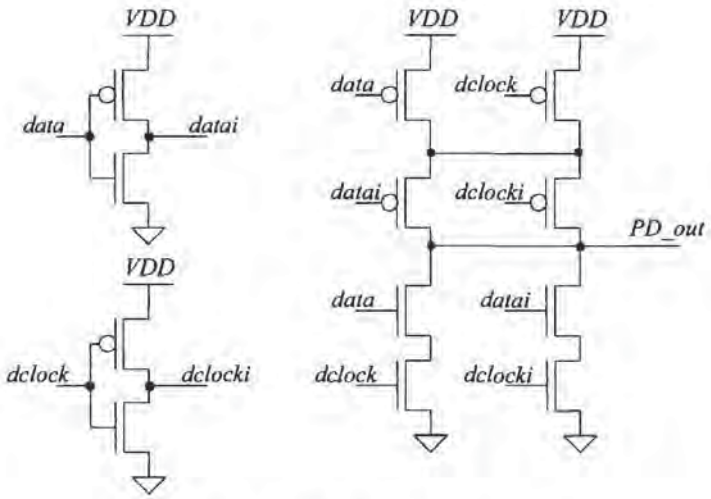



**Figure 19.23** XOR phase detector.

IPR2023-00697
Theta EX2019

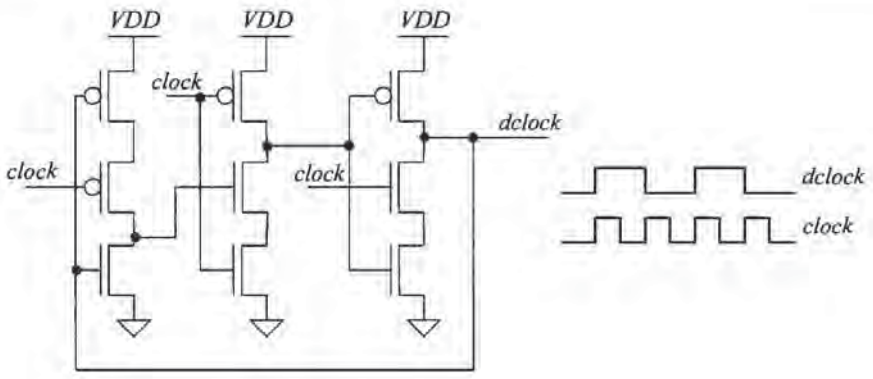



**Figure 19.24** A divide by two circuit using true single phase clocking (TSPC) logic.

and the gain of the XOR phase detector

$$K_{PD} = \frac{VDD}{\pi} = \frac{1}{\pi} \text{ (V/radians)}$$

Let's begin by setting the damping ratio $\zeta$ to 1. Using Eq. (19.36) results in

$$1 = \frac{1}{2} \cdot \sqrt{\frac{2}{\frac{1}{\pi} \cdot (1.57 \times 10^9) RC}} \rightarrow RC = 500\ ps$$

clearly not practical! We must reduce the gain of the VCO. This means the VCO output frequency range must be reduced. The problem with reducing the output frequency range is that it becomes very difficult to fabricate the VCO in CMOS at the precise center oscillation frequency. Nonetheless, let's modify the VCO from Fig. 19.18 so that the output frequency range is limited.

Consider the schematic seen in Fig. 19.25 of the current generator portion of the VCO (see Fig. 19.17). Here, we've added a resistor to set the lower frequency range of the VCO (which makes the VCO very dependent on the power supply voltage). The range is still set by M5R. However, now we've increased the value of the resistor to limit the VCO's output frequency range. The resulting VCO gain, with the values seen in Fig. 19.25, appears in Fig. 19.26. Using this modified VCO gain in the above equation gives

$$RC = 5 \text{ ns}$$

The natural frequency of the second-order system is, from Eq. (19.36),

$$\omega_n = \frac{1}{2 \cdot RC \cdot \zeta} = 100 \times 10^6 \text{ radians/s}$$

The lock range is

$$\Delta\omega_L = \pi\zeta\omega_n = 314 \times 10^6 \text{ radians/s or } \Delta f_L = 50\ MHz$$

which is outside the VCO operating range. Therefore, this DPLL will lock over the entire VCO operating range, that is, from 95 to 105 Mbits/s. We do not need

IPR2023-00697
Theta EX2019