CMOS

Circuit Design, Layout, and Simulation

Second Edition

R. Jacob Baker

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

IEEE Solid-State Circuits Society, *Sponsor*

IEEE
IEEE Press

WILEY-INTERSCIENCE
A JOHN WILEY & SONS, INC., PUBLICATION

0800087554
TK7871.99
M44B168
2005

Copyright © 2005 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

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) 646-8600, 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.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representation 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 please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. 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, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data is available.

ISBN 0-471-70055-X

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

IEEE Press
445 Hoes Lane
Piscataway, NJ 08855

IEEE Press Editorial Board
Stamatios V. Kartalopoulos, *Editor in Chief*

M. Akay	M. E. El-Hawary	F. M. B. Periera
J. B. Andersaon	R. Leonardi	C. Singh
R. J. Baker	M. Montrose	S. Tewksbury
J. E. Brewer	M. S. Newman	G. Zobrist

Kenneth Moore, *Director of Book and Information Services (BIS)*
Catherine Faduska, *Senior Acquisitions Editor*
John Griffin, *Senior Editor*
Anthony VenGraitis, *Project Editor*

IEEE Solid-State Circuits Society, *Sponsor*
SSC-S Liaison to IEEE Press, Stuart K. Tewksbury

Books of Related Interest from IEEE Press

CMOS: Mixed Signal Circuit Design
R. Jacob Baker
2002 Hardcover 532pp 0-471-22754-4

DRAM Circuit Design: A Tutorial
Brent Keeth and R. Jacob Baker
2001 Hardcover 200pp 0-780-36014-I

CMOS Electronics: How It Works, How it Fails
Jaume Segura and Charles F. Hawkins
2004 Hardcover 348pp 0-471-47669-2

Silicon Germanium: Technology, Modeling, and Design
Raminderpal Singh, David L. Harame, and Modest M. Oprysko
2004 Hardcover 340pp 0-471-44653-X

Design of High-Performance Microprocessor Circuits
Anantha Chandrakasan, William J. Bowhill, and Frank Fox
2001 Hardcover 557pp 0-780-36001-X

Low-Power CMOS Design
Anantha Chandrakasan and Robert Brodersen
1998 Hardcover 644pp 0-780-36001-X

To 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	931
Chapter 29 Data Converter Architectures	965

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 Using the LASI Program 6
1.2.1 The Basics of LASI 6
Starting LASI 6
Getting Help 7
Cells in LASI I 7
Navigating 8
Drawing a Box 8
Moving and Resizing 9
The Drawing Grids 10
Making Measurements 10
Key Assignments 11
Cells in LASI II 11
Moving a Cell 14
Viewing or Editing Specific Layers 14
The Polygon and Path Shapes 15
Using Text in LASI 16

Some Features to Speed Up Layout Design 17
Understanding the Cpy and Copy Commands 18
Transporting Cells in LASI 19
Edit-in-Place 19
Backing Up Your Work 20
1.2.2 Common Problems 20
After adding an object, the object cannot be seen 20
Cannot Get an object 20
Cells are drawn as outlines, or the perimeter of a cell has a dashed line 21
Fit command causes the drawing window to expand much larger than the current cell 21
Cursor movement is not smooth 21
1.2.3 Sending the Layout to the Mask Maker 21
Checking to Make Sure the Layout Scaled Correctly 21
1.3 An Introduction to WinSPICE 23
Generating a Netlist 23
Transient Analysis 24
Other Analysis 27
Convergence 27
Some Common Mistakes and Helpful Techniques 28
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 Using the LasiDrc Program 52
Design Rules for the Well 53
Running a Design Rule Check 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 LASI Layout Examples 75
3.4.1 Laying Out the Pad II 75
3.4.2 Introduction to LasiCkt 78
Drawing a Schematic 78
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
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 LASI Design Examples 124
A Resistor Divider 125
Layout of a MOSFET 127

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
Level 1 Model Parameters Related to V_{THN} 145
Long-Channel MOSFET Models 146
Level 1 Model Parameters Related to Transconductance 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
Lightly Doped Drain (LDD) 151
6.5.1 MOSFET Scaling 152
6.5.2 Short-Channel Effects 153
Hot Carriers 153
Oxide Breakdown 154
Drain-Induced Barrier Lowering 154
Substrate Current-Induced Body Effect 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
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
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 $V_{onoise,RMS}$ om 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
Burst (Popcorn) Noise 252
Excess Noise (Flicker Noise) 253
Avalanche Noise 253
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
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
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
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
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
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
18.4.3 Example 546
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
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
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
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
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
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
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
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
Increasing the Widths of the Diff-Pair 903
A Final Note Concerning Biasing 903
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 931
28.1 Analog Versus Discrete Time Signals 931
28.2 Converting Analog Signals to Digital Signals 932
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 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
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
Index **1023**
About the Author **1039**

Preface

For the last 25 years, CMOS (complementary metal oxide semiconductor) technology has been the dominant technology for fabricating integrated circuits (ICs or chips). For the next 25 years, CMOS will, no doubt, continue to be the dominant integrated circuit technology. 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 transistors on a chip will double every 18 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 (it's getting cut in half every 18 months). 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, I take a two-path approach to custom CMOS integrated circuit design. Design techniques are developed for both and then compared. This comparison gives readers deep insight into the circuit design process.

As an academic text, the book is filled with practical design examples, discussions, and problems. The assumed background of the reader is a knowledge of linear circuits, microelectronics, and digital logic design. The solutions to the end-of-chapter problems (for self-study) and the netlists used when simulating the circuits are found at **http://cmosedu.com**. Those interested in gaining an in-depth knowledge of CMOS analog and digital design will be greatly aided by downloading, modifying, and simulating the netlists of the book's circuits.

Several courses can be taught using this book including VLSI or physical IC design (chapters 1–7 and 10–15), analog IC design (chapters 8–9 and 20–24), memory circuit design (16–19), and advanced analog IC design (25–29). A graduate course in mixed-signal design following the advanced analog IC design course can be taught using the second volume of this book entitled *CMOS Mixed-Signal Circuit Design*.

In this edition, LASI CAD software is used once again to introduce circuit layout and WinSPICE is used for the circuit simulations. The material covering the LASI layout tool can be skipped if other layout software packages are used. I like using LASI in a first course on IC design because it takes very little class time to learn. Further, other SPICE programs can be used to simulate the circuit netlists with little or no modifications. Both the LASI CAD software and the WinSPICE circuit simulator run under Windows and can be downloaded at http://cmosedu.com.

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

I've put a great deal of effort 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 of my students 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) have been added to this edition. I've also tried to anticipate future design paradigms. For example, the addition of DSM has been applied to CMOS image sensors, Flash memory, and memory using thin oxides (direct tunneling). Sensing using DSM was included because it makes use of the fact that as CMOS clock speeds are going up, the gain and matching of transistors is deteriorating. Further, I've tried to anticipate 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, I've tried to provide 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. I've focused the discussions and examples found in this book 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.

Acknowledgments

I would like to acknowledge the support of Micron Technology, Inc., and, in particular, Mary Miller for her assistance with the technical writing review. Further, I would like to thank the reviewers, students, colleagues, and friends that have helped to make this book a possibility: Rupa Balan, David M Binkley, Bill Black, Dave Boyce, Elizabeth Brauer, J. W. Bruce, Kloy Debban, Ahmad Dowlatabadi, Kevin Duesman, Krishna Duvvada, Surendranath Eruvuru, Cathy Faduska, Paul Furth, Neil Goldsman, Tyler Gomm, Kory Hall, Wes Hansford, David Harris, Jeff Jessing, Brent Keeth, Howard Kirsch, Bill Knowlton, Bhavana Kollimarla, Harry W. Li, Matthew Leslie, Song Liu, Amy Moll, Sugato Mukherjee, Ward Parkinson, Terry Sculley, Brian Shirley, Harish Singidi, Mike Smith, Mark Tuttle, Vance Tyree, Gary VanAckern, Indira Vemula, Tony VenGraitis, and Joseph J. Walsh.

R. Jacob (Jake) Baker

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 discuss setting up the LASI (layout system for individuals) layout software for Windows and provide a very brief introduction to SPICE (simulation program with integrated circuit emphasis).

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 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.

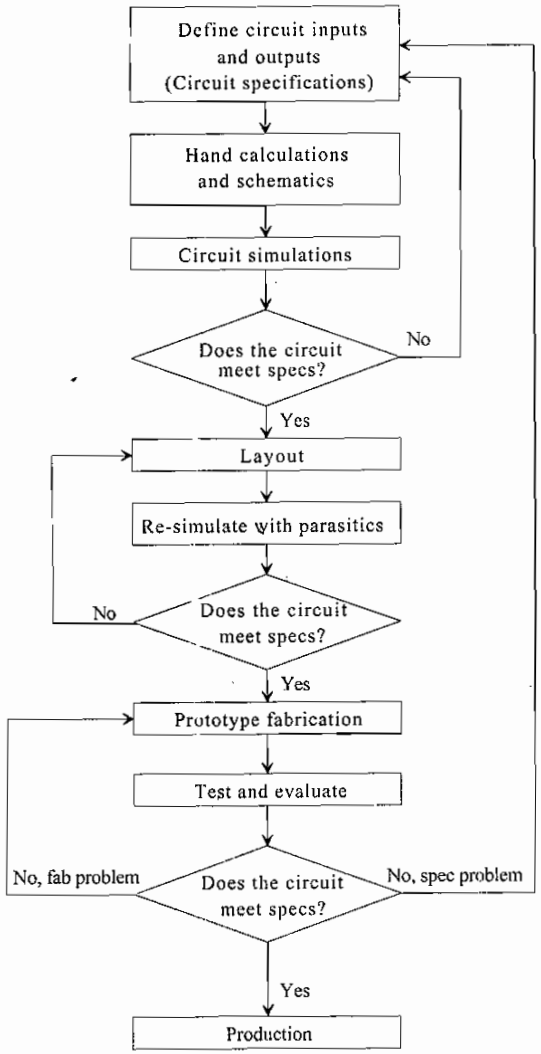


Figure 1.1 Flowchart for the CMOS IC design process.

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. In Fig. 1.2, an individual die has been outlined with a black marker and appears next to a dime. Added to the wafer are test structures and process monitor plugs (sections of the wafer used to monitor process parameters).

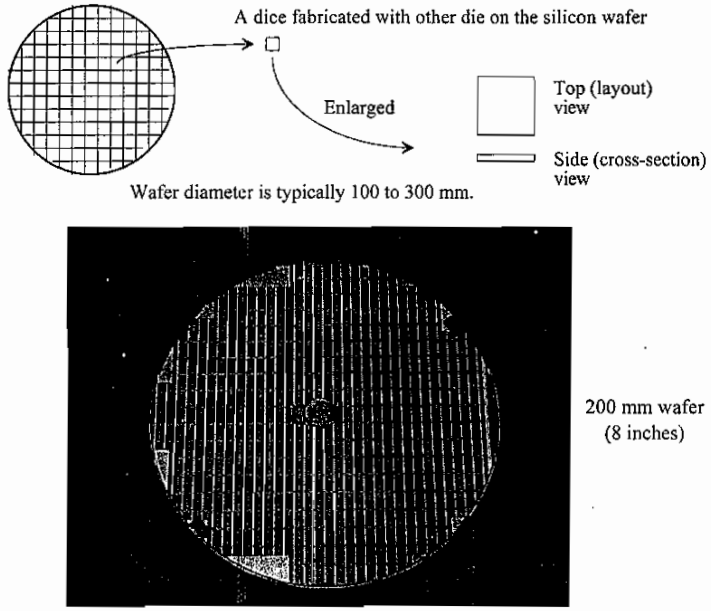


Figure 1.2 CMOS integrated circuits are fabricated on and in a silicon wafer.

The ICs we design and lay out using a layout program can be fabricated through MOSIS (http://mosis.org) 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.

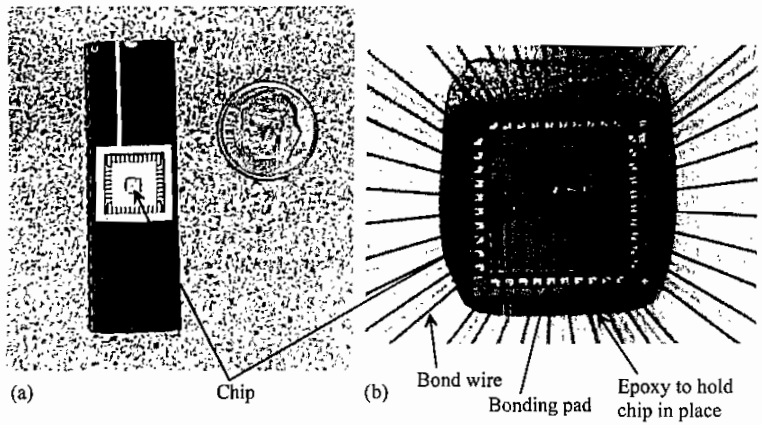


Figure 1.3 How a chip is packaged (a) and (b) a closer view.

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).

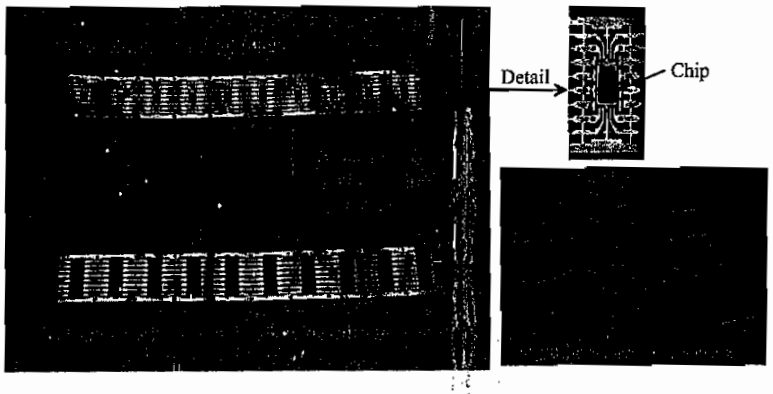


Figure 1.4 Plastic packages are used (generally) when the chip is mass produced.

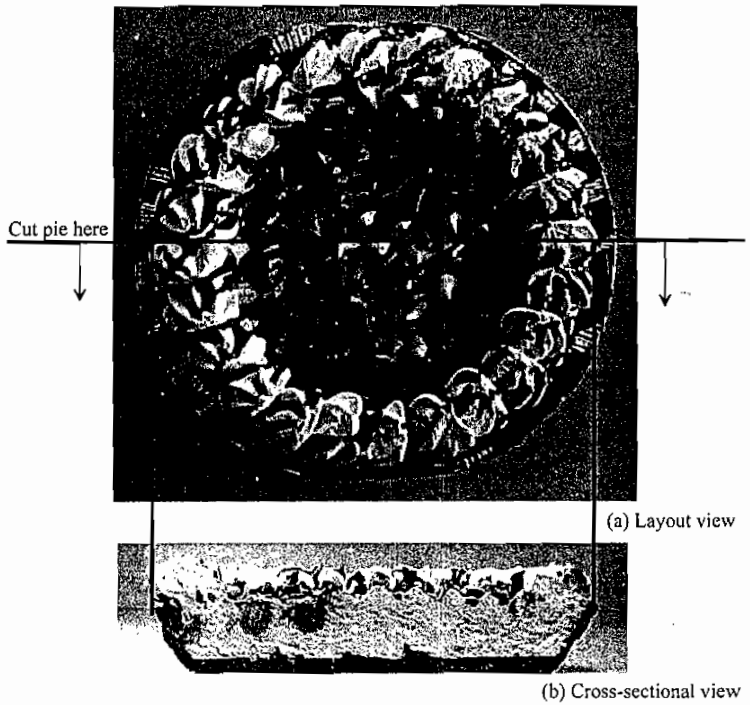


Figure 1.5 Layout and cross sectional view of a pie (minus pie tin).

1.2 Using the LASI Program

The LASI program discussed in this section can be used for the layout and design of CMOS integrated circuits. This section provides a brief tutorial covering the use of LASI. We assume that the reader has downloaded and installed LASI and the MOSIS setups following the instructions at cmosedu.com. We use the MOSIS setups (with a design directory of, for example, C:\Lasi7\Mosis) for our examples in this section.

1.2.1 The Basics of LASI

A chip design should reside in a design directory. The LASI system executables are located in C:\Lasi7 (which *cannot* be used as a design directory). Information on setting up a design directory can be found at cmosedu.com.

Starting LASI

Going to the Window's start button, then Lasi 7, and then MOSIS results in LASI starting, Fig. 1.6. Notice how, at the top of the window in Fig. 1.6, the design directory (Folder) is shown. The menu items (**Help**, **Run**, **Print**, etc.) are executable by either clicking on the word (e.g., **Help**) or on the menu button (e.g., the picture of a question mark). The buttons on the right of the display can be toggled by either pressing the **Menu1** or **Menu2** buttons or by clicking the right mouse button while in the drawing area. We'll talk about the items on the bottom of the screen in a moment.

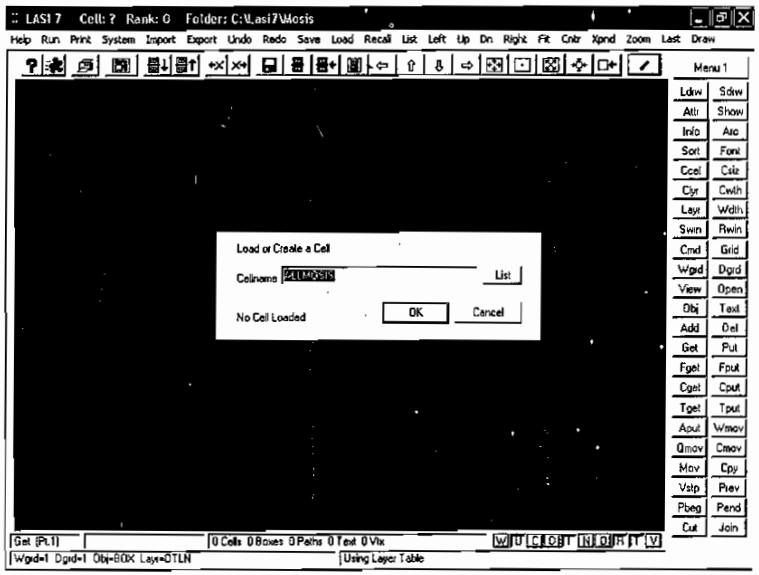


Figure 1.6 Starting LASI using the MOSIS setups.

Getting Help

The LASI layout system comes with a complete on-line manual. This manual is accessible, while LASI is running, by pressing F1 on the keyboard and clicking the command button (simultaneously) with which the user needs help or by pressing the **Help** button. Alternatively, the user can open the help file in the directory C:\Lasi7\help without LASI running.

Cells in LASI I

Complex IC designs are made from simpler objects called cells. A cell might be a logic gate or an op-amp. When LASI starts-up, as seen in Fig. 1.6, the last cell loaded (prior to exiting LASI) appears in the load cell window. In Fig. 1.6, this cell's name is "*ALLMOSIS*." Let's create a new cell called "*test*" with a rank of 1. Figure 1.7 shows the resulting LASI screen. Notice that the current cell's name and rank are displayed at the top of the window. If we want to load or create a different cell, we press either the **Load** menu item or the corresponding picture, as indicated in Fig. 1.7. The drawing area in Fig. 1.7 shows a reference indicator (the origin of the drawing). To toggle displaying this indicator, we can either press **r** on the keyboard or the **R** button in the lower right corner of the display followed by redrawing the cell. To redraw the cell, we press **Enter** on the keyboard or **Draw** or the picture of the pencil at the top of the window.

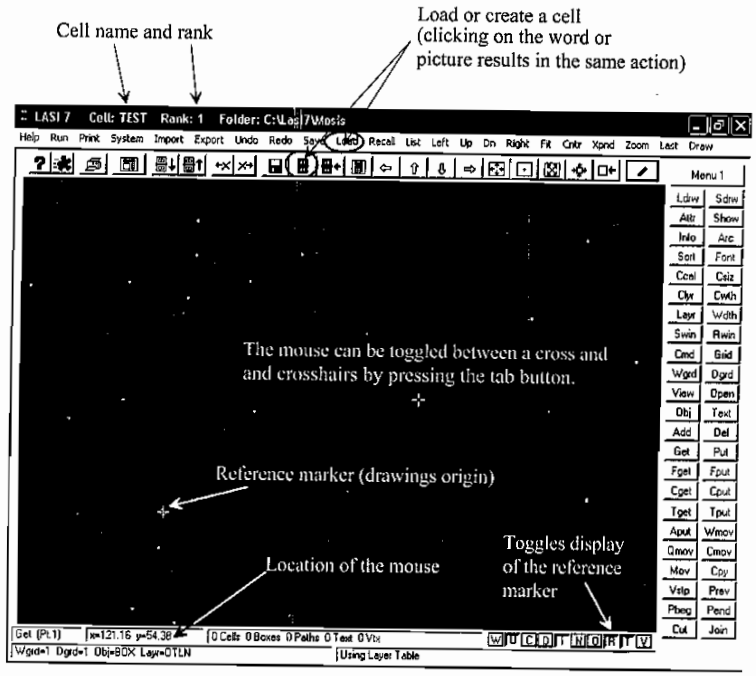


Figure 1.7 Screen after making a cell called "test" with a rank of 1.

Navigating

Pressing the **Grid** button on the right of the screen followed by zooming in around the reference indictor by pressing **Zoom** (and two clicks of the mouse to indicate the zoom area) on the top menu may result in a screen like the one seen in Fig. 1.8. The reader should experiment with the **Fit** (alt+f), **Xpnd** (alt+x or expand), and the arrow (**Left**, **Up**, **Dn**, and **Right**) commands. The center command, **Cntr**, centers the screen around a point set by the mouse. Pressing **Last** on the top menu brings up the previous (or last) view.

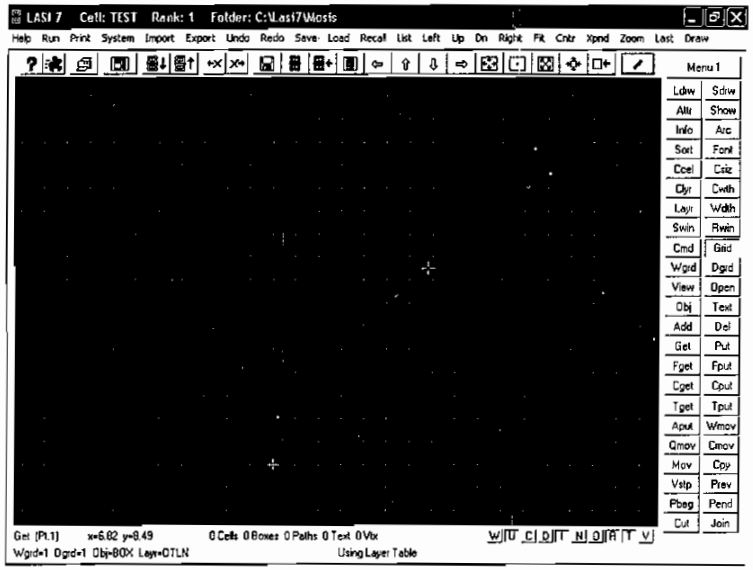


Figure 1.8 Navigating in LASI (see text).

Drawing a Box

Let's draw a box. To start, select the **Layr** command and select "NWEL" or layer 42. This is the n-well layer discussed in the next chapter. Remember that if a command isn't showing, toggle the menus by right clicking the mouse in the drawing area or pressing the buttons **Menu 1** or **Menu 2**. Next select the **Obj** (object) command followed by selecting BOX (double click on the word BOX). The lower left of the display should indicate a working grid (Wgrd) of 1, a dot grid (Dgrd) of 1, a box object, and the n-well layer. Next click on the **Add** button (we want to add a box on the n-well layer). The lower left of the display should indicate that LASI is in the "Add Box" mode. By clicking the left mouse once in the drawing area, moving the mouse, and clicking the left mouse button again, we draw a box on the n-well layer. Figure 1.9 shows a possible resulting display.

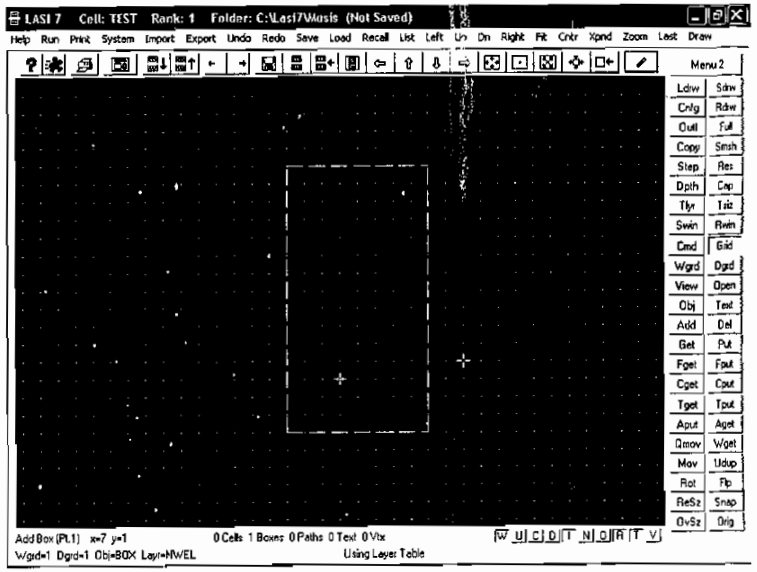


Figure 1.9 Drawing a box using LASI.

Moving and Resizing

Moving or resizing an object consists of *getting* the object using, among others, the **Get** command, *moving* the object using the **Mov** command, and *putting* (or deselecting) the object using the **Put** command. For example, say we want to move the right edge of the n-well box in Fig. 1.9. To begin, select the **Get** command. This is followed by clicking the left button of the mouse on one side of the edge followed by clicking (the mouse) on the other side of the edge (the selected or "Got" edge then becomes highlighted). Using the **Mov** command, we can then use the mouse in the same manner to move the edge. To deselect the edge, we use the **Put** command (again, in the same manner with the two clicks of the left mouse button) or we simply press the all put (**Aput**) command. Note that movement is relative to the clicking of the mouse (we don't have to click on the selected edge). If, for example, the mouse is left-clicked somewhere in the drawing window and then, in a horizontal distance of 5, left-clicked again, the selected edge will move horizontally a distance of 5. (The user should experiment with these commands until they feel comfortable.) To move the entire box, we get all four sides of the box or any part of the box using the **Fget** (full get) command.

To reduce the amount of mouse clicking it is helpful to use the **Qmov** (quick move) command. This command automatically performs the get, move, and put commands in sequence so the user doesn't have to keep clicking menu items. The reader should try out the **Qmov** command in LASI. **Qmov** can save considerable time when making edits to layouts.

The Drawing Grids

The position of the cursor in the drawing window is continuously read out at the bottom of the screen. The value of the working grid (what the cursor snaps to) and the dot grid (the dots we see when the **Grid** button is depressed) are also seen at the bottom of the window. The coordinates of the cursor are either in *working grid* units or in the smallest possible grid unit, the *unit grid* (so that the cursor appears to move smoothly instead of jumping around). For the MOSIS setups, there are 100 grid units between each working grid point when the working grid is 1. By pressing the **Wgrd** and **Dgrd** buttons, the respective working or dot grids are changed between one of ten possible values. These values are set using the **Cnfg** command. (Again, if a command isn't showing, toggle the menus by right clicking in the drawing window.) Figure 1.10 shows the **Cnfg** (configure drawing parameters) window. In the MOSIS setups, we've limited the grids to either 1 or 0.5. Until the user gets familiar with layout, it's not a good idea to change these values. Note that it's possible to have a dot grid of 1 while the working grid (the grid the cursor snaps to) is 0.5.

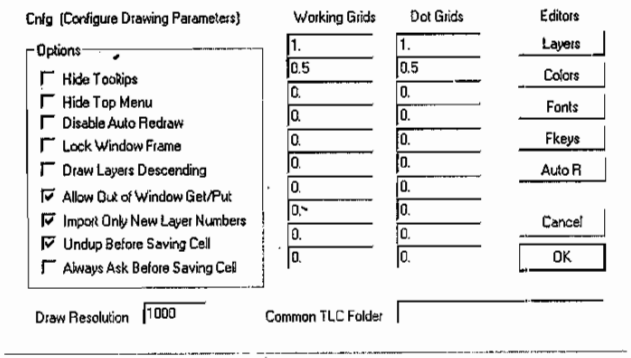


Figure 1.10 Showing how the working and dot grids are set in LASI using the **Cnfg** command.

Keyboard button **a** is a toggle between the working and unit grids. Pressing keyboard button **w** causes the cursor to snap to the working grid while pressing **u** causes the cursor to snap to the unit grid (i.e., appear to move in a continuous movement).

Making Measurements

LASI makes measurements using the keyboard buttons, **z** and **spacebar**. Pressing the **z** button sets a zero reference point. Pressing the **spacebar** displays the distance between the cursor and the zero reference point in the middle bottom of the window. It's useful to remember at this point is that pressing **w** forces the cursor to snap to the working grid so that measurements can be made based on the grid and independent of the state (the current selected command) of LASI. Note that if the **spacebar** is held down, a continuous measurement is displayed on the bottom of the window.

Key Assignments

Below is a summary of the keyboard (button) commands.

Enter does a **Draw** command (redraws the display)
Delete does a **Del** command (deletes the selected objects)
Insert does a **Xpnd** command (expands the current view)
Home does a **Cntr** 0 0 command (centers the display on the origin [reference marker])
End does a **Last** command (switches back to the previous displayed view)
pgup pans up by a full screen
pgdn pans down by a full screen
arrows pan the drawing window in the direction of the pressed arrow
Tab toggles the mouse cursor between a small cross and crosshairs
z sets a measurement zero point
spacebar gives a measurement from the zero point

The following keyboard buttons are also executed by pressing on the button in the lower right portion of the display

w forces the cursor to the working grid
u forces the cursor to the unit grid
c toggles the path center line on and off in a path object
d toggles the distance marker on and off
i toggles the cell image on and off (a dotted line around the silhouette of the cell)
n toggles the outline name on and off
o toggles the octagonal cursor mode on and off
r toggles the 0,0 reference mark on and off
t toggles the text reference point on and off
v toggles viewing vertices of a path or polygon object

The **alt** button can be used with the indicated letter in the top menu commands for execution directly from the keyboard. For example, using **alt+u** (pressing **alt** followed by **u** or pressing both at the same time) results in executing an **Undo** command. Other multipurpose buttons include the **Ctrl**, **Esc**, and **Shift** buttons (see the LASI help manual for additional information).

Finally note that pressing **Esc** (the escape key) aborts a command including a **Draw** command. Using this command can be useful when the layout gets complicated and large. Pressing escape stops the redrawing and shows the cells as simple outlines.

Cells in LASI II

Let's take the *test* cell in Fig. 1.9 and add it to a cell called *test2*. To begin, we press the **Load** command button and create a cell called *test2*. LASI will ask if we want to save the current cell, *test*, (select yes). Next LASI will ask what rank we want to assign the *test2* cell. Select a rank of 2. A new cell will be created and the drawing window will become blank.

The rank is used in the cell collection's (seen by pressing **List**) hierarchy. We might rank a *MOSFET* cell with 1, an *inverter* cell with 2, and a *counter* cell with 5. The *MOSFET* cell can be placed in the *inverter* or the *counter* but the *inverter* or *counter* can't be placed in the *MOSFET*. Similarly, we can place the *inverter* (2) in the *counter* (5) but we can't place the *counter* in the *inverter*. Note that to organize the cell collection we go to the system menu (by pressing **System**) and then press **Organize**.

We can add a box, as we did before, or we can add rank 1 cells to the rank 2 cell. Let's add our *test* cell seen in Fig. 1.9 to this *test2* cell. Press the **Add** command button. Next select the **Obj** command button and double click on the cell *test*. Moving the mouse cursor into the drawing display shows something similar to what's seen in Fig. 1.11. Notice the small outline of the *test* cell. Before adding the cell (by clicking the left mouse button in the drawing window), let's use the zoom command to zoom in around the reference indicator. Adding several *test* cells to the *test2* cell may look something like what is seen in Fig. 1.12.

We can go back to the *test* cell by pressing **List** (saving the *test2* cell) and double clicking on the cell *test*. Press **alt+f** (or **Fit**) at the top of the menu to fit the cell's contents to the drawing window. Let's add another box on the n-well layer (select **Add**, then **Obj**, double clicking on BOX, and then **Layr** followed by selecting the layer NWEL). Adding another n-well box to our layout may result in a layout similar to what appears in Fig. 1.13.

Going back to the *test2* cell, Fig. 1.14, we see that the changes in the layout are automatically updated in the higher ranking cell. If, for example, a change was made to the layout of an inverter, then going to the lower ranking inverter cell and making the change (once) will automatically update the layout wherever the inverter cell is used.

Notice, in Fig. 1.14, that there is a dotted line surrounding the shape of the *test* cell. These dotted lines are the image of the cell. In addition, there are small diamonds in each of the added *test* cells. These diamonds indicate where the reference indicator is in

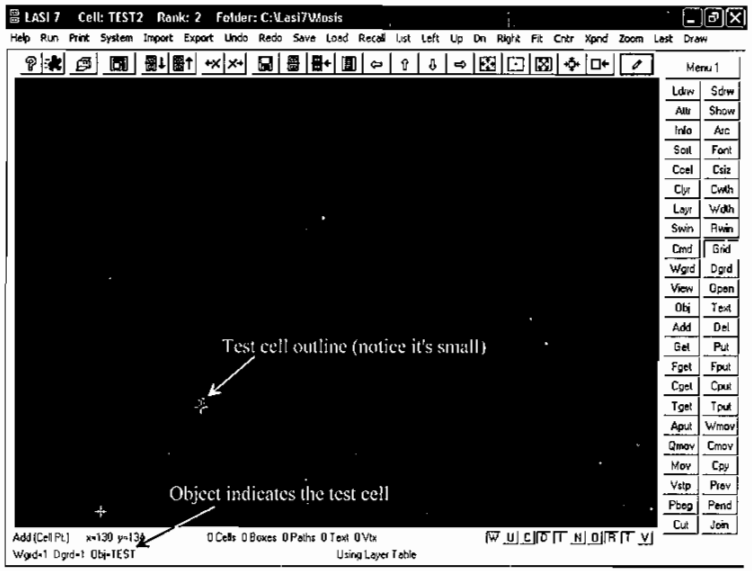


Figure 1.11 Adding the test cell with a rank of 1 to the test2 cell with a rank of 2.

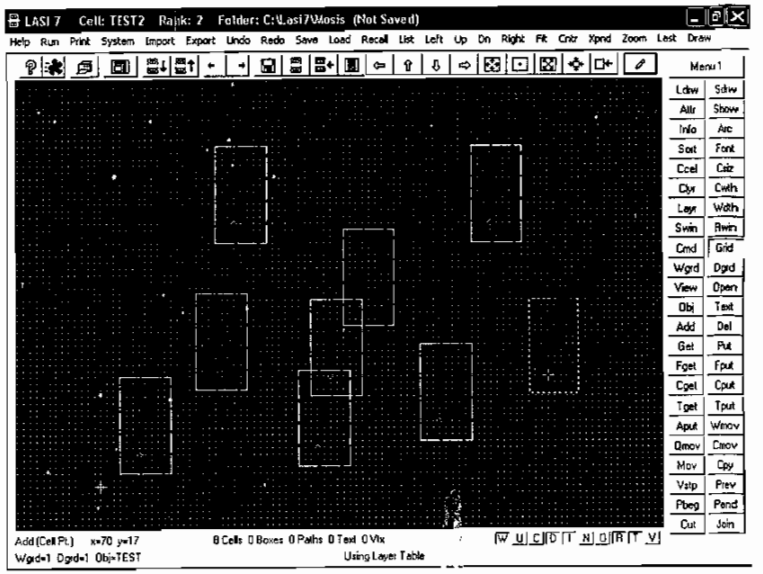


Figure 1.12 Adding several "test" cells with a rank of 1 to the "test2" cell with a rank of 2.

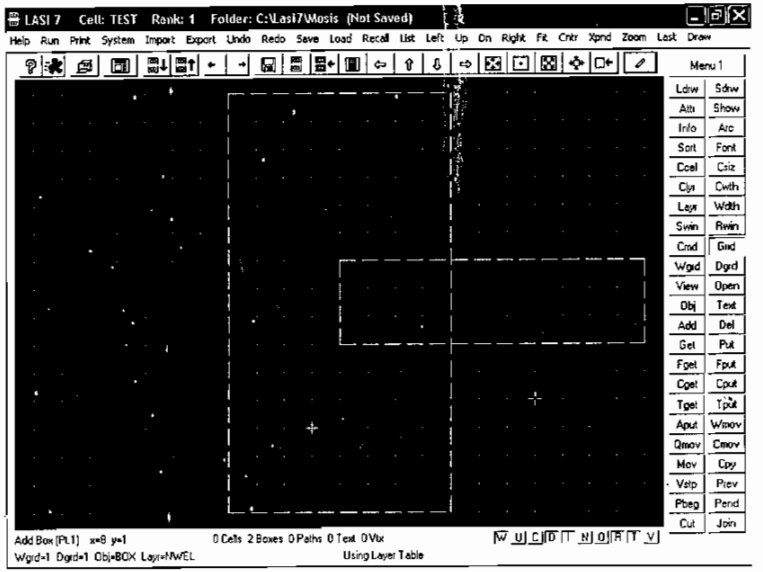


Figure 1.13 Going back to the "test" cell and adding a additional n-well box.

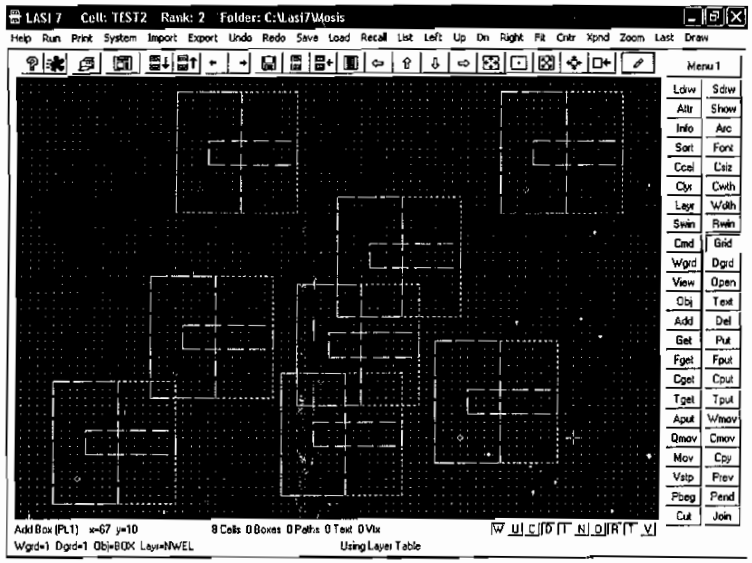


Figure 1.14 How the change to the "test" cell propagates up through the hierarchy.

the *test cell*. Moving or redefining the location of the reference indicator in the lower ranking cell, using the **Orig** command, also shifts the cells' locations in the higher-ranking cells where it is used. Pressing **i** on the keyboard or the **I** button at the bottom of the display toggles the cell's image on and off. (To see the change, the drawing window must be redrawn by pressing **·** on the keyboard or using the **Draw** command.)

Moving a Cell

To move a cell, the **Cget** (cell get) command is used first. Double clicking on the cell or using the mouse to surround a cell (or cells) "gets" the cell. This is followed by using the **Mov** command. Deselecting the cells can be accomplished using the **Cput** (cell put) command or, more often, using the **Aput** (all put) command.

Viewing or Editing Specific Layers

When layouts get complicated, the ability to look at or edit specific layers becomes important. However, someone learning layout can also feel frustrated by not knowing how the viewing and editing of specific layer commands operate. Below we summarize these commands.

Ldrw Draws only one layer of the layout. Useful to quickly view a single layer.

View Selects the layers LASI displays in the drawing window. This command can be frustrating. For example, suppose the NWEL layer is deselected in the view menu. If we were to draw a box on the NWEL layer and then redraw the layout, the box we just drew wouldn't show up!

Open Selects the layers LASI will allow the user to "get." This command, again, can result in frustration. If, for example, the NWEL layer is not selected in the open menu, then trying to "get" the layer will result in a waste of time.

The Polygon and Path Shapes

The only drawing shape we've discussed so far is the box. LASI can also be used to make polygons and path shapes. Let's start out by drawing a polygon (say a triangle). Create a new cell (using the **Load** command) called *test3* with a rank of 1. Select **Add**, then **Obj** (double clicking on PATH/POLY). Let's also select a different layer using the **Layr** command (select POL1 or layer 46). Next click the **Wdth** command and make sure that width is set to 0. When the width is zero, the object is a polygon. When the width is nonzero, the object is a path. Figure 1.15 shows the layout of a polygon (not closed). Notice the location of the polygon's vertices. To move (or change) the shape of a polygon, we must get a vertex. Using the **Get** command on a side of a polygon, and not encompassing a vertex, can result in frustration. To show the polygon's (or path's) vertices in the drawing display, press **v** on the keyboard (or the **V** in the lower right corner of the display) followed by **Enter**. (Try this now.) To close the polygon object in Fig. 1.15, we place the last (fourth) vertex on the first vertex as seen in Fig. 1.16. Note that this last vertex is still active and so it appears that we are still drawing the same polygon (even though our triangle is closed). To draw another shape, click the **Aput** command.

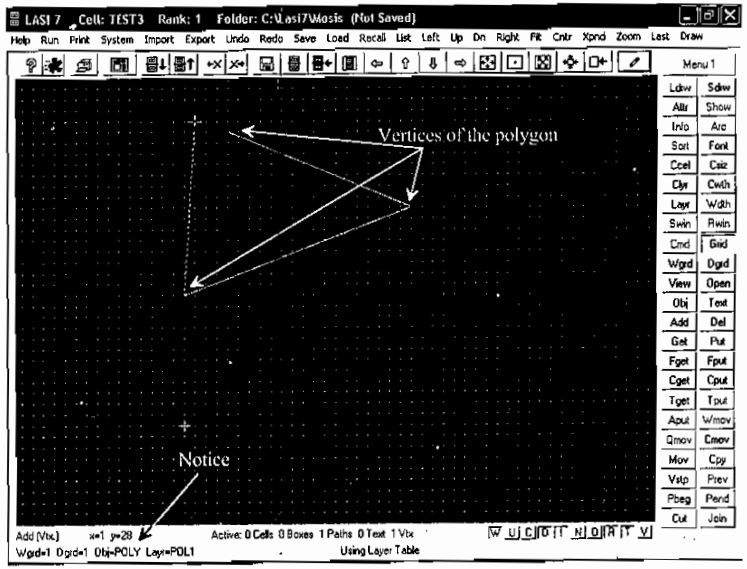


Figure 1.15 Drawing a polygon in LASI on the POL1 (polysilicon) layer.

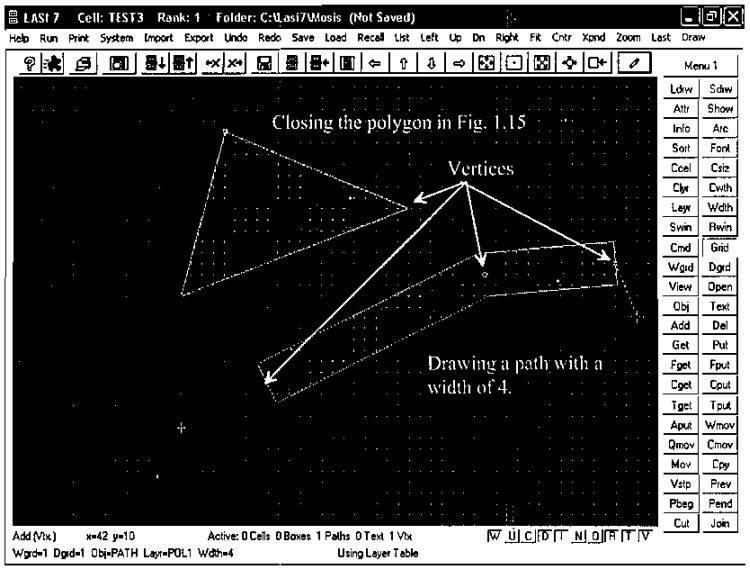


Figure 1.16 Closing the polygon in Fig. 1.15 and drawing a path object with a width of 4.

Also seen in Fig. 1.16 is a path object. To draw a path, after drawing the polygon shape, all we have to do is use the **Wdth** command to set the width of the path to a nonzero number (zero width indicates that we're drawing a polygon). For the path in Fig. 1.16, we used a width of 4. To terminate drawing a path, we can use the **Aput** command. Notice how the vertices of a path are located in the center of the path. Again, to toggle between displaying or not displaying vertices, we can press **v** on the keyboard. Also, again, if we don't encompass a vertex when we are trying to get the object, nothing will get selected. (Trying to get the side of a path or polygon can result in frustration.) To move an entire path or polygon, we can use the **Fget** command and encompass at least one of the path's or polygon's vertices.

Using Text in LASI

Labeling layout with text is important for documenting how the IC is assembled. To add text to a layout, we select the **Text** command. This is followed by selecting the text's layer using the **Tlyr** command (select POL1 for the example to follow). Next the size of the text can be set using **Tsiz**. The command **Font** (remembering if the command isn't showing to right-click the mouse in the drawing display) selects one of three font files. The first font (C:\LASI7\TFF1.DBD) file uses zero-width letters (so they don't have any fabrication significance). The other files (TFF2.DBD and TFF3.DBD) do have widths, and so they can be seen in the fabricated chips. Clicking the left mouse button in the drawing display may result in the text seen in Fig. 1.17 (where we've changed the size

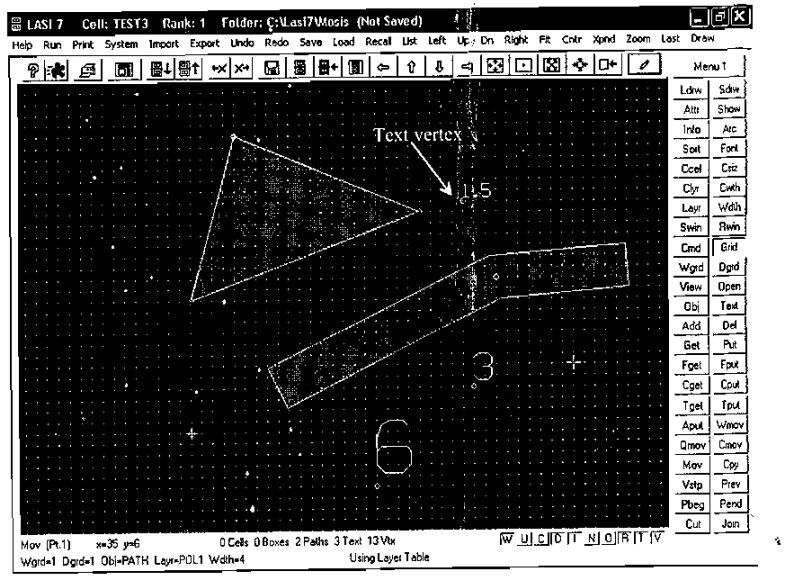


Figure 1.17 Adding text to a layout.

using **Tsiz**). To move the text, we must encompass a text vertex. To show the text vertices, press **t** on the keyboard or **T** in the lower right corner of the display followed by a **Draw** command. Again, not knowing to get a vertex can lead to frustration. To get text in a layout without getting any of the other objects, we can use the **Tget** command. To change the size of the text, we "get" the text by encompassing a vertex using the **Get**, **Fget**, or **Tget** commands, and then using the **Csiz** command. Using the **Text** command and clicking on an existing text vertex results in changing the displayed text but not the size or layer of the existing text. The **Clyr** command can be used to change the layer the text is drawn on or any other shape's layer.

Some Features to Speed Up Layout Design

The reader's right hand should be used for the mouse (assuming the reader is right-handed) while the left hand should be used for pressing keys on the keyboard. To assign more commands to the keyboard, press the **Cnfg** button followed by the **Fkeys** button. Figure 1.18 shows the result. The **F1** key is used for getting help, as discussed earlier. Now, as seen in Fig. 1.18, pressing **F2**, **F3**, and **F4** executes the commands **Aput**, **Qmov**, and **Fget**, respectively. It's important, when adding commands, to ensure that the changes are saved prior to exiting the window. To add a command to the list, click on one of the existing commands. Next type the new command in the command argument field at the bottom of the window. Pressing **Add** places the new command after the selected existing command, while pressing **Insert** places the command before the existing command. To edit an existing command, double click on it. After editing, press **Paste** to update the existing command. Again, it's important to **Save** the changes prior to exiting.

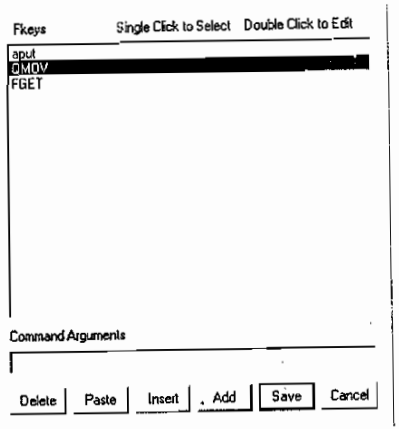


Figure 1.18 Setting up the Fkeys to speed up layout.

Some other useful commands that can speed up layout, when the layout contains a large number of cells, are the **Outl**, **Full**, and **Dpth** commands. To avoid having to redraw the contents of a cell or cells, the **Outl** (outline) command can be used. Either double-clicking on a cell or surrounding the cell's contents with the mouse, when the **Outl** command is selected, causes the cells to be drawn as outlines. Pressing **n** on the keyboard or **N** with the mouse cursor in the lower right portion of the display, toggles the cell's name in its outline on and off. Using the **Full** command shows the contents of the cell.

If we are editing a cell with lower ranking cells nested in it, then we can use the **Dpth** (depth) command to limit the full display of these lower ranking cells. If depth is set to 0, then only the basic objects (boxes, paths, text) in the current-edited cell are shown in detail. The lower level cells are shown as dashed outlines. If the depth is set to 1, then the basic objects in the next level of nested cells are shown ... and so on. Note that cell nesting levels are not the same as cell rank. To show the basic objects of all possible nested cells, the depth should set to 15. To see a listing of nested cells within other cells, use the **Show** command.

Finally, when editing both cells and objects (boxes, polygons, and paths) the window get, **Wget** (window get) and, **Wmov** (window move) commands can be very useful. While in the **Wget** command mode, the mouse is used to select an area of layout containing both cells and drawn objects. The **Wmov** command moves this layout. Note that when "getting" the layout a cell won't be selected unless it is completely surrounded in the mouse selection area.

Understanding the **Cpy** *and* **Copy** *Commands*

The **Cpy** command is used to copy layout within a cell. The layout is selected using, for example, the **Fget**, **Cget**, or **Wget** commands. The **Cpy** command then behaves just like a move command except that the selected layout isn't moved to the new location: it is copied.

It's important to understand that using the **Cpy** command frequently is not a good idea. For example, say a memory cell containing 50 objects (boxes, polygons, or paths) is laid out. If the memory cell is used in a 1 Mbit memory, and the **Cpy** command is used, then the resulting number of objects in the memory cell is 50 million (this is bad). If, on the other hand, the 1 Mbit memory is made with the 1,000,000 memory cells, then the size (the number of objects) is 1,000,050. We can take this a step further. We can make a cell containing a row of memory cells (say 1,000) and use this row cell 1,000 times to make a 1 Mbit memory. The number of objects in the 1 Mbit memory cell is now only 2,050 (1,000 instances of the memory row, 1,000 instances of the memory bit, and 50 objects in the memory bit). We can take this even further to reduce the size of the layout file. **Using cell hierarchy is extremely important.** Do not lay out a large chip unless the material in this paragraph is understood! Else the design file sent to the mask maker will be unreasonably large. Use the **Cpy** command as little as possible.

Using the **Save** command saves the contents of the cell we are working on. It can also be used to save the cell under a different name. However, sometimes we want to take some portion of the layout we are working on and move it (append it) to a different cell (or a new cell). The **Copy** command is useful for this task. To take some layout and append it to a different cell, we first "get" the layout we want to copy to a different cell. Next we select the **Copy** command and the cell we want to move the selected layout into (or the name of a new cell).

Transporting Cells in LASI

To share files between other users or between design directories, *Transportable LASI Cell* files (*.TLC) files or *Transportable LASI Drawing* files (*.TLD) files are used. The files we draw in LASI are saved on the hardisk as text files with the TLC extension.

Copying TLC files into a directory will **not** make them show up in the design directory's cell listing (and so we won't be able to edit the cells). To register a cell in a design directory, we must first **Import** the cell. Note, in the **System** menu, the **Organize** command can be used to organize the cells in a design directory according to rank or name.

The TLC files contain references to other cells but not the actual layouts of the referenced cells themselves. (For example, a reference may indicate "place the inverter cell at the location 12,13.") To get a file that contains the cell's hierarchy, we use TLD files. To generate a TLD file, we go to the **Export** command and select the name of the cell, TLD file, and the location we want to place the TLD file. TLD files can be shared between users or used as backups of work. Note that while LASI imports a TLD file, it still uses TLC files for editing. Every time **Save** is pressed, the TLC file in the design directory (being edited) is updated, leaving the TLD file unchanged (until the **Export** command is used).

Edit-in-Place

Consider the layout seen in Fig. 1.19. In this layout we have two boxes and two cells. Let's say we want to align the cell on the right to the two boxes. The simplest way to do this (keeping in mind that modifying the cell propagates through the hierarchy) is to get the cell using the **Cget** command. Then press **Ctrl** on the keyboard while pressing **Load**. This causes LASI to enter the "Edit-in-Place" (EIP) mode. A lower ranking cell can then be modified while showing the objects from the higher ranking cell. To exit EIP, use the

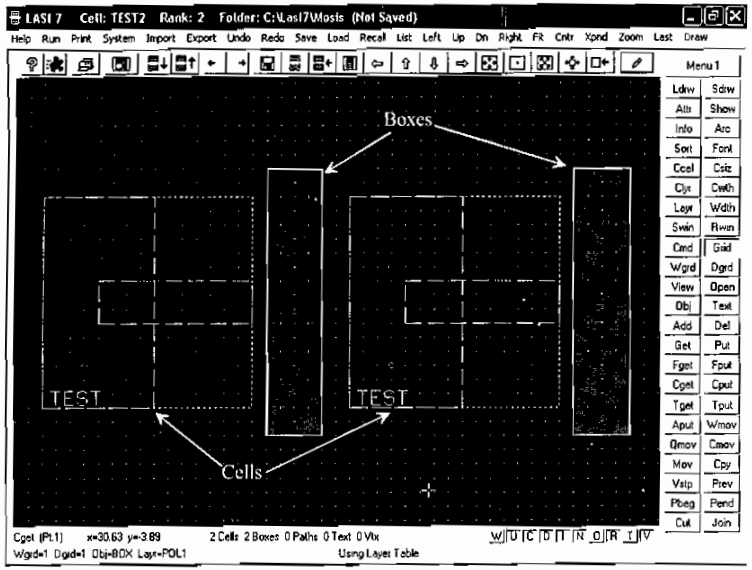


Figure 1.19 Editing a cell in place.

Load or **List** commands. Note that the other cells in the layout will not be shown when in EIP mode (only the cell we are editing and the drawn objects from the higher ranking cell).

Backing Up Your Work

It's very important to constantly back up your work in a location other than the hard disk you are running LASI out of. Backups can be as simple as exporting TLD files to a DVD or as complete as copying the design directory (or zipping it up) to a network drive. The point is layout that is laborious so not backing up the work is a mistake.

1.2.2 Common Problems

After adding an object, the object cannot be seen

Check the **View** layers in the drawing display to ensure that the layer is not in hidden mode. The **Draw** command must be used after using **View**.

Cannot **Get** *an object*

(1) Check, using the **Open** command, that the layer can be opened (or moved). (2) Verify that the object is not part of another cell. (3) When trying to get an object made using the path or polygon object make sure that the cursor encompasses a vertex. (4) If the object is text, then text vertex must be encompassed.

Cells are drawn as outlines, or the perimeter of a cell has a dashed line

(1) Use the **Full** command to show the contents of a cell. (2) Use the **Dpth** command to limit the depth of the cells shown. Increasing the depth to the rank of the cell shows all cells. (3) Press **i** on the keyboard to force an outline to be drawn around a cell. This is indicated by the **I** in the bottom right corner of the drawing display. The letter **n** (or **N** in the lower right of the display) toggles the display of the cell's name.

Fit *command causes the drawing window to expand much larger than the current cell*

There is an unknown object someplace in the cell. Use the **Fget** command to get any objects outside the main cell area. Use the **Del** command to delete the unknown object.

Cursor movement is not smooth

(1) The cursor may be in the octagonal mode. Press **o** on the keyboard or the button **O** in the lower right portion of the display to toggle this mode on and off. (2) Make sure that the display isn't zoomed in too much. (3) Pressing **a** can be used to toggle the mouse cursor between the working and unit grids.

1.2.3 Sending the Layout to the Mask Maker

Once we have a completed layout we can convert the resulting TLC files into a format the mask maker will accept. These formats are either CIF (CalTech intermediate format) or GDS (graphical design system, which is a derivative of the older Calma stream format, CSF). We'll focus on using GDS (at the time of this writing the format is actually a second-generation standard called GDSII) since it is what is used in industry. Generating a *.GDS file in LASI (or any other layout program) is often called *streaming the layout out*. Because GDS files can be large, and they are often stored on magnetic storage tapes, generating and storing the GDS file is often called "taping out" or simply "tape-out."

Assuming the layout is saved, we enter the **System** mode in LASI, Fig. 1.20. Next, we select the **Tlc2Gds** command button, Fig. 1.21. The specific details concerning using this utility or the **Gds2Tlc** utility can be found in the on-line manual. One of the important parameters in the setups is the scale factor Lambda. This takes our layout drawn on a "1" grid and scales it to the appropriate final size.

Checking to Make Sure the Layout Scaled Correctly

To ensure that the layout scales correctly when making the GDS file, copy the design directory into a temporary design directory (TDD). LASI is then set up to run from this TDD. This ensures that all of the original layout isn't corrupted by mistakes in the checking process we're about to discuss (**important**). It's also a good idea to back up the directory at this point as well.

In this TDD we can generate a GDS file, using the **Tlc2Gds** utility. Unless, the user has some atypical needs (e.g., eliminating layers in the final GDS file), this is a simple matter of specifying the name of the TLC file to convert (ensuring the TLC file in the dummy directory is used), specifying the name of the GDS file to create from the TLC file, specifying the scale factor, and starting the conversion (pressing **Go**) after exiting the setups. Both GDS and layer data map (*.ldm) files are generated. (The user will have to select the layers to add to the ldm file while the conversion is in progress unless an ldm file already exists.) The *.ldm file is used when converting the GDS back into a TLC file discussed below.

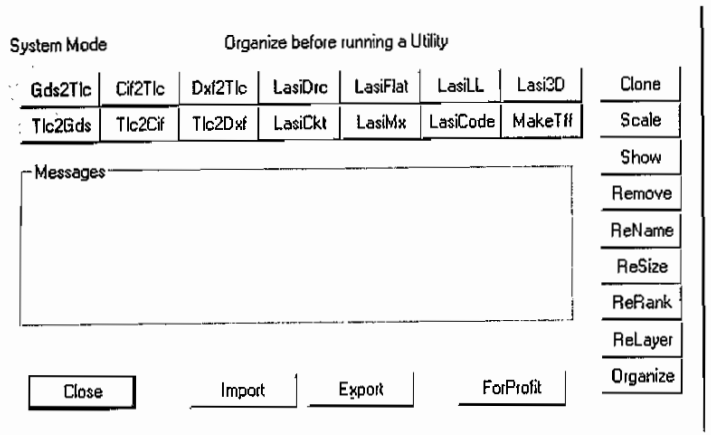


Figure 1.20 The LASI System Screen.

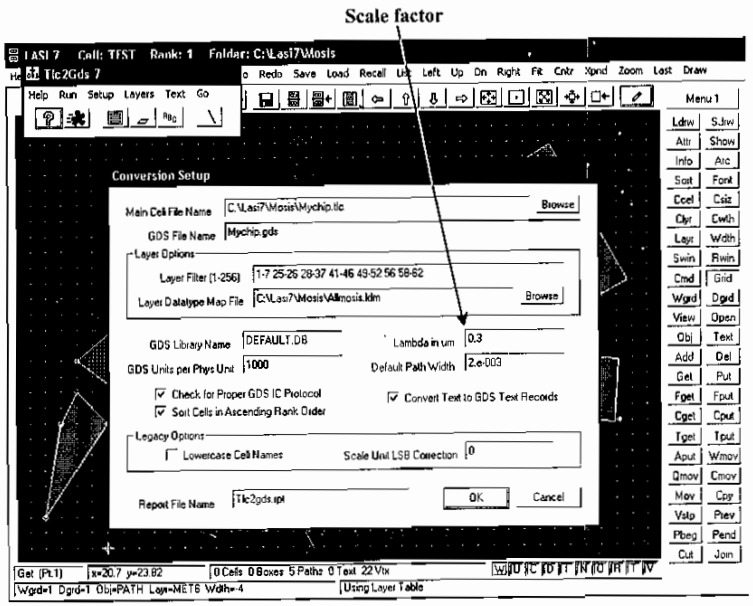


Figure 1.21 Generating a GDS file from a TLC file.

After the GDS file is generated, we can use the **Gds2Tlc** program to convert the GDS file back into TLC files. In the setups we must specify a directory where the TLC files will be written, for example, C:\temp. We can't use a drawing directory because then the existing TLC files would be overwritten. After we've converted the GDS back into TLC files, we can **Import** the (scaled) TLC files into the dummy directory to make sure the generated GDS file is scaled correctly.

Note, for the sake of feeling comfortable with this process, that it's easy to take a simple cell, like the *test* cell in Fig. 1.9 and convert it back and forth between a GDS file and a TLC file (with scale factor). Also note that we can use the **Resize** command on the system menu to change the size of the layout if needed.

1.3 An Introduction to WinSPICE

The simulation program with an integrated circuit emphasis (SPICE) is a ubiquitous software tool for the simulation of circuits. In this book we'll use WinSPICE. See the links at cmosedu.com for download and installation information. WinSPICE, like all SPICE engines, uses a text file netlist for simulation input.

Generating a Netlist File

We can use, among others, the Window's notepad or wordpad programs. WinSPICE likes to see files with a "*.cir" extension. To save a file with this extension, place the file name and extension in quotes as seen in Fig. 1.22. If quotes are not used, then Windows will tack on ".txt" to the filename. This can make finding the file difficult when we open the netlist with WinSPICE (see Fig. 1.23).

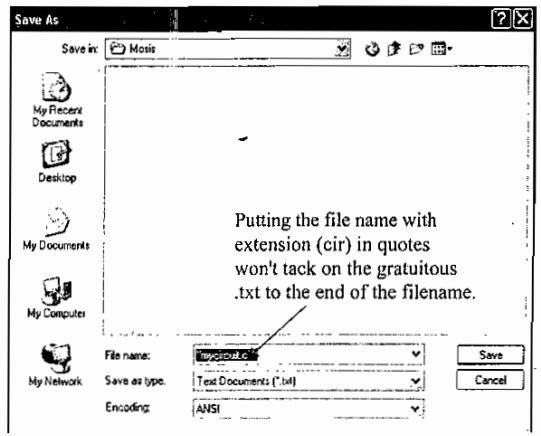


Figure 1.22 Saving a text file with a ".cir" extension.

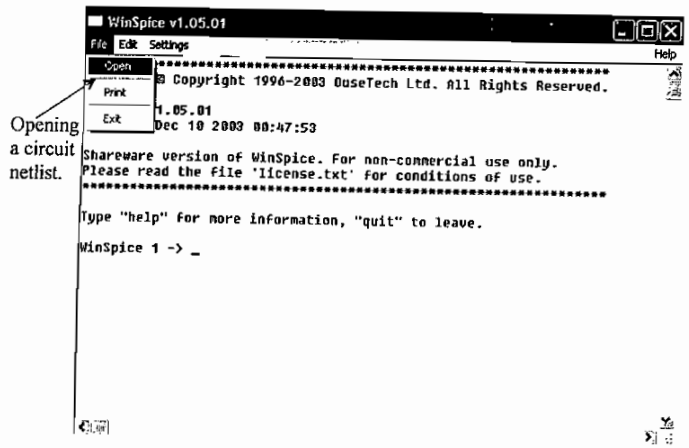


Figure 1.23 Opening a file with WinSPICE.

Transient Analysis

A SPICE transient analysis simulates circuits in the time domain (like an oscilloscope the x-axis is time). Let's simulate the simple circuit seen in Fig. 1.24. A simulation netlist (the text file) may look like:

```
*** Figure 1.25 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
plot vin vout
.endc

.tran 100p 100n

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

.end
```

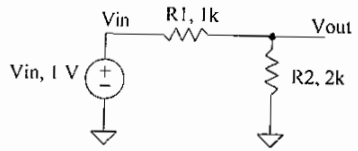


Figure 1.24 Simulating the operation of a resistive divider.

The simulation results using this netlist are seen in Fig. 1.25. The first line in a netlist is a title line. This line is ignored by SPICE (important). The next five lines are control commands. Notice how the end of the control statement is terminated with an ".endc" not an ".end" like at the end of the netlist. Placing ".end" at the end of the control statement causes SPICE to ignore all of the lines containing the circuit information. The statements in the control block can be run directly from the command line in the WinSPICE command window seen in Fig. 1.23. The *destroy all* command destroys all of the previous simulation results (so we don't display old data). The *run* command runs the simulation. The *plot* command plots the voltages on the nodes Vin and Vout.

Note that the DC voltage source Vin is connected to the node Vin. We could have labeled the Vin node with a number like "1." However it is nice to have node names that correspond with signals.

The connection of the resistors and how they are specified should be easy to determine. A line starting with an "R" indicates a resistor specification. A line beginning with an "*" indicates a comment. Node 0 (zero) is always reserved for ground.

The form of the transient statement (this is the type of analysis) 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 (again this term is optional). 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.

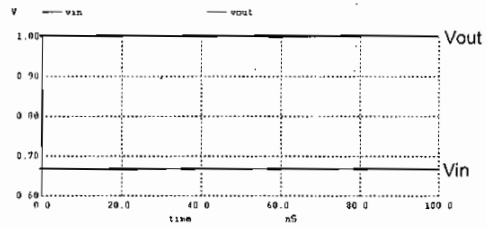


Figure 1.25 Simulating the circuit in Fig. 1.24.

To illustrate a simulation using a sinewave, examine the schematic in Fig. 1.26. 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. To simulate the circuit in Fig. 1.26, we use a netlist of

```
*** Figure 1.26 CMOS: Circuit Design, Layout, and Simulation ***

.control
destroy all
run
plot vin vout
.endc

.tran 1n 3u

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

.end
```

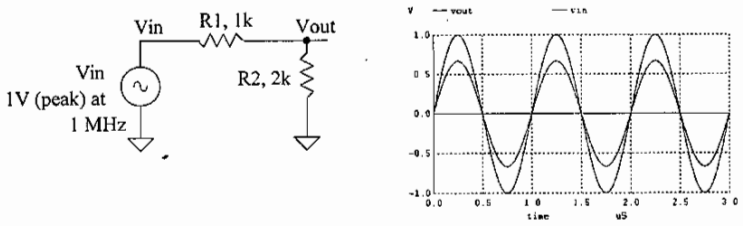


Figure 1.26 Simulating the operation of a resistive divider with a sinewave input.

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. It is **not** used in an AC analysis.

Before leaving this introduction to transient analysis, let's introduce the SPICE pulse statement. This statement has a format given by

```
PULSE VINIT VFINAL TD TR TF PW PER
```

VINIT is the pulse's initial voltage, 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 1.27 provides an example of a simulation that uses the pulse statement. A section of the netlist used to generate the waveforms in this figure is seen below.

```
.tran 100p 30n

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

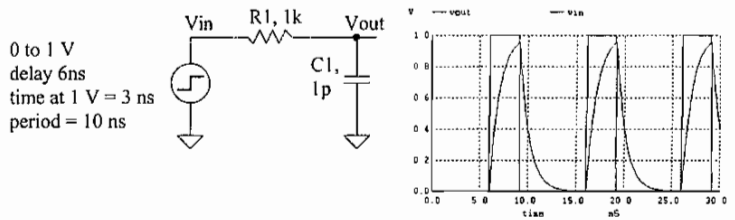



Figure 1.27 Simulating the step response of an RC circuit using a pulsed source voltage.

Other Analysis

Besides the transient analysis presented in this section, we frequently use the SPICE DC and AC analyses. The AC analysis has an x-axis of frequency. This type of analysis is the common "small-signal" analysis used in a basic introductory microelectronics course. The DC analysis has a DC voltage source for the x-axis. The value of the DC source is swept while either a current of voltage is plotted on the y-axis. We don't go into these analyses here but provide numerous examples later in the book.

Convergence

A netlist that doesn't simulate isn't converging numerically. *Assuming* 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
```

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 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] D. E. Boyce, *LASI User's Manual*, available while LASI is running by pressing F1 on the keyboard and the button the user needs help with (at the same time). Alternatively, the user can open the help file in the directory C:\Lasi7\help.

[2] M. Smith, *WinSPICE User's Manual*, Available for download (with the WinSPICE simulation program) at http://www.winspice.co.uk/

PROBLEMS

In the following solutions, it can be very helpful to use the **Prt Sc** (print screen) button on the keyboard to copy contents displayed on the computer's display to the clip board. The image of the display can then be pasted into a document. For ease of viewing (the resulting pasted image in the document), it may also be useful reduce the display resolution prior to using the **Prt Sc** button (right click on the desktop, select properties, then settings).

In the following problems, the solutions should include pictures of the LASI layout screen and discussions indicating that you know what you are doing.

1.1 Create a cell called *mycell* with a rank of 1 using LASI. In this cell, draw a 10 by 10 box using the POL1 layer. Place the lower left corner of the box at the origin. Show how to measure the diagonal distance of the box.

1.2 Explain how the **Qmov** command can be used to edit the box in Problem 1 so that it measures 5 by 8. How would this be accomplished using **Get**, **Mov**, and **Put**?

1.3 What functions do the **Swin** and **Rwin** commands perform? Give an example of where these would be useful.

1.4 What functions do the **Cget** and **Cput** commands perform? Can any other commands be used in place of **Cget**?

1.5 Write the word "test" on the MET1 layer with sizes of 3, 9, and 24 in the cell of Problem 1 (*mycell*). Discuss the possible ways of "getting" the text and "moving " text.

1.6 Using a path and a polygon in the cell (*mycell*) of Problem 1, generate two objects with widths of 10 and lengths of 10 directly next to the box from Problem 1. Discuss the differences in editing boxes, polygons, and paths.

1.7 Explain, in your own words, the difference between the **Cpy** and **Copy** commands. Give examples to demonstrate the differences.

1.8 Demonstrate how the **Wget** and **Wmov** commands function.

1.9 Using the POL1 layer, draw a triangle that measures nominally 10 on each side (make sure the layout is snapped to the working grid). How many vertices does the triangle have? Using LASI's measurement tools, measure the length of the triangle's sides.

1.10 Circles can be drawn using a polygon (path with zero width) and the **Arc** command. Show how to draw a circle using LASI. Make sure the steps are clear.

1.11 What does the **Clone** system command do? Give an example of how to use this command.

1.12 Discuss the different ways cells can be drawn as outlines.

1.13 What do the keyboard buttons **w**, **u**, **a**, **z**, and **space** do in LASI? Give examples to illustrate your understanding of each key's function.

1.14 Describe how to add text in LASI and how to set the text size and layer.

1.15 Explain, in your own words and with examples, how to edit a cell in place.

1.16 Simulate the operation of the circuit in Fig. 1.28. What happens if you use a pulse frequency that's too high? What happens if your simulation time is too short? Give examples to illustrate your understanding.

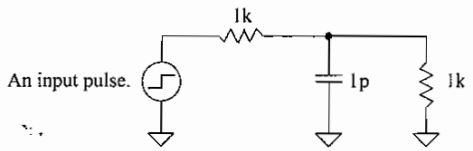


Figure 1.28 Circuit used in Problem 1.16

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).

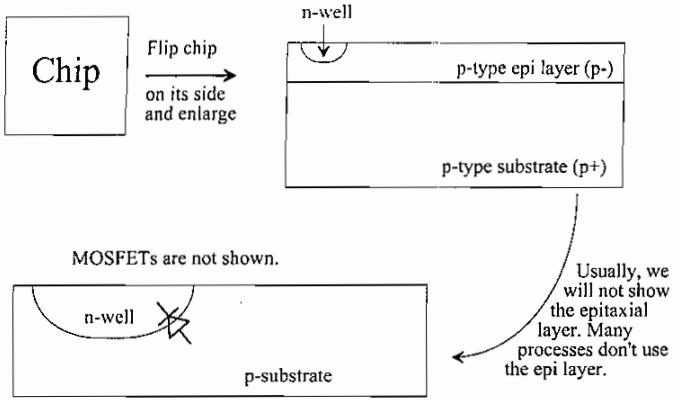


Figure 2.1 The top (layout) and side (cross-sectional) view of a die.

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.

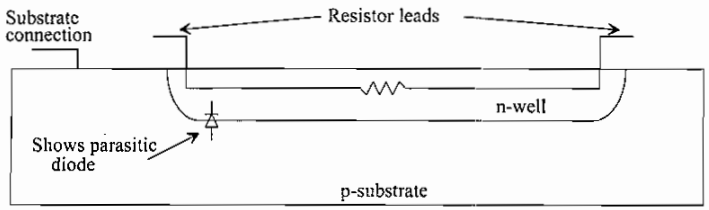


Figure 2.2 The n-well can be used as a resistor.

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.

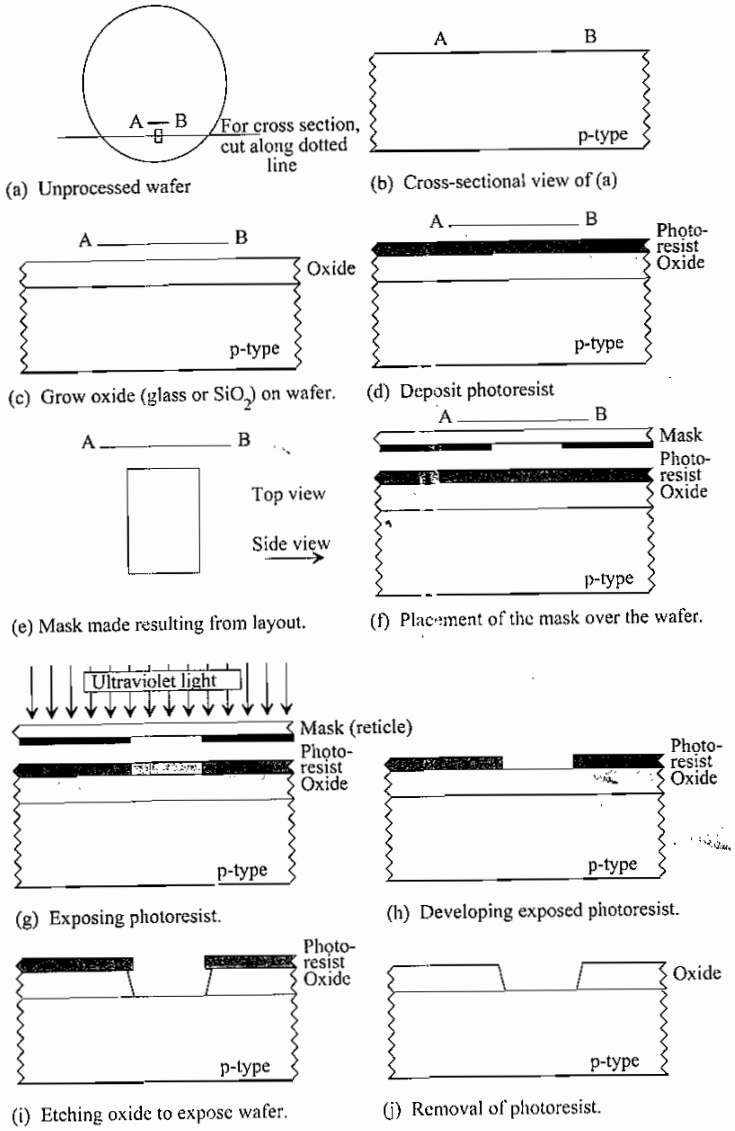


Figure 2.3 Generic sequence of events used in photo patterning.

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

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

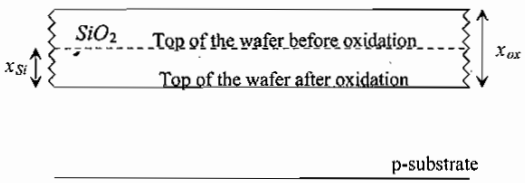


Figure 2.4 How growing oxide consumes silicon.

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.

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).

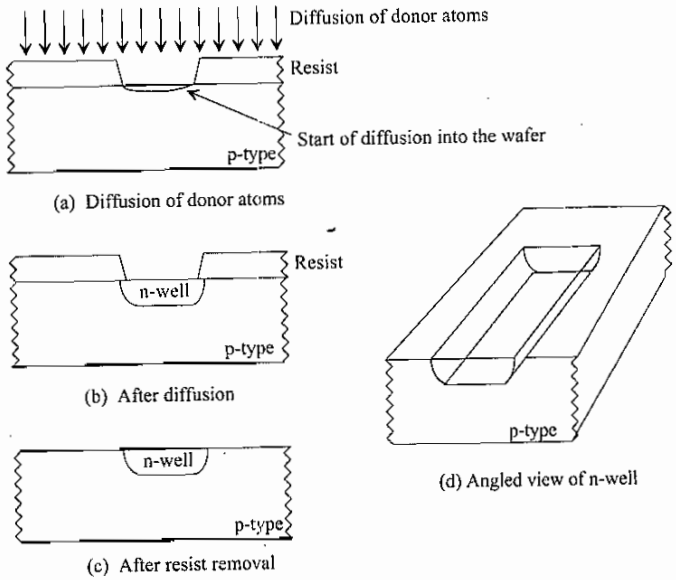


Figure 2.5 Formation of the n-well.

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 = 50×10^{-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.

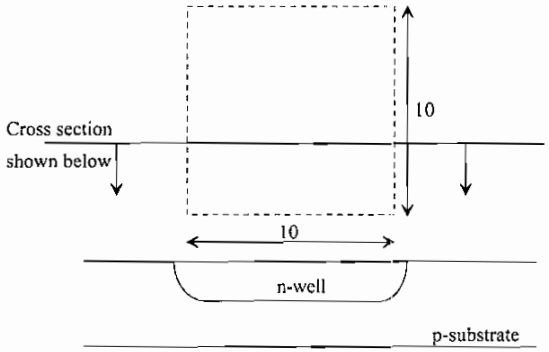


Figure 2.6 Layout and cross-sectional view of a 10 by 10 (drawn) n-well.

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.)

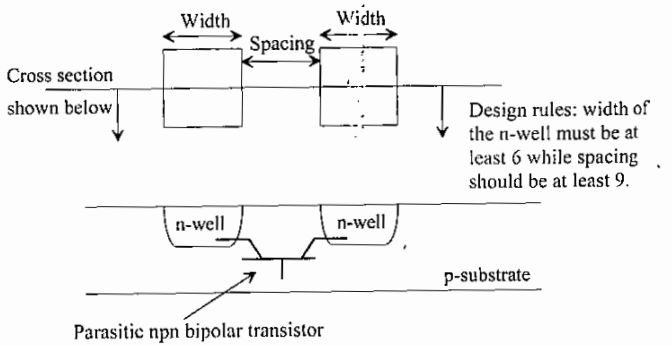


Figure 2.7 Sample design rules for the n-well.

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 $R = R_{square}$).

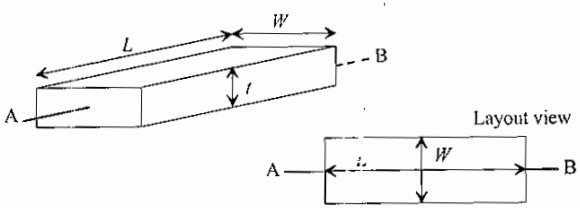


Figure 2.8 Calculation of the resistance of a rectangular block of material.

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_I.

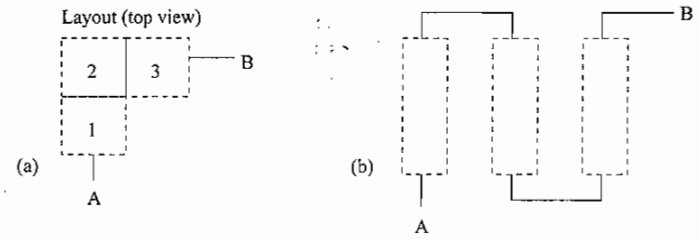


Figure 2.9 (a) Calculating the resistance of a corner section and (b) layout to avoid corners.

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.

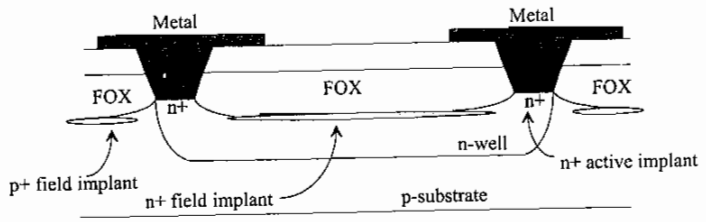


Figure 2.10 Cross-sectional view of n-well showing field implant. The field implantation is sometimes called the "channel stop implant."

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

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

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 (1.3806×10^{-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 1.6022×10^{-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

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.

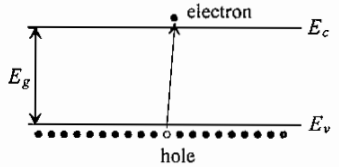


Figure 2.11 An electron moving to the conduction band, leaving behind a hole in the valence band.

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, τ_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

$$n_i \approx 14.5 \times 10^9 \ carriers/cm^3 \qquad (2.5)$$

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*,

$$n = p = n_i \approx 14.5 \times 10^9 \ carriers/cm^3 \qquad (2.6)$$

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

$$N_{Si} = 50 \times 10^{21} \ atoms/cm^3 \qquad (2.7)$$

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

$$n \approx N_D \ \text{when} \ N_{Si} >> N_D >> n_i \qquad (2.8)$$

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

$$p \approx N_A \ \text{when} \ N_{Si} >> N_A >> n_i \qquad (2.9)$$

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*

$$pn = n_i^2 \qquad (2.10)$$

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

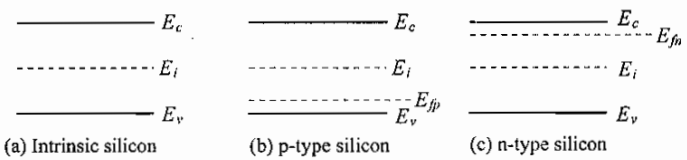
$$p = \frac{n_i^2}{n} = \frac{(14.5 \times 10^9)^2}{10^{18}} = 210 \text{ holes/cm}^3$$

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 determing 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.



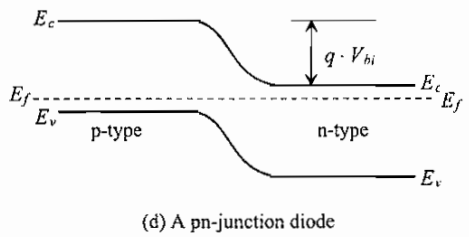


Figure 2.12 The Fermi energy levels in various structures.

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.

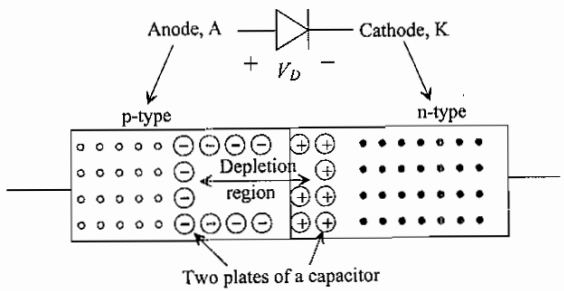


Figure 2.13 Depletion region formation in a pn junction.

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 (= 100×10^{-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)).

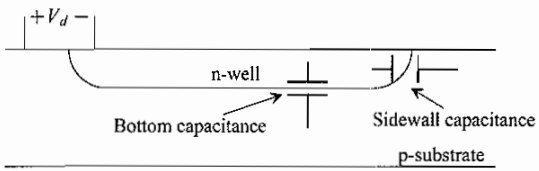


Figure 2.14 A pn junction on the bottom and sides of the junction.

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)

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

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

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

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. ■

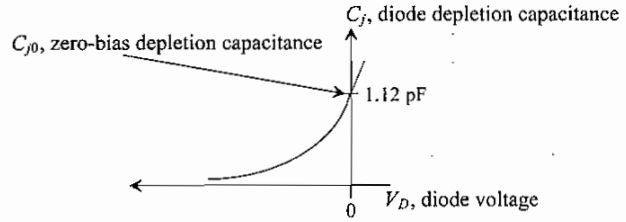


Figure 2.15 Diode depletion capacitance against diode reverse voltage.

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, τ_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 τ_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.

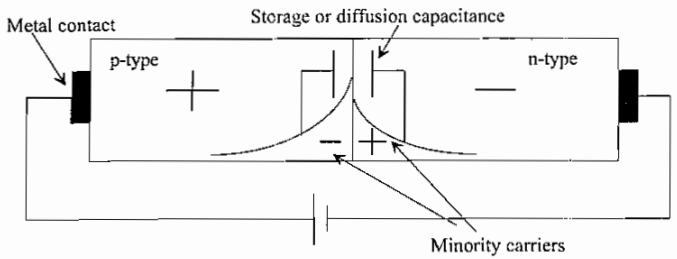


Figure 2.16 Charge distribution in a forward-biased diode.

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 >> 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

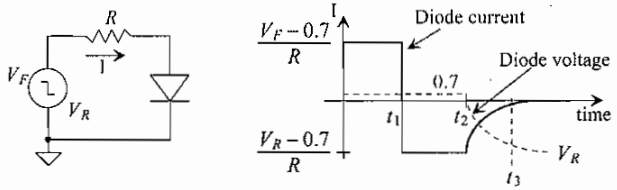


Figure 2.17 Diode reverse recovery test circuit.

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
τ_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.

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.

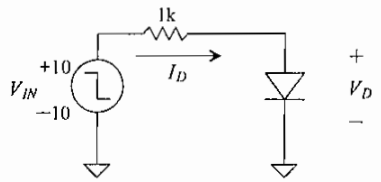


Figure 2.18 Circuit used in Ex. 2.4 to demonstrate simulation of a diode's reverse recovery time.

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. ■

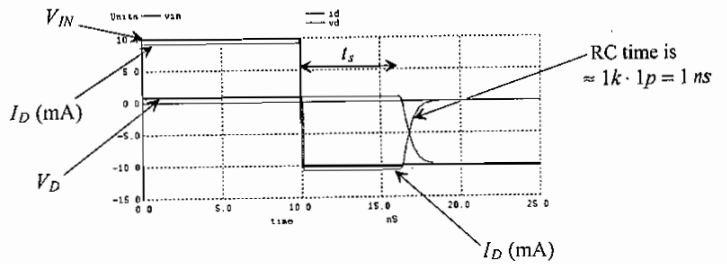


Figure 2.19 The simulation results for Ex. 2.4.

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.

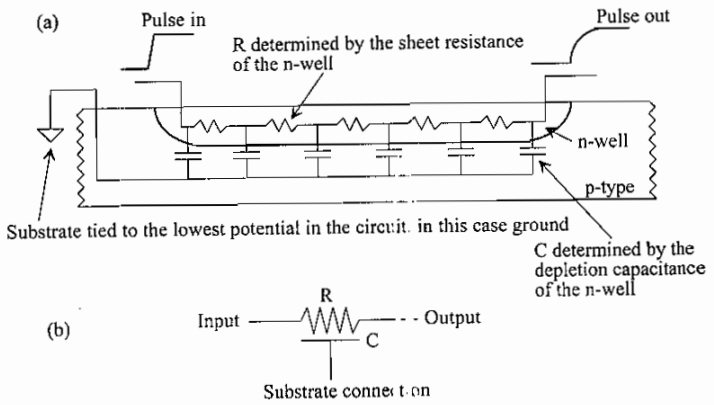


Figure 2.20 (a) Parasitic resistance and capacitance of the n-well and (b) schematic symbol.

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).

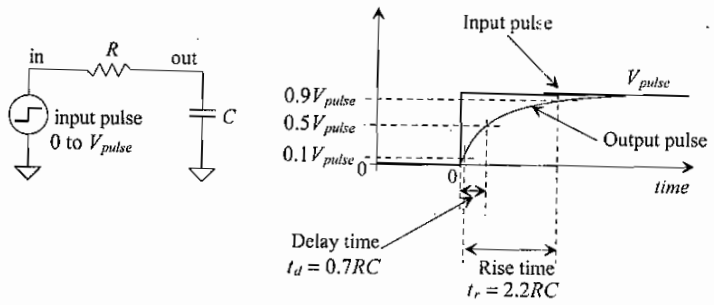


Figure 2.21 Rise and delay times in an RC circuit.

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

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

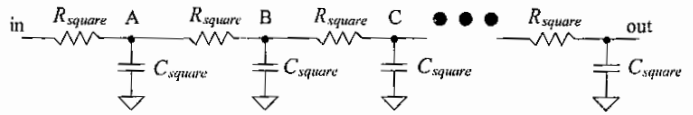


Figure 2.22 Calculating the delay through a distributed RC delay.

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.

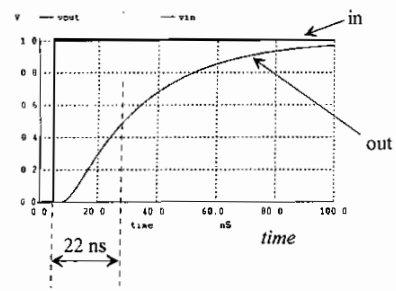
We can divide the n-well up into 50 squares each having a size of 10 by 10 and a resistance of 5 kΩ. The delay through the resistor is then (remembering 1 femto = 10^{-15})

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

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. ■




```
*** 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 Using the LasiDrc Program

Before presenting the use of the LasiDrc program, 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 (a) 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.

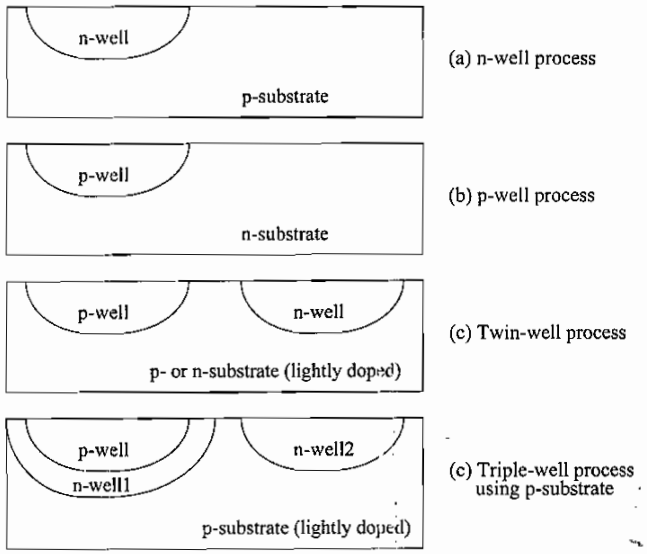


Figure 2.24 The different possible wells used in a bulk CMOS process.

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. 2.24c. 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, λ, is used in the rules. All of the layouts are drawn on a λ 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 10λ by 10λ

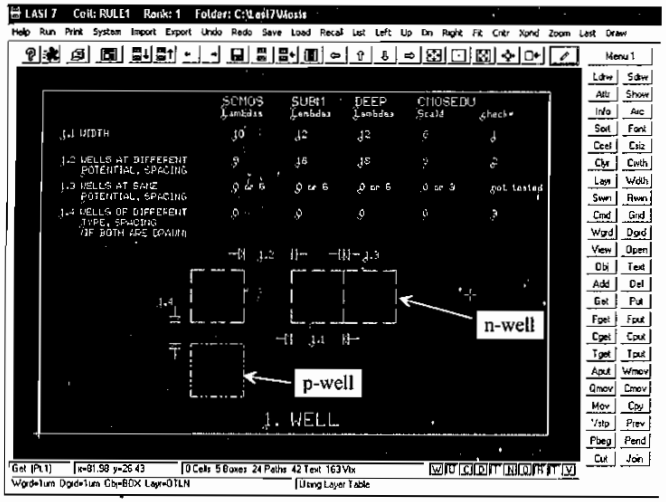


Figure 2.25 LASI layouts showing the MOSIS design rules for the n-well.

with 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 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

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.

Running a Design Rule Check

Going to the **System** menu in LASI and then pressing on the **LasiDrc** button starts the LASI DRC utility program. Figure 2.26 shows the setup screen in this utility. When starting a DRC, it's important to make sure that the correct cell name is specified, the correct checks are selected, and the correct DRC file name is used. Also, if the entire cell is to be DRCed, then the **Fit** command button should be pressed (to fit the layout to the DRC check area). If this button isn't pressed, then the last view seen in LASI when the cell is saved, prior to calling the DRC program, will be used (this can be helpful to reduce DRC time when the layout is large). Further, if the DRC is aborted (by pressing **ESC** on the keyboard), then the LasiDRC remembers where the checking left off, allowing the DRC program to continue if errors are found and fixed in the middle of the DRC run. If the DRC program is run on a cell after making edits and the cell is not *first saved*, the edits won't be seen by the DRC program (**this is a common user mistake**). Finally, errors are both reported in an output file (lasidrc.rpt) and as pictures. Images of the layouts and errors can be viewed with the button indicated in Fig. 2.26.

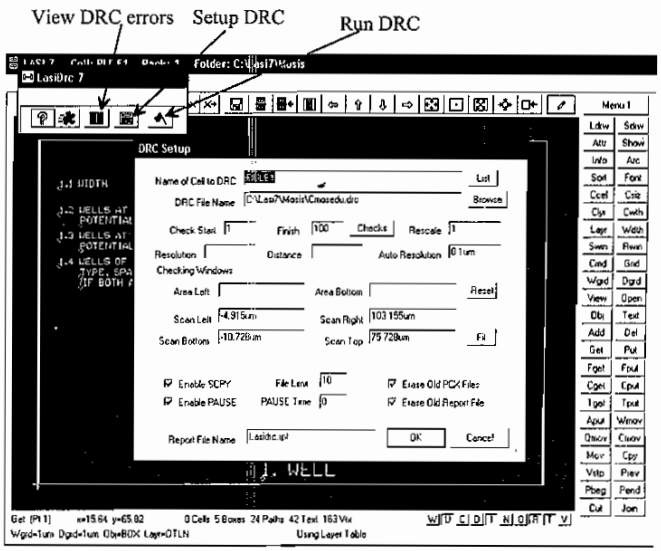


Figure 2.26 Setting up a DRC run.

ADDITIONAL READING

[1] J. P. Uyemura, *Introduction to VLSI Circuits and Systems*, John Wiley and Sons Publishers, 2002. ISBN 0-471-12704-3

[2] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, Cambridge University Press, 1998. ISBN 0-521-55056-4 (hardback) or ISBN 0-521-55959-6 (paperback)

[3] M. Smith, *WinSPICE User's Manual*, Available for download (with the WinSPICE simulation program) at http://www.winspice.co.uk/

[4] J. Jessing, *Introduction to CMOS Process Integration: Front-End of the Line (FEOL)*, course notes at Boise State University.

[5] D. E. Boyce, *LASI User's Manual*, available while LASI is running by pressing F1 on the keyboard and the button the user needs help with (at the same time). Alternatively, the user can open the help file in the directory C:\Lasi7\help.

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?

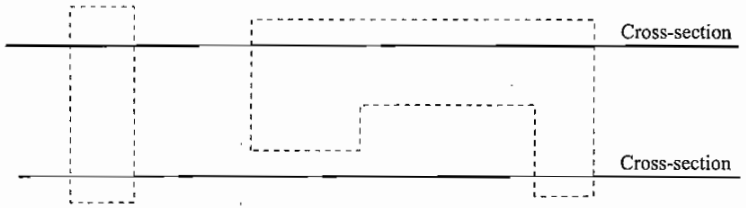


Figure 2.27 Layout used in problem 2.1.

2.2 Sketch (or use LASI) 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 (using a path object in LASI or any other layout program) 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?

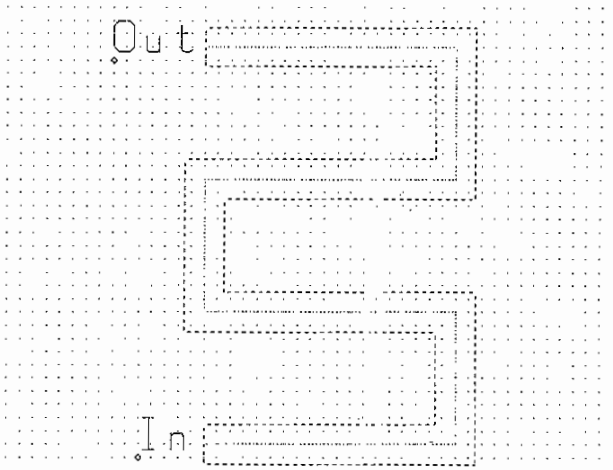


Figure 2.28 Layout of an n-well resistor using a serpentine pattern.

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 $|v_{out}/v_{in}|$).

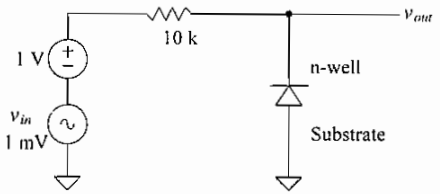


Figure 2.29 Treating the diode as a capacitor. See Problem 2.7.

2.8 Verify the answer given in problem 2.7 with SPICE.

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.

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.

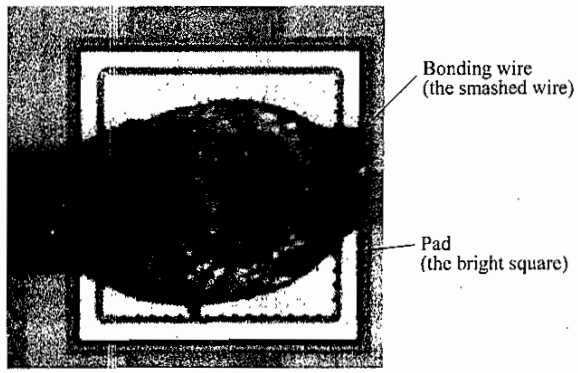


Figure 3.1 The bonding wire connection to a pad.

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 (MET2 in LASI) 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.

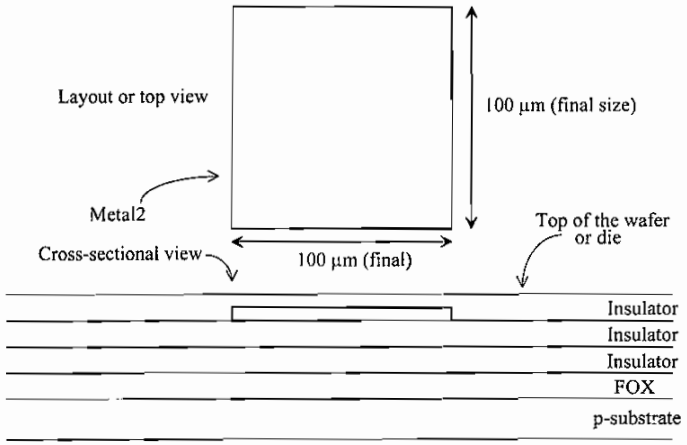


Figure 3.2 Layout of metal2 used for bonding pad with associated cross-sectional view.

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

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

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 metal1 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

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

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." ■

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 (OVGL). 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.

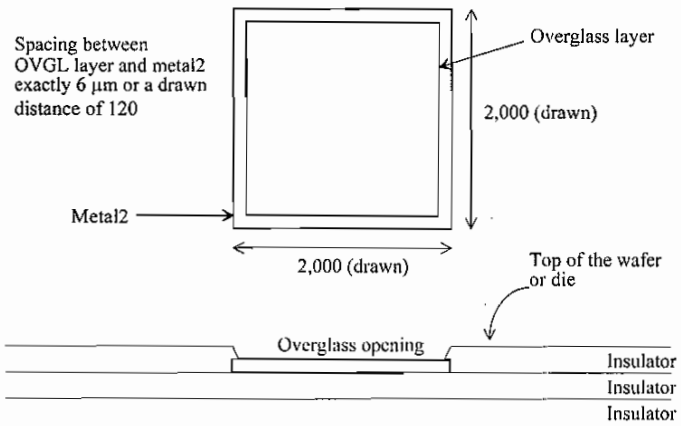


Figure 3.3 Layout of a metal2 pad with pad opening for bonding connection in a 50 nm (scale factor) CMOS process.

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 (NWEL), metal2 (MET2), overglass (OVGL), and pad (PAD). In this section we'll also learn about the metal1 (MET1) and via1 (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.

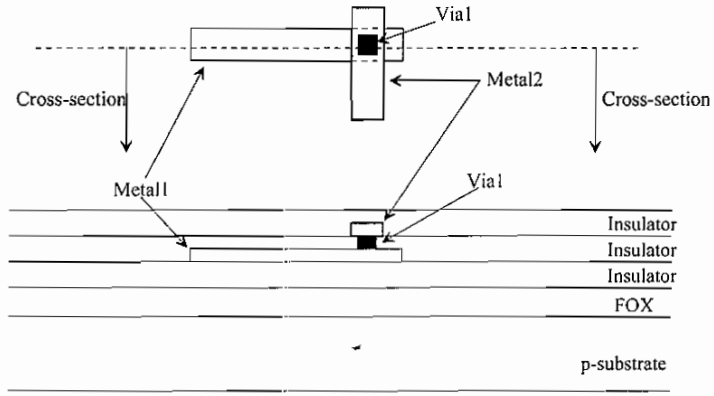


Figure 3.4 Layout and cross-sectional views.

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.

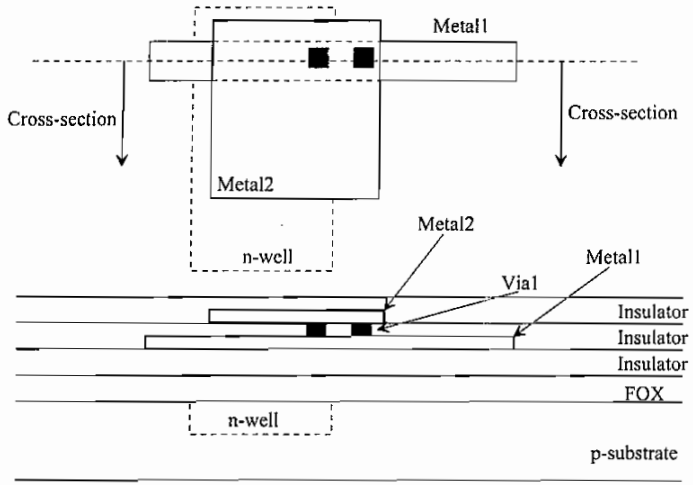


Figure 3.5 An example layout and cross-sectional view using including the n-well.

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)

$$C = (1,000 \cdot 0.2) \cdot 23\ aF + (2,000.4) \cdot 79\ aF = 162\ fF$$

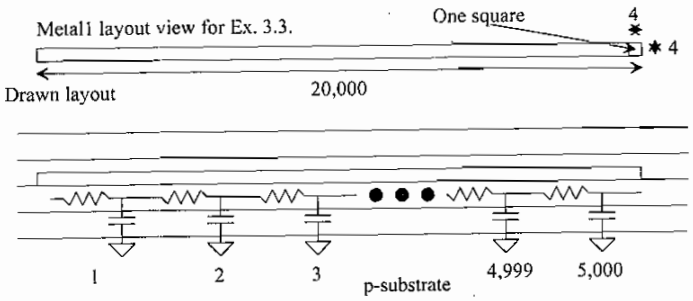


Figure 3.6 Layout and cross-sectional view with parasitics for the metal line in Ex. 3.3.

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, ε_r (no parasitic resistance). The velocity, v, of the signal in this situation is related to the speed of light, c, by

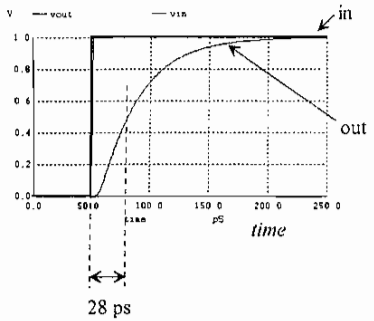
$$v = \frac{c}{\sqrt{\varepsilon_r}}\ (\text{meters/second}) \qquad (3.3)$$

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

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

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).




```
*** 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 5n 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

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

■

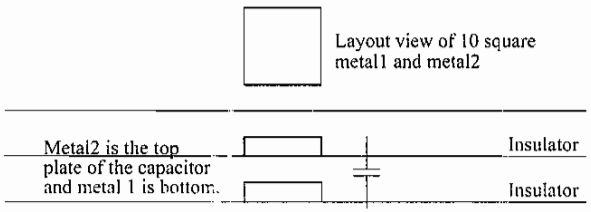


Figure 3.8 Capacitance between metal1 and metal2.

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.

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. The voltage on C_{1sub} is given by

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

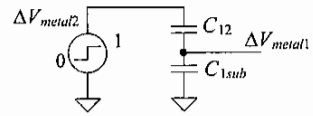
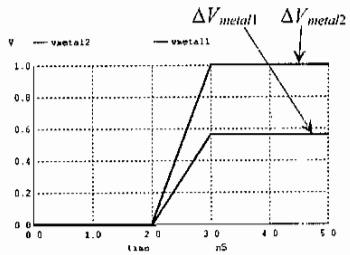


Figure 3.9 Equivalent circuit used to calculate the change in metal1 voltage, see Ex. 3.5.

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 SPICE doesn'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.

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 μm^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.

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).

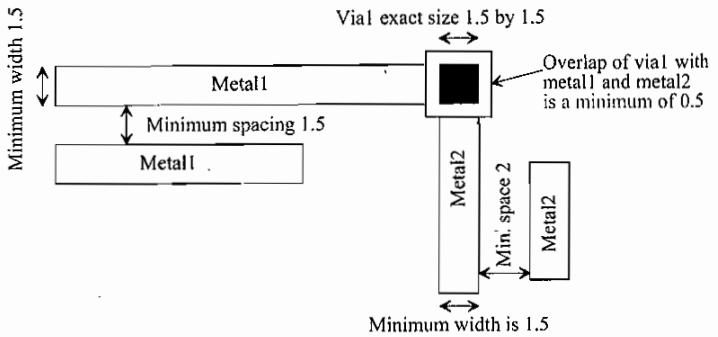


Figure 3.11 Design rules for the metal layers using the CMOSEDU rules.

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" with a rank of 1. 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.

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.

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.

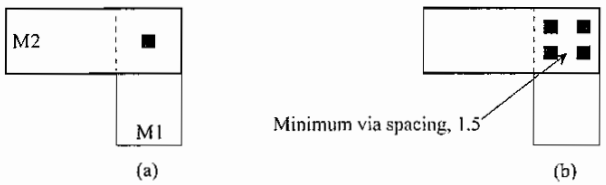
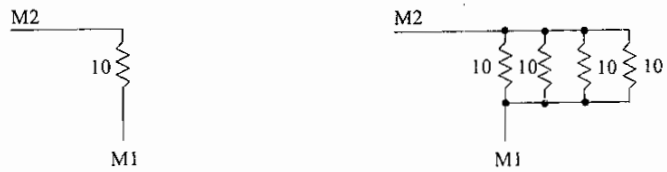


Figure 3.14 Layouts used in Ex. 3.8.

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). ■



(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

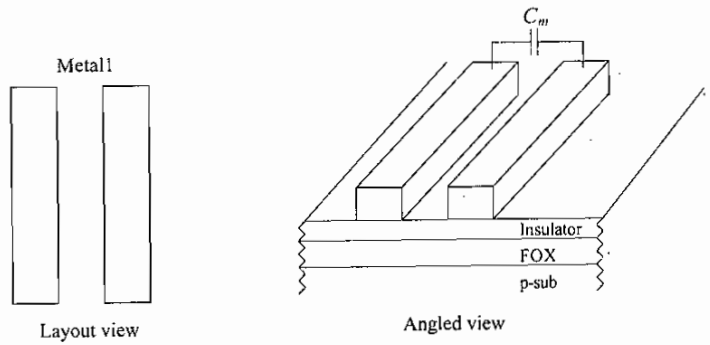


Figure 3.16 Conductors used to illustrate crosstalk.

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 ΔV 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 down during this short time. The average current supplied by *VDD* may be well under a

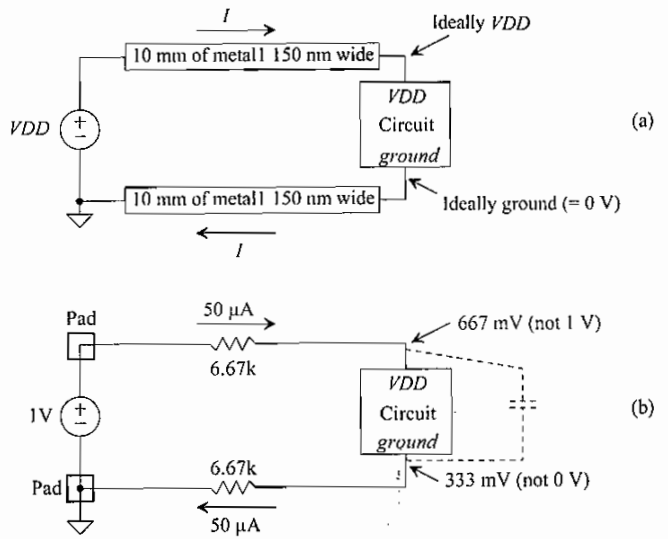


Figure 3.17 Illustrating problems with incorrectly sized conductors.

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

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

A reasonably large capacitor. ■

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.

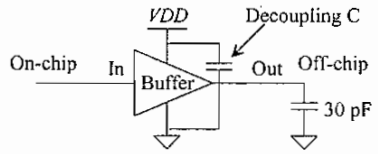


Figure 3.18 Estimating the decoupling capacitance needed in an output buffer.

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.

3.4 LASI Layout Examples

In this section we provide some additional LASI layout examples. In the first section we discuss laying out a pad and a padframe. In the next section we introduce the use of LasiCkt.

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 CMOSEDU 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 making a rank 1 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 the 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 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. Dividing 1/0.13 we get, after rounding, eight pads on a side. However, the corners don't contain a pad so the actual number of pads on a side is six, see Fig. 3.23. Note, in this figure, that we made sure that the pad cell's image was showing (by pressing **i** on the keyboard). The small diamond in the image indicates the location of the reference marker in the cell. As seen in the figure, all of the reference markers in the pads are located close to the edge of the chip (use the **Rot** or **LASI** rotate command). The CMOS circuits are placed in the area inside the pads, while the part that gets cut up when the wafer is sawed into chips is the area outside the pads (the scribe), see Fig. 1.2.

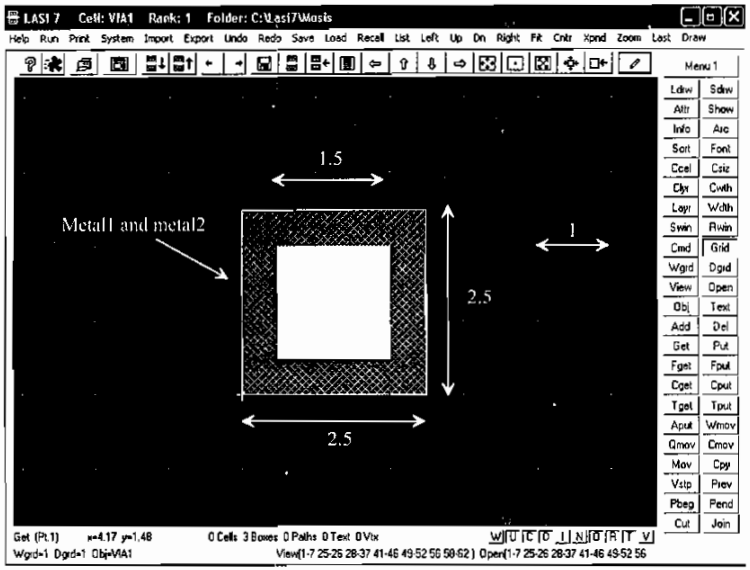


Figure 3.19 Layout of a Via1 cell.

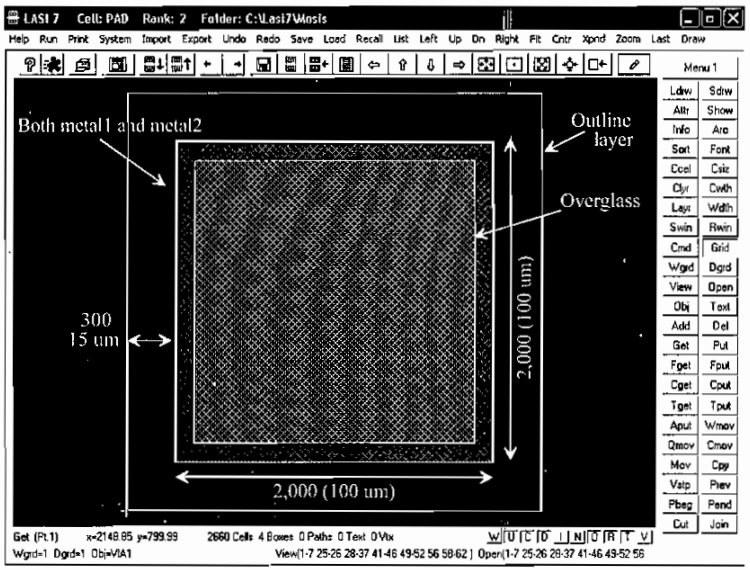


Figure 3.20 Layout of the bonding pad.

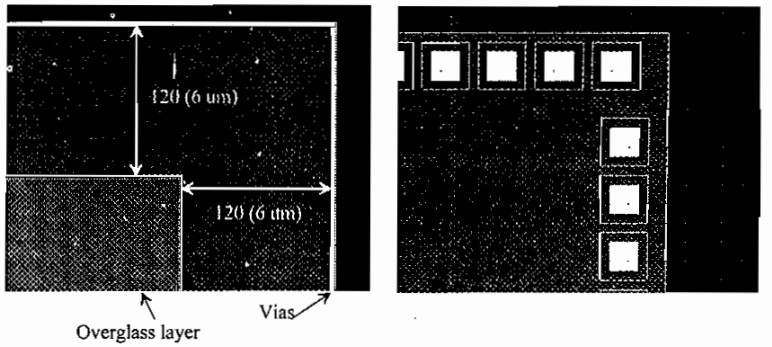


Figure 3.21 Corner detail for the pad in Fig. 3.20.

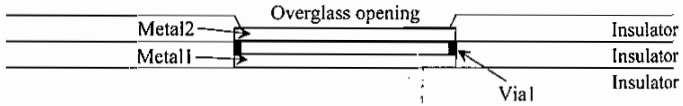


Figure 3.22 Simplified cross-sectional view of the bonding pad discussed in this section.

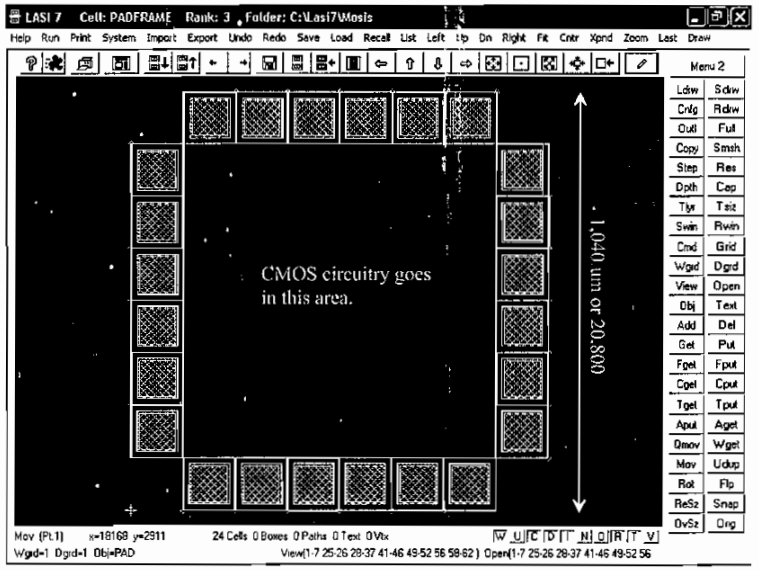


Figure 3.23 The layout of a padframe.

3.4.2 Introduction to LasiCkt

More than a layout tool, LASI can be used to draw schematics. LasiCkt can then be used to generate SPICE netlists from either a layout or a schematic. LASI uses its own format (TLC) for schematics (the schematics are drawn using the same tools and objects used to draw the layouts). Unlike some schematic tools that automatically label MOSFETs or resistors, the user needs to add certain text labels to a LASI drawing using the **Text** command. This text is used by LasiCkt to generate SPICE netlist files. In this section we give an introduction to LasiCkt along with a simple example.

Drawing a Schematic

Let's start by drawing the resistive divider schematic seen in Fig. 3.24 (a rank 2 cell called RDIV_SCH).

1. We can start by adding a rank 1 cell called SCH_RES to RDIV_SCH. This cell (SCH_RES) is a drawing of a resistor using the layer SCHM (schematic layer used for drawing symbols) and two connectors (using the connector text layer, CTXT).

2. Next we use a zero-width path (a polygon) on the MET1 layer (metal1) to connect the cells. We also add a ground cell to the schematic. Note that using a zero width box will **not** connect the symbols.

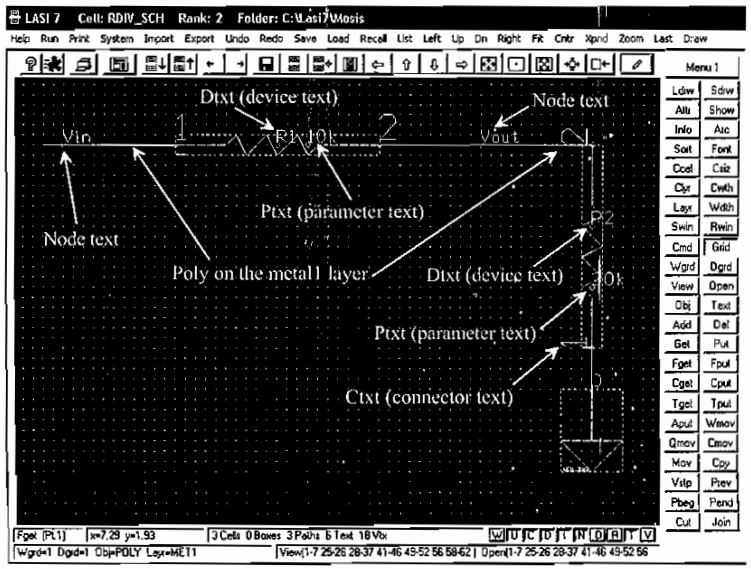


Figure 3.24 Drawing a schematic in LASI.

3. Next we label the nodes (the metal1 polygons) with names using text on the NTXT layer (node text layer). Here we use Vin and Vout for node names. If we don't label the wires, LasiCkt will select a virtual node name (like vn1, vn2, etc.)

4. Next we ensure that each cell's image is visible (the dotted lines around the cells). We place DTXT (device text) and PTXT (parameter text) to indicate the resistor's name and value. The name of a resistor must start with an "R."

We are now ready to run LasiCkt on the schematic. LasiCkt is started from the system menu. Figure 3.25 shows LasiCkt's setup menu. For the moment we'll simply enter the cell's name into the "Name of Cell" field and select "Schematic". Pressing OK and selecting "Go" on the LasiCkt window generates a text file called "rdiv_sch.cir." The contents of the file are

```
*** SPICE Circuit File of RDIV_SCH

* MAIN RDIV_SCH
R1 Vin Vout 10k
R2 0 Vout 20k
.END
```

While this example is simple, we'll see that as our circuits get complicated the ability to generate a netlist from a schematic or a layout allows us to verify that the layout matches the schematic. Sometimes the comparison is called "layout versus schematic" or LVS.

Note that the header file is a text file indicating the type of analysis and the simulation parameters (see the .control line to the .tran line in the netlist on page 24). The footer file is used for models (most often MOSFET models). For more information, see LasiCkt's online help manual.

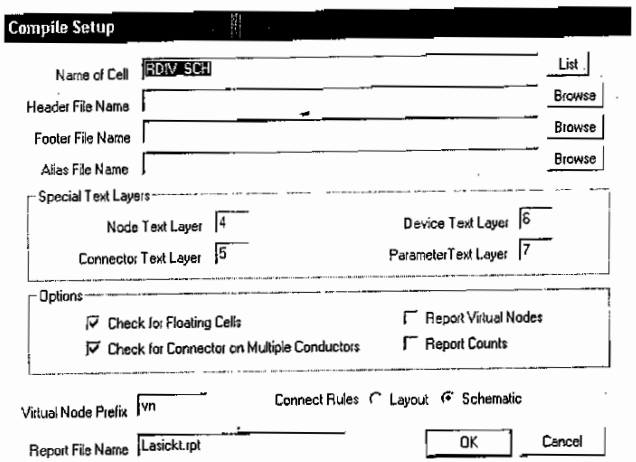


Figure 3.25 LasiCkt setup menu.

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] D. E. Boyce, *LASI User's Manual*, available while LASI is running by pressing F1 on the keyboard and the button the user needs help with (at the same time). Alternatively, the user can open the help file in the directory C:\Lasi7\help.

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.

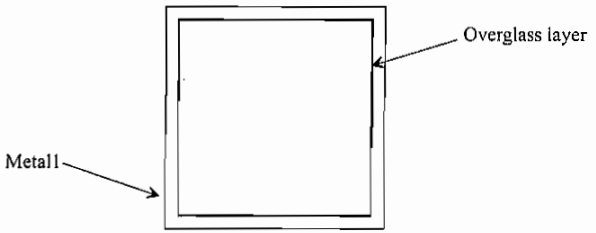


Figure 3.26 Layout used in Problem 3.4.

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?

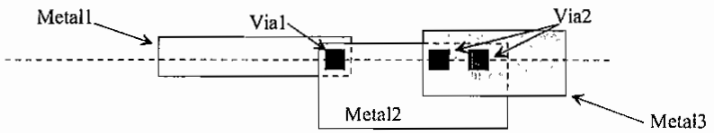


Figure 3.27 Layout for Problem 3.5.

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.

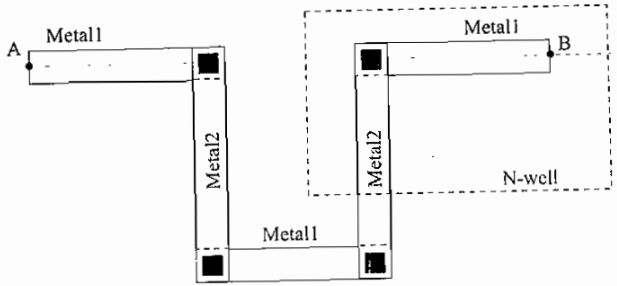


Figure 3.28 Layout for Problem 3.8.

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.

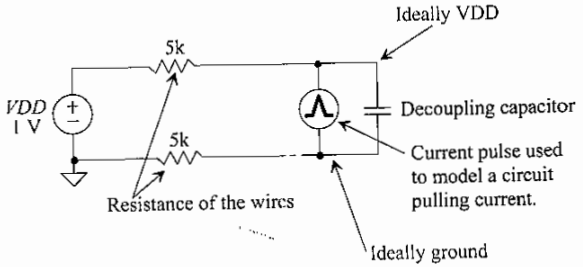


Figure 3.29 Circuit used to show the benefits of a decoupling capacitor.

3.11 Use LasiCkt to draw the schematic seen in Fig. 1.27. The input pulse source should be added into the schematic. Generate a header file that can be used to generate the waveforms seen in the figure. Simulate the output (from LasiCkt) netlist file.

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 (NWEL), metal1 (MET1), via1 (VIA1), metal2 (MET2), and overglass (OVGL). In this section we cover the following additional fabrication layers: active (ACTV), n-select (NSEL), p-select (PSEL), poly1 (POL1), silicide block (SILI), and contact (CONT).

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.

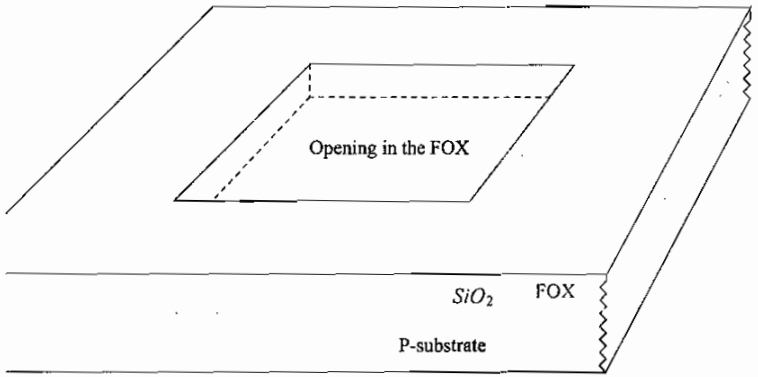


Figure 4.1 How the active layer specifies where to open holes in the field oxide (FOX).

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.

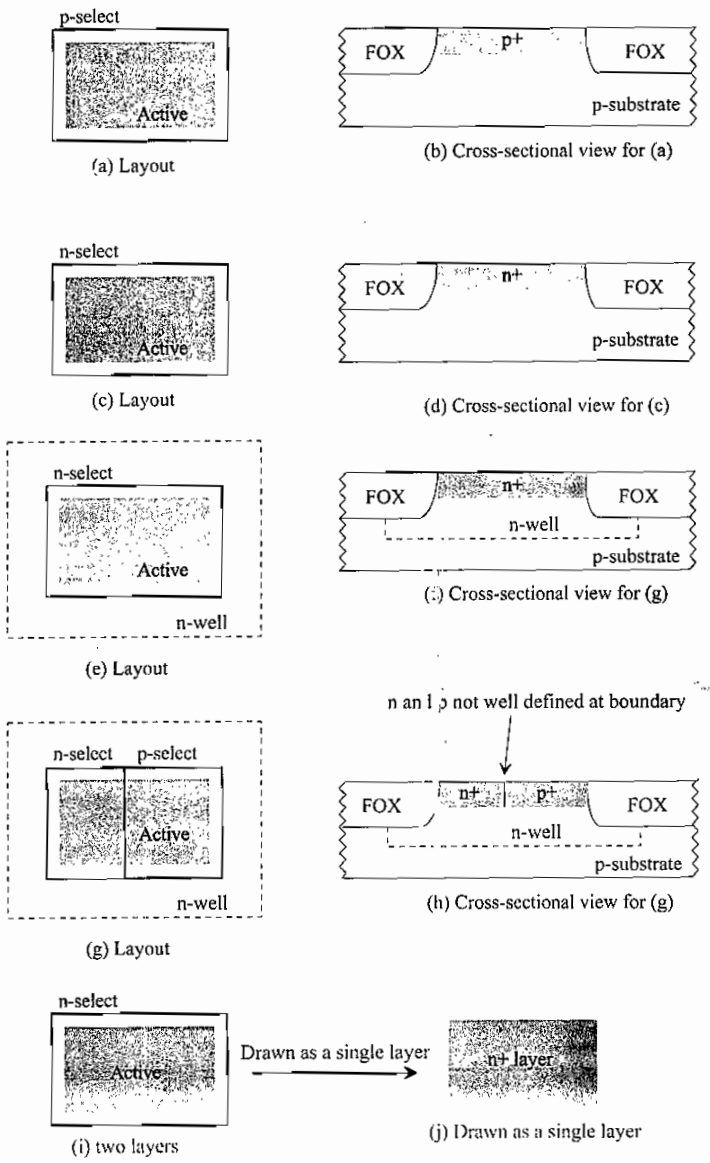


Figure 4.2 Combinations of active, selects, and n-wells.

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).

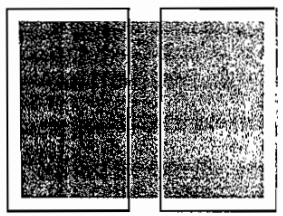
Self-Aligned Gate

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 select masks.

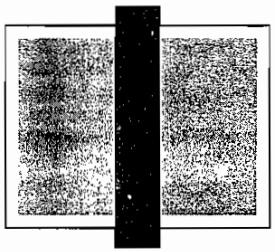
Example 4.1

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. ■

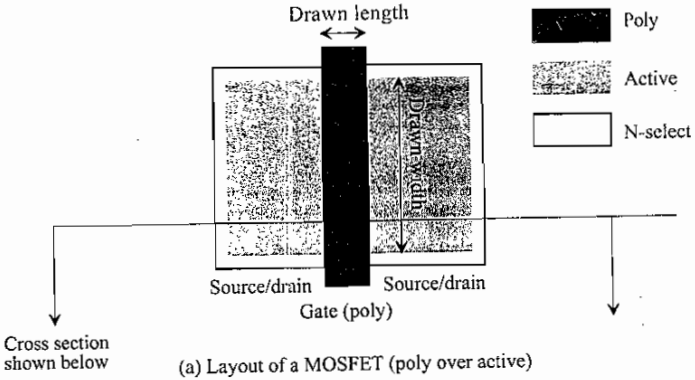


(a) Drawing active and select for a MOSFET (bad).

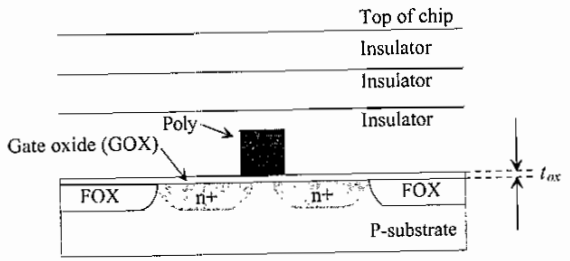


(b) Placing poly over the layout in (a) (bad).

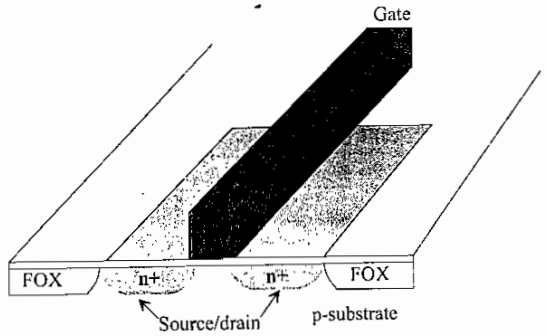
Figure Ex4.1 Bad layout examples (what NOT to do).



(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.

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.

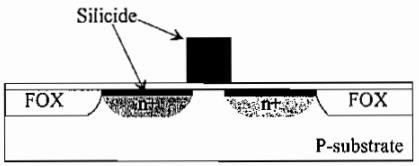


Figure 4.4 How the gate and drain/source of a MOSFET are silicided to reduce sheet resistance.

Table 4.1 gives some typical values of sheet resistance for poly, n+, and p+. Note that when we use the poly as a mask to self-align the source and drain regions of the MOSFET to the gate we dope the poly either n-type (for an NMOS device) or p-type (for a PMOS device). The other specifications (temperature coefficient, TCR, or voltage coefficients, VCR) seen in the table will be used in later chapters.

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

Silicide	Resistor type	Rs (ohms/sq) AVG.	TCR1 (ppm/C) AVG.	TCR2 (ppm/C²) AVG.	VCR1 (ppm/V) AVG.	VCR2 (ppm/V²) AVG.	Mismatch % ΔR/R
	n-well	500 ± 10	2400 ± 50	7 ± 0.5	8000 ± 200	500 ± 50	< 0.1
	n+ poly	200 ± 1	20 ± 10	0.6 ± 0.03	700 ± 50	150 ± 15	< 0.5
	p+ poly	400 ± 5	160 ± 10	0.8 ± 0.03	600 ± 50	150 ± 15	< 0.2
	n+	100 ± 2	1500 ± 10	0.04 ± 0.1	2500 ± 50	350 ± 20	< 0.4
	p+	125 ± 3	1400 ± 20	0.4 ± 0.1	80 ± 80	100 ± 25	< 0.6
*	n+ poly	5 ± 0.3	3300 ± 90	1.0 ± 0.2	2500 ± 125	3800 ± 400	< 0.4
*	p+ poly	7 ± 0.1	3600 ± 50	1.0 ± 0.2	2500 ± 400	5500 ± 250	< 0.7
*	n+	10 ± 0.1	3700 ± 50	1.0 ± 0.2	350 ± 150	600 ± 60	< 1.0
*	p+	20 ± 0.1	3800 ± 40	1.0 ± 0.2	150 ± 50	800 ± 40	< 1.0

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 (SLLI in the LASI setups) 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.

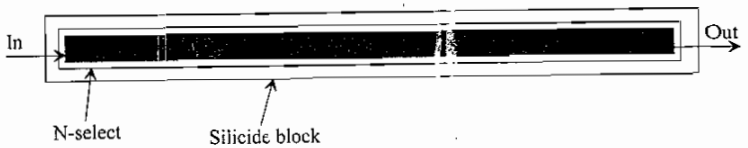


Figure 4.5 Estimating the delay through a polysilicon line with and without a silicide.

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

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

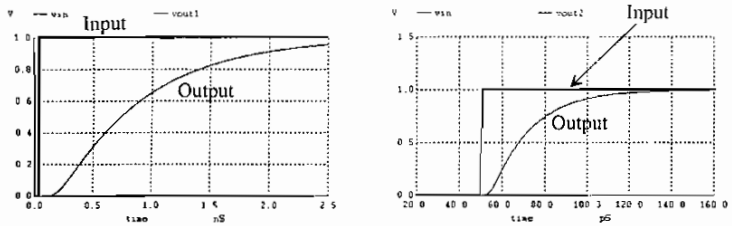
From the data in Table 4.1, the resistance of the wire is either 200k (no silicide) or 5k (with silicide). The delays are then calculated as

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

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). ■

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.



(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.

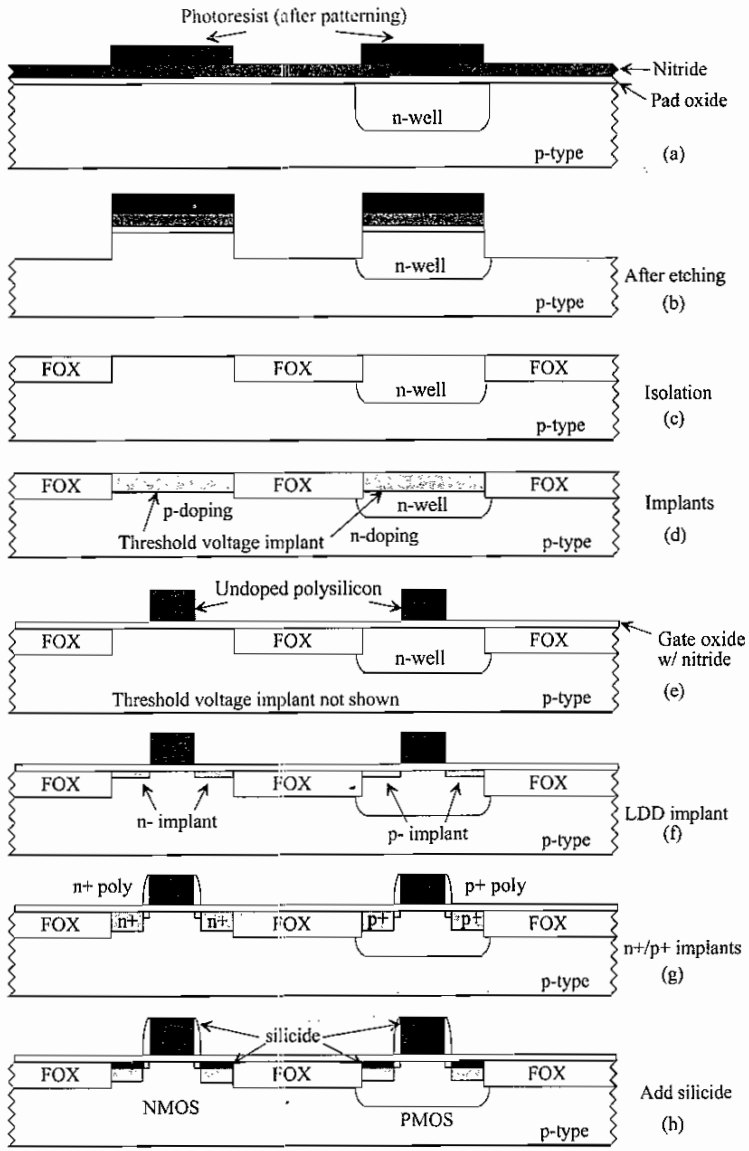
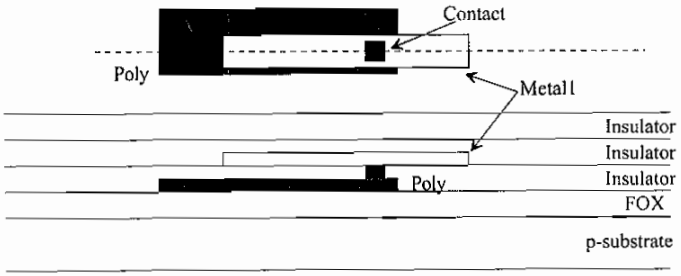


Figure 4.7 General CMOS process flow.

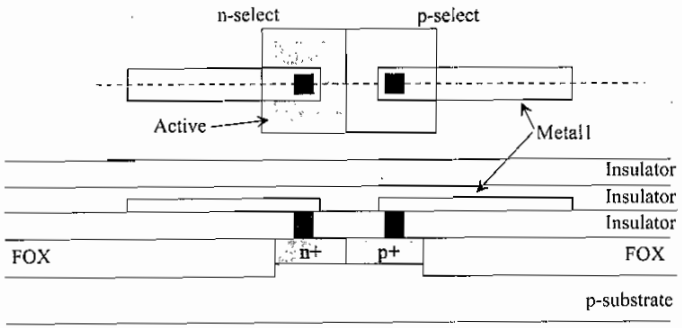
4.2 Connecting Wires to Poly and Active

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 (CONT), 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.

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.

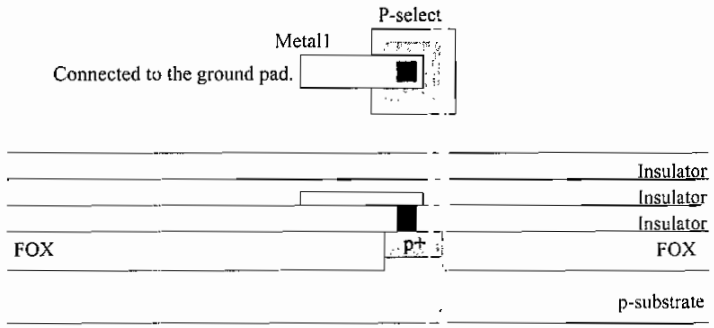


Figure 4.9 Connecting the substrate to ground.

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.

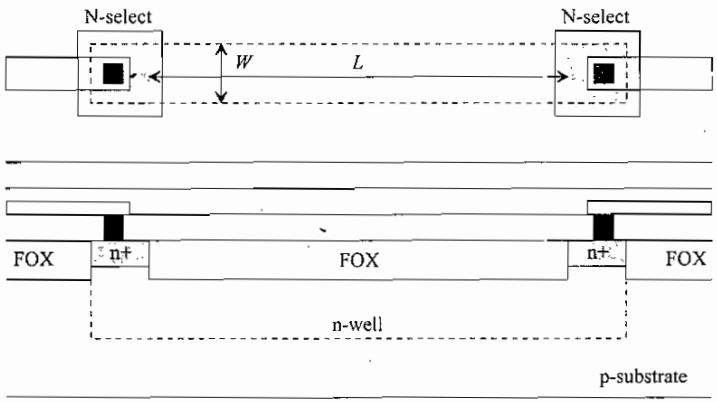


Figure 4.10 Layout of an n-well resistor and the corresponding cross-sectional view.

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. ■

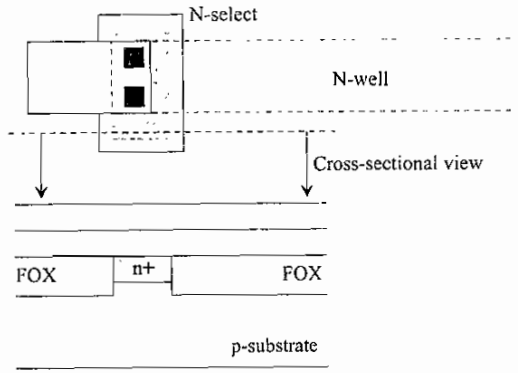


Figure 4.11 Active extending beyond the edges of the n-well.

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

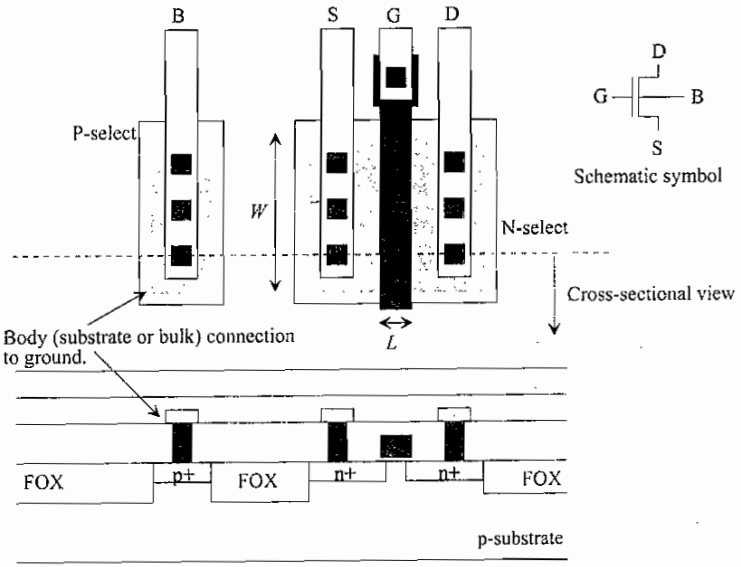


Figure 4.12 Layout and cross-sectional views of an NMOS device.

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.

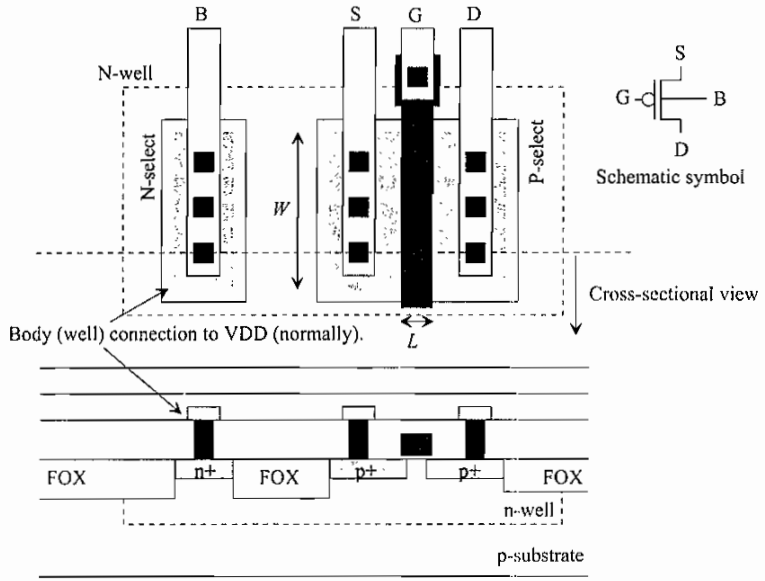


Figure 4.13 Layout and cross-sectional views of an PMOS device.

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,

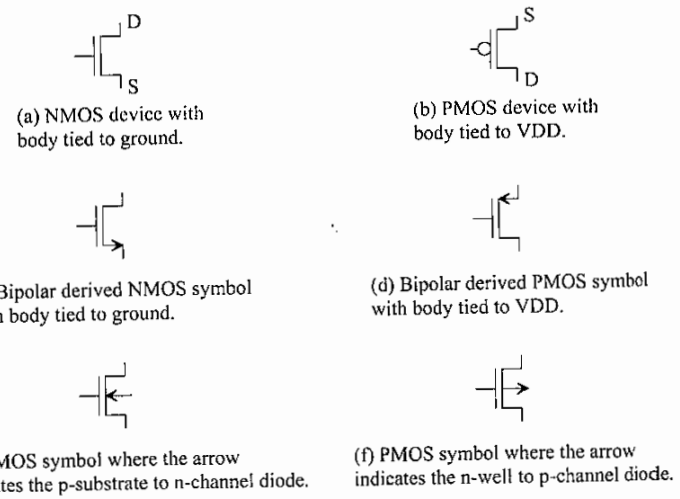


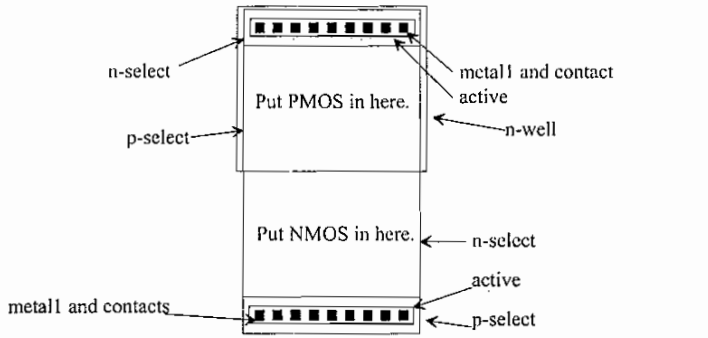
Figure 4.14 Symbols commonly found in schematics to represent a MOSFET.

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

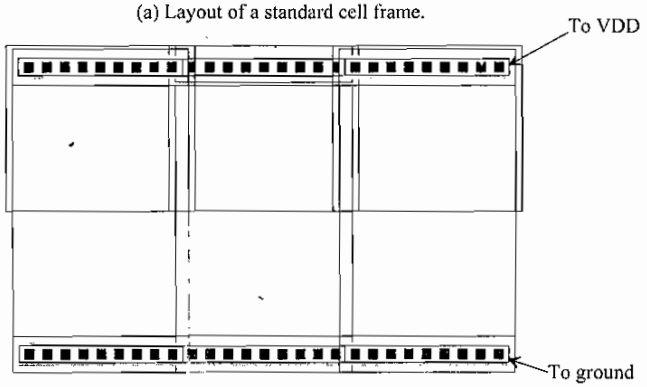
Standard Cell Frame

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 are 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).



(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) or with the LASI setups. 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).

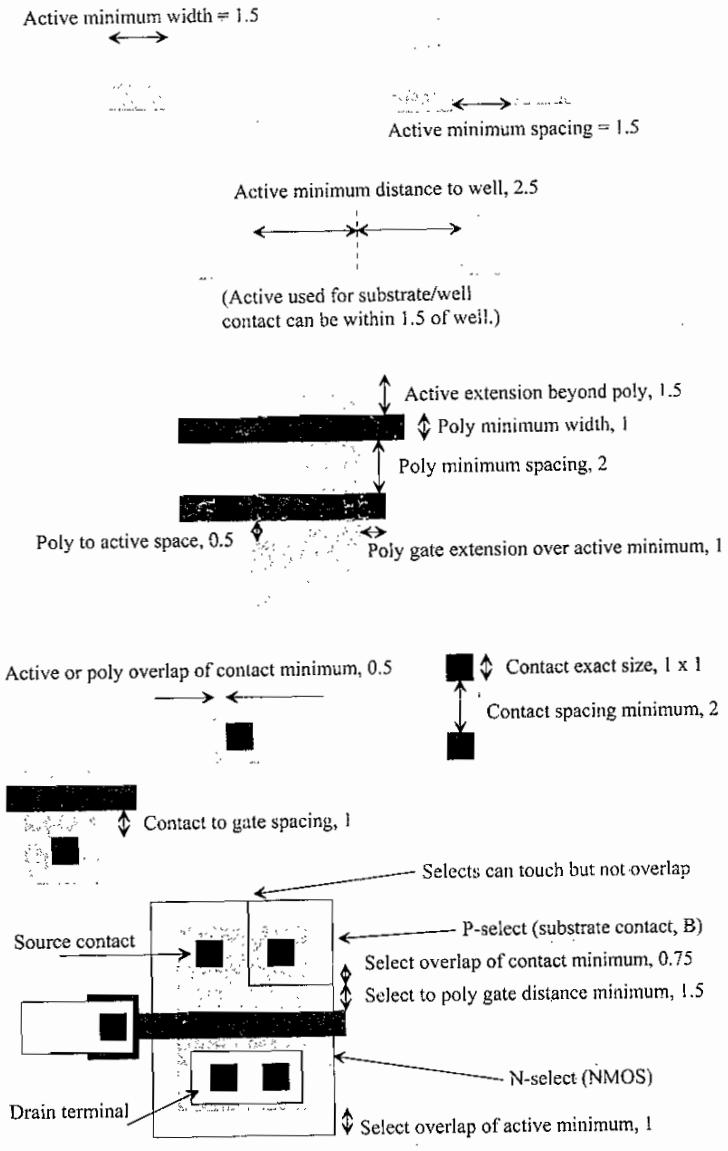


Figure 4.16 Design rules for active, selects, poly, and contacts.

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.

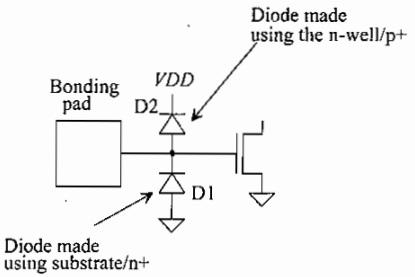


Figure 4.17 Adding diodes to the pads to protect the MOSFET gates from ESD damage.

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

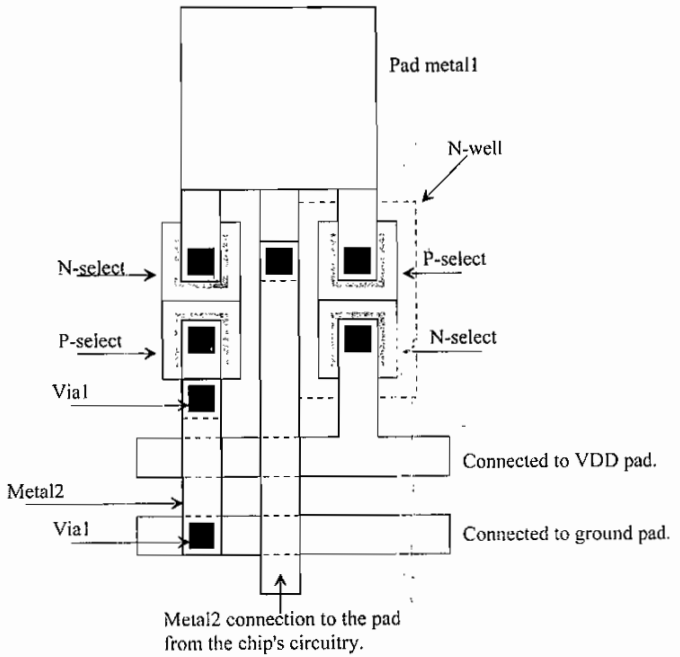


Figure 4.18 Conceptual layout of a pad ESD protection circuit.

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).

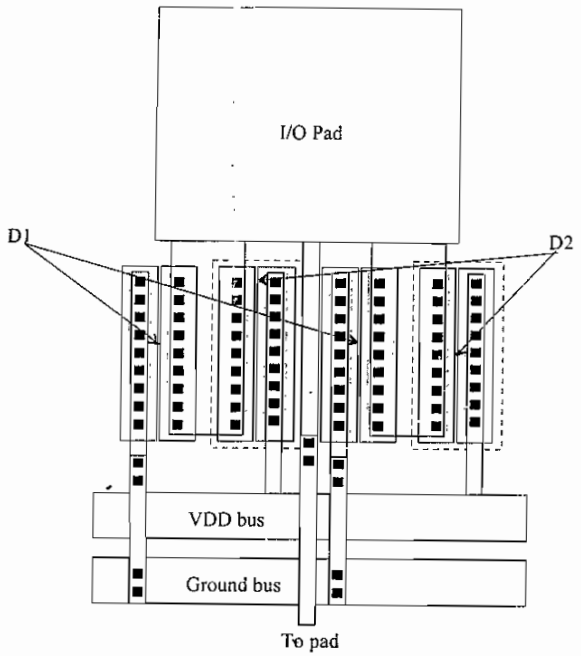


Figure 4.19 More detailed implementation of a pad with ESD protection.

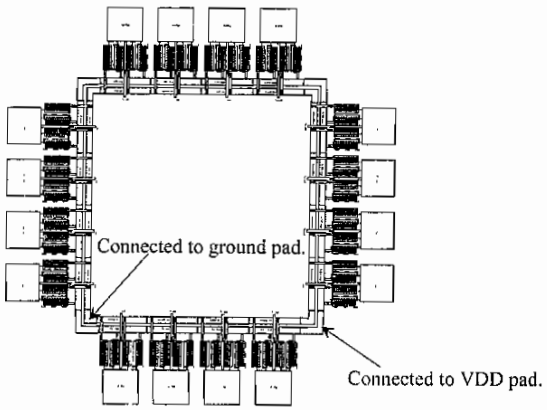


Figure 4.20 Layout of a padframe using pads with ESD diodes.

ADDITIONAL READING

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

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

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

[4] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, Cambridge University Press, 1998. ISBN 0-521-55959-6

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.

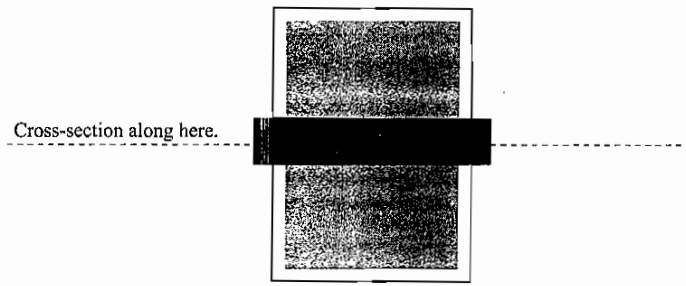


Figure 4.21 Layout used in Problem 4.2.

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.

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.

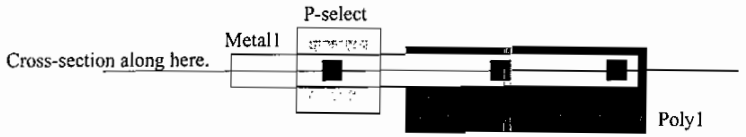


Figure 4.22 Layout used in Problem 4.10.

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

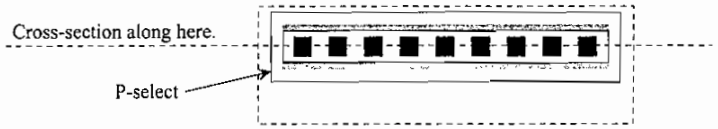


Figure 4.23 Layout used in Problem 4.11.

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

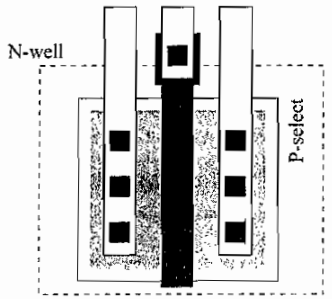


Figure 4.24 Flawed layout of a PMOS device. What is the error?

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

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

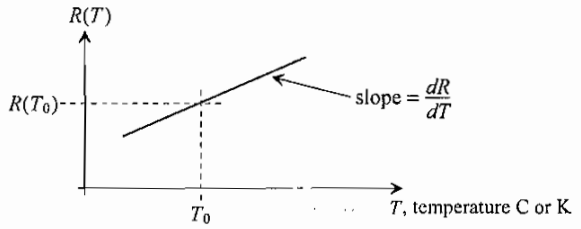


Figure 5.1 Resistor's value change with temperature.

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

$$TCR1 = \frac{1}{R} \cdot \frac{dR}{dT} \qquad (5.2)$$

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

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

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, μ_n or μ_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


```
*** 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
```

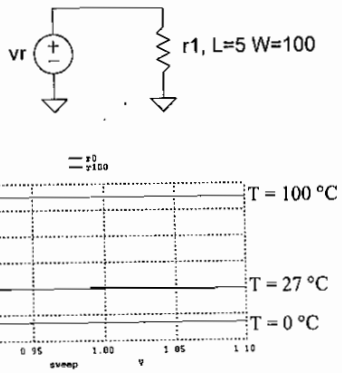


Figure 5.2 Simulating the temperature dependence of an n-well resistor.

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

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

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

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

$$VCR = \frac{1}{R} \cdot \frac{dR}{dV} \qquad (5.6)$$

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 50 k 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.

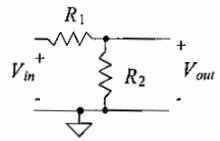


Figure 5.3 Comparing a resistive divider's temperature performance to its voltage performance.

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

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.

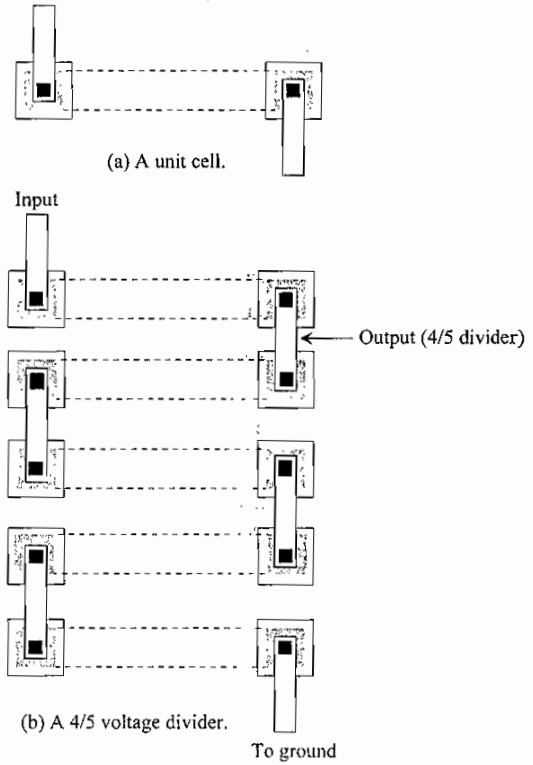


Figure 5.4 (a) Layout of a unit resistor cell, and (b) layout of a divider.

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*.

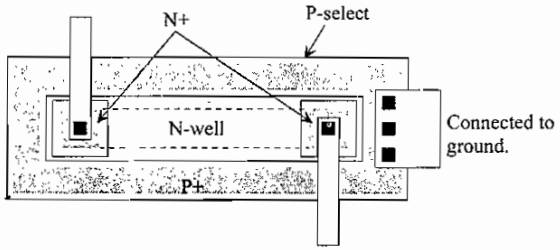


Figure 5.5 Guard ringing an n-well resistor.

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.

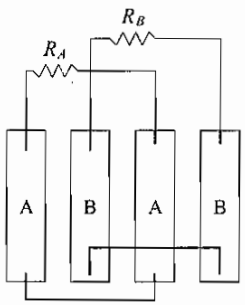


Figure 5.6 Layout of interdigitated resistors.

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.

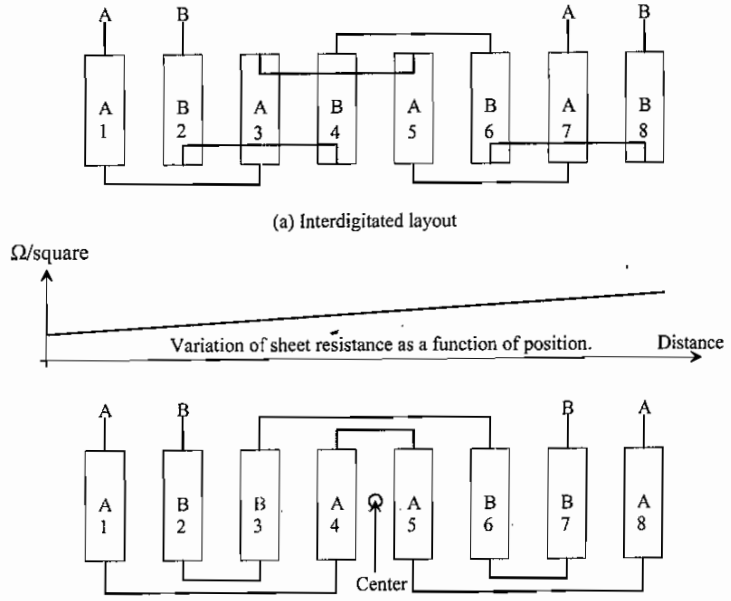


Figure 5.7 (a) Interdigitated layout and (b) common-centroid layout.

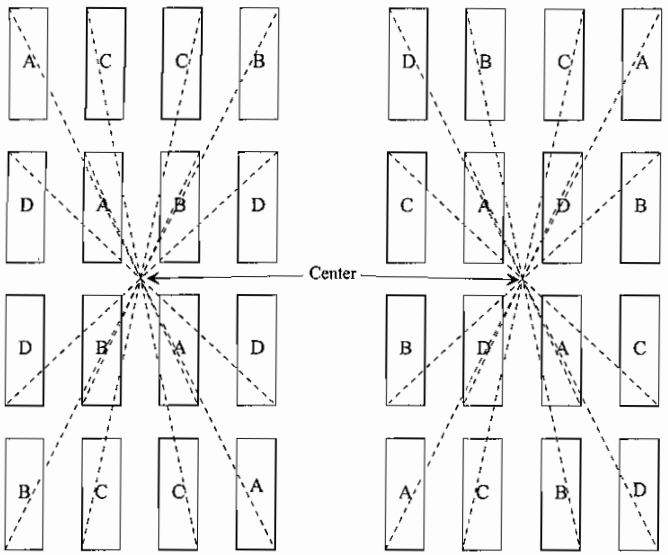


Figure 5.8 Common-centroid layout of four matched resistors (or elements).

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

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

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.

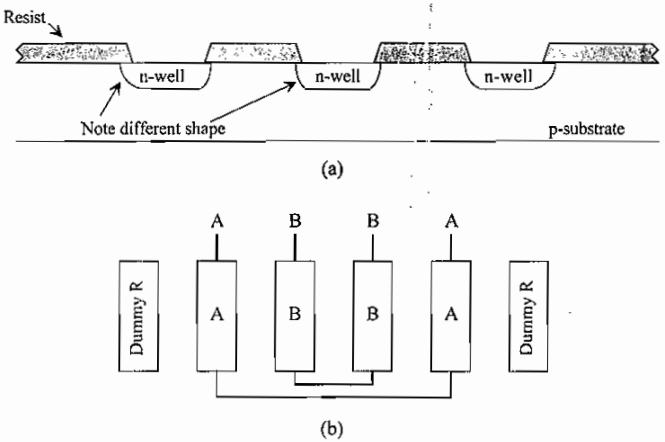


Figure 5.9 (a) Edge effects and (b) a common-centroid layout with dummy R.

5.2 Capacitors

An added layer of polysilicon, called poly2 (POL2), 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.

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

$$C'_{ox} = \frac{\varepsilon_r \cdot \varepsilon_0}{t_{ox}} = \frac{\varepsilon_{ox}}{t_{ox}} \qquad (5.8)$$

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

$$C_{ox} = C'_{ox} \cdot A \qquad (5.9)$$

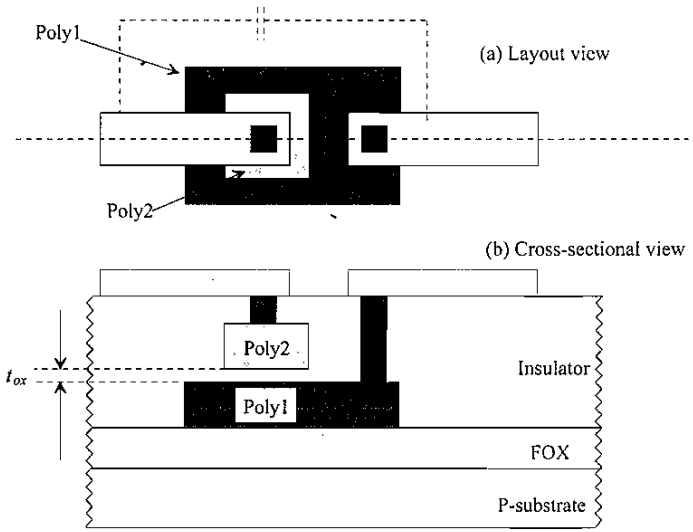


Figure 5.10 Layout and cross-sectional view of a poly-poly capacitor.

If the scale factor is included, then we can write this equation as

$$C_{ox} = C'_{ox} \cdot A \cdot (scale)^2 \qquad (5.10)$$

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

$$C = C_{ox} = 25\,fF/\mu m^2 \cdot 200 \cdot (0.05\ \mu m)^2 = 12.5\,fF \qquad (5.11)$$

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	$1.75\,fF/\mu m^2$
50 nm (short channel)	14 Å	$25\,fF/\mu m^2$

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.

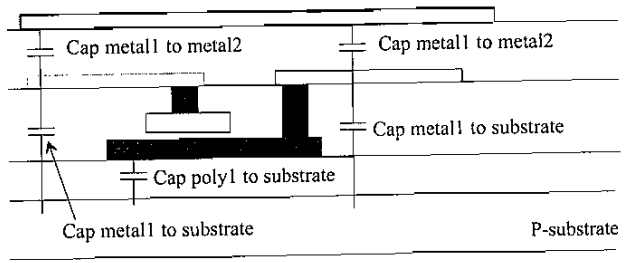


Figure 5.11 The parasitic capacitances of the poly-poly capacitor.

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 $aF/\mu m^2$ (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.

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.

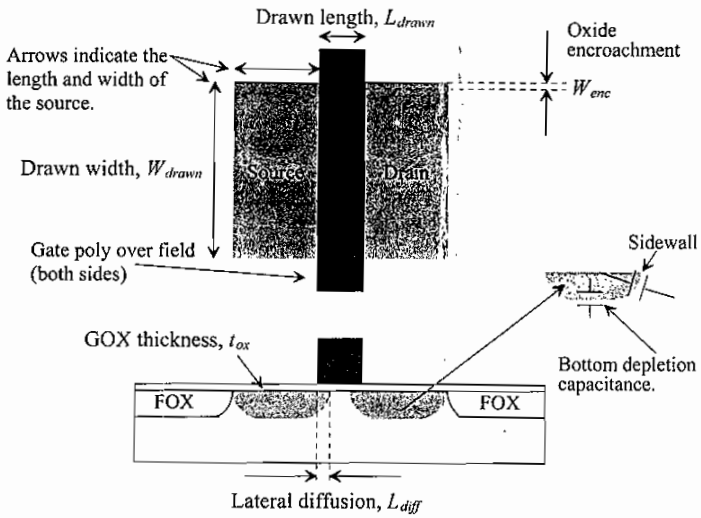


Figure 5.13 Lateral diffusion in a MOSFET.

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

$$C_{js,d} = \frac{cj \cdot A_{S,D} \cdot (scale)^2}{\left(1 + \frac{V_{S,DB}}{pb}\right)^{mj}} + \frac{cjsw \cdot P_{S,D} \cdot (scale)}{\left(1 + \frac{V_{S,DB}}{pbsw}\right)^{mjsw}} \tag{5.17}$$

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).

Note that some authors 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.

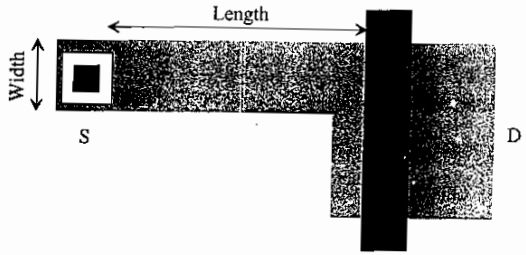


Figure 5.14 Determining a resistance in series with the drain of the MOSFET.

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).

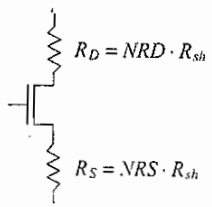


Figure 5.15 How source/drain series parasitic resistance is modeled in SPICE.

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

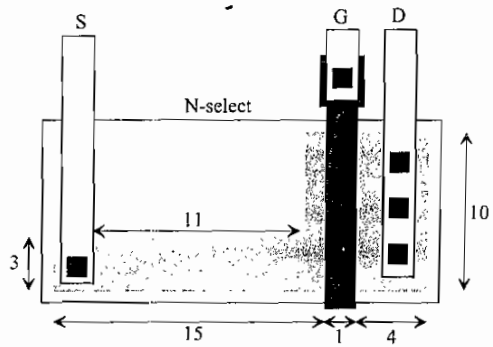


Figure 5.16 Layout of the MOSFET used in Ex. 5.5.

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 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.

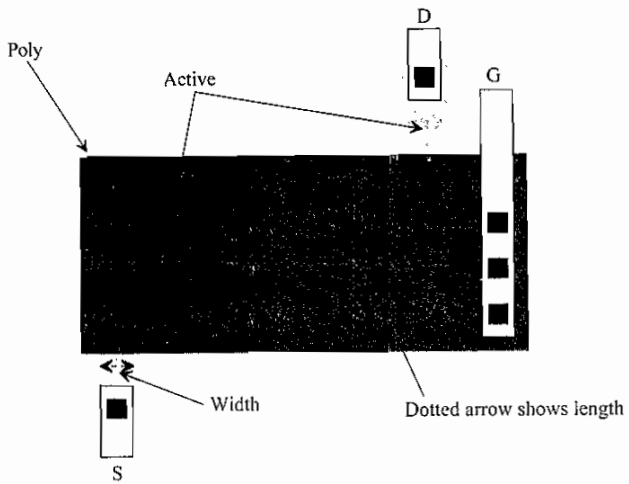


Figure 5.17 Layout of a long-length MOSFET.

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"

$$\text{Width of MOSFET} = (\text{number of fingers}) \cdot W \qquad (5.20)$$

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

$$\text{Width-to-length ratio} = W/L = 10/2 \qquad (5.21)$$

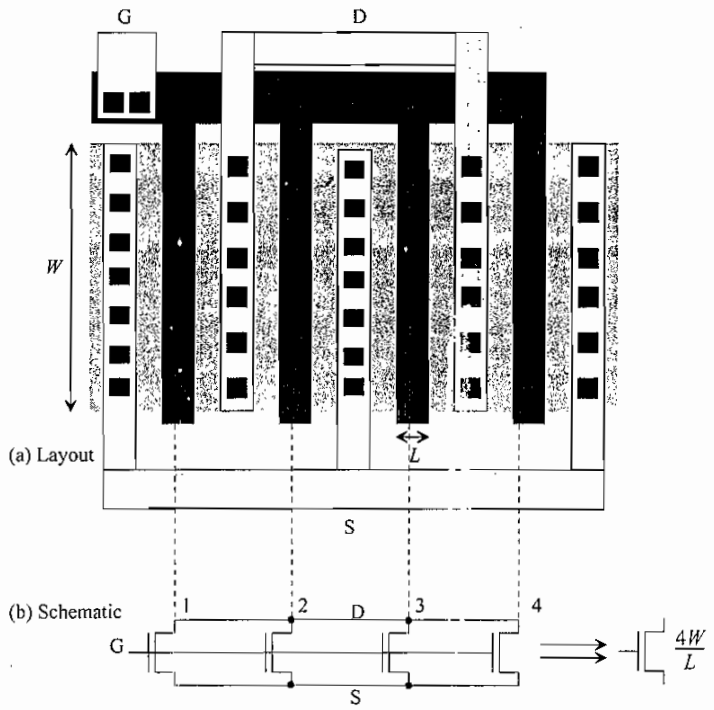


Figure 5.18 Layout and equivalent schematic of a large-width MOSFET.

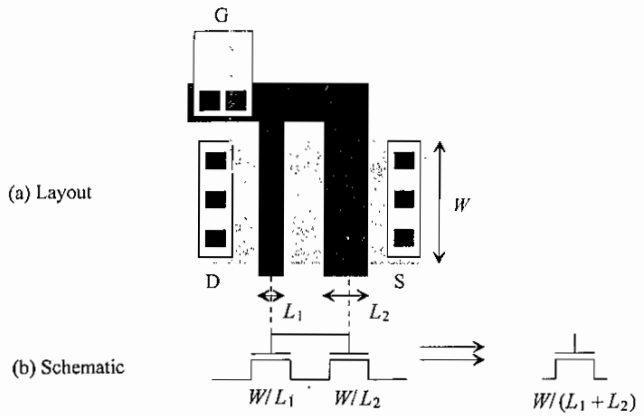
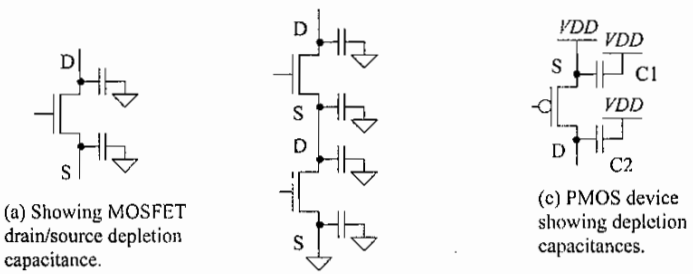


Figure 5.19 MOSFETs in series with gates tied together behave as a single MOSFET with the sum of the lengths.

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



(a) Showing MOSFET drain/source depletion capacitance.

(b) Series connection of MOSFETs.

(c) PMOS device showing depletion capacitances.

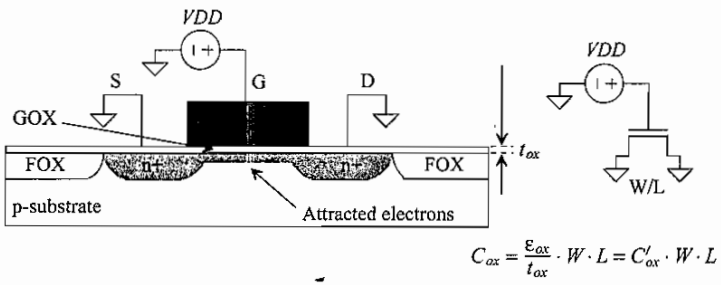
Figure 5.20 Showing depletion capacitance on the MOSFET symbols.

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*.

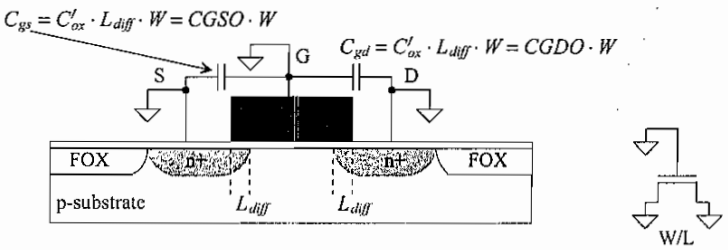
A Qualitative Description of MOSFET Capacitances

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

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \cdot W \cdot L = C'_{ox} \cdot W \cdot L \qquad (5.22)$$



(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.

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.

5.4 LASI Design Examples

In this section we present some additional design examples using LASI. We'll show how to draw both schematics and layouts. Then we'll compare the two cells to make sure that the layout matches the schematic (layout versus schematic, LVS). Some things to remember and know when using LasiCkt:

1. Basic SPICE device cells like resistors, capacitors, and MOSFETs are always rank 1 cells. We can't lay out a MOSFET or resistor in a rank 2 cell. This means we can't use a layout like the via1 cell seen in Fig. 3.19 in a MOSFET layout.

2. The device text (e.g., for a resistor name R1) and parameter text (e.g., specifying a resistor's value, 2k) must be placed within the perimeter of the cell they are labeling. It is useful to show the cell's image by pressing **i** on the keyboard.

3. Connectors (connector text) are used on a cell to connect the cell to wires (e.g., metal1) in higher ranking cells. For example, connector text is placed in rank 1 cell, like a resistor, to show where to connect the wires in a rank 2 (or higher) cell.

4. Care must be exercised if cells are placed in a layout where they overlap. The parameter text or device text shouldn't be placed in the overlap area.

5. When drawing schematics, zero-width paths (poly object) should be used with the metal1 layer to connect the cells. Boxes (zero area) should not be present in a schematic. To verify that there aren't any boxes in a schematic, use the **Wget** command on the entire schematic cell followed by the **Info** command.

6. Parameter text (ptxt) labels on rank 2 and higher cells are always the cell's name. Rank one cells have user inputs. For example, a resistor may have a ptxt value of "2k" (the resistor's value) and a MOSFET may use "NMOS L=1 W=10".

A Resistor Divider

An example schematic, similar to the one we implemented back in Fig. 3.24, is seen in Fig. 5.22. Here, we use 2 kΩ unit cells to implement a divider (a 2/3 divider). The corresponding layout is seen in Fig. 5.23. For every text label (node text, parameter text, connector text, and device text) in the schematic there is an identical label in the layout. Once the schematic is drawn, we can use the **Save** command to save the schematic. We can then use the **Save** command again to save the schematic as a layout (followed by deleting all contents of the cell but the text). We can then place the layout of the unit resistor into the cell and connect the cells together with metal1 boxes (and then move the text into the desired locations). **Note** the metal1 boxes must overlap the connectors on the cells. Figure 5.24 shows the rank 2 layout in Fig. 5.23 redrawn with the cells drawn as outlines. The metal1 in the rank 2 cell must overlap the connector in the lower ranking cell (the resistor layout).

One of the important additions to the schematic and layout in Figs. 5.22–5.24 is node text. We use node text with a header file to specify the inputs and outputs of the circuit. A header file may look like

```
.control
destroy all
run
plot vin vout
.endc
.dc vin 0 1 1m
vin vin 0 DC 0
```

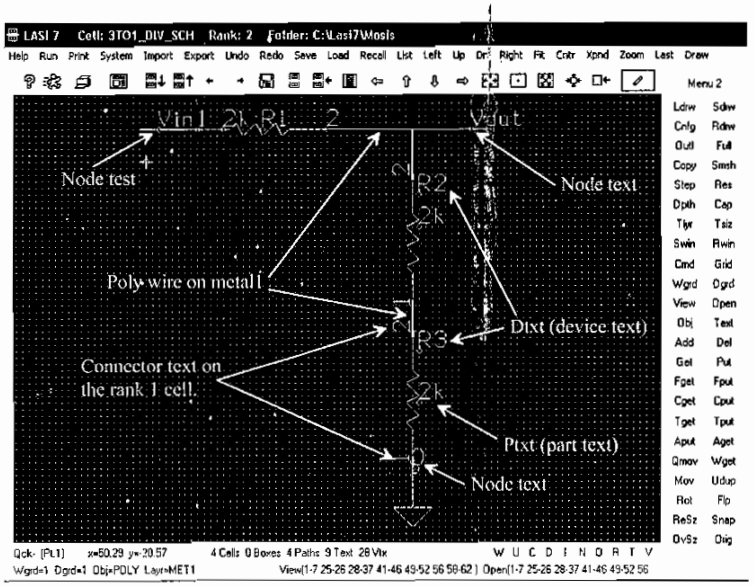


Figure 5.22 Schematic of a resistive divider.

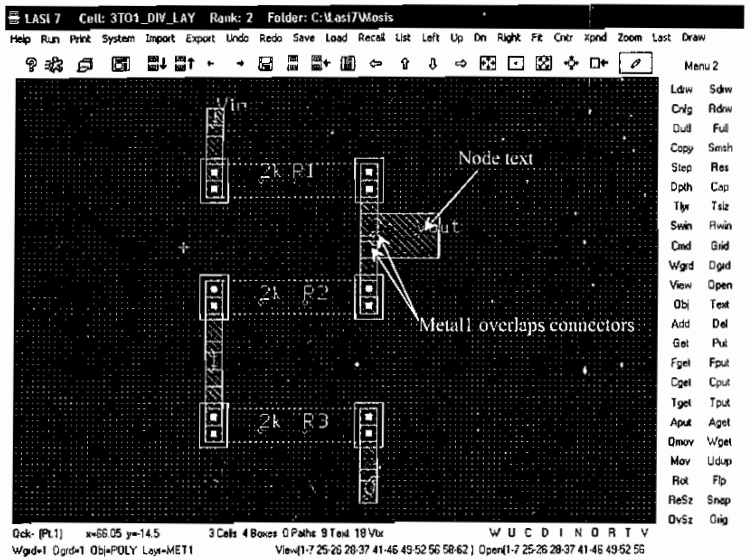


Figure 5.23 Layout corresponding to the schematic in Fig. 5.22.

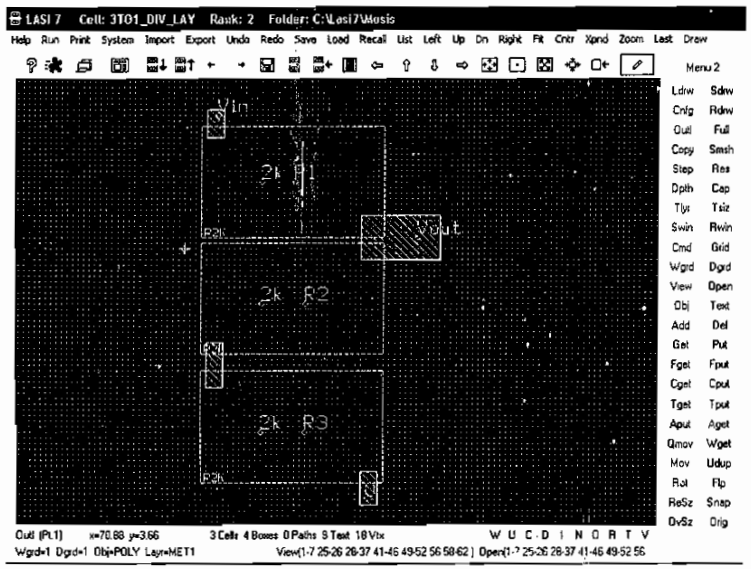


Figure 5.24 Drawing the cells in Fig. 5.23 as outlines.

Layout of a MOSFET

Figure 5.25 shows the schematic of an NMOS transistor with wire connections. This schematic representation is useful for a MOSFET test structure (used to measure the current-voltage characteristics of the MOSFET). The corresponding layout is seen in Fig. 5.26. These cells, as well as the others seen in this book, are available in the file cmosedu_lasi.zip available at cmosedu.com.

An **important point** to note in Fig. 5.25 or any other schematic or layout implemented in LASI is that placing node text directly on (the vertices at the same location) a connector plugs the connector. In other words, if we plug the connector, then running a wire over it has no effect. This can lead to frustration when drawing schematics or layouts and then extracting a netlist.

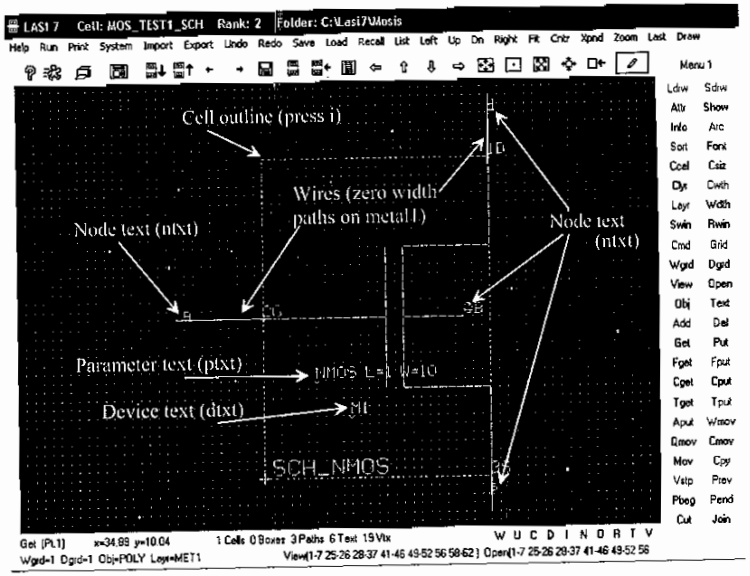


Figure 5.25 Schematic of an NMOS transistor and wire connections.

To simulate the circuit in Fig. 5.25, we can use the following header file (in the file Mos_test1.txt)

```
.control
destroy all
run
let id=-i(vds)
plot id
.endc
```

```
vds d s DC 0
vgs g s DC 0

.dc vds 0 1 1m vgs 0 1 0.25

*Important
.options scale=50nm

*ground the source
Vs  s 0 DC 0
```

We specify the voltage sources in the schematic and the type of analysis. Here, in this header, we sweep the drain-source voltage from 0 to 1 V in 1 mV steps, while the gate-source voltage is stepped in 250 mV increments. Notice the statement showing the scale factor (allowing us to draw the layout with an integer grid). Not remembering to include this specification in a header file can lead to wasted time when running simulations.

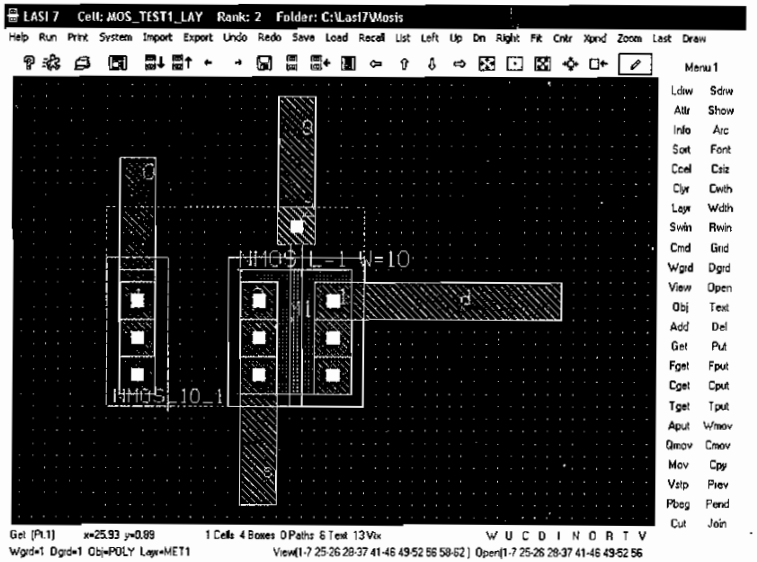


Figure 5.26 Layout of the MOSFET in Fig. 5.25.

Figure 5.27 shows the LasiCkt setup screen for extracting a SPICE netlist from the layout in Fig. 5.26. Note how we specified SPICE models for the footer file. These SPICE models can be downloaded from cmosedu.com. Table 5.2 shows the SPICE file names for the long- and short-channel CMOS processes that **we use in this book**. For the layout in Fig. 5.26, we use the 50 nm CMOS process.

We can extract SPICE netlists from either the layout or the schematic. The SPICE netlists can be compared and the simulation results can be used to perform a heuristic LVS. However, a more complete LVS can be performed by using the node list files in

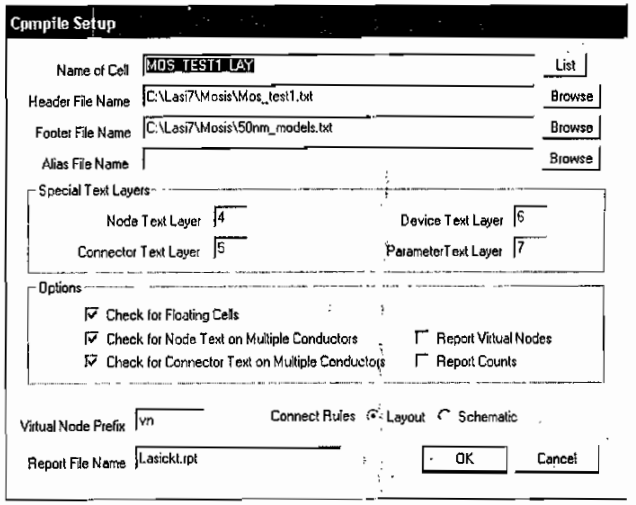


Figure 5.27 LasiCkt setup screen (for extracting the SPICE netlist from a layout).

LASI. By pressing the LVS command in LasiCkt and selecting the file names for the two cells to be compared, LasiCkt can perform an LVS.

Table 5.2 Table listing the SPICE file names and corresponding scale factors (with the .options statement SPICE) used in this book.

CMOS technology	Scale factor used with .options	SPICE file name	*VDD*
1 µm (long channel)	1um	1u_models.txt	5 V
50 nm (short channel)	50nm	50nm_models.txt	1 V

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.

[3] D. E. Boyce, *LASI User's Manual*, available while LASI is running by the help menu item. This help file (which includes links to help for LasiCkt) can also be accessed by opening the lhh7 file in C:\Lasi7\.

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.

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.

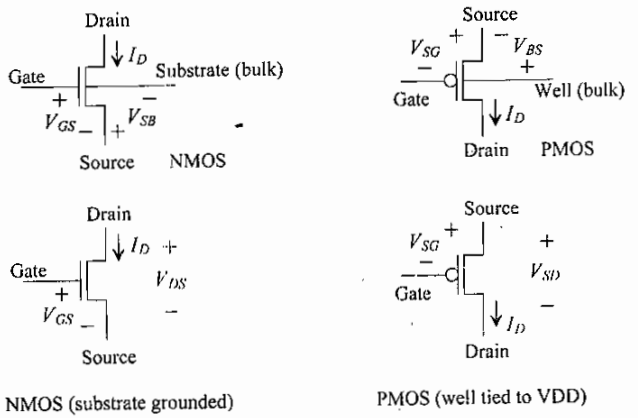


Figure 6.1 Voltage and current designations for MOSFETs in this chapter.

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 ε_{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.

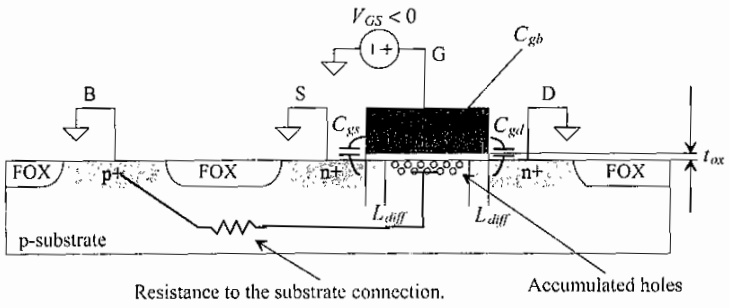


Figure 6.2 Cross-sectional view of a MOSFET operating in accumulation.

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+.

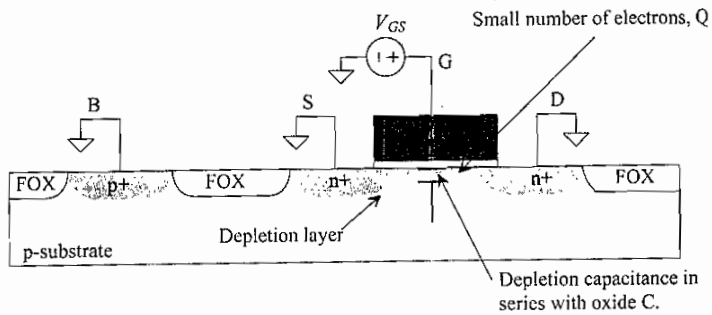


Figure 6.3 Cross-sectional view of a MOSFET operating in depletion.

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.

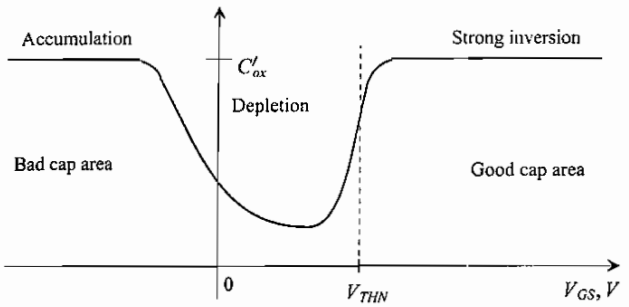


Figure 6.4 The variation of the gate capacitance with DC gate-source voltage.

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?

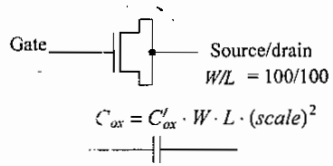


Figure 6.5 Using the MOSFET as a capacitor.

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. ■

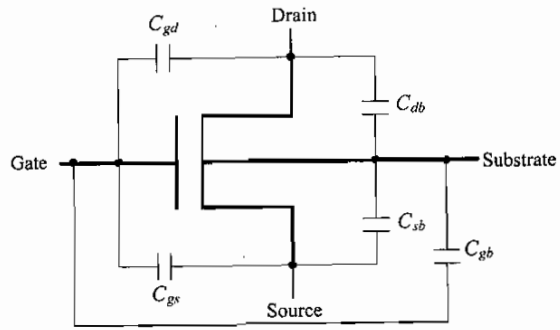


Figure 6.6 MOSFET capacitances.

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

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 depletion 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

$$V_{fp} = -\frac{kT}{q}\ln\frac{N_A}{n_i} = -26\text{ mV}\cdot\ln\frac{10^{15}}{14.5\times10^9} = -290\text{ mV}$$

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 $V_{GS} = V_{THN}$. ■

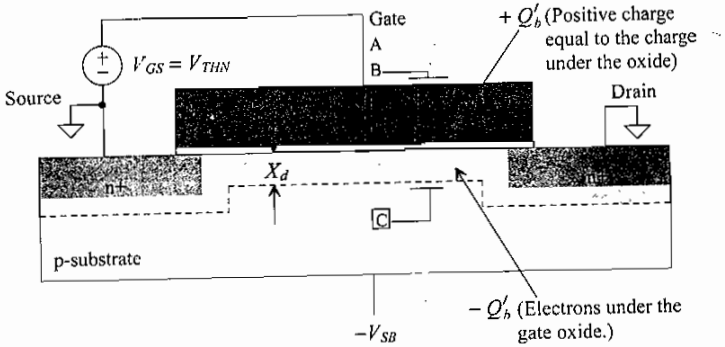


Figure 6.7 Calculation of the threshold voltage.

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 (= ΔV_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

$$V_B = \frac{Q'_b}{C'_{ox}} - 2V_{fp} \tag{6.11}$$

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

$$V_{THN} = \frac{Q'_b - Q'_{ss}}{C'_{ox}} - 2V_{fp} - V_{ms} \tag{6.14}$$

$$= -V_{ms} - 2V_{fp} + \frac{Q'_{bo} - Q'_{ss}}{C'_{ox}} - \frac{Q'_{bo} - Q'_b}{C'_{ox}} \tag{6.15}$$

$$= -V_{ms} - 2V_{fp} + \frac{Q'_{bo} - Q'_{ss}}{C'_{ox}} + \frac{\sqrt{2q\varepsilon_{si}N_A}}{C'_{ox}}\left[\sqrt{|2V_{fp}| + V_{SB}} - \sqrt{|2V_{fp}|}\right] \tag{6.16}$$

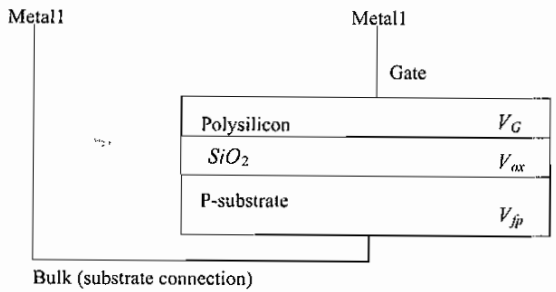


Figure 6.8 Determining the contact potential between poly and substrate.

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

$$\gamma = \frac{\sqrt{2 \cdot 1.6\times10^{-19}\frac{columbs}{atom} \cdot 11.7 \cdot 8.85\frac{aF}{\mu m} \cdot 10^{16}\frac{atoms}{cm^3} \cdot \frac{cm^3}{10^{12}\mu m^3}}}{1.75\frac{fF}{\mu m^2}} = 0.330\ V^{1/2} \quad \blacksquare$$

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

$$-V_{ms} = V_{fp} - V_G = -290\ mV - 26\ mV \cdot \ln\frac{10^{20}}{14.5\times10^{9}} = -879\ mV$$

$$-2V_{fp} = 580\ mV$$

$$\frac{Q'_{bo}}{C'_{ox}} = \frac{139\ aC/\mu m^2}{1.75\ fF/\mu m^2} = 79\ mV$$

$$\frac{Q'_{ss}}{C'_{ox}} = 23\ mV$$

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).

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

$$dR = \overbrace{\frac{1}{\mu_n Q'_I(y)}}^{\text{eff. sheet Res.}} \cdot \frac{dy}{W} \tag{6.27}$$

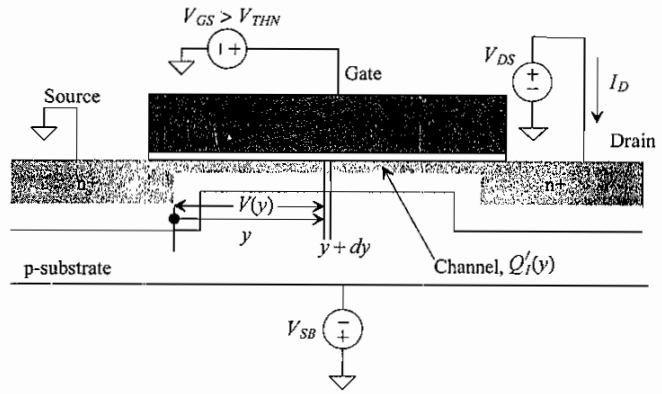


Figure 6.9 Calculation of the large-signal behavior of the MOSFET in the triode (ohmic) region.

where μ_n is the average electron mobility through the channel with units of cm²/V·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

$$dV(y) = I_D \cdot dR = \frac{I_D}{W\mu_n Q_i'(y)} \cdot dy \qquad (6.28)$$

or substituting Eq. (6.26) and rearranging

$$I_D \cdot dy = W\mu_n C_{ox}'(V_{GS} - V(y) - V_{THN}) \cdot dV(y) \qquad (6.29)$$

At this point, let's define the transconductance parameter, *KP*, for a MOSFET. For an n-channel MOSFET, this parameter is given by

$$KP_n = \mu_n \cdot C_{ox}' = \mu_n \cdot \frac{\varepsilon_{ox}}{t_{ox}} \qquad (6.30)$$

and for a p-channel MOSFET, it is given by

$$KP_p = \mu_p \cdot C_{ox}' = \mu_p \cdot \frac{\varepsilon_{ox}}{t_{ox}} \qquad (6.31)$$

where μ_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 μA/V² and 40 μA/V² 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:

$$I_D \int_0^L dy = W \cdot KP_n \cdot \int_0^{V_{DS}} (V_{GS} - V(y) - V_{THN}) \cdot dV(y) \qquad (6.32)$$

or

$$I_D = KP_n \cdot \frac{W}{L} \cdot \left[(V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2} \right] \text{ for } V_{GS} \geq V_{THN} \text{ and } V_{DS} \leq V_{GS} - V_{THN} \qquad (6.33)$$

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

$$\beta = KP_n \cdot \frac{W}{L} \qquad (6.34)$$

or

$$I_D = \beta \cdot \left[(V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2} \right] \qquad (6.35)$$

The equivalent equation for the PMOS device operating in the triode region is

$$I_D = KP_p \cdot \frac{W}{L} \cdot \left[(V_{SG} - V_{THP})V_{SD} - \frac{V_{SD}^2}{2} \right] \text{ for } V_{SG} \geq V_{THP} \text{ and } V_{SD} \leq V_{SG} - V_{THP} \qquad (6.36)$$

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 (again, pinching off the channel). Further increases in V_{DS} do not cause an increase in the drain current[1]. Figure 6.10 shows that the depletion region, with a thickness of X_{dl}, between the drain and substrate increases, causing the channel to pinch off. If V_{DS} is increased until the drain-substrate depletion region 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+) implant/substrate diode.

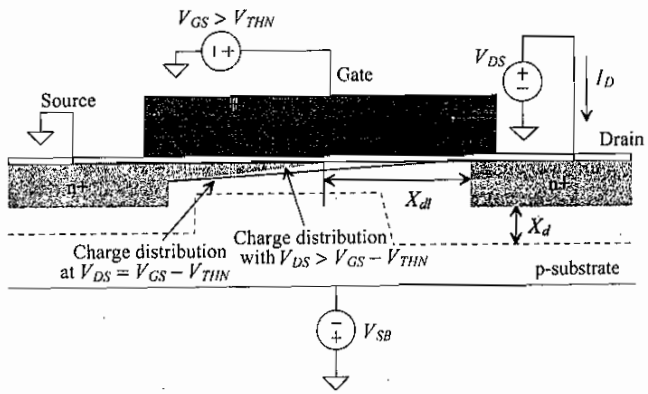


Figure 6.10 The MOSFET in saturation (pinched off).

[1] We will see that this is not entirely true. An effect called channel length modulation causes the drain current to increase with increasing drain-source voltage.

When a MOSFET is operated with its channel 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

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN})^2 = \frac{\beta}{2}(V_{GS} - V_{THN})^2$$

$$\text{for } V_{DS} \geq V_{GS} - V_{THN} \text{ and } V_{GS} \geq V_{THN} \qquad (6.37)$$

We can define an electrical channel length of the MOSFET as the difference between the drawn channel length, neglecting lateral diffusion, and the depletion layer width, X_{dl}, between the drain n+ and the channel under the gate oxide by

$$L_{elec} = L_{drawn} - X_{dl} \qquad (6.38)$$

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). Note that as L_{drawn} 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}, or

$$\frac{\partial I_D}{\partial V_{DS}} = -\frac{KP_n}{2}\frac{W}{L_{elec}^2}(V_{GS} - V_{THN})^2 \cdot \frac{dL_{elec}}{dV_{DS}} = I_D \cdot \left[\frac{1}{L_{elec}}\frac{dX_{dl}}{dV_{DS}}\right] \qquad (6.40)$$

where it is common to let

$$\lambda = \frac{1}{L_{elec}} \cdot \frac{dX_{dl}}{dV_{DS}} \qquad (6.41)$$

Typical values for λ, called the channel length modulation parameter, range from greater than 0.1 V^{-1} for short-channel devices to 0.01 V^{-1} for long-channel devices. Equation (6.37) can be rewritten for a device operating in the saturation region, taking into account channel length modulation as

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - V_{THN})^2[1 + \lambda(V_{DS} - V_{DS,sat})]$$

$$\text{for } V_{DS} > V_{DS,sat} = V_{GS} - V_{THN} \text{ and } V_{GS} > V_{THN} \qquad (6.42)$$

When the drain-source voltage is $V_{DS,sat}$ (the drain current at the triode/saturation region border), the drain current is sometimes specified as

$$I_{D,sat} = I_D \text{ when } V_{DS} = V_{DS,sat} = V_{GS} - V_{THN} \qquad (6.43)$$

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 $I_{D,sat}$).

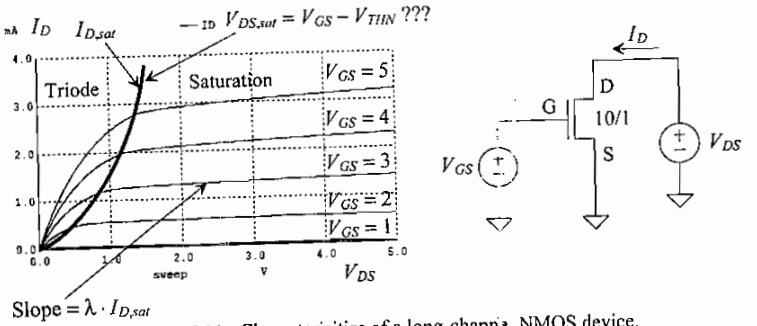


Figure 6.11 Characterisitics of a long-channe NMOS device.

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,

$$Q_I = \int_0^L W \cdot Q'_I(y) \cdot dy = WC'_{ox}\int_0^L (V_{GS} - V(y) - V_{THN})\,dy \qquad (6.44)$$

or solving Eq. (6.29) for dy and substituting yields

$$Q_I = \frac{(W \cdot C'_{ox})^2 \cdot \mu_n}{I_D}\int_0^{V_{GS}-V_{THN}} (V_{GS} - V(y) - V_{THN})^2 \cdot dV(y) \qquad (6.45)$$

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

$$Q_I = \frac{2}{3} \cdot W \cdot L \cdot C'_{ox} \cdot (V_{GS} - V_{THN}) \qquad (6.46)$$

We can determine the gate-to-source capacitance while in the saturation region by,

$$C_{gs} = \frac{\partial Q_I}{\partial V_{GS}} = \frac{2}{3} \cdot W \cdot L \cdot C'_{ox} \qquad (6.47)$$

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 consider the level 1 SPICE model and how it relates to the equations derived in the last section. The level 1 model is a subset of the level 2 and 3 models.

Level 1 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
γ	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 γ 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<Ldrawn<200 10<Wdrawn<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
```

Level 1 Model Parameters Related to Transconductance

The following SPICE parameters are related to the calculation of transconductance.

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
λ	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

The following SPICE model parameters are related to calculating the parasitics associated with the source/drain implant regions.

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 shows a summary of the device characteristics for the long-channel CMOS process.

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 fF/µm²	1.75 fF/µm²	$C_{ox} = C'_{ox}WL \cdot (scale)^2$
λ_n and λ_p	0.01 V^{-1}	0.0125 V^{-1}	at $L = 2$
γ_n and γ_p	0.5 $V^{-1/2}$	0.6 $V^{-1/2}$	Body factor

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.

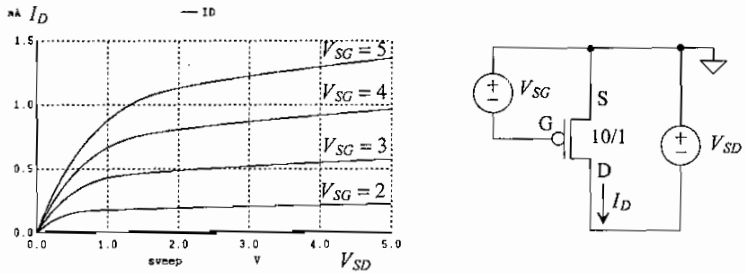


Figure 6.12 Characterisitics of a long-channel PMOS device.

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).

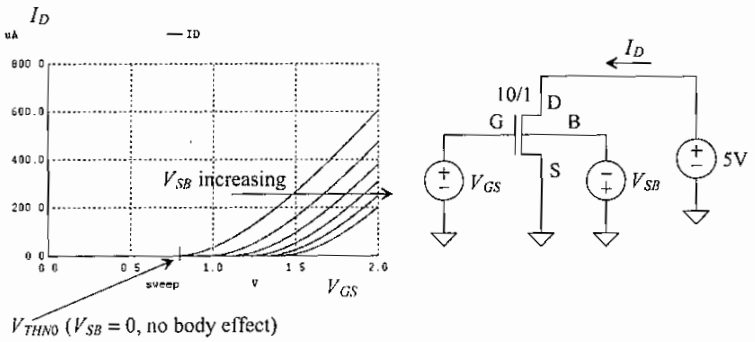


Figure 6.13 Threshold voltage and body effect.

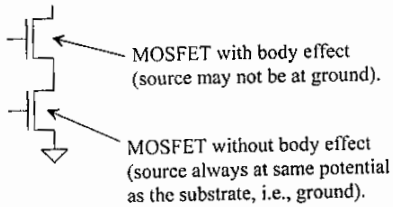


Figure 6.14 How an NMOS can have body effect.

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's threshold voltage.

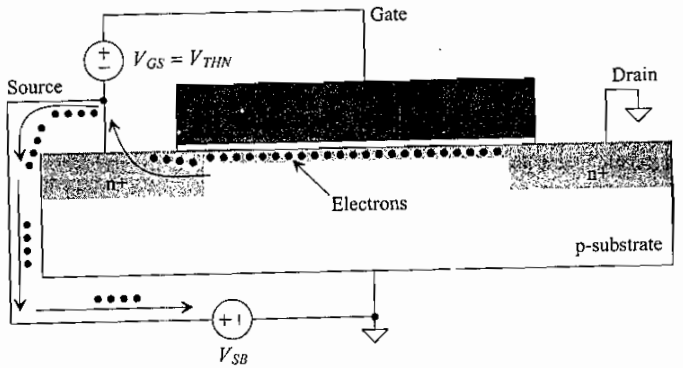


Figure 6.15 Qualitative description of body effect.

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.

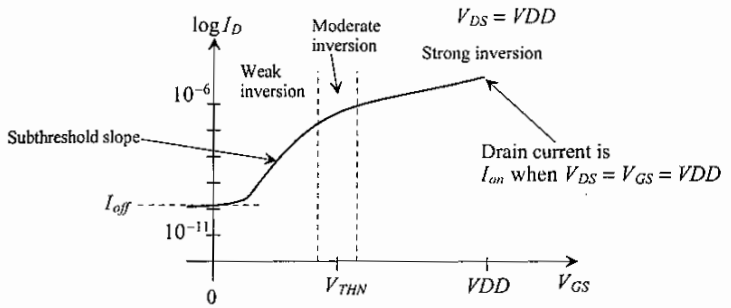


Figure 6.16 Drain current plotted from weak to strong inversion.

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}.

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 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 saturates, Fig. 6.17, an effect called *carrier velocity saturation*, v_{sat}. The result is a reduction in electron or hole mobility, μ_n or μ_p, and thus an increase in the channel's effective sheet resistance, Eq. (6.27). Since the mobility of the electron decreases with increasing temperature[2], this effect is sometimes referred to as the hot-carrier effect. 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 (μ_n) and 250 cm^2/Vs (μ_p). See also Eq. (5.4).

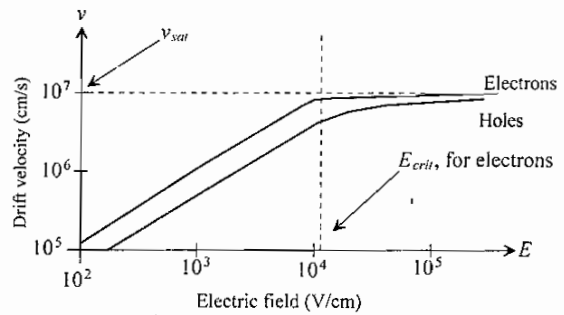


Figure 6.17 Drift velocity plotted against electric field. The slope of these curves is the mobility of the carriers, see Eq. (5.4)

Lightly-Doped Drain (LDD)

A cross-sectional view of an NMOS device using a lightly doped drain (LDD) structure to reduce the effects of hot carriers, is shown in Fig. 6.18. We also show this structure in Fig. 4.7. The LDD is formed by implanting a shallow n-, forming the spacer adjacent to the poly, and implanting the source/drain n+. The addition of the lightly doped n- provides a resistive buffer between the channel and the higher-doped source/drain. The effect of this resistive buffer is to drop more voltage, across a smaller distance, between the channel and the drain. In effect the electric field gets reduced (the drain doesn't see as high of an electric field). The effect, as seen in Fig. 6.17, is to keep (or at least help) the carriers' velocity from saturating.

[2] See Ch. 9 for a discussion of the temperature dependence of the mobility.

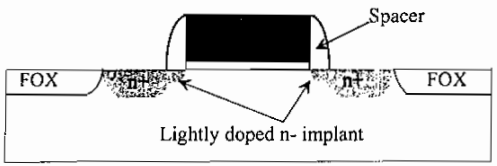


Figure 6.18 Lightly doped drain (LDD) implant.

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

6.5.2 Short-Channel Effects

The average drift velocity, v, of an electron plotted against electric field, E, is 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. 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.

Hot Carriers

The low field mobility decreases with increasing temperature (again see Ch. 9). This decrease in mobility can be used to describe the decrease in mobility associated with velocity saturation. Because of this, carriers traveling at v_{sat} are sometimes called hot carriers. Note that the drift velocity and temperature are not proportional. *Hot carriers* is simply a term used to velocity saturation and thus the associated reduction in mobility. Hot carriers can tunnel through the gate oxide and cause gate current or become trapped in the gate oxide, having the effect of changing the threshold voltage. Hot carriers can also cause impact ionization (avalanche breakdown).

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.

Substrate Current-Induced Body Effect

Substrate current-induced body effect, (SCBE), is the result of hot carriers causing impact ionization and generating a substrate current. This occurs for electric fields greater than 10^5 V/cm. The substrate current flows through the resistance of the substrate, increasing the substrate potential and further decreasing the threshold voltage. The result, again, is an increase in drain current and a decrease in the MOSFET's output resistance.

Gate Tunnel Current

As the oxide thickness scales downwards, the probability of carriers 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 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[3] 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
* Change to level=54 when using HSPICE

.model      nmos     nmos     level = 14

+binunit = 1          paramchk= 1          mobmod  = 0
+capmod  = 2          igcmod  = 1          igbmod  = 1          geomod  = 1
+diomod  = 1          rdsmod  = 0          rbodymod= 1          rgatemod= 1
```

[3] 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


```
+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
```

```
+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 = 14

+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
```



```
+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

$$I_{on,p} \approx 150\ \mu A/(W \cdot scale) = 300\ \mu A/\mu m \tag{6.61}$$

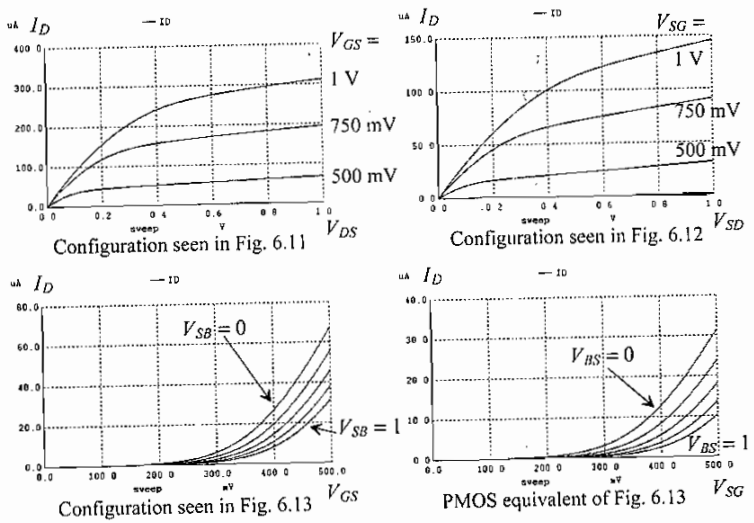


Figure 6.19 Current-voltage characteristics for 50 nm MOSFETs.

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	$C_{ox} = C'_{ox}WL \cdot (scale)^2$
λ_n and λ_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] 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] W. Liu, "MOSFET Models for Spice Simulation, Including BSIM3v3 and BSIM4," John Wiley and Sons Publishers, 2001. ISBN 0-471-39697-4

[3] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, Cambridge University Press, 1998. ISBN 0-521-55056-4 (hardback) or ISBN 0-521-55959-6 (paperback).

[4] D. Foty, *MOSFET Modeling with SPICE: Principles and Practice,* Prentice-Hall, 1997. ISBN 0-13-227935-5.

[5] M. Bohr, "MOS Transistors: Scaling and Performance Trends," *Semiconductor International*, pp. 75–79, June 1995.

[6] D. A. Neamen, *Semiconductor Physics and Devices-Basic Principles,* Richard D. Irwin, 1992. ISBN 0-256-08405-X.

[7] R. F. Pierret, *Volume IV in the Modular Series on Solid State Devices-Field Effect Devices,* Addison-Wesley, 1990.

[8] R. C. Jaeger, *Modular Series on Solid State Devices-Introduction to Microelectronic Fabrication,* Addison-Wesley, 1989.

[9] D. K. Schroder, *Modular Series on Solid State Devices-Advanced MOS Devices,* Addison-Wesley, 1987.

[10] Y. P. Tsividis, *Operation and Modeling of the MOS Transistor,* McGraw-Hill, 1987. ISBN 0-07-065381-X.

[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.

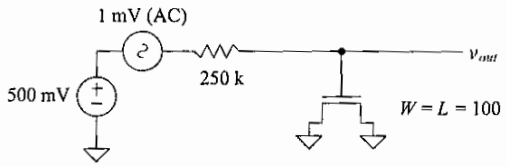


Figure 6.20 Circuit used in Problem 6.1.

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 100×10^9 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.

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 ($V_{DS1} < V_{GS1} - V_{THN}$).

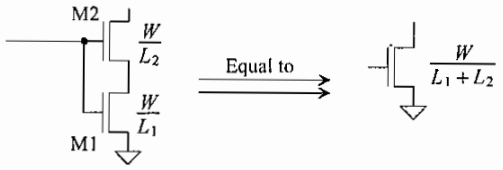


Figure 6.21 MOSFETs operating in series.

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.

This chapter was contributed by

Dr. Jeff Jessing, Boise State University

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 all 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.

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

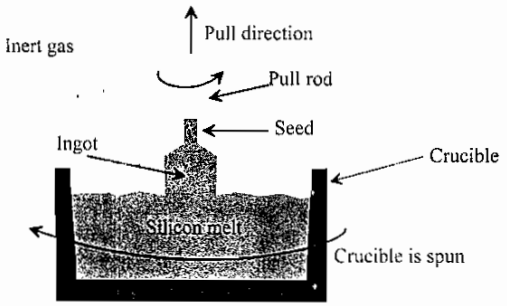


Figure 7.1 Simplified diagram showing Czochralski (CZ) crystal growth.

(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

$$Si + O_2 \rightarrow SiO_2 \text{ (dry oxidation)} \qquad (7.3)$$

$$Si + 2H_2O \rightarrow SiO_2 + 2H_2 \text{ (wet oxidation)} \qquad (7.4)$$

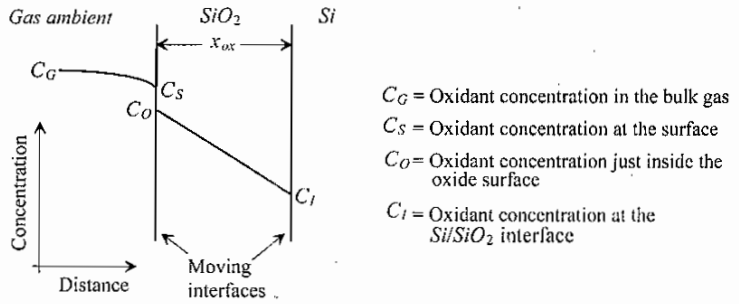


Figure 7.2 A simple model for thermal oxidation of silicon. Notice the oxidant concentrations (boundary conditions) in the gas, oxide, and silicon.

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.

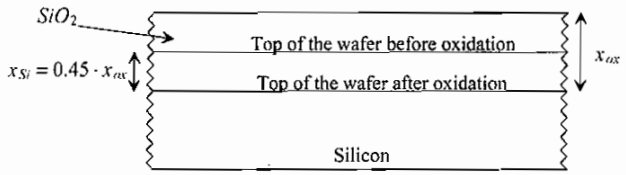


Figure 7.3 Silicon/oxide growth interface. See also Fig. 2.4.

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.

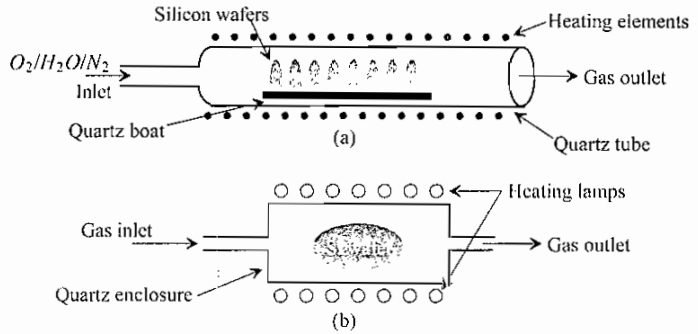


Figure 7.4 (a) Simplified representation of an oxidation tube furnace and (b) simplified diagram for rapid thermal processing.

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.

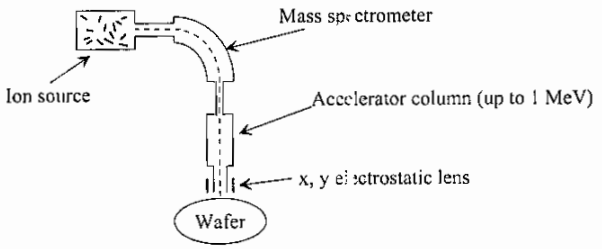


Figure 7.5 Simplified diagram of an ion implanter. The ions are created by an RF field where they are extracted into a mass spectrometer. An electrostatic lens scans the ion beam on the surface of a wafer to achieve the appropriate dose. Electrostatically, the ions can be counted to provide the real-time 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 ΔR_p is called the straggle. By inspection, R_p should be identified as the mean distance the ions travel into the silicon and ΔR_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

implant curve corresponds to what is referred to as the implant dose Q_{imp} , given mathematically as

$$Q_{imp} = \int_0^\infty N(x) \cdot dx \qquad (7.6)$$

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.

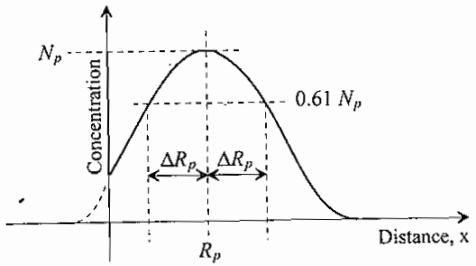


Figure 7.6 Ideal implant profile representing Eq. (7.5). Notice that the peak concentration occurs below the surface and depends on the implant energy.

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

$$N(x,t) = \frac{Q_{imp}}{\sqrt{\pi \cdot D \cdot t} \cdot \exp(-x^2/4Dt)} \qquad (7.7)$$

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.

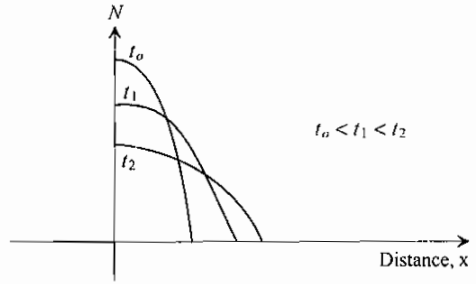


Figure 7.7 Idealized limited-source diffusion profile showing the effects of drive-in time on the profile. Notice that the peak concentration occurs at the surface of the substrate (x = 0) and that the area under the curves is constant.

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.

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.

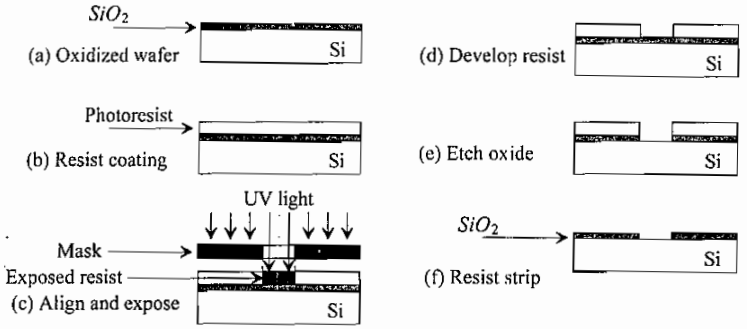


Figure 7.8 Simplified representation of the primary steps required for the implementation of photolighography and pattern transfer.

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, λ, 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 θ 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

$$\text{DOF} = \frac{c_2 \lambda}{NA^2} \tag{7.10}$$

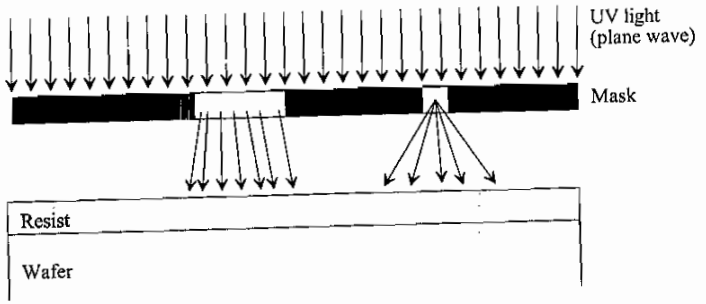


Figure 7.9 The diffraction effects become significant as the mask feature dimensions approach the wavelength of UV light. Notice that the diffraction angle is larger for the smaller opening.

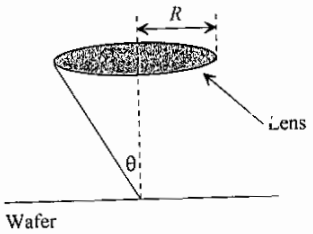


Figure 7.10 The relationship of the lens radii to the angle used to compute NA.

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.

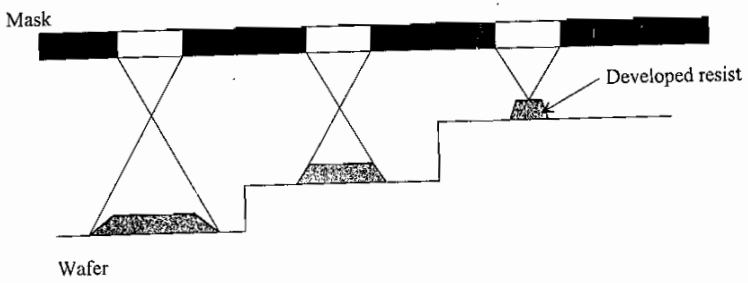
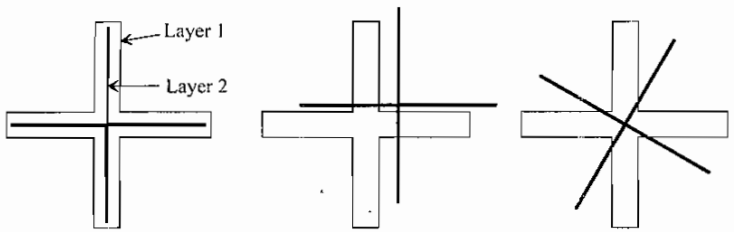


Figure 7.11 Depth of focus diagram illustrating the need to have planar surfaces (minimized topography) during high resolution photopatterning.

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

$$A_f = 1 - \frac{R_l}{R_v} \tag{7.12}$$

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.

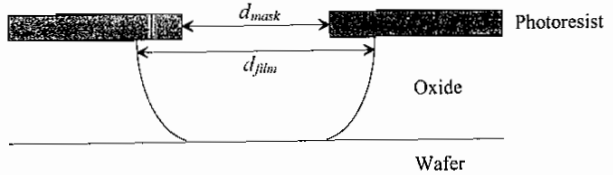


Figure 7.13 Diagram showing a post-etch profile. Notice that because of isotropy in the etch process the mask opening does not match the fabricated opening in the underlying oxide film. The difference between these dimensions is called etch-bias.

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

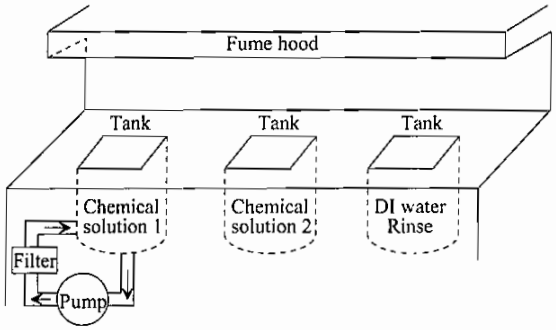


Figure 7.14 Simplfied diagram of a wet bench used for wet chemical cleaning and etching.

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.

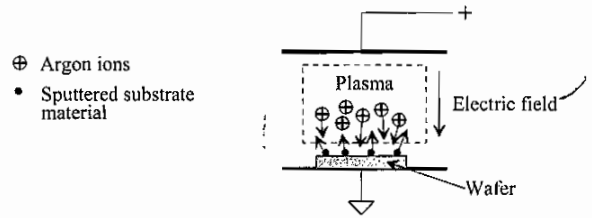


Figure 7.15 Simplified schematic diagram of the sputter etch process. This process is dominated by the physical bombardment of ions on a substrate.

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.

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.

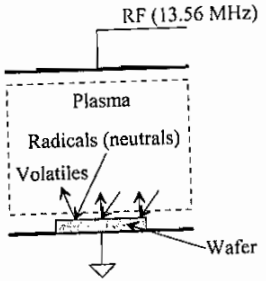


Figure 7.16 Simplified schematic diagram of a plasma etch process. This process is dominated by the chemical reactions of radicals at the surface of the substrate.

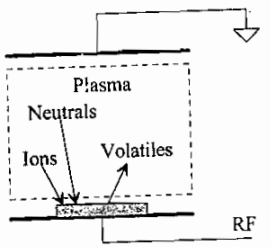


Figure 7.17 Simplified schematic diagram of an RIE etch process. This process has both physical (ion bombardment) and chemical (reaction of radicals) components.

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

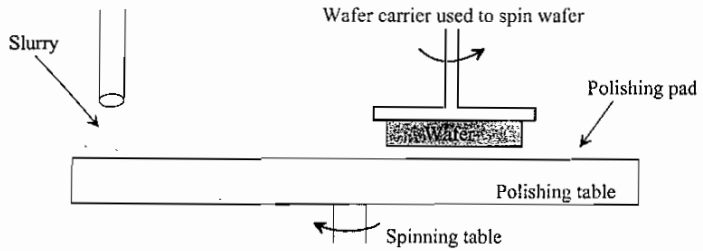


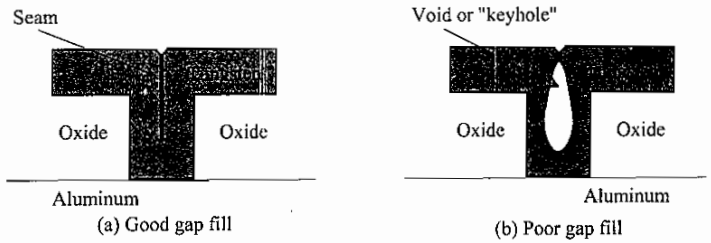
Figure 7.18 Simplified representation of a chemical mechanical polishing process used in the fabrication process.

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.



(a) Good gap fill (b) Poor gap fill

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.

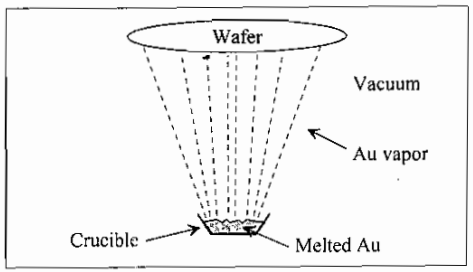


Figure 7.21 Simplified diagram of an evaporation deposition process.

In simple terms, sputter deposition is similar to sputter etching, as discussed in Sec. 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

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.

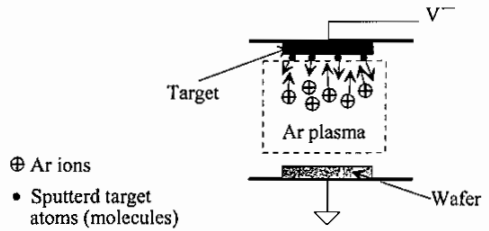


Figure 7.22 Simplified diagram of a sputter deposition process.

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

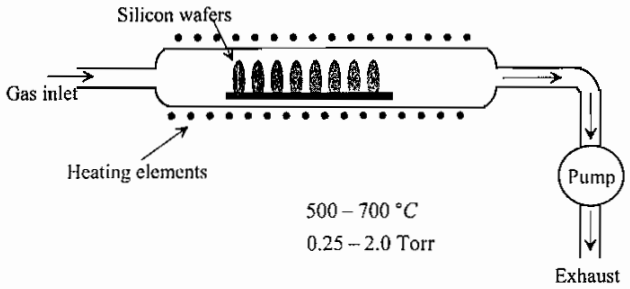


Figure 7.23 Simplified schematic diagram of a LPCVD.

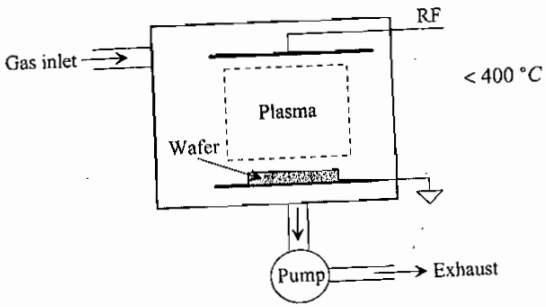


Figure 7.24 Simplified schematic diagram of a PECVD reactor.

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

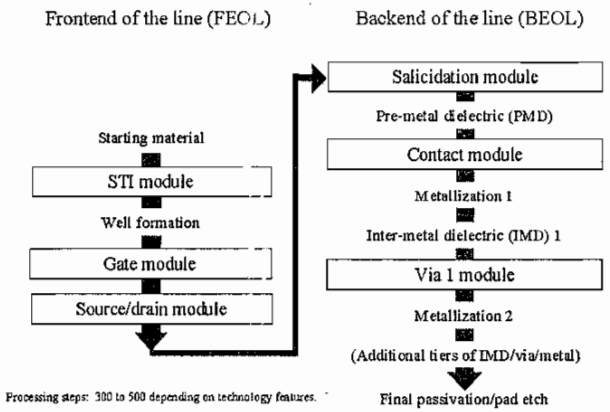


Figure 7.25 A typical CMOS process flow illustrating the difference between FEOL and BEOL processes.

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

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

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.

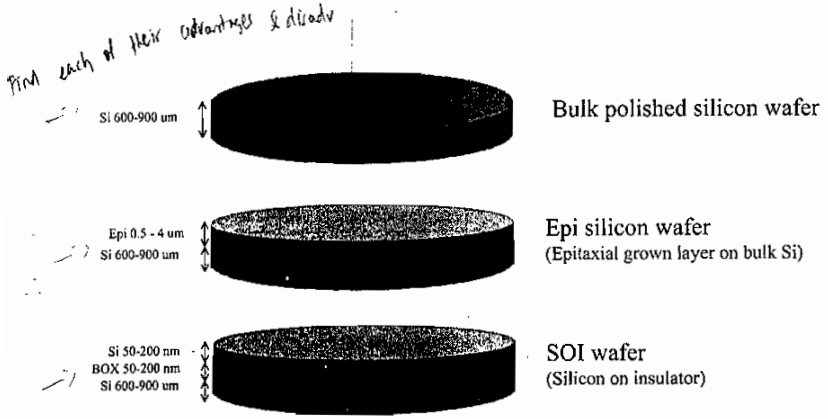


Figure 7.26 The three general types of silicon wafers used for CMOS fabrication.

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.

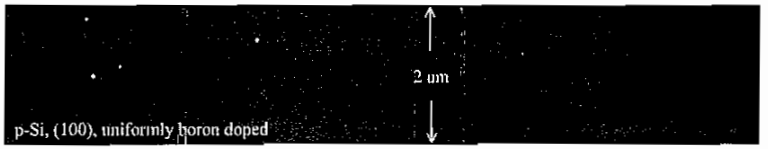


Figure 7.27 Simulated cross-sectional view (the top 2 μm) of the bulk wafer in Fig. 7.26.

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

can be implemented with minimal active-area encroachment. Moreover, it has a relatively low thermal budget.

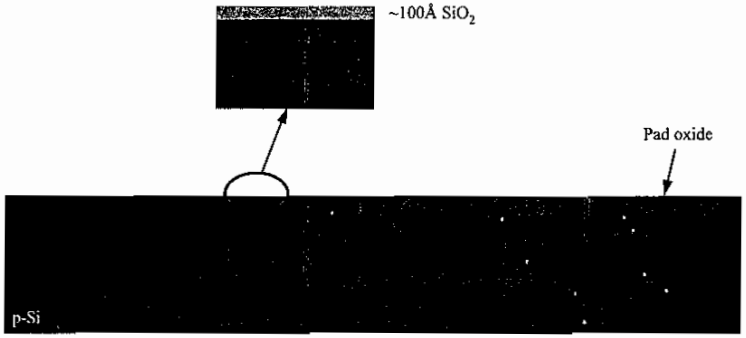


Figure 7.28 STI film stack. The oxide is thermally grown at approximately 900 °C with dry O_2.

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

Figure 7.29 STI film stack. Silicon nitride deposited by LPCVD at approximately 800 °C.

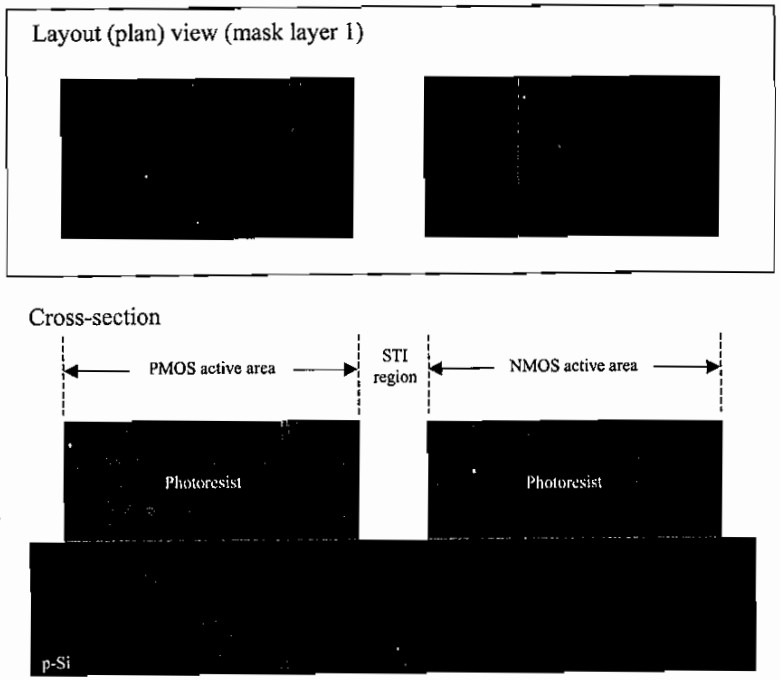


Figure 7.30 STI definition, photolithography and nitride/pad oxide etch with fluorocarbon- based RIE.

timed RIE with the photoresist 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

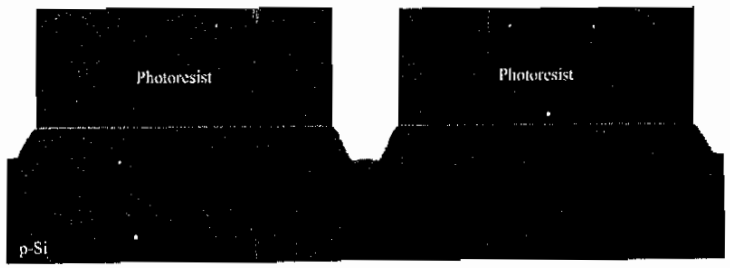


Figure 7.31 Timed silicon trench reactive ion etch.

parasitic corner transistors. 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.

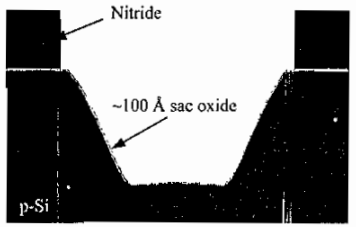


Figure 7.32 Cross-section showing post STI resist strip followed by the dry thermal oxidation (at 900 °C) of a sacrificial (sac) oxide in the trench.

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

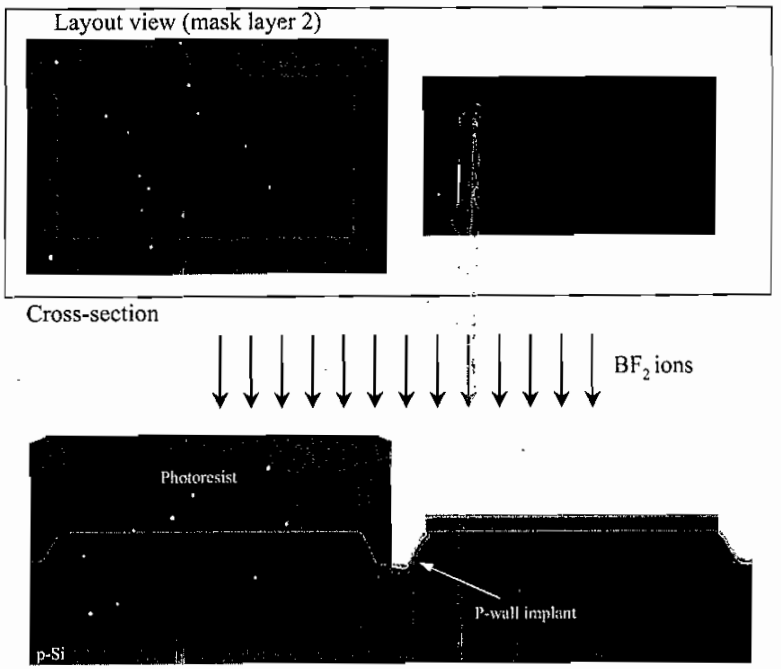


Figure 7.33 P-wall sidewall formation via photolithography and BF_2 implantation.

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

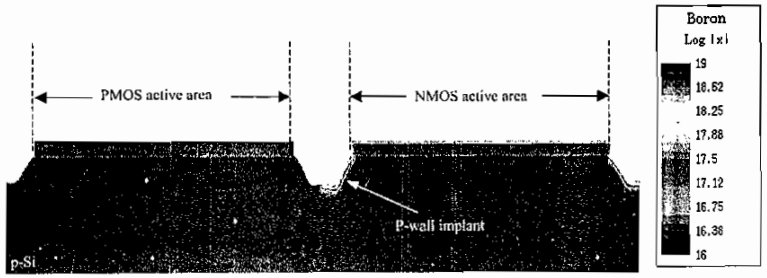


Figure 7.34 Post p-wall photoresist strip using O_2 plasma and wet processing.

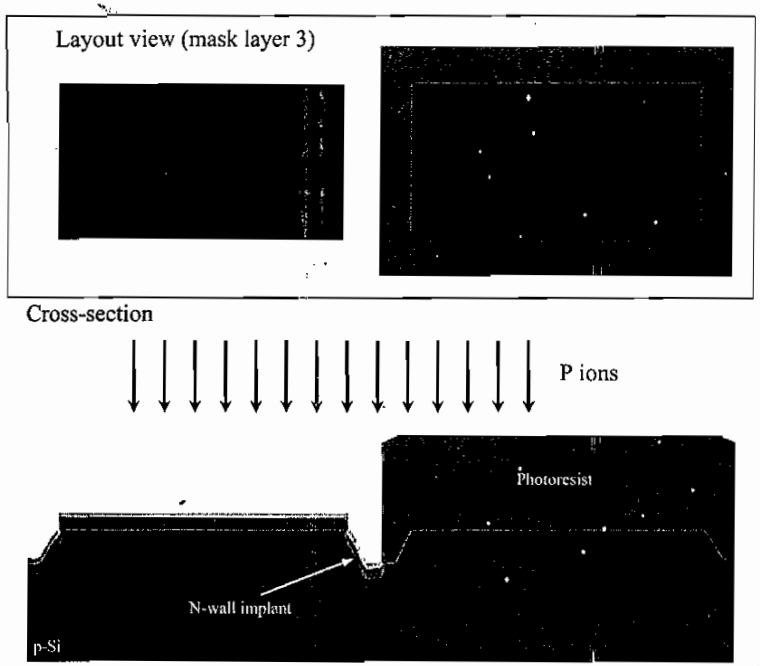


Figure 7.35 N-wall sidewall formation via photolithography and *P* implantation.

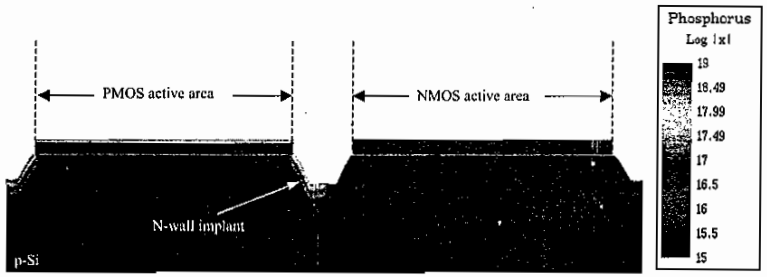


Figure 7.36 Post n-wall photoresist strip using O_2 plasma and wet processing.

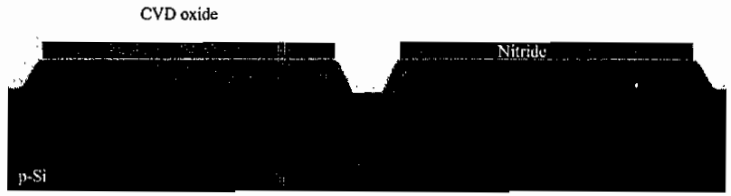


Figure 7.37 High quality, 100 Å thick liner oxide is thermally grown at 900 °C. High density plasma (HDP) CVD trench fill at room temperature. Notice that the trenches are overfilled.

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.

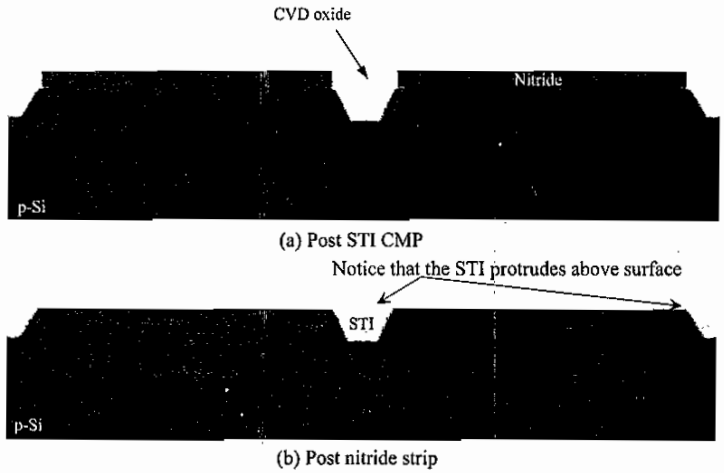


Figure 7.38 STI CMP (a) where the nitride acts like a polish stop and (b) wet nitride etch in hot phosphoric acid and/or dry nitride etch in NF_3/Ar/NO. The remaining oxide will be used as a sacrificial oxide for subsequent implants.

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

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.

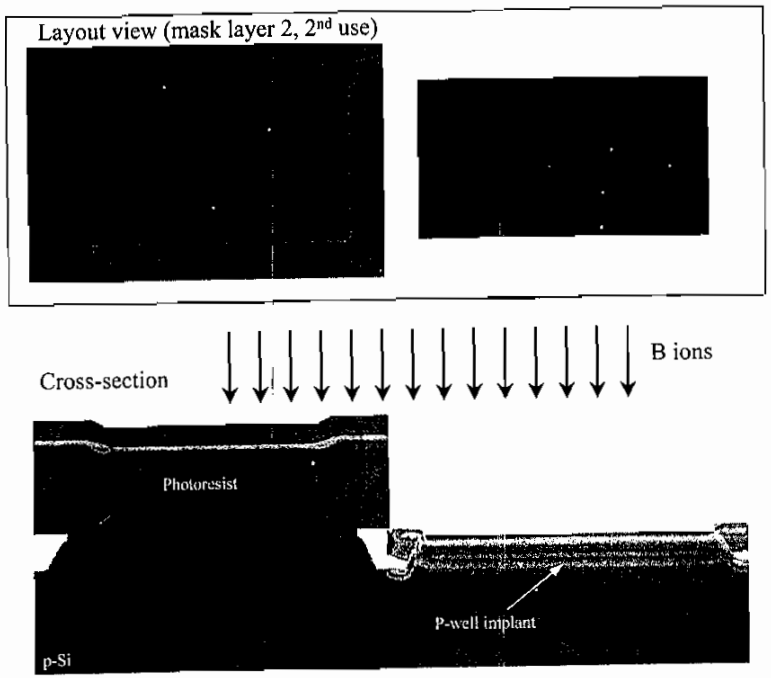


Figure 7.39 P-well formation via photolithography and *B* implantation.

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

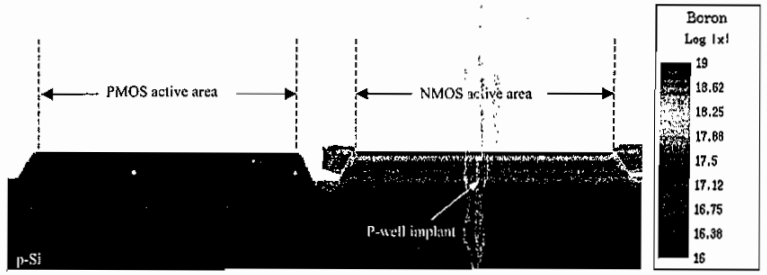


Figure 7.40 Post p-well photoresist strip using O_2 plasma and wet processing.

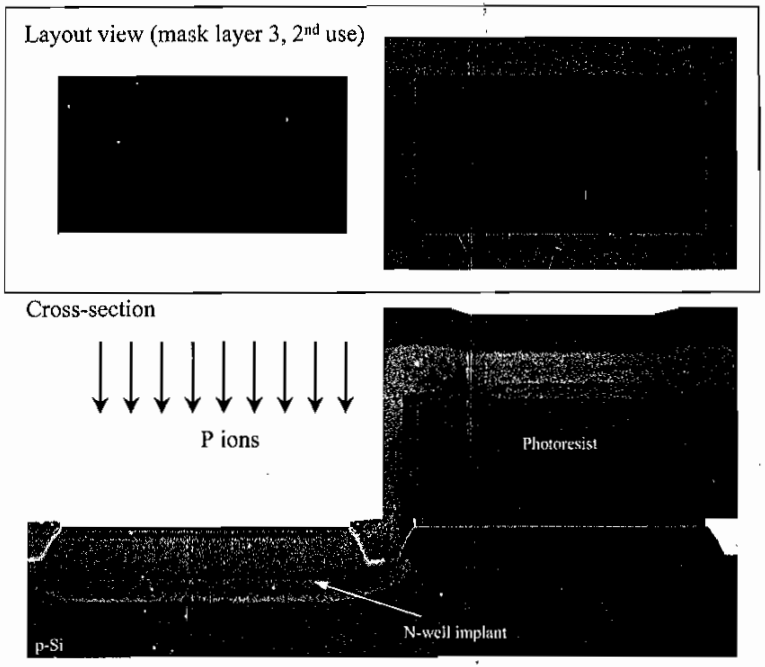


Figure 7.41 N-well formation via photolithography and *P* implantation.

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.

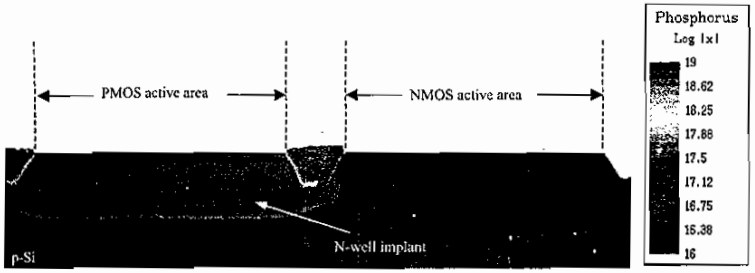


Figure 7.42 Post n-well photoresist strip using O_2 plasma and wet processing.

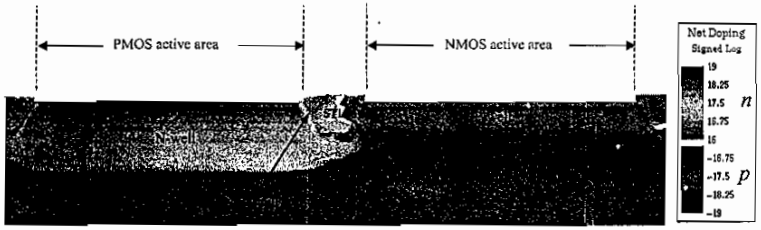


Figure 7.43 Net doping profile of both the n-well and the p-well.

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

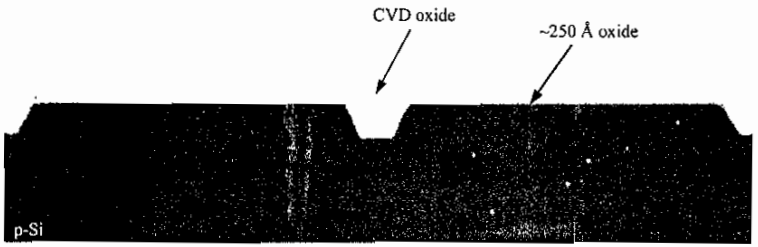


Figure 7.44 Wet etch the remaining trench stack oxide using buffered *HF*. Sacrificial oxide formation using dry thermal oxidation at approximately 900 °C.

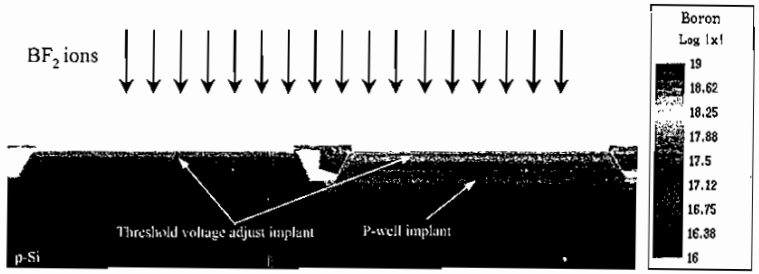


Figure 7.45 Blanket low-energy BF_2 implant for NMOS and PMOS threshold voltage adjust.

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

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.

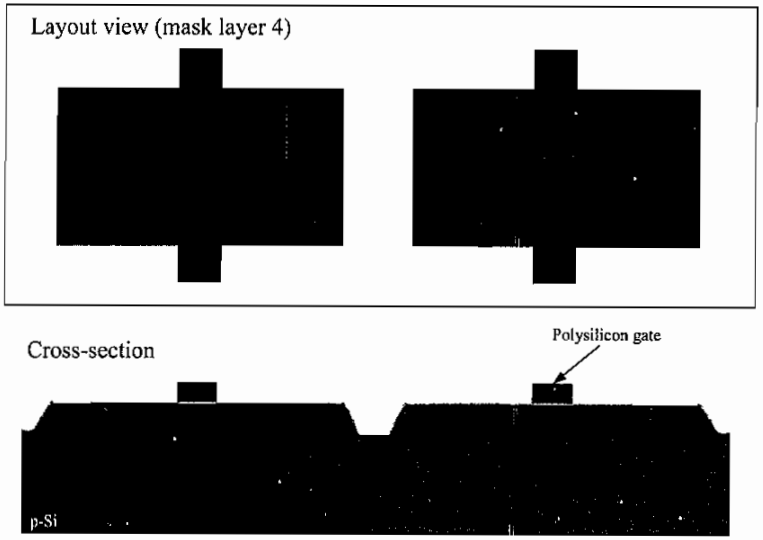


Figure 7.48 Gate electrode and local interconnect photolithography and polysilicon reactive ion etching.

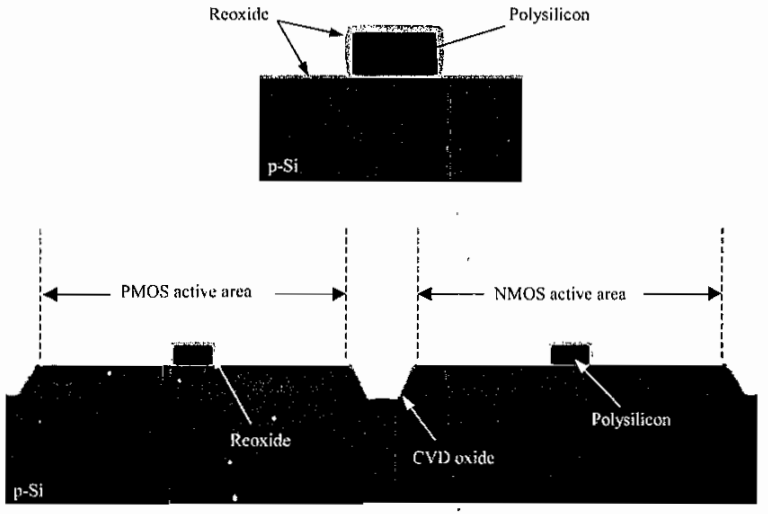


Figure 7.49 Polysilcion reoxidation using dry O_2 at approximately 900 °C. Notice that the resulting oxide is thicker on the polysilicon than on the active silicon.

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.

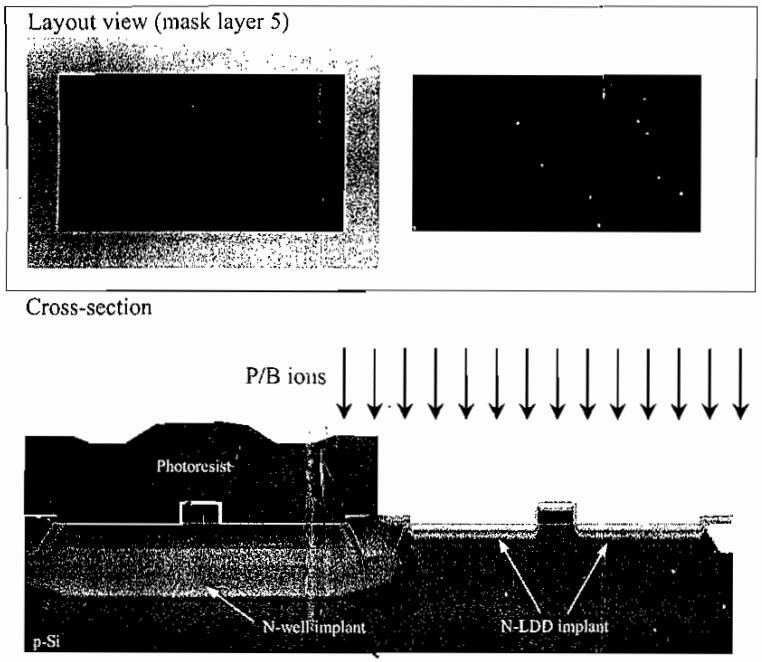


Figure 7.50 N-LDD/n-pocket formation using low energy implantation of *P* and *B*, respectively.

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

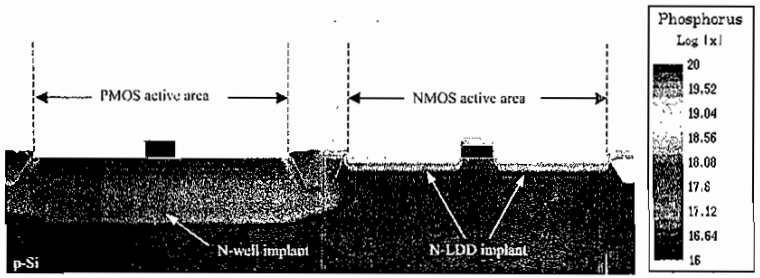


Figure 7.51 Post-n-LDD resist strip using O_2 plasma and wet processing.

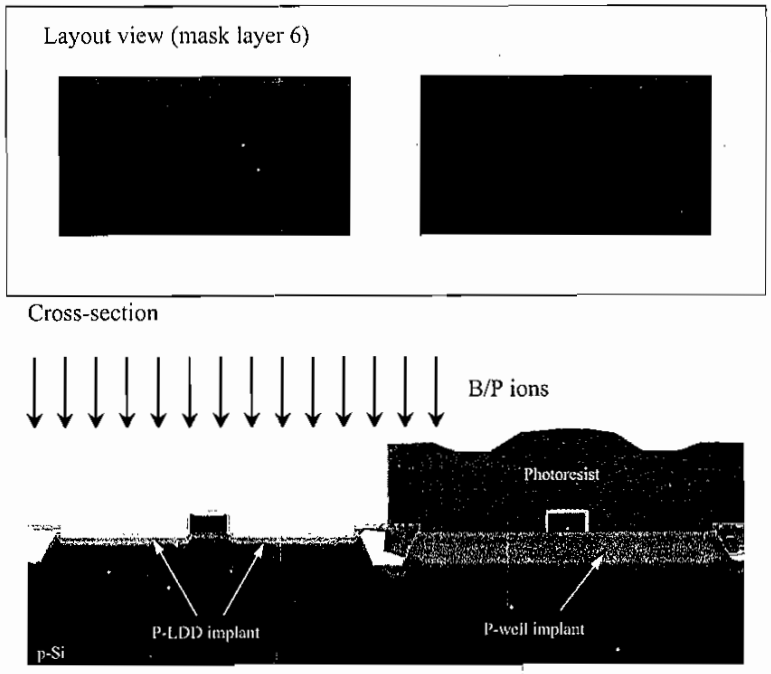


Figure 7.52 P-LDD/p-pocket formation using low energy implantation of *B* and *P*, respectively.

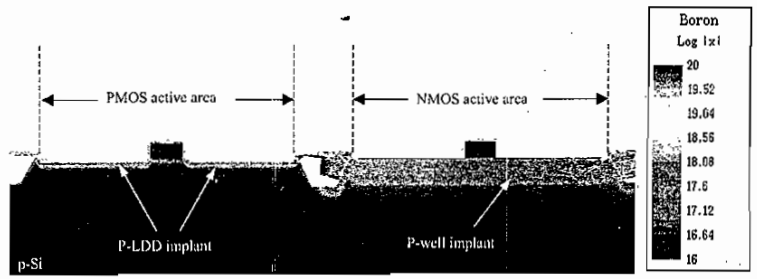


Figure 7.53 Post-p-LDD resist strip using O_2 plasma and wet processing.

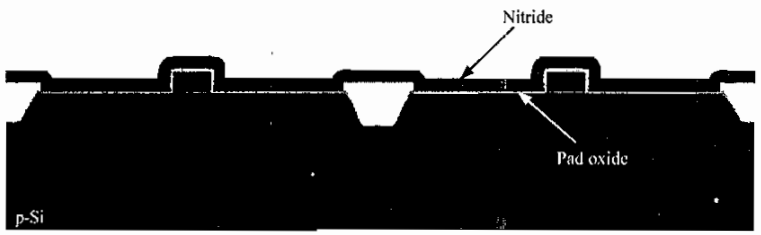


Figure 7.54 Sidewall spacer nitride deposition using LPCVD at approximately 800 °C.

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.

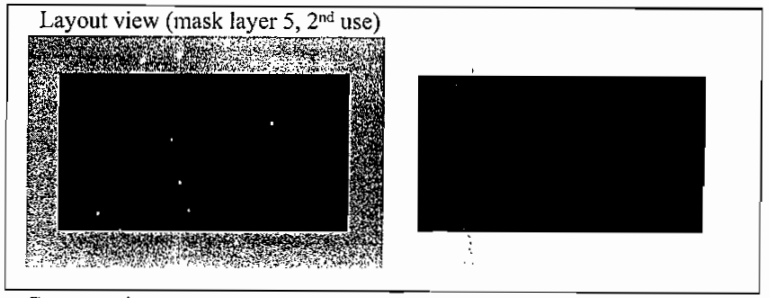


Figure 7.56 N+ source/drain formation using a low energy, high dose implantation of *As*.

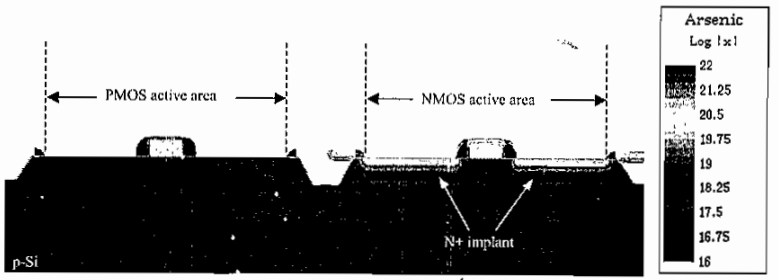


Figure 7.57 Post-n+ resist strip using O_2 plasma and wet processing.

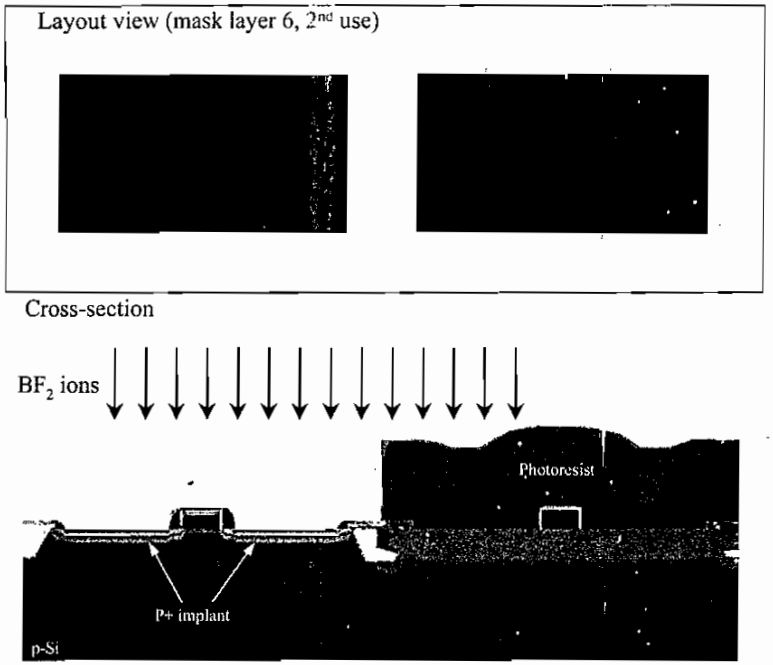


Figure 7.58 P+ source/drain formation using a low energy, high dose implantation of BF_2.

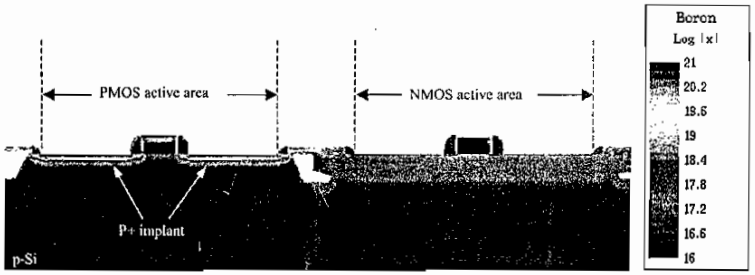


Figure 7.59 Post p+ resist strip using O_2 plasma and wet processing.

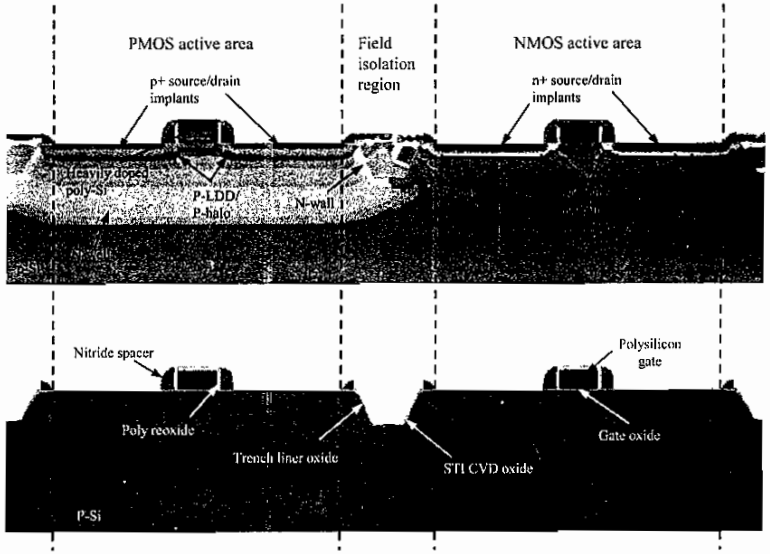


Figure 7.60 Summary of the FEOL features.

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

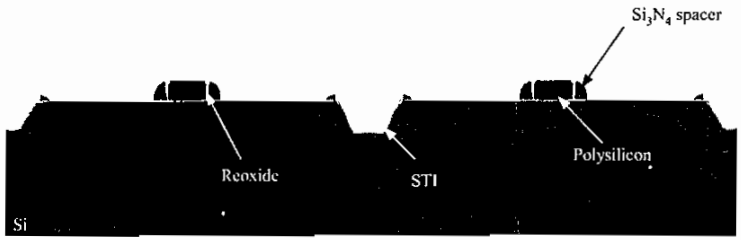


Figure 7.61 Removal of the exposed reoxide (present from the FEOL) using buffered-*HF*.

a high resistivity 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.

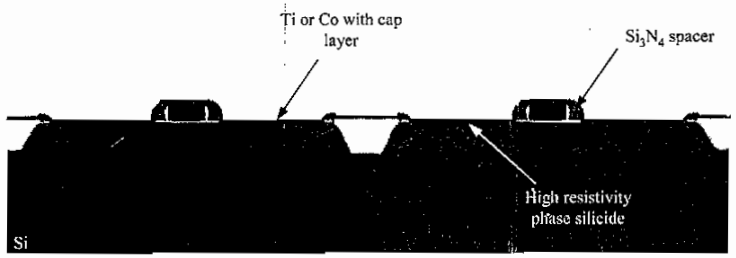


Figure 7.62 Titanium or cobalt deposited by PVD followed by the first salicide rapid thermal anneal.

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

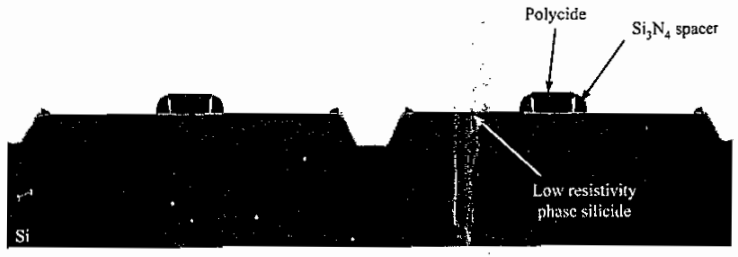


Figure 7.63 Wet chemical etch of the unreacted titanium or cobalt followed by the second salicide rapid thermal anneal.

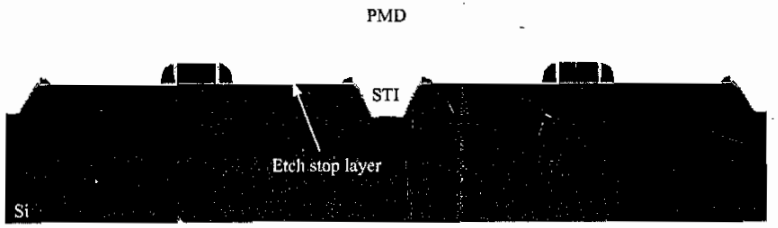


Figure 7.64 Pre-metal dielectric (PMD) deposition using high density plasma

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.

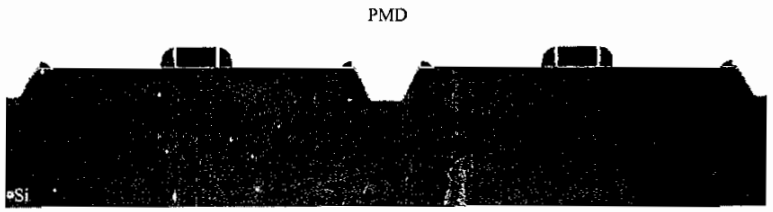


Figure 7.65 Planarizing of the PMD using CMP.

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.

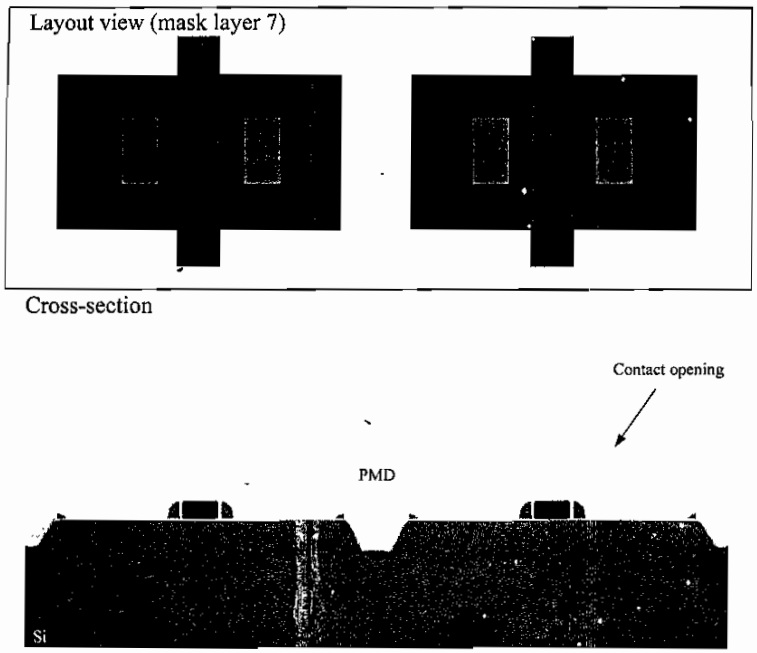


Figure 7.66 Contact definition using photolithography and RIE of the PMD and stop layer. Notice that the contacts to the poly are formed at the same time but not shown.

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 photolithographic alignment of the metal layer to the contacts, a tungsten recess etch is

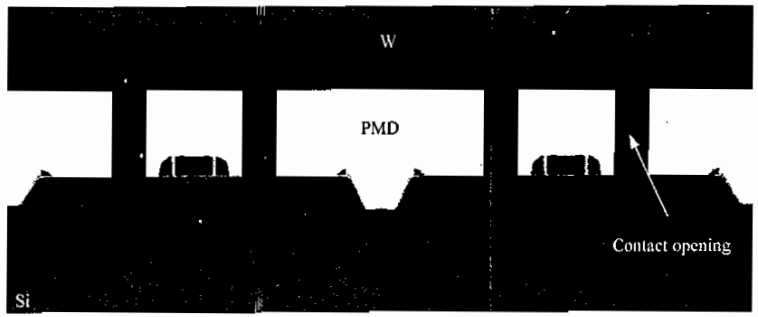


Figure 7.67 *Ti/TiN* liner deposition using IMP and CVD, respectively. *W* contact fill deposition using WF_6 CVD.

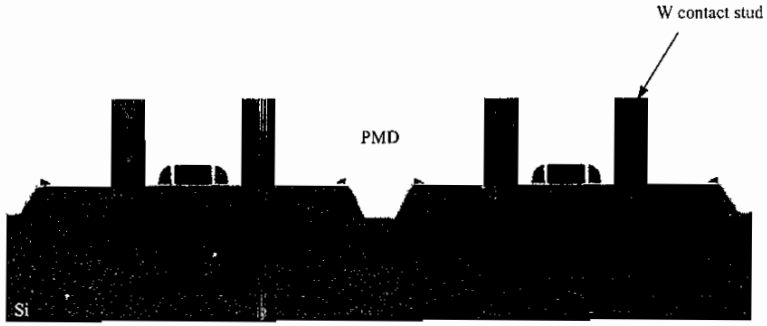


Figure 7.68 *W* CMP to form defined contacts.

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.

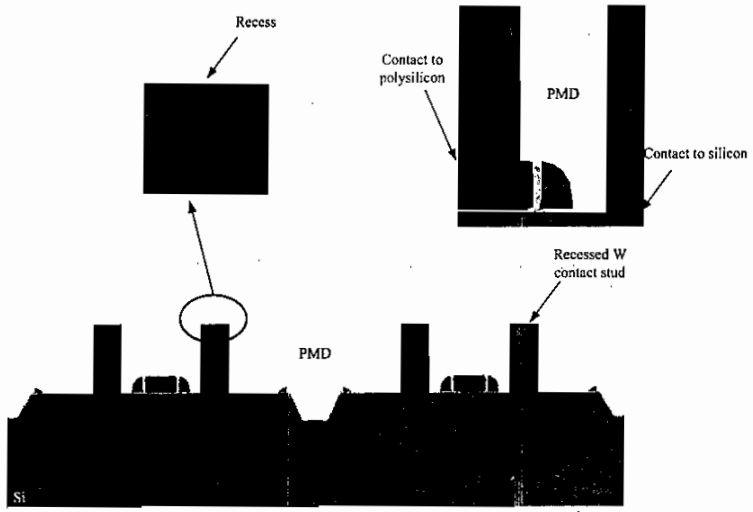


Figure 7.69 *W* recess etch using "buff" polish or dry *W* etch.

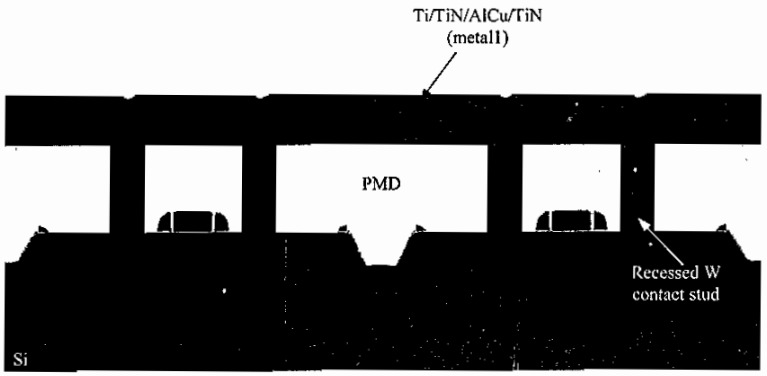


Figure 7.70 Metal1 stack deposition using PVD.

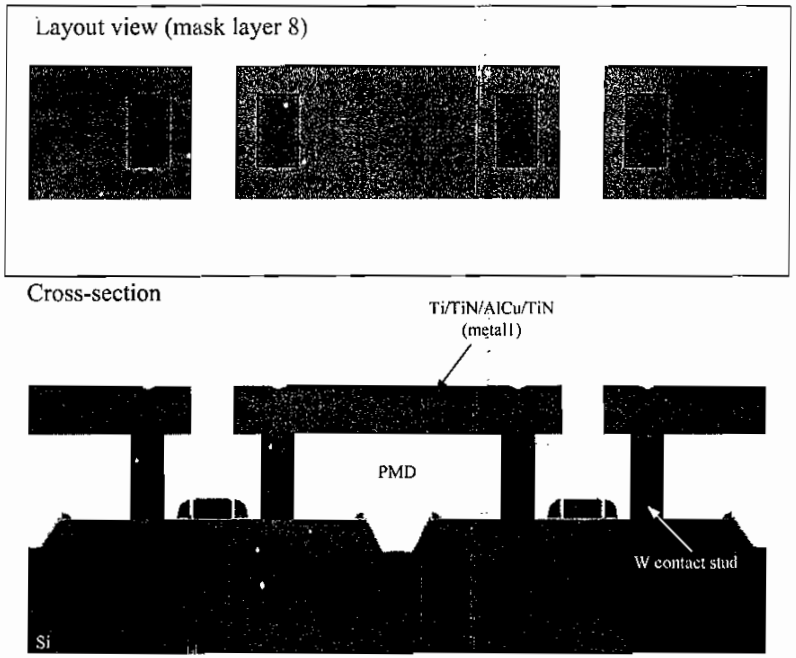


Figure 7.71 Metal1 definition using photolithography and dry metal etch.

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).

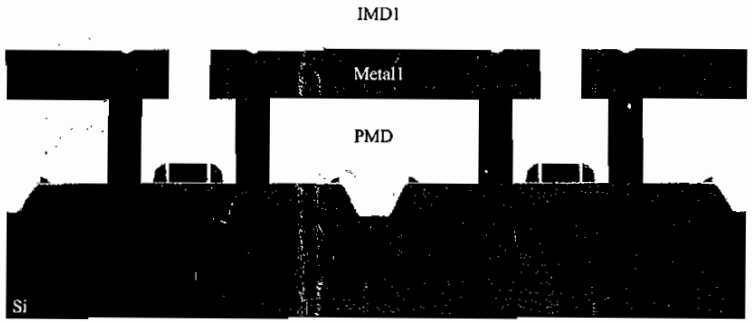


Figure 7.72 Intra-metal dielectric 1 (IMD1) deposition using HDP CVD. This is followed by IMD1 planarization using CMP.

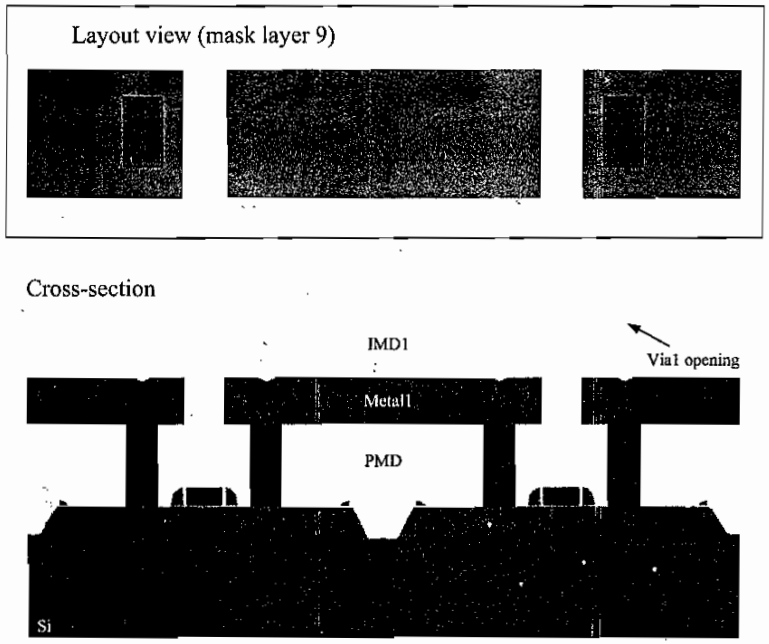


Figure 7.73 Via1 definition using photolithography and dry IMD1 etch.

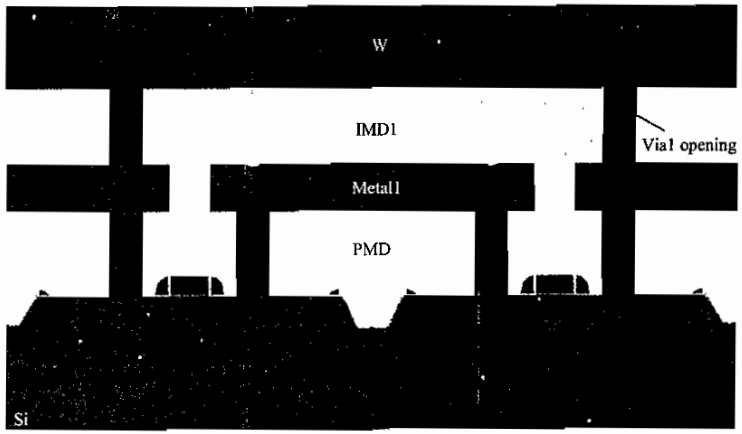


Figure 7.74 *Ti/TiN* liner deposition using IMP and CVD, respectively. *W* via1 fill deposition using WF_6 CVD.

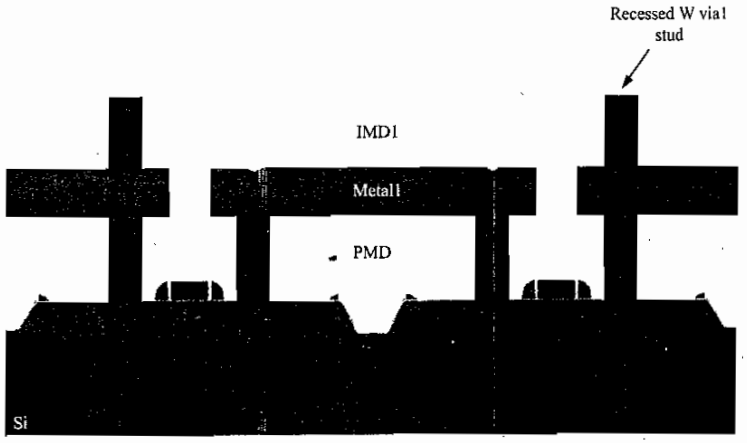


Figure 7.75 *W* CMP to form defined vias. This is followed by *W* recess etch using "buff" polish or dry *W* etch.

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.

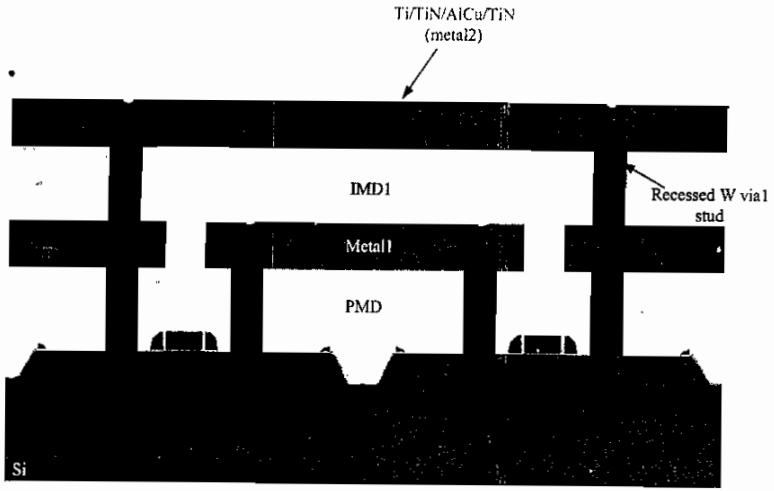


Figure 7.76 Metal2 stack deposition using PVD.

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

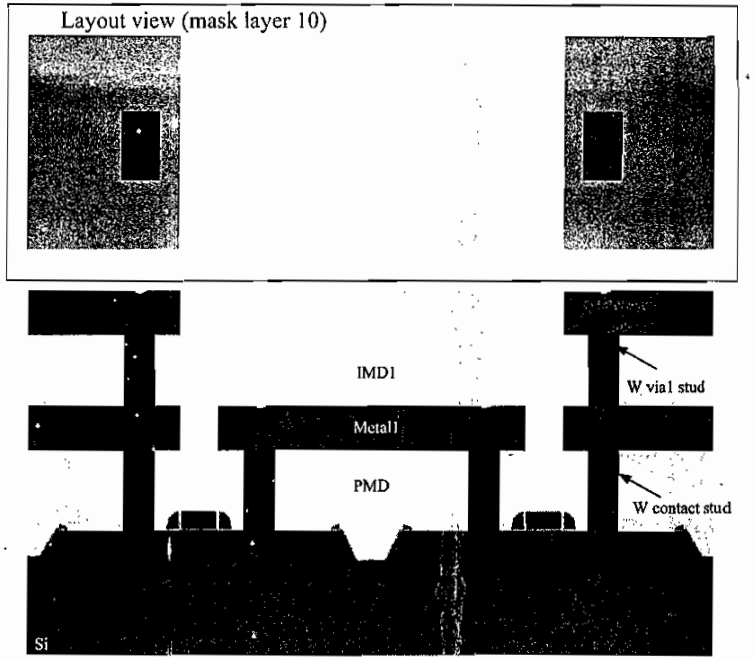


Figure 7.77 Metal2 definition using photolithography and dry metal etch.

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

$$Y = \frac{\#\text{ of die passing all tests}}{\text{total }\#\text{ of die}} \tag{7.13}$$

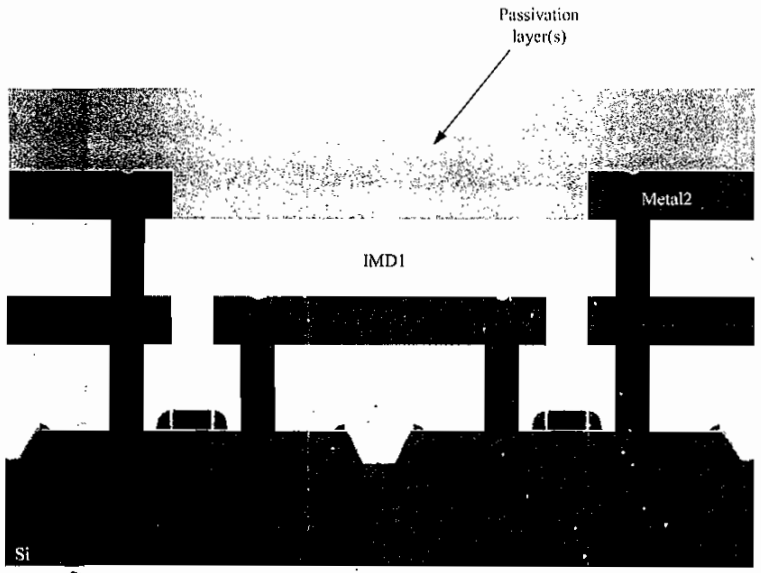


Figure 7.78 Deposition of final passivation.

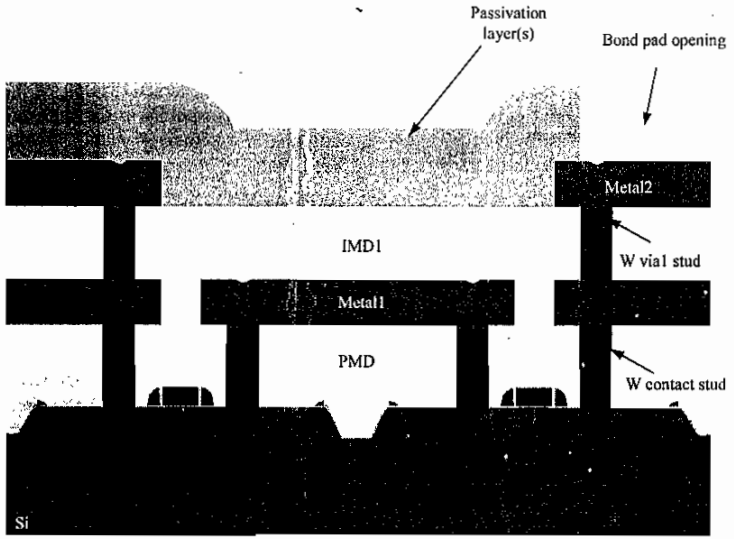


Figure 7.79 Bond pad definition using photolithography and dry etch of passivation.

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.

ADDITIONAL READING

[1] S. A. Campbell, *The Science and Engineering of Microelectronic Fabrication*, 2nd ed, Oxford University Press, 2001. ISBN 0-19-513605-5

[2] 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

[3] 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

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 sine wave. 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

$$P_{inst}(t) = \frac{v^2(t)}{R} = \frac{V_P^2 \sin^2(2\pi \cdot \frac{t}{T})}{R} \quad (8.1)$$

Summing this power over time results in

$$Energy = \int_0^{\infty} \frac{V_P^2 \sin^2(2\pi \cdot \frac{t}{T})}{R} \cdot dt \quad (8.2)$$

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 sine wave 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.)

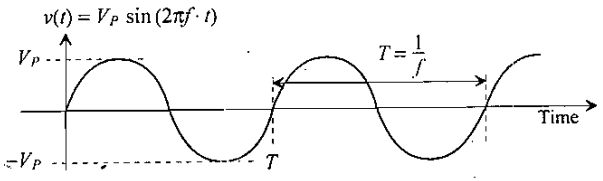


Figure 8.1 A sine wave.

Examining Fig. 8.1 for a moment, we realize that because the sine wave 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 sine wave source during this cycle) and then dividing by the cycle time T. This can be written as

$$\text{Average power dissipated} = \frac{1}{T}\int_0^T \frac{V_P^2 \sin^2(2\pi \cdot \frac{t}{T})}{R} \cdot dt \quad (8.3)$$

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

$$\text{Mean-squared voltage} = \overline{v^2} = \frac{1}{T}\int_0^T V_P^2 \sin^2(2\pi \cdot \frac{t}{T}) \cdot dt = \frac{V_P^2}{2} \quad (8.4)$$

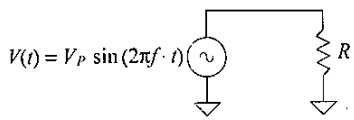


Figure 8.2 A sine wave driving a resistor.

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

$$\text{RMS voltage of a sine wave, } \sqrt{\overline{v^2}} = V_{RMS} = \sqrt{\frac{1}{T}\int_0^T V_P^2 \sin^2(2\pi f \cdot t) \cdot dt} = \frac{V_P}{\sqrt{2}} \quad (8.5)$$

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 sine wave (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 sine wave 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.

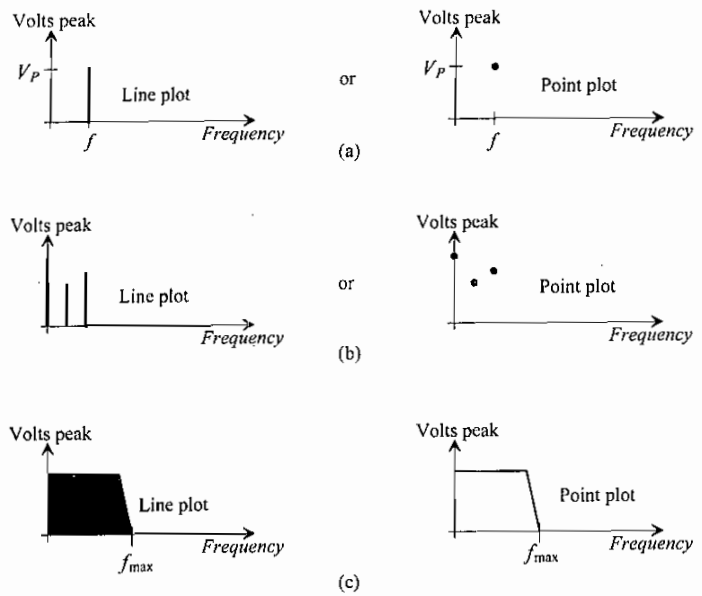


Figure 8.3 Spectrums of various signals (see text).

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.

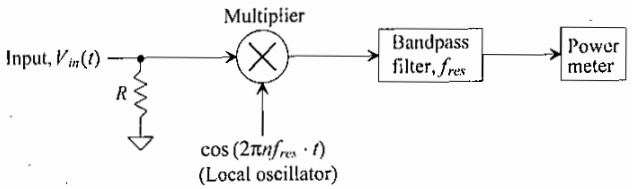


Figure 8.4 Block diagram of a spectrum analyzer.

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 sine wave 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 sine wave 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).

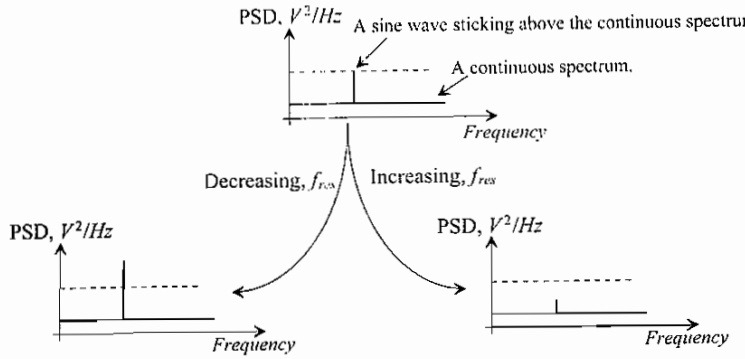


Figure 8.5 Changing the resolution bandwidth doesn't affect the PSD amplitude of the continuous spectrum but does affect the amplitude of the sine wave sticking up above the continuous spectrum.

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 sine wave 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 second volume of this book entitled *CMOS Mixed-Signal Circuit Design*. In this book additional information and discussions are given.

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.

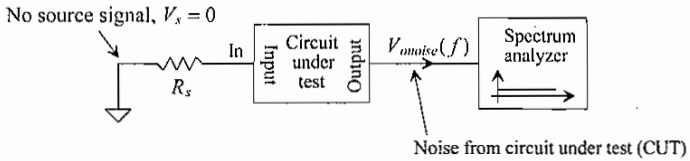


Figure 8.6 Circuit used for noise measurement.

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

$$V_{RMS} = \sqrt{\int_{f_L}^{f_H} V^2_{noise}(f) \cdot df} \quad \text{Volts} \qquad (8.10)$$

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

$$V_{RMS} = \sqrt{(f_H - f_L) \cdot V^2_{noise}(f)} = \sqrt{B} \cdot \sqrt{V^2_{noise}(f)} \qquad (8.11)$$

where the bandwidth of the measurement, B, is $f_H - f_L$.

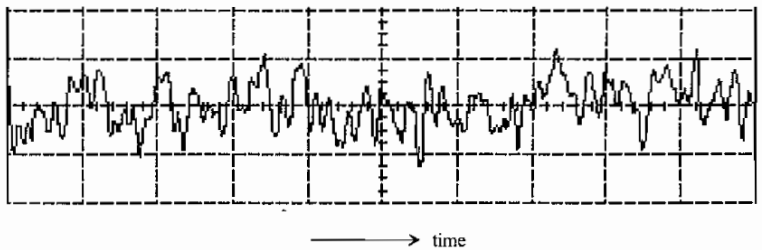


Figure 8.7 Time domain view of circuit noise.

Input-Referred Noise I

Noise is always 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 (and so it can't be measured directly). This is especially true for CMOS circuits where the input to a CUT may be the polysilicon gate of a MOSFET (which is, of course, surrounded by an insulating dielectric).

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).

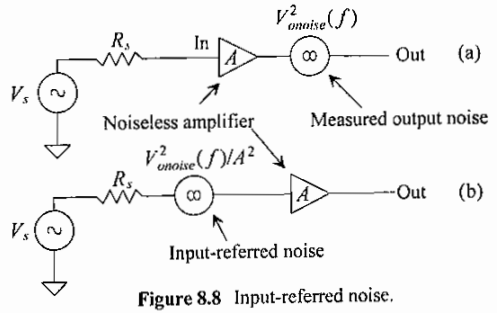


Figure 8.8 Input-referred noise.

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 ($B = f_H - f_L = 100\ MHz$) 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

$$V_{onoise,RMS} = \sqrt{\int_0^{100MHz} 100 \times 10^{-18} df} = 100\ \mu V$$

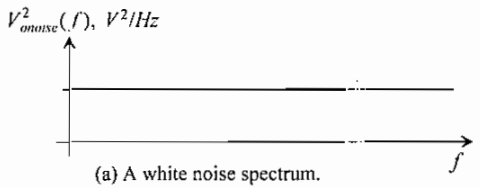
Referring this noise back to the input of the amplifier results in

$$V_{inoise,RMS} = \frac{V_{onoise,RMS}}{A} = 1\ \mu V \qquad (8.12)$$

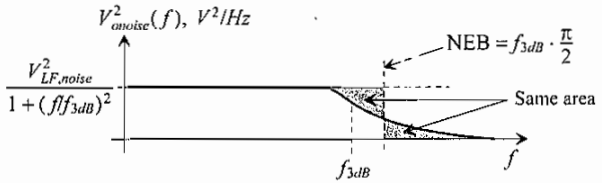
■

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 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

$$V^2_{onoise,RMS} = \int_0^{\infty} \frac{\overbrace{V^2_{onoise}(f)}}{\left(\sqrt{1+(f/f_{3dB})^2}\right)^2} df \qquad (8.13)$$

If we use

$$\int \frac{du}{a^2+u^2} = \frac{1}{a}\tan^{-1}\frac{u}{a} + C \qquad (8.14)$$

then

$$V^2_{onoise,RMS} = V^2_{LF,noise} \cdot f_{3dB} \cdot [\tan^{-1} 2\pi f/f_{3dB}]_0^{\infty}$$

$$= V^2_{LF,noise} \cdot \overbrace{f_{3dB} \cdot \frac{\pi}{2}} \qquad (8.15)$$

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. ■

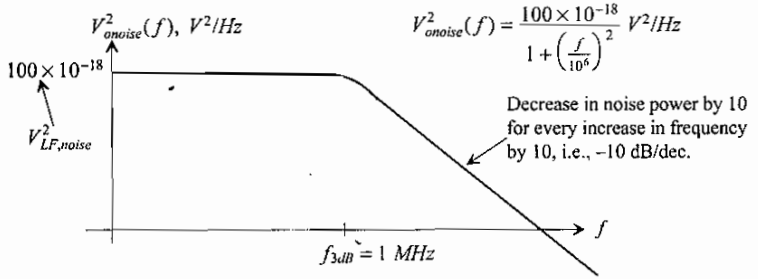


Figure 8.10 Measured output noise spectrum for the CUT discussed in Ex. 8.2.

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!).

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

$$V_{inoise,RMS} = \frac{\sqrt{NEB \cdot V^2_{LF,noise}}}{A_{DC}} = \frac{V_{onoise,RMS}}{A_{DC}} \tag{8.20}$$

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

$$V^2_{onoise,RMS} = (A_1A_2A_3)^2 V^2_{inoise,RMS1} + (A_2A_3)^2 V^2_{inoise,RMS2} + A^2_3 V^2_{inoise,RMS3} \tag{8.21}$$

again remembering that we add the noise powers from each amplifier. The input-referred noise in (b) is given by

$$V^2_{inoise,RMS} = V^2_{inoise,RMS1} + V^2_{inoise,RMS2}/A^2_1 + V^2_{inoise,RMS3}/(A_1A_2)^2 \tag{8.22}$$

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.

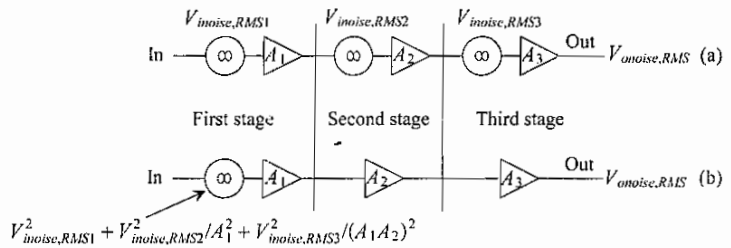


Figure 8.11 Noise performance of a cascade of amplifiers.

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

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 sine wave 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 sine wave amplitude to the input sine wave 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.

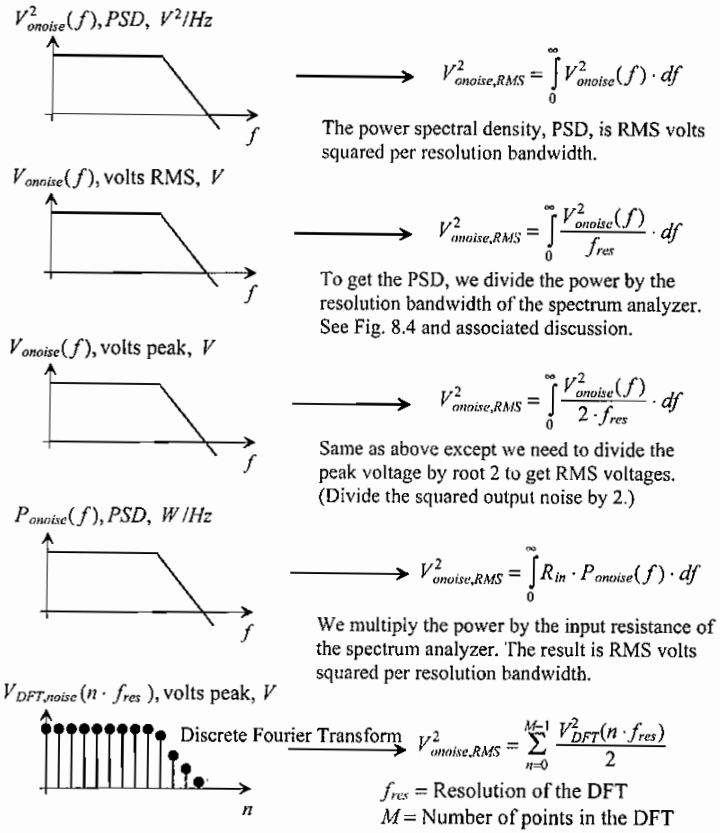


Figure 8.12 Calculating RMS output noise from a noise spectrum.

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

$$V_R^2(f) = 4kTR \text{ with units of } V^2/Hz \qquad (8.23)$$

(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, units = joules = watt·sec = volts·amps·sec = 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,

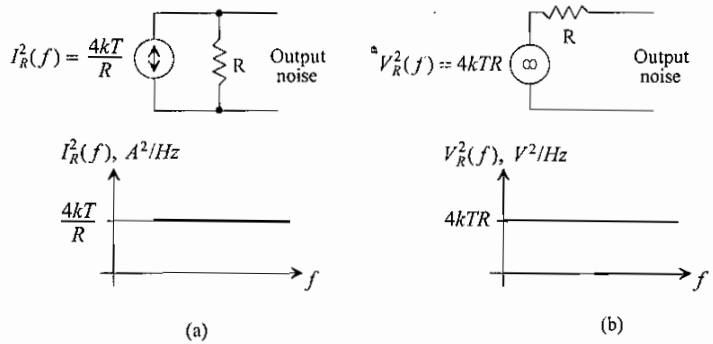


Figure 8.13 Circuit representations and their corresponding PSDs for thermal noise.

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 4.14×10^{-21} joules. For a one-second time frame, again at 300 °K, the resistor dissipates 4.14×10^{-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.

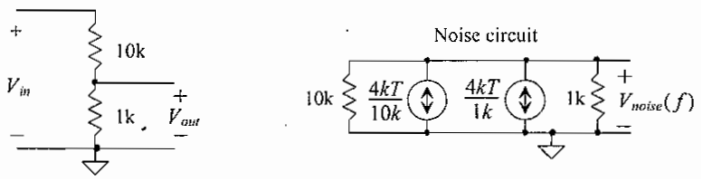


Figure 8.14 Circuit used in Ex. 8.5.

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).

The mean-squared output noise voltage over a bandwidth of 1 kHz is

$$V_{onoise,RMS}^2 = \int_{f_L}^{f_H} V_{onoise}^2(f) \cdot df = \int_0^{1\text{kHz}} \left(V_{10k}^2(f) + V_{1k}^2(f)\right) \cdot df = 15.1 \times 10^{-15}\ V^2$$

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 1.8215×10^{-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. ■

Example 8.6

Estimate $V_{onoise,RMS}$ for the circuit seen in Fig. 8.15. Verify the answer with

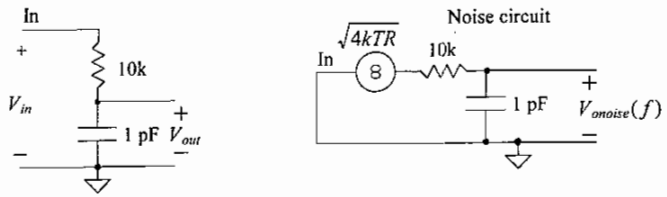


Figure 8.15 Circuit used in Ex. 8.6.

SPICE.

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%
```



```
inoise_total = 1.695223e-07
onoise_total = 4.101864e-09
```

Using Eq. (8.24) at 300 °K, we get $V^2_{onoise,RMS} = 4.14\times10^{-9}\ V^2$. 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 $V_{inoise,RMS} = V_{onoise,RMS}$. ■

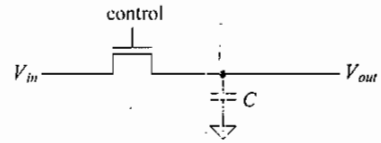


Figure 8.16 A sample-and-hold circuit.

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

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

$$SNR = 10 \log \frac{P_s}{P_{noise}} = 20 \log \frac{\sqrt{P_s}}{\sqrt{P_{noise}}} = 20 \log \frac{V_{s,RMS}}{V_{noise,RMS}} \tag{8.27}$$

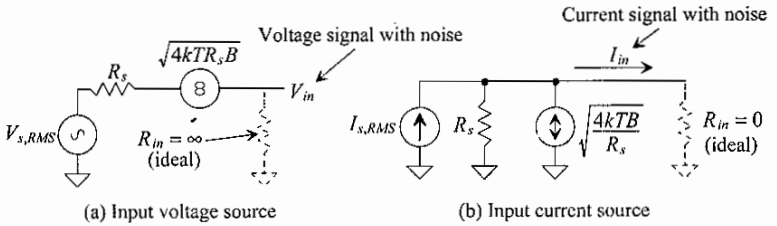


Figure 8.17 Calculating input SNR.

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

$$SNR_{out} = \frac{V_{s,RMS}^2 \cdot \left[\frac{R_{in}}{R_{in}+R_s}\right]^2 \cdot A^2}{V_{onoise,RMS}^2} \tag{8.30}$$

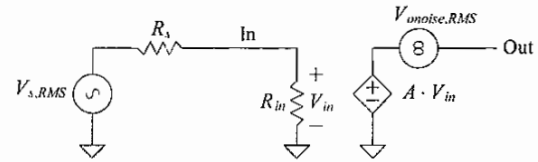


Figure 8.18 Calculating output SNR.

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.)

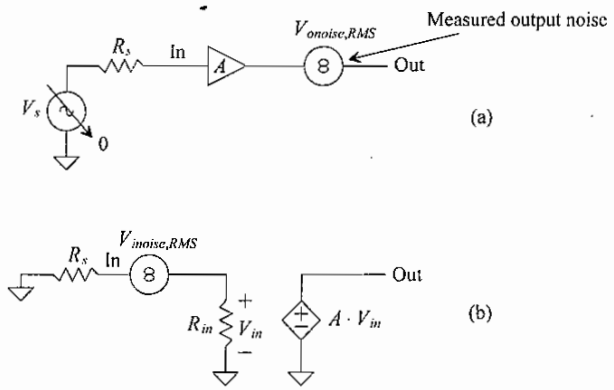
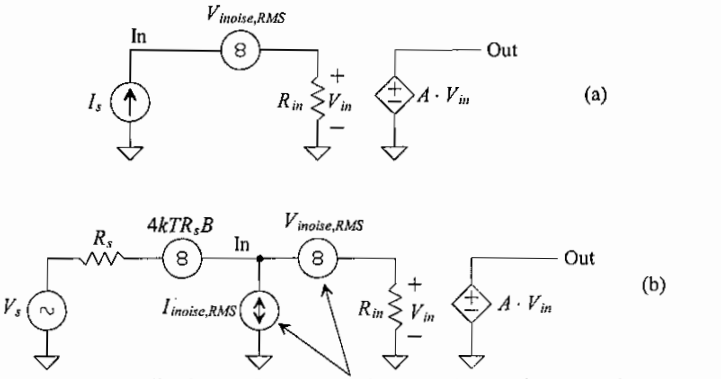


Figure 8.19 Modeling measured output noise with a single input-referred source.

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

$$V_{inoise,RMS} = \frac{V_{onoise,RMS,Rs=0}}{A} \quad (V_{onoise,RMS} \text{ measured with } R_s \text{ shorted}) \tag{8.33}$$

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

$$I_{inoise,RMS} = \frac{V_{onoise,RMS,Rs=\infty}}{AR_{in}} \quad (V_{onoise,RMS} \text{ measured with } R_s \text{ opened}) \tag{8.34}$$

■

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

$$NF = 10 \log \frac{V_{onoise,RMS}^2}{4kTR_sB \cdot \left[\frac{R_{in}}{R_{in}+R_s}\right]^2 \cdot A^2} \tag{8.36}$$

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

$$F = \frac{V_{onoise,RMS}^2}{4kTR_sB \cdot A^2} \tag{8.38}$$

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 $R_{in} = \infty$.

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.

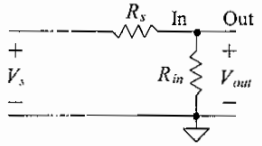


Figure 8.21 Circuit used in Ex. 8.9.

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

$$V^2_{onoise,RMS} = \int_{f_L}^{f_H} V^2_{onoise}(f)\cdot df = 4kTBR_s\left[\frac{R_{in}}{R_{in}+R_s}\right]^2 + 4kTBR_{in}\left[\frac{R_s}{R_{in}+R_s}\right]^2$$

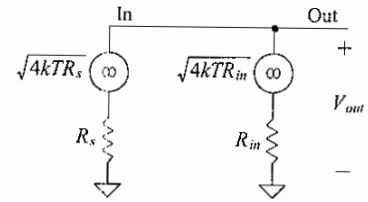
Noting our gain A ($= V_{out}/V_{in}$ not V_{out}/V_s) is one, we can use the model shown in Fig. 8.22b. To determine the input-referred noise sources, we can use Eq. (8.32) and the results in Ex. 8.7. To determine $V_{inoise,RMS}$, we short the input to ground ($R_s = 0$ in Fig. 8.21 and the equation above), Fig. 8.22c, and equate the circuit output to $V_{onoise,RMS}$. This gives

$$V_{inoise,RMS} = V_{onoise,RMS}$$

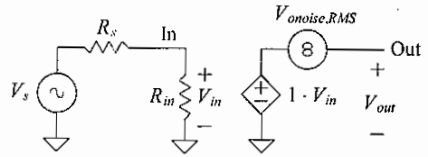
To determine $I_{inoise,RMS}$, we open the input ($R_s = \infty$), Fig. 8.22d, and equate the circuit's output to $V_{onoise,RMS}$ (from the equation above). This gives

$$R^2_{in}\cdot I^2_{inoise,RMS} = V^2_{onoise,RMS,Rs=\infty} = 4kTBR_{in} \rightarrow I_{inoise,RMS} = \sqrt{\frac{4kTB}{R_{in}}}$$

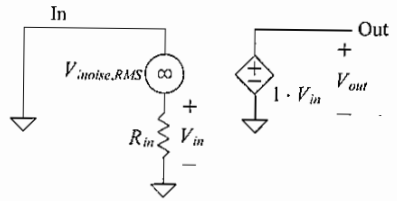
The input SNR is given in Eq. (8.29). The output SNR, Fig. 8.22e, is



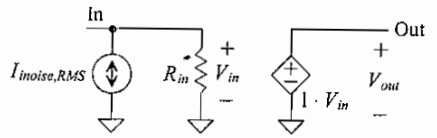
(a) Adding noise sources to the circuit.



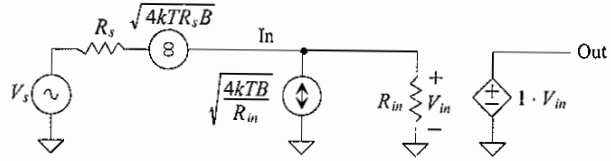
(b) Model showing output noise.



(c) Shorting the input.



(d) Opening the input.



(e) Input-referred noise model.

Figure 8.22 Noise analysis procedure for Ex. 8.9.

$$SNR_{out} = \frac{V_{s,RMS}^2 \cdot \left[\frac{R_{in}}{R_s+R_{in}}\right]^2}{V_{onoise,RMS}^2} = \frac{V_{s,RMS}^2}{4kTB \cdot R_s(1+R_s/R_{in})} \tag{8.40}$$

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 node1 to node2 in the resistor of (a) while it flows from nplus to nminus in the VCCS

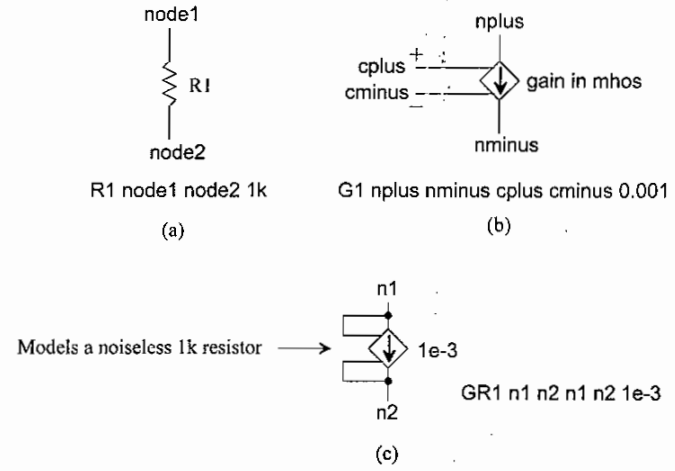


Figure 8.23 Modeling a noiseless resistor in SPICE.

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 $V_{onoise,RMS}^2$ 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
```

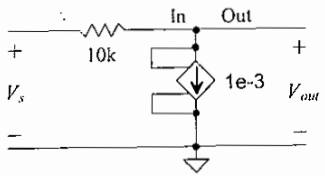


Figure 8.24 Replacing Rin with a noiseless resistor.

```
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. ■

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

$$F = \frac{\overbrace{4kTR_sB \cdot A^2 \cdot \left(\frac{R_{in}}{R_{in}+R_s}\right)^2}^{\text{Output noise from } Rs} + \overbrace{4kT_nR_sB \cdot A^2 \cdot \left(\frac{R_{in}}{R_{in}+R_s}\right)^2}^{\text{Amplifier's output noise alone}}}{\underbrace{4kTR_sB \cdot A^2 \cdot \left(\frac{R_{in}}{R_{in}+R_s}\right)^2}_{\text{Output noise from } Rs}} = 1 + \frac{T_n}{T} \quad \text{where } T_n \geq 0 \tag{8.47}$$

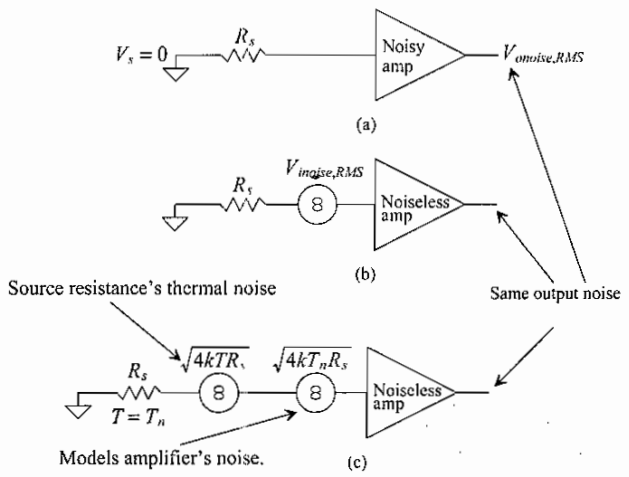


Figure 8.25 Determining the noise temperature of an amplifier.

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.

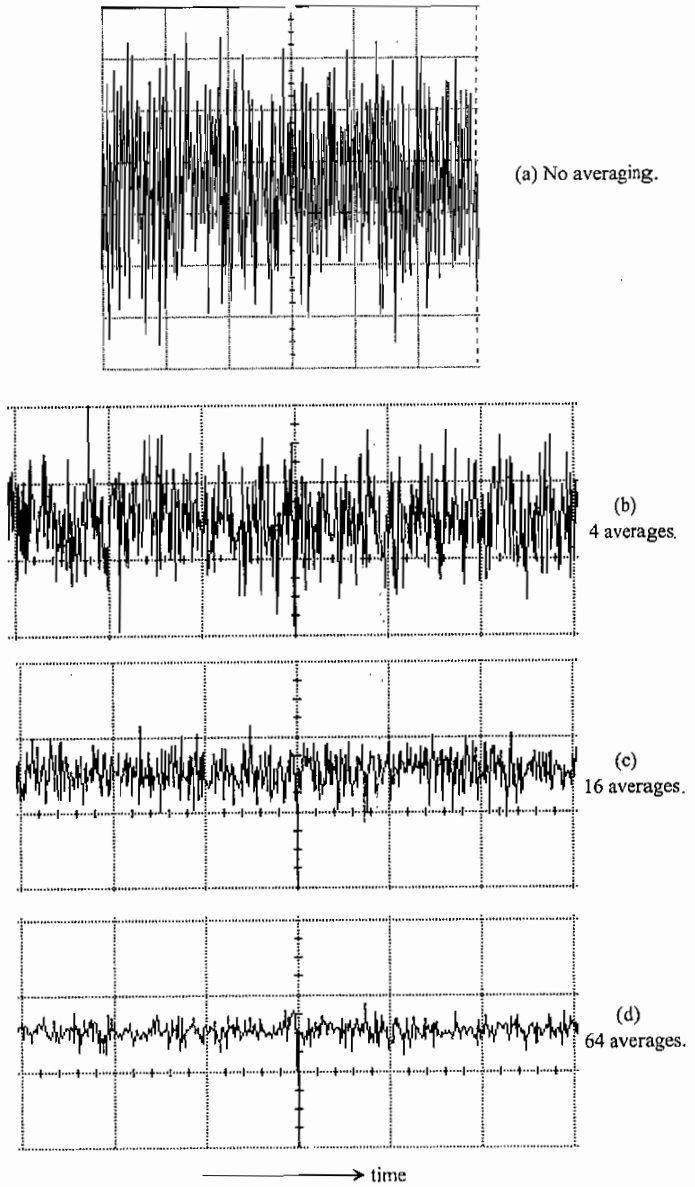


Figure 8.26 Averaging reduces the amplitude of white noise.

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 [9]). 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.

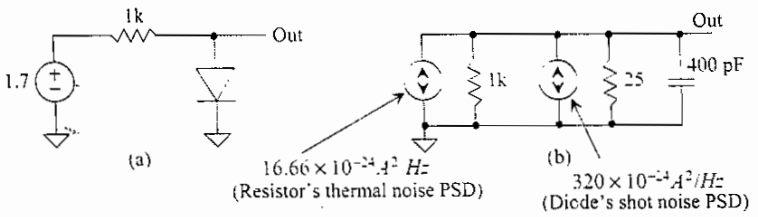


Figure 8.27 Calculating output noise for Ex. 8.12.

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

$$I_{shot}^2(f) = 2qI_{DC} = 2 \cdot (1.6 \times 10^{-19}) \cdot 1mA = 320 \times 10^{-24} A^2/Hz$$

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

$$V_{onoise}^2(f) = 336.6 \times 10^{-24} \cdot (25||1k)^2 = 200 \times 10^{-21} V^2/Hz$$

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
```

```
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 $\sqrt{5.26\times10^{-12}} = 2.28\,\mu V$. 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. ■

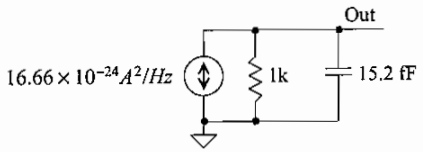


Figure 8.28 Calculating output noise for Ex. 8.12.

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 (pronouced "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.

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).

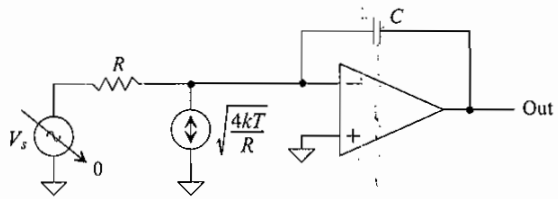


Figure 8.29 Integrating thermal noise.

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 sine wave for that matter, see Fig. 8.5) results in an infinite PSD point if we take $f_{res} \to 0$.

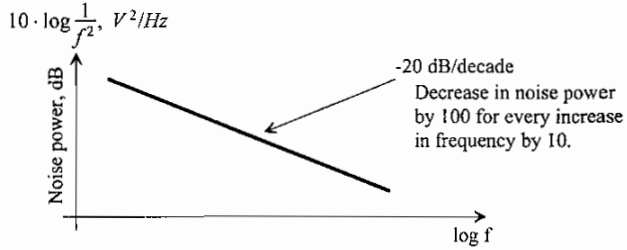


Figure 8.30 PSD of integrated thermal noise (red noise).

Let's use Eq. (8.10) to determine the RMS output noise voltage for this circuit

$$V^2_{onoise,RMS} = \int_{f_L}^{f_H} V^2_{onoise}(f) \cdot df = \frac{4kT}{R} \cdot \frac{1}{(2\pi C)^2} \cdot \left(\frac{1}{f_L} - \frac{1}{f_H}\right) \tag{8.50}$$

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. (8.50),

$$V_{onoise,RMS} \approx \sqrt{\frac{4kT}{R}} \cdot \frac{1}{2\pi C} \cdot \sqrt{T_{meas}} \tag{8.53}$$

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

$$I^2_{1/f}(f) \text{ or } V^2_{1/f}(f) = \frac{FNN}{f} \text{ units of } A^2/Hz \text{ or } V^2/Hz \tag{8.54}$$

where FNN is the flicker noise numerator. The RMS value of a $1/f$ noise signal is

$$V^2_{noise,RMS} = \int_{f_L}^{f_H} \frac{FNN}{f} \cdot df = FNN \cdot \ln\frac{f_H}{f_L} \text{ units of } V^2 \tag{8.55}$$

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.

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.

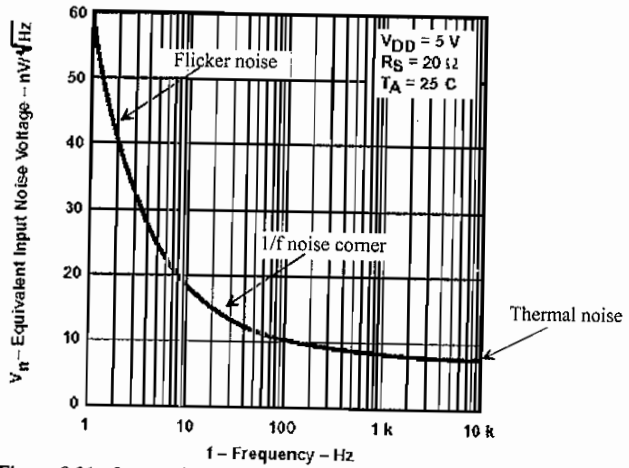


Figure 8.31 Input-referred noise for the TLC220x low-noise op-amp.

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

$$FNN = (56\,nV)^2 = 3.14 \times 10^{-15} V^2$$

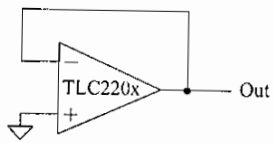


Figure 8.32 Estimating RMS output noise in a voltage follower.

As a quick check we see, in Fig. 8.31, that at 4 Hz the spectral density is roughly 28 nV/$\sqrt{Hz}$ or

$$V_{inoise,1/f}(f) = \sqrt{\frac{FNN}{f}} = \sqrt{\frac{3.14\times10^{-15}V^2}{4\ Hz}} = \frac{56\ nV}{\sqrt{4\ Hz}} = 28\ nV/\sqrt{Hz}$$

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

$$V^2_{onoise}(f) = V^2_{inoise}(f) = \overbrace{\frac{3.14\times10^{-15}}{f}}^{V^2_{inoise,1/f}(f),\ \text{flicker noise}} + \overbrace{64\times10^{-18}}^{\text{thermal noise}} \text{ with units of } V^2/Hz$$

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

$$V^2_{onoise,RMS} = V^2_{inoise,RMS} = \overbrace{49\cdot(3.14\times10^{-15})}^{\text{using Eq. (8.56)}} + \overbrace{(64\times10^{-18})\cdot(2\times10^{6})\cdot\frac{\pi}{2}}^{\text{using Eq. (8.15)}}$$

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).

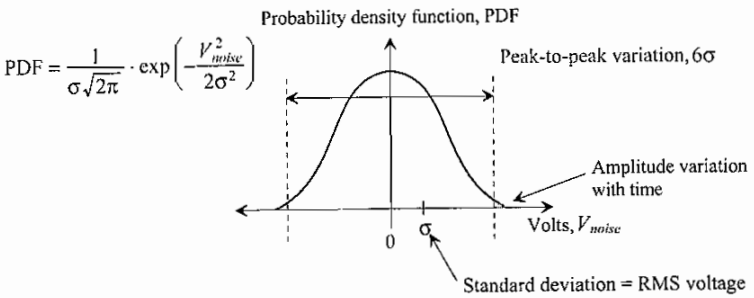


Figure 8.33 Gaussian probability distribution.

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

$$V^2_{inoise,RMS} = \int_{f_L}^{f_H}\frac{3.14\times10^{-15}}{f}\cdot df = 3.14\times10^{-15}\cdot\ln\frac{1}{0.1}$$

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).

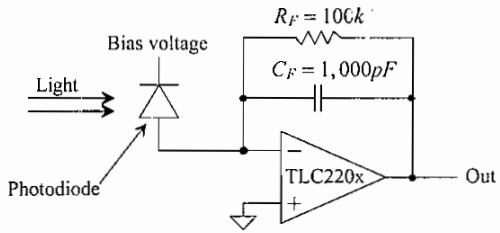


Figure 8.34 Estimating RMS output noise in a transimpedance amplifier.

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 1,000pF.

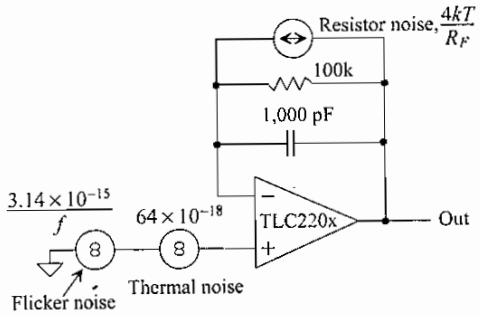


Figure 8.35 Noise model for the transimpedance amplifier in Fig. 8.34.

The output noise PSD can then be written as

$$V^2_{onoise}(f) = \frac{3.14\times10^{-15}}{f} + 64\times10^{18} + \overbrace{\frac{4kT}{R_F}\cdot\left|\frac{R_F\cdot 1/j\omega C_F}{R_F+1/j\omega C_F}\right|^2}^{\frac{4kTR_F}{1+(2\pi fR_FC_F)^2}}$$

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

$$A(f) = R_m = \frac{v_{out}}{i_s} = R_F \| \frac{1}{j\omega C_F} = \frac{R_F}{1+j\omega R_F C_F}$$

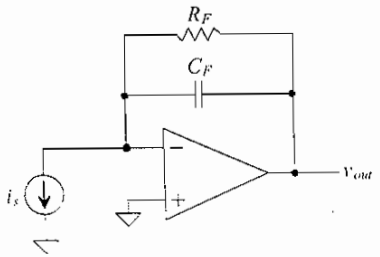


Figure 8.36 Determining the transfer function of a transimpedance amplifier.

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 as

$$V^2_{onoise,RMS} = \overbrace{49\cdot 3.14\times10^{-15}}^{=154\times10^{-15}} + \overbrace{64\times10^{-18}\cdot\frac{\pi}{2}\cdot 1.59\,kHz}^{=160\times10^{-15}} + \overbrace{\frac{kT}{C_F}}^{=4\times10^{-12}}$$

or

$$V_{onoise,RMS} \approx 2.07\,\mu V$$

noting that the thermal noise from R_F dominates the output RMS noise (the *kT/C* 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 to approximately 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 20.7 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, $> 6\times2.07\,\mu V$), then $i_s > 125\,pA$. ■

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

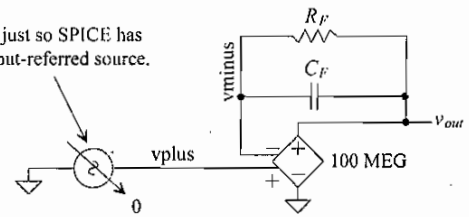


Figure 8.37 Using a voltage-controlled voltage source for an op-amp in SPICE.

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
Noise analysis ... 100%
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 burst (popcorn) and avalanche noise.

Burst (Popcorn) Noise

Burst noise is seen, in the time-domain, in Fig. 8.38. Burst noise is often called popcorn noise. When a signal containing burst noise is played on a speaker, the popcorn noise component sounds like popping corn. The origin of popcorn 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 a popcorn 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 popcorn noise doesn't have a Gaussian noise distribution, its PSD can be modeled using

$$I^2_{popcorn}(f) = \frac{K_{popcorn} \cdot I_{DC}}{1 + \left(\frac{f}{f_{3dB}}\right)^2} \text{ units of } A^2/Hz \qquad (8.58)$$

where $K_{popcorn}$ 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, popcorn noise is white (like thermal and shot noises). However, at higher frequencies, the noise rolls off as $1/f^2$.

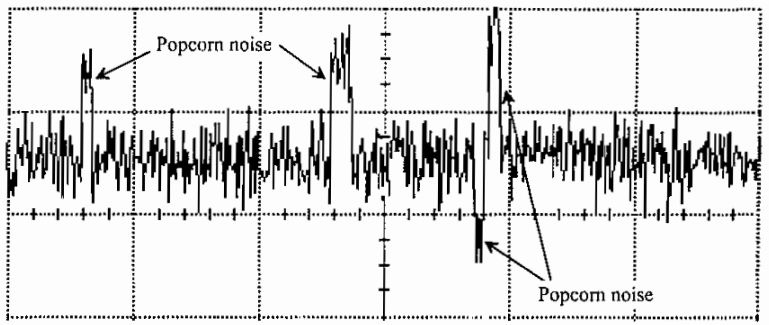


Figure 8.38 Popcorn (burst) noise viewed in the time domain.

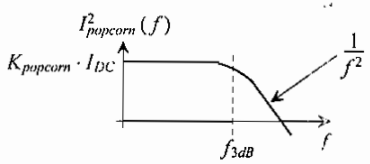


Figure 8.39 PSD of popcorn noise.

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.

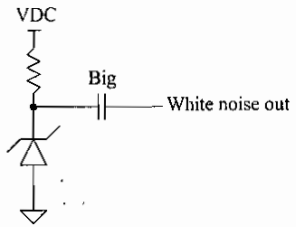


Figure 8.40 A Zener diode-based white noise generator.

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

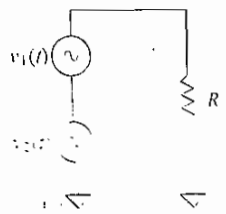


Figure 8.41 Series combination of voltage signals.

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 sine wave 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.

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.

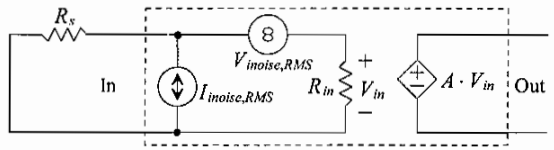


Figure 8.42 Correlation calculation of input-referred noise sources.

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

$$|R_s + Z_{in}|^2 = \left(\sqrt{R_s^2 + (1/\omega C_{in})^2}\right)^2 = R_s^2 + (1/\omega C_{in})^2 \qquad (8.72)$$

The incorrect way is

$$|R_s + Z_{in}|^2 \neq |(R_s + 1/j\omega C_{in})(R_s + 1/j\omega C_{in})| = |\overbrace{R_s^2 + 1/\omega^2 C_{in}^2}^{\text{real term}} + j \cdot \overbrace{(-2/\omega C_{in})}^{\text{imaginary term}}|$$
$$= \sqrt{\left(R_s^2 + 1/\omega^2 C_{in}^2\right)^2 + (2/\omega C_{in})^2} \text{ (again wrong!)} \qquad (8.73)$$

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,

$$V_{onoise,RMS,Rs=\infty} = V_{onoise,RMS,Rs=0} = V_{onoise,RMS} \qquad (8.74)$$

and the input-referred noise sources are uncorrelated then, for a real input resistance,

$$R_{s,opt} = R_{in} \qquad (8.75)$$

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

$$R_{s,opt} = |Z_{in}| = |R_{in} + jX_{in}| = \sqrt{R_{in}^2 + X_{in}^2} \qquad (8.76)$$

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 acnieve 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.

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

$$I_{inoise}(f) = 62.8 \times 10^{-21} \cdot f\ A/\sqrt{Hz}$$

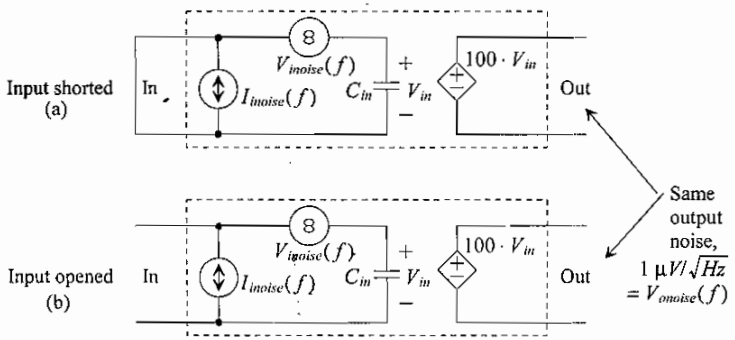


Figure 8.43 Determining input-referred noise in Ex. 8.19.

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. ■

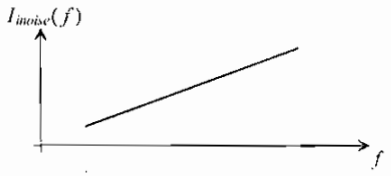


Figure 8.44 Input-referred noise current increasing with frequency when the input impedance is capacitive.

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

$$V_{out} = [(V_{in} - \beta V_{out}) + V_{inoise,RMS}] \cdot A \qquad (8.77)$$

or

$$V_{out} = \frac{A}{1 + \beta A} \cdot (V_{in} + V_{inoise,RMS}) \qquad (8.78)$$

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.

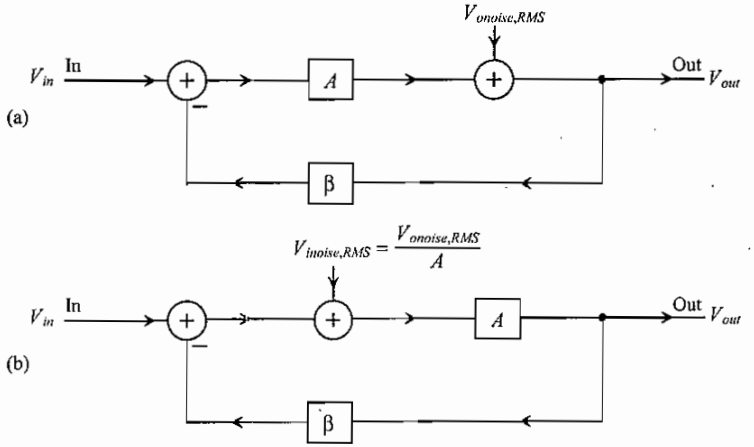


Figure 8.45 Noise in a feedback circuit.

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.

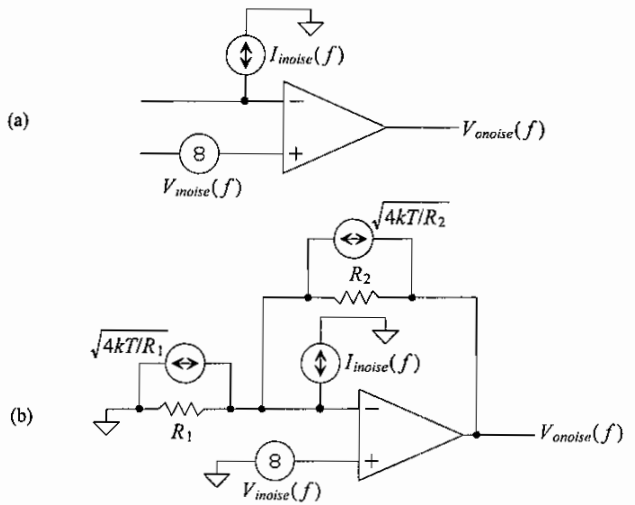


Figure 8.46 Modeling op-amp noise.

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

$$V^2_{onoise}(f) = V^2_{inoise}(f) \cdot \left(1 + \frac{R_2}{R_1}\right)^2 + \left[I^2_{inoise}(f) + \frac{4kT}{R_1} + \frac{4kT}{R_2}\right] \cdot R_2^2 \qquad (8.79)$$

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

$$V_{onoise,RMS} = \sqrt{V^2_{onoise}(f) \cdot \frac{\pi}{2} \cdot f_{un} \frac{R_1}{R_1 + R_2}} \qquad (8.82)$$

where $V^2_{onoise}(f)$ is assumed to be white until it roll-offs with the amplifer's bandwidth, Fig. 8.9b.

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 $I_{inoise}(f) = 8.5\ pA/\sqrt{Hz}$ (= 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.

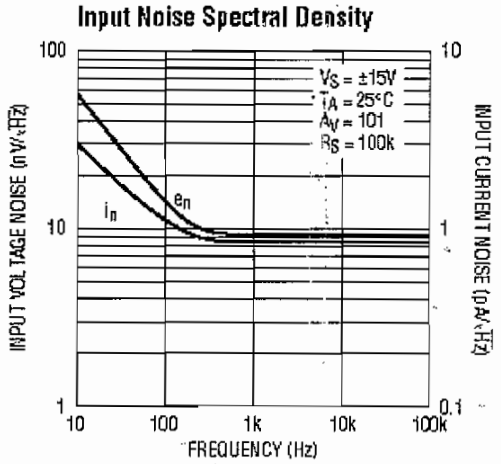


Figure 8.47 Input-referred noise sources for the LT1364 high-speed op-amp.

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

$$\overline{v^2} = V_{RMS}^2 = \frac{1}{T_{meas}} \int_0^{T_{meas}} v^2(t) \cdot dt \qquad (8.83)$$

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

$$\overline{v^2} = 4kTR \text{ or } \overline{i^2} = \frac{4kT}{R} \qquad (8.84)$$

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, Δf, is used for the resolution bandwidth f_{res}. It is correct to write the PSD of thermal noise as

$$V_R^2(f) = \frac{V_{RMS}^2}{f_{res}} = \frac{\overline{v^2}}{\Delta f} = 4kTR \text{ units of } \frac{V^2}{Hz} \qquad (8.85)$$

It is also correct to write

$$\overline{v^2} = 4kTR \cdot \Delta f \text{ volts} \qquad (8.86)$$

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

$$V_{RMS}^2 = \overline{v^2} = \int_{f_L}^{f_H} V_R{}^2(f) \cdot df = V^2(f) \cdot B = 4kTRB \qquad (8.87)$$

where $B = f_H - f_L \neq \Delta f = f_{res}$. Note that Δf 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

$$V_{1/f}^2(f) = \frac{\overline{v^2}}{\Delta f} = \frac{FNN}{f} \qquad (8.88)$$

at each frequency f. The measured power in the narrow bandwidth Δf (= f_{res}), $\overline{v^2}$ (= V_{RMS}^2), decreases as f goes up resulting in the $1/f$ PSD.

ADDITIONAL READING AND/OR INFORMATION

[1] R. J. Baker, *CMOS: Mixed-Signal Circuit Design*, John Wiley and Sons, 2002. ISBN 0-471-22754-4. Volume 2 of this book. Covers averaging and how it affects circuit noise.

[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] 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.

[5] 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.

[6] 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.

[7] A. Van der Ziel, *Noise; Sources, Characterization, Measurement*, Prentice-Hall. Also written by a pioneer in noise theory.

[8] 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.

[9] 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.

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.

Δf - 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 sine wave.

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 Δf).

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.

$I_{popcorn}^2(f)$ PSD of popcorn (burst) noise current, units of A^2/Hz.

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 $\overline{i^2}$.

$I_{shot}^2(f)$ - 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.

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.

$P_{instr}(t)$ - 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 1.6×10^{-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.

$V_{1/f}^2(f)$ - 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.

ω - $2\pi f$.

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.)

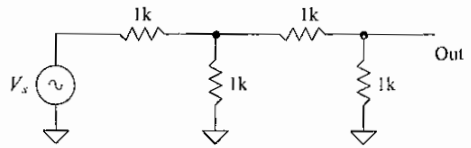


Figure 8.48 Circuit for Problem 8.5.

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.

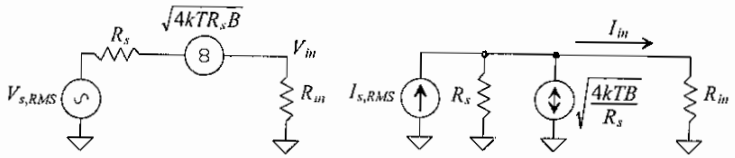


Figure 8.49 Calculating input SNR in Problem 8.7.

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).

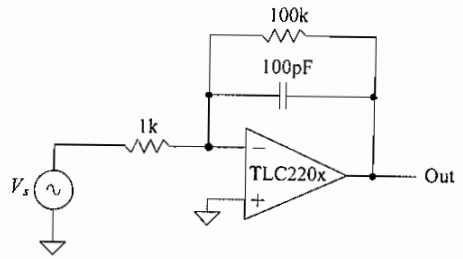


Figure 8.50 Circuit for Problem 8.15.

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?

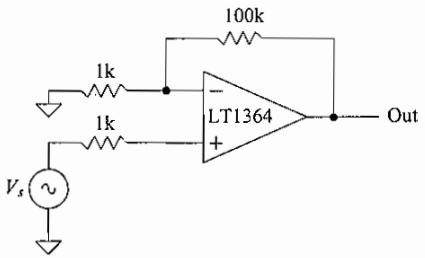


Figure 8.51 Circuit for Problem 8.18.

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

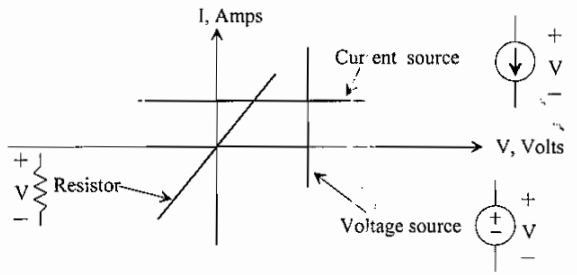


Figure 9.1 Current-voltage (IV) plots for various electrical components.

plot). 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.

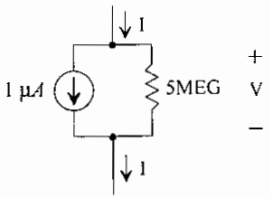


Figure 9.2 Circuit for Ex. 9.1.

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. ■

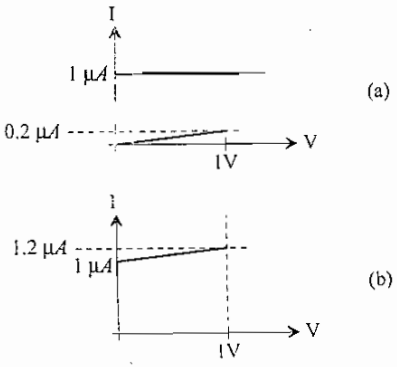


Figure 9.3 IV plots for Ex. 9.1.

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.

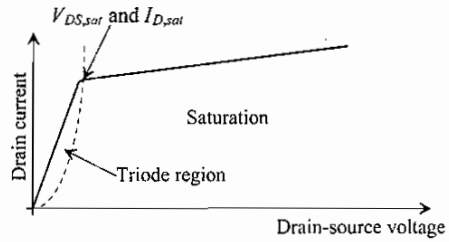


Figure 9.4 IV curves of a MOSFET.

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

$$I_D = I_{D,sat} + I_{D,sat}\lambda \cdot (V_{DS} - V_{DS,sat}) \qquad (9.5)$$

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 series with a resistor of value

$$r_o = \frac{1}{\lambda I_{D,sat}} \qquad (9.6)$$

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

$$\overbrace{V_D - V_S}^{V_{DS}} \geq \overbrace{V_G - V_S}^{V_{GS}} - V_{THN} \text{ or } V_D \geq V_G - V_{THN} \qquad (9.7)$$

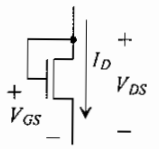


Figure 9.5 Gate-drain connected MOSFET. Also known as a diode-connected MOSFET.

PMOS Square-Law Equations

The PMOS equivalents of Eqs. (9.1) and (9.2) are (for completeness)

$$I_D = \frac{KP_p}{2} \cdot \frac{W}{L}(V_{SG} - V_{THP})^2(1 + \lambda(V_{SD} - V_{SD,sat})) \qquad (9.8)$$

and

$$V_{SD,sat} = V_{SG} - V_{THP} \qquad (9.9)$$

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

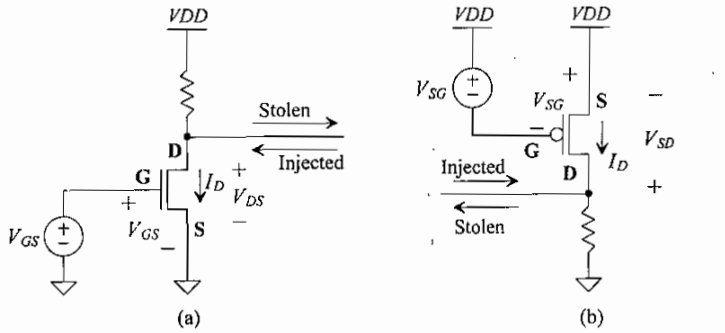


Figure 9.6 Movement of voltages and currents in a MOSFET, see text.

of 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, λ, 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 $-1\,\mu A \leq I \leq 1\,\mu A$. 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

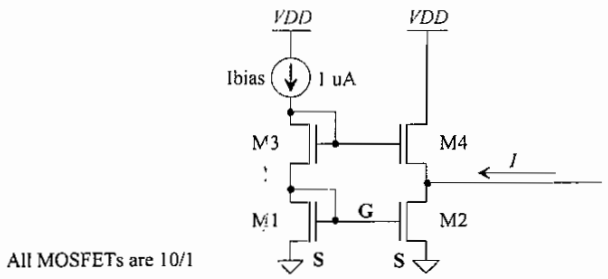


Figure 9.7 Schematic for Ex. 9.3.

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. ■

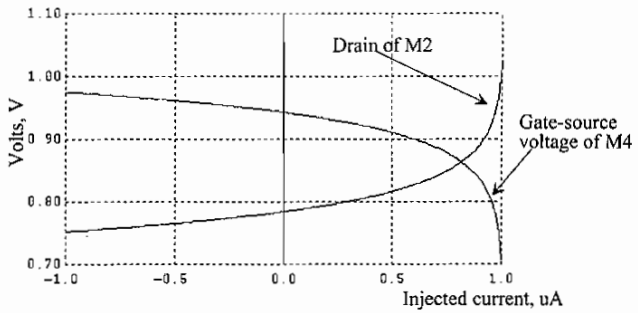


Figure 9.8 SPICE simulations verifying the discussions in Ex. 9.3.

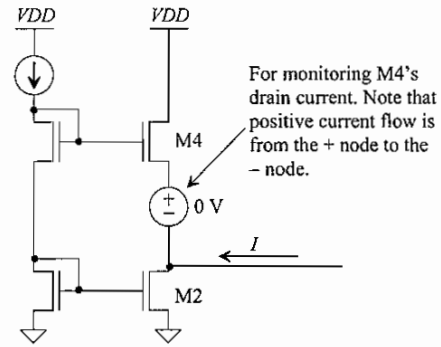


Figure 9.9 Adding zero volt sources to monitor currents in SPICE.

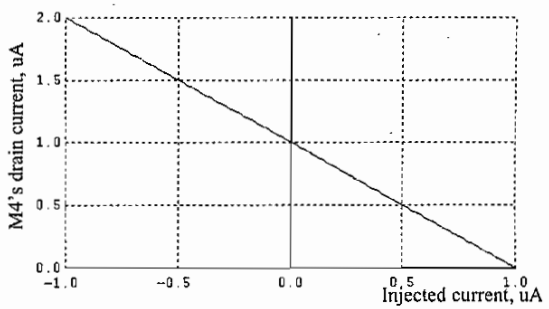


Figure 9.10 How the drain current of M4 changes with injected current.

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)

$$V_{THP}(V_{BS}) = V_{THP0} + \gamma_p\left(\sqrt{2|V_{fn}| + V_{BS}} - \sqrt{2|V_{fn}|}\right) \qquad (9.11)$$

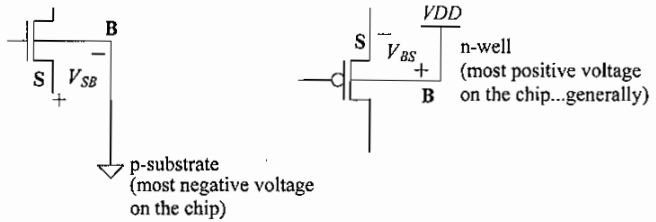


Figure 9.11 The body connection of a MOSFET.

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.

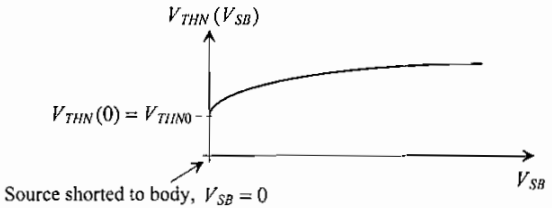
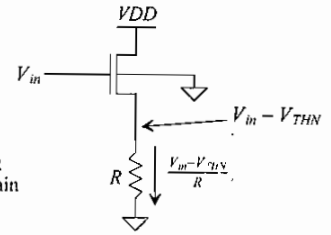


Figure 9.12 Variation in threshold voltage with source to bulk potential.



The resistor is big so that $V_{GS} \approx V_{THN}$ (a small drain current).

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, 100×10^{-9} A/V. ■

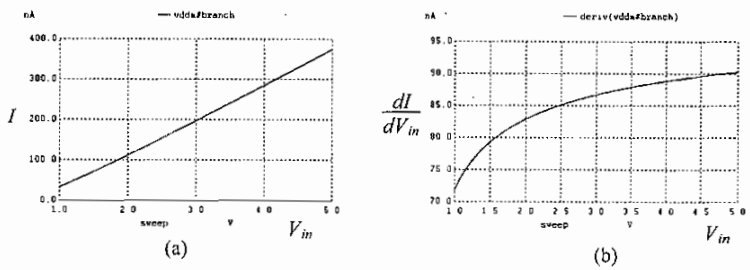


Figure 9.14 (a) Current flowing in the circuit of Fig. 9.13 and (b) its linearity.

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

$$I_D = KP_n \frac{W}{L} \cdot \left((V_{GS} - V_{THN}) V_{DS} - \frac{V_{DS}^2}{2} \right) \qquad (9.12)$$

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

$$I_D = KP_p \frac{W}{L} \cdot \left((V_{SG} - V_{THP}) V_{SD} - \frac{V_{SD}^2}{2} \right) \qquad (9.13)$$

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

$$R_{ch}^{-1} = \frac{\partial I_D}{\partial V_{DS}} = KP_n \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN}) - KP_n \cdot \frac{W}{L} \cdot V_{DS} \qquad (9.14)$$

or

$$R_{ch} = \frac{1}{KP_n \cdot \frac{W}{L} \cdot (V_{DS,sat} - V_{DS})} \qquad (9.15)$$

If $V_{DS,sat} >> V_{DS}$, this equation can be written as

$$R_{ch} \approx \frac{1}{KP_n \cdot \frac{W}{L} (V_{GS} - V_{THN})} \qquad (9.16)$$

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

$$I_D = I_{D0} \cdot \frac{W}{L} \cdot e^{(V_{GS} - V_{THN})/nV_T} \qquad (9.17)$$

$V_{GS} \leq V_{THN}$ 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).

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 [$V_{GS} < V_{THN}$].)

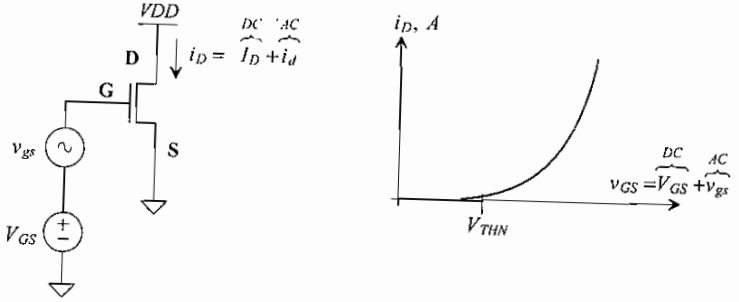


Figure 9.15 IV curves of a MOSFET in saturation.

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).

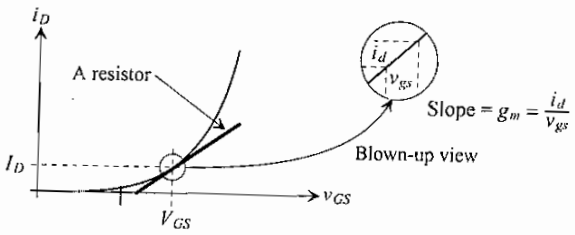


Figure 9.16 How small-signal parameters are calculated.

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

$$\beta_n = KP_n \cdot \frac{W}{L} \quad \text{and} \quad |v_{gs}| << V_{GS} \tag{9.21}$$

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 $V_{DS,sat}$.

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.

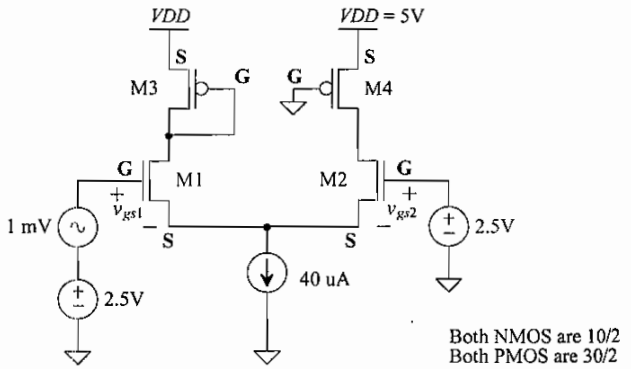


Figure 9.17 Circuit discussed in Ex. 9.5.

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

$$V_{GS} = \overbrace{\sqrt{\frac{2I_D}{KP_n} \cdot \frac{L}{W}}}^{V_{DS,sat}} + V_{THN} \tag{9.23}$$

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)

$$V_{D1} \geq V_{G1} - V_{THN} \rightarrow 3.842 \geq 1.7\ V \text{ (yes, M1 is in saturation)}$$

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

$$\overbrace{V_S - V_D}^{V_{SD}} \geq \overbrace{V_S - V_G}^{V_{SG}} - V_{THP} \rightarrow V_D \leq V_G + V_{THP} \qquad (9.24)$$

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
```



```
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

$$\frac{1}{g_m} = \frac{v_{sd}}{i_d} = \frac{v_{sg}}{i_d} \tag{9.25}$$

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).

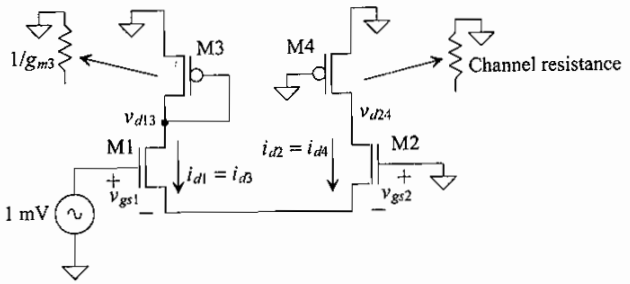


Figure 9.18 AC schematic for the circuit in Fig. 9.17.

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

$$i_{D1} = 20\ \mu A + 0.075 \sin 2\pi f$$

where f is the frequency of our 1 mV input signal and

$$i_{D2} = 20\ \mu A - 0.075 \sin 2\pi f$$

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
```

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.

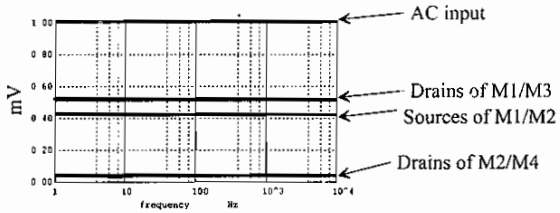


Figure 9.19 AC simulation results for the circuit in Fig. 9.17.

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

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.

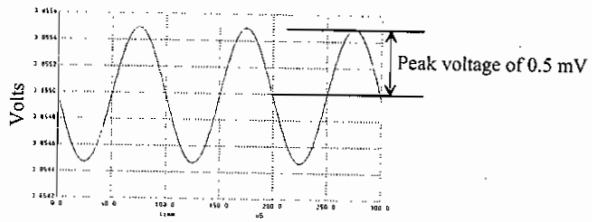


Figure 9.20 Transient simulation showing the drain voltages of M1 and M3 from Fig. 9.17.

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.

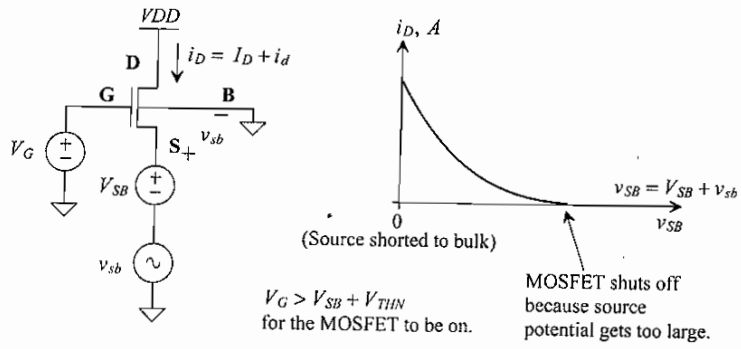


Figure 9.21 How the drain current changes with body-effect.

Remembering that the body-effect is the variation of the threshold voltage with V_{SB}, we can write

$$g_{mb} = \left[\frac{\partial i_D}{\partial v_{SB}}\right]_{V_{SB}=\text{constant}}^{I_D=\text{constant}} = \frac{\partial}{\partial v_{SB}}\left[\frac{KP_n}{2}\cdot\frac{W}{L}(V_{GS}-V_{THN})^2\right]_{V_{SB}=\text{constant}}^{I_D=\text{constant}} \qquad (9.26)$$

or

$$g_{mb} = \overbrace{KP_n \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN})}^{g_m} \cdot \left(-\frac{\partial V_{THN}}{\partial v_{SB}}\right) \qquad (9.27)$$

or (noting there is a variation in v_{GS} with v_{SB} but this is simply the forward g_m).

$$g_{mb} = g_m \cdot \eta \qquad (9.28)$$

The factor η 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.

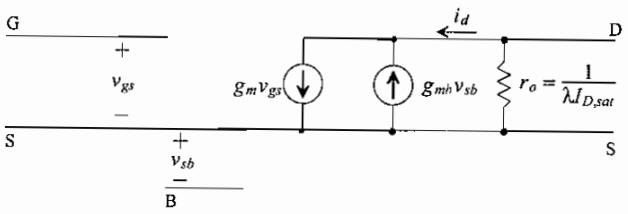


Figure 9.22 Small-signal MOSFET with both transconductances.

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

$$r_o^{-1} = \left[\frac{\partial i_D}{\partial v_{DS}}\right]_{\substack{I_D = \text{constant} \\ V_{DS} = \text{constant}}} = \frac{\partial}{\partial v_{DS}}\left(\frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - V_{THN})^2\left(1 + \lambda\left(\overbrace{v_{ds} + V_{DS}}^{v_{DS}} - V_{DS,sat}\right)\right)\right) \qquad (9.29)$$

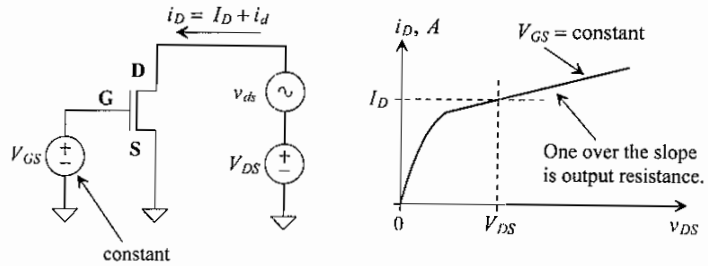


Figure 9.23 How the drain current changes with drain-to-source voltage.

or, once again,

$$r_o = \frac{1}{\lambda I_{D,sat}} \qquad (9.30)$$

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. ■

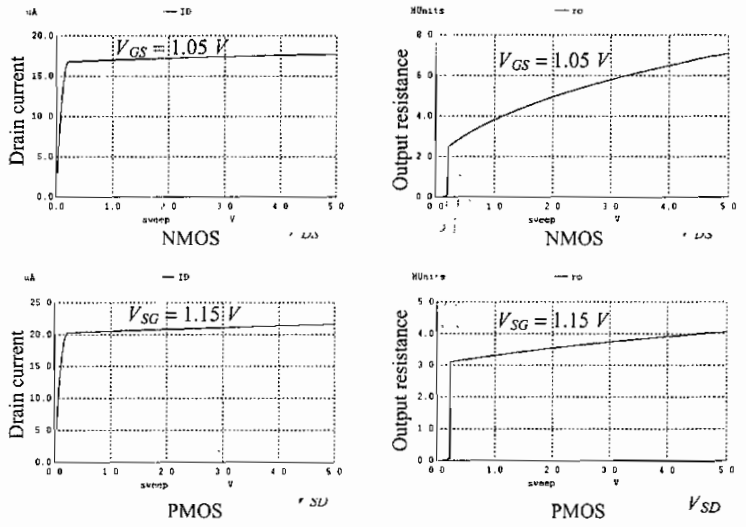


Figure 9.24 Using simulations to determine lambda.

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

$$\left|\frac{i_d}{i_g}\right| = \frac{g_m}{2\pi f \cdot (C_{gs} + C_{gd})} \tag{9.35}$$

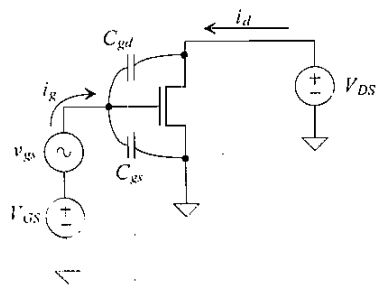


Figure 9.25 [illegible]

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])

$$f_T \approx \frac{g_m}{2\pi C_{gs}} = \frac{3KP_n \cdot (V_{GS} - V_{THN})}{4\pi \cdot L^2 C'_{ox}} = \frac{3\mu_n}{4\pi} \cdot \frac{V_{DS,sat}}{L^2} \tag{9.36}$$

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 μ_n/L as a relatively constant value and rewrite Eq. (9.36) as

$$f_T \propto \frac{V_{DS,sat}}{L} \text{ (Short-channel devices)} \tag{9.37}$$

Again, for high-speed, we need to use the smallest possible channel length and large $V_{DS,sat}$.

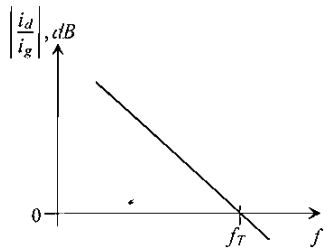


Figure 9.26 The transition frequency of a MOSFET.

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.

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 *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 fF/µm²	1.75 fF/µm²	$C_{ox} = C'_{ox}WL \cdot (scale)^2$
C_{oxn} and C_{oxp}	35 fF	105 fF	PMOS is three times wider
C_{gsn} and C_{sgp}	23.3 fF	70 fF	$C_{gs} = \frac{2}{3}C_{ox}$
C_{gdn} and C_{dgp}	2 fF	6 fF	$C_{gd} = CGDO \cdot W \cdot scale$
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
λ_n and λ_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

$$g_m = \left[\frac{\delta i_D}{\delta v_{GS}}\right]_{\substack{I_D = \text{constant} \\ V_{GS} = \text{constant}}} = I_{D0} \cdot \frac{W}{L} \cdot e^{\left(\frac{v_{GS}}{V_{GS} + v_{gs}} - V_{THN}\right)\frac{1}{nV_T}} \cdot \left(\frac{1}{nV_T}\right) \quad (9.38)$$

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.

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).

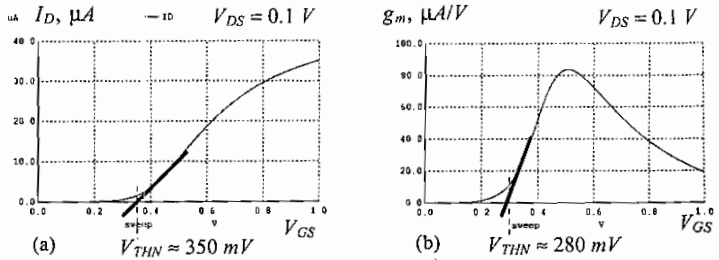


Figure 9.27 (a) Drain current plotted against gate-source voltage with small Vds and (b) transconductance plotted against gate-source voltage.

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

$$I_{DS,sat} = \frac{\mu_n \cdot \varepsilon_{ox}}{2t_{ox}} \cdot \frac{W}{L} \cdot (V_{GS} - V_{THN})^2 \quad (9.40)$$

(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

$$V_{THN} = -V_{ms} - 2V_{fp} + \frac{Q'_{b0} - Q'_{ss}}{C'_{ox}} \quad (9.41)$$

noting Q'_{ss} is a constant and $Q'_{b0} = \sqrt{2qN_A\varepsilon_{si}|-2V_{fp}|}$. Knowing

$$V_{fp} = -\frac{kT}{q}\ln\frac{N_A}{n_i} \text{ and } V_{ms} = V_G - V_{fp} = \frac{kT}{q}\ln\frac{N_{D,poly}}{n_i} - V_{fp} \quad (9.42)$$

the change in the threshold voltage with temperature is

$$\frac{\partial V_{THN}}{\partial T} = -\frac{k}{q} \cdot \ln \frac{N_{D,poly}}{N_A} + \frac{Q'_{h0}}{C'_{ox} \cdot 2T} \approx -\frac{k}{q} \cdot \ln \frac{N_{D,poly}}{N_A} \qquad (9.43)$$

The term k/q is the change in the thermal voltage with temperature, that is,

$$\frac{\partial V_T}{\partial T} = \frac{\partial}{\partial T}\left(\frac{kT}{q}\right) = \frac{k}{q} = 0.085\ mV/C° \qquad (9.44)$$

(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

$$\frac{\partial V_{THN}}{\partial T} \approx -1\ mV/C° \qquad (9.45)$$

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

$$\frac{\partial V_{THN}}{\partial T} \approx -0.6\ mV/C° \qquad (9.46)$$

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

$$TCV_{THN} = \frac{1}{V_{THN}} \cdot \frac{\partial V_{THN}}{\partial T} \qquad (9.47)$$

so that the threshold voltage can be written as a function of temperature as

$$V_{THN}(T) = V_{THN}(T_0) \cdot (1 + TCV_{THN} \cdot (T - T_0)) \qquad (9.48)$$

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/$C°$. For a short-channel process with V_{THN} = 0.28 V, we get a temperature coefficient of –2,143 ppm/$C°$. 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).

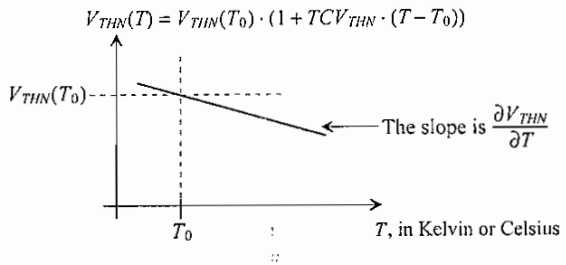


Figure 9.28 Temperature dependence of the threshold voltage.

Mobility Variation with Temperature

The reduction in mobility with increasing temperature is modeled using

$$\mu(T) = \mu(T_0) \cdot \left(\frac{T_0}{T}\right)^{1.5} \qquad (9.49)$$

and so it follows that the reduction in the transconductance parameter with increasing temperature is

$$KP(T) = KP(T_0) \cdot \left(\frac{T_0}{T}\right)^{1.5} \qquad (9.50)$$

where, once again, the mobility is measured at temperature T_0. The change in the MOSFET transconductance with temperature around T_0 is

$$\left[\frac{\partial KP(T)}{\partial T}\right]_{T_0 = \text{constant}} = KP(T_0) \cdot (-1.5)\left(\frac{T_0}{T}\right)^{2.5} \cdot \frac{1}{T_0} \qquad (9.51)$$

The temperature coefficient of the transconductance parameter around the temperature T_0 is then

$$\frac{1}{KP(T)} \cdot \frac{\partial KP(T)}{\partial T} = -\frac{1.5}{T} \qquad (9.52)$$

The transconductance parameter at a particular temperature close to T_0 is given by

$$KP(T) = KP(T_0) \cdot \left(1 - 1.5 \cdot \frac{T - T_0}{T}\right) \qquad (9.53)$$

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). As a simple example of when Eq. (9.53) doesn't work well, consider what happens when $T = 3T_0$. The mobility and KP go to zero. What happens if the measured temperature is 10° C (yes, cool, but it does illustrate the point). Does the mobility go to zero at 30° C?

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).

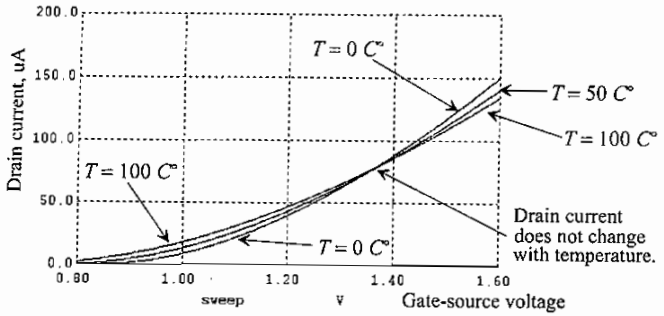
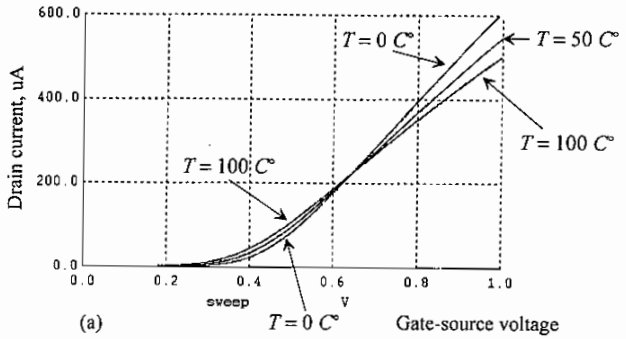
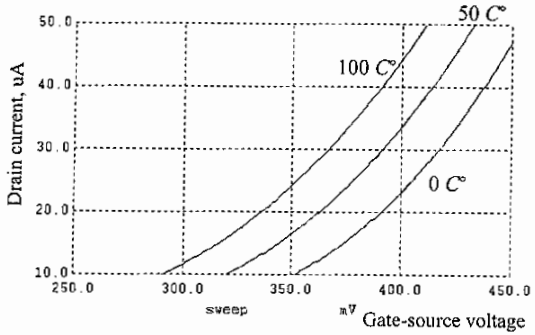


Figure 9.29 Long-channel drain current change with temperature.



(a)



(b) Zoomed-in view of (a).

Figure 9.30 Short-channel drain current change with temperature.

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

$$g_m = \left[\frac{\partial i_D}{\partial v_{GS}}\right]_{\substack{I_D = \text{constant} \\ V_{GS} = \text{constant}}} = v_{sat} \cdot C'_{ox} \cdot W \tag{9.57}$$

The g_m of a short-channel device operating in saturation depends only on the width. If $v_{sat} = 2.5 \times 10^5\ cm/s = 2.5 \times 10^9\ \mu m/s$ (a reasonable value for a short-channel MOSFET), $t_{ox} = 14$ Å so that $C'_{ox} = 25\ fF/\mu m^2$ and we want a g_m of 150 µA/V (picked [arbitrarily] so that our short-channel MOSFET has the same transconductance as the long-channel MOSFETs described in Table 9.1), then $W = 2.4$ µm. With a scale factor of 50 nm, that makes (without scaling) $W = 48$. As a first cut, let's use $V_{ovn} = 70$ mV, $W = 50$, and $L = 2$. 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).

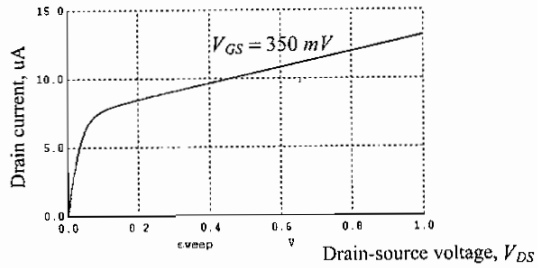


Figure 9.31 IV curves for a 50/2 NMOS with VGS = 350 mV.

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}.

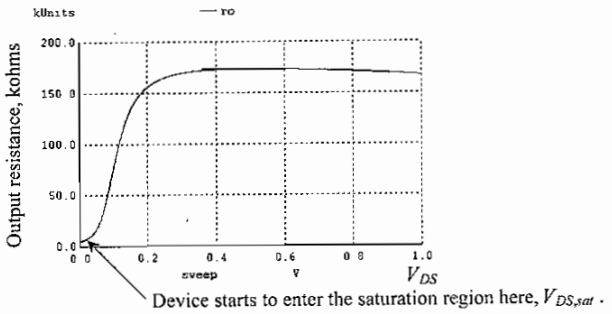


Figure 9.32 Output resistance of the MOSFET in Fig. 9.31 against V_{DS}.

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_{GT}, 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

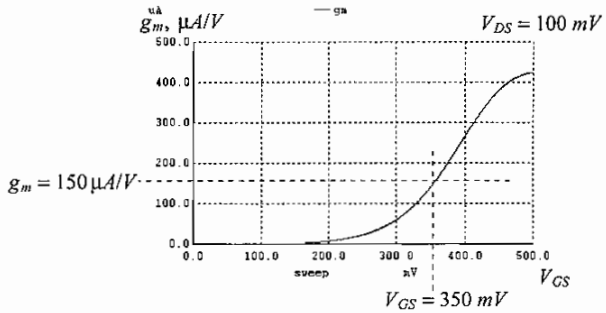


Figure 9.33 How transconductance changes with gate-source bias.

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.

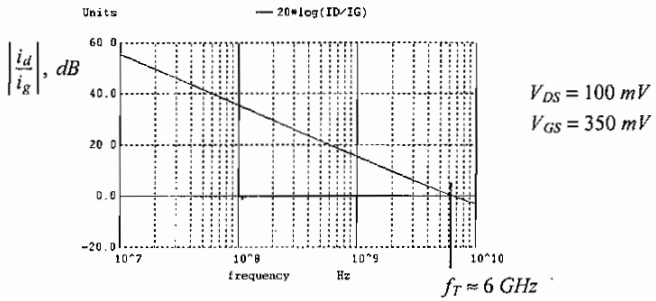


Figure 9.34 MOSFET transition frequency.

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.

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 *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 $V_{SD,sat}$ V_{ovn} and V_{ovp}	50 mV 70 mV	50 mV 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!!
λ_n and λ_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 can be written as a product of the open-circuit gain of the MOSFET with its 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], and [9.36]) knowing $C_{gs} = \frac{2}{3}C'_{ox}WL$ as

$$\text{GFT} = g_m r_o \cdot f_T = \frac{g_m^2}{2\pi C_{gs}} \cdot \frac{1}{\lambda I_D} = \frac{3\mu_n}{2\pi \cdot L^2 \lambda} \tag{9.59}$$

Notice that the expression is *independent of drain current*. It is based entirely on the channel length 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 lead to *wasted time* when trying to increase both the gain and bandwidth of an amplifier.

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, 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. We can also write

$$g_m r_o = \frac{KP_n\frac{W}{L}(V_{GS} - V_{THN})}{\lambda \cdot \frac{KP_n}{2}\frac{W}{L}(V_{GS} - V_{THN})^2} = \frac{2}{\lambda(V_{GS} - V_{THN})} = \frac{2}{\lambda V_{DS,sat}} \tag{9.61}$$

showing gain decreases with increasing gate-source voltage (but, of course, f_T goes up). When operating in the *subthreshold region*, 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, see Fig. 9.35.

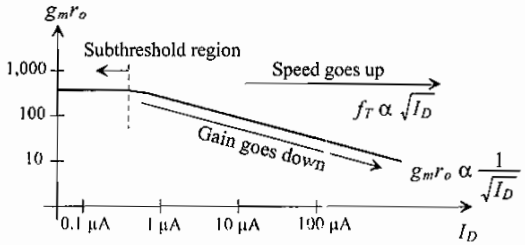


Figure 9.35 Gain falling off with bias current.

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

$$\overbrace{g_m r_o}^{\text{open circuit gain} = 25} \cdot f_T = 150\tfrac{\mu A}{V} \cdot 167\ k\Omega \cdot 6\ GHz = 150\ GHz$$

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): $I_D = 4\ \mu A$ (V_{ovn} is still 70 mV), $r_o = 435\ k\Omega$, $g_m = 55\ \mu A/V$, and $f_T = 6\ GHz$. The GFT with the device width of 20 is

$$\overbrace{g_m r_o}^{\text{open circuit gain} \;=\; 24} \cdot f_T = 55\frac{\mu A}{V} \cdot 435\ k\Omega \cdot 6\ GHz = 144\ GHz$$

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 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, $I_D = 45\ \mu A$ (V_{ovn} is now 170 mV), $r_o = 55\ k\Omega$, $g_m = 390\ \mu A/V$, and $f_T = 10\ GHz$. The GFT with the increased V_{ovn} is

$$\overbrace{g_m r_o}^{\text{open circuit gain} \;=\; 21.5} \cdot f_T = 390\frac{\mu A}{V} \cdot 55\ k\Omega \cdot 10\ GHz = 215\ GHz$$

Note that the open circuit gain only decreased a modest 3.5 from the value in Table 9.2. Using the long-channel theory we would expect, with a roughly 4 increase in the drain current, the open circuit gain to be cut in half (Eq. [9.60]).

By using higher gate overdrives, in a short-channel process, we can improve the speed of our circuits at the cost of power and overhead (the devices will go into triode at higher V_{DS}) with small degradation to the gain (the GFT increases with increasing gate overdrive). It might appear that, because the GFT in a long-channel CMOS technology is independent of bias conditions (see Eq. [9.59]), it is better to use a short-channel process. However, we must remember that the open circuit gains of designs using the short-channel processes are considerably lower than those of designs using the long-channel technology (see Tables 9.1 and 9.2). The GFT, for the NMOS device in Table 9.1 for example, is 675 GHz (compared to 150 GHz for the one in Table 9.2). The short-channel device is better for high-speed design because its f_T is higher (it will still behave like an amplifier in the GHz frequency region) but worse for lower speed design.

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 by a resistor of $\frac{2}{3} \cdot \frac{1}{g_m}$ while in the saturation region and R_{ON} in the triode region, is given, in the saturation region, by a power spectral density, PSD, of

$$I_{D,th}^2(f) = \frac{4kT}{\frac{2}{3}\frac{1}{g_m}} = \frac{8kT}{3} \cdot g_m \qquad (9.63)$$

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).

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

$$I_{1/f}^2(f) = \frac{KF \cdot I_D^{AF}}{f \cdot (C'_{ox} \cdot L)^2} \qquad (9.64)$$

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,

$$I_{1/f}^2(f) = \frac{KF \cdot I_D^{AF}}{f \cdot (C'_{ox})^2 LW} \qquad (9.65)$$

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).

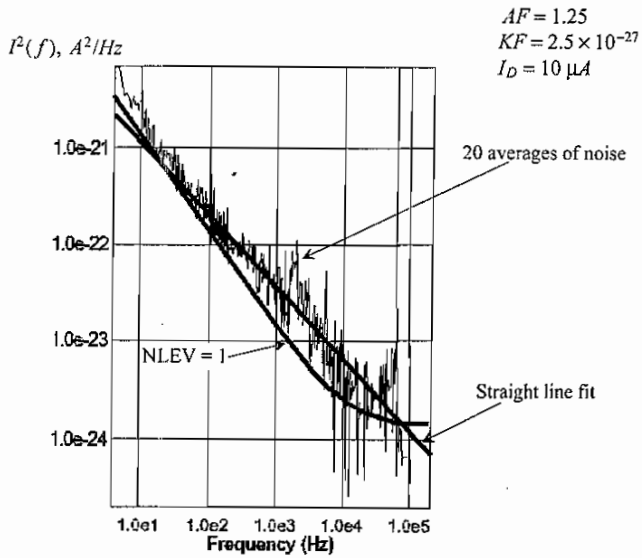


Figure 9.36 Measured MOSFET flicker noise spectrum.

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.)

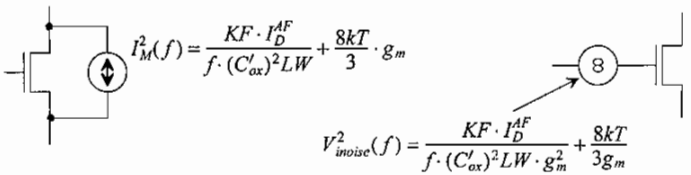


Figure 9.37 (a) Modeling MOSFET noise current and (b) input-referred noise voltage.

ADDITIONAL READING

Textbooks Covering Small-Signal Analysis

[1] R. Jaeger and T. Blalock, *Microelectronic Circuit Design*, McGraw-Hill Publishers, 2nd ed., 2003. ISBN 0-072-50503-6.

[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.

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.

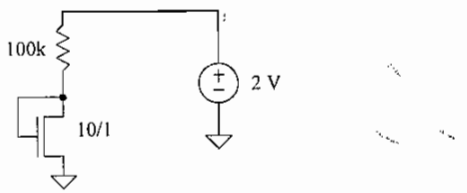


Figure 9.38 Circuit used in Problem 9.1

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.

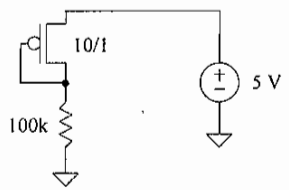


Figure 9.39 Circuit used in Problem 9.3

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.

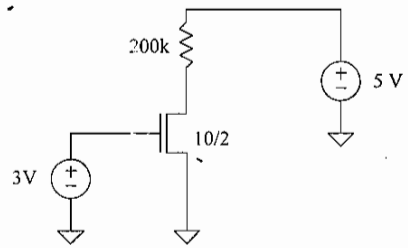


Figure 9.40 Circuit used in Problem 9.5

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?

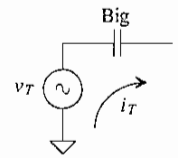


Figure 9.41 Using a test voltage to determine a resistance in a simulation.

9.7 Explain qualitatively what happens to V_{GS4} and V_{DS4} in Fig. 9.42 as the bias current is increased.

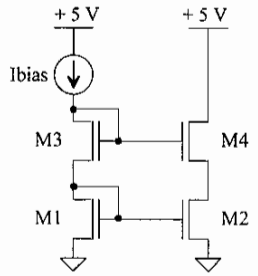


Figure 9.42 Schematic for Problem 9.7.

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.

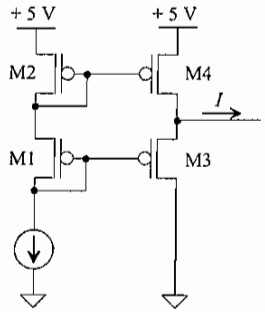


Figure 9.43 Schematic for Problem 9.8.

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.

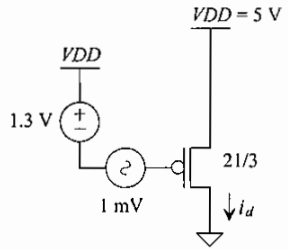


Figure 9.44 Schematic for Problem 9.13.

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.

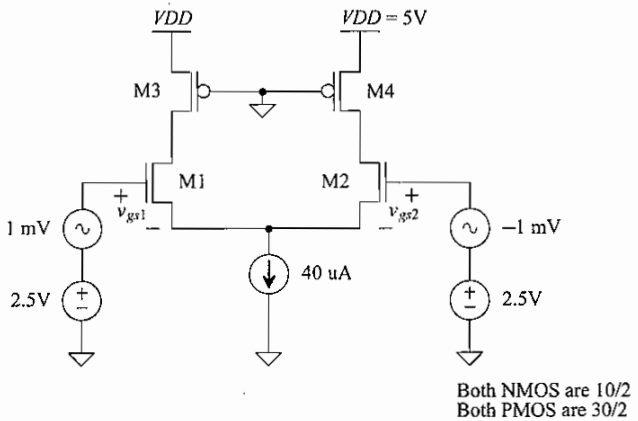


Figure 9.45 Circuit used in Problem 9.14.

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.

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.

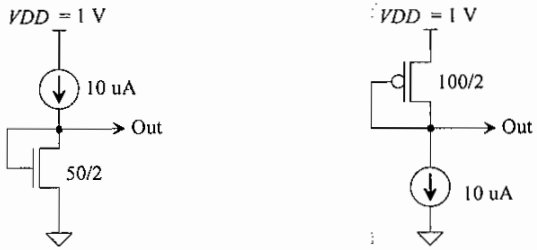


Figure 9.46 Circuits for Problem 9.23.

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 9.1, calculate the drain voltage of M1 and its drain current (both AC and DC components) for the circuit seen in Fig. 9.47.

9.27 What is the PSD of a MOSFET's thermal noise when it is operating in the triode region?

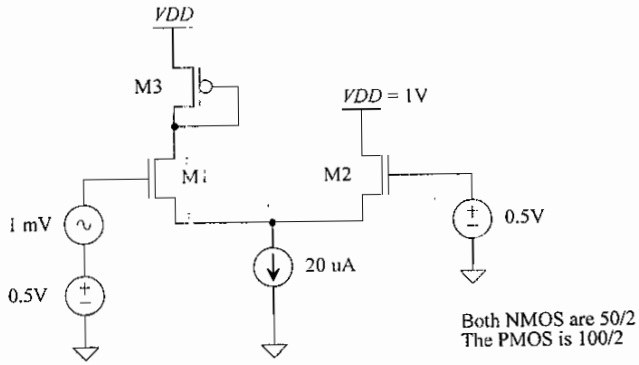


Figure 9.47 Circuit used in Problem 9.26.

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?

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 a 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

$$Q_{final} = C \cdot (VDD - 0) = C \cdot VDD \qquad (10.2)$$

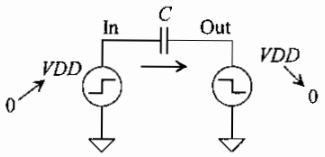


Figure 10.2 Determining the charge through a capacitor.

The total charge supplied by the input or output voltage source to the capacitor after the transition is then

$$Q_{tot} = Q_{final} - Q_{init} = 2C \cdot VDD \qquad (10.3)$$

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

$$Q_{tot} = 2C \cdot VDD \qquad (10.4)$$

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

$$I_D = \frac{KP_n}{2} \cdot \frac{W}{L} \cdot (VDD - V_{THN})^2 = \frac{\beta}{2} \cdot (VDD - V_{THN})^2 \qquad (10.5)$$

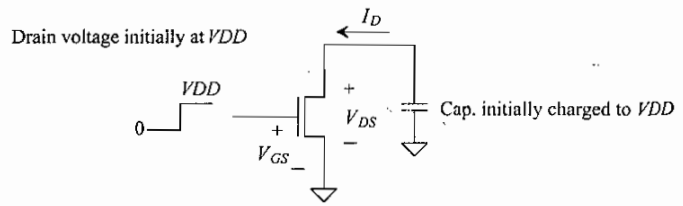


Figure 10.4 MOSFET switching circuit.

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.

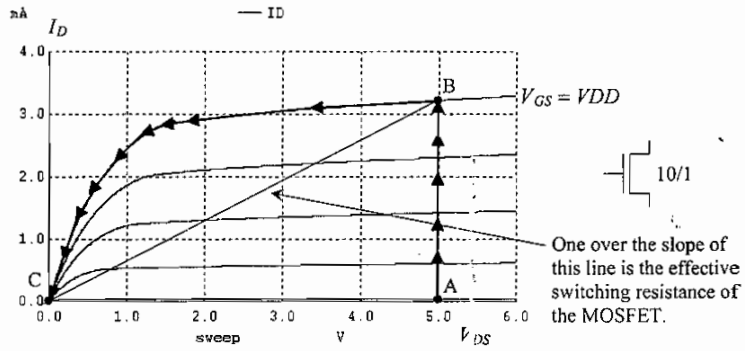


Figure 10.5 IV plot for a 10/1 NMOS device to estimate average switching resistance.

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.

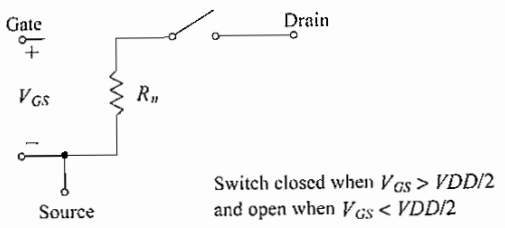


Figure 10.6 Simple digital MOSFET model.

From the data seen in Fig. 10.5 we can estimate the effective digital resistance as

$$R_n = R_n' \cdot \frac{L}{W} = \frac{VDD}{I_{D,sat}} = \frac{5}{3.3\ mA} \rightarrow R_n' \approx 15\ k\Omega \qquad (10.7)$$

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

$$R_n \approx 15k \cdot \frac{L}{W} \qquad (10.8)$$

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

$$R_p \approx 45k \cdot \frac{L}{W} \qquad (10.9)$$

The effective resistance of the PMOS device is three times as large as the NMOS's effective switching resistance 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),

$$R_n = \frac{VDD}{I_{D,n}} = \frac{VDD}{I_{on,n} \cdot W \cdot scale} = \frac{1\ V}{600\ \frac{\mu A}{\mu m} \cdot W \cdot scale} \rightarrow R_n \approx \frac{1.7k \cdot \mu m}{W \cdot scale} \qquad (10.10)$$

and

$$R_p = \frac{VDD}{I_{D,p}} = \frac{VDD}{I_{on,p} \cdot W \cdot scale} = \frac{1\ V}{300\ \frac{\mu A}{\mu m} \cdot W \cdot scale} \rightarrow R_p \approx \frac{3.4k \cdot \mu m}{W \cdot scale} \qquad (10.11)$$

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 causes the MOSFET's resistance to start to increase.

Note that in the long-channel equations, Eqs. (10.8) and (10.9), the scale factor doesn't affect the resistance. However, in the short-channel equations, the scale factor is important and does affect the switching resistance. 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)

$$R_n \approx \frac{34k}{W} \qquad (10.12)$$

and

$$R_p \approx \frac{68k}{W} \qquad (10.13)$$

where W is the drawn width of the devices.

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

$$I = C_{gd} \cdot \frac{dV_{gd}}{dt} = \frac{C_{ox}}{2} \cdot \frac{VDD-(-VDD)}{\Delta t} = C_{ox} \cdot \frac{VDD}{\Delta t} = C_{ox} \cdot \frac{dV_{DS}}{dt} \qquad (10.14)$$

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.

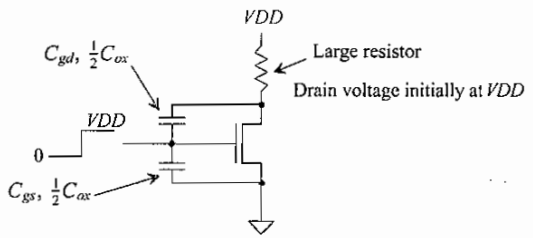


Figure 10.7 MOSFET switching circuit with capacitances.

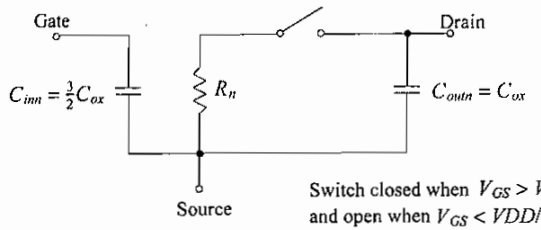


Figure 10.8 Simple digital MOSFET model.

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*, τ_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

$$\tau_p \approx 75\ ps$$

For the short-channel process, using the data in Table 5.1 and Eq. (10.10), we can write

$$\tau_n = R_nC'_{ox} \cdot WL \cdot (scale)^2 = \frac{1.7k \cdot \mu m}{W \cdot scale} \cdot 25\frac{fF}{\mu m^2} \cdot WL \cdot (scale)^2 = 2.1\ ps$$

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

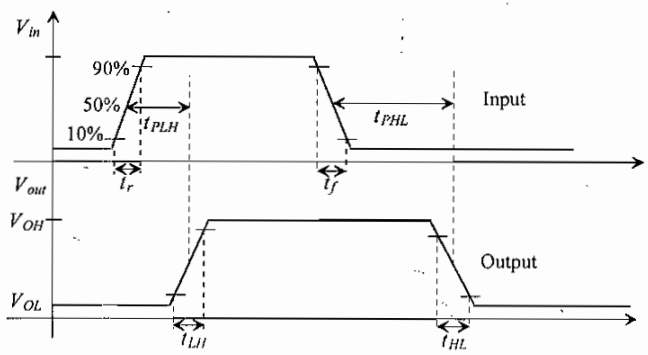


Figure 10.9 Definition of delays and transition times.

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.

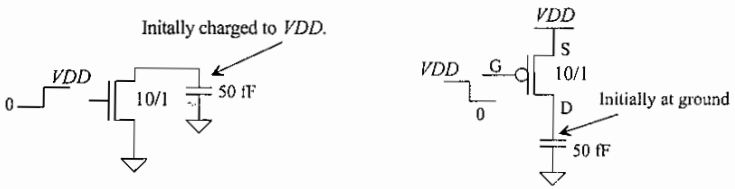


Figure 10.10 Circuits used in Example 10.2.

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

$$t_{PLH} = 0.7 \cdot R_p C_{tot} = 0.7 \cdot R_p \cdot (C_{ox} + C_L) \tag{10.22}$$

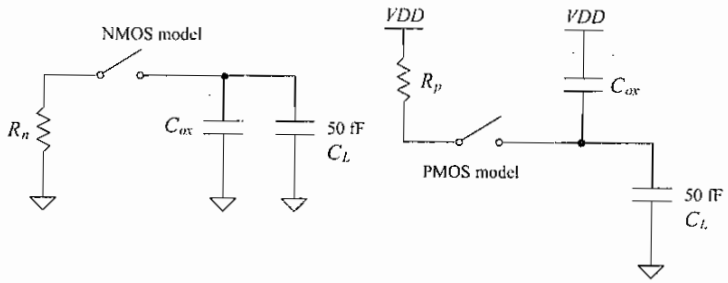


Figure 10.11 Models used to determine switching times in Example 10.2.

In many practical situations the load capacitance is much larger than the output capacitances of the switches. These equations can then be reduced to

$$t_{PHL} = 0.7 \cdot R_n \cdot C_L \text{ and } t_{PLH} = 0.7 \cdot R_p \cdot C_L \text{ for } C_L >> C_{ox} \tag{10.23}$$

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

$$t_{HL} = 377\,ps \text{ and } t_{LH} = 754\,ps \blacksquare$$

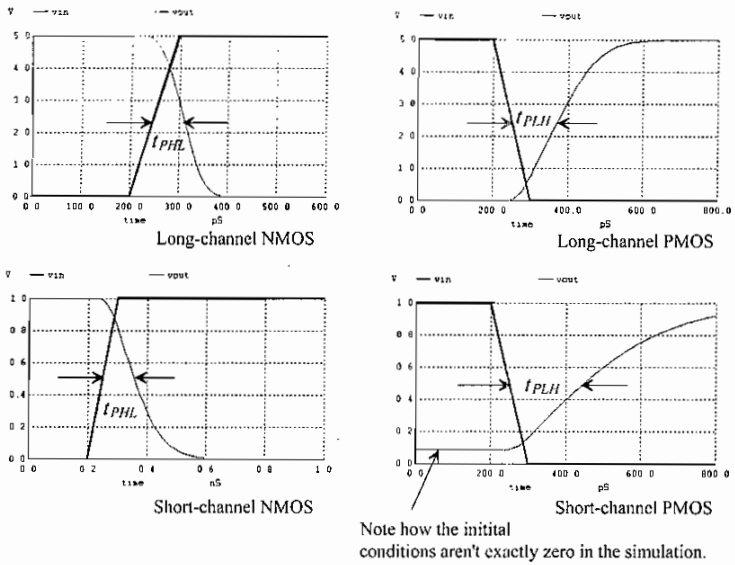


Figure 10.12 Simulating the circuits in Fig. 10.10.

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*

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.

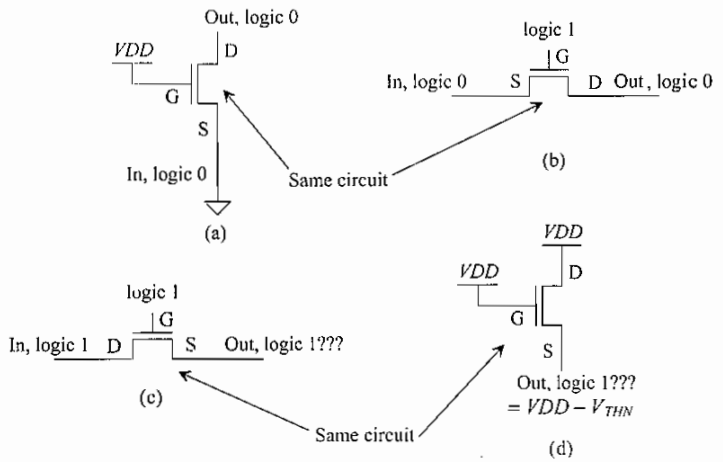


Figure 10.13 Using the NMOS switch as a pass gate.

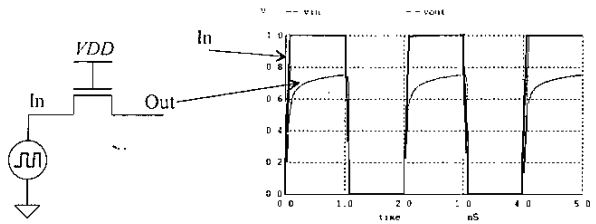


Figure 10.14 The input and output of an NMOS pass gate.

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.*

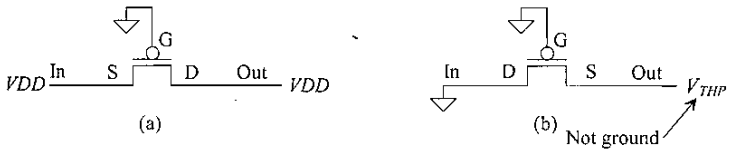


Figure 10.15 How the PMOS device does not pass a logic 0 well.

Example 10.3
Estimate the output voltages in the circuits seen in Fig. 10.16.

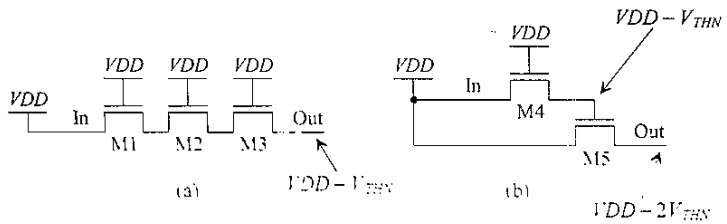


Figure 10.16 Circuits used in Ex. 10.3.

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

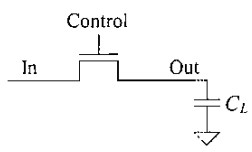
Consider the PG configuration seen in Fig. 10.17a. Let's estimate the delay between the input and the output of the PG. Note that if the input to the circuit is a "0" the PG behaves like the configuration seen in Fig. 10.4 and the output gets pulled all the way down to ground. Let's consider the configuration 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

$$C_{tot} = C_L + \frac{C_{ox}}{2} \tag{10.24}$$

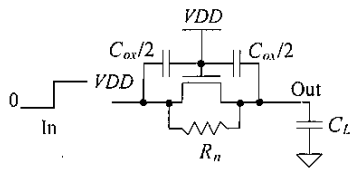
The delay through the PG can be estimated as

$$t_{delay} = 0.7 \cdot R_n C_{tot} = 0.7 \cdot R_n \cdot \left(C_L + \frac{C_{ox}}{2}\right) \tag{10.25}$$

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

$$t_{delay} \approx 0.7 \cdot R_{n,p} C_L \tag{10.26}$$

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. ■

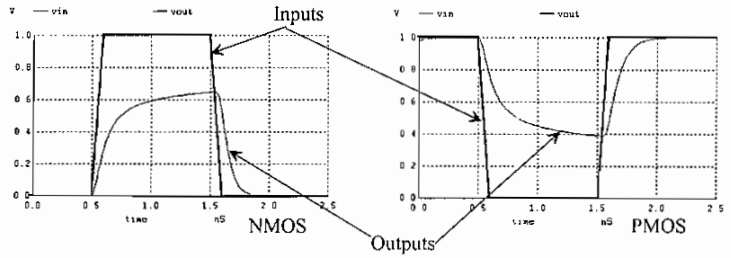


Figure 10.19 The delay through the PGs in Fig. 10.18.

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.

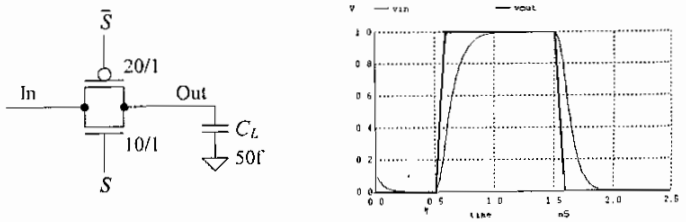


Figure 10.20 Simulating the operation of a transmission gate.

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

$$t_{delay} = 0.7 \cdot (R_n || R_p) \cdot C_L \qquad (10.27)$$

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

$$t_{delay} \approx 0.35 \cdot R_n \cdot C_{ox} \cdot l^2 \qquad (10.28)$$

Noting $R_n C_{ox}$ is the process characteristic time constant, τ_n, we can quickly estimate delays through series-connected PGs without doing much of a calculation. For example,

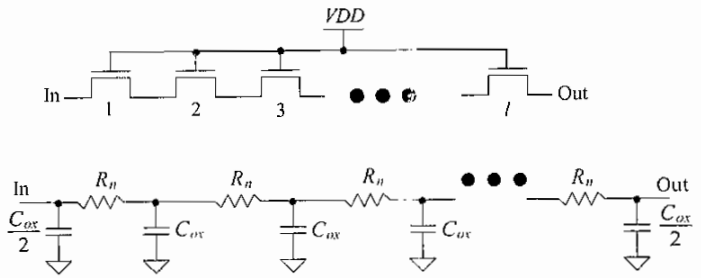


Figure 10.21 How a series connection of MOSFETs behaves like an RC transmission line (see Fig. 2.22).

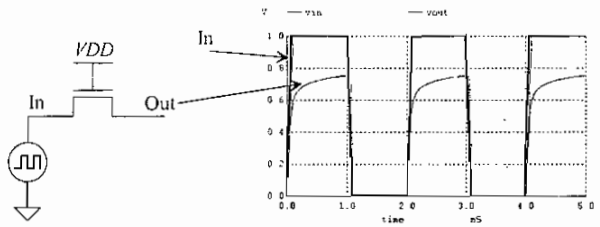


Figure 10.14 The input and output of an NMOS pass gate.

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.*

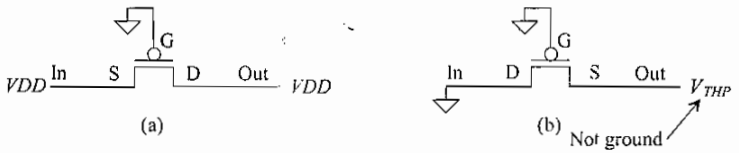


Figure 10.15 How the PMOS device does not pass a logic 0 well.

Example 10.3
Estimate the output voltages in the circuits seen in Fig. 10.16.

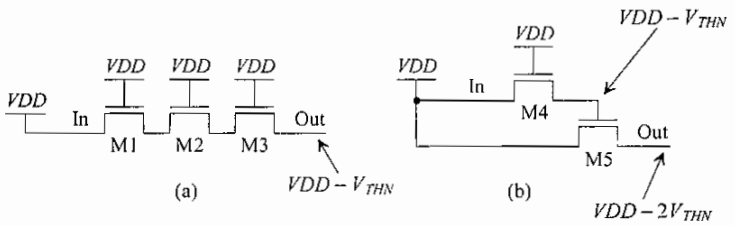


Figure 10.16 Circuits used in Ex. 10.3.

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

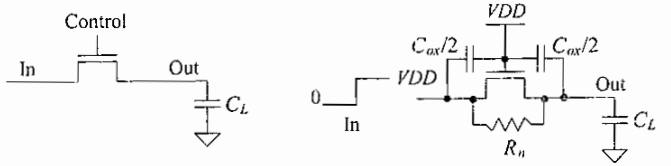
Consider the PG configuration seen in Fig. 10.17a. Let's estimate the delay between the input and the output of the PG. Note that if the input to the circuit is a "0" the PG behaves like the configuration seen in Fig. 10.4 and the output gets pulled all the way down to ground. Let's consider the configuration 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

$$t_{delay} \approx 0.7 \cdot R_{n,p} C_L \tag{10.26}$$

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.

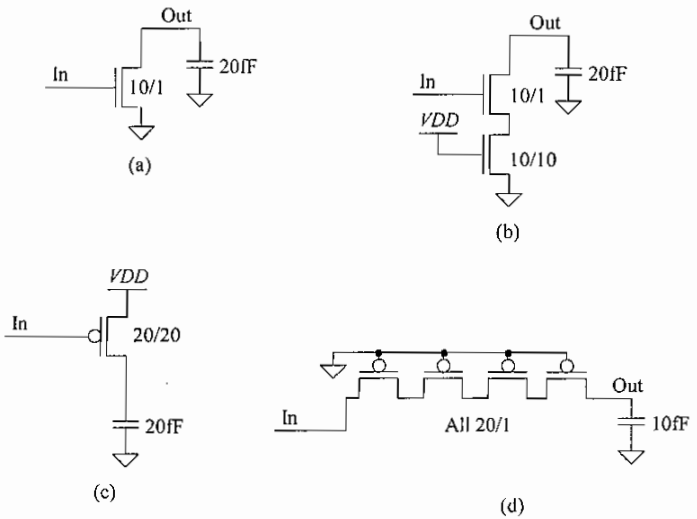


Figure 10.25 Circuits used in Problem 10.5.

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.

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 4 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.

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.

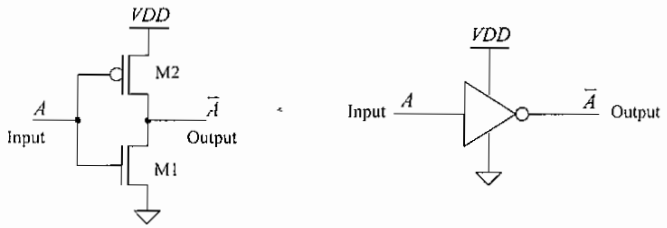


Figure 11.1 The CMOS inverter, schematic, and logic symbol.

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.

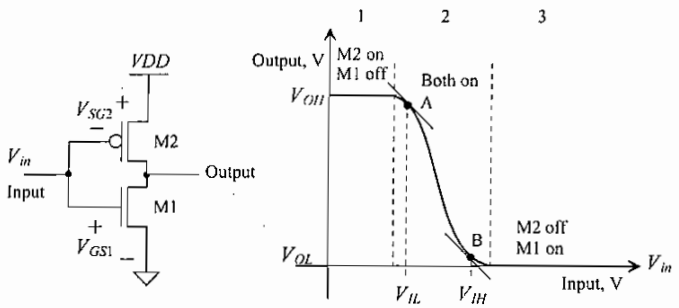


Figure 11.2 The CMOS inverter transfer characteristics.

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}.

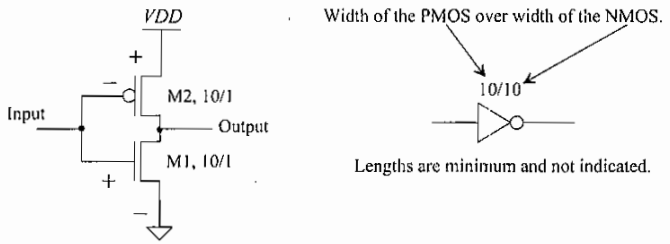


Figure 11.3 Inverter used in Ex. 11.1.

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

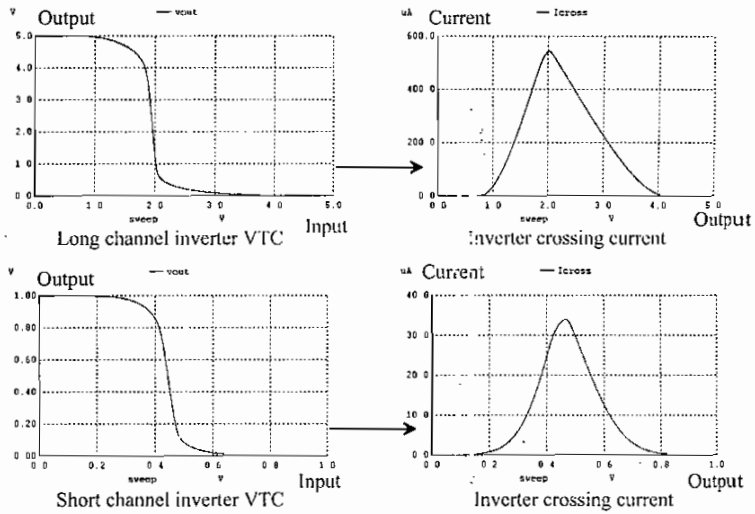


Figure 11.4 Voltage transfer curves (VTCs) for a long and a short channel inverter.

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, $NM_L = NM_H = VDD/2$.

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:

$$\frac{\beta_n}{2}(V_{SP} - V_{THN})^2 = \frac{\beta_p}{2}(VDD - V_{SP} - V_{THP})^2 \qquad (11.3)$$

Solving for V_{SP} gives

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \qquad (11.4)$$

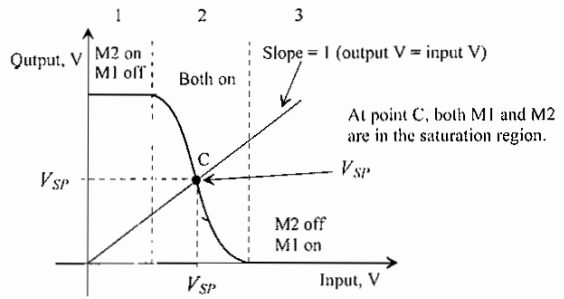


Figure 11.5 Transfer characteristics of the inverter showing the switching point.

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 β_n and β_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 β_n/β_p gives a value of approximately unity. That is,

$$\beta_n = \beta_p = KP_n\frac{W_1}{L_1} = KP_p\frac{W_2}{L_2}$$

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). ■

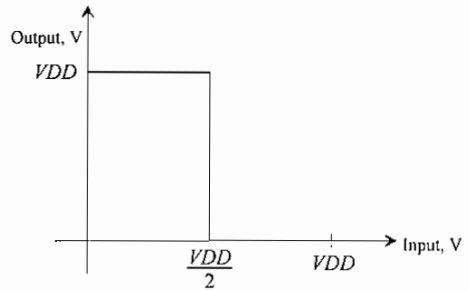


Figure 11.6 Ideal VTCs for the inverter.

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

$$V_{SP} = VDD \cdot \frac{R_n}{R_n + R_p} \qquad (11.5)$$

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

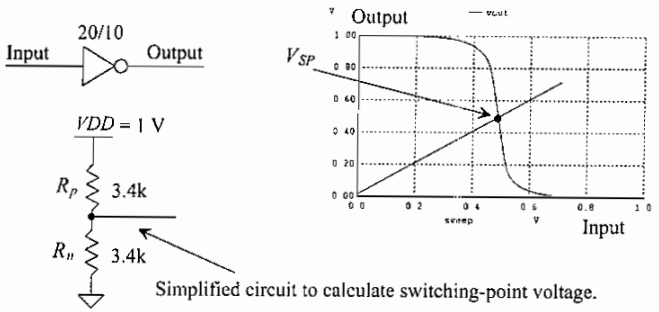


Figure 11.7 Switching point voltage for an inverter in the short-channel process.

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 β_n/β_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). ■

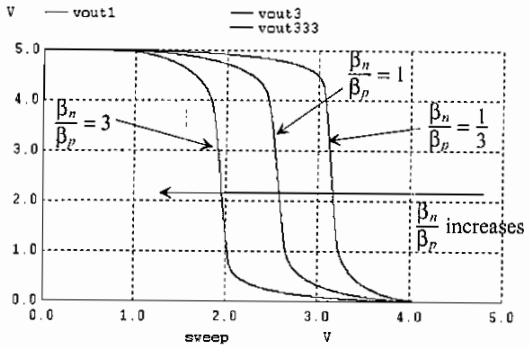


Figure 11.8 Sizing the inverter changes the switching point voltages.

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

$$t_{PLH} = 0.7 \cdot R_{p2} \cdot C_{out} \text{ and } t_{PHL} = 0.7 \cdot R_{n1} \cdot C_{out} \tag{11.8}$$

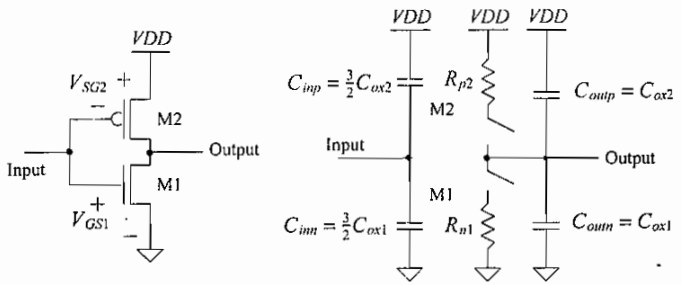


Figure 11.9 The CMOS inverter switching characteristics using the digital model.

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. ■

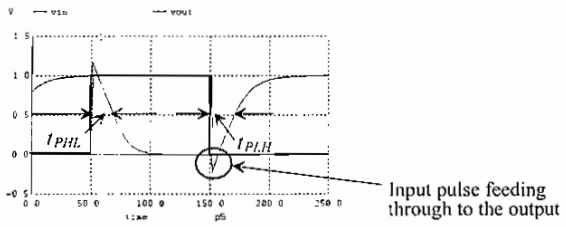


Figure 11.10 The intrinsic propagation delays of an inverter.

The propagation delays for an inverter driving a capacitive load are

$$t_{PLH} = 0.7 \cdot R_{p2} \cdot C_{tot} = 0.7 \cdot R_{p2} \cdot (C_{out} + C_{load}) \qquad (11.9)$$

and

$$t_{PHL} = 0.7 \cdot R_{n1} \cdot C_{tot} = 0.7 \cdot R_{n1} \cdot (C_{out} + C_{load}) \qquad (11.10)$$

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. ■

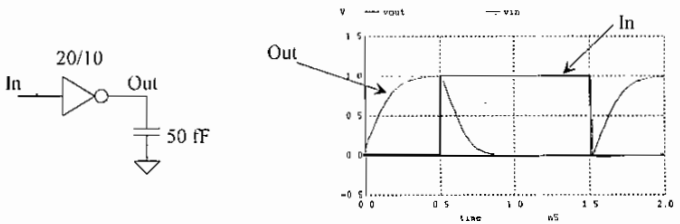


Figure 11.11 The delay associated with an inverter driving a 50 fF load.

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

$$f_{osc} = \frac{1}{n \cdot (t_{PHL} + t_{PLH})} \qquad (11.11)$$

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.

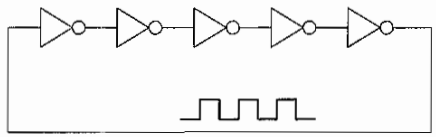


Figure 11.12 A five-stage ring oscillator.

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

$$C_{tot} = \overbrace{C_{oxp} + C_{oxn}}^{C_{out}} + \overbrace{\frac{3}{2} \cdot (C_{oxp} + C_{oxn})}^{C_{in}} = \frac{5}{2} \cdot (C_{oxp} + C_{oxn}) \qquad (11.12)$$

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

$$t_{PHL} + t_{PLH} = 0.7 \cdot (R_n + R_p) \cdot C_{tot} \qquad (11.13)$$

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

$$I_{avg} = \frac{Q_{Ctot}}{T} = \frac{VDD \cdot C_{tot}}{T} \qquad (11.14)$$

The average dynamic power dissipated by the inverter is

$$P_{avg} = VDD \cdot I_{avg} = \frac{C_{tot} \cdot VDD^2}{T} = C_{tot} \cdot VDD^2 \cdot f_{clk} \qquad (11.15)$$

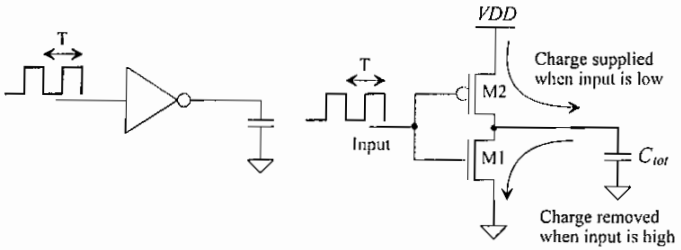


Figure 11.13 Dynamic power dissipation of the CMOS inverter.

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

$$PDP = P_{avg} \cdot (t_{PHL} + t_{PLH}) \qquad (11.16)$$

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 $431 \times 10^{-18}\,J$. ■

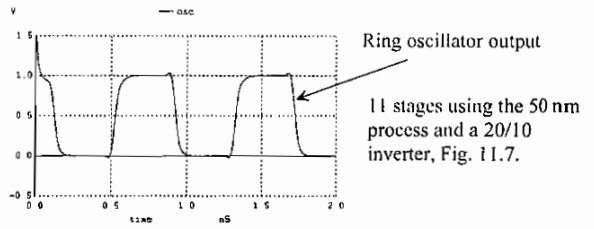


Figure 11.14 Oscillation frequency for the ring oscillator described in Ex. 11.7.

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-Ip

Figure 11.13 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.

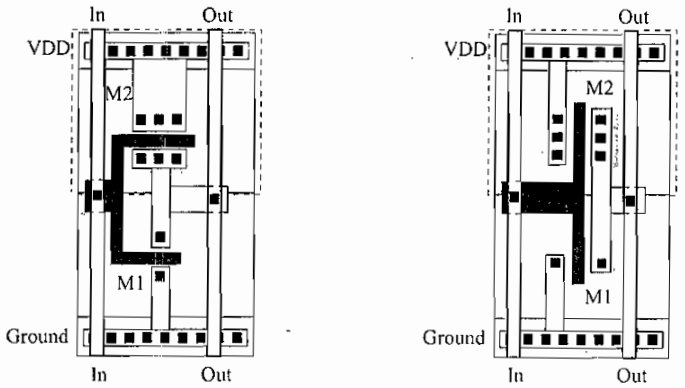


Figure 11.15 Layout styles for inverters.

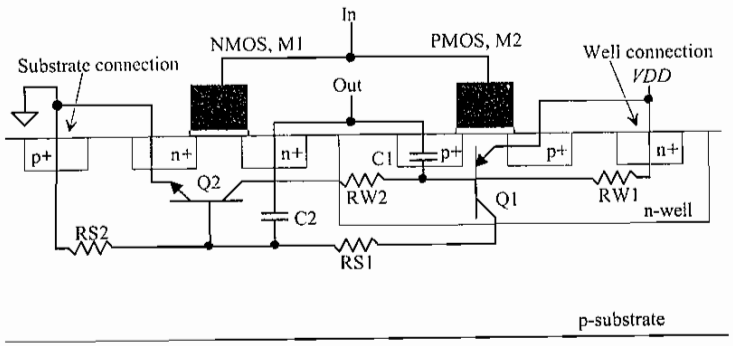


Figure 11.16 Cross-sectional view of an inverter showing parasitic bipolar transistors and resistors.

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.

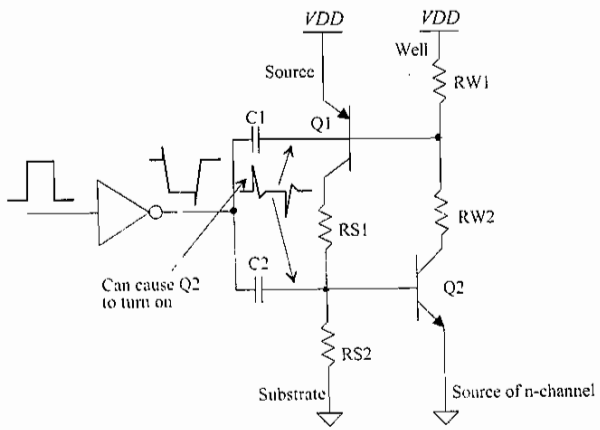


Figure 11.17 Schematic used to describe latch-up.

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.

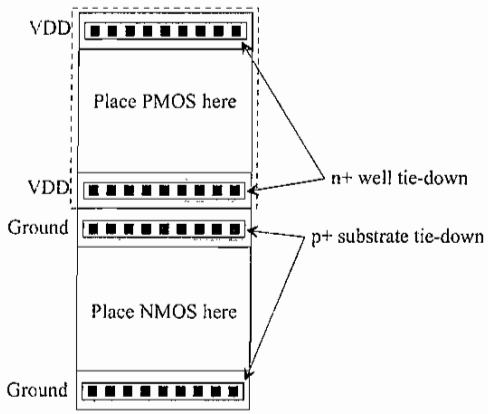


Figure 11.18 Adding an extra implant between NMOS and PMOS to reduce latch-up.

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,

$$C_{in2} = A \cdot C_{in1} \text{ and } C_{in3} = A \cdot C_{in2} = A^2 \cdot C_{in1}, \text{ etc.} \qquad (11.17)$$

The effective switching resistances are also divided by a factor of A, resulting in the same delay for each stage of the buffer,

$$R_{n,p2} = \frac{R_{n,p1}}{A} \text{ and } R_{n,p3} = \frac{R_{n,p2}}{A} = \frac{R_{n,p1}}{A^2} \qquad (11.18)$$

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.

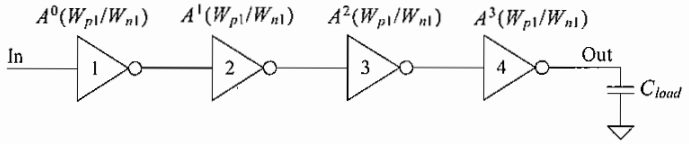


Figure 11.19 Cascade of inverters used to drive a large load capacitance.

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

$$\text{Input C of the final inverter} = C_{in1} \cdot A^N = C_{load} \qquad (11.19)$$

or

$$A = \left\lceil \frac{C_{load}}{C_{in1}} \right\rceil^{\frac{1}{N}} \qquad (11.20)$$

Again, the delays of each stage in the buffer are equal. The total delay of the inverter string is given by

$$(t_{PHL} + t_{PLH})_{total} = 0.7 \cdot \underbrace{(R_{n1} + R_{p1})(C_{out1} + AC_{in1})}_{\text{First-stage delay}} + \underbrace{0.7 \cdot \frac{(R_{n1} + R_{p1})}{A} \cdot (AC_{out1} + A^2 C_{in1})}_{\text{Second-stage delay}} \ldots \qquad (11.21)$$

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

$$(t_{PHL} + t_{PLH})_{total} = 0.7 \cdot \sum_{k=1}^{N} (R_{n1} + R_{p1})(C_{out1} + AC_{in1}) = 0\ 7 \cdot N(R_{n1} + R_{p1})(C_{out1} + AC_{in1}) \qquad (11.22)$$

or with the help of Eq. (11.20):

$$(t_{PHL} + t_{PLH})_{total} = 0.7 \cdot N(R_{n1} + R_{p1}) \cdot \left(C_{out1} + \left(\frac{C_{load}}{C_{in1}} \right)^{\frac{1}{N}} \cdot C_{in1} \right) \qquad (11.23)$$

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

$$0.7 \cdot \left((R_{n1} + R_{p1})C_{out1} + (R_{n1} + R_{p1})C_{in1} \left(\left(\frac{C_{load}}{C_{in1}} \right)^{\frac{1}{N}} + N \cdot \left(\frac{C_{load}}{C_{in1}} \right)^{\frac{1}{N}} \frac{\ln (C_{load}/C_{in1})}{-N^2} \right) \right) = 0 \qquad (11.24)$$

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

$$N = \ln \frac{C_{load}}{C_{in1}} \qquad (11.25)$$

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

$$N = \ln \left(\frac{20\ pF}{2.81\ fF} \right) = 8.87 \rightarrow 9 \text{ stages}$$

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. ■

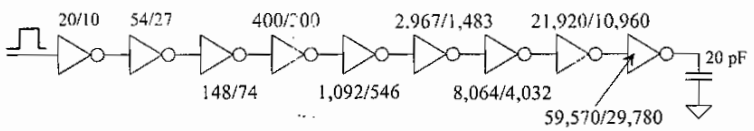


Figure 11.20 Buffer designed in Ex. 11.8. Attains the minimum delay but the buffer is not practical.

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

$$N = \frac{1}{2} \cdot \ln \frac{C_{load}}{C_{in1}} = 4.43$$

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. ■

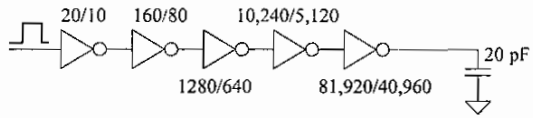


Figure 11.21 Buffer designed in Ex. 11.9.

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

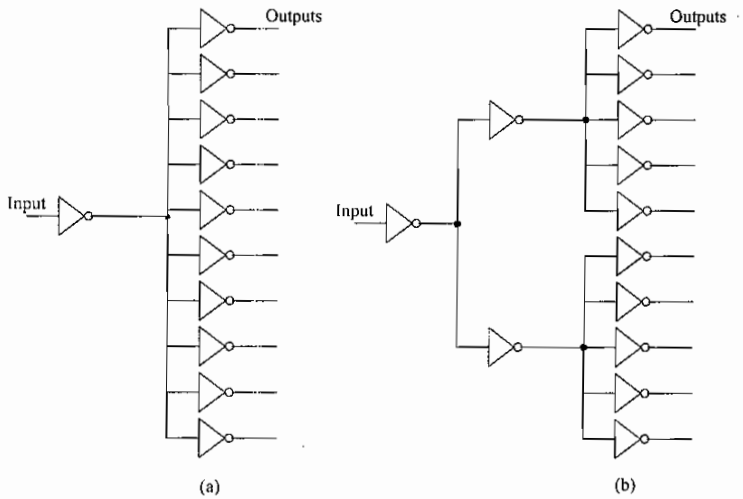


Figure 11.22 Distributed drivers.

$$t_{PHL} + t_{PLH} = 0.7 \cdot (R_n + R_p)[(C_{out} + 2C_{in}) + (C_{out} + 5C_{in})] = 0.7 \cdot (R_n + R_p)[2C_{out} + 7C_{in}] \quad (11.27)$$

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

$$t_{PHL} + t_{PLH} = 2 \cdot 0.7 \cdot [(R_n + R_p)(C_{out} + c \cdot l + C_{load}) + (r \cdot l)(C_{load})] + 2 \cdot 0.35 \cdot rcl^2 \quad (11.28)$$

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 through the RC line, while the last term is an estimate of the delay associated with driving a capacitive load through the line's resistance. The most common method of reducing the delay

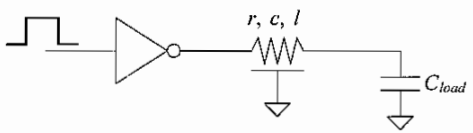


Figure 11.23 Driving an RC transmission line.

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.

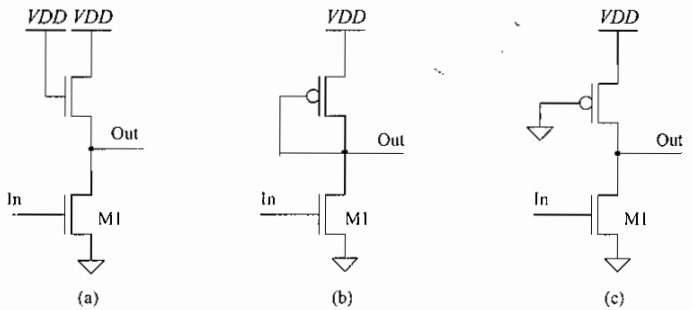


Figure 11.24 Other inverter configurations.

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*.

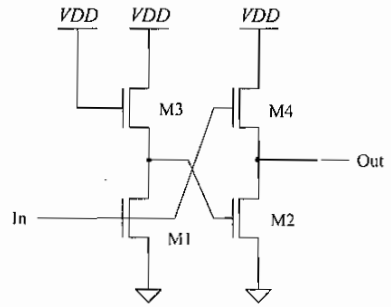


Figure 11.25 NMOS super buffer.

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.

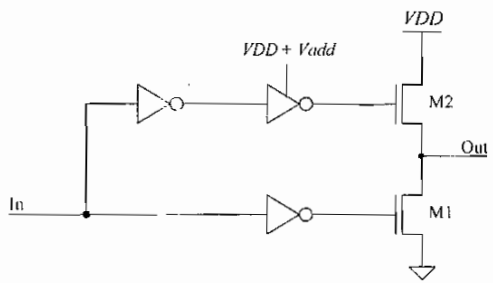


Figure 11.26 Output buffer using a pumped voltage.

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).

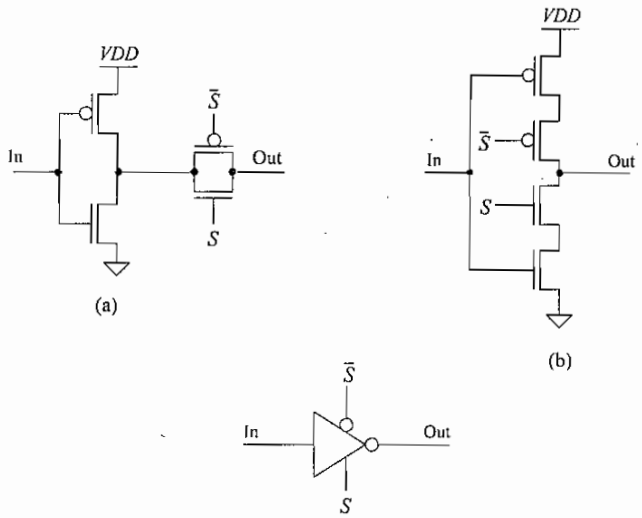


Figure 11.27 Circuits and logic symbol for the tri-state inverter.

Additional Examples

Additional delay calculations using inverters (and other CMOS circuit building blocks) can be found in Sec. 13.4.

ADDITIONAL READING

[1] J. P. Uyemura, *Introduction to VLSI Circuits and Systems*, John Wiley and Sons Publishers, 2002. ISBN 0-471-12704-3.

[2] N. H. E. Weste and D. Harris, *Principles of CMOS VLSI Design*, Addison-Wesley, 3rd ed., 2004. ISBN 0-321-14901-7.

[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}.

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.

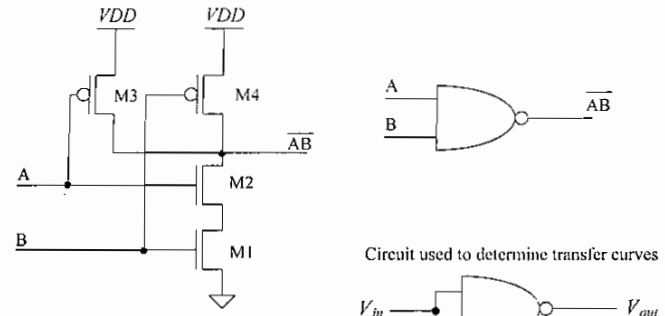
To determine the gate switching point voltage, V_{SP}, we must remember that two MOSFETs in parallel behave like a single MOSFET with a width equal to the sum of the individual widths. For the two parallel PMOS devices in Fig. 12.1a, we can write

$$W_3 + W_4 = 2W_p \tag{12.1}$$

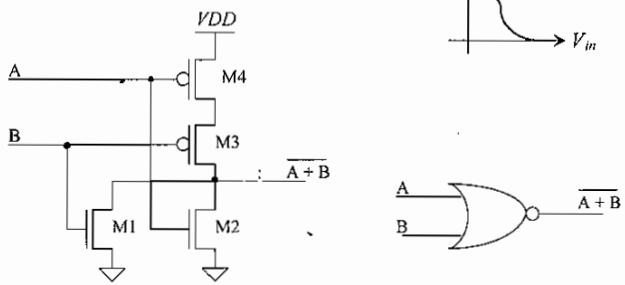
again assuming that all PMOS devices are of the same size. The transconductance parameters can also be combined into the transconductance parameter of a single MOSFET, or

$$\beta_3 + \beta_4 = 2\beta_p \tag{12.2}$$

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



(a) Schematic of a NAND gate



(b) Schematic of a NOR gate

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

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{4\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{\beta_n}{4\beta_p}}} \tag{12.6}$$

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 $VDD/2$.

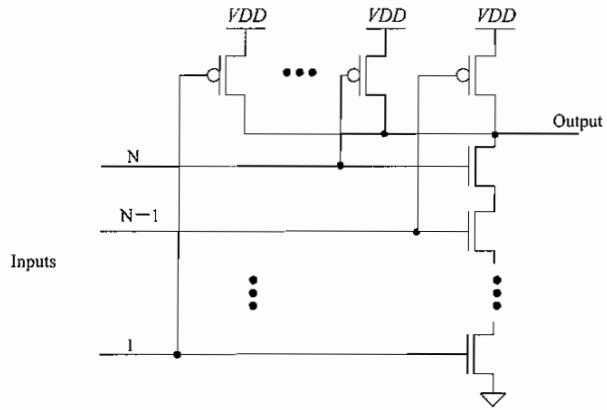


Figure 12.2 Schematic of an n-input NAND gate.

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.

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. ■

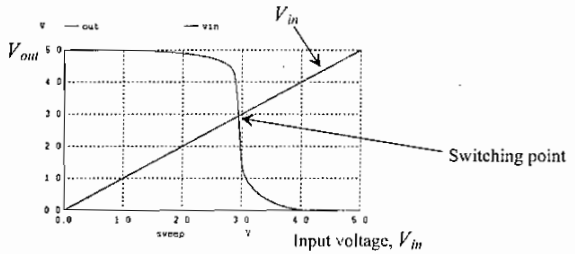


Figure 12.3 VTCs of the three-input minimum-size (using 10/1 MOSFETs) NAND gate.

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

$$V_{SP} = \frac{\sqrt{\frac{N^2\beta_n}{\beta_p}} \cdot V_{THN} + (VDD - V_{THP})}{1 + \sqrt{\frac{N^2\beta_n}{\beta_p}}} \qquad (12.8)$$

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

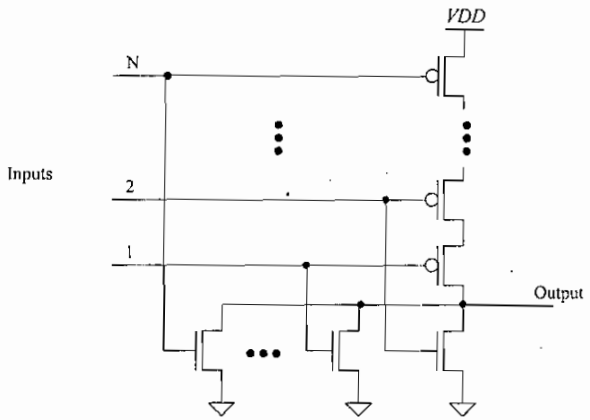


Figure 12.4 Schematic of an n-input NOR gate.

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. ■

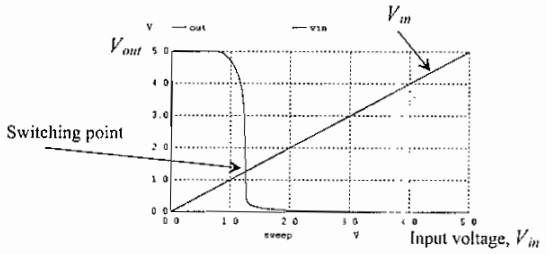


Figure 12.5 VTCs of the three-input minimum-size (using 10/1 MOSFETs) NOR gate.

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 $V_{SP} \cong (VDD - V_{THN})/2$.

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).

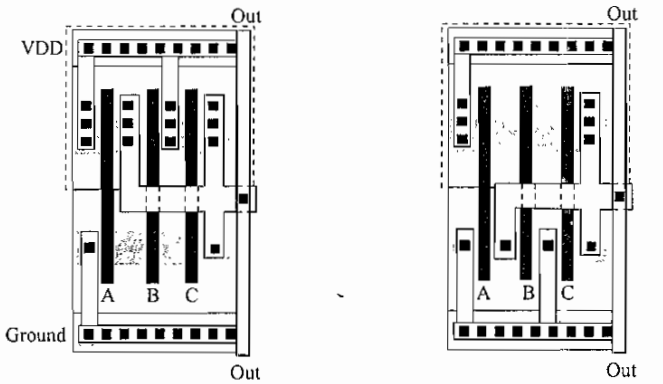


Figure 12.6 Layouts of NAND (a) and NOR (b) gates.

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

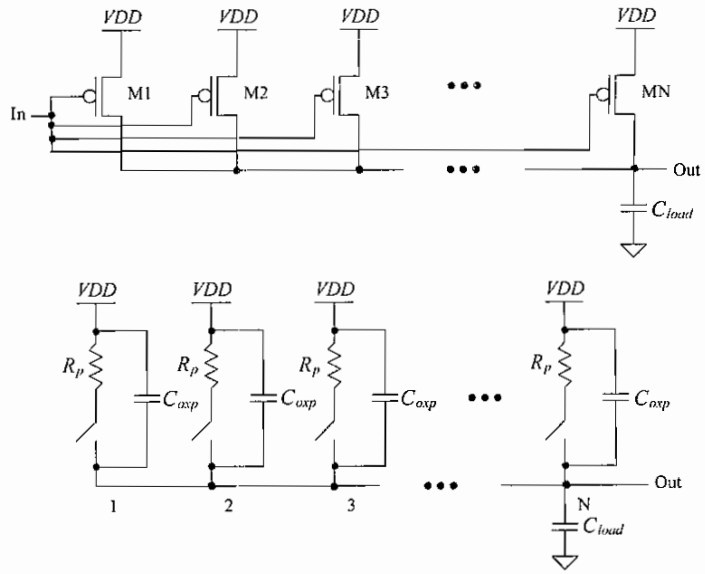


Figure 12.7 Parallel connection of MOSFETs and equivalent digital model.

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N} \cdot (N \cdot C_{oxp} + C_{load}) \qquad (12.10)$$

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

$$t_{PHL} = 0.7 \cdot \frac{R_n}{N} \cdot (N \cdot C_{oxn} + C_{load}) \qquad (12.11)$$

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

$$t_{PHL} = 0.35 \cdot R_n C_{oxn} \cdot N^2 \qquad (12.12)$$

as discussed back in Sec. 10.2.2. With an external load capacitance, the high-to-low delay-time becomes

$$t_{PHL} = 0.35 \cdot R_n C_{oxn} \cdot N^2 + 0.7 \cdot N \cdot R_n \cdot C_{load} \qquad (12.13)$$

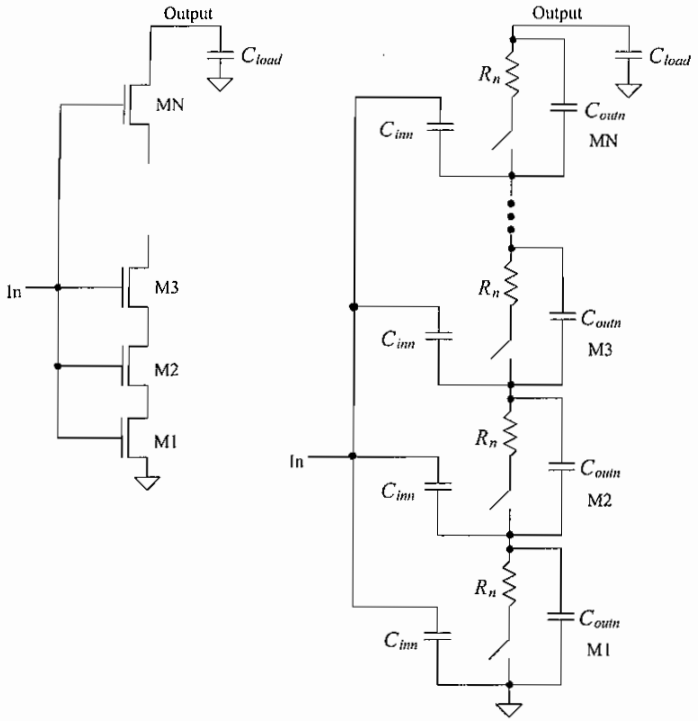


Figure 12.8 Series connection of MOSFETs and equivalent digital model.

For PMOS devices in series, a similar analysis yields

$$t_{PLH} = 0.35 \cdot R_p C_{oxp} \cdot N^2 + 0.7 \cdot N \cdot R_p \cdot C_{load} \qquad (12.14)$$

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

$$t_{PLH} = 0.7 \cdot \frac{R_p}{N}\left(N \cdot C_{outp} + \frac{C_{outn}}{N} + C_{load}\right) \qquad (12.15)$$

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

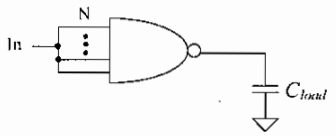


Figure 12.9 An n-input NAND gate driving a load capacitance.

capacitance is much greater than the output capacitance of the NAND gate, the low-to-high propagation time can be estimated by

$$t_{PLH} \approx 0.7 \cdot \frac{R_p}{N} \cdot C_{load} \qquad (12.16)$$

The high-to-low propagation time, using Eq. (12.13), is given by

$$t_{PHL} = 0.7 \cdot N \cdot R_n\left[N \cdot C_{outp} + \frac{C_{outn}}{N} + C_{load}\right] + 0.35 \cdot R_n C_{oxn} \cdot N^2 \qquad (12.17)$$

If C_{load} is much larger than the output capacitance of the NAND gate, then

$$t_{PHL} \approx 0.7 \cdot N \cdot R_n \cdot C_{load} \qquad (12.18)$$

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. ■

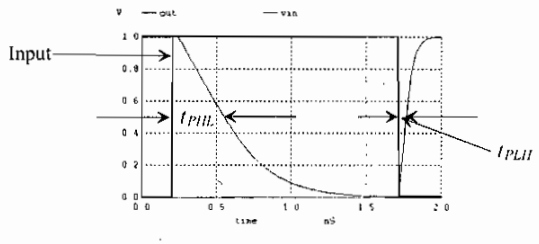


Figure 12.10 Simulating the operation of a 3-input NAND gate in 50 nm CMOS driving a 50 fF load capacitance.

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,

$$t_{PHL} \approx 0.7 \cdot N \cdot R_n \cdot C_{load} \qquad (12.19)$$

The case when the output of the NAND gate changes from a low to a high is somewhat different then 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

$$t_{PLH} \approx 0.7 \cdot R_p \cdot C_{load} \qquad (12.20)$$

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.)

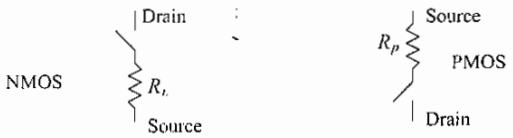


Figure 12.11 Further simplification of digital models not showing input capacitance.

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. ■

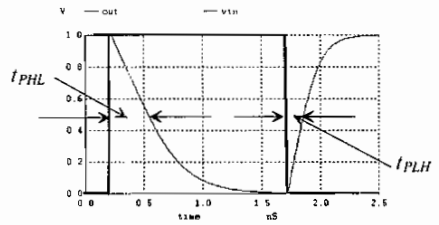


Figure 12.12 Switching delays in a 3-input NAND gate with only one changing states and driving a 50 fF load capacitance.

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.

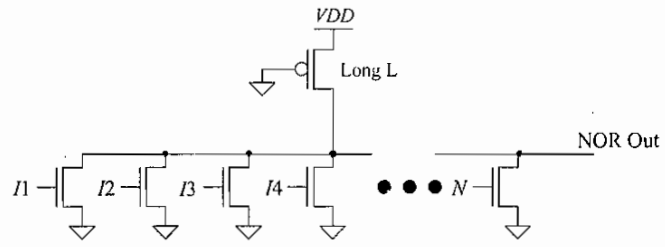


Figure 12.13 NOR configuration used for a large number of inputs.

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.

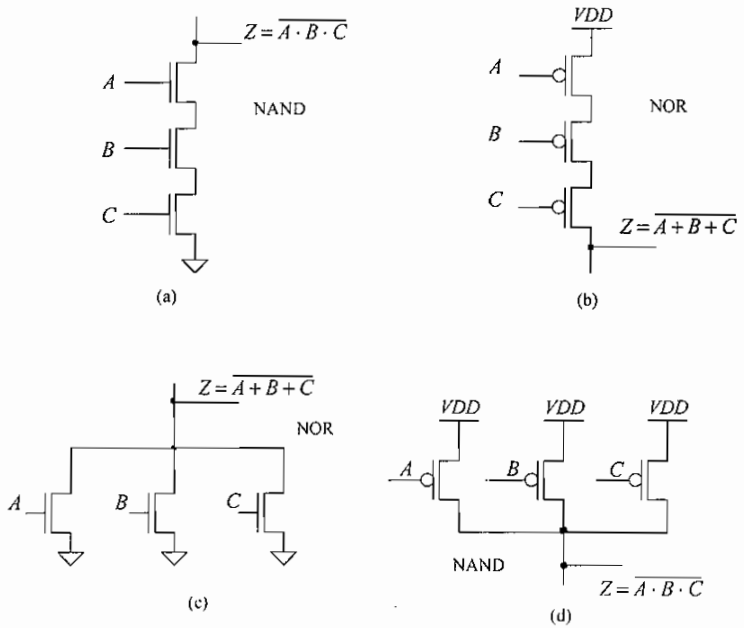


Figure 12.14 Logic implementation in CMOS.

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

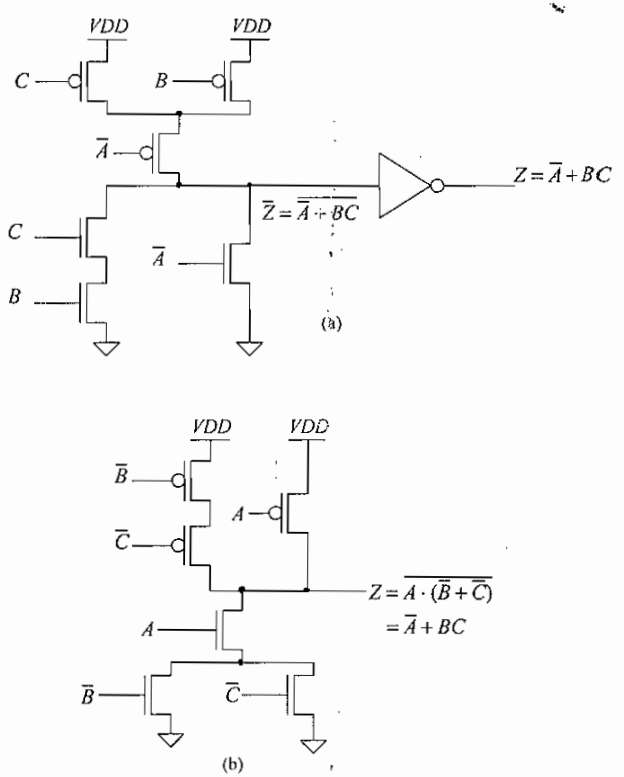


Figure 12.15 First logic gate of Ex. 12.5.

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

$$Z=A+\overline{B}C+CD=A+C(\overline{B}+D) \Rightarrow \overline{Z}=\overline{A+C(\overline{B}+D)}=\overline{A}\cdot(\overline{C}+B\overline{D})$$

or

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

The logic implementation is given in Fig. 12.16. ■

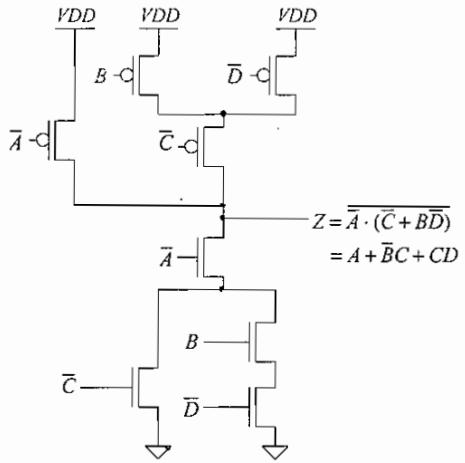
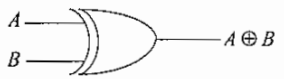


Figure 12.16 Second logic gate of Ex. 12.5.

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.

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

$$Z = \overline{\overline{A} \cdot \overline{B} + A \cdot B} = A \oplus B \qquad (12.23)$$

The CMOS AOI implementation of an XOR gate is shown in Fig. 12.18. ■

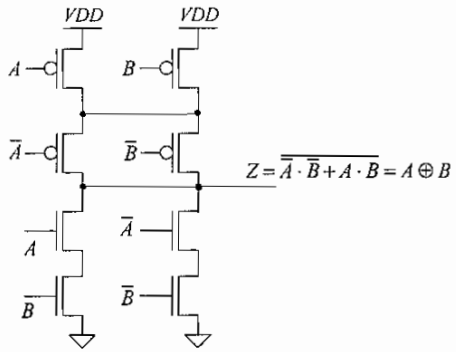
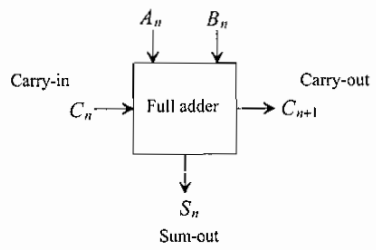


Figure 12.18 CMOS AOI XOR gate.

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.

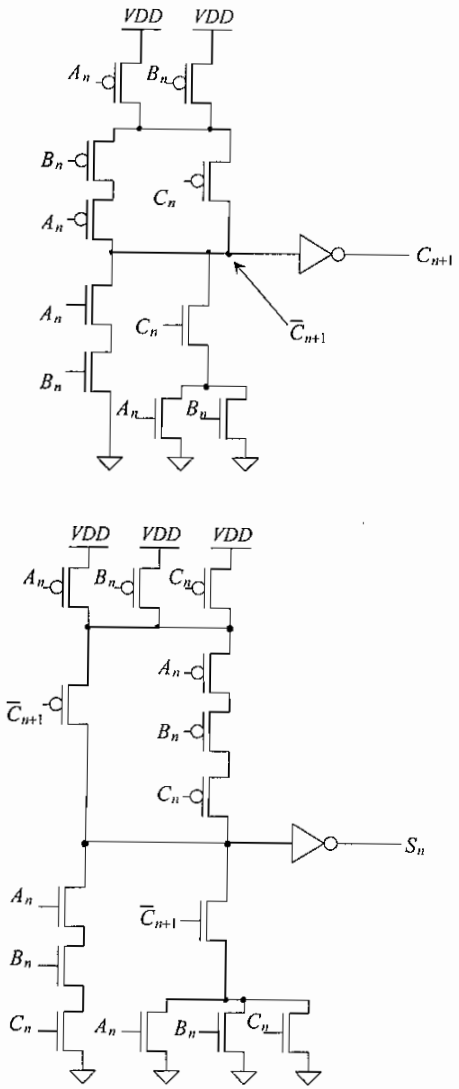


Figure 12.20 AOI implementation of a full adder.

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.

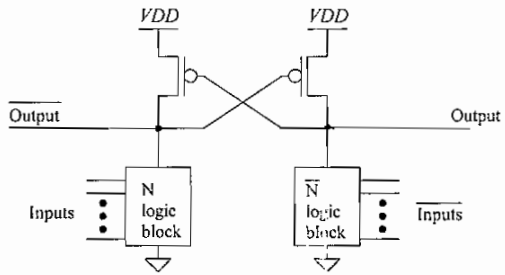


Figure 12.21 CVSL block diagram.

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.

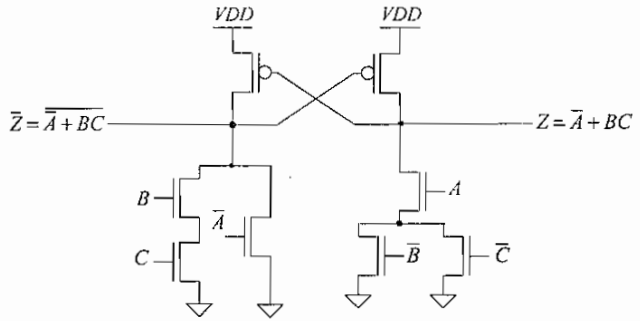


Figure 12.22 CVSL logic gate.

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.

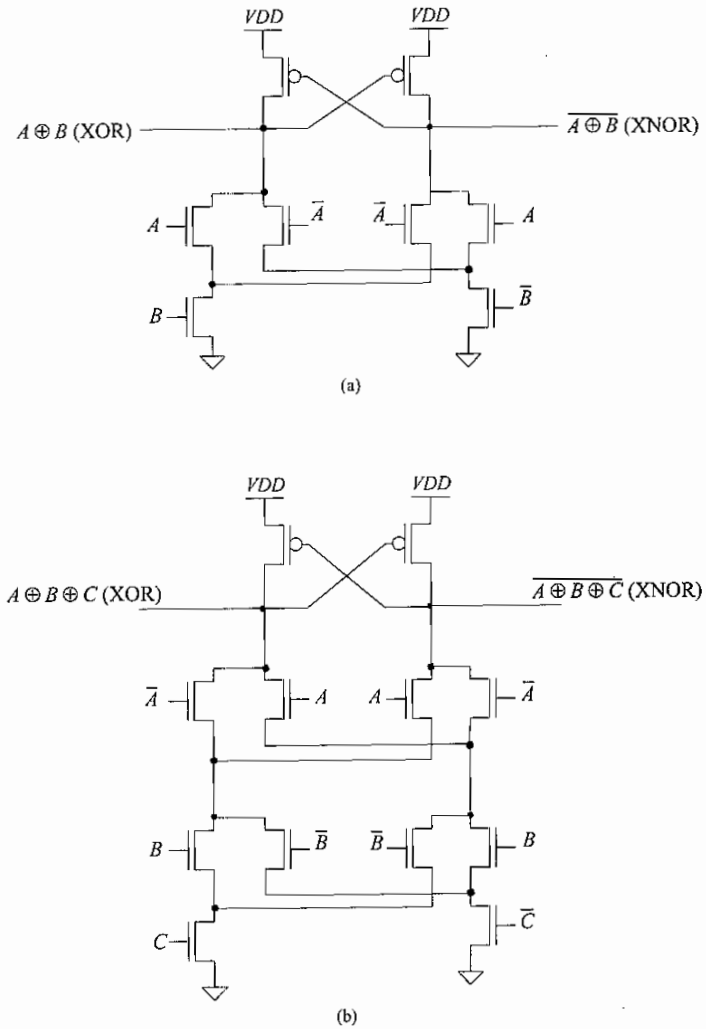


Figure 12.23 (a) Two-input and (b) three-input XOR/XNOR gates.

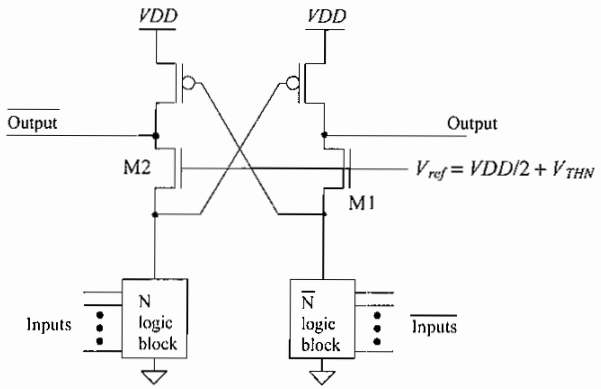


Figure 12.24 DSL block diagram.

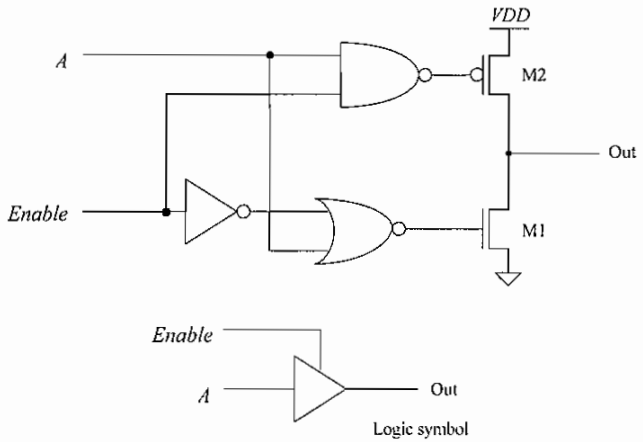


Figure 12.25 Tri-state buffer.

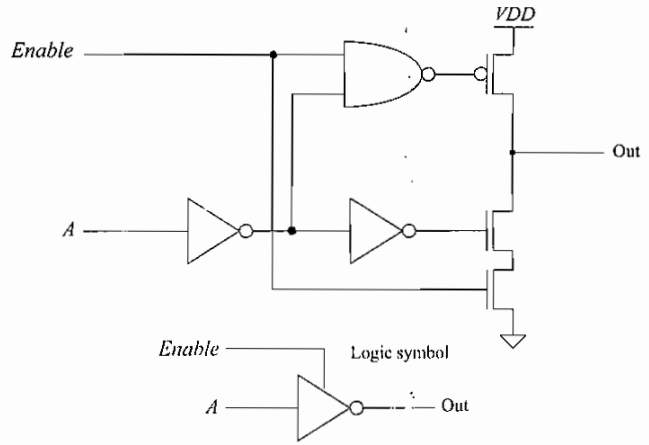


Figure 12.26 Tri-state inverting buffer.

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

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}.)

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.

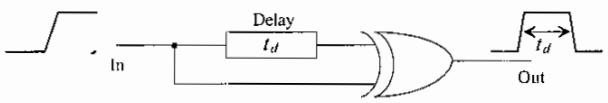


Figure 12.27 An edge detector circuit.

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 $\overline{S}$ input of the TG is the input capacitance of the PMOS device, or C_{inp}.

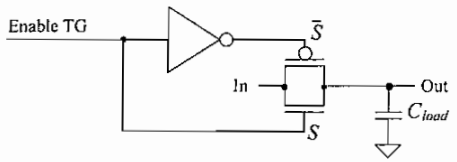


Figure 13.2 The transmission gate with control signals shown.

Evaluating 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.

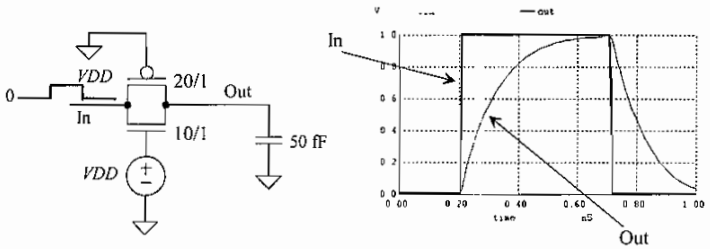


Figure 13.3 TG circuit discussed in Ex. 13.1.

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. ■

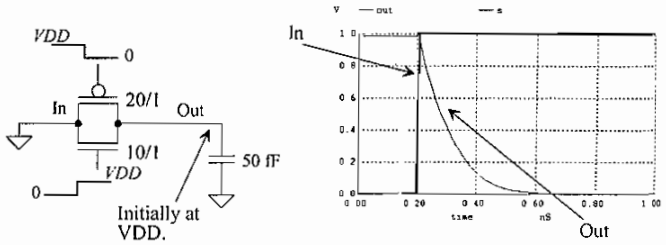


Figure 13.4 TG circuit discussed in Ex. 13.2.

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.

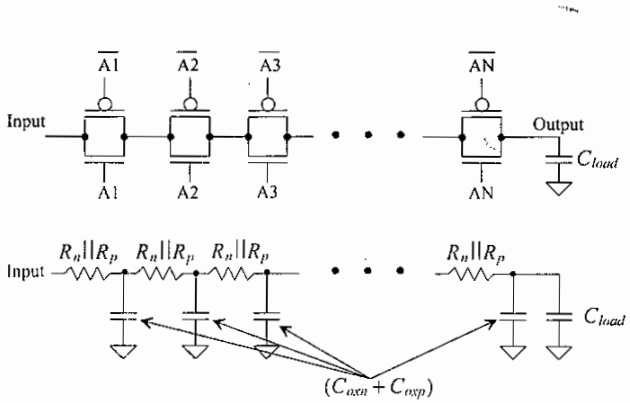


Figure 13.5 Series connection of TGs with digital model.

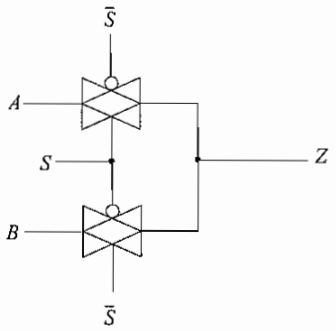


Figure 13.6 Path selector.

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

$$Z = AS + B\overline{S} \tag{13.3}$$

When the selector signal S is high, A is passed to the output while a low on S passes B to the output.

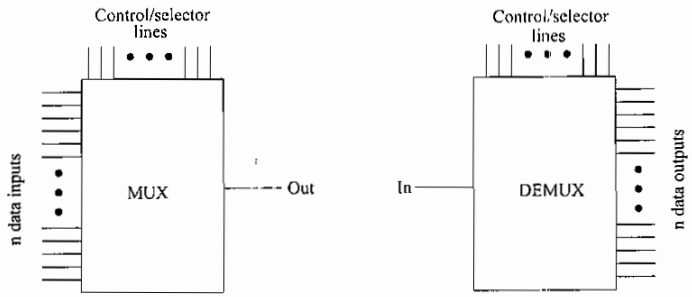


Figure 13.7 Block diagram of MUX/DEMUX.

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

$$2^m = n \tag{13.4}$$

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.

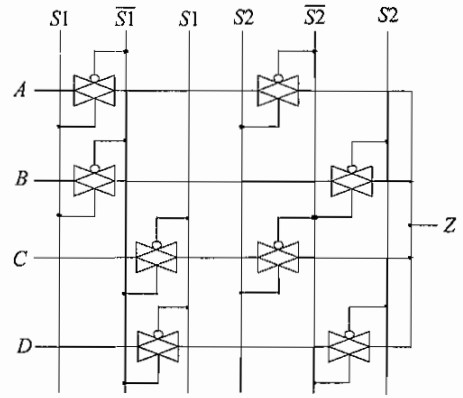
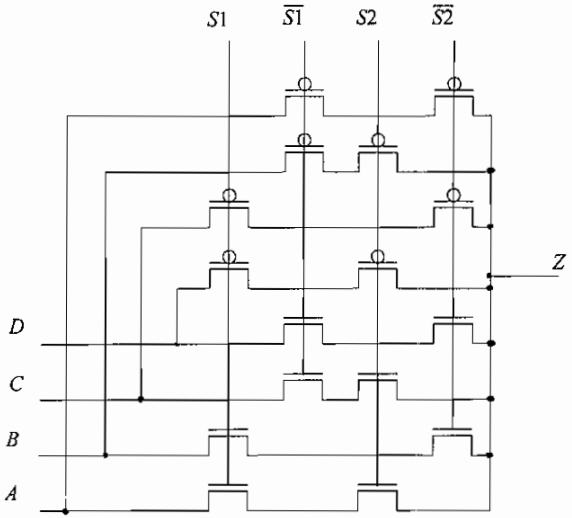


Figure 13.8 Circuit implementation of a 4-to-1 MUX.

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.



Figuer 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.

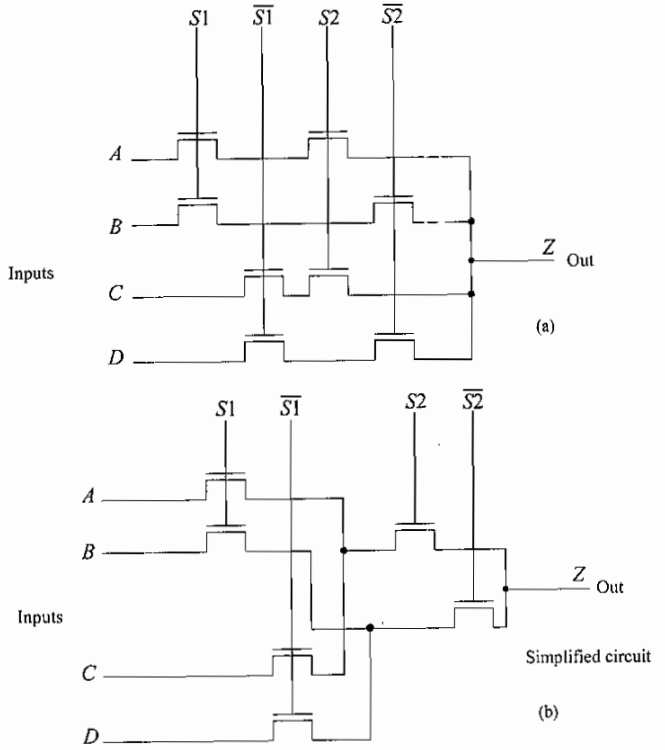


Figure 13.10 MUX/DEMUX using pass transistors.

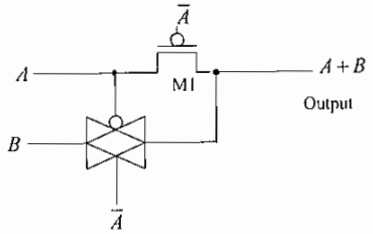


Figure 13.11 TG-based OR gate.

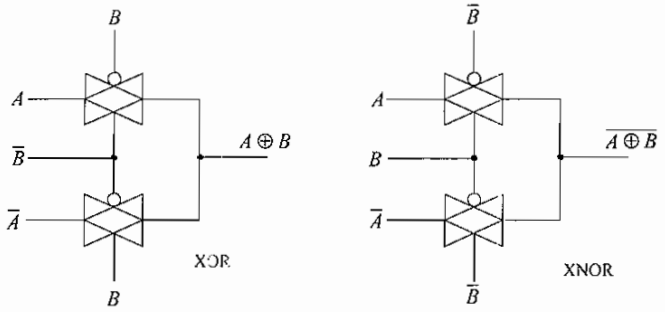
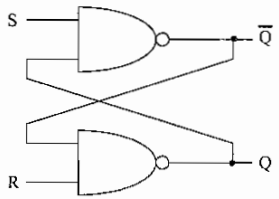


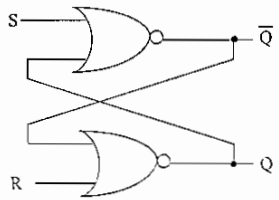
Figure 13.12 TG implementation of XOR/XNOR gate.



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.

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.*

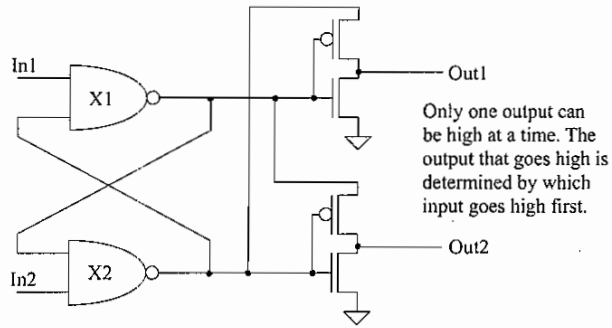


Figure 13.15 An arbiter made using NAND gates.

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

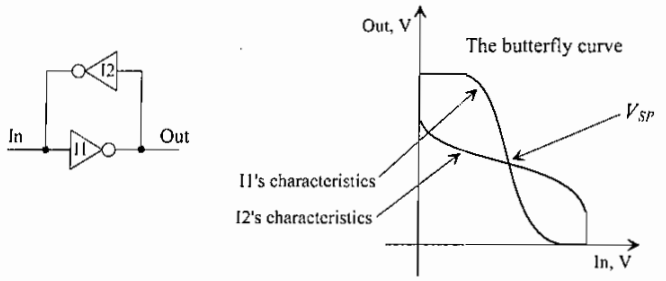


Figure 13.16 A cross-coupled inverter (a latch) and the input output characteristics.

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.

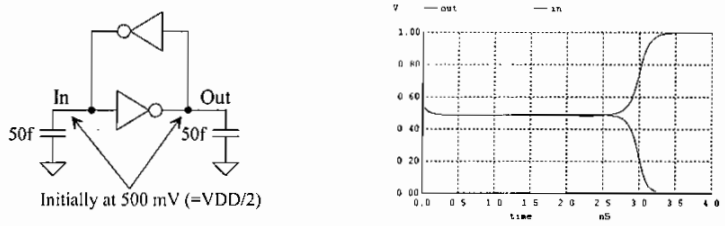


Figure 13.17 The delay associated with metastability.

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.

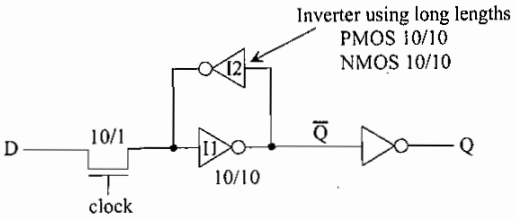


Figure 13.18 A level-sensitive latch.

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.

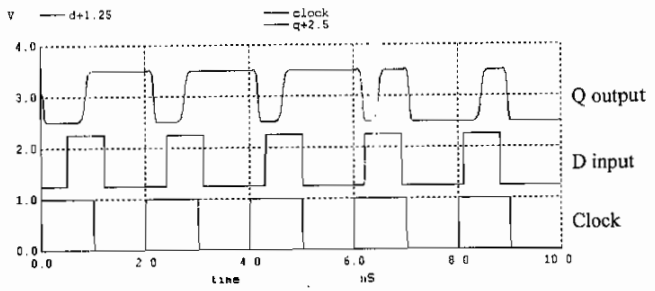


Figure 13.19 Simulating the level-sensitive latch in Fig. 13.18.

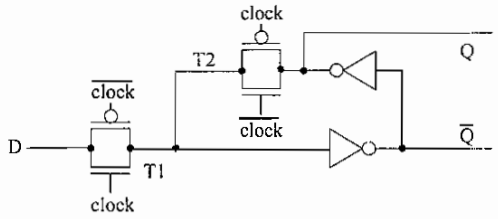


Figure 13.20 A higher performance level-sensitive latch.

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.

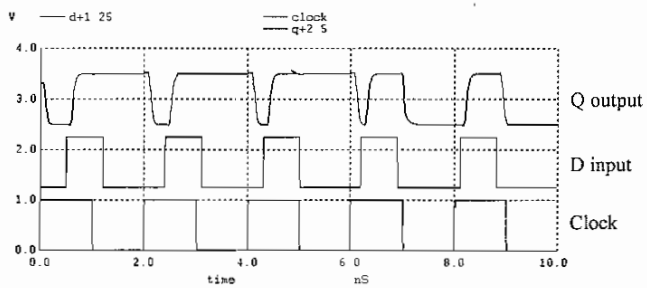


Figure 13.21 Simulating the level-sensitive latch in Fig. 13.20.

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

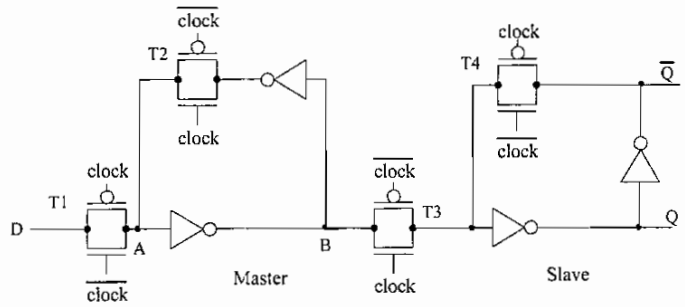


Figure 13.22 An edge-triggered D-FF.

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.

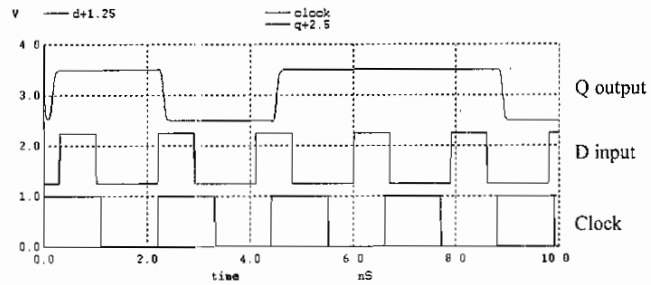


Figure 13.23 Simulating the operation of the edge-triggered FF in Fig. 13.22.

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.

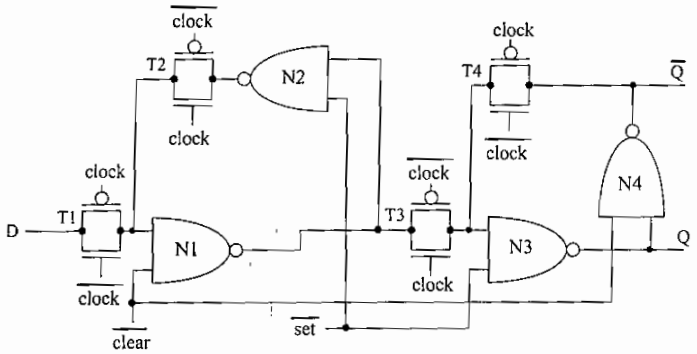


Figure 13.24 An edge-triggered FF with asynchrounous set and clear.

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.

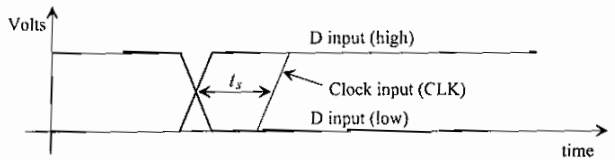


Figure 13.25 Illustrating D FF setup time.

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.

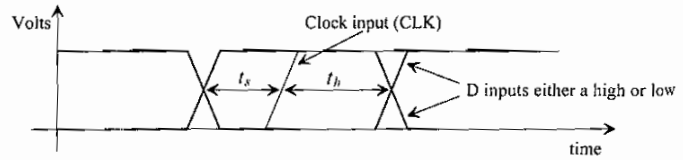


Figure 13.26 Illustrating D FF hold time.

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 $t_{PHL1} = t_{PLH1} = 71.5\,ps$.

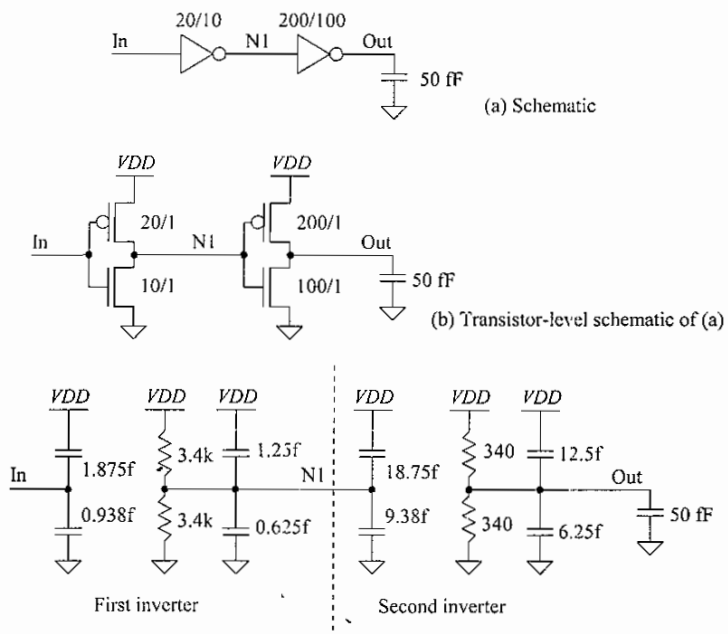


Figure 13.27 Circuit used in Ex. 13.3.

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. ■

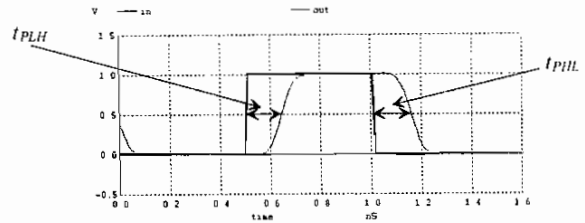


Figure 13.28 Simulating the circuit in Ex. 13.3 and Fig. 13.27.

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.

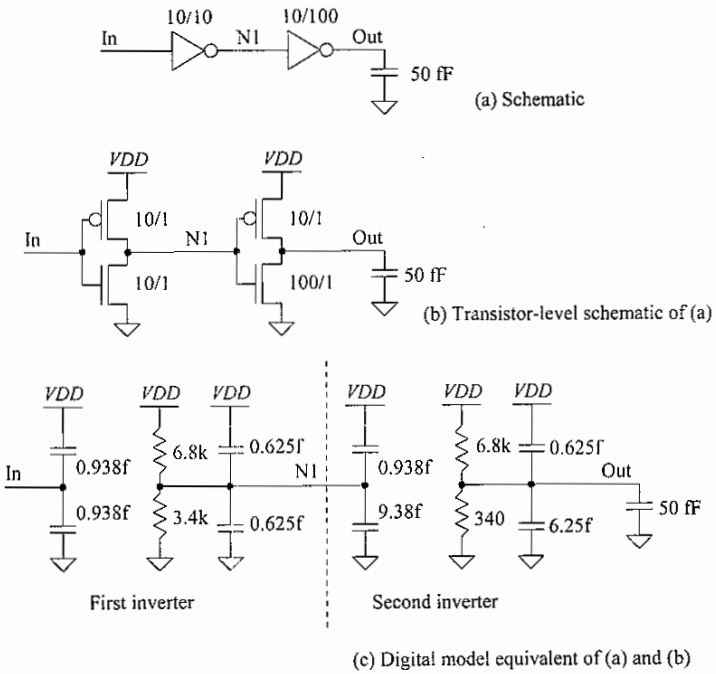


Figure 13.29 Circuit used in Ex. 13.4.

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. ■

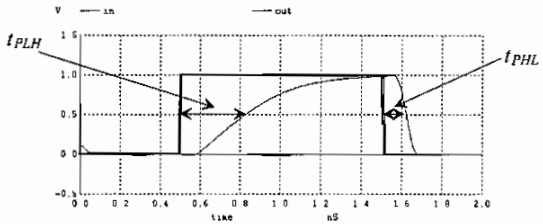


Figure 13.30 Simulating the operation of the circuit in Fig. 13.29.

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. ■

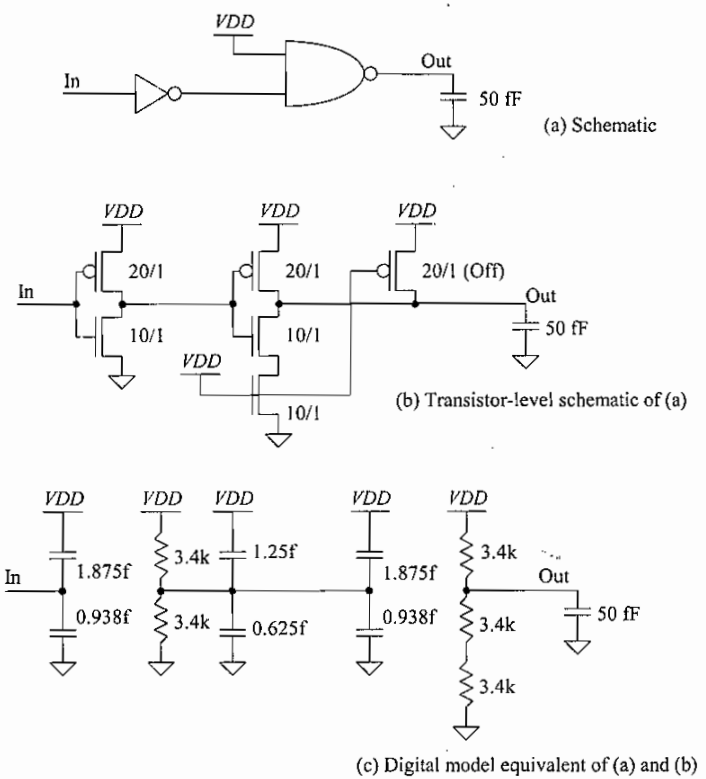


Figure 13.31 Circuit used in Ex. 13.5.

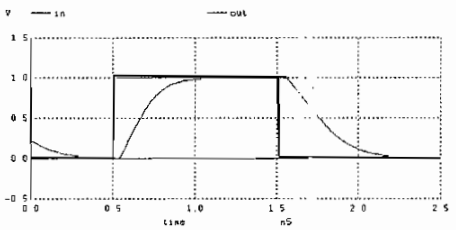


Figure 13.32 Simulating the circuit in Fig. 13.31.

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.

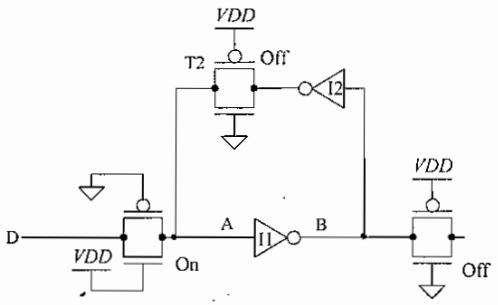


Figure 13.33 Section of a D-FF used in Ex. 13.5.

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

$$t_{PLH} = t_{PHL} = 13.5\,ps$$

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. ■

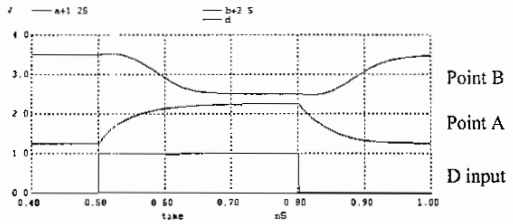


Figure 13.34 Simulating the delays through the latch seen in Fig. 13.33.

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.

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.

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).

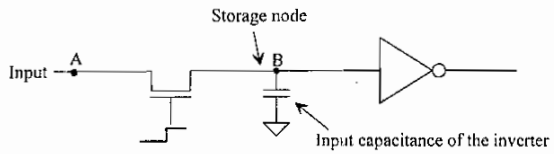


Figure 14.1 Example of a dynamic circuit and associated storage capacitance.

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

$$\log I_D = -8.45 \rightarrow I_D = 3.55\ nA = I_{off} \cdot W \cdot scale \qquad (14.1)$$

noting that the worst case off current occurs when the minimum channel length is used (L = 1). The off current can be estimated as

$$I_{off} = 7.1\ nA/\mu m \qquad (14.2)$$

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.

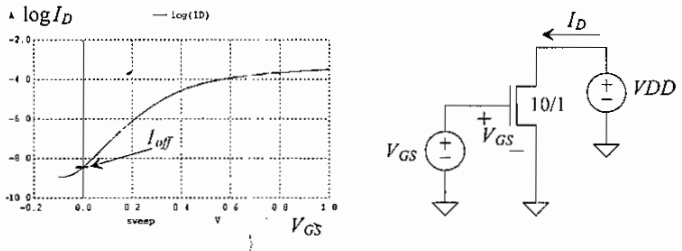


Figure 14.2 Drain current of an NMOS device plotted from weak to strong inversion. See Fig. 6.16. PMOS netlist is found at cmosedu.com.

The rate at which the storage node seen in Fig. 14.1 discharges is given by

$$\frac{dV}{dt} = \frac{I_{off} \cdot W \cdot scale}{C_{node}} \qquad (14.3)$$

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

$$C_{node} \approx C_{in} \text{ of the inverter} \qquad (14.4)$$

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

$$\frac{dV}{dt} = \frac{3.55\ nA}{50\ fF} = 71\ mV/\mu s$$

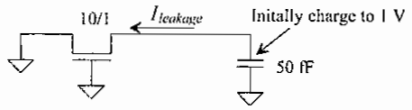


Figure 14.3 Circuit used in Ex. 14.1.

For the capacitor to discharge to ground (from an initial voltage of 1 V) takes, roughly, (1 V)/(71 $mV/\mu s$) = 14 μs. 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. ■

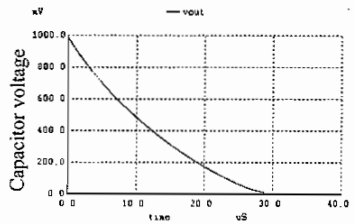


Figure 14.4 Time it takes the capacitor to discharge due to the off current of the MOSFET in Fig. 14.3.

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

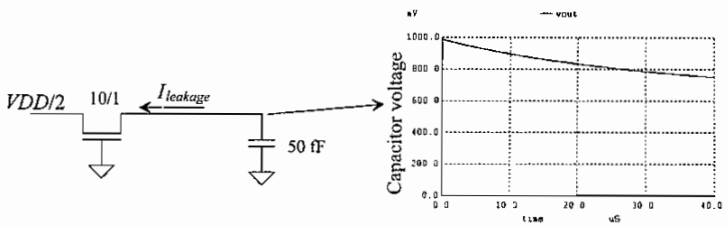


Figure 14.5 Circuit used in Ex. 14.2.

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.

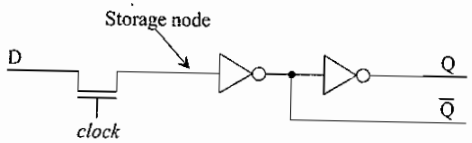


Figure 14.6 A dynamic level-sensitive latch.

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). ■

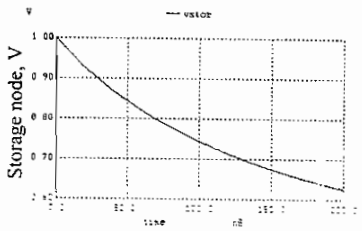


Figure 14.7 How the voltage on the storage node varies in the dynamic latch circuit seen in Fig. 14.6.

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 (like WinSPICE) are also capable of adding a resistor at every node to ground called RSHUNT (to help with convergence, see page 78). 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 ϕ_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 ϕ_2 is low while ϕ_1 is high, the data cannot pass from A0 to A1 and from A2 to A3. If ϕ_1 goes low and ϕ_2 goes high, data is passed from A0 to A1 and from A2 to A3. If both ϕ_1 and ϕ_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, ϕ_1 and ϕ_2. Note that a simple logic inversion will not generate nonoverlapping clock signals. Consider the schematic of the

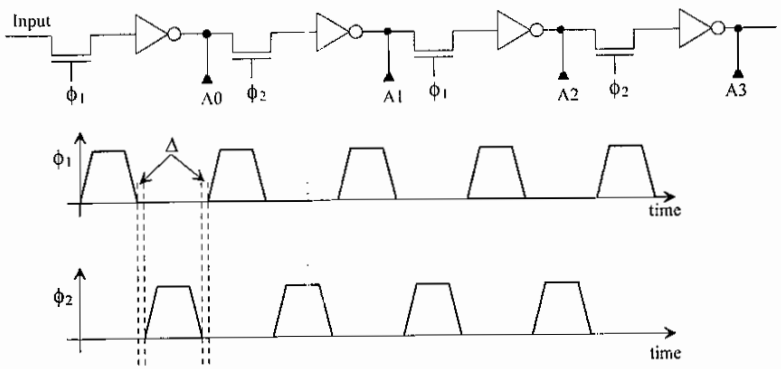


Figure 14.8 Dynamic shift register with associated nonoverlapping clock signals.

nonoverlapping clock generator shown in Fig. 14.9. This circuit takes a clock signal and 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 ϕ_1 high and ϕ_2 low. When the input clock goes low, ϕ_1 goes low. After ϕ_1 goes low, ϕ_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.

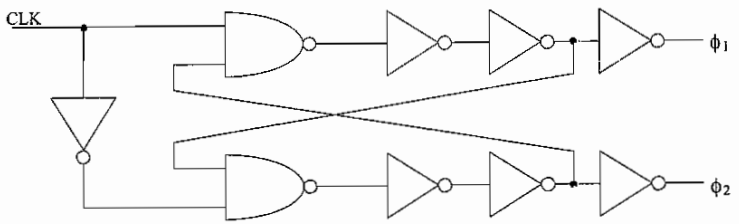


Figure 14.9 Nonoverlapping clock generation circuit.

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.

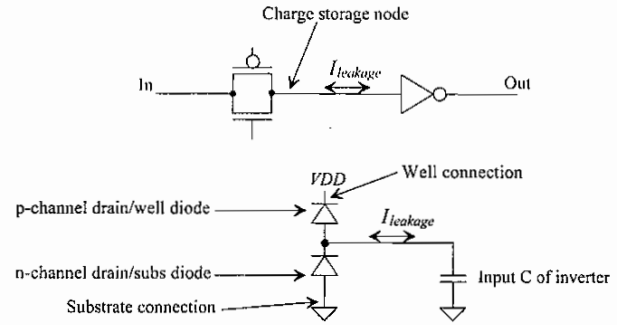


Figure 14.10 CMOS TG used in dynamic logic.

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 ϕ_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 ϕ_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 ϕ_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.

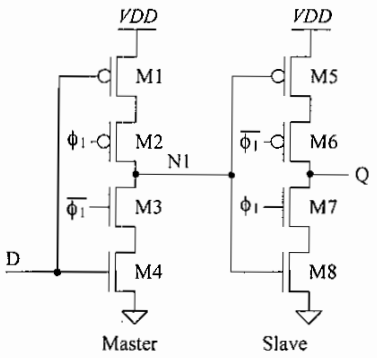


Figure 14.11 A clocked CMOS latch. The clock signals can be generated with an RS latch so that the edges occur essentially at the same moment in time.

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 ϕ_1 is low, the output node capacitance is charged to *VDD* through M5. During the evaluate phase, ϕ_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 ϕ_1 is high. The output is a logic one when ϕ_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.

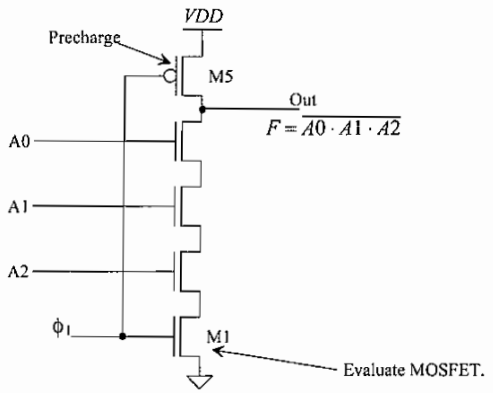


Figure 14.12 Precharge-evaluate three-input NAND gate.

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.

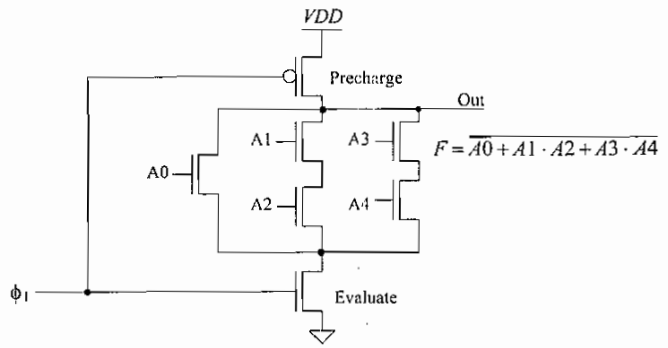


Figure 14.13 A complex PE gate.

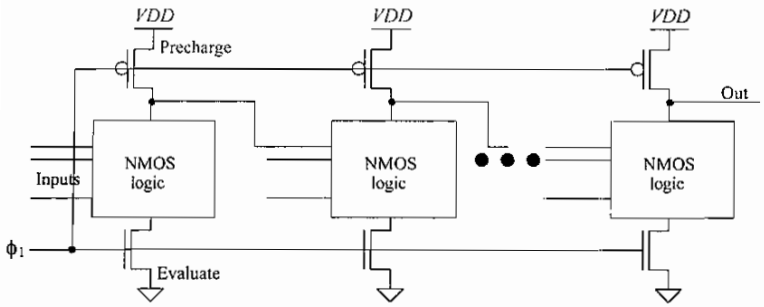


Figure 14.14 Problems with a cascade of PE gates.

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.

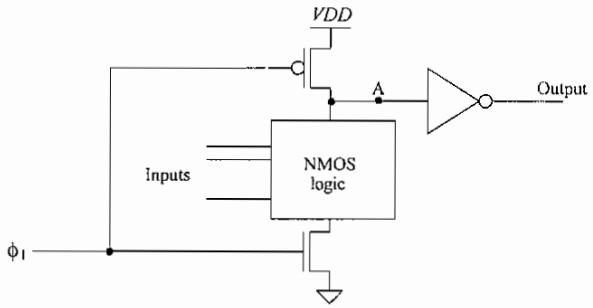
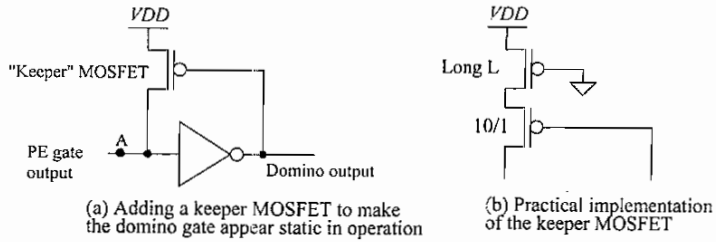


Figure 14.15 Domino logic gate.



(a) Adding a keeper MOSFET to make the domino gate appear static in operation

(b) Practical implementation of the keeper MOSFET

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.

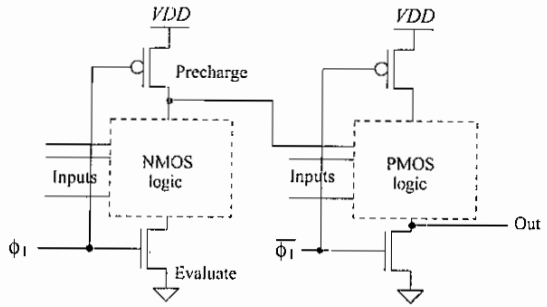


Figure 14.17 NP logic.

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

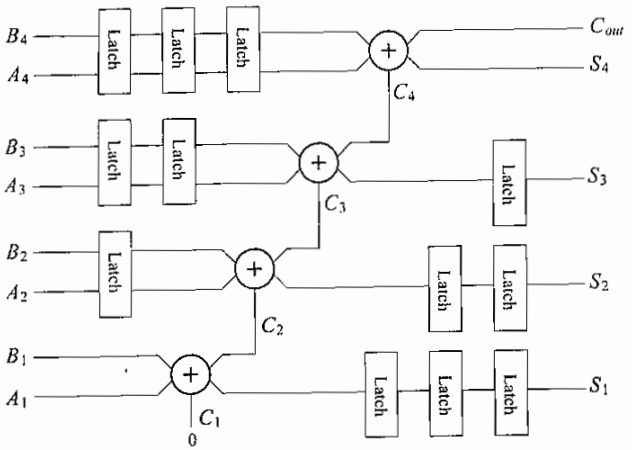


Figure 14.18 A pipelined adder. The latches (clocked) behave as delay elements.

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] 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.

[2] N. H. E. Weste and D. Harris, *Principles of CMOS VLSI Design,* Addison-Wesley, 3rd ed., 2004. ISBN 0-321-14901-7.

[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}.

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 ϕ_1 and ϕ_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 [1]. Note that a single-phase clock signal is used.

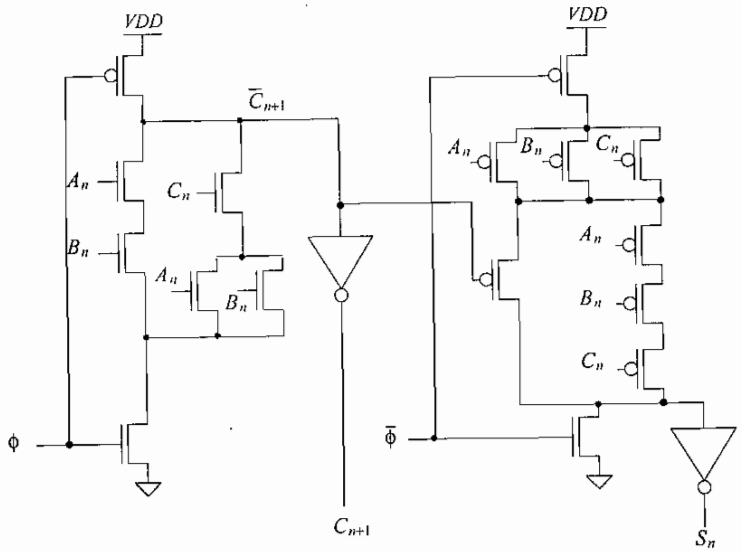


Figure 14.19 A high speed adder cell. See Problem 14.10.

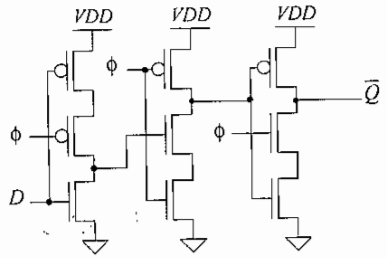


Figure 14.20 A true-single phase clocked FF, see Problem 14.15.

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 and, thus, a failing chip. 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 device dimensions

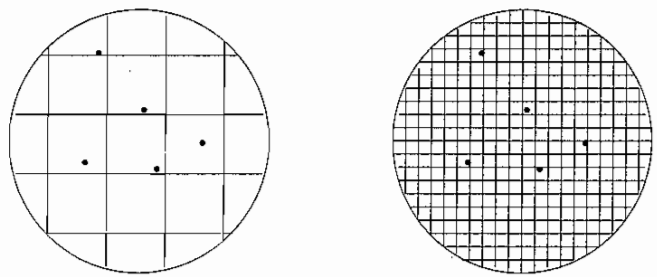


Figure 15.1 Defect density effects on yield.

(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.

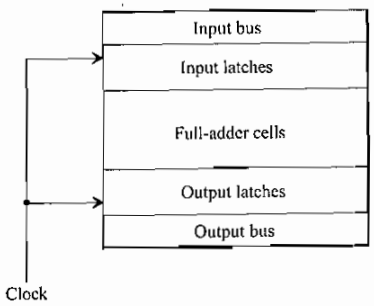


Figure 15.2 Floor plan for an adder.

[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.

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.

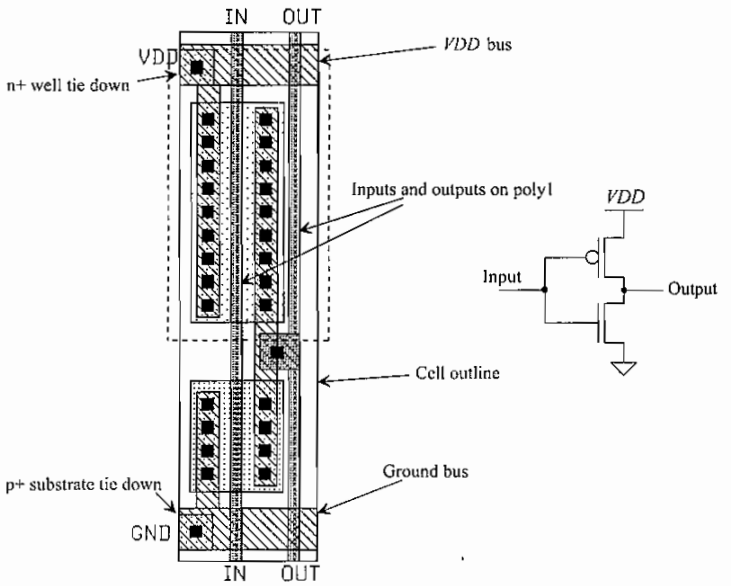


Figure 15.3 Standard cell layout of an inverter.

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.

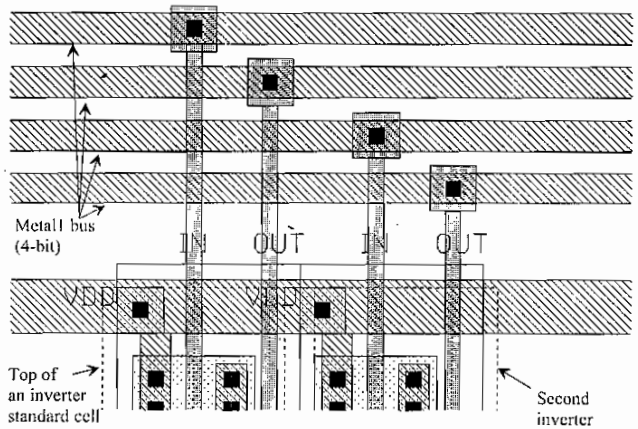


Figure 15.4 Connection of two inverter standard cells to a bus.

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,

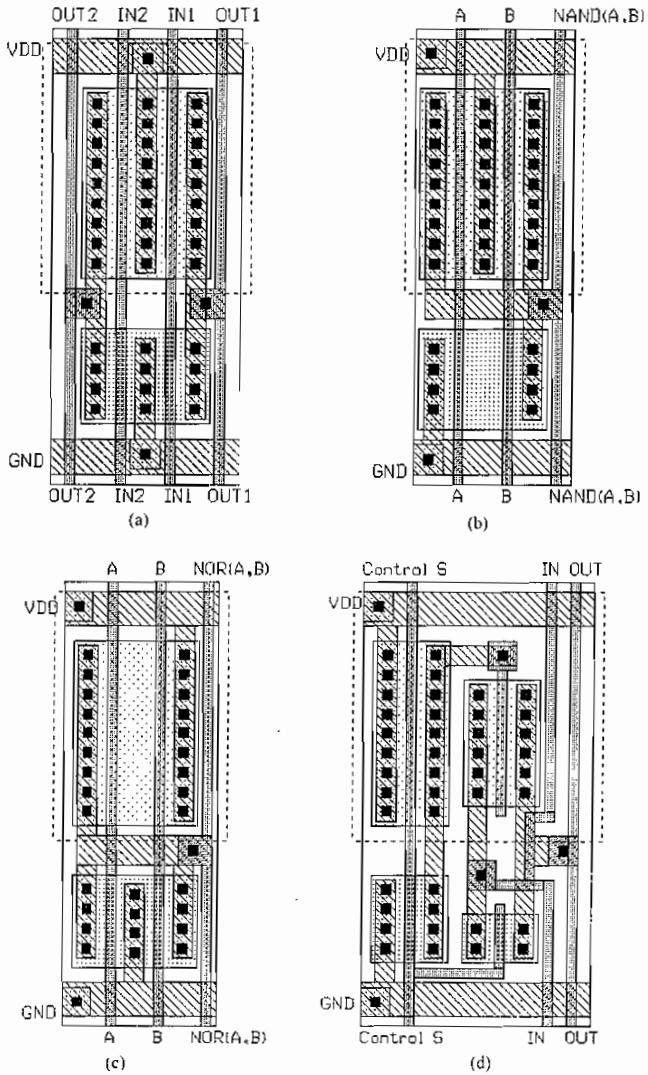


Figure 15.5 (a) Double inverter, (b) two-input NAND, (c) two-input NOR, and (d) transmission gate.

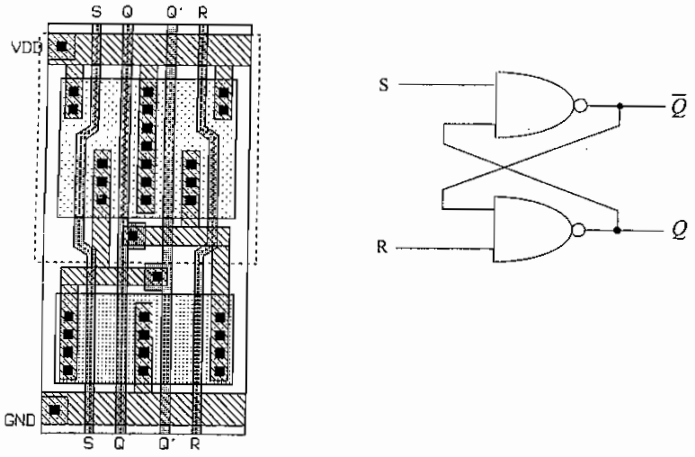


Figure 15.6 SR latch using NAND gates.

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.

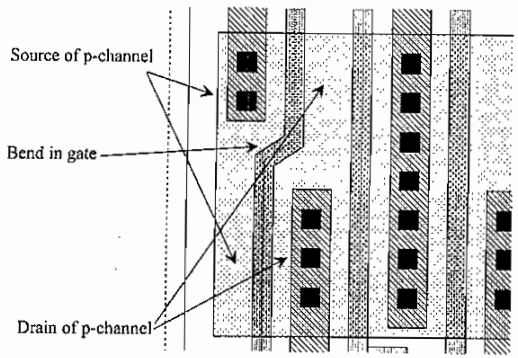


Figure 15.7 Section of the layout shown in Fig. 15.6.

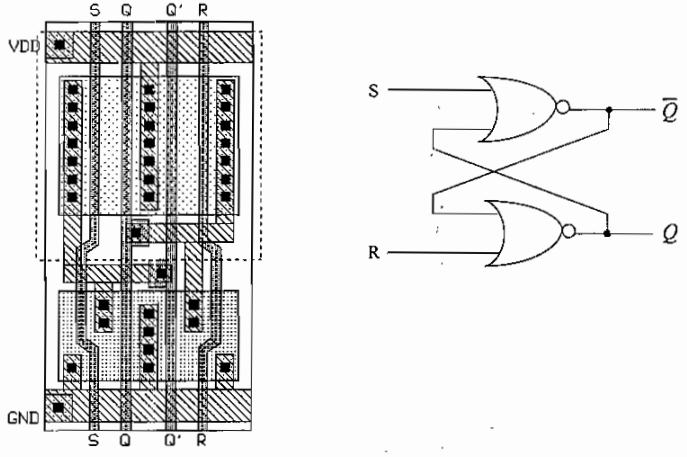


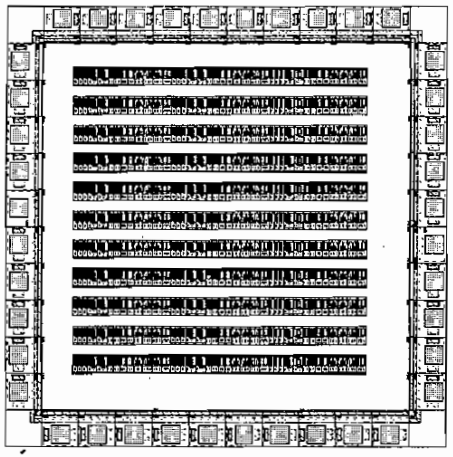
Figure 15.8 SR latch using NOR gates.

Power and Ground Connections

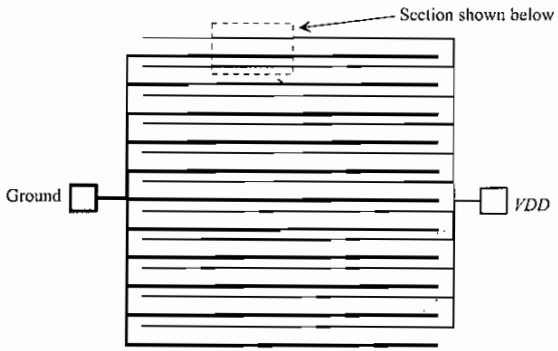
Many of the problems encountered when designing a chip can be related to the distribution of power and ground. When power and ground are not distributed properly, noise can be coupled from one circuit onto the power and ground conductors and injected into some other circuit.

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.

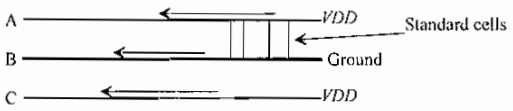
Coupling is a problem on signal buses as well. Figure 15.11 shows a simple scheme to reduce coupling. The length of a section, where two wires are adjacent, is reduced by routing the wire to other locations at varying distances along the bus. The inductive or capacitive coupling between two conductors is directly related to the length of the wire runs.



(a)



(b)



(c)

Figure 15.9 Connection of power and ground to standard cells.

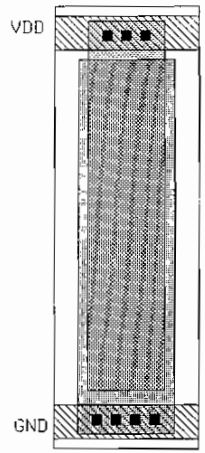


Figure 15.10 Decoupling capacitor.

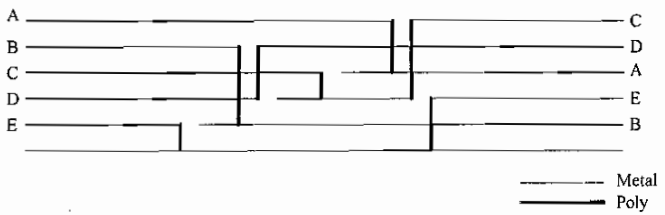
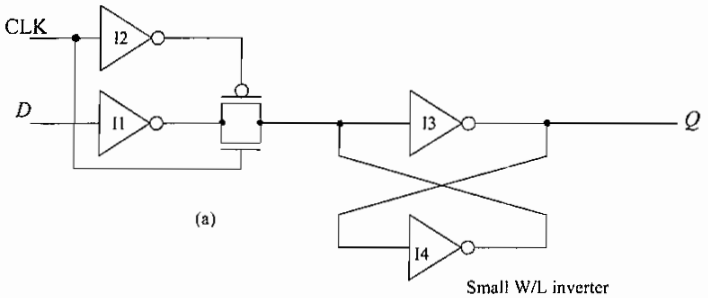


Figure 15.11 Busing structure used to decrease signal coupling.

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.



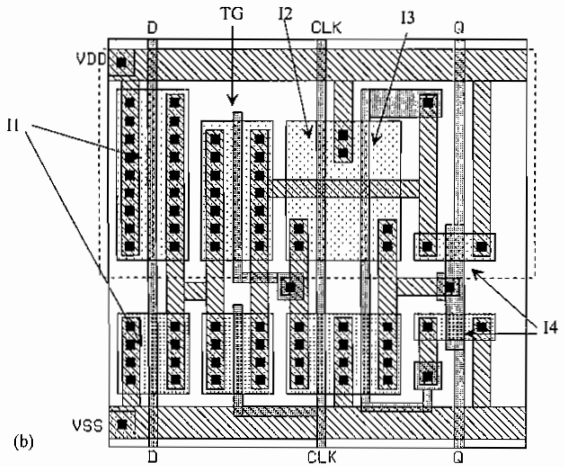


Figure 15.12 Schematic and layout of a latch.

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 $\overline{CLK}$ 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.

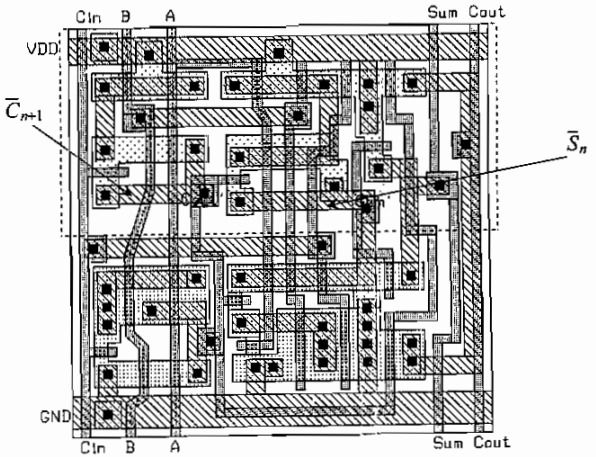


Figure 15.13 Layout of an AOI static adder.

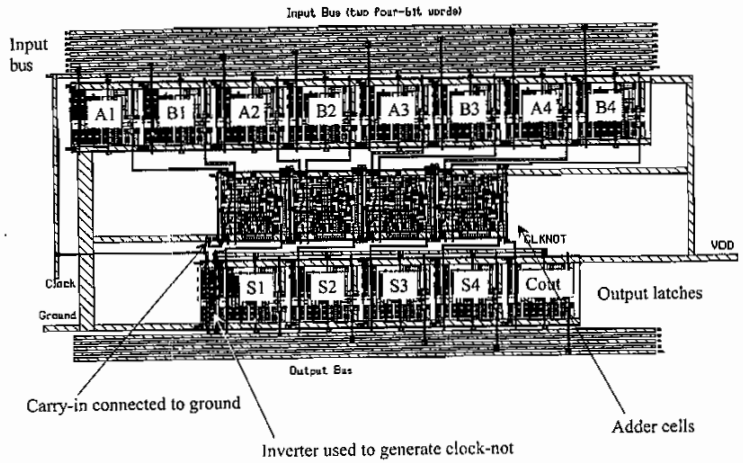


Figure 15.14 Layout of the complete adder.

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.

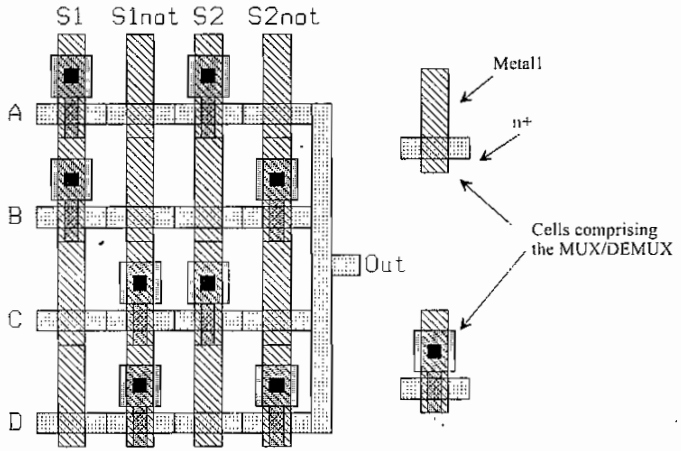


Figure 15.15 Layout of a 4-to-1 MUX/DEMUX.

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.

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.

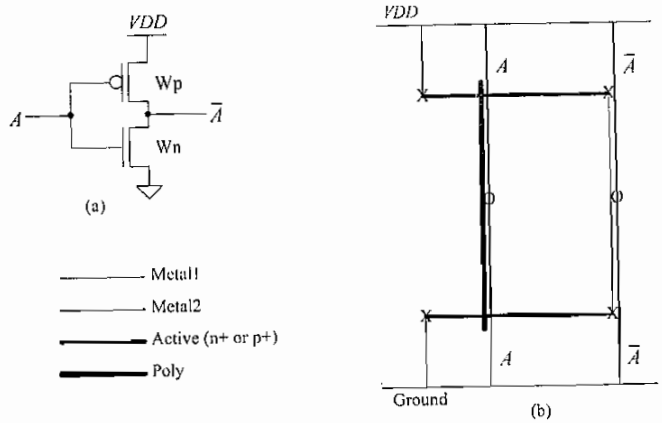


Figure 15.16 (a) Inverter and (b) stick diagram used for layout.

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.

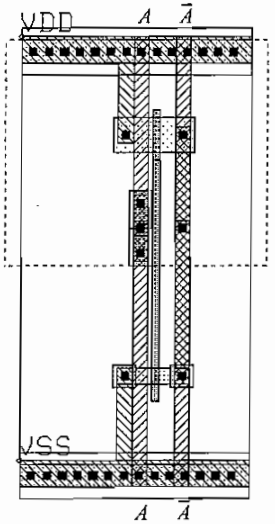
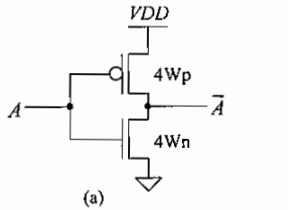


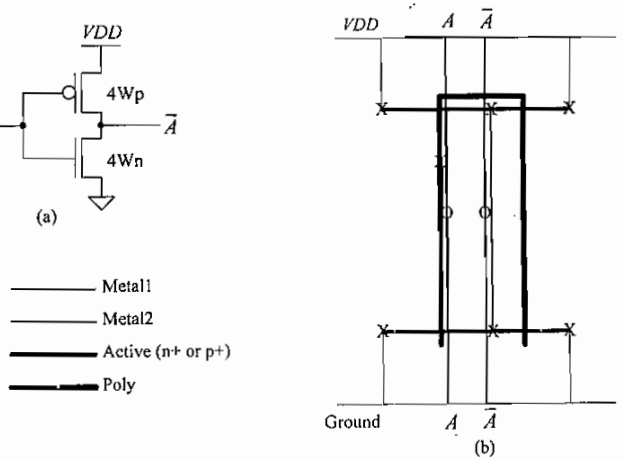
Figure 15.17 Layout of the inverter shown in Fig. 15.16.

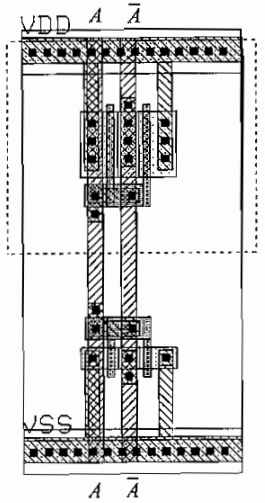
Take another look at the two circuits from a geometrical rather than an electrical viewpoint. Compare the NAND gate layout to the NOR gate layout. Each can be created from the other by simply "flipping" the metal and poly connections about the x-axis.

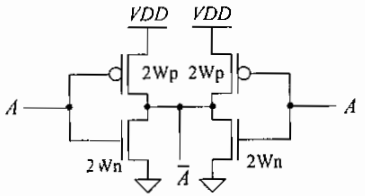
Device Placement

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.









(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.

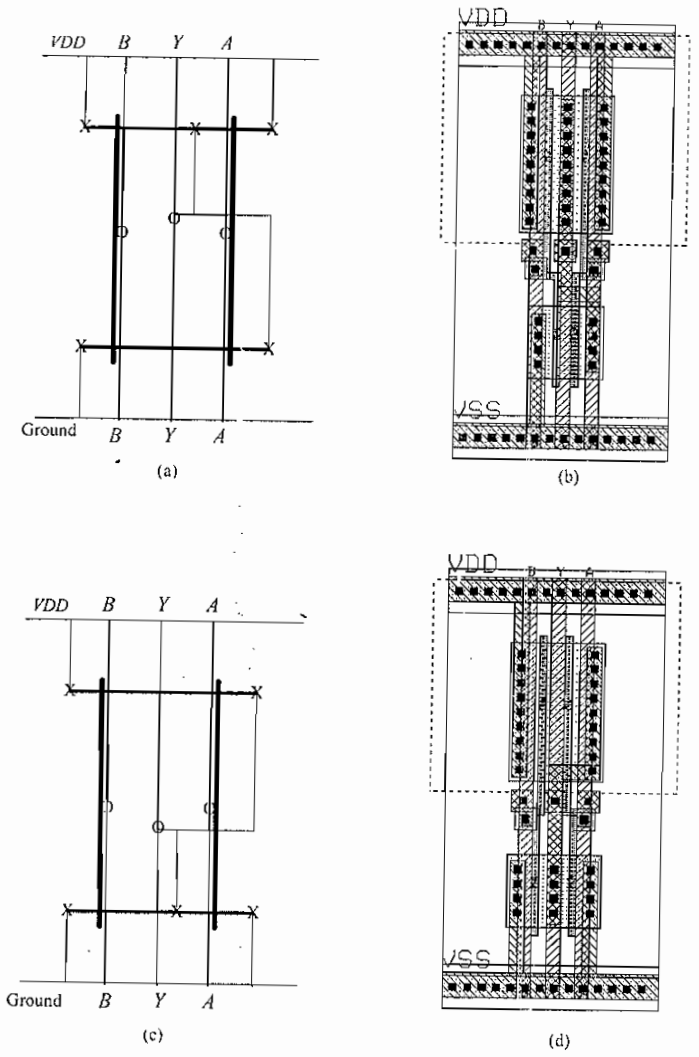


Figure 15.19 (a) NAND stick diagram, (b) layout, (c) NOR stick diagram, and (d) layout.

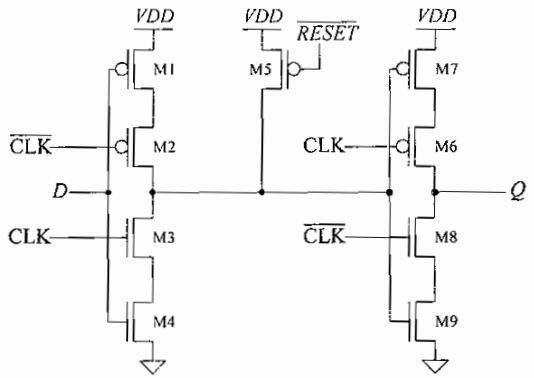


Figure 15.20 Schematic of a dynamic register cell.

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 part of the standard cells themselves. All remaining connections are made via the

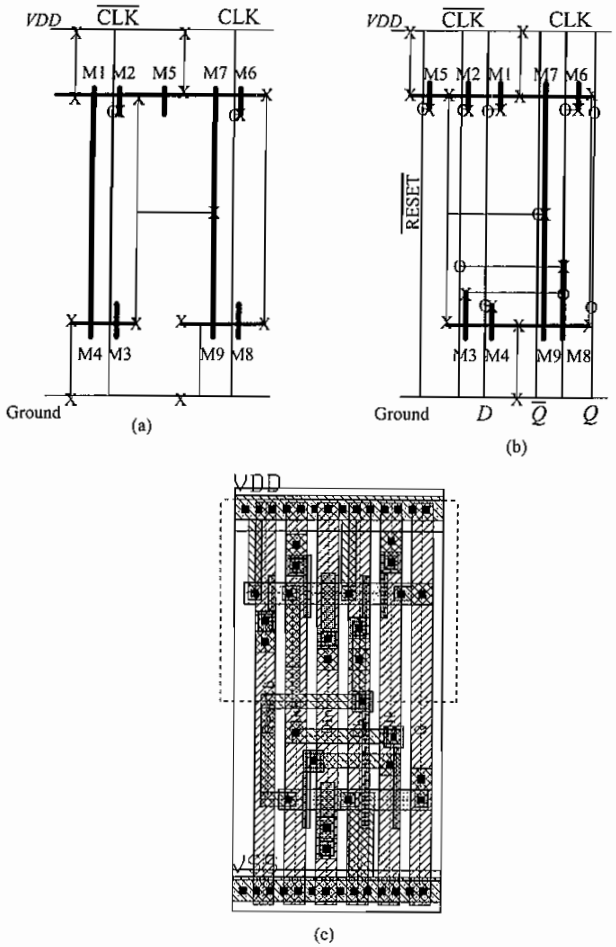


Figure 15.21 Layout of a dynamic register cell.

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.

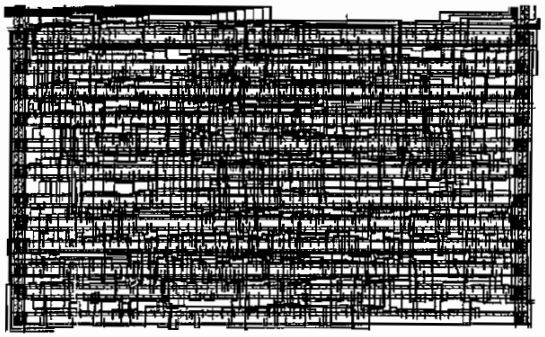


Figure 15.22 Layout based on standard cells.

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 case, the width of each cell depended on the maximum allowable width of the widest cell

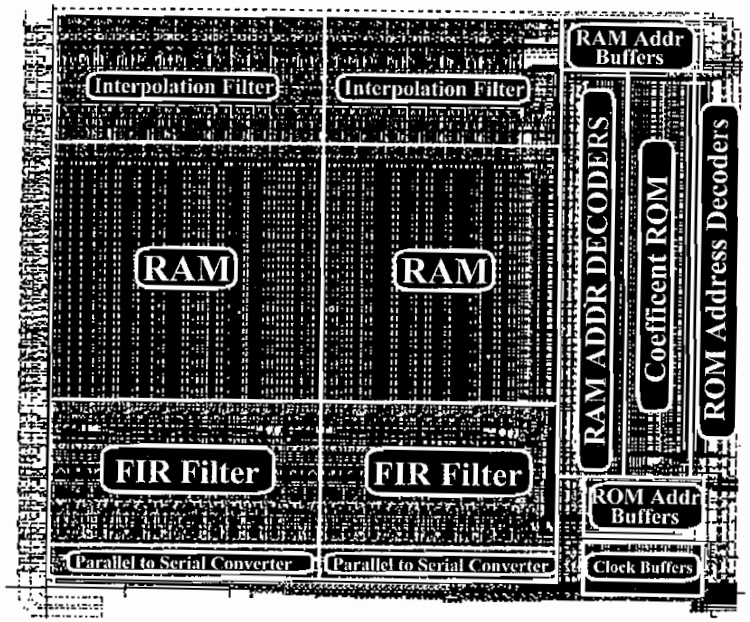


Figure 15.23 Full custom layout of a digital filter.

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 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

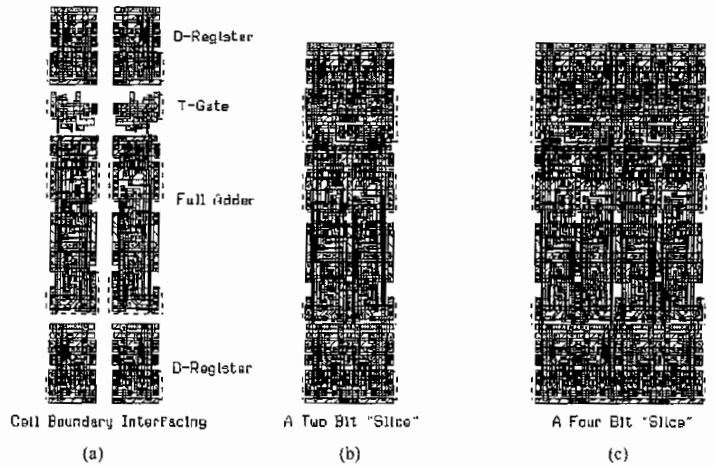


Figure 15.24 Sections of the digital interpolation filter.

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*, Addison-Wesley, 3rd ed., 2004. ISBN 0-321-14901-7.

[2] J. P. Uyemura, *Introduction to VLSI Circuits and Systems*, John Wiley and Sons Publishers, 2002. ISBN 0-471-12704-3.

[3] Kerth, Donald A. *"Floorplanning-Lecture Notes,"* Crystal Semiconductor, Inc.

[4] Kerth, Donald A. *"Analog Tricks of the Trade-Lecture Notes."* Crystal Semiconductor, Inc.

[5] J. Uyemura, *Physical Design of CMOS Integrated Circuits Using L-EDIT*, PWS Publishing Co., 1995. ISBN 0-534-94326-8.

[6] D. V. Heinbuch, *CMOS3 Cell Library*, Addison-Wesley, 1988. ISBN 0-201-11257-4.

[7] 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.

[8] 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.

[9] C. Saint and J. Saint, *IC Mask Design: Essential Layout Techniques*, McGraw-Hill, 2002. ISBN 0-071-38996-2. Companion book for reference [8].

[10] G. Petley, *The Art of Standard Cell Library Design*, to be published. See the website: http://www.vlsitechnology.org/ for more information (including example layouts and schematics).

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

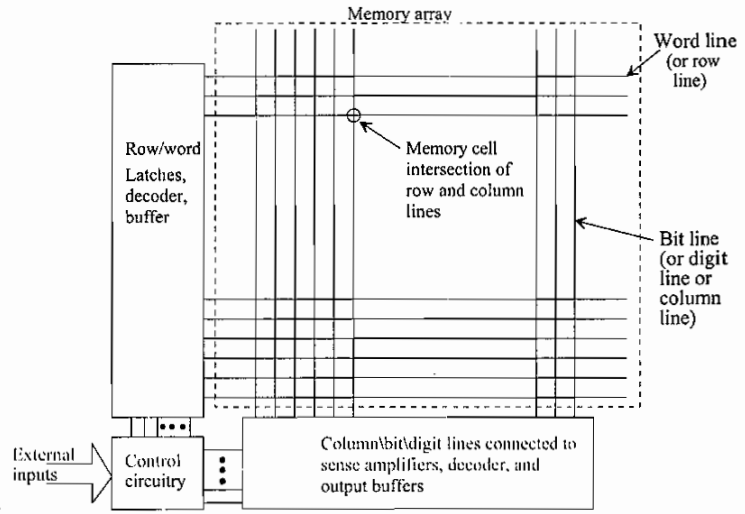


Figure 16.1 Block diagram of random access memory.

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/μm², 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.

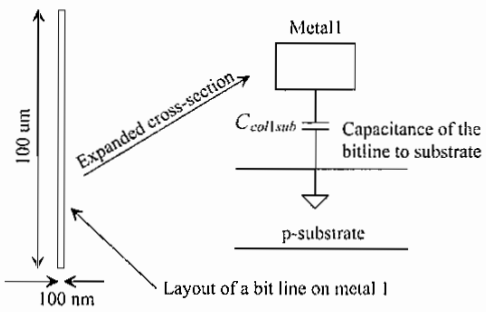


Figure 16.2 The parasitic capacitance of a bitline to ground (substrate).

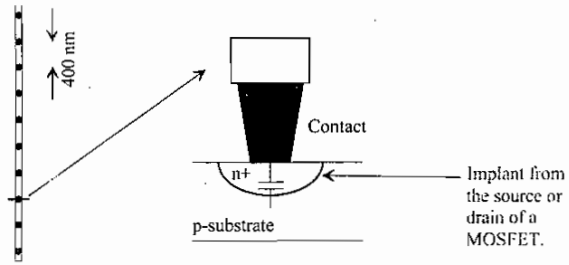


Figure 16.3 How a bit line is loaded with implants (depeletion capacitance).

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, ΔV_{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.

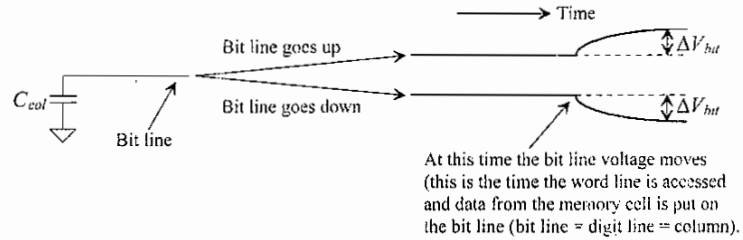


Figure 16.4 Sensing a change on the bit line in an array.

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?

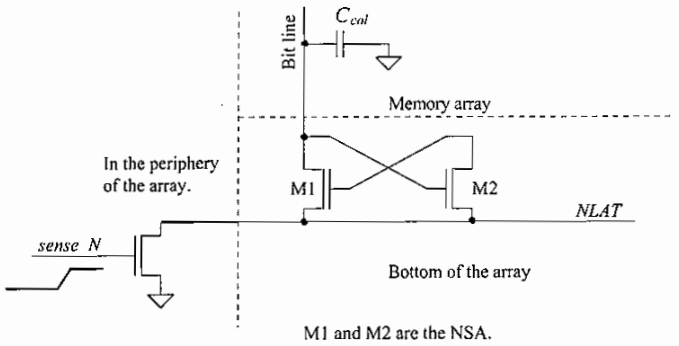


Figure 16.5 Development of an NMOS sense amplifier (NSA).

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.

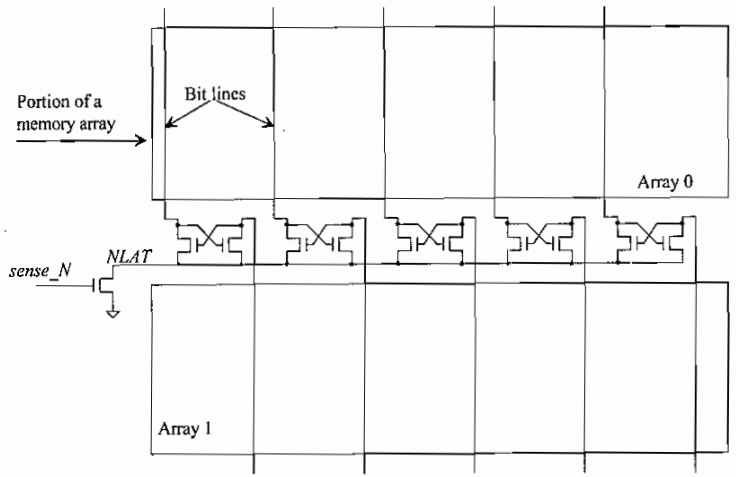


Figure 16.6 How the NSA is placed between two memory arrays in the so-called open memory array architecture.

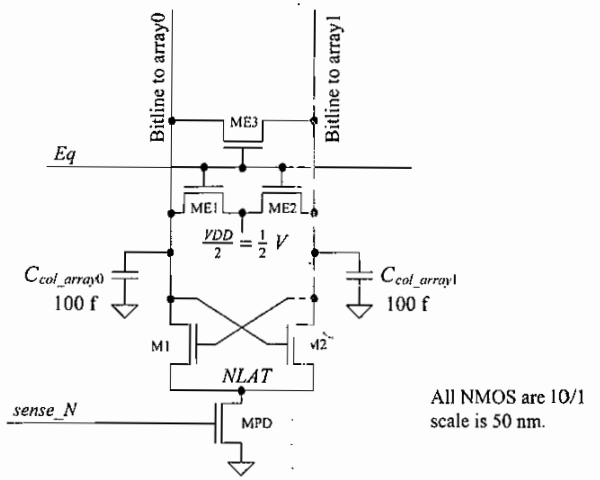


Figure 16.7 An NSA with equilibration circuitry and connections to two bit lines.

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

$$P_{avg} = (\text{number of sense amplifiers}) \cdot C_{col} \cdot (VDD/2)^2 \cdot f \qquad (16.4)$$

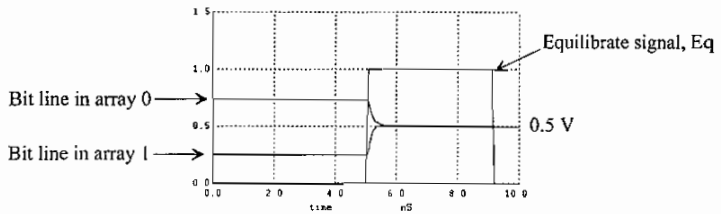


Figure 16.8 How the equilibrate circuitry operates.

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

$$Q_{tot} = C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col_array} \qquad (16.5)$$

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

$$V_{final}(C_{mbit} + C_{col_array}) = C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col_array} \qquad (16.6)$$

or

$$V_{final} = \frac{C_{mbit}V_{mbit} + (VDD/2) \cdot C_{col_array}}{C_{mbit} + C_{col_array}} \qquad (16.7)$$

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

$$\Delta V_{bit} = V_{final} - VDD/2 \qquad (16.8)$$

or here $\Delta V_{bit} = 83\ mV$.

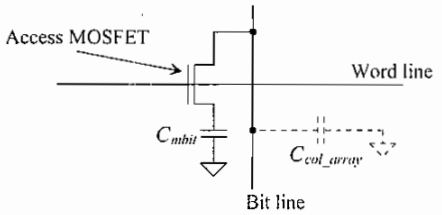


Figure 16.9 The one-transistor, one-capacitor (1T1C) DRAM memory cell.

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

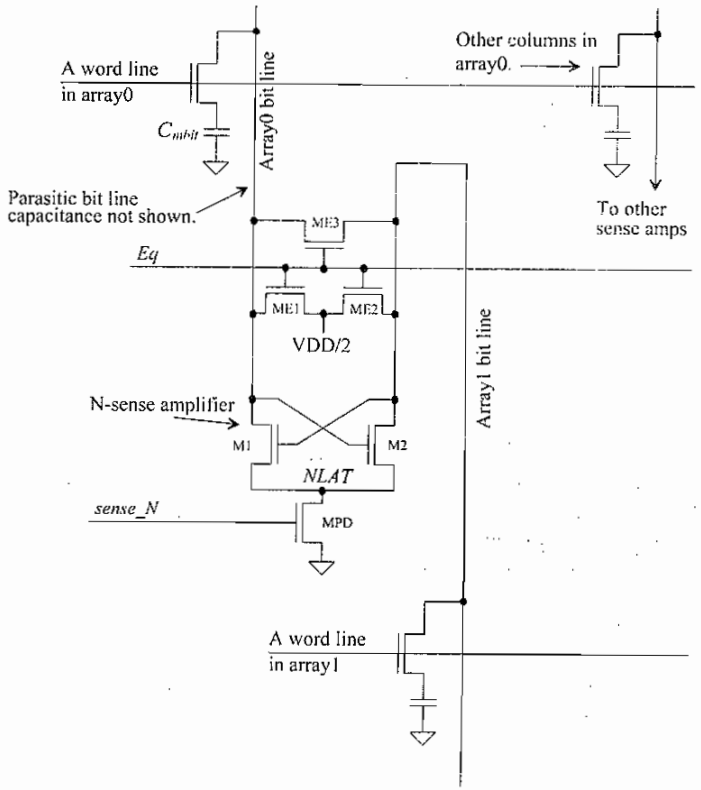


Figure 16.10 The connection of the NSA to the memory arrays.

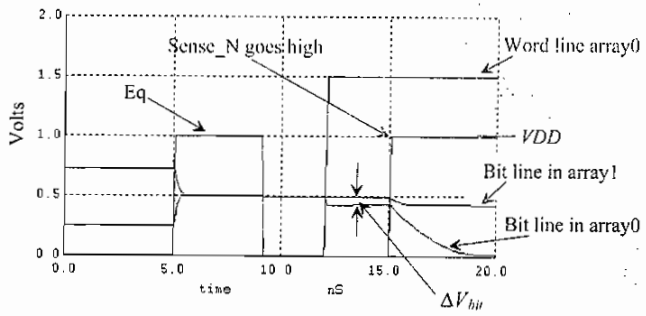


Figure 16.11 N-sense amp's operation.

out zero. 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).

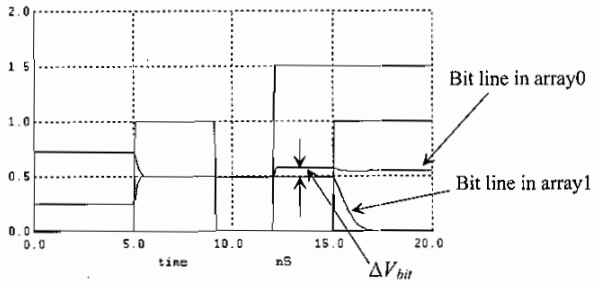


Figure 16.12 Reading out a "1" from the cell in array0.

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 the matching of the PMOS. It's not common to fire the sense amplifiers at the same time

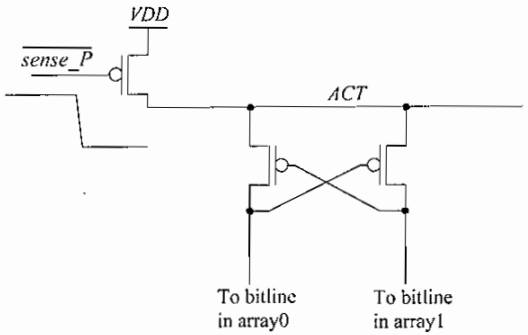


Figure 16.13 Schematic diagram of a PMOS sense amplifier.

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.

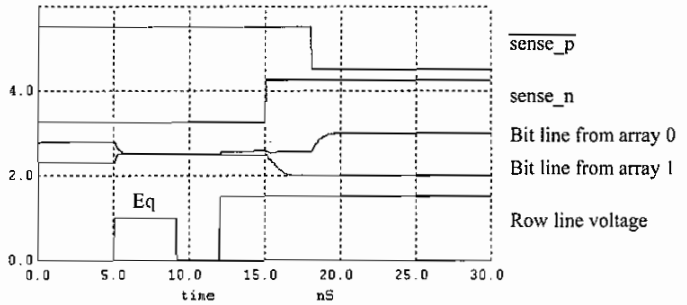


Figure 16.14 How the PSA pulls the bit line from array 0 high.

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

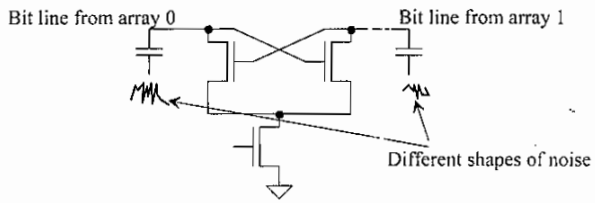


Figure 16.15 Different amplitudes of coupled noise into physically separated bit line.

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.

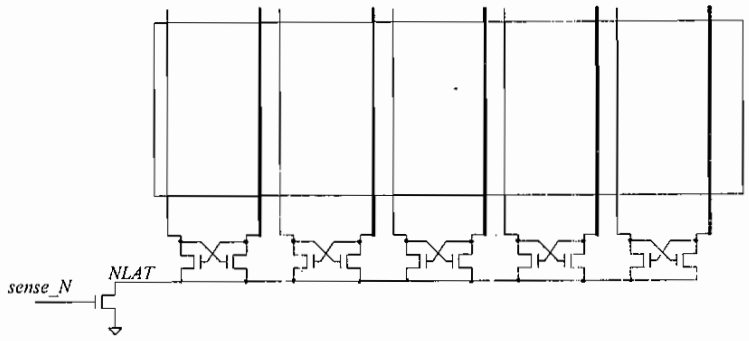


Figure 16.16 The folded array is formed by taking the open array architecture (open book) topology seen in Fig. 16.6 and "closing the book," that is, folding array 1 on top of array 0. Note that the bold lines indicate the bitlines from array 1 in the newly formed array.

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.

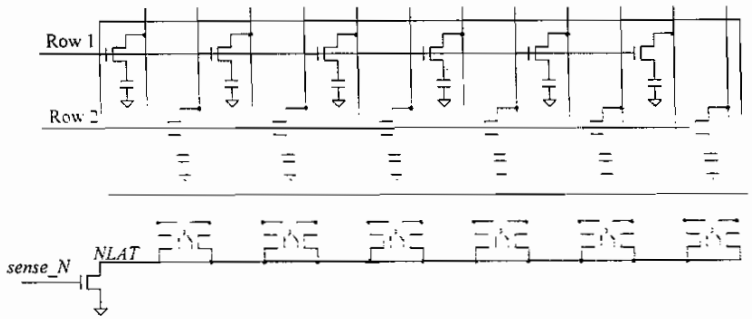


Figure 16.17 How a memory cell is located at every other intersection of a row line and a column line in a folded-area architecture.

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

$$t_d = \overbrace{(\text{number of columns}) \cdot \left(WL\frac{\varepsilon_{ox}}{t_{ox}} + C_{parasitic}\right)}^{\text{total capacitance on the word line}} \cdot \overbrace{(\text{number of columns}) \cdot R_{gat}}^{\text{total resistance of the word line}} \quad (16.9)$$

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.

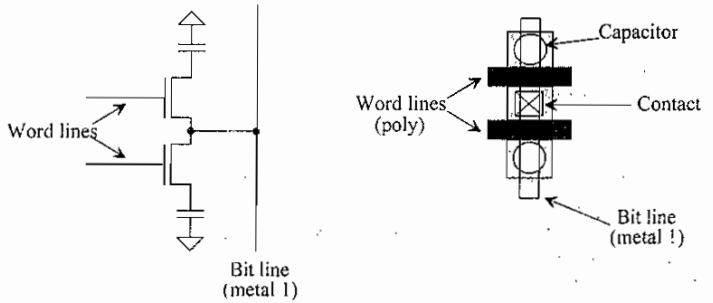


Figure 16.18 Two mbits sharing a contact to the bit line.

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).

F is feature size, which is half the bit line pitch; that is, F = pitch/2.

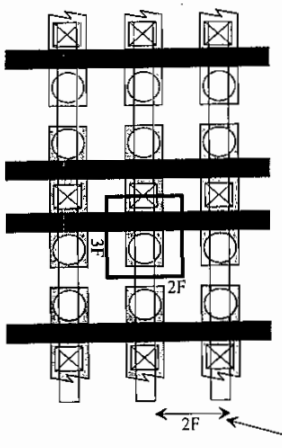


Figure 16.19 Layout of mbit used in an open bit line configuration.

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

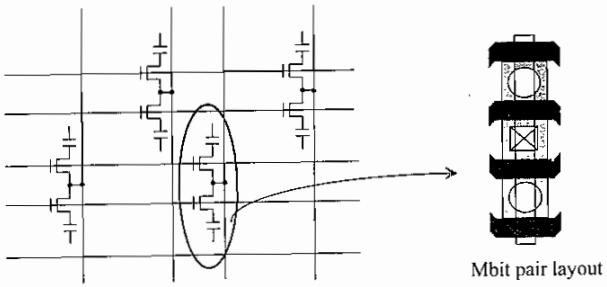


Figure 16.20 The mbit pair used for a folded architecture.

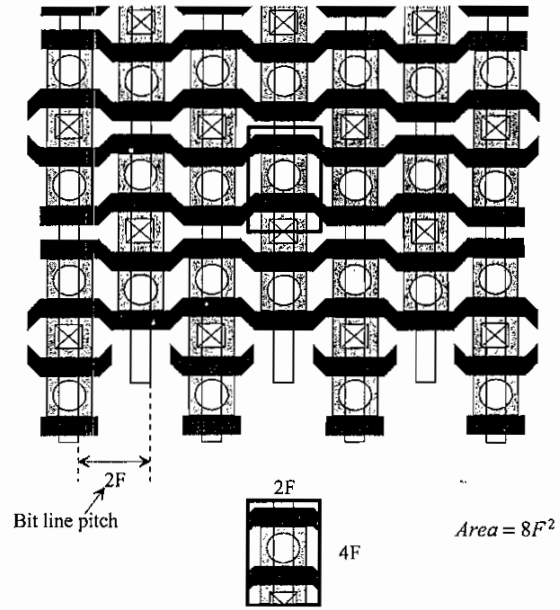


Figure 16.21 Folded architecture layout and cell size.

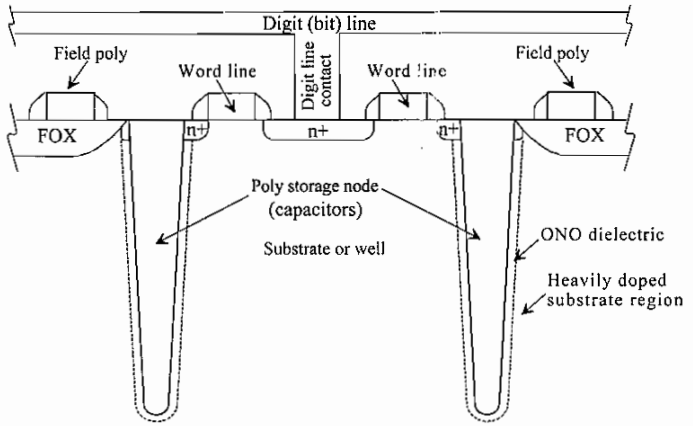


Figure 16.22 Cross-sectional view of a trench capacitor cell.

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 trench 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.

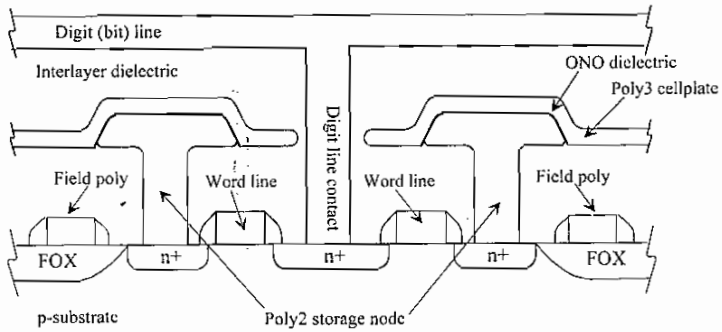


Figure 16.23 Cross-sectional view of a buried capacitor cell.

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.

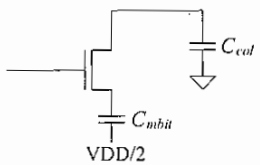


Figure 16.24 Holding the capacitor's common plate at VDD/2 to minimize oxide stress.

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 8-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.

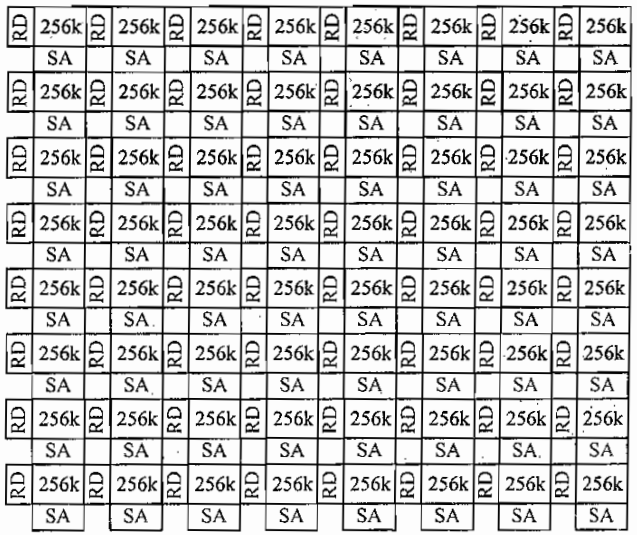


Figure 16.25 A 16-Mbit array block.

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.

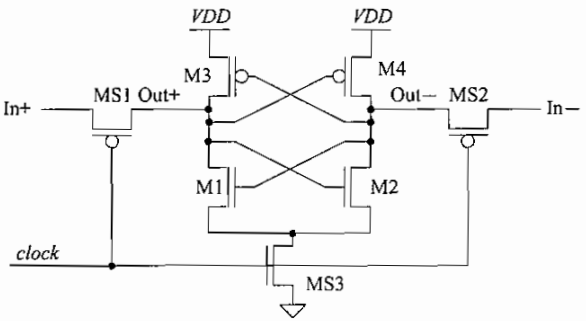


Figure 16.26 Clocked sense amplifier.

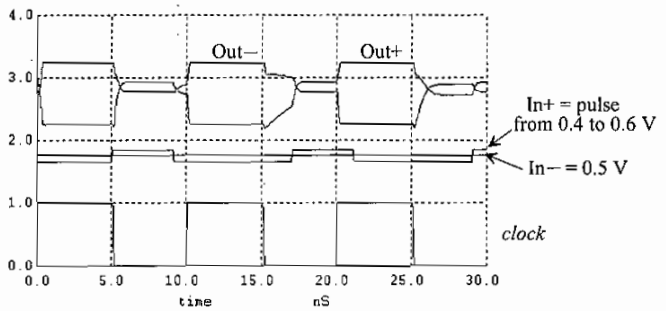


Figure 16.27 Simulating the operation of the circuit in Fig. 16.26.

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.

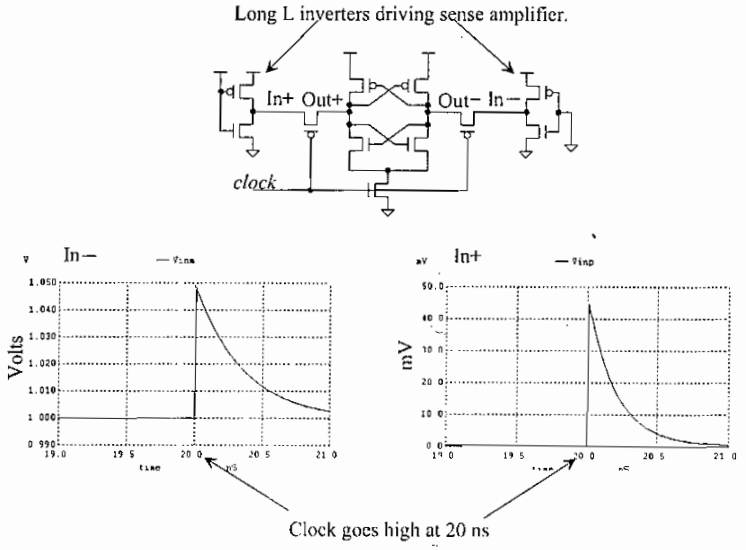


Figure 16.28 Clock feedthrough noise.

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 large). 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.*

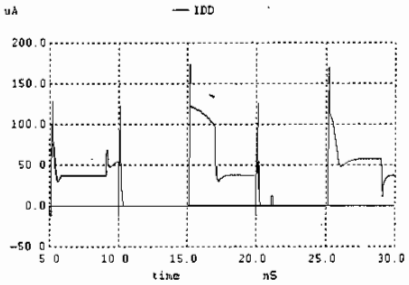


Figure 16.29 The current that flows in the sense amplifier in Fig. 16.27

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

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.

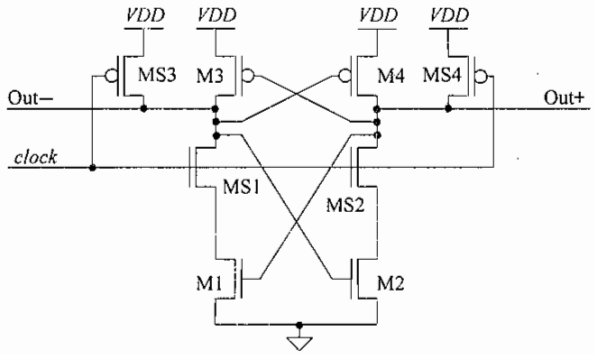


Figure 16.30 Removing memory in a sense amplifier.

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 power 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*.

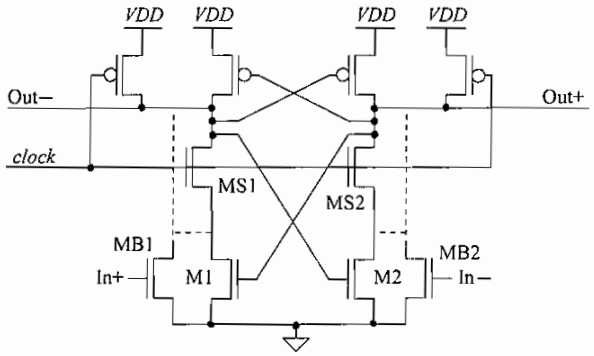


Figure 16.31 Creating an imbalance in a sense amplifier.

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.

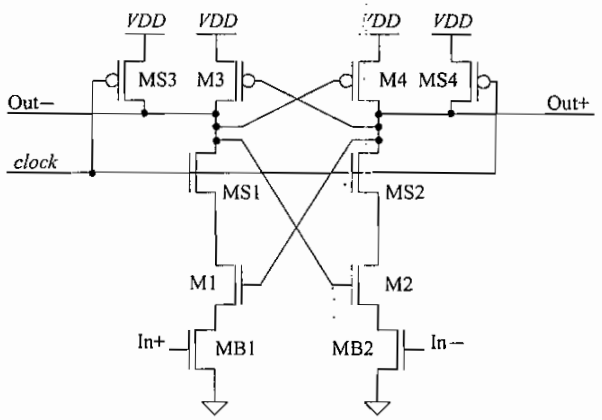


Figure 16.32 Reducing power in a sense amplifier.

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.

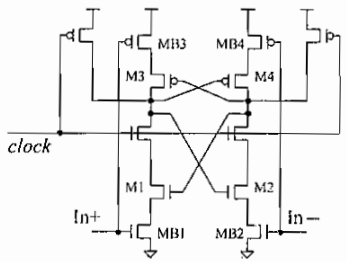


Figure 16.33 A bad design: how not to get wide-swing operation.

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).

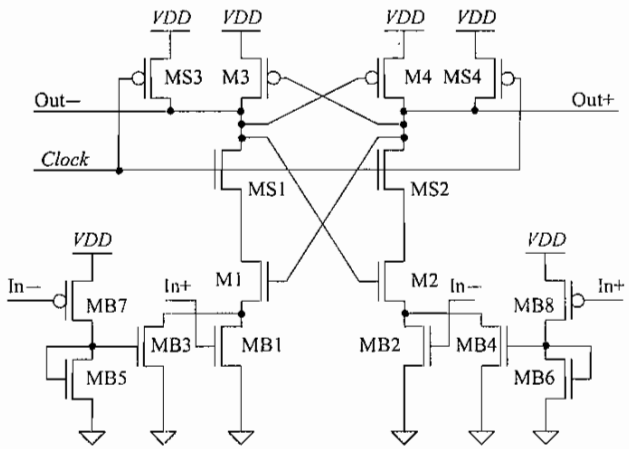


Figure 16.34 Rail-to-rail input signal range. See also Fig. 27.16.

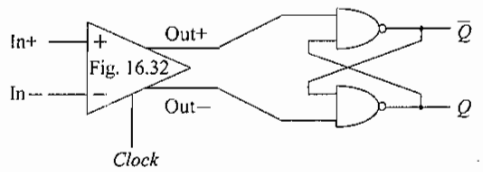


Figure 16.35 Adding an SR latch so output only changes on rising edge of the clock.

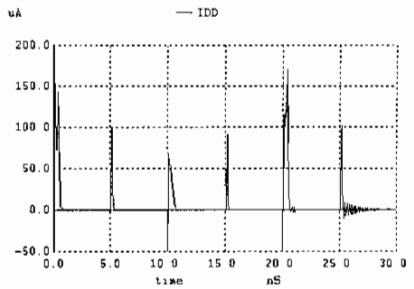


Figure 16.36 Regenerating the plot seen in Fig. 16.29 with the sense amplifier from Fig. 16.35. The current includes the switching current from the NAND gates.

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.

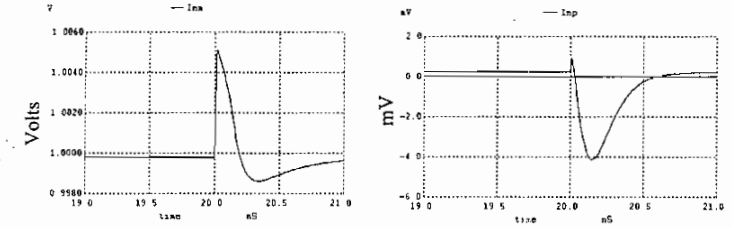


Figure 16.37 Kickback noise.

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.

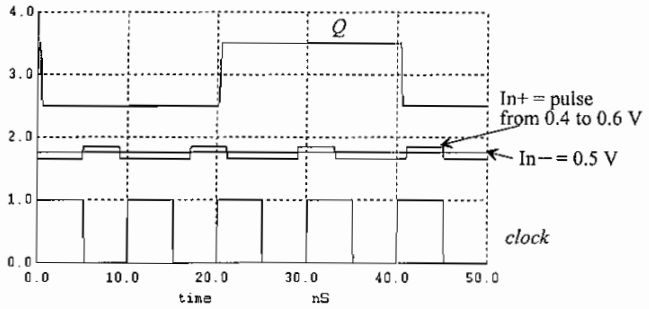


Figure 16.38 The operation of the circuit in Fig. 16.35.

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.

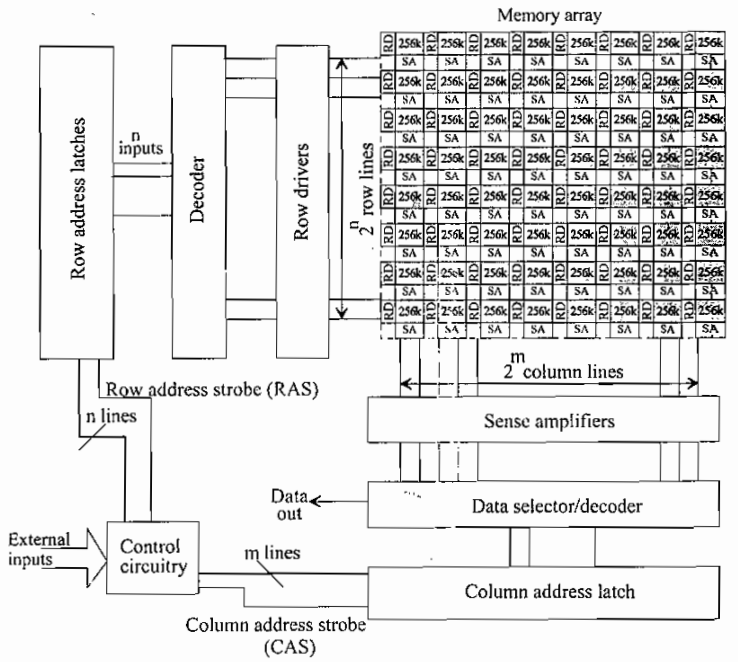


Figure 16.39 Detailed block diagram of a RAM.

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.

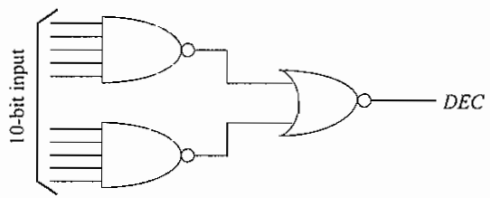


Figure 16.40 A static decoder.

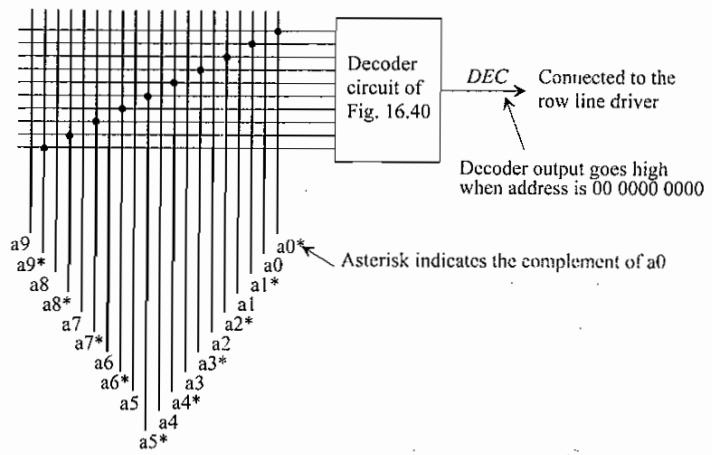


Figure 16.41 How the address lines are connected to a decoder element.

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.

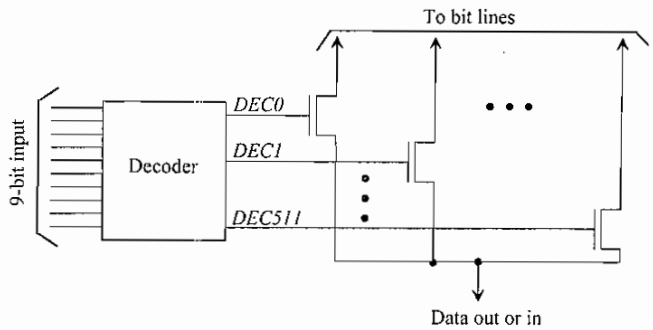


Figure 16.42 Addition of pass transistors to implement a column decoder.

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.

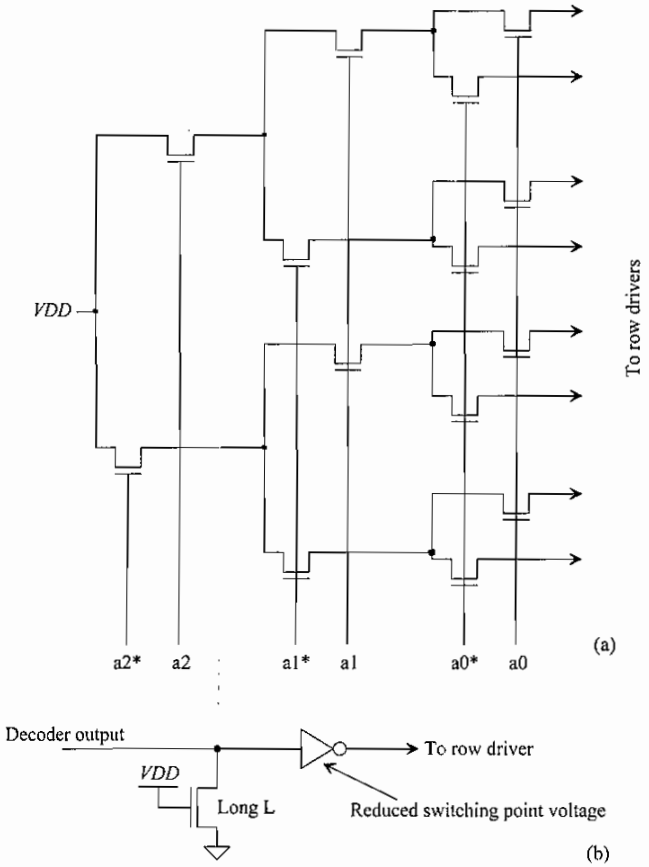


Figure 16.43 A tree decoder used to reduce layout area.

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.

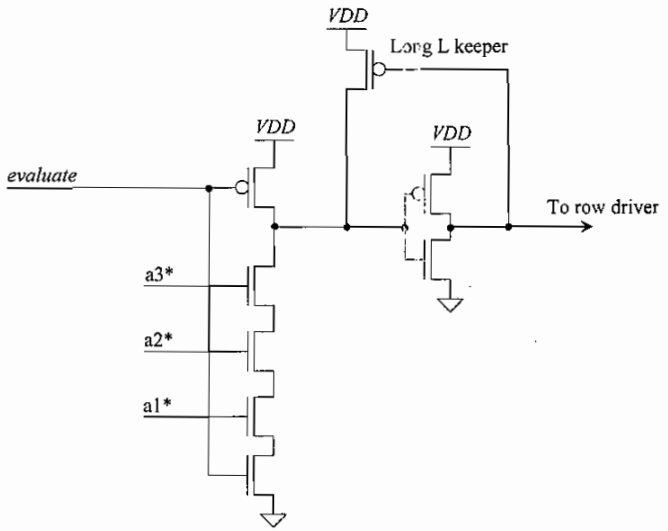


Figure 16.44 3-bit PE decoder.

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.

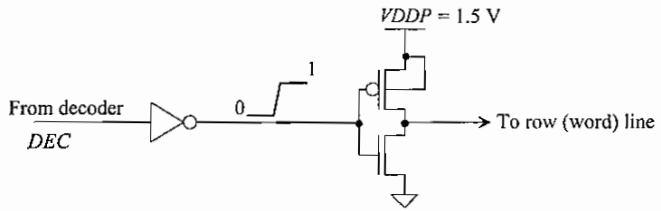


Figure 16.45 Problems with using an inverter for a row driver.

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.

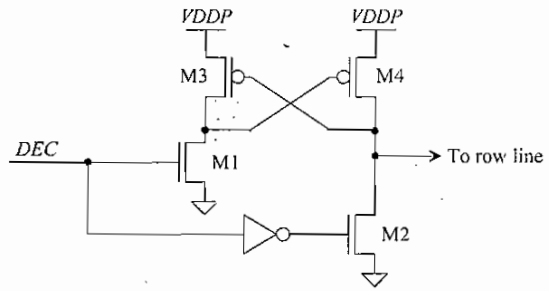


Figure 16.46 A CMOS word line driver.

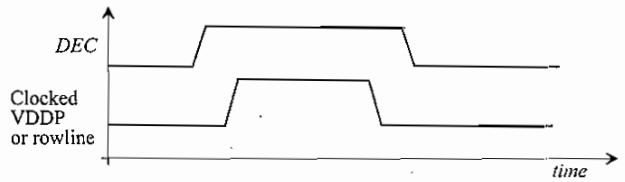


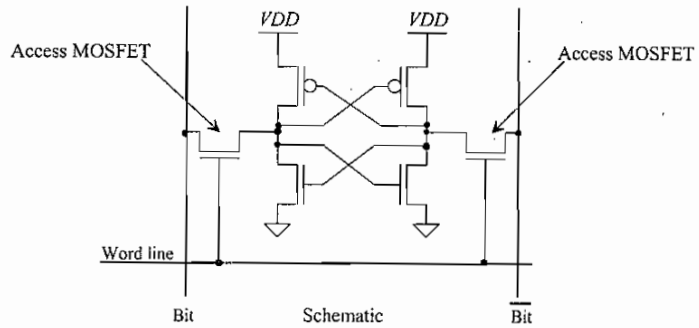
Figure 16.47 Using a clocked VDDP to reduce contention current in a row line driver.

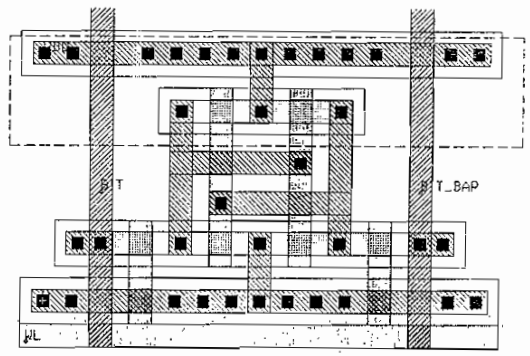
16.3 Memory Cells

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.

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}.

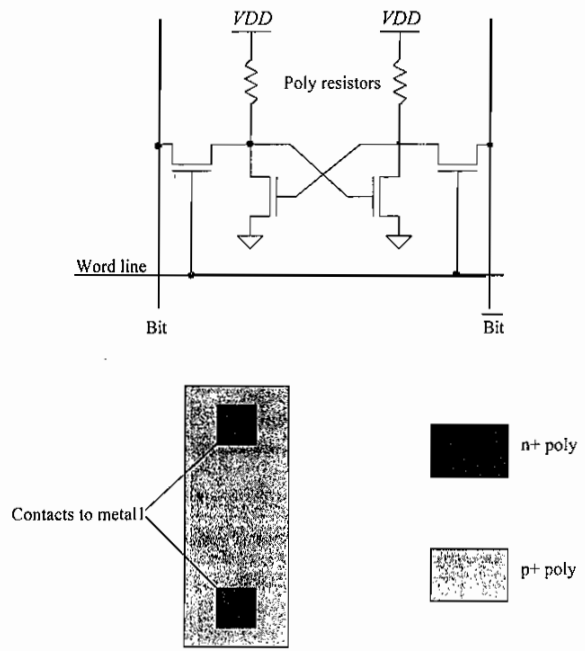


Figure 16.49 SRAM memory cell with poly resistors.

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

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.

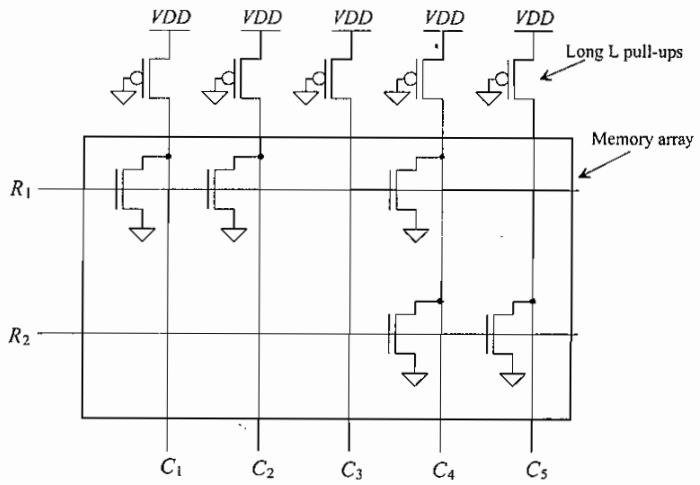


Figure 16.50 A ROM memory array.

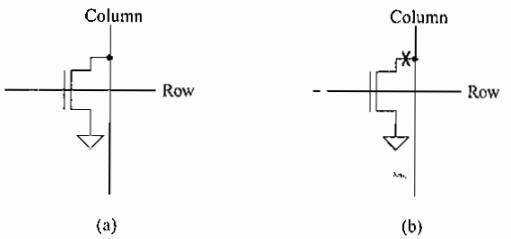


Figure 16.51 (a) n-channel MOSFET at the intersection of every column and row line and (b) eliminating the connection between the drain and column line to program the ROM.

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.

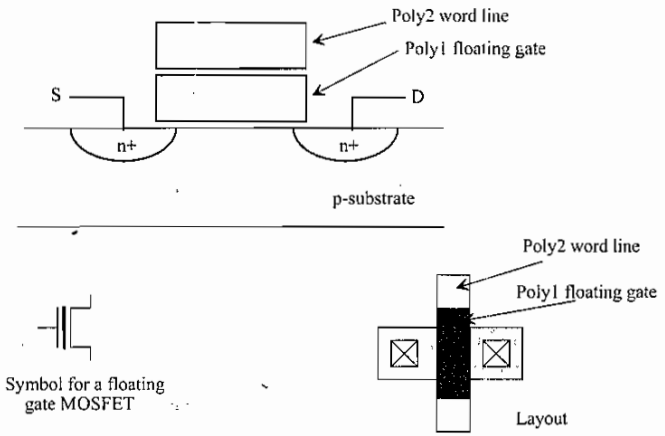


Figure 16.52 A floating gate MOSFET, its symbol and layout.

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).

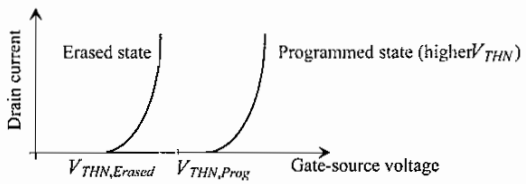


Figure 16.53 Programmed and erased states of a floating gate memory.

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'_i, 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.

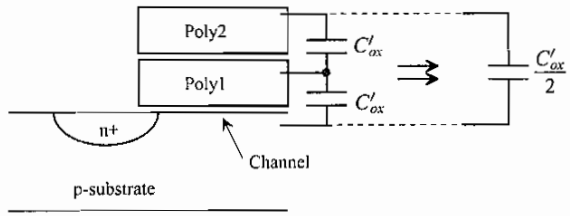


Figure 16.54 Oxide capacitance estimation for calculating threshold voltage..

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

$$V_{THN,Prog} = -V_{ms} - 2V_{fp} + 2 \cdot \left(\frac{Q'_{b0}}{C'_{ox}} + \frac{|Q'_{poly1}|}{C'_{ox}} \right) \tag{16.12}$$

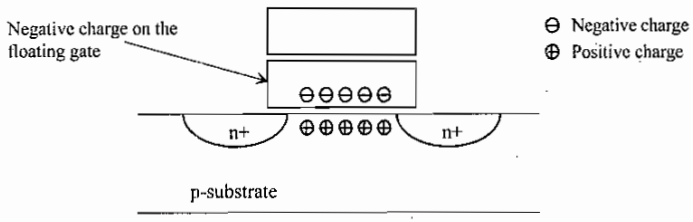


Figure 16.55 Trapped negative charge on the floating gate.

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.

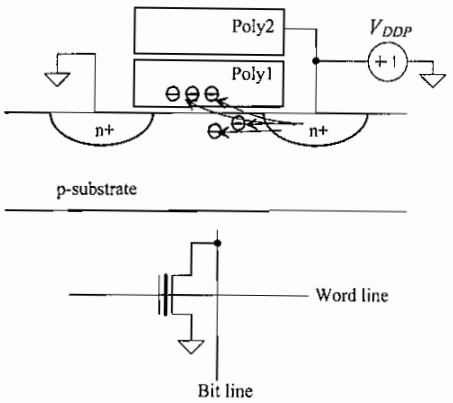


Figure 16.56 How charge is trapped on the floating gate using channel hot electron (CHE) injection.

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.

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).

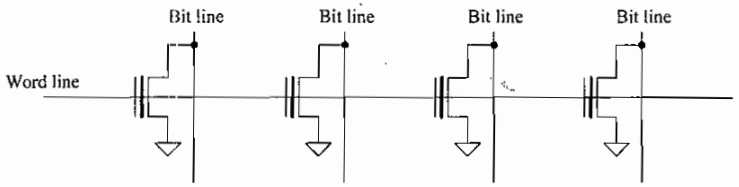


Figure 16.57 Row of floating gate devices. When programming the word line is driven to a high voltage. To program a specific cell, its bit line is also driven to a high voltage. To leave the cell erased, the bit line is held at ground.

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).

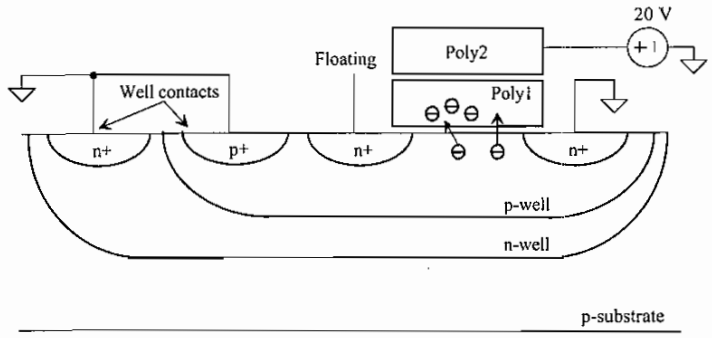


Figure 16.58 FNT of electrons from the p-well to a floating gate to increase threshold voltage (showing programming).

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 ± 3 *V* as typical values).

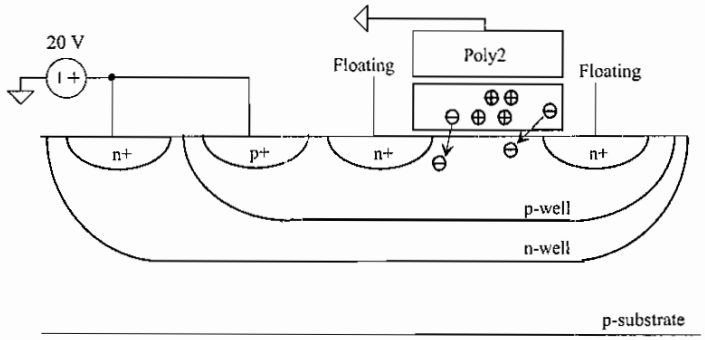


Figure 16.59 FNT of electrons from the floating gate to p-well to decrease threshold voltage (showing erasing).

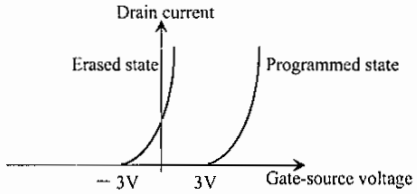


Figure 16.60 Programmed and erased states of a flash memory.

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).

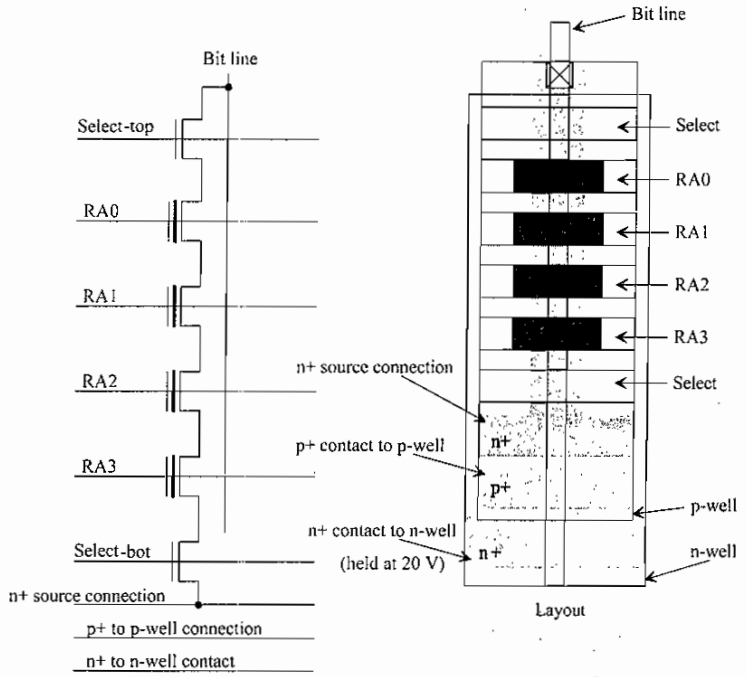


Figure 16.61 A 4-bit Nand Flash memory cell.

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).

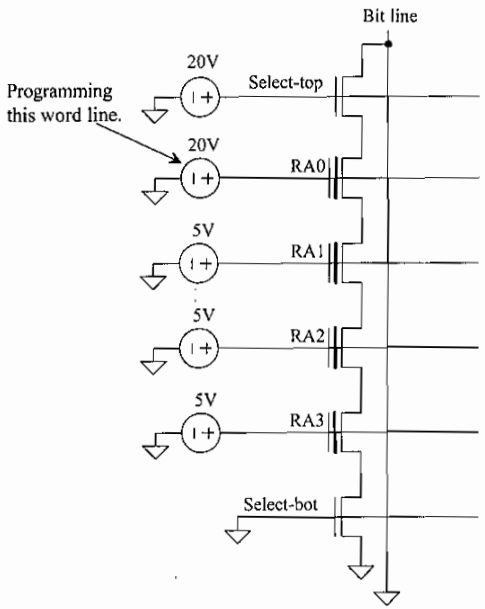


Figure 16.62 Programming in a Flash NAND cell.

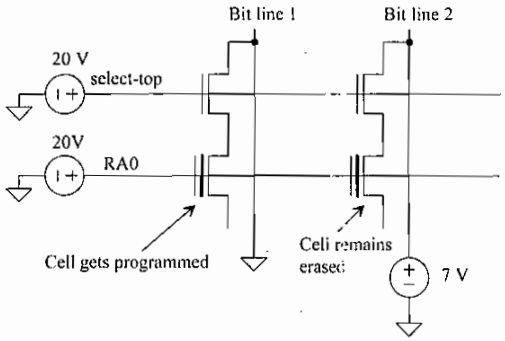


Figure 16.63 How cells are programmed (or not) in a NAND Flash memory array.

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

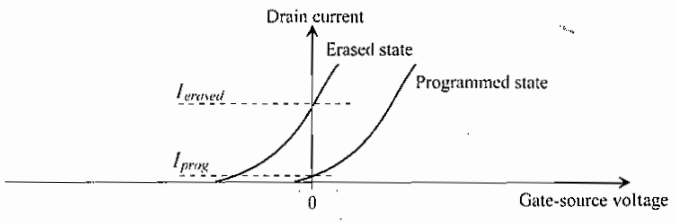


Figure 16.64 Expanded view showing erased and programmed IV curves.

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.

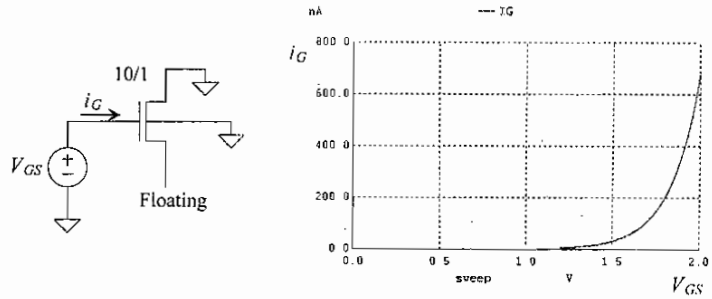


Figure 16.65 How gate current changes with gate voltage with either the gate or source grounded.

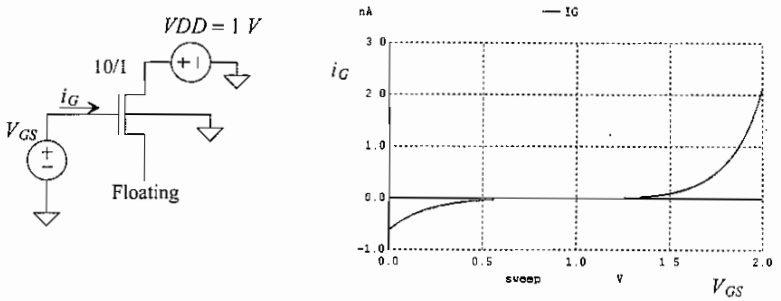


Figure 16.66 How gate current changes with gate voltage when drain is floating and drain is at VDD.

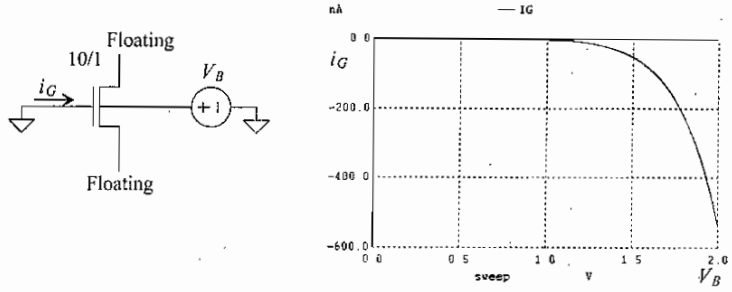


Figure 16.67 Erasing. Since electrons are being removed from the gate (they are tunneling through the gate oxide), we know the gate current will flow out of the device.

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 and R. J. Baker, *DRAM Circuit Design: A Tutorial*, John Wiley and Sons Publishers, 2001. ISBN 0-7803-6014-1

[2] 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

[3] B. Prince, *Semiconductor Memories: A Handbook of Design Manufacture, and Application*, (second edition), John Wiley and Sons Publishers, 1991. ISBN 0-471-92465-2

[4] A. K. Sharma, *Semiconductor Memories: Technology, Testing and Reliability*, John Wiley and Sons Publishers, 1997. ISBN 0-7803-1000-4

[5] K. Itoh, *VLSI Memory Chip Design*, Springer-Verlag Publishers, 2001. ISBN 3-5406-7820-4

[6] T. Hamada, "A Split-Level Diagonal Bit-Line (SLDB) Stacked Capacitor Cell for 256Mb DRAMs," *1992 IEDM Technical Digest*, pp. 799–802.

[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] 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.

[9] 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.

[10] M. I. Elmasry, *Digital MOS Integrated Circuits II*, IEEE Press, 1992. ISBN 0-87942-275-0

[11] R. D. Pashley and S. K. Lai, "Flash Memories: The Best of Two Worlds," *IEEE Spectrum*, 1989, pp. 30–33.

[12] D. Frohman-Bentchkowsky, "FAMOS-A New Semiconductor Charge Storage Device," *Solid-State Electronics*, vol. 17, pp. 517–529, 1974.

[13] 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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] J. Ramírez-Angulo, "Ultracompact Low-voltage Analog CMOS Multiplier Using Multiple Input Floating Gate Transistors,". 1996 European Solid State Circuits Conference, pp. 99–103.

[19] 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.

[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] 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.

[22] 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.

[23] 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.

[24] 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.

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.

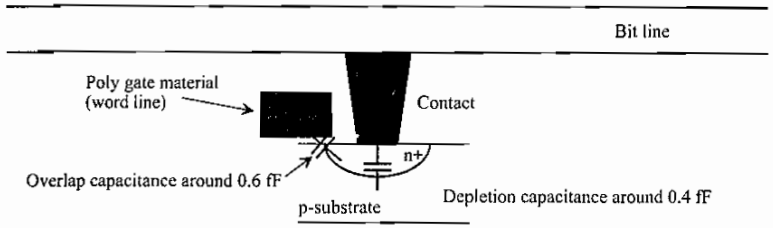


Figure 16.68 How the gate-drain overlap capacitance loads the bit line. Note that this cross-sectional view is rotated 90 degrees from the one seen in Fig. 16.3.

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.

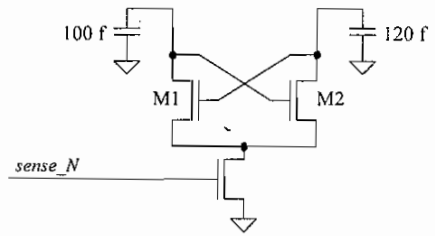


Figure 16.69 Problem 16.2 showing how a mismatch in the load capacitance of an NSA can result in in sensing errors.

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 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.

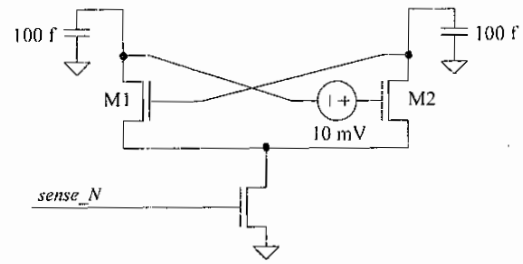


Figure 16.70 Circuit for Problem 16.3.

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.

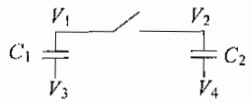


Figure 16.71 Circuit for Problem 16.6.

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.

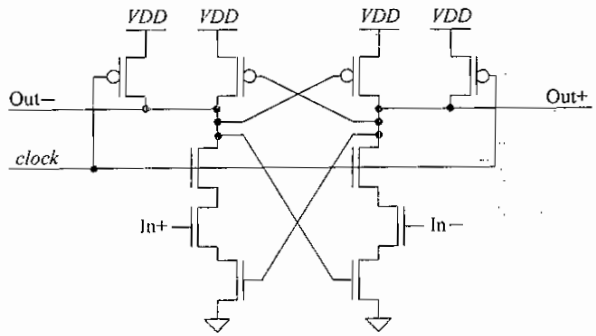


Figure 16.72 An alternative clocked sense amplifier.

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.

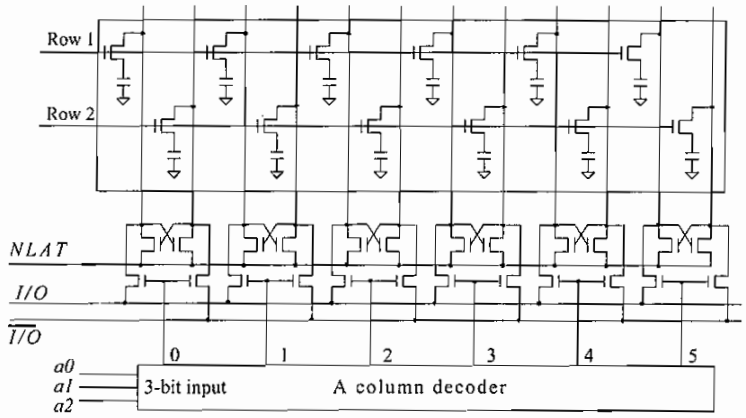


Figure 16.73 Design of a column decoder for problem 16.8.

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?

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 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.

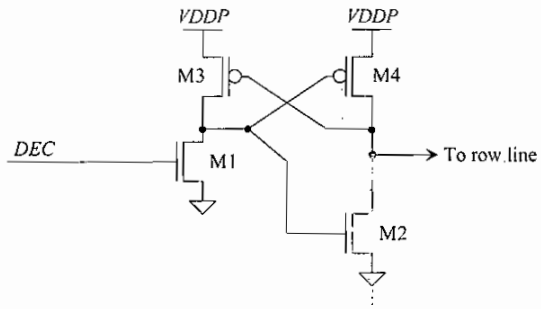


Figure 16.74 A (bad) CMOS word line driver, Problem 16.12.

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?

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.

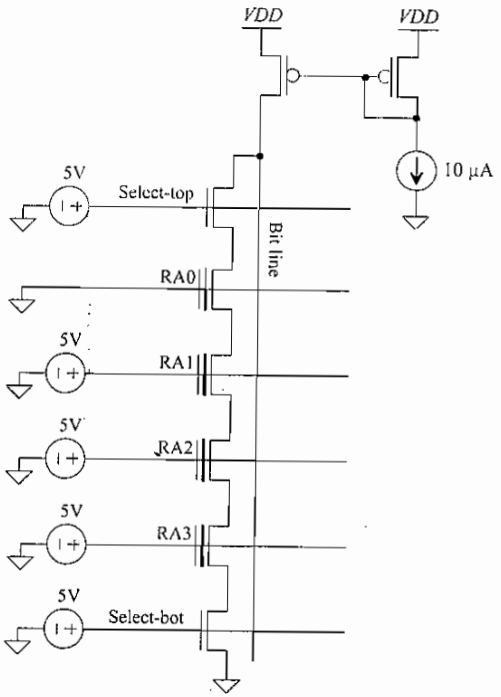


Figure 16.75 Reading the contents of RA0 in a NAND Flash cell.

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.

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).

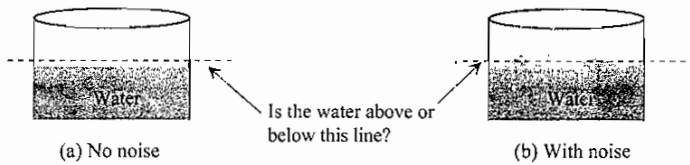


Figure 17.1 Using a water analogy to illustrate the problems with sensing.

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).

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.

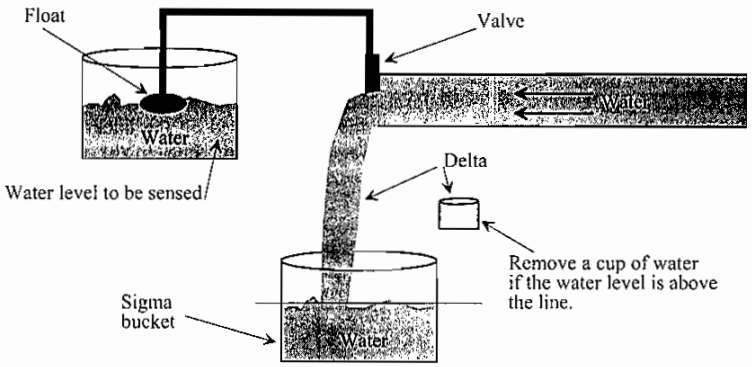


Figure 17.2 Changing the water height (pressure or voltage) into a water flow (current) using the float and valve. The delta comes from the difference in the number of cups we remove from the bucket with the water we add to the bucket. The sigma (sum) is the storage of the difference in the bucket.

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

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

$$\text{Input signal} = \text{Ouput number} \times \text{fed back signal size} \qquad (17.2)$$

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.

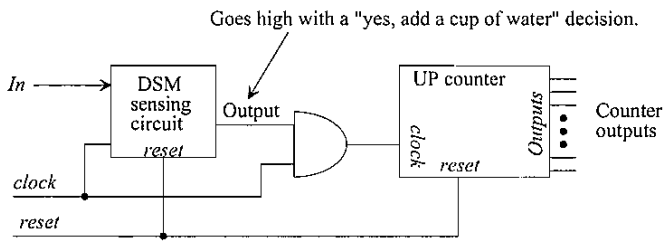


Figure 17.3 How a counter is used to average the outputs of a DSM sensing circuit.

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.

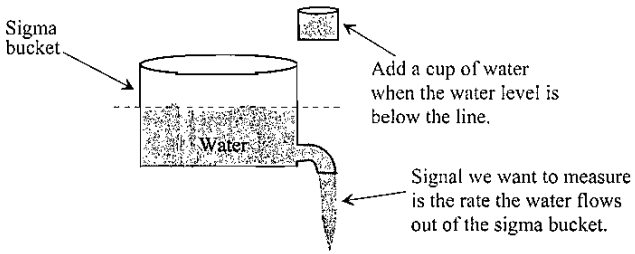


Figure 17.4 Using DSM to measure the water leaving a bucket.

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.

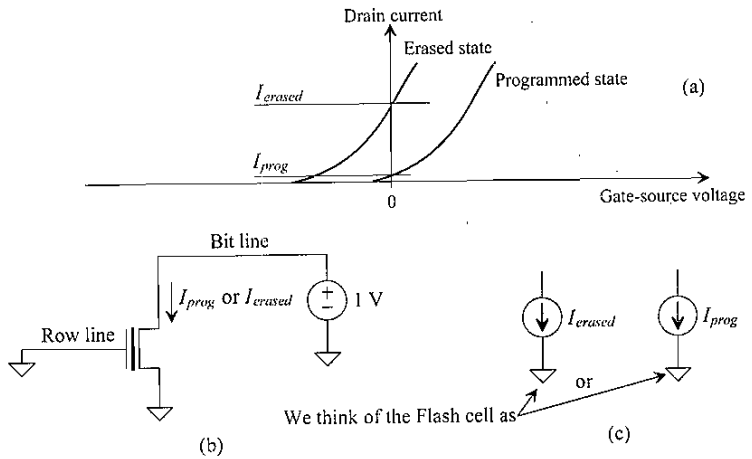


Figure 17.5 The IV characteristics of a NAND memory cell and how we think of the cell when sensing.

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,

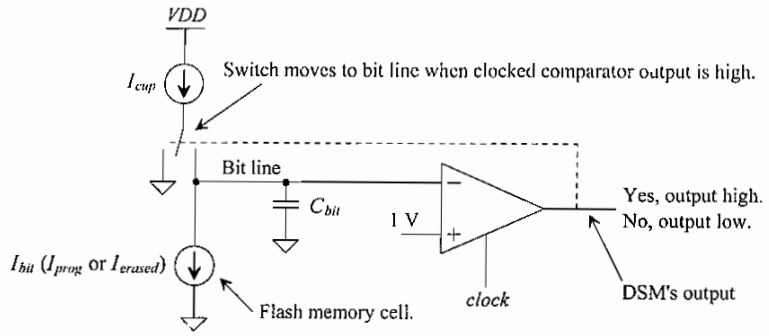


Figure 17.6 Sensing a Flash memory cell using DSM.

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 ΔV_{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

$$\frac{I_{bit}}{I_{cup}} = \frac{M}{N} \tag{17.7}$$

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

$$\Delta V_{bit,\max} = \frac{9\mu A \cdot 10\ ns}{500\ fF} = 180\ mV$$

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.

The maximum input signal, $I_{bit,\max}$, equals the fed-back signal I_{cmp} ($I_{cmp} \geq I_{bit}$). 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)

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.

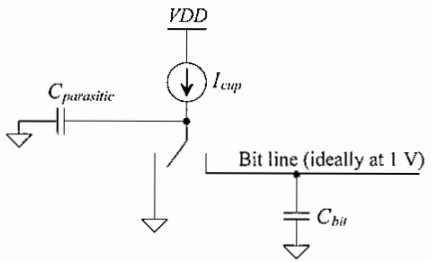


Figure 17.7 How the parasitic capacitance on the output of the current source causes errors.

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, ϕ_1 and ϕ_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 ϕ_1 is low, C_{cup} charges to *VDD*. Next, ϕ_1 goes high. Between ϕ_1 going high and ϕ_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

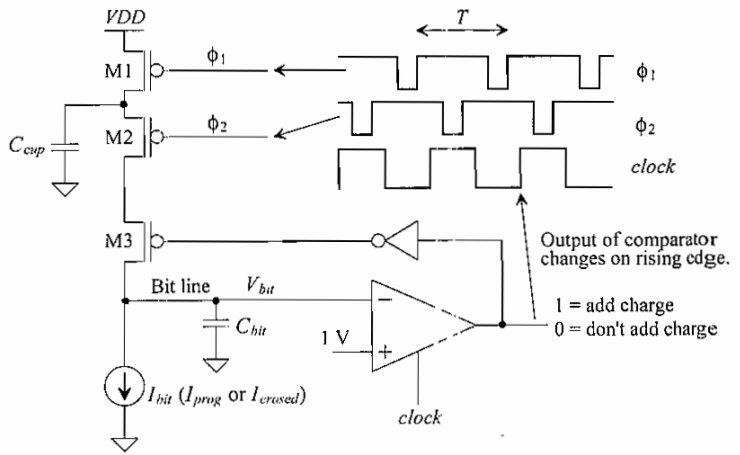


Figure 17.8 Using a switched-capacitor to add charge to the bit line.

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 ϕ_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 ϕ_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, $V_{bit,min} > V_{THP}$).

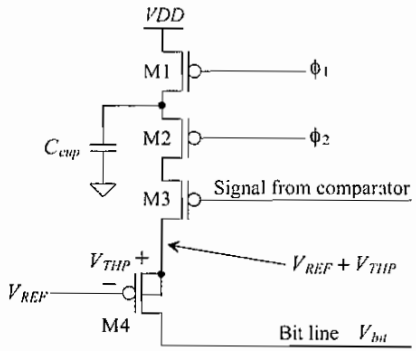


Figure 17.9 Adding a MOSFET to set the swing across the capacitor.

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 (ϕ_1 or ϕ_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

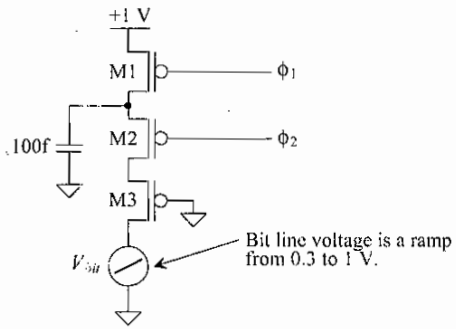


Figure 17.10 Verifying Eq. (17.11), see Ex. 17.3.

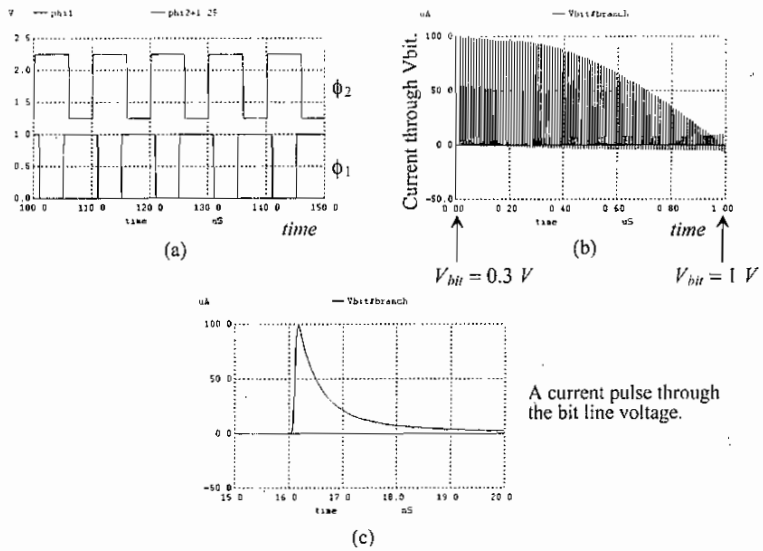


Figure 17.11 The operation of the circuit in Fig. 17.10.

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

$$Q_{cup} = 100\,fF \cdot (1 - 0.3) = 70\,fC \qquad (17.14)$$

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

$$V_{REF} < VDD - V_{THP} \qquad (17.15)$$

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. 17.11a, 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. ■

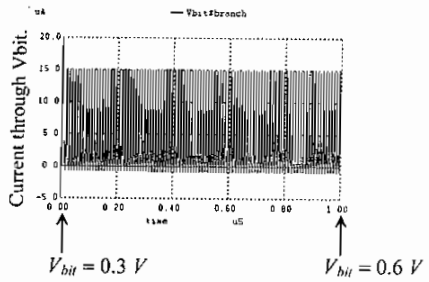


Figure 17.12 How the charge supplied to Vbit becomes linear if we add M4, from Fig. 17.9, to the simulation in Fig. 17.10. Note the reduced bit line voltage swing and the reduction in the current pulse amplitudes.

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 ϕ_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).

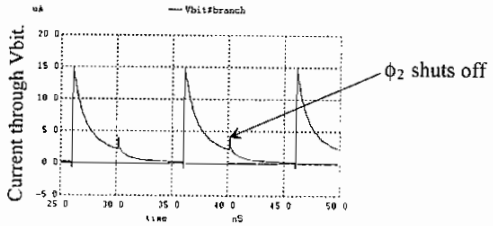


Figure 17.13 The incomplete settling in Fig. 17.12.

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).

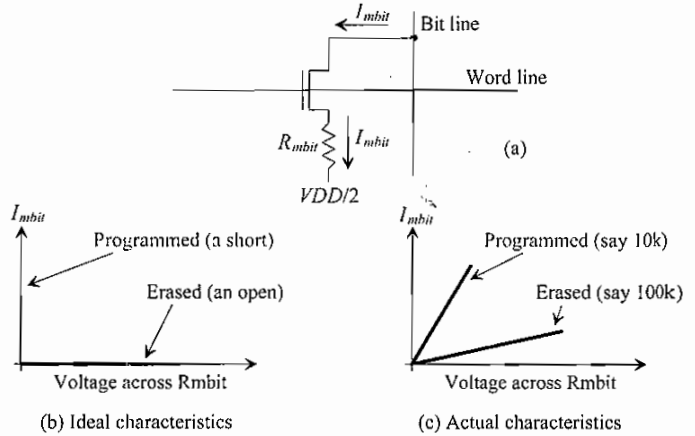


Figure 17.14 The one-transistor, one-resistor (1T1R) RAM memory cell.

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

$$I_{mbit} = \frac{V_{os}}{R_{mbit}} \qquad (17.16)$$

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).

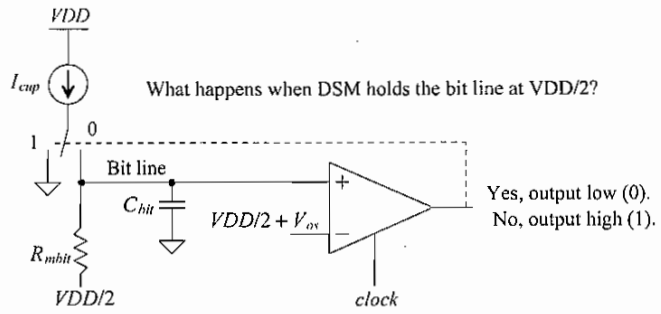


Figure 17.15 Sensing a resitive memory cell using DSM. Notice that we've eliminated the inverter from Fig. 17.8 by switching the input terminals of the comparator.

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 ϕ_2, remove the NAND gates on the output, and combine M2 and M3 in Fig. 17.9. In a

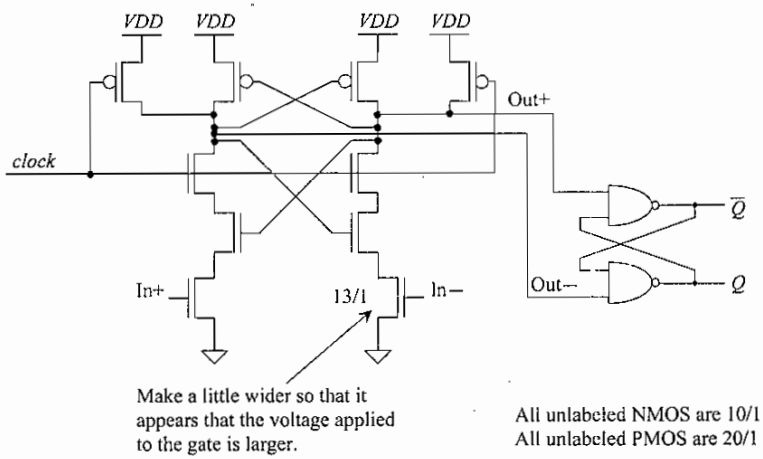


Figure 17.16 Designing a clocked comparator with a built-in offset.

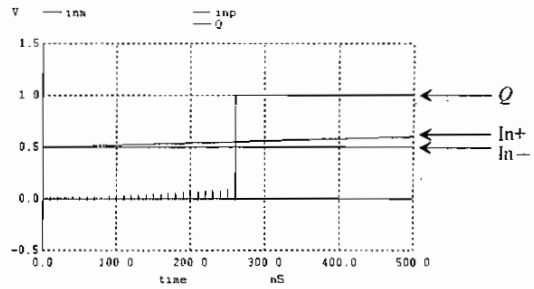


Figure 17.17 The output of the comparator switching states when the + input is 50 mV above the - input.

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

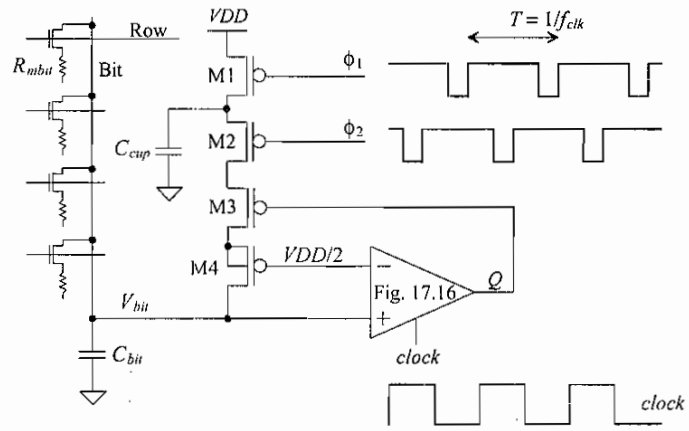


Figure 17.18 Sensing circuit for resistive memory.

current through the cell is then given by Eq. (17.15). 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 high (M is always a 1). If R_{mbit} is infinite, then the output of the DSM is always low (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 ΔV_{bit}, we can write (knowing charge must be conserved)

$$C_{cup} \cdot (VDD - VDD/2 - V_{THP}) = \Delta V_{bit} \cdot (C_{bit} + C_{cup}) \qquad (17.21)$$

or

$$\Delta V_{bit} = \frac{C_{cup}}{C_{cup} + C_{bit}} \cdot (VDD/2 - V_{THP}) \qquad (17.22)$$

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.

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 $R_{mbit} = 25$k, 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),

$$R_{mbit} = 25k \cdot \frac{50}{10} = 125k \text{ (actual value 100k)}$$

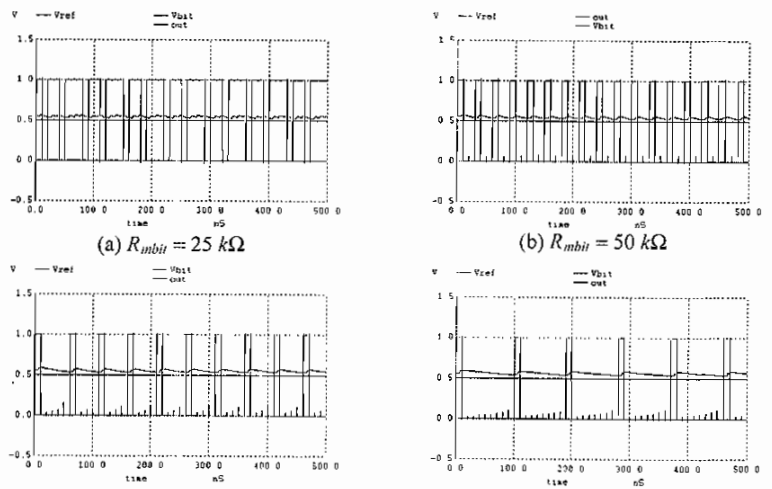
$$R_{mbit} = 25k \cdot \frac{50}{6} = 208k \text{ (actual value 200k)}$$

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

$$R_{mbit,\max} = \frac{V_{OS} \cdot T \cdot N}{(VDD/2 - V_{THP}) \cdot C_{cup}} \ (= 1.25\text{M here}) \qquad (17.23)$$

Again, the minimum resistance is

$$R_{mbit,\min} = \frac{V_{OS} \cdot T}{(VDD/2 - V_{THP}) \cdot C_{cup}} \ (= 25\text{k here}) \qquad (17.24)$$



(a) $R_{mbit} = 25\ k\Omega$ (b) $R_{mbit} = 50\ k\Omega$

(c) $R_{mbit} = 100\ k\Omega$ (d) $R_{mbit} = 200\ k\Omega$

Figure 17.19 Outputs of the DSM for various Rmbit values.

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

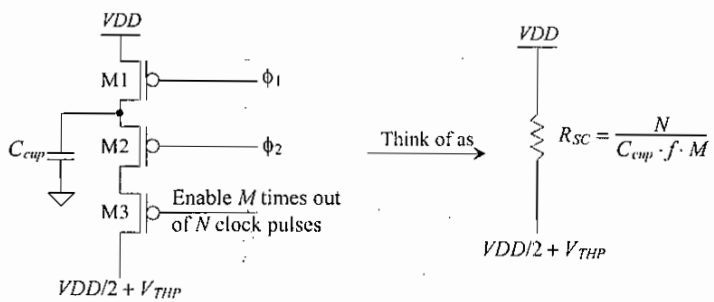


Figure 17.20 How a switched-capacitor resistor is modeled.

$$I_{avg} = \frac{Q_{cup}}{T} \cdot \frac{M}{N} = Q_{cup} \cdot f \cdot \frac{M}{N} \tag{17.27}$$

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

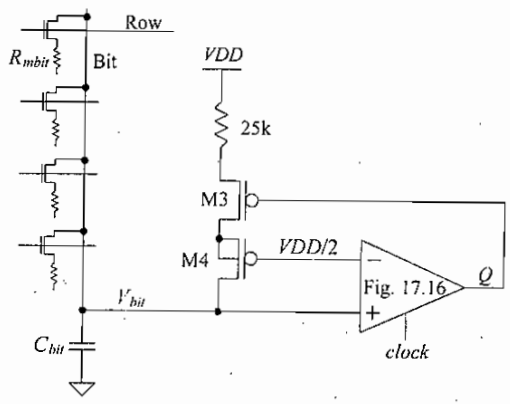
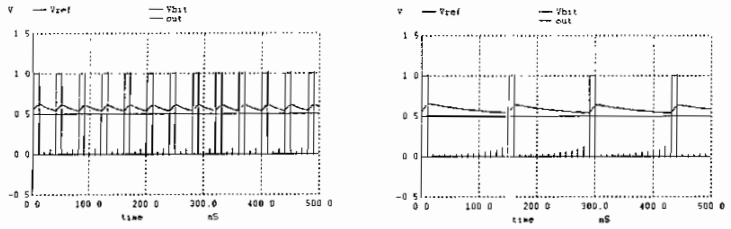


Figure 17.21 Simpler DSM for sensing resistive memory.



(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

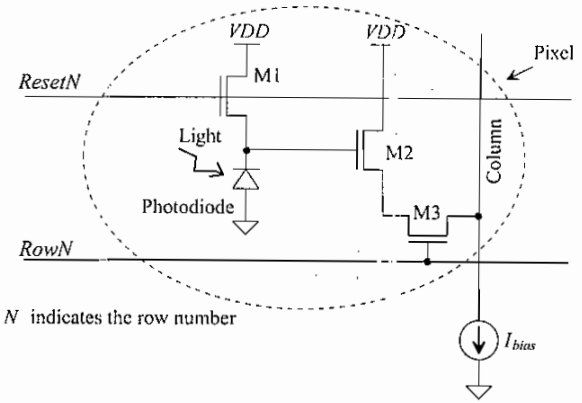


Figure 17.23 A CMOS active pixel sensor (APS).

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

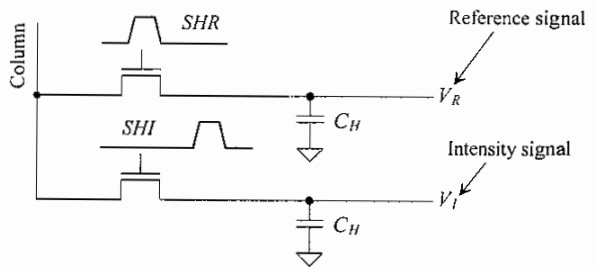


Figure 17.24 Sampling the reference and intensity signals.

of the light) goes 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 second volume of this 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 capacitor results in a voltage signal containing $1/f^3$ noise. The ramp, with noise, voltage

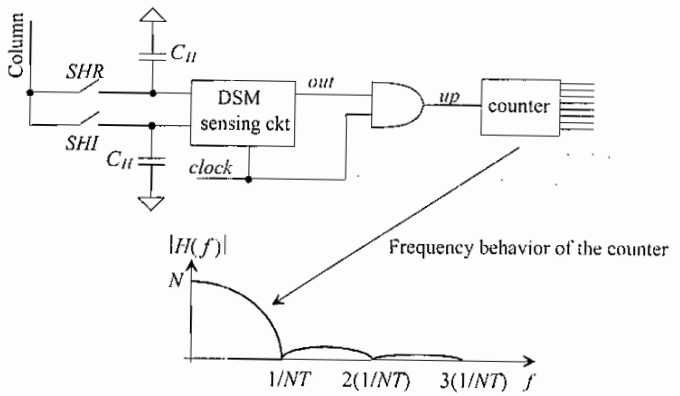


Figure 17.25 Thinking of the counter as a digital low pass filter

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.

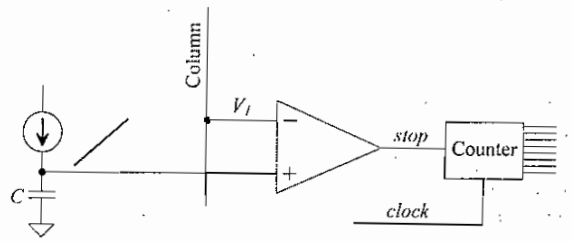


Figure 17.26 A circuit where the counter doesn't behave like a lowpass filter.

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.

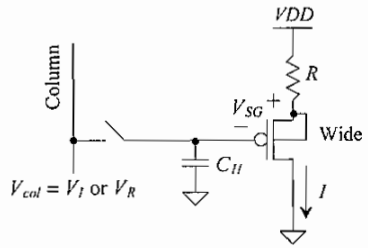


Figure 17.27 Linear voltage-to-current conversion.

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. ■

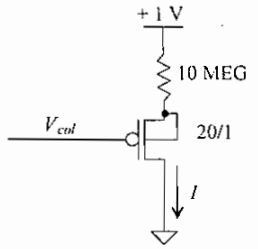
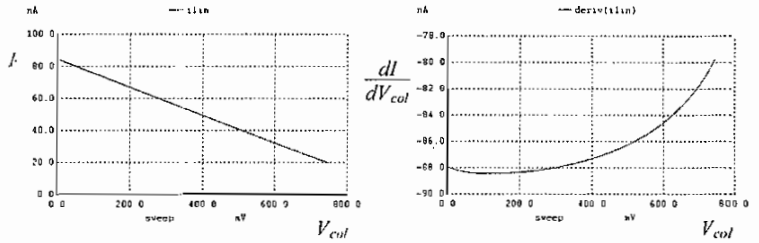


Figure 17.28 Circuit used in Ex. 17.6.



(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).

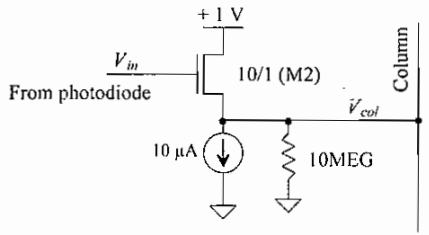


Figure 17.30 Looking at the linearity of the source follower in the pixel.

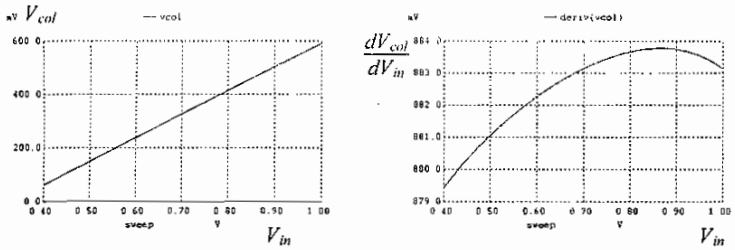


Figure 17.31 Showing the transfer curve and linearity of the circuit in Fig. 17.30.

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

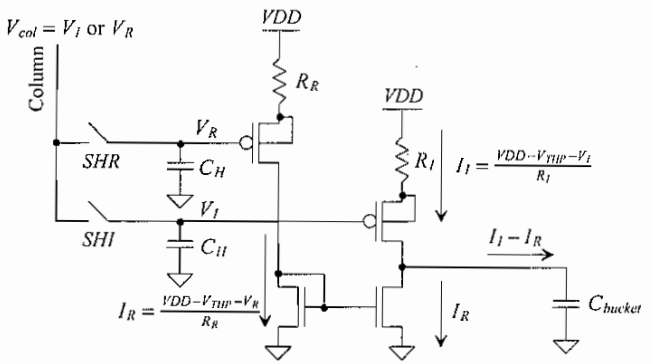


Figure 17.32 Subtracting the currents.

$$V_{I,shift} = \frac{R_I}{R_R} \cdot V_{R,shift} \tag{17.35}$$

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)

$$R_R = \frac{1}{f \cdot C_{cup}} \tag{17.37}$$

and for R_I (which *will* be enabled via M3 when its gate goes low)

$$R_I = \frac{1}{f \cdot C_{cup} \cdot \frac{\overline{M}}{N}} \tag{17.38}$$

Rewriting Eq. (17.35), knowing $\overline{M} = N - M$, gives

$$V_{I,shift} = V_{R,shift} \cdot \frac{N}{N - M} \tag{17.39}$$

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.

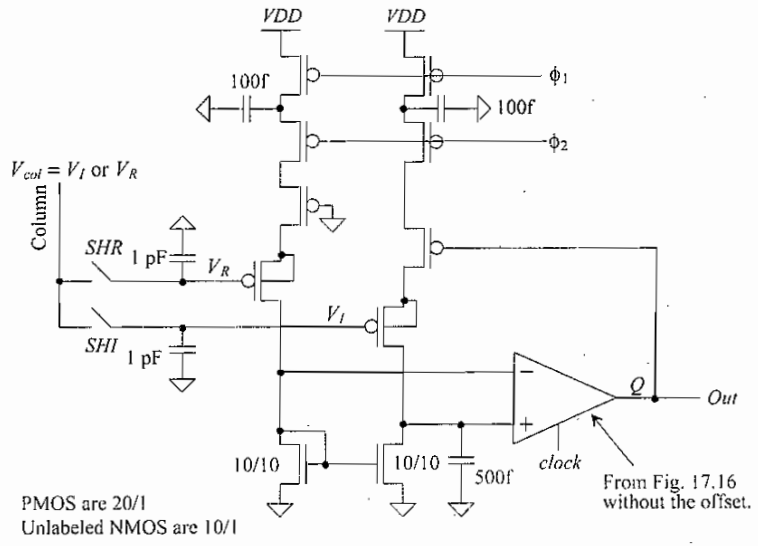
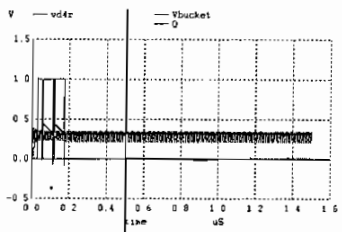


Figure 17.33 Schematic of a DSM for sensing in CMOS imaging chips.

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 reference and intensity signals are sampled onto the hold capacitors. If V_R = 650 mV, then

$$V_{R,shift} \approx 1 - 0.25 - 0.65 = 100\ mV$$



(a)

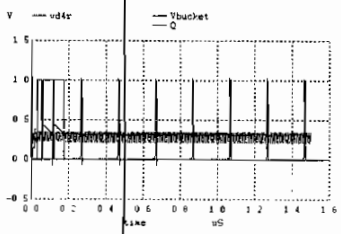
$V_R = 650\ mV \qquad V_I = 650\ mV$

$V_{R,shift} = VDD - V_{THP} - V_R = 100\ mV$

$V_{I,shift} = VDD - V_{THP} - V_I = 100\ mV$

$$V_{I,shift} = 100\ mV \cdot \frac{100}{100-0} = 100\ mV$$

(sensed)



(b)

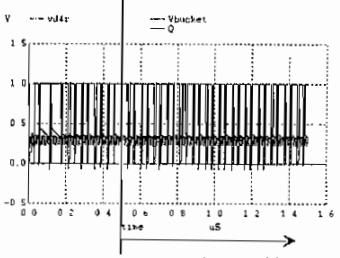
$V_R = 650\ mV \qquad V_I = 645\ mV$

$V_{R,shift} = VDD - V_{THP} - V_R = 100\ mV$

$V_{I,shift} = VDD - V_{THP} - V_I = 105\ mV$

$$V_{I,shift} = 100\ mV \cdot \frac{100}{100-5} = 105\ mV$$

(sensed)



(c)

start sensing at 500 ns

$V_R = 650\ mV \qquad V_I = 400\ mV$

$V_{R,shift} = VDD - V_{THP} - V_R = 100\ mV$

$V_{I,shift} = VDD - V_{THP} - V_I = 350\ mV$

$$V_{I,shift} = 100\ mV \cdot \frac{100}{100-72} = 357\ mV$$

(sensed)

Figure 17.34 How the DSM sensing circuit in Fig. 17.33 operates.

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

$$V_{I,shift} = V_{R,shift} \cdot \frac{N-M}{M} \tag{17.42}$$

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 $N/2 \leq M \leq N$ (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.

Note that it's a bad idea to ground the gate of M31 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.

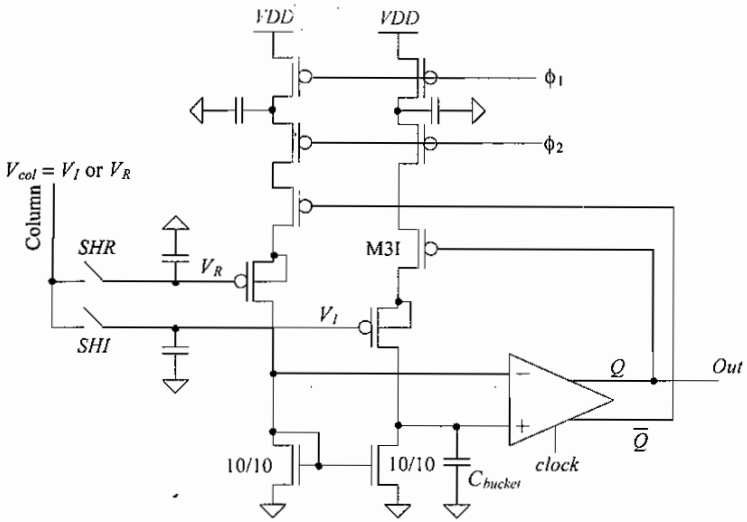


Figure 17.35 Using both comparator outputs for a DSM sensing circuit (bad if $V_R \geq V_I$).

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 ϕ_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).

In Fig. 17.36 we show a comparator design using PMOS imbalance MOSFETs (as seen in Fig. 16.32 for the NMOS flavor). When ϕ_2 goes high, the outputs of the comparator, *Out* and $\overline{Out}$ are driven low. When ϕ_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

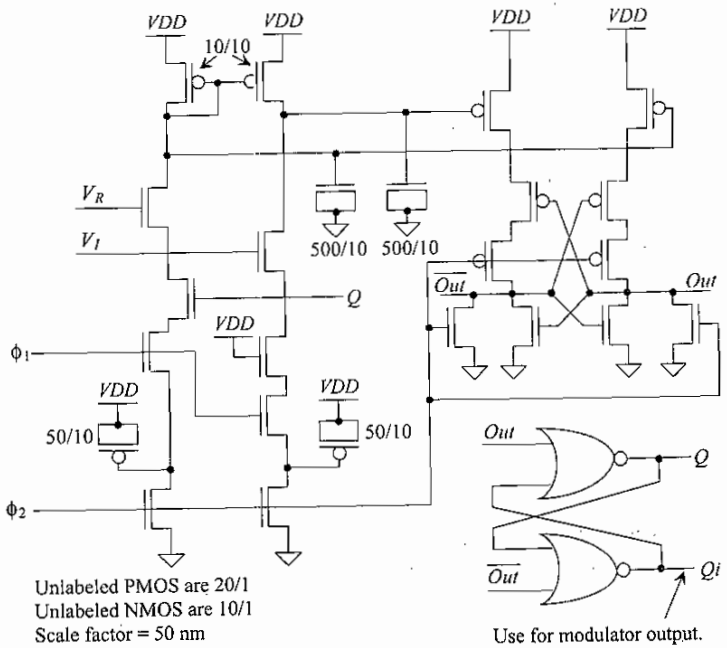


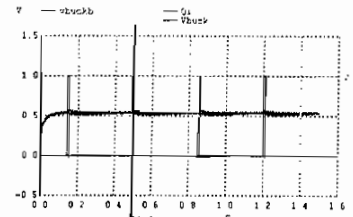
Figure 17.36 DSM circuit for sensing in a CMOS imaging chip.

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

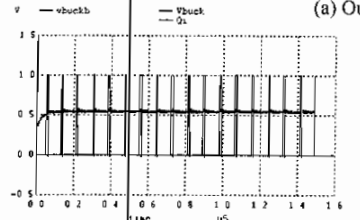
$$V_{res} = \frac{V_{R,shift}}{N} \rightarrow 4\ mV \qquad (17.43)$$

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,



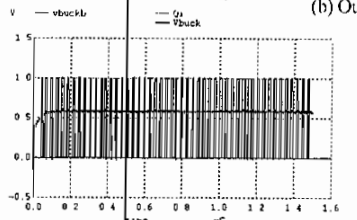
$V_R = 650\ mV \quad V_I = 640\ mV$
$V_{R,shift} = V_R - V_{THN} = 400\ mV$
$V_{I,shift} = V_I - V_{THN} = 390\ mV$
$V_{I,shift} = 400\ mV \cdot \frac{98}{100} = 392\ mV$ (sensed)

(a) Output goes high 2 times out of 100 (M = 98).



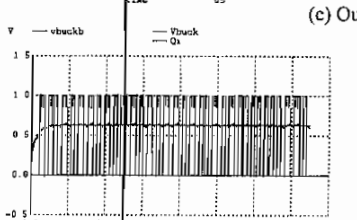
$V_R = 650\ mV \quad V_I = 600\ mV$
$V_{I,shift} = V_I - V_{THN} = 350\ mV$
$V_{I,shift} = 400\ mV \cdot \frac{88}{100} = 352\ mV$ (sensed)

(b) Output goes high 12 times out of 100 (M = 88).



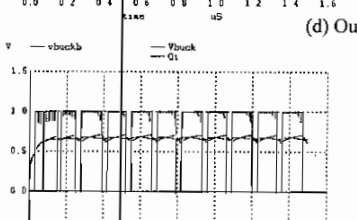
$V_R = 650\ mV \quad V_I = 500\ mV$
$V_{I,shift} = V_I - V_{THN} = 250\ mV$
$V_{I,shift} = 400\ mV \cdot \frac{65}{100} = 264\ mV$ (sensed)

(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.

$$V_{res} = \frac{100\ mV}{23} = 4.35\ mV \tag{17.44}$$

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 thermal noise from the capacitors 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

$$V_{onoise,RMS} = \sqrt{\frac{kT}{NC}} \tag{17.46}$$

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 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 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.

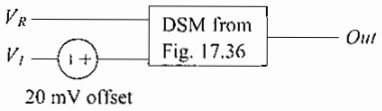


Figure 17.38 Modeling the differences in the signal paths in the DSM of Fig. 17.36 with a voltage in series with the intensity signal path.

To illustrate this problem, let's resimulate the DSM in Fig. 17.33 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.

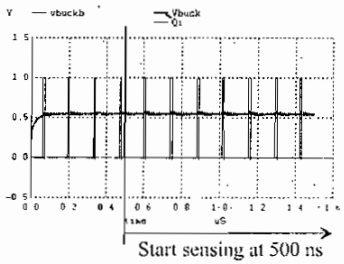


Figure 17.39 How a 20 mV offset changes the outputs in Fig. 17.37a.

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. 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

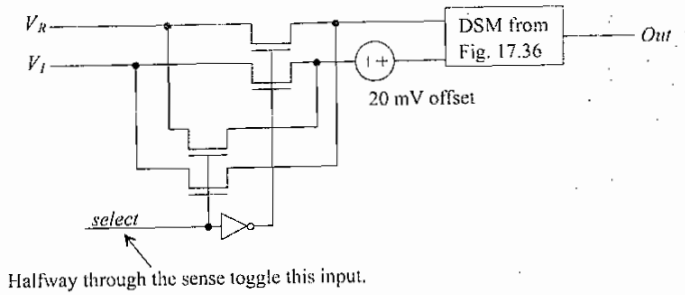


Figure 17.40 Switching the inputs of a DSM to eliminate path mismatch.

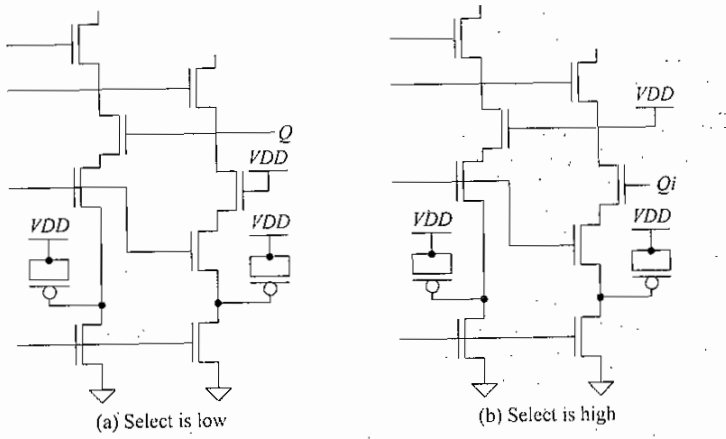


Figure 17.41 The change in the feedback with change in select.

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).

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

See Volume 2 of this book entitled *CMOS Mixed-Signal Circuit Design*, John Wiley and Sons, 2002. ISBN 0-471-22754-4

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 that the operation of 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?

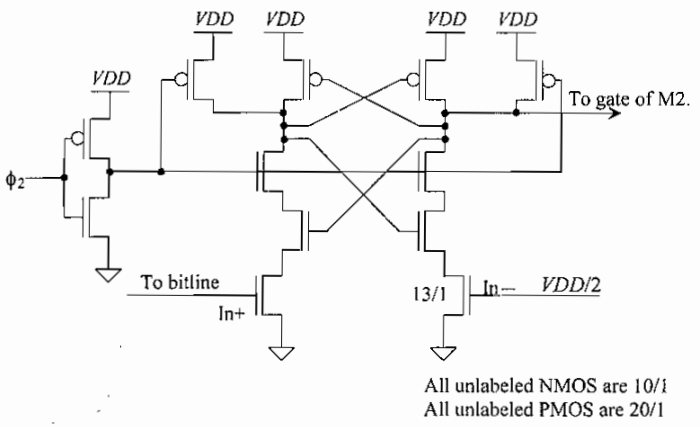


Figure 17.42 Simplifying the comparator in Figs. 17.16 and 17.18.

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 following circuit, 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 (the WinSPICE command "print mean(vdd# branch)" can be used to get the average current supplied by *VDD*).

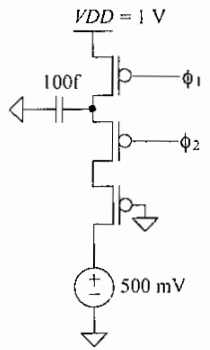


Figure 17.43 Determining the average current that flows in a switched-capacitor resistor. See Problem 17.11.

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 ϕ_2 goes high so that $\overline{Out}$ can be used to clock a counter directly.

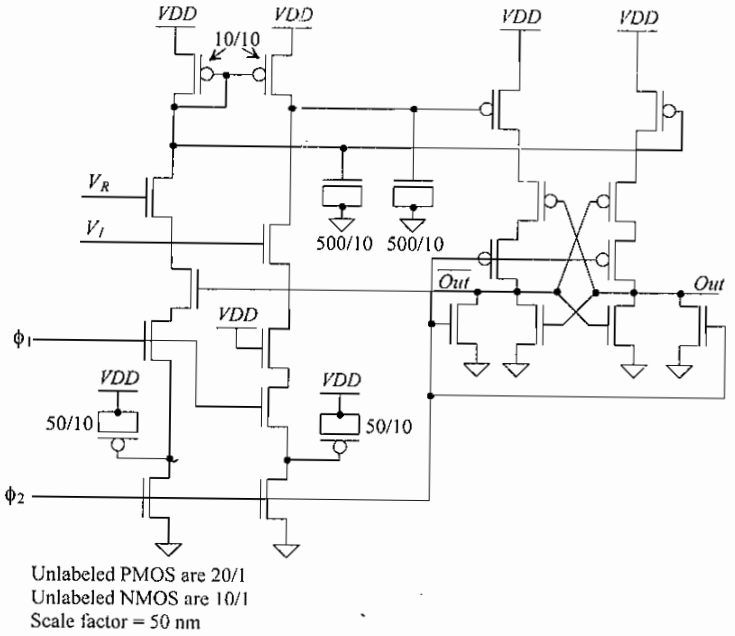


Figure 17.44 Simplifying the DSM sensing circuit, see Problem 17.13.

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.

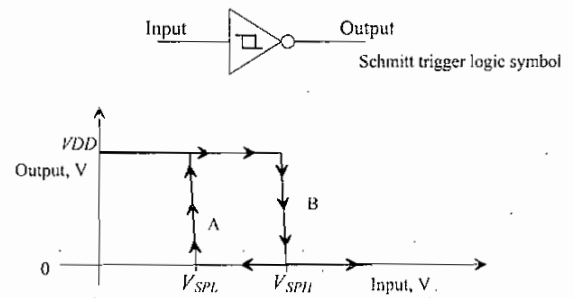


Figure 18.1 Transfer characteristics of a Schmitt trigger.

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

$$V_H = V_{SPH} - V_{SPL} \tag{18.1}$$

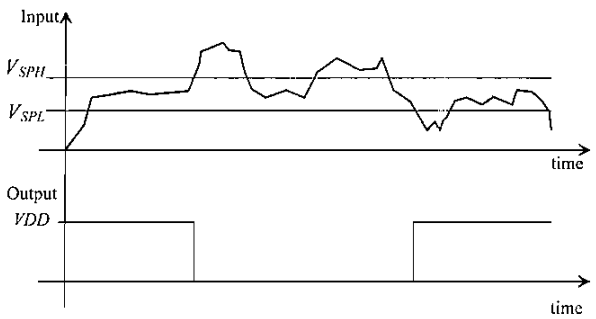


Figure 18.2 Input, top trace, and output of a Schmitt trigger.

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*.

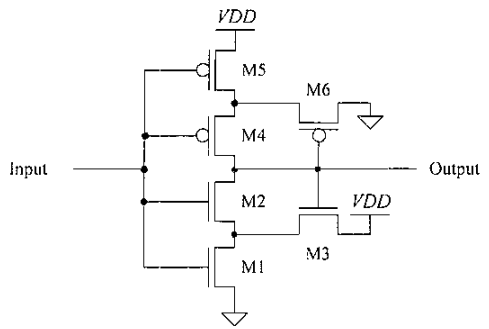


Figure 18.3 Schematic of the Schmitt trigger.

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.

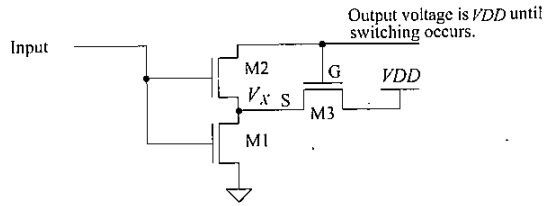


Figure 18.4 Portion of the Schmitt trigger schematic used to calculate upper switching point voltage.

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, β_2, is to require that

$$\beta_2 \geq \beta_1 \text{ or } \beta_3 \tag{18.5}$$

since M2 is used as a switch.

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). ■

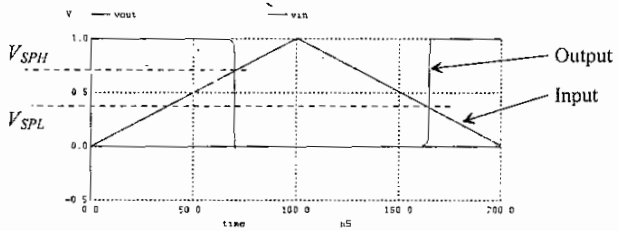


Figure 18.5 The input and output of the Schmitt trigger designed in Ex. 18.1.

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

$$t_{PLH} = 0.7 \cdot (R_{p4} + R_{p5}) \cdot C_{load} \tag{18.8}$$

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.

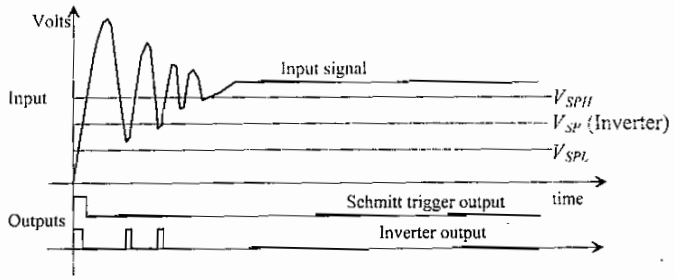


Figure 18.6 Applying a Schmitt trigger to clean up an interconnecting signal.

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

$$V_c(t) = V_{SPL} + (VDD - V_{SPL})\left(1 - e^{\frac{-t}{RC}}\right) \tag{18.11}$$

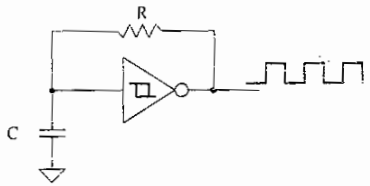


Figure 18.7 Oscillator design using a Schmitt trigger.

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.

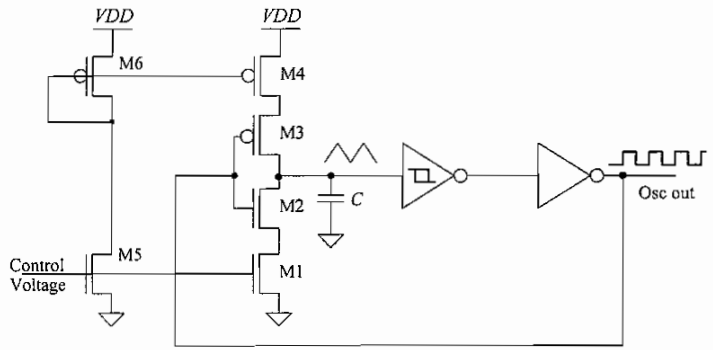


Figure 18.8 Voltage-controlled oscillator using Schmitt trigger and current sources. MOSFETs M2 and M3 are used as switches.

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.

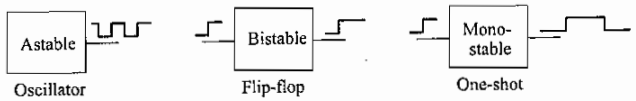


Figure 18.9 Multivibrator circuits.

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

$$V_c(t) = V_2(t) - \overbrace{V_1(t)}^{=0} = VDD \cdot (1 - e^{\frac{-t}{RC}}) \tag{18.16}$$

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

$$t = RC \cdot \ln \frac{VDD}{VDD - V_{SP}} = RC \cdot \ln(2) \approx 0.7RC \tag{18.17}$$

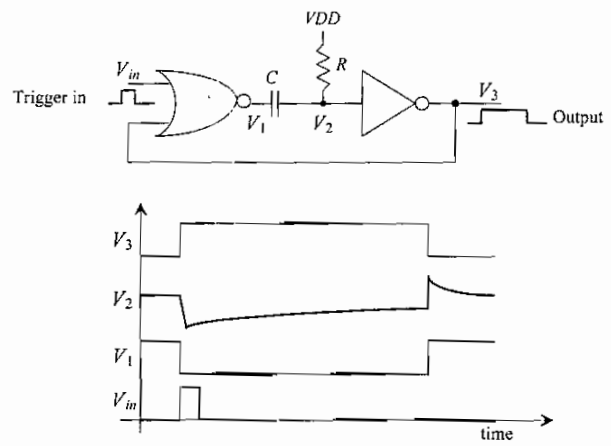


Figure 18.10 Operation of the monostable multivibrator.

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

$$t_1 = t_2 = 1.1RC \qquad (18.20)$$

and the frequency of oscillation is

$$f_{osc} = \frac{1}{t_1 + t_2} = \frac{1}{2.2RC} \qquad (18.21)$$

Again, if the resistor and capacitor are bonded out, the ESD diodes on the pads will limit the voltage swing of V_1.

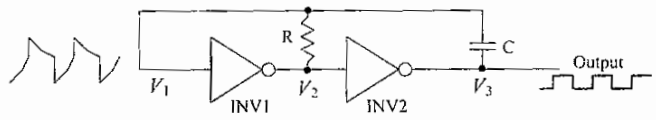


Figure 18.11 An astable multivibrator.

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.

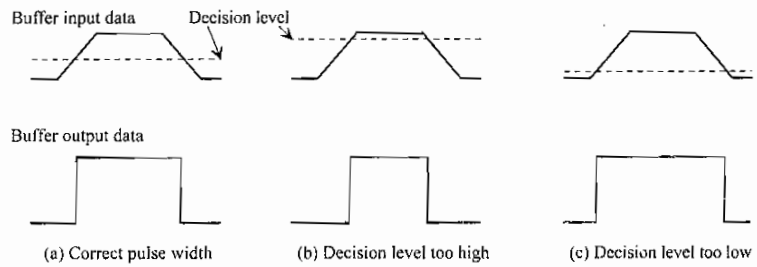


Figure 18.12 Timing errors in regenerating digital data.

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 400 ps. However, the delay time between the input and the output going low is

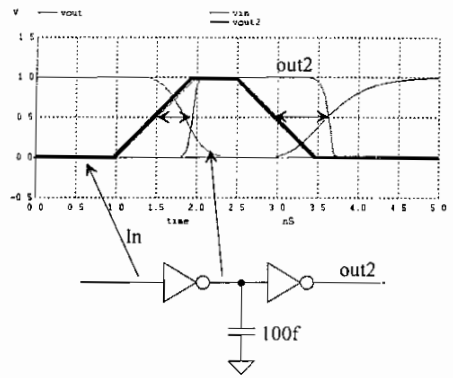


Figure 18.13 How skew is introduced into a high-speed signal.

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.

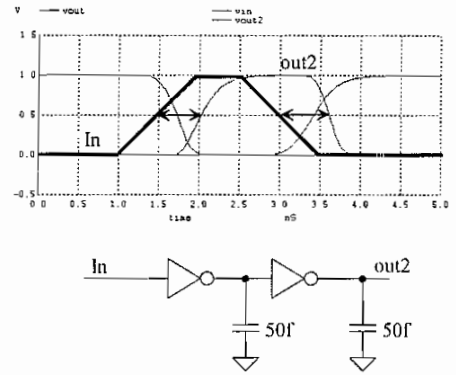


Figure 18.14 How equalizing delays can be used to reduce skew.

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).

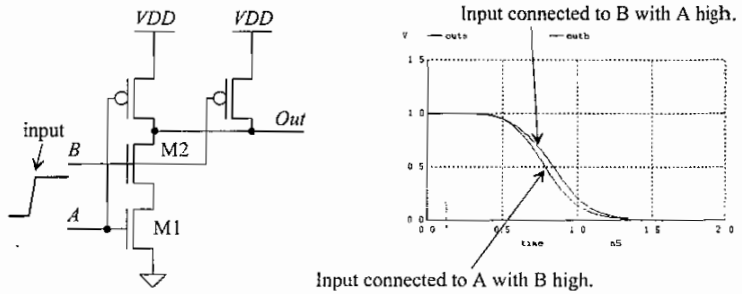


Figure 18.15 The skew introduced by using different inputs.

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.

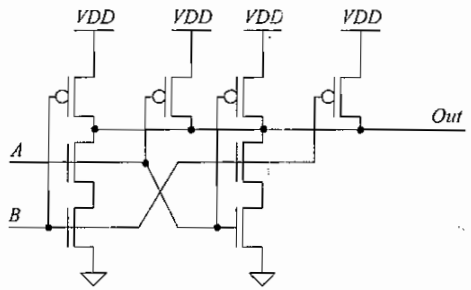


Figure 18.16 Using two NAND gates in parallel to reduce input-dependent skew.

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.

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

$$Vinp > Vinm \rightarrow \text{out} = \text{"1"} \qquad (18.22)$$

and

$$Vinp < Vinm \rightarrow \text{out} = \text{"0"} \qquad (18.23)$$

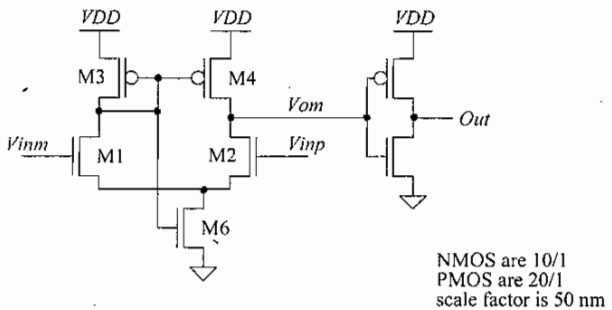


Figure 18.17 An (n-flavor) input buffer for high-speed digital design.

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).

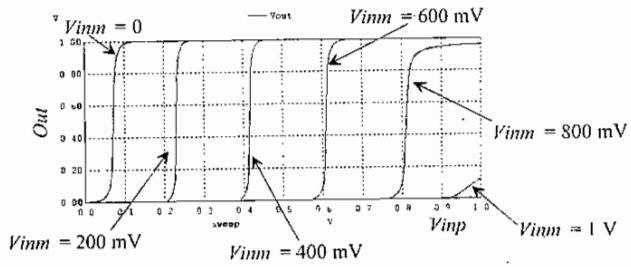


Figure 18.18 Simulating the DC behavior of the buffer in Fig. 18.17.

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.

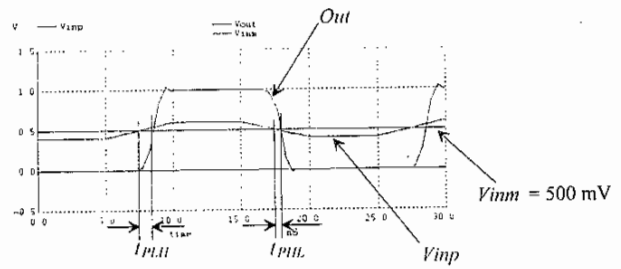


Figure 18.19 Transient response of the input buffer in Fig. 18.17 driving a 50 fF load.

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

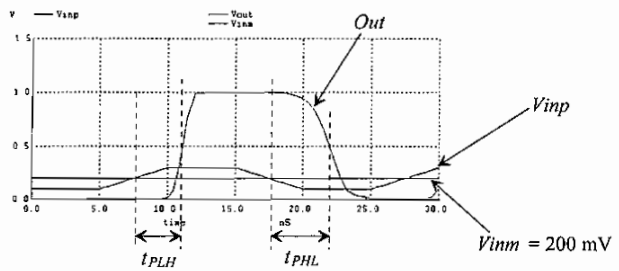


Figure 18.20 Resimulating with lower input signal voltages.

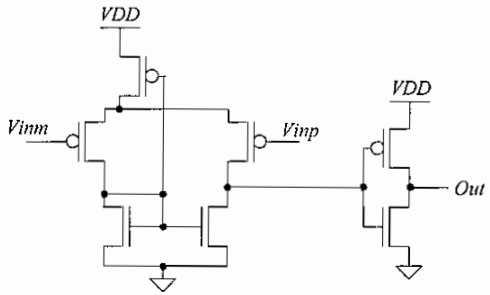


Figure 18.21 A PMOS input buffer for high-speed digital design.

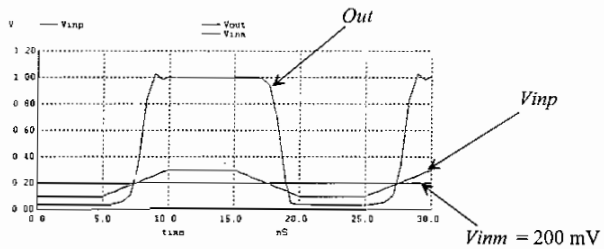


Figure 18.22 Repeating the simulation in Fig. 18.20 with the buffer in Fig. 18.21.

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.

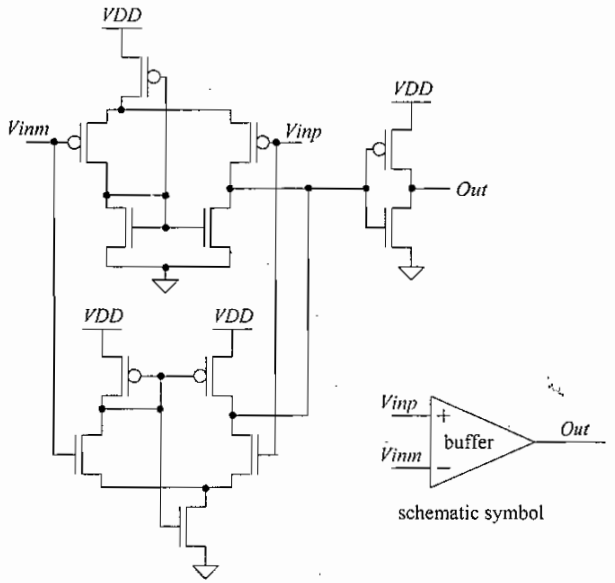


Figure 18.23 A rail-to-rail input buffer based on the topologies in Figs. 18.17 and 18.21.

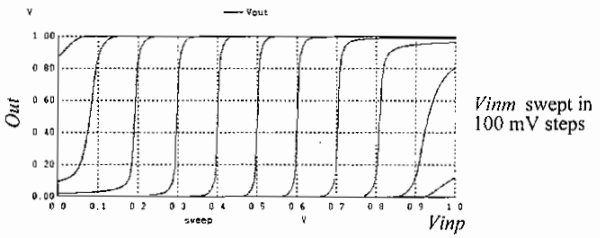


Figure 18.24 DC characterisitics of the buffer in Fig. 18.23.

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. ■

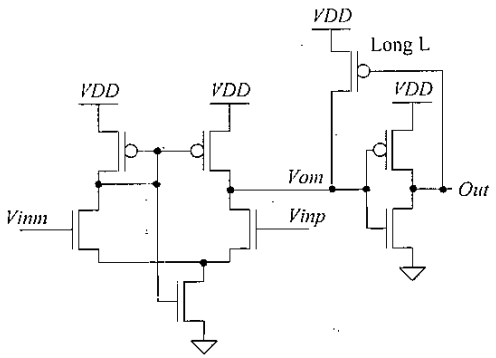


Figure 18.25 An input buffer for high-speed digital design with hysteresis.

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.

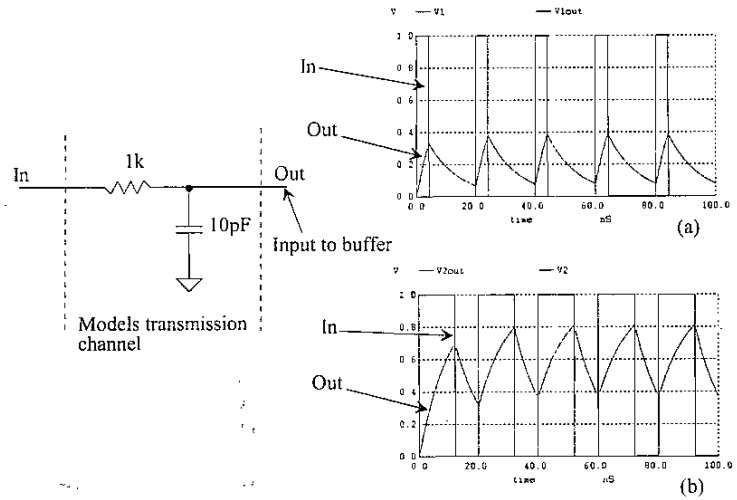


Figure 18.26 How the shape of the data changes through a communication channel.

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.

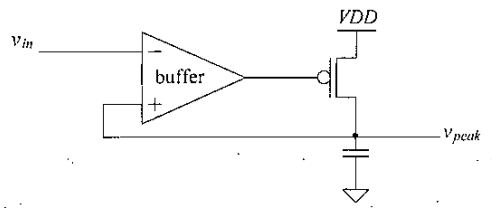


Figure 18.27 CMOS peak detector.

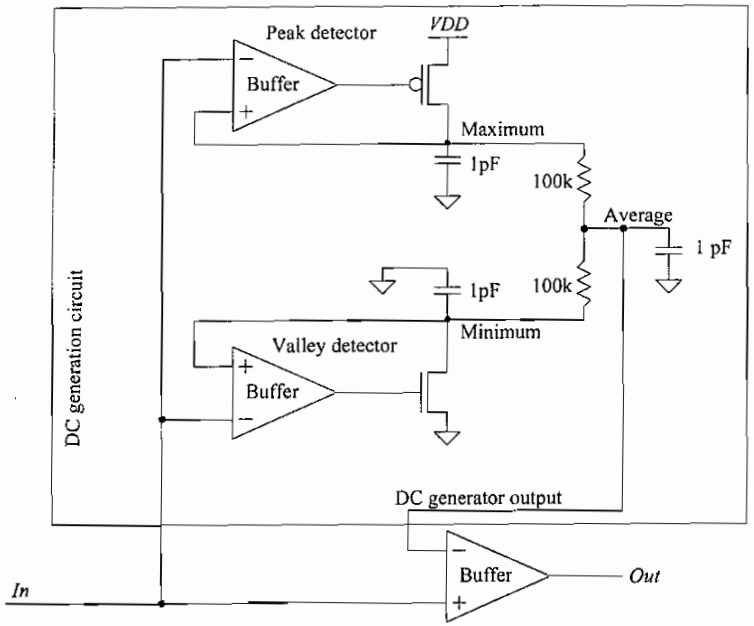


Figure 18.28 A DC generation circuit.

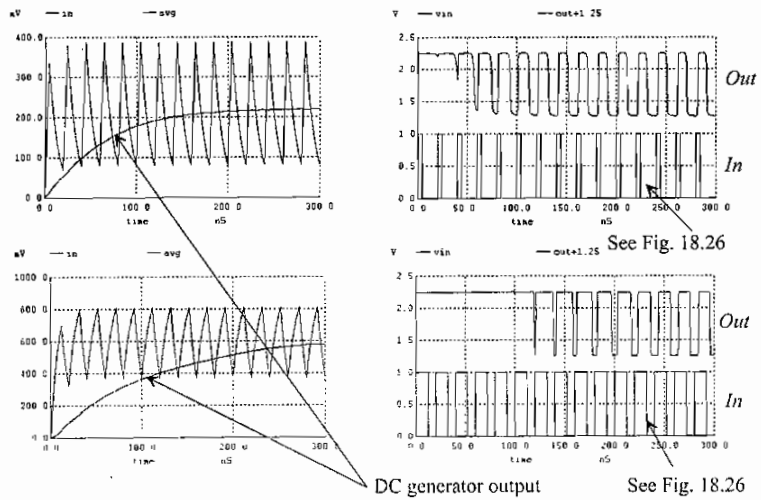


Figure 18.29 The output of the DC restore circuit in Fig. 18.28.

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.

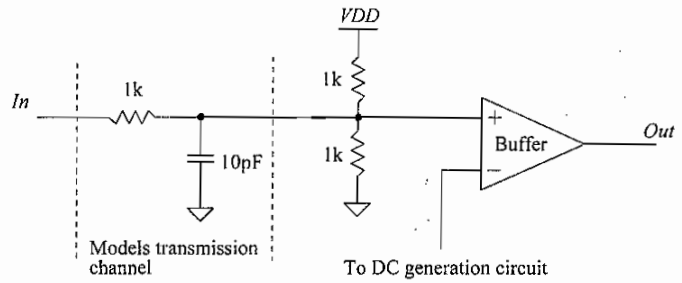


Figure 18.30 Reducing the input resistance of the input buffer.

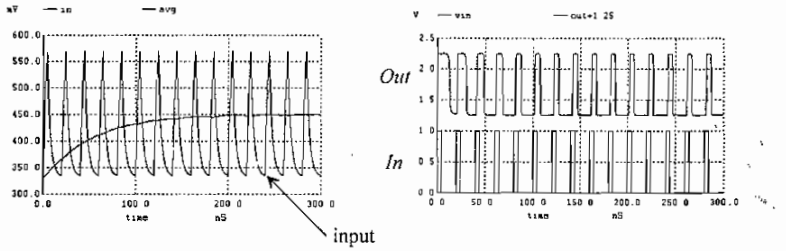


Figure 18.31 How reducing the input resistance of the buffer reduces the input signal amplitude (bad) and distortion (good).

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.

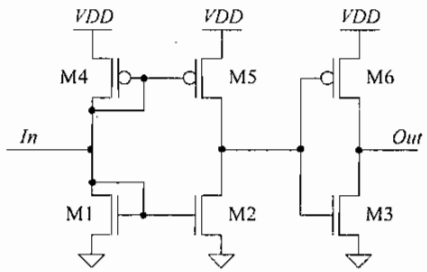


Figure 18.32 A low-input resistance input buffer.

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.

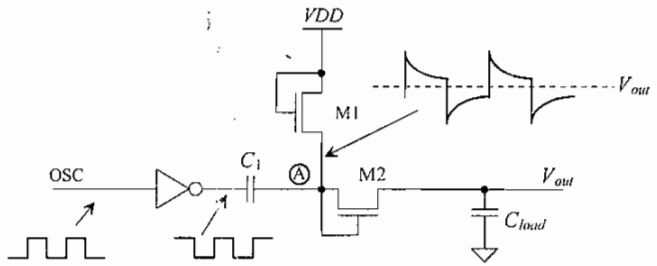


Figure 18.33 Pump used to generate a voltage greater than *VDD*.

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 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. 18.33 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.

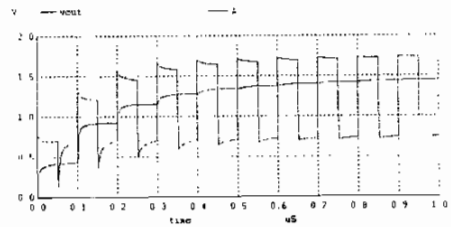


Figure 18.34 Simulating the operation of the voltage pump in Fig. 18.33.

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).

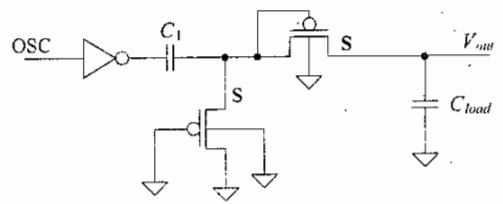


Figure 18.35 Negative voltage pump.

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.

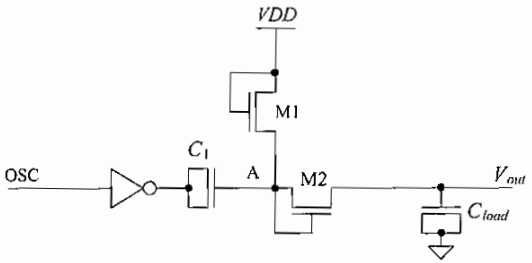


Figure 18.36 Using MOSFETs as capacitors.

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.

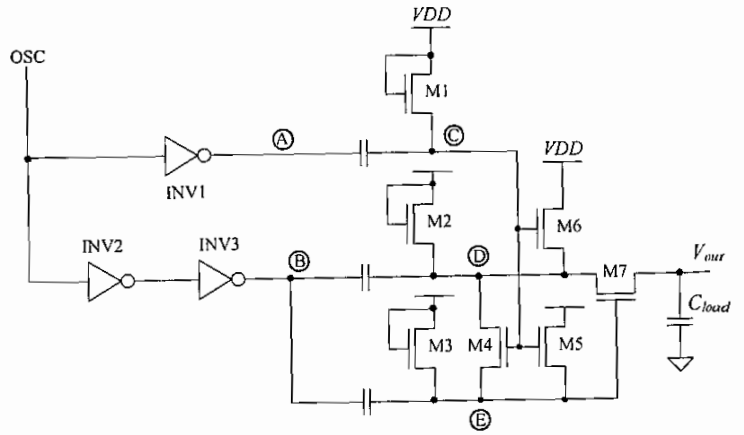


Figure 18.37 Increased output voltage pump.

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. $\overline{CLK}$ goes high and point B swings up to $3 \cdot VDD - 2 \cdot V_{THN}$. When $\overline{CLK}$ 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 $N \cdot VDD - N \cdot V_{THN}$ down to $N \cdot VDD - (N + 1) \cdot V_{THN}$. C_N can be made larger than the other capacitors to reduce this ripple.

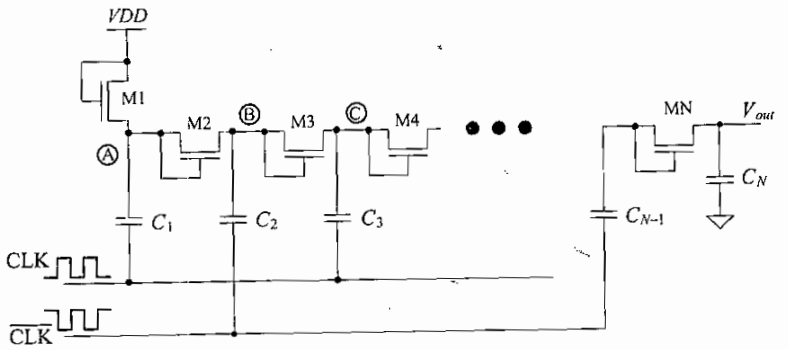


Figure 18.38 N-stage voltage multiplier (the Dickson charge pump).

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 A 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.

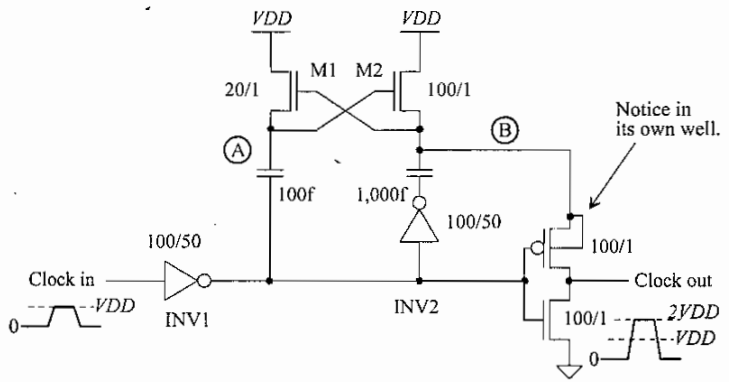


Figure 18.39 Charge-pump clock driver.

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. 18.40. 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.

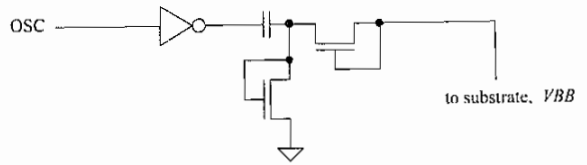


Figure 18.40 Simple substrate pump.

An example oscillator circuit that is used to drive the substrate pump is shown in Fig. 18.41. 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.

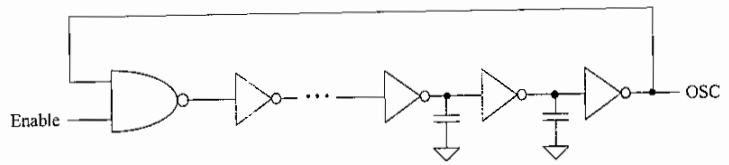


Figure 18.41 Ring oscillator with enable.

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 a comparator with hysteresis, a precision voltage reference, and a level shifting circuit, the enable signal can be generated with the circuit of Fig. 18.42. 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.

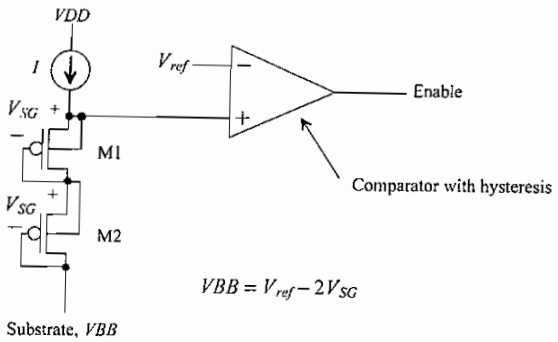


Figure 18.42 Regulator circuit used in substrate pump.

A simpler and less accurate implementation of the regulator is shown in Fig. 18.43. 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. 18.42 is implemented with the long L MOSFET M4 in Fig. 18.43. 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.

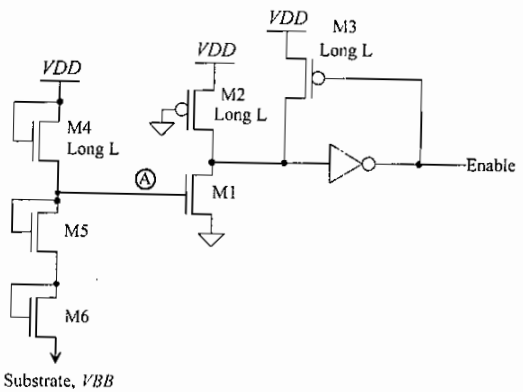


Figure 18.43 Simpler substrate regulator.

ADDITIONAL READING

[1] O. H. Schmitt, "A Thermionic Trigger," *Journal of Scientific Instruments*, Vol. XV, pp. 24–26, no. 1, Jan. 1938. Information on the "Schmitt trigger."

[2] J. Millman and H. Taub, *Pulse, Digital, and Switching Waveforms*, McGraw-Hill Publishers, 1965. ISBN 07-042386-5. Information on multivibrators and oscillators.

[3] 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.

[4] K. Sawada, Y. Sugawara, and S. Masui, "An on-chip high-voltage generator circuit for EEPROMs with a power supply voltage below 2 V," *Symp. VLSI Circuits Dig.* 9, pp. 75–76, June 1995.

[5] 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.

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.

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*.

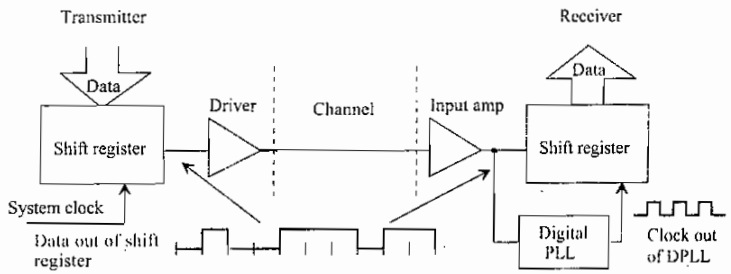


Figure 19.1 Block diagram of a communication system using a DPLL for the generation of a clock signal.

[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".

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].

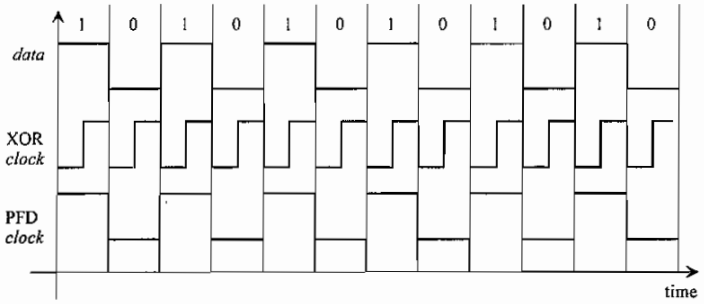


Figure 19.2 Data input to DPLL in lock and possible clock outputs using the XOR phase detector and PFD.

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.

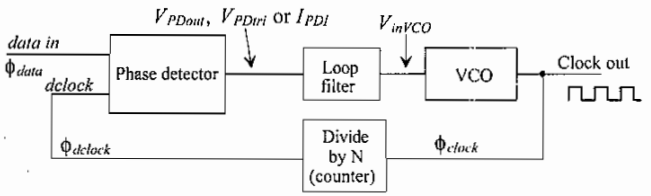


Figure 19.3 Block diagram of a digital phase-locked loop.

[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.

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 cababilities 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.

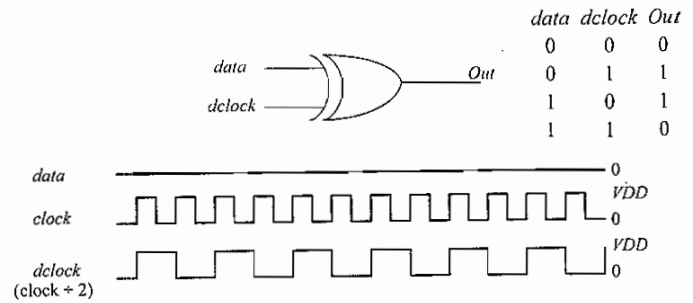


Figure 19.4 Operation of the XOR phase detector.

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.

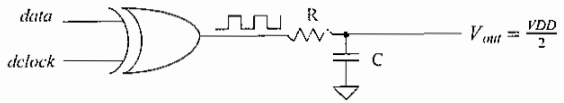


Figure 19.5 How the filtered output of the phase detector becomes VDD/2.

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 continue 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).

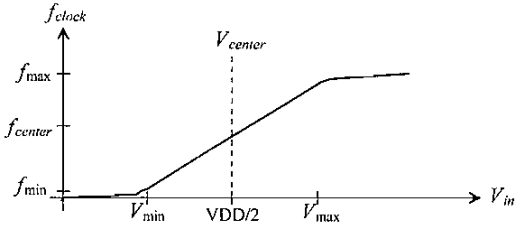


Figure 19.6 Output frequency of VCO versus input control voltage.

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.

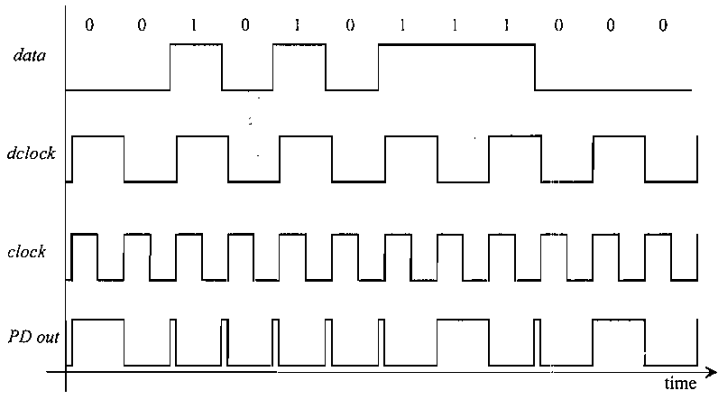


Figure 19.7 Possible XOR phase-detector inputs and the resulting output.

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 Δt. 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, Δt, 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 π. 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

$$K_{PD} = \frac{VDD}{\pi} \text{ (V/radians)} \qquad (19.6)$$

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 $\pi/2$.

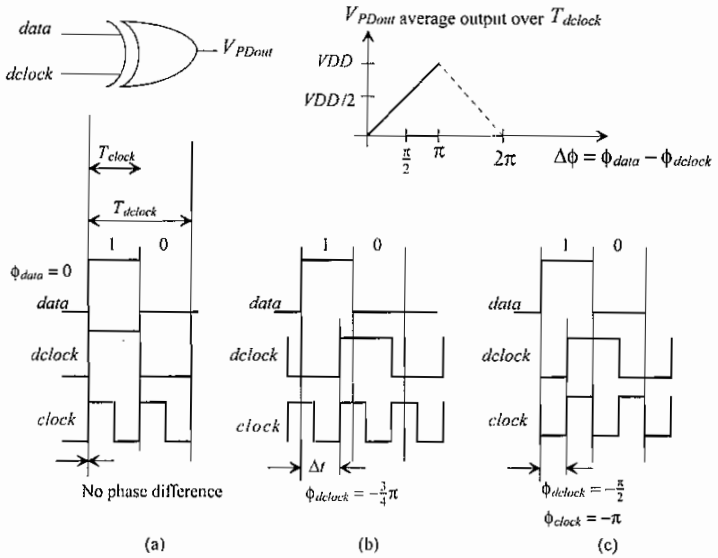


Figure 19.8 XOR PD output for various inputs (assuming input data are a string of alternating ones and zeros).

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.

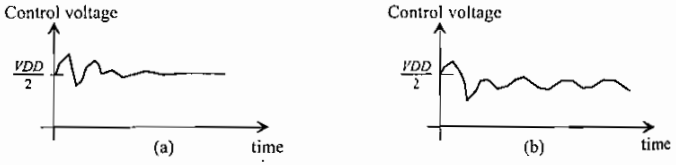


Figure 19.9 Average output voltage of phase detector during acquisition.

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.

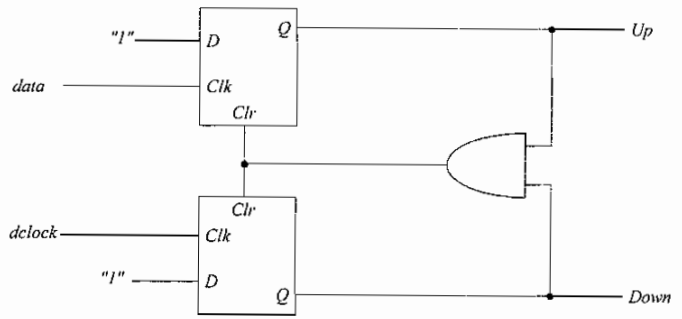


Figure 19.10 Phase frequency detector (PFD).

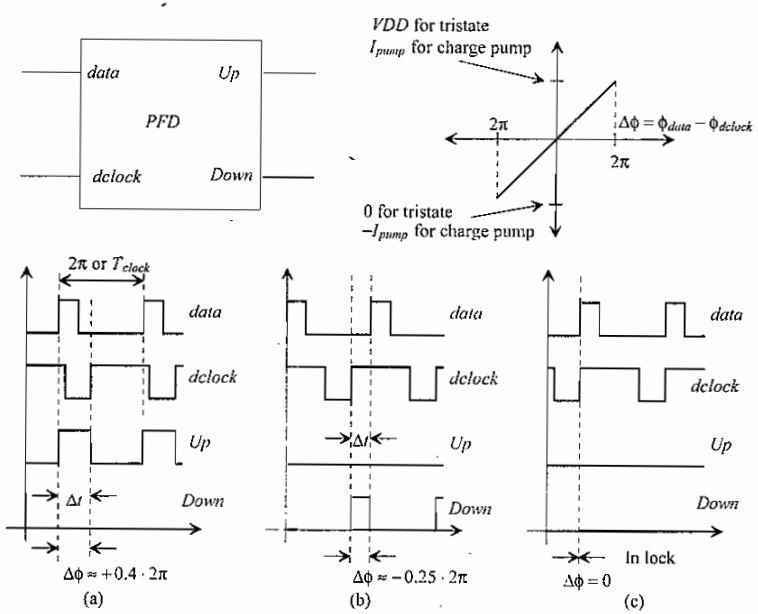


Figure 19.11 PFD phase-detector inputs and outputs.

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.

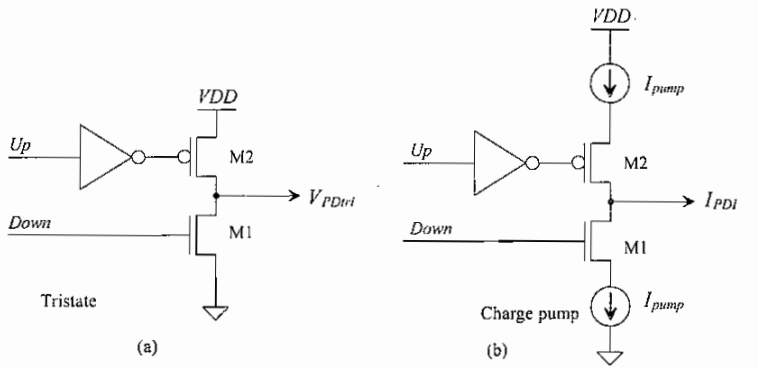


Figure 19.12 (a) Tri-state and (b) charge pump outputs of the PFD.

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 Δt 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

$$\Delta\phi = \frac{\Delta t}{T_{clock}} \cdot 2\pi \text{ (radians)} \qquad (19.7)$$

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

$$V_{PDtri} = \frac{VDD - 0}{4\pi} \cdot \Delta\phi = K_{PDtri} \cdot \Delta\phi \qquad (19.8)$$

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.

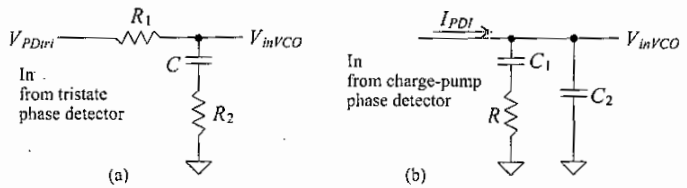


Figure 19.13 Loop filters for (a) tristate output and (b) charge-pump output.

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 ω_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 ϕ_{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 β 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

$$C_{tot} = C_{out} + C_{in} = \overbrace{C'_{ox}(W_pL_p + W_nL_n)}^{C_{out}} + \overbrace{\frac{3}{2}C'_{ox}(W_pL_p + W_nL_n)}^{C_{in}} \quad (19.18)$$

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 ≥ 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 $I_D = I_{Dcenter}$)

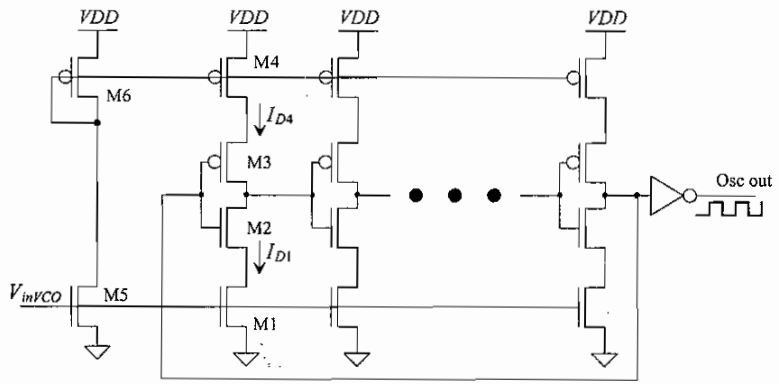


Figure 19.14 Current-starved VCO.

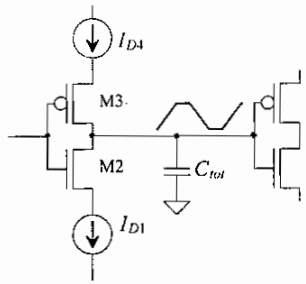


Figure 19.15 Simplified view of a single stage of the current-starved VCO.

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

$$V_{\min} = V_{THN} \text{ and } f_{\min} = 0 \quad (19.24)$$

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

$$I_{avg} = N \cdot \frac{VDD \cdot C_{tot}}{T} = N \cdot VDD \cdot C_{tot} \cdot f_{osc} \quad (19.25)$$

or

$$I_{avg} = I_D \quad (19.26)$$

The average power dissipated by the VCO is

$$P_{avg} = VDD \cdot I_{avg} = VDD \cdot I_D \quad (19.27)$$

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

$$C_{tot} = \frac{5}{2} \cdot 25\frac{fF}{\mu m^2} \cdot (10 \cdot 1 + 20 \cdot 1) \cdot \overbrace{(0.050\ \mu m)^2}^{\text{scale factor}} = 4.7\,fF$$

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. ■

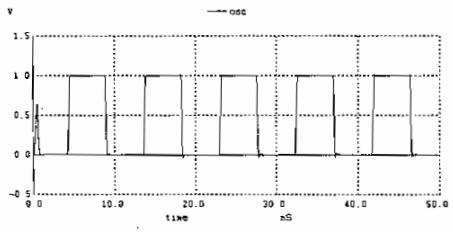


Figure 19.16 Output of the oscillator designed in Ex. 19.1.

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 1 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.*

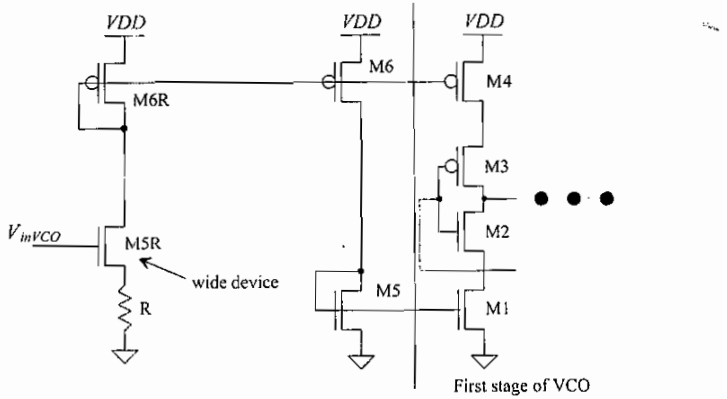


Figure 19.17 Linearizing the current in a current-starved VCO.

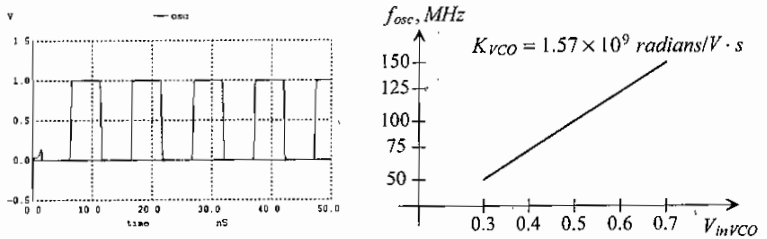


Figure 19.18 Gain of the VCO in Fig. 19.16 after linearizing with Fig. 19.17.

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.

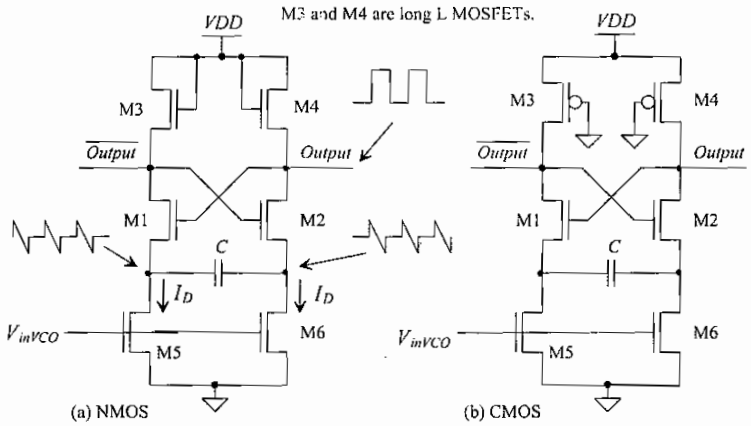
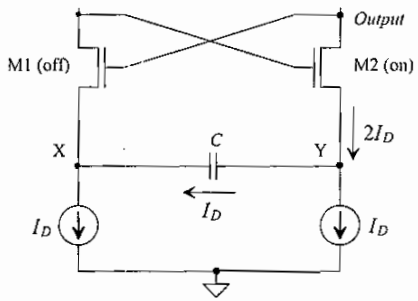


Figure 19.19 Source coupled voltage-controlled oscillators (also known as source coupled multivibrators).

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



Note: C should be laid out with the same parasitic capacitance at X and Y.

Figure 19.20 Simplified schematic of source coupled oscillator, M1 is on and M2 is off.

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.

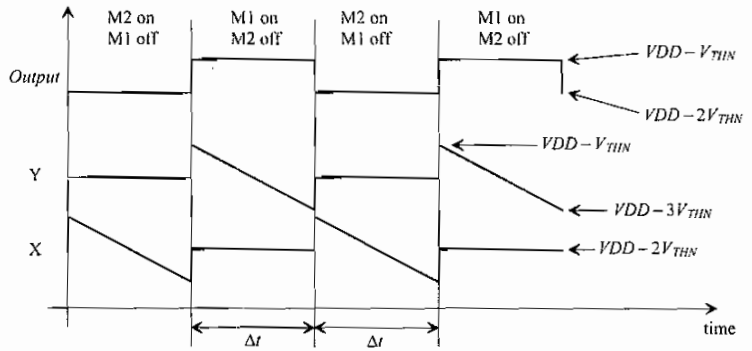


Figure 19.21 Voltage waveforms for the NMOS source-coupled VCO.

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, $\Delta\omega/t < \omega_n^2$.

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, β, 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, ω_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, ζ, 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

$$\Delta\omega_P = \frac{\pi}{2} \cdot \sqrt{2\zeta\omega_n K_{VCO}K_{PD} - \omega_n^2} \tag{19.37}$$

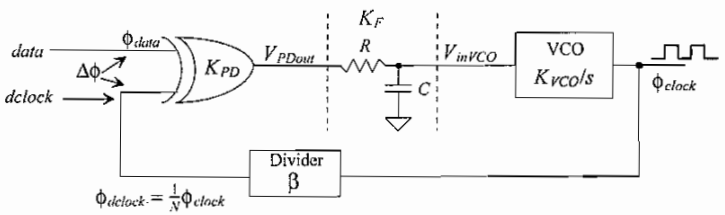


Figure 19.22 Block diagram of a DPLL using a XOR phase detector

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

$$K_{VCO} = 1.57 \times 10^9 \; radians/V \cdot s$$

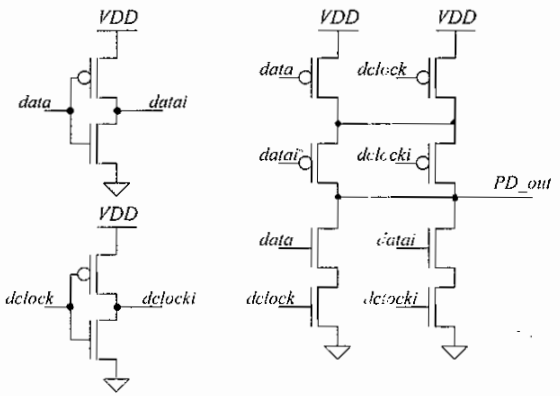


Figure 19.23 XOR phase detector.

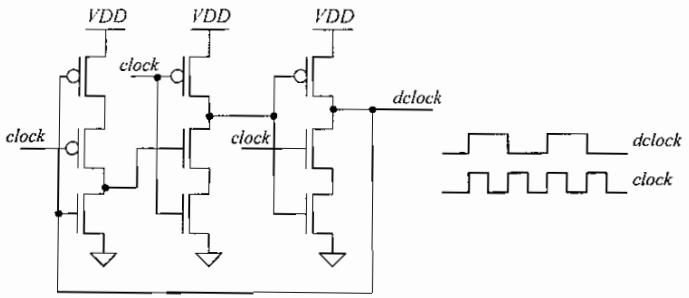


Figure 19.24 A divide by two circuit using true single phase clocking (TSPC) logic.

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 ζ 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

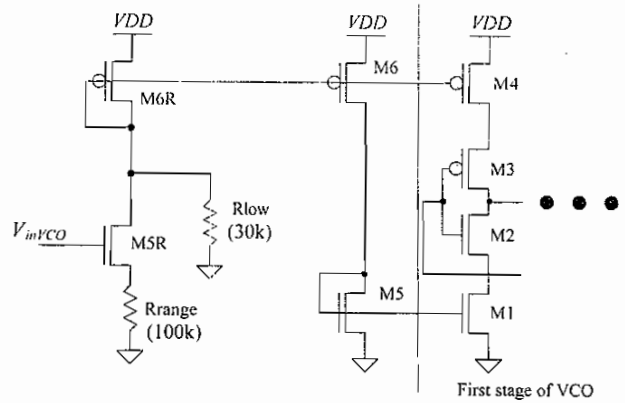


Figure 19.25 Limiting the current in a current-starved VCO.

to calculate the pull-in range since it is greater than the lock-in range. Again, the design of the oscillator is important in this DPLL. The lock time, T_L, of the DPLL is 40 ns. SPICE simulation results of this DPLL are seen in Fig. 19.27. The top waveform is the input control voltage of the VCO. The time that it takes the loop to pull-in the frequency and lock, from this figure, is roughly 250 ns. The lock time would be the time it takes the loop if the input frequency jumped after the loop was locked. Note how, in the bottom traces, the output clock rising edge is centered on the data. Notice that the VCO's control voltage varies by 200 mV. From Fig. 19.26, we can see a 5 MHz variation, Δf_{VCO}, in the VCO's output frequency. To estimate the jitter, Δt_{jitter}, in the output, we can use

$$\Delta t_{jitter} = \frac{1}{f_{center}} - \frac{1}{f_{center} + \Delta f_{VCO}} = \frac{1}{100\ MHz} - \frac{1}{105\ MHz} = 480\ ps \qquad (19.41)$$

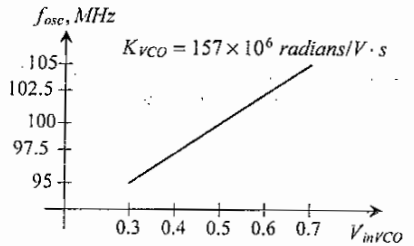


Figure 19.26 Gain of the VCO in Fig. 19.16 after linearizing and limiting with Fig. 19.25.

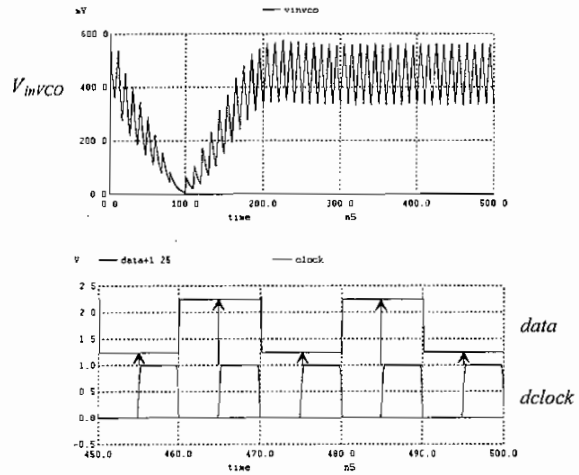


Figure 19.27 Simulating the DPLL in Ex. 19.2.

In this simulation (Fig. 19.27), we used the ideal input data, that is, a string of alternating ones and zeroes. Figure 19.28 shows what happens when the data is a one followed by seven zeroes. The rising edge of the clock is occurring a little offset from the center of the data. This error is sometimes called a *static phase error*. As mentioned earlier, when the input data is not changing, the VCO control voltage starts to wander back towards *VDD*/2.

Finally, *it's very important* to realize that if we were to increase the *RC* time constant of the filter, as seen above, the damping factor, ζ , would decrease and the loop would never lock. This is counter-intuitive and the reason we should *always use the design equations when designing DPLLs.* ■

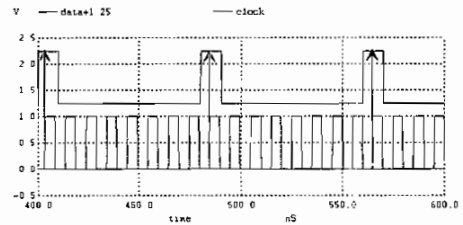
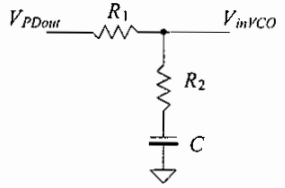


Figure 19.28 Showing how the DPLL doesn't lock up to the center of the data when the data isn't alternating ones and zeroes.

Adding a zero to the simple passive RC loop filter, Fig. 19.29 (called a passive lag loop filter), the loop-filter pole can be made small (and thus the gain of the VCO can be made larger) while at the same time a reasonable damping factor can be achieved. The result is an increase in the lock range of a DPLL using the XOR PD and a shorter lock time see Eqs. (19.37) – (19.40). The passive lag loop filter is, in most situations preferred over the simple RC. Again, as Ex. 19.2 showed, if the center frequency of the VCO doesn't match the input frequency, the clock will not align at $\pi/2$ (in the center of the data).



$$K_F = \frac{V_{inVCO}}{V_{PDout}} = \frac{1 + j\omega R_2 C}{1 + j\omega(R_1 + R_2)C}$$

$$\omega_n = \sqrt{\frac{K_{PD}K_{VCO}}{N(R_1 + R_2)C}} \qquad \Delta\omega_L = \pi\zeta\omega_n$$

$$\zeta = \frac{\omega_n}{2} \cdot \left(R_2 C + \frac{N}{K_{PD}K_{VCO}}\right)$$

Figure 19.29 Passive lag loop filter used to increase DPLL lock range.

Active PI Loop Filter

The clock misalignment encountered in a DPLL using an XOR PD and passive loop filter can be minimized by using the active proportional + integral (active PI) loop filter shown in Fig. 19.30. The integrator allows the V_{inVCO} voltage to move, and stay, away from *VDD*/2. This then eliminates the static phase error (the clock will align to the middle of the data). The transfer function of this filter is given by

$$K_F = \frac{1 + sR_2C}{sR_1C} \tag{19.42}$$

The natural frequency of the resulting second-order system is given by

$$\omega_n = \sqrt{\frac{K_{PD}K_{VCO}}{NR_1C}} \tag{19.43}$$

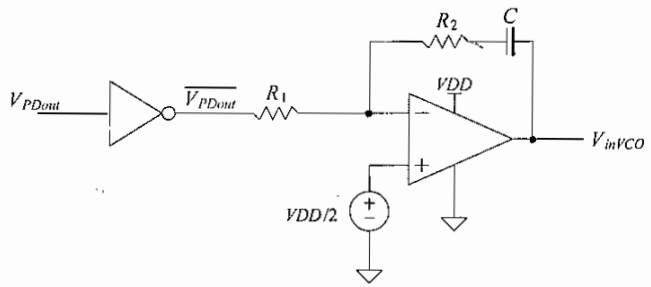


Figure 19.30 Active PI loop filter.

and the damping ratio is given by

$$\zeta = \frac{\omega_n R_2 C}{2} \tag{19.44}$$

The lock range is

$$\Delta\omega_L = 4\pi\zeta\omega_n \tag{19.45}$$

while the lock time remains $2\pi/\omega_n$. The pull-in range, using the active PI loop filter, is limited by the VCO oscillator frequency. Consider the following example.

Example 19.3

Repeat Ex. 19.2 using the active PI loop filter. The SPICE model for an op-amp (a voltage-controlled voltage source) seen in Fig. 20.19 can be used for the op-amp.

Using Eq. (19.43) in (19.44) gives

$$\zeta = \frac{R_2 C}{2}\sqrt{\frac{K_{PD}K_{VCO}}{NR_1 C}}$$

If we set the damping factor again to 1 we can write

$$4 = \frac{157\times 10^6 R_2^2 C}{2\pi R_1}$$

Setting C to 10 pF and R_2 to 25k, then R_1 is 39k. The simulation results are seen in Fig. 19.31. The output clock is aligned to the center of the data. It's important to make sure that the step size used in the SPICE transient simulation isn't too big. For example, if a 100 MHz clock is used with a step size of 1 ns, it is likely that the loop either won't lock or if it does lock, both the VCO control voltage and thus the output data will oscillate. A maximum step size for 100 MHz signals may be 100 ps.

Finally, it is important to remember that the VCO tuning scheme in Fig. 19.25 is not practical unless the components can be set manually (either off-chip or using fuses or switches on chip). Also, something very important that we are not discussing (yet) is power supply noise. The current-starved VCO's output frequency is not very stable with changes in *VDD*. ■

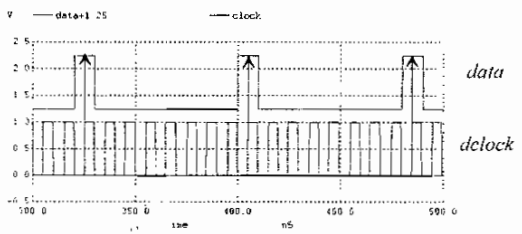


Figure 19.31 How the loop locks up in the center of the data.

19.3.2 PFD DPLL

Tri-State Output

A block diagram of a DPLL using the PFD with tri-state output is shown in Fig. 19.32. The phase transfer function is the same as Eq. (19.30)

$$H(s) = \frac{\phi_{clock}}{\phi_{data}} = \frac{K_{PDtri}K_F K_{VCO}}{s + \beta\cdot K_{PDtri}K_F K_{VCO}} \tag{19.46}$$

where

$$K_F = \frac{1+sR_2C}{1+s(R_1+R_2)C} \tag{19.47}$$

When this filter is driven with the tri-state output, no current flows in R_1 or R_2 with the output in the high impedance state. The voltage across the capacitor remains unchanged. We can think of the filter, tri-state output as an ideal integrator with a transfer function

$$K_F' = \frac{1+sR_2C}{s(R_1+R_2)C} \tag{19.48}$$

Substituting this equation into Eq. 19.46 and rearranging results in

$$H(s) = \frac{K_{PDtri}K_{VCO}\frac{1+sR_2C}{(R_1+R_2)C}}{s^2 + s\frac{K_{PDtri}K_{VCO}R_2C}{N(R_1+R_2)C} + \frac{K_{PDtri}K_{VCO}}{N(R_1+R_2)C}} = \frac{\phi_{clock}}{\phi_{data}} = \frac{f_{clock}}{f_{data}} \tag{19.49}$$

From this equation, the natural frequency is

$$\omega_n = \sqrt{\frac{K_{PDtri}K_{VCO}}{N(R_1+R_2)C}} \tag{19.50}$$

and the damping factor is determined by solving

$$2\zeta\omega_n = \frac{K_{PDtri}K_{VCO}R_2C}{N(R_1+R_2)C} \tag{19.51}$$

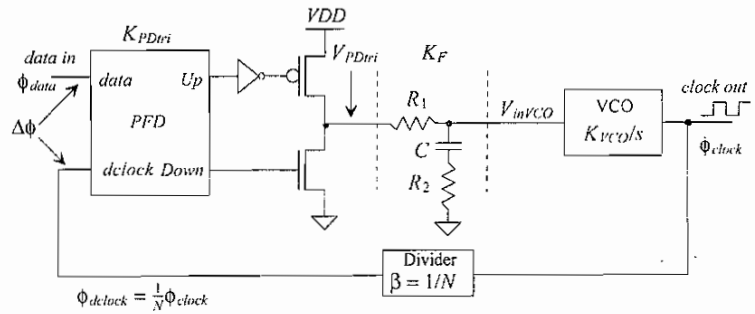


Figure 19.32 Block diagram of a DPLL using a sequential phase detector (PFD).

which results in a damping factor given by

$$\zeta = \frac{\omega_n}{2} \cdot R_2 C \tag{19.52}$$

The lock range is given by

$$\Delta\omega_L = 4\pi\zeta\omega_n \tag{19.53}$$

while the lock time, T_L, remains $2\pi/\omega_n$. The pull-in range is limited by the VCO operating frequency. The pull-in time is given by

$$T_P = 2R_1 C \cdot \ln \frac{(K_{VCO}/N) \cdot (VDD/2)}{(K_{VCO}/N)(VDD/2) - \Delta\omega} \tag{19.54}$$

where $\Delta\omega$ is the magnitude of the *input* frequency step.

Implementing the PFD in CMOS

The PFD seen in Fig. 19.10 can be implemented using inverters and nand gates as seen in Fig. 19.33. Simulating this circuit gives the results seen in Fig. 19.34. When the *dclock* is lagging, the *data* the output of the PFD is the *up* pulse (indicating that the edge of *dclock* needs to speed up or occur earlier in time, that is, the VCO's control voltage should increase). When *data* is lagging *dclock* the *down* pulse goes high indicating *dclock* should slow down.

We have some fine points that need to be discussed here. For example, what happens to *up* and *down* as the rising edges of *dclock* and *data* move close together? What we may get is some small glitches in the *up* and *down* signals. Or we may see both *up* and *down* staying low as the two pulses move closer together. In the PFD of Fig. 19.33, the delay through the two inverters can be used to set how *up* and *down* behave as

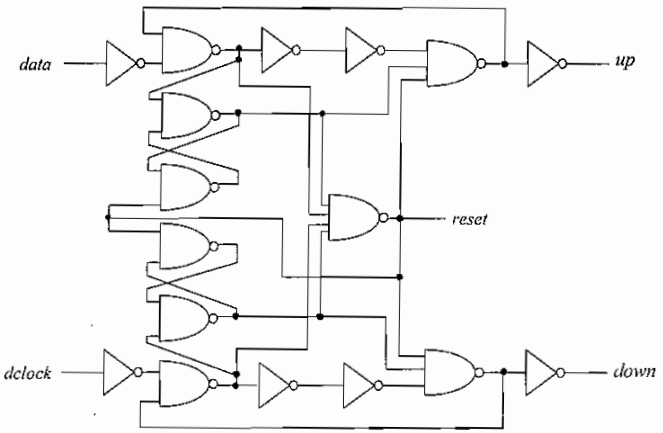


Figure 19.33 CMOS implementation of a PFD.

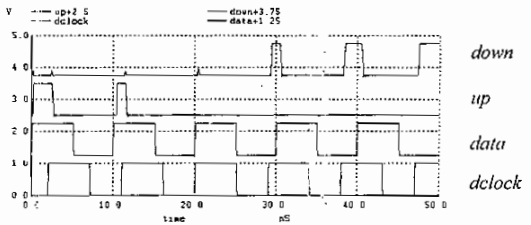


Figure 19.34 Simulation results for the PFD in Fig. 19.33.

the PFD's inputs move close together. If, for example, both stay low when the PFD's inputs get within 100 ps of each other, we get a *static phase error*. The loop won't lock any tighter than within 100 ps of its ideal position. Another way of looking at this is that as the phase difference, $\Delta\phi$, moves towards zero (see Fig. 19.11), the gain (the slope of the curve in Fig. 19.11) decreases. The phase detector is then said to have a *dead zone* where the gain of the phase detector (the slope of the curves) decreases.

Example 19.4

Design a DPLL using the tri-state topology seen in Fig. 19.32 that generates a clock signal at a frequency of 100 MHz from a 50 MHz square wave input. This application of the DPLL is called *frequency synthesis*.

The feedback path contains a divide by 2 circuit ($N = 2$). Let's use the VCO with the characteristics seen in Fig. 19.18

$$K_{VCO} = 1.57 \times 10^9 \; radians/V \cdot s$$

The gain of the phase detector (knowing *VDD* is 1 V) is

$$K_{PDtri} = \frac{1}{4\pi}$$

The lock range, Δf_L will be set to 20 MHz. Again let's set $\zeta = 1$. Using Eq. (19.53)

$$\Delta\omega_L = 4\pi\zeta\omega_n \rightarrow \omega_n = 10 \times 10^6 \; radians/V \cdot s$$

Using Eq. (19.52) gives

$$R_2 C = 200 \; ns$$

Let's set the capacitor to 10 pF and R_2 to 20k. Solving Eq. (19.50) for R_1 gives 42.5k ($= R_1$).

The simulation results are seen in Fig. 19.35. We should first notice that the VCO's control voltage doesn't have the excessive ripple like we had in the DPLL using the XOR gate phase detector. The response of the loop shows a nice $\zeta = 1$ shape. Again note that a DPLL loop's pull-in range is limited by the VCO's operating frequency (which, in this example uses the VCO from Fig. 19.18, which ranges from 50 to 150 MHz). A good exercise to perform at this point is to change the divider in the feedback path (to divide by 1, 2, 4, etc.) and the input frequency (a signal we've called *data*) and look at the robustness of the loop. ■

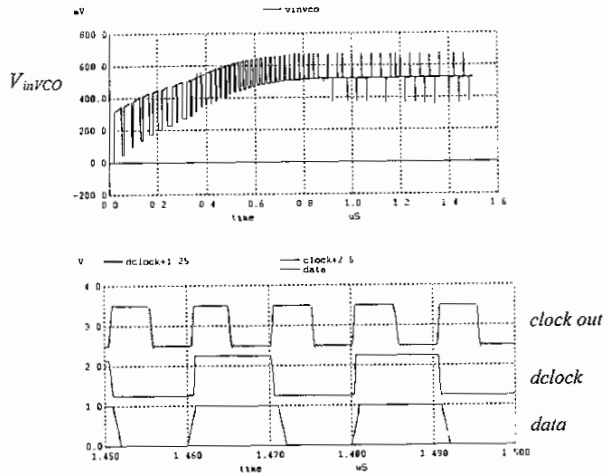


Figure 19.35 Simulation results for Ex. 19.4.

PFD with a Charge Pump Output

The PFD with a charge-pump output is seen in Fig. 19.36. A CMOS implementation of a DPLL using this configuration is, in general, preferred over the tri-state output because of the better immunity to power supply variations. In the tristate configuration seen in Fig. 19.32 note that when either the NMOS or PMOS switches are on, either *VDD* or ground are connected directly to the loop filter. Power supply or ground noise can thus feed directly into the loop filter and then into the VCO's control voltage.

The loop filter integrates the charge supplied by the charge pump. The capacitor C_2 prevents $I_{pump}{\cdot}R$ from causing voltage jumps on the input of the VCO and thus frequency jumps in the DPLL output. In general, C_2 is set at about one-tenth (or less) of C_1. The loop-filter transfer function neglecting C_2 is given by

$$K_F = \frac{1 + sRC_1}{sC_1} \tag{19.55}$$

The feedback loop transfer function is given by

$$H(s) = \frac{\phi_{clock}}{\phi_{data}} = \frac{K_{PDI}K_{VCO}(1 + sRC_1)}{s^2 + s(\frac{K_{PDI}K_{VCO}R}{N}) + \frac{K_{PDI}K_{VCO}}{NC_1}} \tag{19.56}$$

From the transfer function the natural frequency is given by

$$\omega_n = \sqrt{\frac{K_{PDI}K_{VCO}}{NC_1}} \tag{19.57}$$

and the damping factor is

$$\zeta = \frac{\omega_n}{2} \cdot RC_1 \tag{19.58}$$

The lock range and lock time remain the same (using the different values for the natural frequency and damping ratio) as the PFD with the tri-state output. Again, the pull-in range is set by the VCO oscillator frequency range. The pull-in time is given by

$$T_P = 2RC_1 \ln\left[\frac{(K_{VCO}/N) \cdot (I_{pump})}{(K_{VCO}/N) \cdot (I_{pump}) - \Delta\omega}\right] \tag{19.59}$$

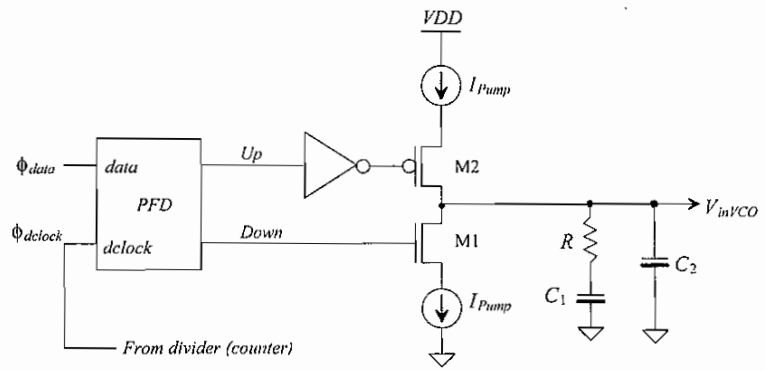


Figure 19.36 PFD using the charge pump.

Practical Implementation of the Charge Pump

The circuit seen in Fig. 19.36 is useful for illustrating the concept of the PFD with charge pump. However, in a practical circuit the fact that the sources of M1 and M2 charge to ground and *VDD* respectively when M1 and M2 are off creates some design issues. For example, suppose that the source of M1 is discharged to ground when the *down* signal goes high. When M1 turns on, it doesn't supply the charge set by I_{pump} to the loop filter but rather, until the voltage across the current sink increases, behaves like a switch simply connecting the filter's input to ground (meaning that we are not controlling the signal that is applied to the loop filter). To get around this problem, consider the circuit seen in Fig. 19.37. The bias voltages come from diode-connected MOSFETs (to form current mirrors, see Ch. 20). When *up* and *down* are low, M1L and M2L are on. The pump-up current source, MPup, is driving the pump-down current source, MNdwn. When either *up* or *down* goes high, the output is connected to one of the current sources. The MOSFETs M1L,R and M2L,R simply steer the current to the loop filter. The only other concern we have with this topology is the fact that when M1L and M2L are both on, the voltage on their drains won't precisely match the voltage across the loop filter (V_{inVCO}). In other words, the voltage across each of the current sources (MPup and MNdwn) won't be the same as it is when they are connected to the output node. The result is that charge sharing between the parasitic capacitance on the drains of MPup and MNdwn and the capacitance used in the loop filter cause a static phase error or jitter. To eliminate this problem, an amplifier (see dashed lines in the figure) can be inserted to set the drain voltage of M1L and M2L to the same value V_{inVCO}.

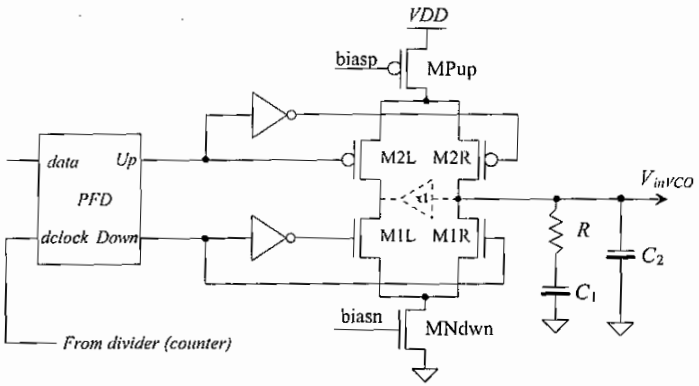


Figure 19.37 Practical implementation of the charge pump.

Example 19.5

Repeat Ex. 19.4 using the PFD and charge pump seen in Fig. 19.37.

One of the other benefits of using the charge pump is the fact that we can select I_{pump}, that is,

$$K_{PDI} = \frac{I_{pump}}{2\pi}$$

Again, let's set the lock range to 20 MHz. Using Eq. (19.53) with $\zeta = 1$ gives, again, $\omega_n = 10 \times 10^6$ *radians/V · s*. From Eq. (19.58) we get

$$RC_1 = 200\ ns \text{ so let's use } R = 20\text{k},\ C_1 = 10\text{ pF, and } C_2 = 1\text{ pF}$$

remembering that we generally set C_2 to one-tenth of C_1. Using Eq. (19.57), we can now find the value of I_{pump}

$$10 \times 10^6 = \sqrt{\frac{I_{pump} \cdot 1.57 \times 10^9}{2\pi \cdot 2 \cdot 10 \times 10^{-12}}} \rightarrow I_{pump} = 8\ \mu A$$

Because this value isn't that critical, we'll round it up to 10 μA. Figure 19.38 shows the simulation results. Notice the $\zeta = 1$ (or maybe a little less because the voltage does overshoot its final value by a little bit) shape of the VCO's control voltage. Let's modify the loop filter to show what V_{inVCO} would look like if $\zeta = 0.1$. All we have to do, from Eq. (19.58), is drop R by a factor of 10. The result is seen in Fig. 19.39. The control voltage is oscillating and the loop isn't locking. Of course, we can also increase the damping factor. The loop now behaves sluggishly and does not respond to fast changes in the input signal. For general design, where the process and temperature vary, it's better to center the design on a larger damping factor to avoid instability. ■

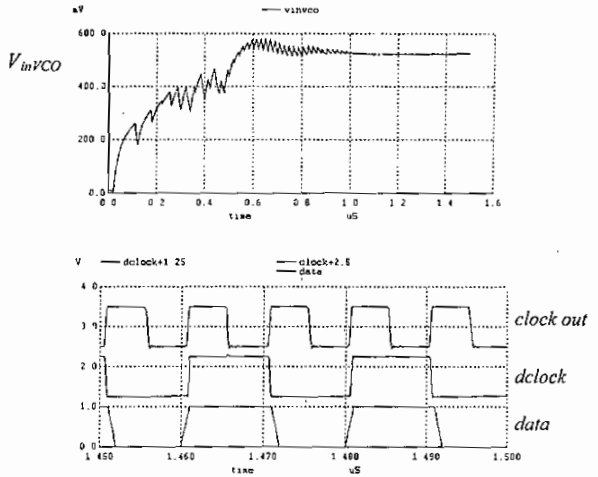


Figure 19.38 Simulation results for Ex. 19.5.

Discussion

When selecting values for the loop filter, we assumed that the output resistance of the phase detector was small (for the XOR and tri-state PD) compared to the impedances used in the loop filter. We also assumed that the input resistance of the VCO was infinite and the input capacitance of the VCO was small compared to the capacitance used in the loop filter. Considering the parasitics present in the DPLL is an important part of the design.

The center frequency of the VCO is critical for good DPLL performance when using the XOR gate with *RC* loop filter. If the center frequency, f_{center}, of the VCO (i.e., $V_{inVCO} = VDD/2$) does not match twice the input data rate, the DPLL will lock up at a phase different from π/2 (the input frequencies to any phase detector must be equal). The need for a precision center frequency is eliminated by using the XOR PD with an active PI loop filter or by using the PFD. The other big benefit of using the PFD is that the VCO's gain can be larger.

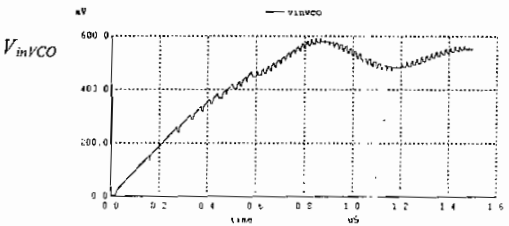


Figure 19.39 Showing what happens when the damping factor is reduced to 0.1.

Finally, selecting of the loop's damping factor, ζ, is very important. If the value of ζ is too small, the loop will have trouble locking or the output will jitter excessively when the loop is locked. This last problem is sometimes (gratuitously) called *jitter peaking* since a step in the DPLL's input frequency causes excessive overshoot in $V_{in\text{-}VCO}$ with small ζ.

19.4 System Considerations

System concerns are often the driving force behind the design of a DPLL. Referring to Fig. 19.1, we observe that the data transmitted through the channel should ideally arrive at the receiver with the same shape with which it was transmitted. In reality, the data becomes distorted. Distortion arises from nonlinearities in the receiver input amplifier and the finite bandwidth of the channel. To understand the conditions for distortionless transmission, consider the block diagram shown in Fig. 19.40. The system has a transfer function in the frequency domain of $H(f)$ and in the time domain $h(t)$. For distortionless transmission, we can relate the input and output of the system by

$$y(t) = K \cdot x(t - t_o) \tag{19.60}$$

where t_o is the time delay through the system and K is a constant. This equation shows that for distortionless transmission through a system the output is simply a scaled, time-delayed version of the input. An interesting observation can be made by taking the Fourier Transform of both sides of this equation,

$$Y(f) = K \cdot X(f) e^{-j2\pi f t_o} \tag{19.61}$$

The transfer function of a distortionless system can then be written as

$$H(f) = \frac{Y(f)}{X(f)} = K e^{-j2\pi f t_o} \tag{19.62}$$

Figure 19.41 shows the magnitude and phase responses of a distortionless system. A system is distortionless when its amplitude response, $|H(f)|$, is a constant, K, and its phase response, $\angle H(f)$, is linear with a slope of $-2\pi t_o$ over all frequencies of interest.

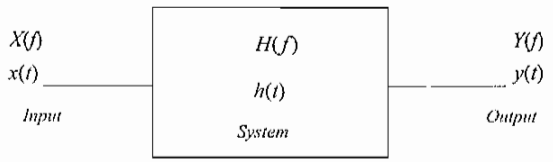


Figure 19.40 Representation of a system with input and output.

The responses shown in Fig. 19.41 are ideal. In practice, the magnitude response of a system may look similar to Fig. 19.42a. At higher frequencies, the magnitude rolls off. To compensate for this roll-off, or other imperfections, a circuit called an equalizer is added in series with the system (Fig. 19.42b). The equalizer has a transfer function in which its magnitude response increases with increasing frequency beyond a point (Fig. 19.42c). If the low-frequency gain of the equalizer is A/K and the low-frequency gain of the system is K, then the resulting gain of the system/equalizer combination is A.

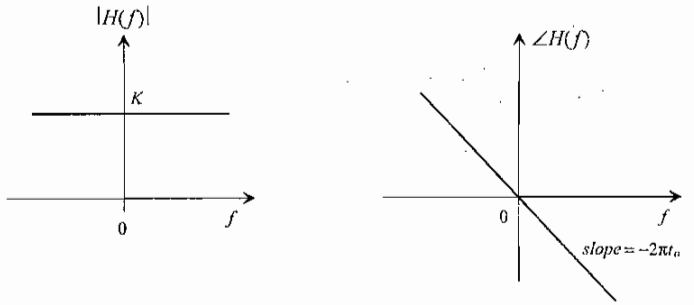


Figure 19.41 Magnitude and phase response of a distortionless system.

Another source of potential distortion occurs when the receiver input data is regenerated into digital levels. This was discussed back in Sec. 18.3. Timing errors occur when the input data is not precisely sliced through its middle (see Fig. 18.12). What makes slicing the data correctly even more difficult is the fact that the amplitude response of the channel can change with time and the data pattern can affect the average level of the data. There are two solutions to this problem. The first uses a circuit (see Fig. 18.29) that determines the peak positive and negative input analog amplitudes, averages the values, and feeds back the result to the comparator in the decision-making circuit. The second method encodes the digital data so that the duty cycle of the resulting encoded data is 50%. The encoding increases the channel bandwidth for a constant data rate.

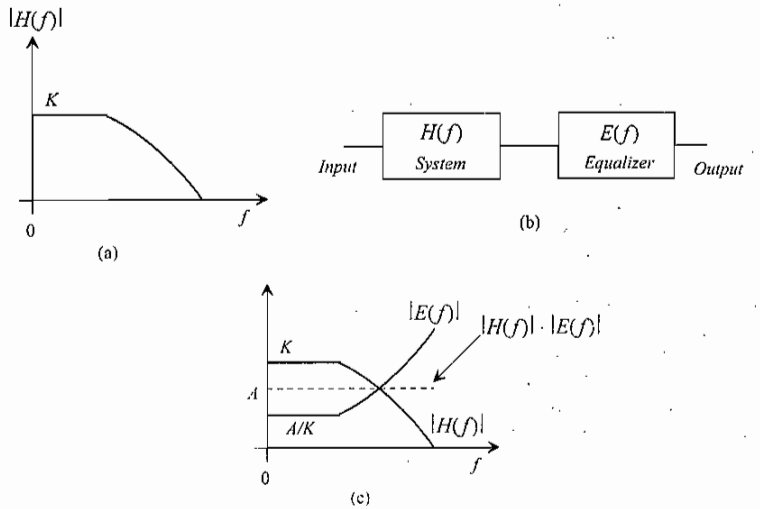


Figure 19.42 Using an equalizer to lower distortion in a system.

Encoding the data can eliminate the need for a decision circuit. If the resulting encoded data has a 50% duty cycle, it can be passed through a capacitor in the receiver, resulting in an analog signal centered around ground. The noninverting input of the comparator will then be connected to ground. The comparator will then be able to slice the data at the correct moments in time (in the middle of the data bit). An example of an encoding scheme is shown in Fig. 19.43. Encoding occurs in the transmitter prior to transmission over the channel. This particular encoding scheme is referred to as the bi-phase format, or more precisely, the bi-phase-level (sometimes called bi-phase-L or Manchester NRZ) format. The cost of using this scheme over the NRZ data format is increased channel bandwidth. Other encoding schemes are shown in Fig. 19.44.

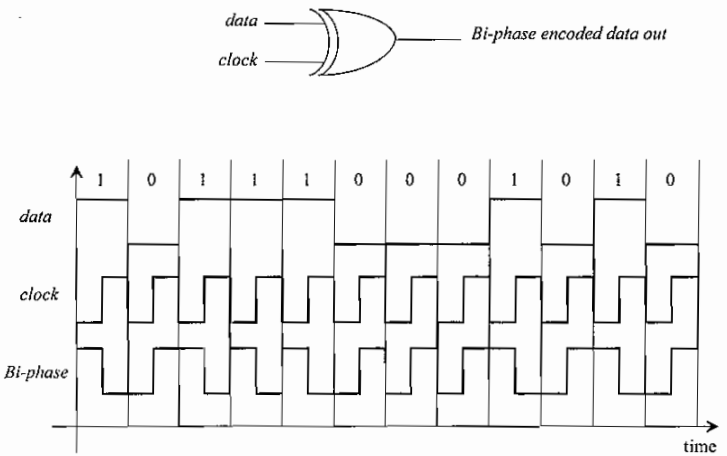


Figure 19.43 Bi-phase data encoding.

19.4.1 Clock Recovery from NRZ Data

One of the most important steps in the design of a communication system is the selection of the transmission format, that is, NRZ, bi-phase, or some other format, together with use of parity, cyclic redundancy code, or some other encoding format. In this section we discuss some of the considerations that go into the design of a clock-recovery DPLL in a system that uses NRZ.

Let's begin this discussion by considering the NRZ *data* and *clock* shown in Fig. 19.45. Let's further assume that these signals are the inputs to an XOR PD in a DPLL, which is not in lock since the clock is not aligned properly to the data. The resulting output of the XOR PD is shown in this figure as well. If we were to average this output using a loop filter, we would get *VDD*/2. In fact, it is easy to show that shifting the *clock* signal in time has no effect on the average output of the PD. Why? To answer this question, let's use some numbers. Assume that a bit width of *data* is 10 ns (which is also the period of the *clock*). The frequency of the square wave resulting from the alternating

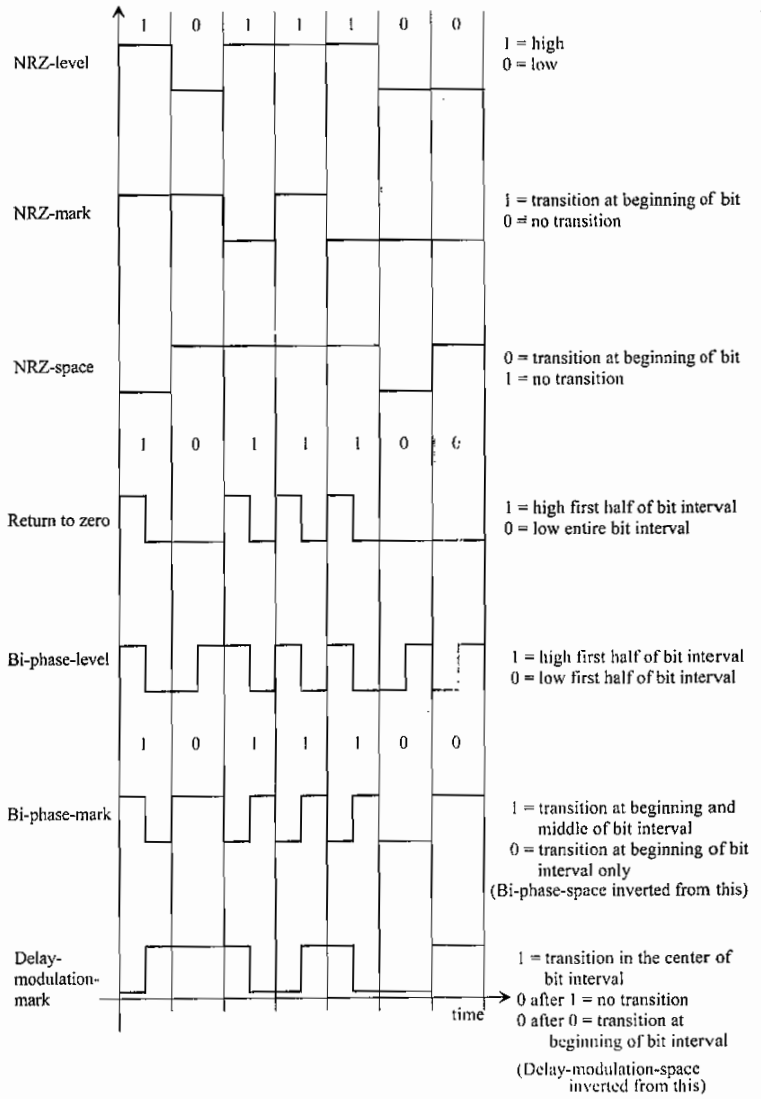


Figure 19.44 Data transmission formats.

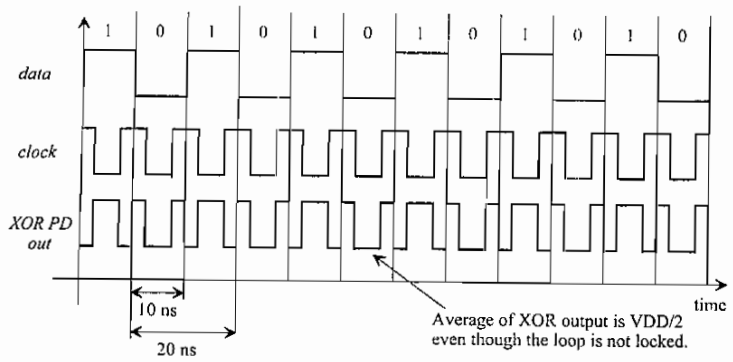


Figure 19.45 The problems of using clock without the divide by 2 to lock on data.

strings of ones and zeros is 50 MHz. We know that if we take the Fourier Transform of a square-wave, only the odd harmonics (i.e., 50, 150, and 250 MHz) are present. Since the *clock* signal is at 100 MHz, there is no energy or information common between the *clock* and *data* signals. To remedy this, we divide the clock down in frequency, *dclock*, so that it is at the same frequency as the alternating ones and zeros of the data (divide by two).

The next problem we encounter, if we use the divide by two in the feedback loop, occurs when we get *data* that is a repeating string of 2 ones followed by 2 zeroes, Fig. 19.46 (the *dclock* is not locked to the data). Again, there is no common information between the two inputs, and the resulting XOR PD output will always have an average of *VDD*/2. In this case, however, the *dclock* is running at 50 MHz, and the *data* is a square-wave of 25 MHz. Should we divide the *clock* down further to avoid this situation? The answer is no. However many times we divide the clock down, we can still come up

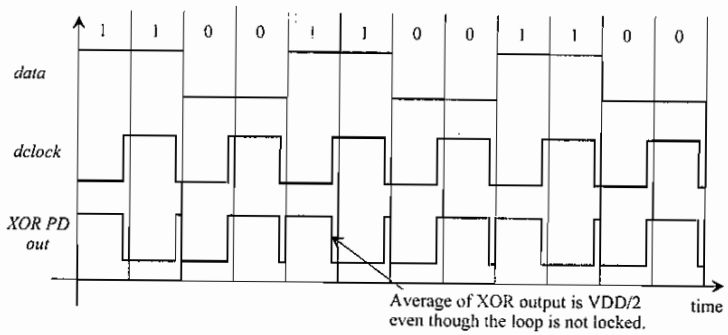


Figure 19.46 Problems trying to lock on a data stream that is one-half the dclock frequency.

with a data string that will not allow the loop to lock. Also, it is the actual edge transitions (the frequency of the *data* and *dclock*) that is used when the inputs are not pure squarewaves. Increasing the width of *dclock* has the effect of removing information and making it more difficult to lock up to the data. One solution to this problem is to use odd-parity with an 8-bit word (9 bits total) and eliminate, at the transmitting end, the possibility of all 8 bits being high, that is, 11111111 or 255. It is then impossible to generate a square-wave.

The restrictions on the data pattern in a communications system using NRZ data can be reduced by detecting the edges of the input data with an edge detector circuit (Fig. 19.47). The delay through the inverters sets the width of the output pulses, labeled "*Edge out*" in this circuit. The frequency content of the output pulses will always contain energy at the *clock* frequency and thus the loop can lock up on the data.

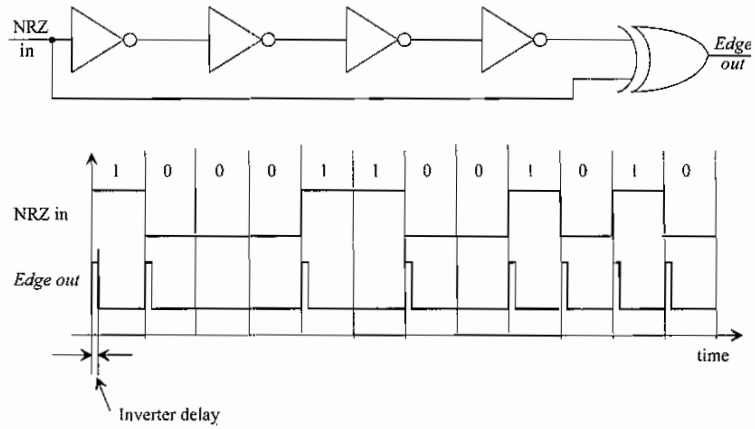


Figure 19.47 Detecting the edges in NRZ data.

As an example, consider the block diagram and data shown in Fig. 19.48. The *Edge output* is connected as the *input* of an XOR-based DPLL. The output of the VCO, *clock*, will lock up on the center of the *Edge output*, that is, the rising edge of the *clock* signal will become aligned with the center of *Edge out*. Averaging *PD out*, in this figure, results in *VDD*/2. If the clock is shifted to the left or right in time, the average value of *PD out* will shift down or up, causing the VCO frequency to change and keep the *clock* aligned to the center of *Edge out*. Several practical problems exist with this configuration. The delay through the inverters should be constant whether a high-to-low or a low-to-high transition is propagating through the inverters. Also, for best performance the delay of the inverters, or whatever element is used for the delay (one common element for high-speed applications is a microstrip line) should be close to one-half of the bit-interval time. This delay is important as it directly affects the gain of the phase detector and therefore the transient properties of the DPLL.

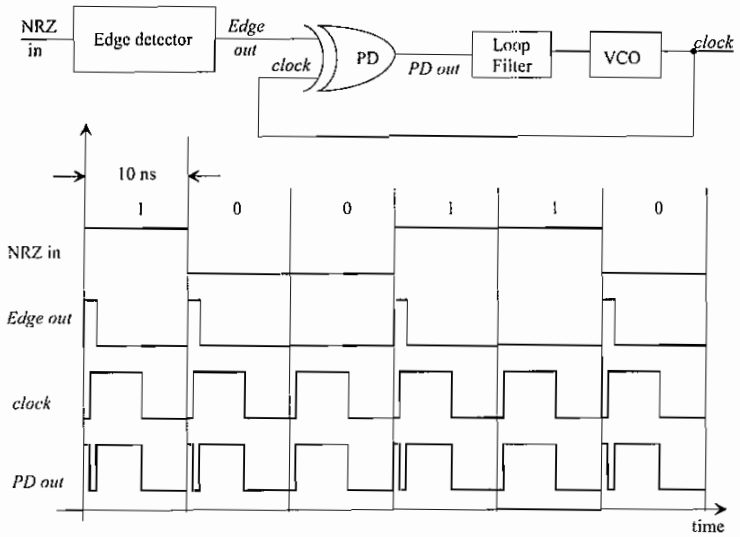


Figure 19.48 Clock-recovery circuit for NRZ using an edge detector. Note that the DPLL is in lock, when the rising edge of clock is centered on the edge output pulse.

Not having the *clock* aligned to the center of the data bit can cause problems in high-speed, clock-recovery circuits. Simply adding a delay in series with the *clock* signal does not solve this problem since the temperature dependence and process variations of the associated circuit do not guarantee proper alignment. What is needed is a circuit that is *self-correcting*, causing the clock signal to align to the center of the data bit independent of the data-rate, the temperature, or process variations.

The Hogge Phase Detector

The PD portion of a self-correcting, clock-recovery circuit is shown in Fig. 19.49 along with associated waveforms for a locked DPLL. Nodes A and B are simply the input NRZ data shifted in time by one-half bit-interval and one bit-interval, respectively. The outputs of the phase detector are labeled *Increase* and *Decrease*. If *Increase* is low more often than *Decrease*, the average voltage out of the loop filter and thus the frequency out of the VCO will decrease. A loop filter that can be used in a self-correcting DPLL is shown in Fig. 19.50a. This filter is the active-PI loop filter discussed in Sec. 19.3.1, with an added input to accommodate both outputs of the PD. Figure 19.50b shows the resulting waveforms in a DPLL where the clock is leading the center of the NRZ data, and thus *Increase* is high less often than *Decrease*. If the NRZ data was lagging the center of the data bit, *Decrease* would be high less often than *Increase*, resulting in an increase in the loop-filter output voltage. Note that in this discussion we have neglected the propagation delays present in the circuit. For a high-speed, self-correcting PD design, we would have to analyze each delay in the PD to determine their effect on the performance of the DPLL.

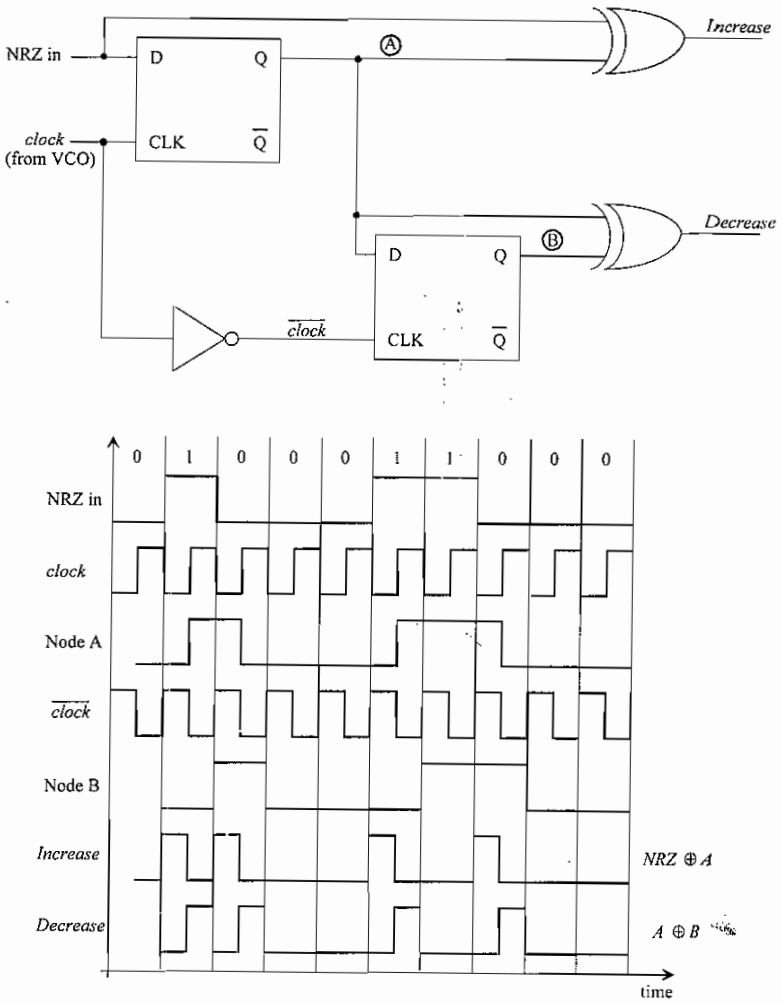


Figure 19.49 The PD (Hogge) portion of a self-correcting, clock-recovery circuit in lock.

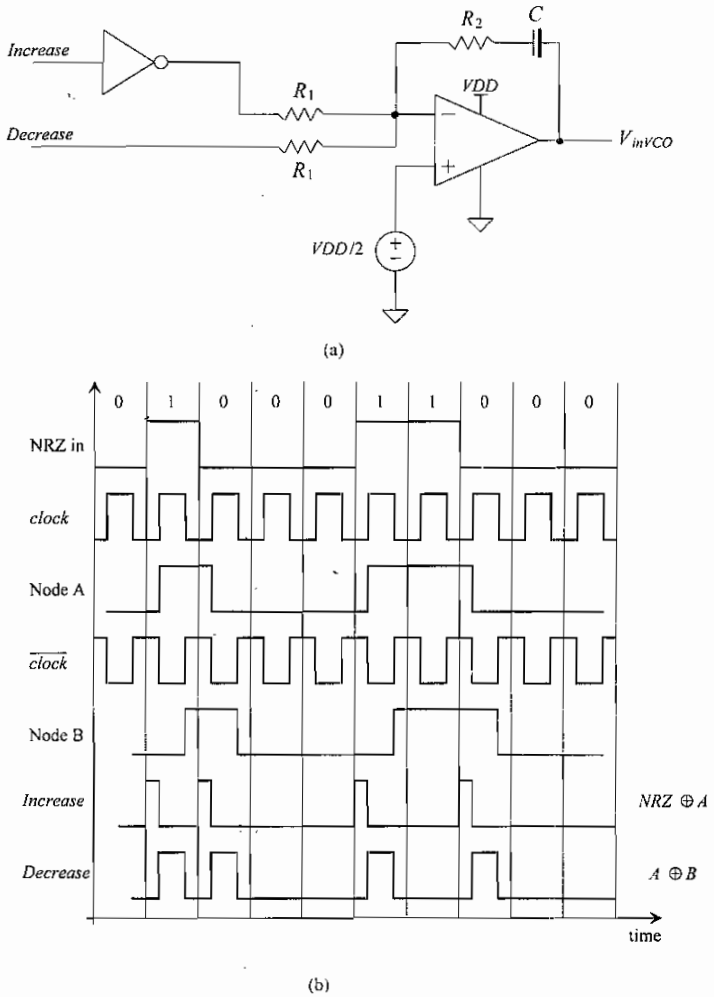


Figure 19.50 (a) Possible loop filter used in a self-correcting (Hogge) DPLL and (b) waveforms when the loop is not in lock and the clock leads the center of the data.

Jitter

Jitter, in the most general sense, for clock-recovery and synchronization circuits, can be defined as the amount of time the regenerated clock varies once the loop is locked. Figure 19.51a shows the idealized case when the *clock* doesn't jitter, while Fig. 19.51b shows the actual situation where the clock-rising edge moves in time (jitters). In these figures, the oscilloscope is triggered by the rising edge of the *data*. In the following discussion, we neglect power supply and oscillator noise, that is, we assume that the oscillator frequency is an exact number that is directly related to the VCO input voltage. In the section following this one, we cover delay-locked loops and further discuss the limitations of the VCO.

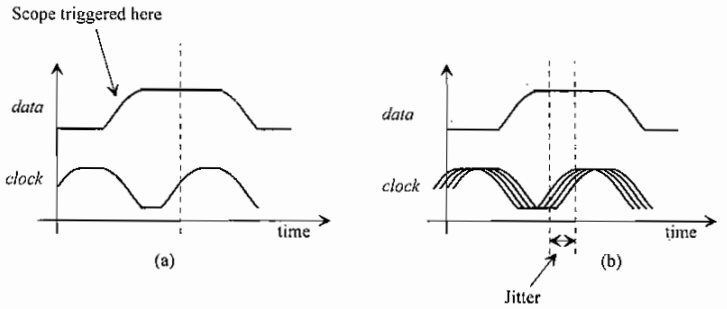


Figure 19.51 (a) Idealized view of clock and data without jitter and (b) with jitter.

Consider using the charge pump with the self-correcting PD shown in Fig. 19.52. When the loop is locked (from Fig. 19.49), both increase and decrease occur (with the same width) for every transition in the incoming data. Note that unlike a PFD/charge pump combination where the output of the charge pump remains unchanged when the loop is locked, the self-correcting PD/charge-pump combination generates a voltage ripple on the input of the VCO (similar to the XOR PD with RC or Active PI filter)[4]. Let's assume that this ripple is 10 mV and use the values for VCO gain and frequencies given in Ex. 19.5 to illustrate the resulting jitter introduced into the output clock. The change in output frequency resulting from this ripple is $10\text{mV}\cdot(1.57 \times 10^9 \text{ radians/V}\cdot\text{s})\cdot(1/2\pi)$ or 2.5 MHz. This means that the output of the DPLL will vary from, say, 100 MHz to 102.5 MHz. In terms of a jitter specification (see Eq. (19.41)), the clock jitter is (roughly) 250 ps (2.5% of the output clock's period).

From this example, *data dependent jitter*, can be reduced by

1. *Reducing the gain of the VCO.* Ripple on the input of the VCO has less of an effect on the output frequency. The main disadvantage, in the most general sense, of reducing the gain of the VCO is that the range of frequencies the DPLL can

[4] Of course, the self-correcting PD should not be used in most frequency synthesis applications. Similarly, the PFD should not be used in most clock-recovery applications.

lock up on is reduced. Also, the VCO gain strongly affects the ability to fabricate the VCO without postproduction tuning.

2. *Reducing the bandwidth of the loop filter.* This has the effect of reducing the actual ripple on the input of the VCO. The main drawback here is the increase in chip real estate needed to realize the larger components used in the filter.

3. *Reducing the gain of the PD.* This also has the effect of reducing the ripple on the input of the VCO. This method is, in general, the easiest when using the charge pump since reducing the gain is a simple matter of reducing I_{pump}. Substrate noise can become more of a factor in the design.

Another way of stating the above methods of reducing jitter is simply to say that the forward loop gain of the DPLL, that is, $K_{PD}K_FK_{VCO}$, should be made small. The main problems with using small forward loop gain are the reductions in lock range and pull-in range coupled with the associated increases in pull-in and lock times.

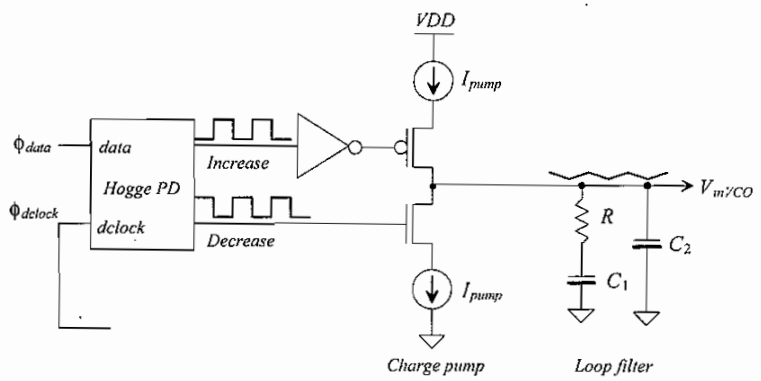


Figure 19.52 Self-correcting PD with charge pump output.

19.5 Delay-Locked Loops

Problems with PLL output jitter resulting from the VCO output frequency changing (often called oscillator or phase noise) with a constant input voltage (V_{inVCO} = constant) has led to the concept of a delay-locked loop (DLL). Figure 19.53 shows the basic block diagram of a DLL. *Assuming* that a reference clock is available at exactly the correct frequency, the input data is delayed through a voltage-controlled delay line (VCDL) a time t_o until it is synchronized with the reference clock. Jitter is reduced by using an element, the VCDL, that does not generate a signal (like the VCO did). The transfer function ϕ_{clock}/ϕ_{out} is zero (the phase of the reference clock is taken as the reference for the other signals in the DLL, i.e., $\phi_{clock} = 0$), so that oscillator noise and the resulting jitter are not factors in DLL design. The jitter considerations discussed in the last section, however, are still a concern since any ripple on the output of the loop filter will cause jitter.

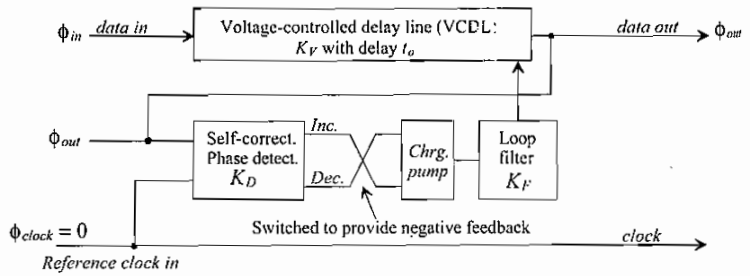


Figure 19.53 Block diagram of a delay-locked loop.

The phase (in radians) of the input data is related to the phase of the output data by

$$\phi_{out} = \phi_{in} + t_o \cdot \frac{2\pi}{T_{clock}} \tag{19.63}$$

where T_{clock} is the period of the reference clock (or half of the period of the *data in* for a string of alternating ones and zeros). The gain of the VCDL can be written in terms of the delay, t_o, by

$$t_o = K_V \cdot V_{indel} \tag{19.64}$$

where K_V has units of seconds/V and V_{indel} is the voltage input to the VCDL from the loop filter. The minimum and maximum delays of the VCDL should, in general, lie between $T_{clock}/2$ and $1.5T_{clock}$ for proper operation. The output of the loop filter (input to the VCDL) can be written as

$$V_{indel} = \phi_{out} \cdot K_D \cdot K_F \tag{19.65}$$

The overall transfer function may now be written as

$$\frac{\phi_{out}}{\phi_{in}} = \frac{1}{1 - K_D K_F K_V \cdot \omega_{clk}} \tag{19.66}$$

where $\omega_{clk} = 2\pi/T_{clock}$. The gain of the self-correcting PD with charge-pump output, with the help of Fig. 19.54 and noting that *Increase* and *Decrease* can occur at the same time, is

$$K_D = -\frac{I_{pump}}{\pi} \text{ (amps/radian)} \tag{19.67}$$

The negative sign is the result of switching the *Increase* and *Decrease* outputs of the self-correcting PD when connecting to the charge pump. This switch is required to provide negative feedback around the loop. Another benefit of the DLL is that the loop filter can be a simple capacitor, which results in a first-order feedback loop, that is,

$$K_F = \frac{1}{sC_1} \tag{19.68}$$

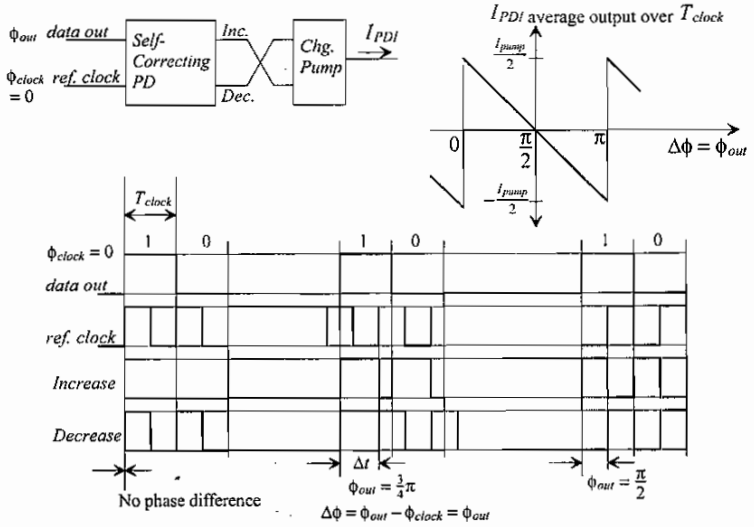


Figure 19.54 Self-correcting PD output for various inputs (assuming input data is a string of alternating ones and zeros).

The transfer function of the DLL relating the input data to the time-shifted output is

$$\frac{\phi_{out}}{\phi_{in}} = \frac{1}{1 + \frac{I_{pump}}{\pi} \cdot \frac{1}{sC_1} \cdot K_V \cdot \omega_{clk}} = \frac{s}{s + K_V \cdot \frac{2I_{pump}}{C_1 T_{clock}}} \tag{19.69}$$

We know that the frequency of the reference clock must be exactly related to the frequency of the input data. However, there will exist instantaneous changes in the phase of the input data which the output of the DLL should follow. Modeling instantaneous changes in ϕ_{in} by $\Delta\phi_{in}/s$ (a step function with an amplitude of $\Delta\phi_{in}$), we get a change in output phase given by

$$\Delta\phi_{out} = \frac{\Delta\phi_{in}}{s + K_V \cdot \frac{2I_{pump}}{C_1 T_{clock}}} \tag{19.70}$$

The time it takes the DLL to respond to an input step in phase is simply

$$T_r = 2.2 \cdot \frac{C_1 T_{clock}}{K_V \cdot 2I_{pump}} = \text{number of clock cycles} \cdot T_{clock} \tag{19.71}$$

This time can be decreased by making C_1/I_{pump} small, which from our discussion in the last section, has the result of increasing the output pulse jitter (i.e., jitter dependent on the input data pattern). Decreasing C_1/I_{pump}, increases the ripple on the control voltage of the VCDL. Similarly, increasing K_V (the time/volt delay of the VCDL) increases jitter since a given ripple on the control voltage of the VCDL will have a larger effect on the delay. Again, trade-offs must be made between responses to input variations and output jitter.

Delay Elements

The VCDL is an important component of the DLL. Figure 19.55a shows the basic implementation of a VCDL using adjustable delay inverters. The last two inverters in the VCDL ensure that clean digital signals are output from the line. Figure 19.55b shows the circuit schematics for possible delay elements. The first delay element should be recognized as a current-starved inverter discussed earlier in the chapter. The second delay element is nothing more than an inverter with a variable load. In practice, these delay elements are rarely used because of the susceptibility to noise and power supply variations. Rather, fully-differential delay elements are used.

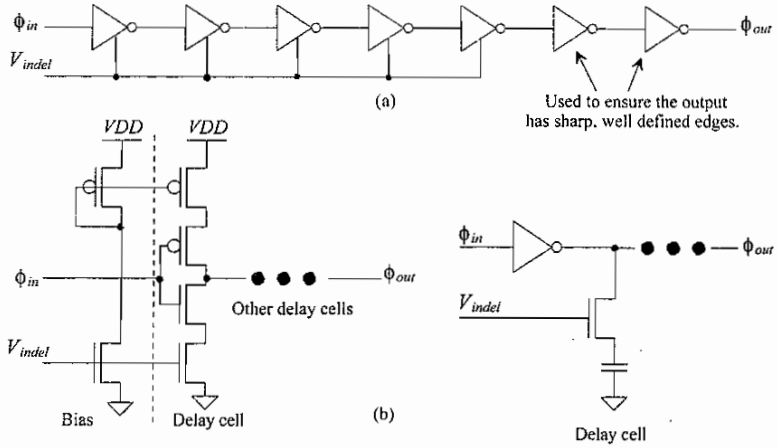


Figure 19.55 (a) VCDL made using inverter delay cells and (b) possible delay cells.

Figure 19.56 shows the connection of a fully differential VCDL. Using any number of stages, a fully differential VCO can be implemented with the delay elements (and the proper feedback), while using an even number with the inverting (noninverting) output fed back and connected to the noninverting (inverting) input in-phase (I) and quadrature (Q) signals can be generated.

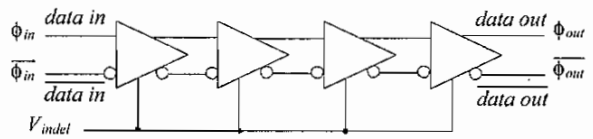


Figure 19.56 Implementation of a fully differential VCDL.

Practical VCO and VCDL Design

An example of a practical delay stage is shown in Fig. 19.57. The control voltage (that controls the delay if used in a VCDL or the oscillation frequency if the element is used in a VCO) can be generated from some linear voltage to current converter (like the one seen in Fig. 19.17 using a resistor, M5R, and M6R). The reason that this circuit is labeled "practical" can be understood by first considering what happens when there is noise on *VDD*. The noise causes a voltage variation across the PMOS current source, MPC. Ideally, this doesn't change the current through MPC. The output voltages of the delay cell swing between V_{REF} and ground. If there is noise on ground, it feeds equally into each output. This common-mode noise is then, ideally, rejected by the differential amplification action of the next stage. The bias circuit is a half-replica of the delay stage and is used to bias the delay elements so that the swing is up to V_{REF} when the gate of one of the PMOS switches is at ground.

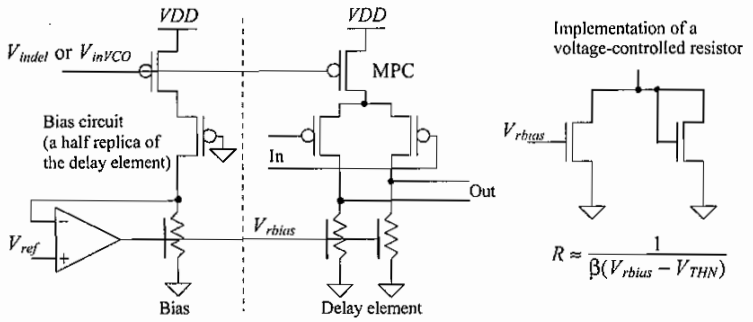


Figure 19.57 A differential delay element based on a voltage-controlled resistor. The bias circuit adjusts the value of the resistors used in the delay elements to sink the current sourced by the p-channel MOSFETs.

19.6 Some Examples

To finish up this chapter, let's present some examples with discussions and simulations.

19.6.1 A 2 GHz DLL

To begin, let's design a 500 ps delay line for a DLL that is used for *clock synchronization*. Let's design the delay line so that when it is used in a DLL, with the input and output of the delay line synchronized, we have eight phases of the input clock (we need eight stages in the delay line).

The first circuits we need to design are the bias circuit and the voltage-to-current converter used to generate $V_{outfilter}$ and V_{rbias} (see Fig. 19.57). We'll use the basic schemes of Fig. 19.17 and 19.57, as seen in Fig. 19.58. The important issues concerning this circuit are: 1) the linearity of the V_{indel} voltage against the generated I_{REF}, 2) the sensitivity of I_{REF} to changes in *VDD*, and 3) how well the amplifier regulates node n2 to V_{REF}.

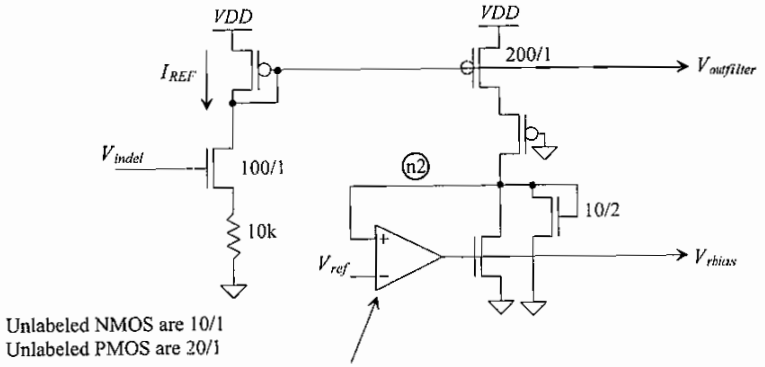


Figure 19.58 Bias circuit for a 500 ps delay line.

Figure 19.59 shows the simulation results for the bias circuit in Fig. 19.58. In (a) the x-axis is the delay line's control voltage, V_{indel}, while the y-axis is the generated reference current. Also seen in the figure, although hard to discern, is the change in *VDD* from 900 mV to 1.1 V (*VDD* is changed from 900 mV to 1.1 V in 50 mV steps, while V_{indel} is swept from 300 mV to 800 mV). Figure 19.59b shows how well node n2 is regulated to V_{REF}. V_{REF} can be generated with the beta-multiplier discussed in the next chapter (see Eq. (20.38) and Fig. 20.22).

Note that we aren't setting a lower value of current in this circuit like we did in Fig. 19.25. If we were to do this (which may be useful to reduce jitter since the resulting delay line or VCO would have lower gain), we would use a topology like the one seen in Fig. 19.60. The reference voltage is used to set the minimum current (when V_{indel} is small) through Rlow. The problem with setting the lower current with the method seen in Fig. 19.25 is that any changes in *VDD* feed directly across Rlow and change the bias current. Using the method in Fig. 19.60 with wide NMOS devices, the lower current is (roughly) $(V_{REF} - V_{THN})$/Rlow.

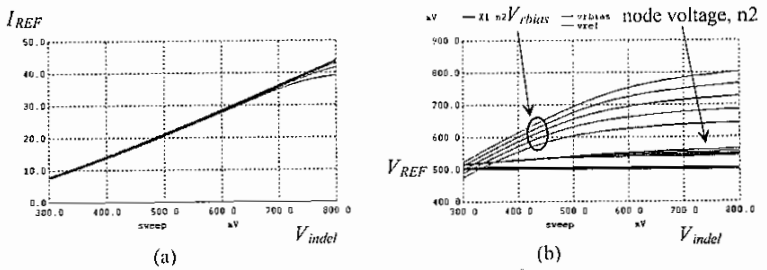


Figure 19.59 The performance of the bias circuit in Fig. 19.58.

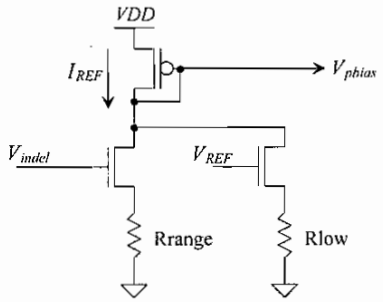


Figure 19.60 Lowering the current range in the bias circuit.

A schematic of the delay line is seen in Fig. 19.61. Notice that the current in the each delay stage was sized up by a factor of ten (the current sources in the delay cell are sized 200/1). This VCDL was simulated using the topology seen in Fig. 19.62. The control voltage was set to *VDD*/2 while the differential inputs were generated using an inverter and a transmission gate (to attempt to equalize the delays that the input signal sees to the VCDL). The outputs of the even stages are shown in this figure. Notice how they are evenly spaced. Notice, also, that the outputs only swing up to V_{REF} (= 500 mV here). As discussed earlier, this was done to minimize the effects of power supply and ground noise on the delay of the circuit.

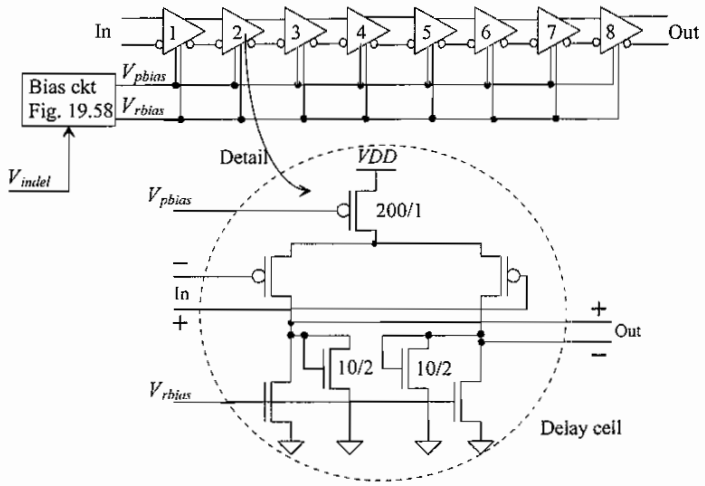
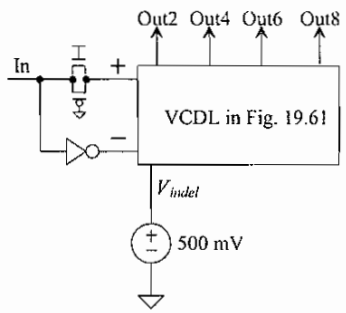


Figure 19.61 An eight-stage VCDL.



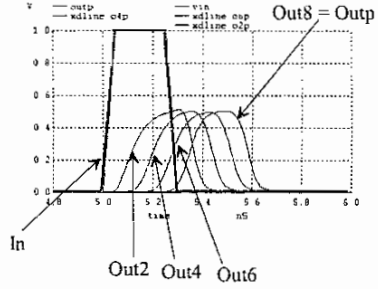


Figure 19.62 Simulating the VCDL in Fig. 19.61.

An important concern is how we regenerate full logic levels without introducing skew into our signals (see Sec. 18.3). Consider the modified VCDL seen in Fig. 19.63. Here we've changed the last stage to two diff-amps with swapped input signals (so we can generate the positive and minus outputs). Figure 19.64 shows the simulation results with this change. The shift in the eighth stage's output is small, in (a), and not much different than what is seen in Fig. 19.62. However, as needed, the output amplitude is larger. To get full logic levels, we pass these signals through inverters (that add more delay as seen Fig. 19.64b).

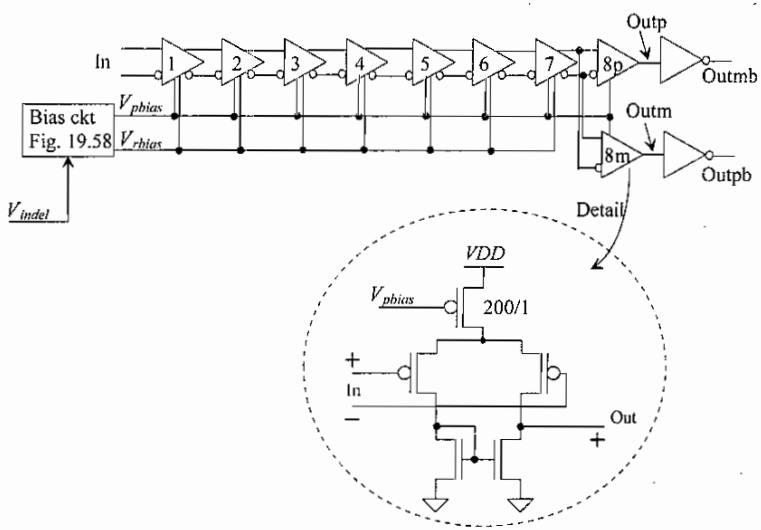


Figure 19.63 Modifying the VCDL to generate full output logic levels.

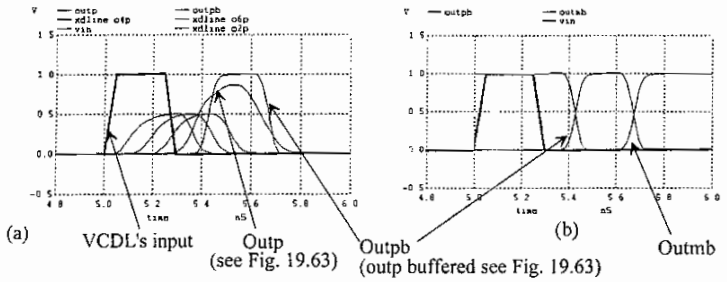


Figure 19.64 The operation of the VCDL in Fig. 19.63.

Using the simulations that generated Fig. 19.64, we can get a reasonable estimate for the VCDL's gain as

$$K_V = \frac{75\ ps}{100\ mV} = 750\ \frac{ps}{V}$$

The minimum delay through the VCDL is roughly 300 ps, while the maximum delay is infinite when V_{indel} moves towards V_{THN}. As seen in Fig. 19.60, we can set the minimum current that flows, I_{REF}, and thus the maximum delay. While this is an important concern we won't address it any further here.

For the PD in our DLL, let's use the PFD detector and charge pump from Ex. 19.5. The PFD's gain can be written as

$$K_D = \frac{I_{pump}}{2\pi} = \frac{10\ \mu A}{2\pi} = 1.59 \times 10^{-6}\ amps/radian$$

Using Eq. (19.71) with a lock time of 50 cycles gives

$$C_1 = \frac{(750 \times 10^{-12}) \cdot 2 \cdot (10 \times 10^{-6}) \cdot 50}{2.2} = 340\ fF$$

As indicated in the discussion following Eq. (19.71), however, this capacitor's selection is based, in some designs, not on lock time but rather the allowable jitter in the output signal once the loop is locked. A block diagram of our DLL is seen in Fig. 19.65. Simulations show that, indeed, the loop locks within 50 cycles with this value of loop filter capacitor (340 *fF*). However, the ripple on the control voltage is 20 mV. With the VCDL gain given above, we can estimate the output jitter as 15 ps. Since the period is 500 ps, this is 3% of the period. This much jitter, in a general application, may not be too bad. When we derived Eq. (19.71), we assumed that the gain of the VCDL was linear. If it is not linear the DLL can exhibit some second-order locking effects (meaning the response isn't truly first-order), resulting in a static phase error or a dead zone (the loop doesn't lock tightly to the input signal). To reduce the jitter and to linearize the loop's response, let's use a large capacitor for the loop filter (a 5 pF) in this example.

The simulated input and output signals of the DLL in Fig. 19.65 using a 5 pF capacitor are seen in Fig. 19.66. Notice that the static phase error. The center of the rising edge of the input signal isn't exactly occurring at the same time as the center of the rising edge of the output signal. Further, notice the asymmetry in the output signal, which stays

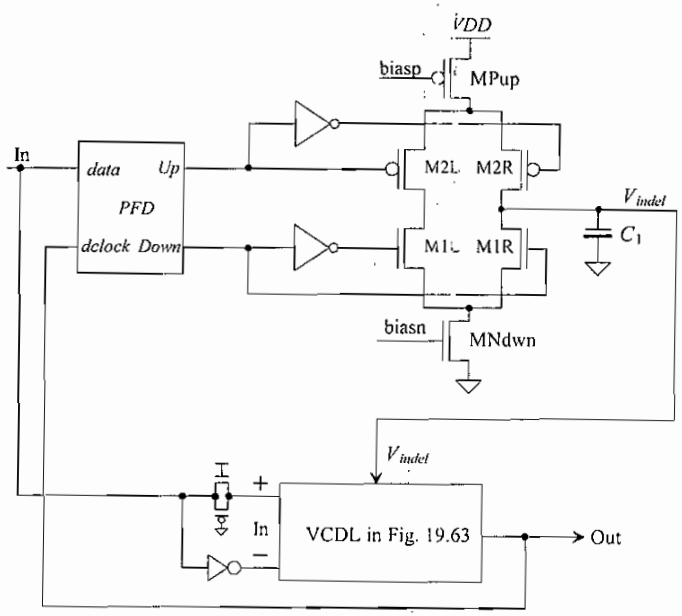


Figure 19.65 Block diagram of a DLL.

high longer than it stays low. This skew, as discussed in Sec. 18.3, is caused by different drive strengths of NMOS and PMOS devices or by differing slopes of input signals. We used the half-replica bias circuit in Fig. 19.58 to make the delay line more tolerant of power supply and ground noise. The drawback is that, at the end of the line, we do have to generate full logic levels. This causes skew between the final output of the VCDL (stage 8 in Fig. 19.63) and the outputs of the other stages in the line. The simple solution

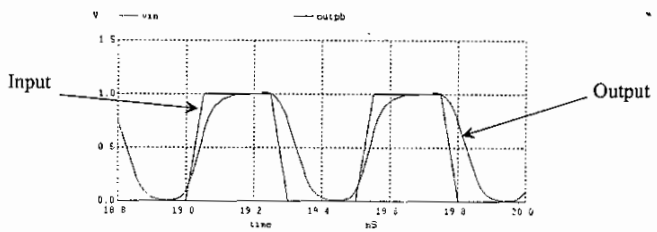


Figure 19.66 Input and output of the DLL in Fig. 19.65.

to this problem is to use a delay stage where the outputs swing to full logic levels. To reduce the DLL's susceptability to power supply and ground noise, we can power it with its own on-chip regulated power supply, supply power and ground to the DLL from separate *VDD* and ground pads, or use some simple filtering to ensure the noise on *VDD* is slow enough that the DLL can respond and correct for variations in the VCDL's delay circuits. For example, we might, in Fig. 19.61, reduce the number of stages to four and add, on the output of each of the delay cells seen in this figure, two of the PMOS buffers seen in Fig. 18.21. We need two buffers because we swap the inputs of each buffer to generate both true and compelement output logic levels. An example of this delay cell is seen in Fig. 19.67. We only use four stages now (for the 500 ps delay line in this section) because of the extra capacitive loading on the output of each stage. The buffered delay cell outputs drive an external load while the basic delay cell outputs are used to drive the input of the next delay cell in the VCDL (only).

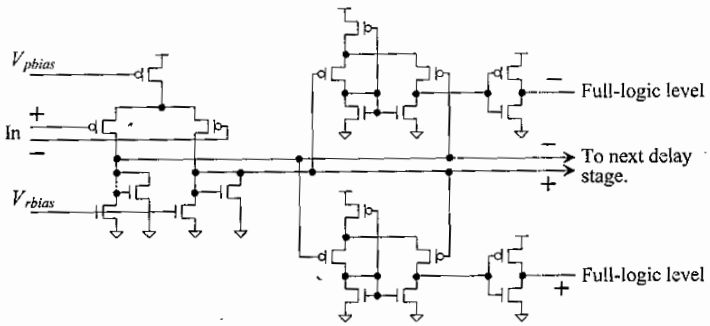


Figure 19.67 Generating full-logic levels in each delay stage.

19.6.2 A 1 Gbit/s Clock-Recovery Circuit

As another example, let's discuss and simulate the design of a 1 Gbit/s clock recovery circuit using the NRZ data format. Let's use the VCDL we've already developed in the VCO (see Fig. 19.68). The output of the VCDL is fed back to its input (with an inversion) to get the positive feedback needed for oscillations. Note that the VCO oscillates at 1 GHz when V_{inVCO} is (roughly) 350 mV (which is not *VDD*/2). While, in a production part, centering the VCO's operating frequency is important, we don't modify the VCDL in Fig. 19.63 for the PLL design here.

We will be using the Hogge phase detector in this design. Since the Hogge PD doesn't perform frequency detection (like the PFD), we must be concerned with locking on harmonics. For example, we might have an input NRZ data stream of 1000 0000. Our loop locks with the clock centered on the one and then six other edges centered on the seven zeroes. The loop is locked just not at the right clock frequency (it's locked at 7/8 of the correct frequency). While this is an **important practical** concern, we won't discuss it further until the end of the section. We will use initial conditions on the loop filter to set the initial clock frequency close to the correct value to avoid locking on a harmonic.

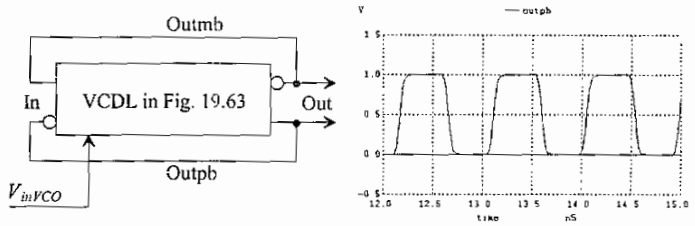


Figure 19.68 Making a VCO with the delay line in Fig. 19.63 and its output when Vinvco = 350 mV.

We know that the gain of the VCDL is 750 ps/V. Since, for one complete oscillation of the VCO, the signal must travel through the VCDL twice, we can estimate the gain of the VCO as

$$K_{VCO} = 2\pi \cdot \frac{1.175\ GHz - 1\ GHz}{100\ mV} = 11 \times 10^9\ radians/V \cdot s$$

where we assume at 350 mV the output frequency is 1 GHz (period is 1 ns). If we increase V_{inVCO} by 100 mV, we get a decrease in the period by 150 ps ($2 \times 100\ mV \cdot 750\ ps/V$) resulting in an output frequency of 1.175 GHz (period is 850 ps).

To implement the Hogge PD seen in Fig. 19.49, we'll use the same TSPC edge-triggered latch that we used for the divide-by-2 seen in Fig. 19.24. The result is seen in Fig. 19.69. We've been very careful to buffer the inputs and outputs of the PD to ensure high-speed output edges and to square up the input signals. To verify the operation, we've simulated the PD with the NRZ and clock signals seen in Fig. 19.49, Fig. 19.70. Looking at the *increase* and *decrease* signals, we see that the width of the *increase* signal is too large. On careful inspection, the cause of this error is the delay through the DFF the NRZ data sees to node A. To compensate for this delay, we'll add a delay in parallel with this path to the XOR gate, as seen in Fig. 19.71. Figure 19.72 shows the resulting simulation output. Again note that unlike the PFD, which only looks at the edges, the Hogge PD, like the XOR PD, uses the widths of its inputs to drive the loop filter.

When the Hogge PD is used with a charge pump the gain is, see Eq. (19.67),

$$K_D = \frac{I_{pump}}{\pi}$$

If we use the charge pump from the DLL in Fig. 19.65 with I_{pump} = 10 μA, the gain of the PD is 3.2 μA/radian. If we set the natural frequency, ω_n, to, 100×10^6 radians/s and ζ = 1, then using Eq. (19.58)

$$RC_1 = 20\ ns$$

and using Eq. (19.57) we can solve for C_1 (knowing that there isn't a feedback divider so N is 1) as

$$C_1 = \frac{K_D K_{VCO}}{\omega_n^2} = \frac{(3.2 \times 10^{-6})(11 \times 10^9)}{(100 \times 10^6)^2} = 3.5\ pF$$

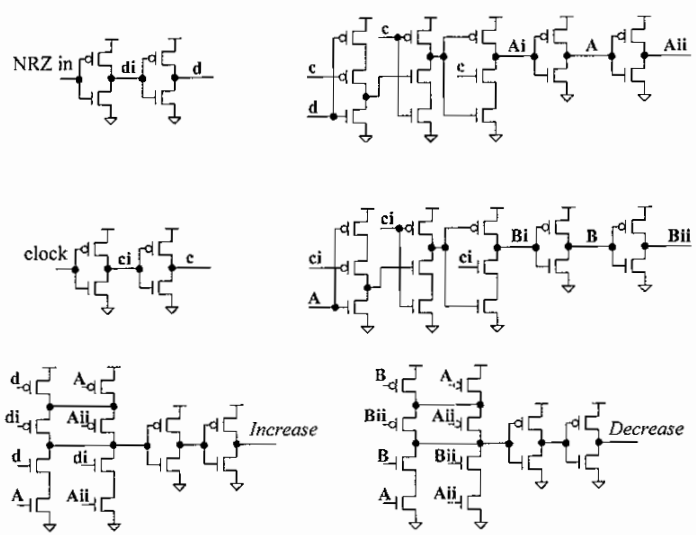


Figure 19.69 CMOS implementation of the Hogge PD.

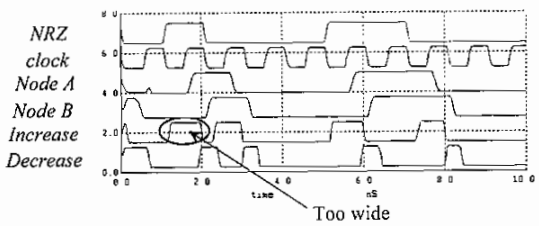


Figure 19.70 Simulating the Hogge PD in Fig. 19.69. These results should be compared to Fig. 19.49. Notice how the increased pulse widths are too wide. The result is that the Vinvco control voltage will increase above the desired value (and cause a static phase error or false locking).

Let's set C_1 to 3.5 pF, R to 10k, and C_2 to 0.35 pF. The schematic of the complete DPLL clock-recovery circuit is seen in Fig. 19.73.

In the first simulation, Fig. 19.74, we apply an alternating string of ones and zeroes to the PLL clock-recovery circuit. Since the data rate is 1 Gbit/s, the width of a one or a zero is 1 ns. In this simulation we didn't apply any initial conditions to the loop filter to start the simulation out. However, because the NRZ data is full of transitions, V_{inVCO} quickly moves to the correct voltage (around 350 mV). The important thing to note is that, again, the lock time depends on the input data.

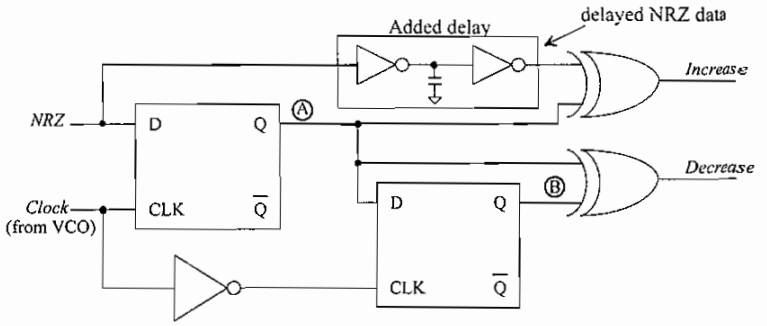


Figure 19.71 Adding a delay to the Hogge PD to compensate for the delay the NRZ data sees to node A.

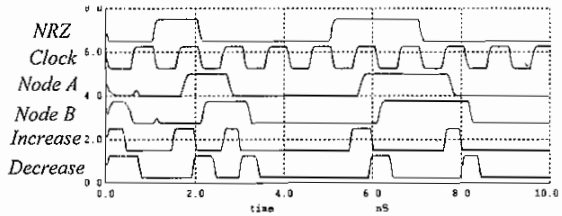


Figure 19.72 Resimulating the operation of the Hogge PD with the delay seen in Fig. 19.71 included.

For the second simulation, Fig. 19.75, we start the voltage across the loop filter, V_{inVCO}, out at 340 mV (slightly below the final value of around 350 mV). After approximately 600 ns, the loop locks. Had we not used this initial condition, the lock time (and the simulation time required to attain lock) would be quite long. Because of the large gain of the VCO, not using an initial voltage for V_{inVCO} would result in the VCO oscillating at the wrong frequency and the loop not locking correctly (locking on the wrong frequency) when the data is, as seen, a single one followed by seven zeroes (or some other sparse pattern). To avoid this problem, we can require the PLL sees a long pattern of alternating ones and zeroes to ensure lock prior to sending data (called sending a preamble) or we can reduce the gain of the VCO. The problem with sending a preamble is that it reduces the data rate through the channel. Further, large VCO gain makes it easier for the PLL to lose lock with steps in the phase shift through a communication channel. The practical solution is to reduce the gain of the VCO. In this example if we were to limit the VCO's oscillation range to, say, 950 MHz to 1050 MHz, then (with proper design of the loop filter) the PLL could acquire lock quickly and stay locked with reasonable variations in the phase shifts of the input NRZ data (these phase shifts are common in channels whose path length is not fixed, as in wireless communications). The practical problem with low VCO gain, again, is manufacturability. It's impossible to

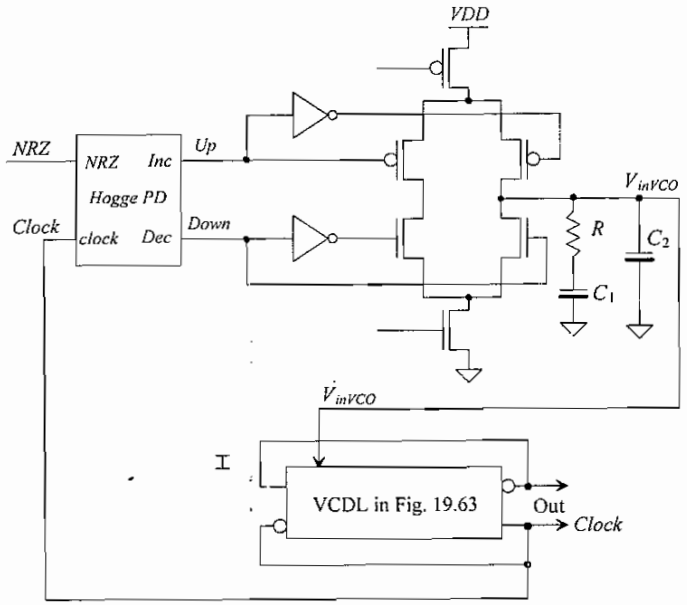


Figure 19.73 Block diagram of the DPLL used for clock-recovery discussed in this section.

design a purely monolithic CMOS oscillator (meaning no off-chip components) without some sort of hand tuning (meaning using fuses or some other sort of digital adjustment to center the oscillation frequency and limit the gain after the chips are made). Further, even if we hand-tailor the performance of each VCO in a production line, the temperature and power supply sensitivity will limit the minimum gain we can attain. Tuned circuits

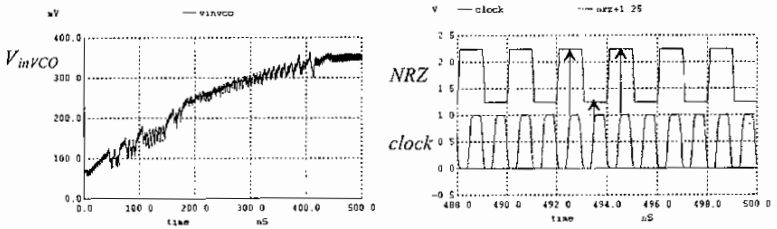


Figure 19.74 Simulating the PLL in Fig. 19.73 when the input NRZ data is an alternating string of ones and zeroes.

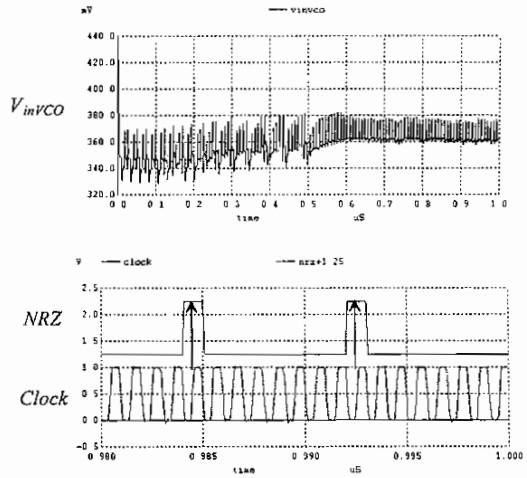


Figure 19.75 Simulating the PLL in Fig. 19.73 when the input data is a string of seven zeroes followed by a single one.

(parallel inductor-capacitor "tanks") are commonly used to help make purely monolithic clock-recovery circuits manufacturable. However, hand-tuning during Probe is still common to set the oscillation frequency and oscillation range.

ADDITIONAL READING

[1] R. E. Best, *Phase-Locked Loops, Theory, Design and Applications,* McGraw-Hill, 2nd ed., 1993. ISBN 0-07-911386-9. Excellent book covering PLL design.

[2] D. H. Wolaver, *Phase-Locked Loop Circuit Design*, Prentice-Hall, 1991. ISBN 0-1366-2743-9. Another excellent book covering PLL design.

[3] W. F. Egan, *Phase-Lock Basics*, John Wiley and Sons, 1998. ISBN 0-4712 -4261-6 Good introduction to PLLs. See also the same author's *Frequency Synthesis by Phase Lock.*

[4] F. M. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Transactions on Communications,* COM-28, no. 11, pp. 1849–1858, November 1980. The paper first discussing charge pump PLLs.

[5] B. Razavi, *Monolithic Phase-Locked-Loops and Clock Recovery Circuits,* IEEE Press, 1996. ISBN 0-7803-1149-3. Good tutorial and collection of papers.

[6] S. Haykin, *An Introduction to Analog and Digital Communications,* John Wiley and Sons, 1989. ISBN 0-471-85978-8. Covers distortionless transmission.

[7] C. R. Hogge, Jr., "A Self Correcting Clock Recovery Circuit," *IEEE Journal of Lightwave Technology*, vol. LT-3, pp. 1312–1314, December 1985. Paper presenting the "Hogge" phase detector.

[8] M. G. Johnson and E. L. Hudson, "A Variable Delay Line PLL for CPU - Coprocessor Synchronization," *IEEE Journal of Solid-State Circuits,* vol. SC-23, pp. 1218–1223, October, 1988. Classic paper presenting the concept of a delay-locked loop.

[9] I. A. Young, J. K. Greason, and K. L. Wong, "A PLL Clock Generator with 5 to 110 MHz of Lock Range for Microprocessors," *IEEE Journal of Solid-State Circuits,* vol. SC-27, pp. 1599–1607, November 1992. Presents the practical design of a CMOS delay element.

PROBLEMS

19.1 The XOR gate PD seen in Fig. 19.4 can exhibit input-dependent skew. In other words, the delay from one of the inputs changing to the output of the PD will not be precisely the same as the delay from the other input to the output. Design an XOR PD that doesn't exhibit input-dependent skew. Hint: see Fig. 18.16.

19.2 Verify, using simulations, that a locked PLL using an XOR PD will exhibit, after RC filtering, an average value of *VDD*/2. Show, using simulations and hand calculations, the filter's average output if the XOR PD sees a phase difference in its inputs of $-\pi/4$.

19.3 Why, in your own words, is it so important for the VCO's center frequency to match the input NRZ data rate when a passive loop filter is used. Use simulations to show the problems. Why does using the active loop filter eliminate this requirement?

19.4 Suppose, for a robust high-speed PLL using an XOR PD, that it desirable to adjust the PD's gain. Show a charge pump can be used towards this goal. Should the loop filter have two outputs? Discuss the PD's gain and how it would be adjusted. What topology would be used for a passive loop filter? for an active loop filter?

19.5 Demonstrate, using simulations, how the outputs of the XOR PD can be equivalent when the loop is true or false locking (locking on a harmonic).

19.6 Describe, in your own words and using simulations, what the dotted line in Fig. 19.8 indicates.

19.7 We know that a PFD isn't generally used in clock recovery applications because both edges (*data* and *dclock*) must present when doing a phase comparison. Suggest a scheme where the edge of the input *data* enables a phase comparison. When an edge of data is not present your circuit will "swallow" the pulse from *dclock* resulting in no edges being applied to the PFD (no phase- frequency comparison).

19.8 Demonstrate, using simulations, charge sharing between the charge pump and loop filter using the topology seen in Fig. 19.12b (and Fig. 19.13b). Show how the topology in Fig. 19.37 helps with charge sharing (simulate with and without the x1 amplifier). For the x1 amplifier, use the n-type diff-amp from Fig. 18.17 without the inverter on its output. Connect the loop filter to the gate of M1 (in the diff-amp). To make the gain "1," tie the output (which is connected to drains of M1L/M2L in Fig. 19.37) of the diff-amp (the drains of M2/M4) to the gate of M2.

19.9 Using the VCO that generated the simulation data in Fig. 19.18, plot the VCO's center frequency against changes in *VDD*. Is this VCO insensitive to changes in *VDD*? Where does the sensitivity come from?

19.10 Discuss, and demonstrate with simulations, how to reduce the gain of the VCO used to generate the data in Fig. 19.18 (see Fig. 19.25 and the associated discussion). Your discussion should include some insight into the manufacturability of low-gain VCOs.

19.11 Design and simulate the operation of a 100 MHz VCO using the topology seen in Fig. 19.19a. The output of your VCO should be full logic levels. Your design should show a linearly frequency against V_{inVCO} curve. What is the gain of your design?

19.12 Using the step response of an RLC circuit, Fig. 19.76, demonstrate how selection of the resistor, inductor, and capacitor affect the output voltage's damping factor and natural frequency. From this plot show how natural frequency and lock time are related.

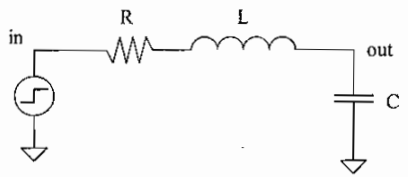


Figure 19.76 Using a second-order circuit to demonstrate how lock time and natural frequency are related.

19.13 Replace, in the DPLL that generated the waveforms in Figs. 19.27 and 19.28, the simple RC loop filter with the passive lag loop filter in Fig. 19.29. Show how the filter's component values are selected. Comment on, using simulations to support your conclusions, how the performance of the DPLL is affected.

19.14 Using the PFD in Fig. 19.33 and the charge pump in Fig. 19.36, demonstrate, with simulations, how the gain of PFD/charge-pump configuration can have a dead zone (the gain, K_{PDI}, decreases) as seen in Fig. 19.77.

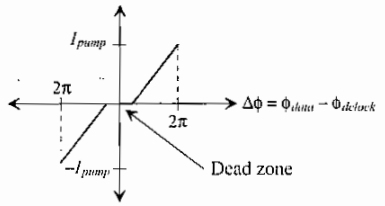


Figure 19.77 Showing the dead zone in a PFD charge-pump.

19.15 Suppose that the pump currents used in the DPLL in Ex. 19.5 are mismatched by 50%. Will this cause a static phase error? Why or why not? Use simulations to support your answer.

19.16 Redesign the DPLL in Ex. 19.5 so that the output remains a 100 MHz square wave signal when the input signal is changed to 25 MHz.

19.17 We used pF values for the capacitors in the loop filters in this chapter. Why not reduce the filter's layout area by using fF size capacitors? Demonstrate the problems with using such small loop filter capacitors.

19.18 What are we sacrificing by using an equalizer? What does the minus sign indicate in the slope of the phase in Fig. 19.41?

19.19 Using the DPLL from Ex. 19.2, demonstrate the false locking as seen in Fig. 19.46.

19.20 Design an edge detector, like the one seen in Fig. 19.47, for use in the DPLL in Ex. 19.2. Regenerate the simulation data seen in Figs. 19.27 and 19.28. Remember to eliminate the divide by two in the feedback path.

19.21 Derive Eq. (19.71).

19.22 Design a nominal delay line of 1 ns (when V_{indel} = 500 mV) using the current starved delay element seen in Fig. 19.55. Determine the delay's sensitivity to variations in *VDD*.

19.23 Repeat Problem 19.22 for the inverter delay cell in Fig. 19.55.

19.24 Suppose, as seen in Fig. 19.78, that instead of using a half-replica of the delay cell in Fig. 19.58, the full delay cell is used to generate V_{rbias}. Electrically is there any difference in V_{rbias} when comparing the full- and half-replica circuits? What may be the benefit of using a full-replica of the delay cell?

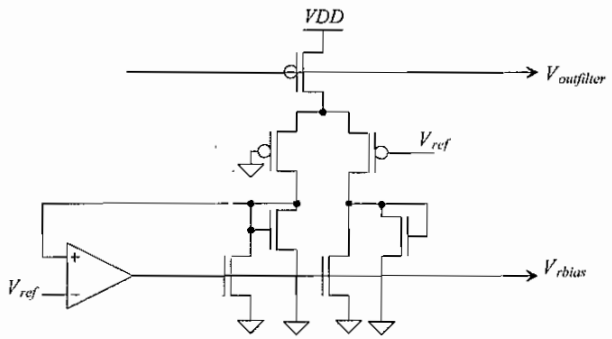


Figure 19.78 Using a full-replica of the delay element for generating Vrbias.

19.25 For the delay line that generated the simulation results in Fig. 19.62 determine, using simulations, the delay's sensitivity to changes *VDD*. Plot the VCDL's delay as a function of *VDD* with V_{indel} held at 500 mV.

19.26 The delay element seen in Fig. 19.79 doesn't use a reference voltage. Show how this element can be used in the VCDL of Fig. 19.61. Note that the input capacitance of this delay stage is twice as large as the input capacitance of the element in Fig. 19.61 (and so you may have to use fewer stages to attain the same overall delay). How does the NMOS gate potential change with the inputs/outputs switching? Why? Is the output amplitude a function of *VDD*?

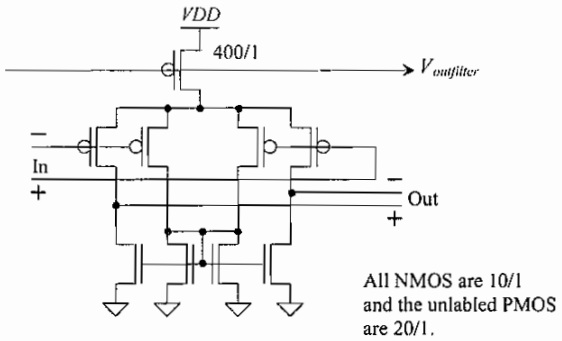


Figure 19.79 A delay element that doesn't use a reference voltage.

19.27 Use the delay element in Fig. 19.67 to implement a VCDL. Use the VCDL in the DLL seen Fig. 19.65 to generate the waveforms in Fig. 19.66. Show the 90, 180, 270, and 360 degree outputs of the VCDL swinging to full-logic levels.

19.28 The VCO output seen in Fig. 19.68 doesn't have an exactly 50% duty cycle. Will this affect a clock-recovery circuit's operation? Why or why not? Use simulations to support you answer.

19.29 Suggest another method, other than the one seen in Fig. 19.72, to equalize the widths of the *Increase* and *Decrease* outputs of the Hogge PD. Verify your design with simulations.

19.30 Must the currents in the charge pump in Fig. 19.73 be equal for proper DPLL clock-recovery operation? Why or why not? What happens if the currents aren't equal? What happens if the NMOS switches turn on at different speeds than the PMOS switches? Use SPICE to support your answers.

19.31 Modify the simulation inputs in Fig. 19.74 to show false locking. Comment on what can be done in a practical clock-recovery circuit to eliminate false locking.

19.32 Replace the charge pump used in the DPLL in Fig. 19.73 with the active-PI loop filter seen in Fig. 19.50. Calculate the loop filter component values. Using the new loop filter, regenerate Figs. 19.74 and 19.75.

Chapter
20

Current Mirrors

In this chapter we turn our attention towards the design, layout, and simulation of current mirrors (a circuit that sources [or sinks] a constant current). As we observed back in Fig. 9.1, and the associated discussion, the ideal output resistance, r_o, of a current source is infinite. Achieving high output resistance (meaning that the output current doesn't vary much with the voltage across the current source) will be the main focus of this chapter.

It's very important that the reader first understand the material in Ch. 9 concerning the selection of biasing currents and device sizes and how they affect the gain/speed of the analog circuits. We'll use the parameters found in Tables 9.1 and 9.2 in many of the examples in this chapter.

20.1 The Basic Current Mirror

The basic NMOS current mirror, made using M1 and M2, is seen in Fig. 20.1. Let's assume that M1 and M2 have the same width and length and note that $V_{GS1} = V_{DS1} = V_{GS2}$. Because the MOSFETs have the same gate-source voltages, we expect (neglecting channel-length modulation) them to have the same drain current. If the two resistors in the drains of M1/M2 are equal, the drain of M2 will be at the same potential as the drain of M1 (this is important). By matching the size, V_{GS}, and I_D of two transistors, we are assured that the two MOSFETs have the same drain-source voltage, $(V_{GS1} = V_{DS1} = V_{GS2} = V_{DS2})$.

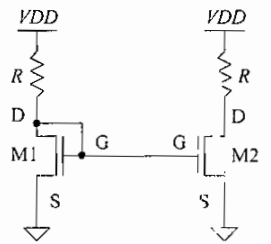


Figure 20.1 A basic current mirror.

20.1.1 Long-Channel Design

Examine Fig. 20.2. In this figure we show a current mirror and the equivalent circuit representation of a current source. Looking at M1, we can write

$$I_{REF} = I_{D1} = \frac{KP_n}{2}\frac{W_1}{L_1}(V_{GS1} - V_{THN})^2(1 + \lambda(V_{DS1} - V_{DS1,sat})) \qquad (20.1)$$

knowing $V_{DS1} = V_{GS1}$ and $V_{DS1,sat} = V_{GS1} - V_{THN}$. For M2, we write

$$I_O = I_{D2} = \frac{KP_n}{2}\frac{W_2}{L_2}(V_{GS1} - V_{THN})^2(1 + \lambda(V_O - V_{DS1,sat})) \qquad (20.2)$$

noting $V_{GS1} = V_{GS2}$, $V_{DS1,sat} = V_{DS2,sat}$, and V_O is the voltage across the current source. Looking at the ratio of the drain currents, we get

$$\frac{I_O}{I_{REF}} = \frac{W_2/L_2}{W_1/L_1}\cdot\frac{1 + \lambda(V_O - V_{DS1,sat})}{1 + \lambda(V_{DS1} - V_{DS1,sat})} \qquad (20.3)$$

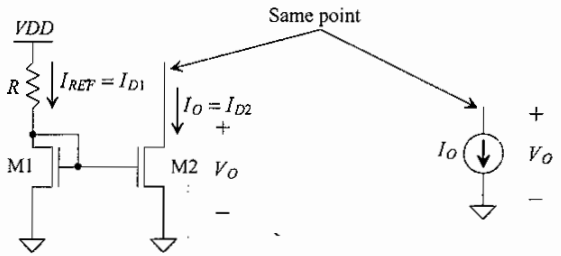


Figure 20.2 The current mirror and how we think about it.

Generally, the lengths of the devices in the current mirror are equal (let's assume they are for the moment). If, also at this time, we don't concern ourselves with channel-length modulation ($\lambda = 0$), we get a very useful result, that is,

$$\frac{I_O}{I_{REF}} = \frac{W_2}{W_1} \qquad (20.4)$$

By simply scaling the width of M2, we can adjust the size of our output current. Figure 20.3 shows an example of this scaling (using PMOS devices).

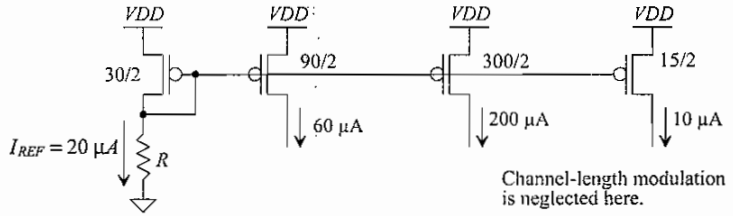


Figure 20.3 How current mirrors are ratioed.

Example 20.1

Determine the value of the resistor, R, needed in Figs. 20.2 and 20.3 so that the reference drain currents are 20 µA. Use the long-channel parameters seen in Table 9.1. Simulate the operation of the NMOS mirror with the calculated resistor value.

In Fig. 20.2 (the NMOS current mirror), we can write

$$I_{REF} = 20\ \mu A = \frac{VDD - V_{GS1}}{R} \approx \frac{KP_n}{2}\cdot\frac{10}{2}\cdot\left(\overbrace{V_{GS1}}^{1.05\ V} - \overbrace{V_{THN}}^{0.8}\right)^2 = \frac{KP_n}{2}\cdot\frac{10}{2}\cdot(0.25)^2$$

(notice how we indicate "approximately" because channel-length modulation is not included in the equations) or

$$R = \frac{5 - 1.05}{20\ \mu A} \approx 200\ k\Omega$$

For Fig. 20.3 (the PMOS current mirror), we can write

$$I_{REF} = 20\ \mu A = \frac{VDD - V_{SG}}{R} \approx \frac{KP_p}{2}\cdot\frac{30}{2}\left(\overbrace{V_{SG}}^{1.15\ V} - \overbrace{V_{THP}}^{0.9}\right)^2 = \frac{KP_p}{2}\cdot\frac{30}{2}\cdot(0.25)^2$$

or

$$R = \frac{5 - 1.15}{20\ \mu A} \approx 200\ k\Omega$$

Simulation of the operation of the NMOS current mirror is seen in Fig. 20.4. The reference current isn't exactly 20 µA (and we shouldn't expect it to be). The x-axis is a sweep of the voltage across the current source, V_O. Note that below $V_{DS,sat}$ (= 250 mV here) M2 triodes and the output current, I_O, goes to zero. The output *compliance* range for this current source (the range of output voltages where the current source behaves like a current source, that is, not an open or a resistor) is between VDD and $V_{DS,sat}$. The point where $V_O = V_{DS1} = V_{GS1}$ is where $I_O = I_{REF}$ (again this is important for matching two currents). Finally, note that I_{REF} and V_{GS1} are not dependent on V_O. ■

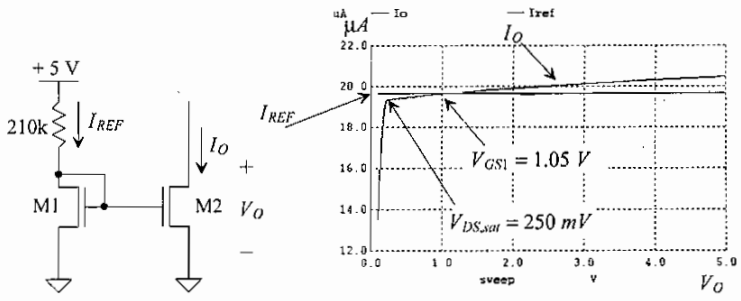


Figure 20.4 The operation of an NMOS current mirror.

20.1.2 Matching Currents in the Mirror

Many analog applications are susceptible to errors due to layout. In circuits in which devices need to be matched, layout becomes a critical factor. For example, in the basic current mirror shown in Fig. 20.2, first-order process errors can cause the output current, I_O, to be significantly different from the reference current. Process parameters such as gate-oxide thickness, lateral diffusion, oxide encroachment, and oxide charge density can drastically affect the performance of a device. Layout methods can be used to minimize the first-order effects of these parameter variations.

Threshold Voltage Mismatch

In a given current mirror application, the values for the threshold voltages are critical in determining the overall accuracy of the mirror. Again, examine the basic current mirror shown in Fig. 20.2. Since both devices have the same value for V_{GS} and assuming the sizes and transconductance parameters for both devices are equal, let's examine the effect of a mismatch in threshold voltages between the two devices. If it is assumed that the threshold mismatch is distributed across both devices such that

$$V_{THN1} = V_{THN} - \frac{\Delta V_{THN}}{2} \tag{20.5}$$

$$V_{THN2} = V_{THN} + \frac{\Delta V_{THN}}{2} \tag{20.6}$$

where V_{THN} is the average value of V_{THN1} and V_{THN2} and ΔV_{THN} is the mismatch, then

$$\frac{I_O}{I_{REF}} = \frac{\frac{KP_n}{2}\frac{W}{L}(V_{GS} - V_{THN} - \frac{\Delta V_{THN}}{2})^2}{\frac{KP_n}{2}\frac{W}{L}(V_{GS} - V_{THN} + \frac{\Delta V_{THN}}{2})^2} = \frac{\left[1 - \frac{\Delta V_{THN}}{2(V_{GS}-V_{THN})}\right]^2}{\left[1 + \frac{\Delta V_{THN}}{2(V_{GS}-V_{THN})}\right]^2} \tag{20.7}$$

If both expressions are squared and the higher order terms are ignored, then the first-order expression for the ratio of currents becomes

$$\frac{I_O}{I_{REF}} \approx 1 - \frac{2\Delta V_{THN}}{V_{GS} - V_{THN}} = 1 - \frac{2\Delta V_{THN}}{V_{DS,sat}} \tag{20.8}$$

Equation (20.8) is quite revealing because it shows that as V_{GS} decreases, the difference in the mirrored currents increases due to threshold voltage mismatch. This is particularly critical for devices that are separated by relatively long distances because the threshold voltage is susceptible to process gradients. *To attain high speed and to reduce the effects of threshold voltage mismatch, a large gate overdrive voltage should be used* (remembering for a long-channel process that $V_{ovn} = V_{DS,sat} = V_{GS} - V_{THN}$). Of course the drawback, for a current mirror, is a reduced range of compliance (the MOSFET enters the triode region earlier).

Transconductance Parameter Mismatch

The same analysis can be performed on the transconductance parameter, KP_n. If $KP_{n1} = KP_n - \Delta KP_n/2$ and $KP_{n2} = KP_n + \Delta KP_n/2$, where KP_n is the average of KP_{n1} and KP_{n2}, then assuming perfect matching on all other parameters, the difference in the currents becomes

$$\frac{I_O}{I_{REF}} = \frac{KP_n + 0.5\Delta KP_n}{KP_n - 0.5\Delta KP_n} \approx 1 + \frac{\Delta KP_n}{KP_n} \tag{20.9}$$

Since KP_n is a process parameter, we might think that as CMOS scales downwards (C'_{ox} goes up) the mismatch due to oxide variations and mobility differences might get better. However, there is less averaging of these variations with the smaller layout sizes. Practically, as we're about to see, differences in V_{DS} dominate the matching behavior.

Drain-to-Source Voltage and Lambda

One aspect of current mirror design that is *critical* for generating accurate currents is the drain-to-source voltage. As seen in Fig. 20.4, the only point where the currents of the devices are actually the same is when their V_{DS} values are equal. In Eq. (20.3) the ratio of the output current to the reference current is affected by both the matching in the drain-to-source voltages (V_O and V_{DS1}) and the device λs. If, for example in the short-channel process (see Table 9.2), V_{DS1} = .35 V, $V_{DS2} = V_O$ = .75 V, and $\lambda_1 = \lambda_2 = 0.6$ V^{-1}, then

$$\frac{I_O}{I_{REF}} = \frac{1 + \lambda_2 \cdot V_O}{1 + \lambda_1 \cdot V_{DS1}} = \frac{1 + 0.6 \cdot 0.75}{1 + 0.6 \cdot 0.35} = 1.20 \tag{20.10}$$

resulting in 20% error! It is extremely important, for good matching, that the V_{DS} values of the MOSFETs in the current mirror are equal.

Layout Techniques to Improve Matching

Most general analog applications require the length of the gate to be longer than minimum since the channel-length modulation, λ , has less effect on longer devices than on shorter ones (as discussed in Ch. 9). As a result, the minimum-sized devices found in digital circuits are not used as often in general analog design (however, see Ch. 26). However, the larger devices can result in larger parasitics if some layout issues are not considered. Figure 20.5a illustrates a basic MOSFET device with a large *W/L*. The implant resistance of the source and drain can be modeled as shown in Fig. 20.5b. The implant resistance can easily be reduced by simply adding as many contacts as possible along the width of both the source and the drain as seen in Fig. 20.5c. The increase in the number of contacts results in lower resistance, more current capability, and a more distributed current load throughout the device. However, as the device width increases, another technique is used which distributes the parasitics (both resistive and capacitive) into smaller contributions.

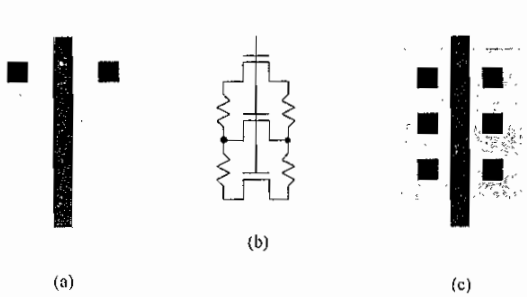


Figure 20.5 (a) Large device with a single contact and (b) its equivalent circuit. (c) Adding more contacts to reduce parasitic resistance.

Examine Fig. 20.6. In this figure, a single device with a large W/L is split into several parallel devices, each with a width one-fourth of the original W. One result of splitting the device into several parts is smaller overall parasitic capacitance associated with the reverse-biased implant substrate diode (the drain or source depletion capacitance to substrate). Since values of C_{db} and C_{sb} are proportional to W, the split device reduces these parasitics by a factor of $(n + 1)/2n$ where n is the number of parallel devices and is odd. If n is even, then the C_{sb} is reduced by one-half and C_{db} is reduced by $(n + 2)/2n$. As seen in Fig. 20.6b, putting the devices in parallel also reduces the parasitic resistance in series with both the source and the drain (it gets cut roughly in half too).

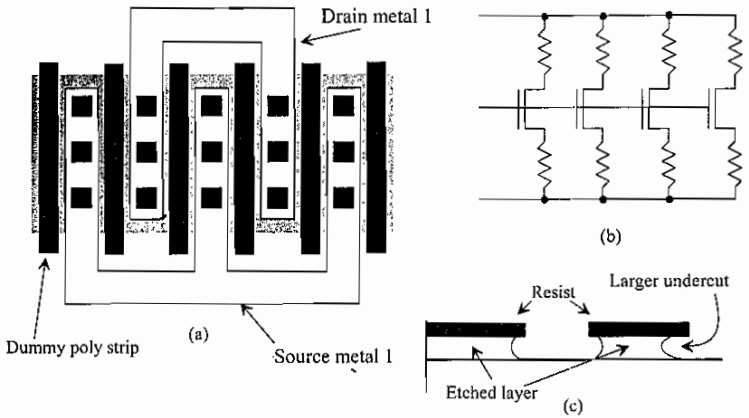


Figure 20.6 (a) A parallel device with dummy strips, (b) the equivalent circuit, and (c) undercutting.

Notice also, that Fig. 20.6a has dummy poly strips on both sides of the device. These strips are used to help minimize the effects of undercutting the poly on the outer edges after patterning, Fig. 20.6c. If the dummy strips had not been used, the poly would have been etched out more under the outermost gates, resulting in a mismatch between the four parallel devices.

When matching two devices, it is imperative that the two devices be as symmetrical as possible. Always orient the two devices in the same direction, unlike those illustrated in Fig. 20.7.

Splitting the devices into parallel devices and interdigitizing them can distribute process gradients across both devices and thus improve matching. An example of this is seen in Fig. 20.8. In (a) the current mirror seen in (b) is laid out. Each MOSFET in (b) is split up into four MOSFETs. If the W/L of each MOSFET in (b) is 80/2, then the size of each MOSFET (finger) in (a) is 20/2. Note the use of dummy poly strips on this layout. A good exercise at this point is to lay out the mirror of Fig. 20.8 in a common-centroid arrangement (see Ch. 5). Using a large layout area (long lengths and wide widths) and common-centroid layouts can result in significant improvements in matching.

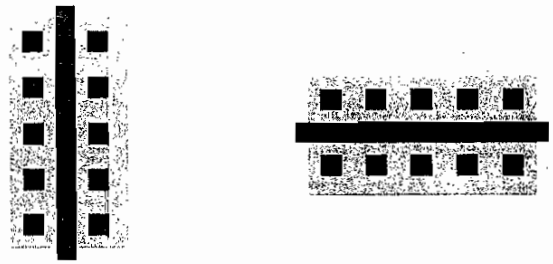


Figure 20.7 Devices with differing orientation (bad).

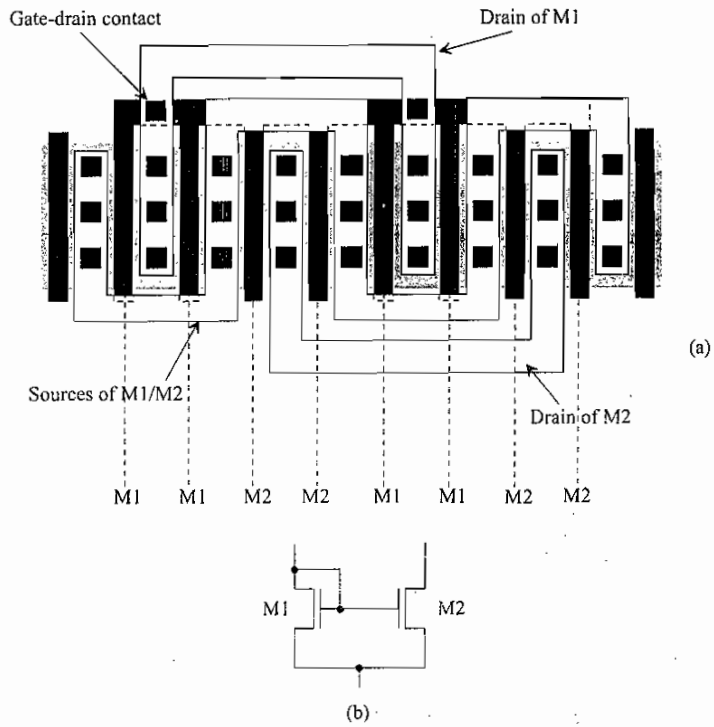


Figure 20.8 (a) Layout of a simple current mirror using interdigitation and (b) equivalent circuit.

Layout of the Mirror with Different Widths

When laying out the current mirror, the lateral diffusion, L_{diff}, under the gate oxide (Fig. 5.13) and the oxide encroachment, W_{enc}, can affect the actual length and width, respectively, of the MOSFET. If not careful with the layout, this will affect the mirror's ratio. Equation 20.3 can be rewritten, without including the differences due to drain-source voltage and lambda, as

$$\frac{I_O}{I_{REF}} = \frac{(W_{2drawn} - 2W_{enc}) \cdot (L_{1drawn} - 2L_{diff})}{(W_{1drawn} - 2W_{enc}) \cdot (L_{2drawn} - 2L_{diff})} \qquad (20.11)$$

If the requirement that $L_{1drawn} = L_{2drawn}$ is imposed, then the widths of the devices determine the relative currents in the mirror. Figure 20.9a shows a current mirror layout without width compensation. If W_{enc} is 0.2, then for this layout,

$$\frac{I_O}{I_{REF}} = \frac{40 - .2}{20 - .2} = 2.01 \text{ (a 1\% error due to poor layout)}$$

Figure 20.9b shows how to lay out a current mirror to avoid these problems. The layout of M2 is two MOSFETs in parallel. This can be specified in SPICE by adding M = X after the MOSFET statement in the netlist, where X is the number of MOSFETs,

```
M1    Vd1   Vd1   0    0    NMOS    L=2   W=20
M2    Vd2   Vd1   0    0    NMOS    L=2   W=20   M=2
```

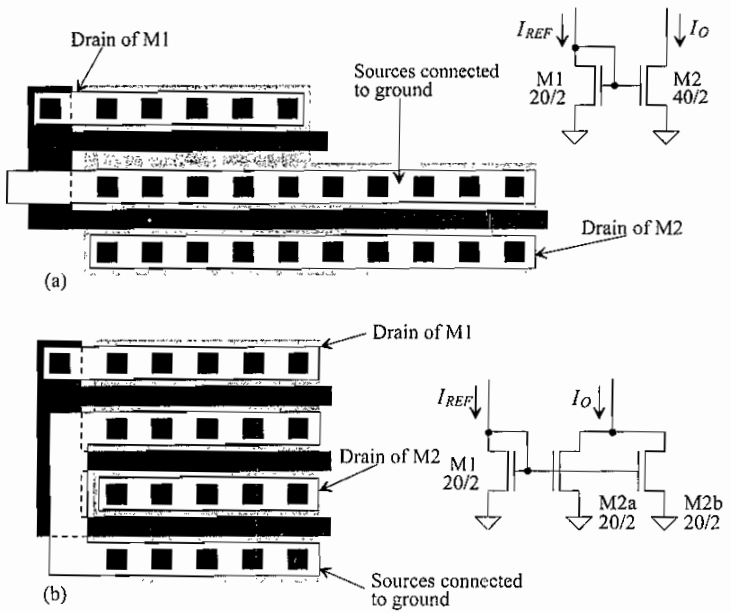


Figure 20.9 Layout of a current mirror (a) without width correction and (b) with width correction.

20.1.3 Biasing the Current Mirror

Using a resistor to set the bias current, as seen in Figs. 20.2–20.4, can result in currents that are too dependent on the power supply value and temperature. Consider the current mirror seen in Fig. 20.10a. In this design we've used the sizes and bias current given in Table 9.2 (the short-channel CMOS process) to select the resistor. In particular, $V_{SG} = 0.35\ V$ so the gate potential is 0.65 V. Figure 20.10b shows how the reference and output currents vary if VDD is swept from 900 mV to 1 V. The reference current is linearly dependent on VDD (as seen in Ex. 20.1). The output current is dependent on both the reference current and the V_{DS} (λ) of M2. In general, we want a reference current that isn't dependent on power supply or ground variations (noise). This is an important point. Consider the following example.

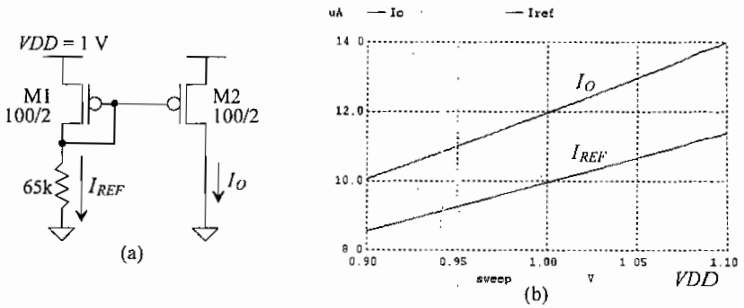


Figure 20.10 How reference and output current vary with VDD.

Example 20.2

Regenerate Fig. 20.10 if the 65k resistor is replaced with an ideal 10 μA current source, Fig. 20.11a. Explain the results.

The simulation results are seen in Fig. 20.11b. Note how the output current variation with VDD is much better than what is seen in Fig. 20.10. The reference current through M1 is, of course, a constant. The output current is larger than the

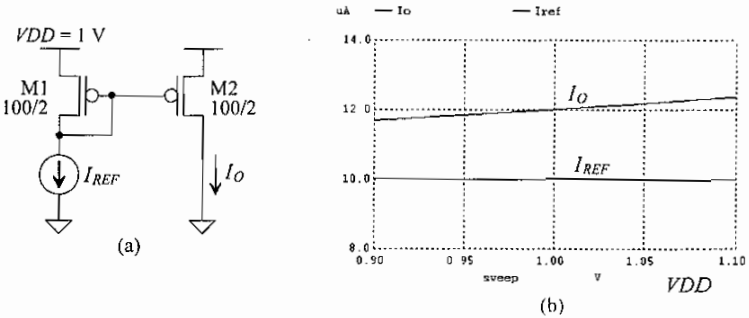


Figure 20.11 Showing that VDD variations don't affect the reference current.

reference current because $V_{SD1} < V_{SD2}$ (notice how often this point is coming up in the design of current mirrors). The reason I_O varies is due to the finite output resistance ($\lambda \neq 0$) of M2, Eq. (20.2).

The point of this example is that it is desirable to have a bias current that doesn't vary with changes in *VDD* or ground. ■

Using a MOSFET-Only Reference Circuit

Let's try replacing the resistor bias with a MOSFET, Fig. 20.12. If we assume long-channel behavior, then

$$VDD = V_{SG3} + V_{GS1} \tag{20.12}$$

or

$$VDD = \sqrt{\frac{2I_{REF}}{KP_p \frac{W_3}{L_3}}} + V_{THP} + \sqrt{\frac{2I_{REF}}{KP_n \frac{W_1}{L_1}}} + V_{THN} \tag{20.13}$$

If we are going to think of M3 as a resistor, as in Fig. 20.2, so that we mirror the current in M1, then we solve for the size of M3. Using the data from Table 9.1, we get

$$5 = \sqrt{\frac{2 \cdot 20}{40 \cdot \frac{W_3}{L_3}}} + 0.9 + \sqrt{\frac{2 \cdot 20}{120 \cdot \frac{10}{2}}} + 0.8 \rightarrow \frac{W_3}{L_3} = 0.11 \approx \frac{10}{90} \tag{20.14}$$

If we are going to think of M1 as a resistor, as in Fig. 20.10a, so that we mirror the current in M3, then we solve for the size of M1 as we did in Eq. (20.14) and get

$$\frac{W_1}{L_1} \approx \frac{10}{270} \tag{20.15}$$

Figure 20.13 shows the operation of these MOSFET bias circuits with *VDD* swept from 4.5 to 5.5 V. Note that, at 5 V, we should have a bias current of 20 μA. What we get is 14 μA. The reason that we have such a large difference is due to neglecting the output resistance of the MOSFETs (and the effects of mobility degradation). Note that the sensitivities, how the output current changes with *VDD*, are roughly 8 μA/V. (Let's compare this to hand-calculated values of sensitivity next.)

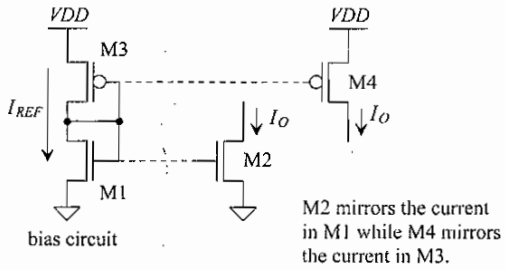


Figure 20.12 A MOSFET-only bias circuit.

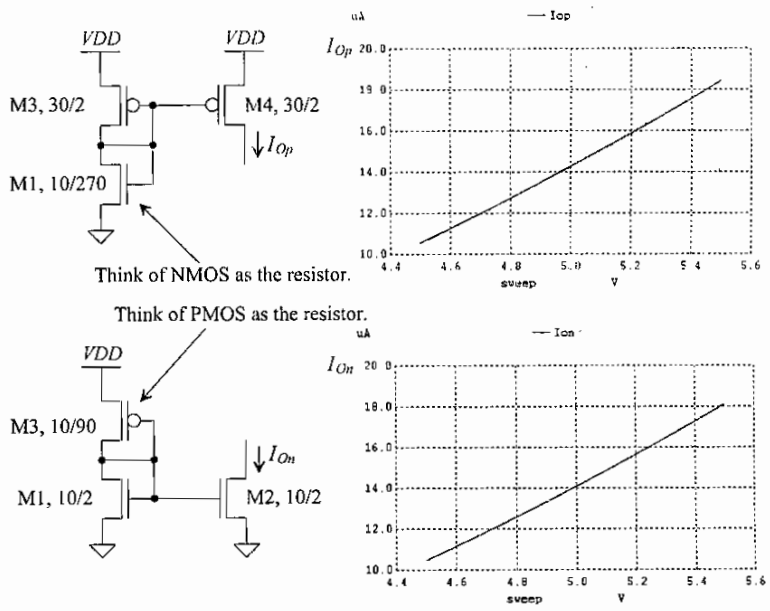


Figure 20.13 Behavior of MOSFET-only bias circuits with changes in VDD.

To determine I_{REF}'s sensitivity to *VDD*, let's take the derivative of the I_{REF} in Eq. (20.13) with respect to *VDD*. First let's write

$$(VDD - V_{THN} - V_{THP})^2 = I_{REF} \overbrace{\left[\sqrt{\frac{2L_3}{KP_p \cdot W_3}} + \sqrt{\frac{2L_1}{KP_n \cdot W_1}}\right]^2}^{K} \tag{20.16}$$

and next

$$\frac{\partial I_{REF}}{\partial VDD} = \frac{2 \cdot VDD}{K} - \frac{2 \cdot (V_{THN} + V_{THP})}{K} \tag{20.17}$$

Using the values in Table 9.1 and Eqs. (20.14) or (20.15), K is approximately 547×10^3 V^2/A so

$$\frac{\partial I_{REF}}{\partial VDD} = 12\ \mu A/V \tag{20.18}$$

For every millivolt change in *VDD* (around *VDD* = 5 V), we get 12 nA change in I_{REF}. As a comparison from Ex. 20.1, we can write (assuming the change in V_{GS} with *VDD* small)

$$\frac{\partial I_{REF}}{\partial VDD} \approx \frac{1}{R} = \frac{1}{200k} = 5\ \mu A/V \tag{20.19}$$

For every millivolt change in *VDD* we get 5 nA change in bias current.

Supply Independent Biasing

Instead of putting the resistor in the drain side of the current mirror, let's consider placing it in the source side, Fig. 20.14a. One of the problems with this approach is that if we try to mirror the current in M2 using M5 we won't know the value of the mirrored current. It's not obvious how V_{GS2} and V_{GS5} are related. In (b) we add a diode connected M1 so that its current can be mirrored (by M5). The next question is how do we force the same current through both M1 and M2? Figure 20.14c shows the addition of a PMOS current mirror to do this. From (c) we can write

$$V_{GS1} = V_{GS2} + I_{REF} \cdot R \tag{20.20}$$

which can only be valid if $V_{GS1} > V_{GS2}$. To ensure that this is the case, we use a larger value of β in M2, that is, we *multiply-up* M1's β in M2 so that less gate-source voltage is needed to conduct I_{REF}. Generally, this is done by simply using a larger width in M2. The resulting circuit is called a *Beta-multiplier* reference circuit. Knowing

$$V_{GS} = \sqrt{\frac{2I_D}{\beta}} + V_{THN} \tag{20.21}$$

$\left(\beta = KP_n \cdot \frac{W}{L}\right)$ and

$$\beta_2 = K \cdot \beta_1 \text{ (which is satisfied by } W_2 = K \cdot W_1) \tag{20.22}$$

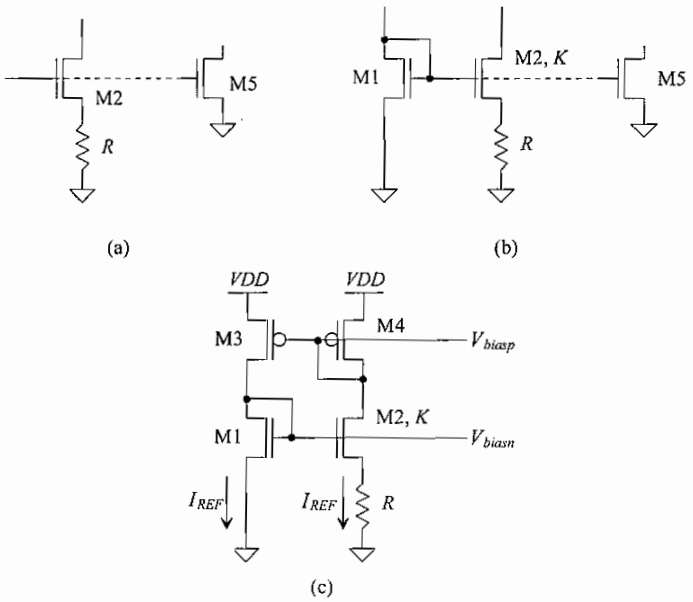


Figure 20.14 Developing the Beta-multiplier reference.

we can write

$$I_{REF} = \frac{2}{R^2 KP_n \cdot \frac{W_1}{L_1}}\left(1 - \frac{1}{\sqrt{K}}\right)^2 \tag{20.23}$$

It's important to note the reference current, neglecting the finite output resistance of the MOSFETs which wasn't included in the derivation of Eq. (20.23), is independent of *VDD*. Solving for R with the values given in Table 9.1 and a K of 4 gives $R = 6.5\ k\Omega$. Note that this circuit, when $K = 4$, is sometimes called a *constant-g_m* bias circuit because

$$g_m = \sqrt{2KP_n \frac{W}{L} \cdot I_{REF}} = \frac{1}{R} \tag{20.24}$$

a constant independent of MOSFET process shifts. Figure 20.15 shows the schematic of a bias circuit based on the sizes, bias currents, and $V_{DS,sat}$ given in Table 9.1.

Note that a "start-up circuit" was included in Fig. 20.15. In any self-biased circuit there are two possible operating points: the one we just described and the unwanted one where zero current flows in the circuit. This unwanted state occurs when the gates of M1/M2 are at ground while the gates of M3/M4 are at *VDD*. When in this state, the gate of MSU1 is at ground and so it is off. The gate of MSU2 is somewhere between *VDD* and $VDD - V_{THP}$. MSU3, which behaves like an NMOS switch, turns on and leaks current into the gates of M1/M2 from the gates of M3/M4. This causes the current to snap to the desired state and MSU3 to turn off. *Note that during normal operation the start-up circuit should not affect the Beta-multiplier's operation.* The current through MSU3 should be zero (or very small).

The Beta-multiplier is an example of a circuit that uses positive feedback. The addition of the resistor kills the closed loop gain (a positive feedback system can be stable if its closed loop gain is less than one). However, if we decrease the size of the resistor, we increase the gain of the loop and push the feedback system closer to instability. An example of when this could occur is if the parasitic capacitance on the source of M2 to ground is large (effectively shorting M2's source to ground). If the resistor, for example, is bonded out off-chip to set the current, it is likely that this bias circuit will oscillate.

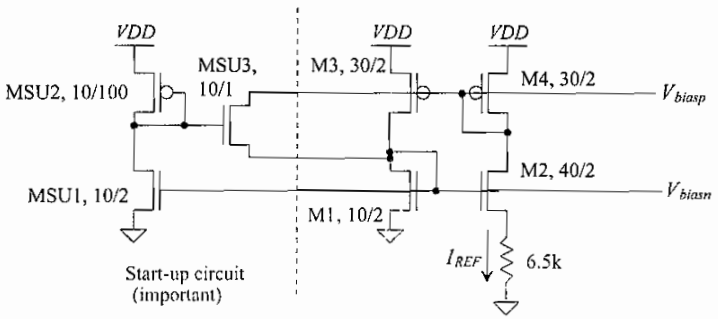


Figure 20.15 Beta-multiplier reference for biasing in the long-channel process described in Table 9.1.

Figure 20.16 shows how the reference currents through M1 and M2 vary with *VDD*. The minimum value of *VDD* can be estimated by looking at the minimum value of voltage across the drain-source of M3 and M1. For M3 this is $V_{SD3,sat}$ or 250 mV (because we are using the parameters from Table 9.1 in the design of this current reference). For M1 this is $V_{DS1} = V_{GS1} = 1.05$ V. We can then write

$$VDD_{min} = V_{SD3,sat} + V_{GS1} = 1.3\ V \qquad (20.25)$$

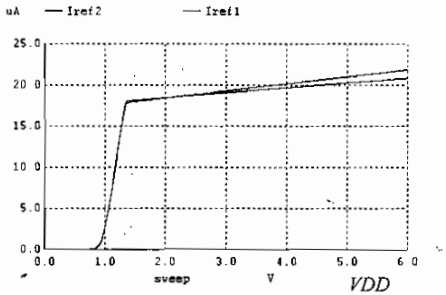


Figure 20.16 The reference currents through M1 and M2 in the Beta-multiplier.

Finally, the sensitivity of I_{REF} is directly dependent on the output resistance of the MOSFETs. From the simulations

$$\frac{\partial I_{REF}}{\partial VDD} \approx 800\ nA/V$$

or almost an order of magnitude better than the previous reference circuits

Example 20.3

Estimate the voltage on the gate of M6 and its drain current in Fig. 20.17.

M3 and M4 are biased to source 20 μA of current. Since $V_{GS1} = V_{GS2} = V_{DS1}$ (and M1/M2 have the same drain current, see Fig. 20.1 and the associated discussion), it follows that $V_{GS6} = V_{DS2} = V_{GS1} = 1.05$ V. *We treat M6, for biasing purposes, as if its gate were tied to the gates of M1 or M2.* It follows then that $I_{D6} = 20$ μA. ■

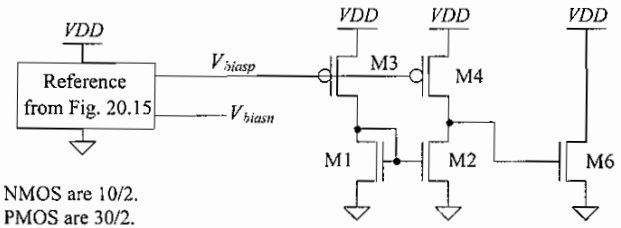
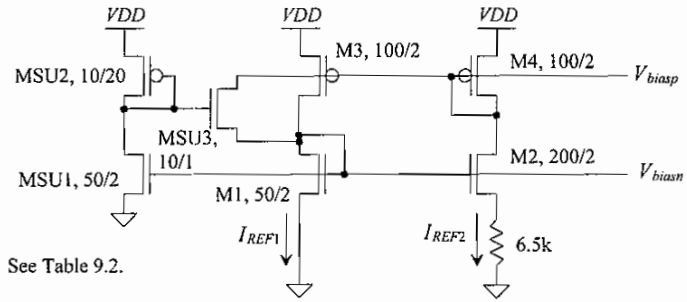


Figure 20.17 Circuit used in Ex. 20.3

20.1.4 Short-Channel Design

Figure 20.18 shows a Beta-multiplier biasing circuit based on the values given in Table 9.2 as well as simulation results showing how the reference currents vary with *VDD*. From Eq. (20.25) and Table 9.2 we would expect VDD_{min} to be 400 mV. At *VDD* = 1 V the reference currents are indeed 10 μA. However, what we see is a horrible sensitivity to changes in *VDD*. Reviewing Figs. 9.31 and 20.10 we see the low output resistance of the short-channel devices causes the drain current to change significantly with changes in drain-to-source voltage. In some analog applications this variation isn't that harmful. However, if the Beta-multiplier circuit is to behave like a true current reference, the reference currents shouldn't vary with changes in *VDD*.



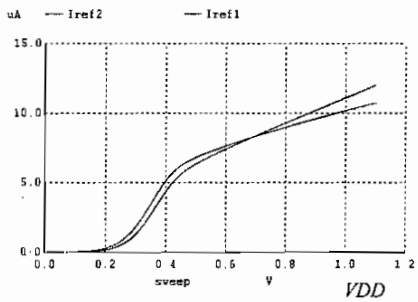


Figure 20.18 Beta-multiplier reference for short-channel design (see Table 9.2).

To reduce the sensitivity, we need to reduce the variations in the drain-to-source voltages of the NMOS devices with changes in *VDD*. Consider adding a differential amplifier (diff-amp) to the basic Beta-multiplier seen in Fig. 20.19. Note that M4 is no longer diode-connected so its drain can move to the same potential as M2's drain, that is, V_{biasn}. The start-up circuit (required) is not shown. The idea is to use the amplifier to compare the drain voltage of M1 (V_{biasn}) with the drain voltage of M2 (V_{reg}) and regulate them to be equal. The result is an effective increase in M2's output resistance. For example, if V_{reg} is above V_{biasn}, the amplifier's output voltage increases. This drives the gate of M4 upwards, lowering the current it supplies and causing V_{reg} to drop back down.

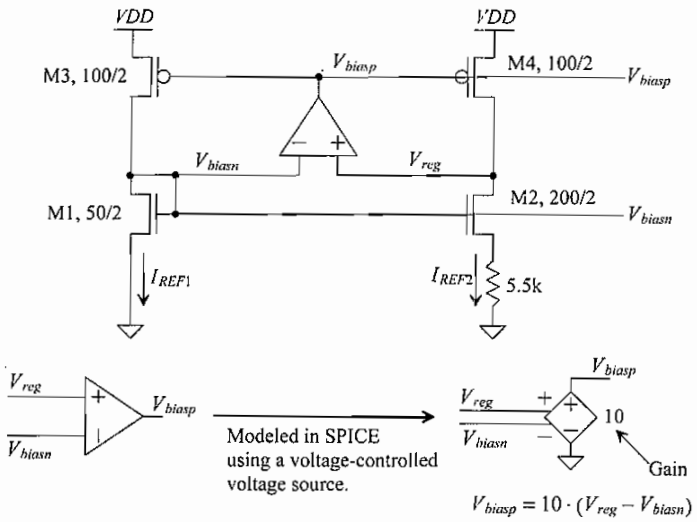


Figure 20.19 Increasing the output resistance of short-channel MOSFETs using feedback. The result, for the Beta-multiplier circuit, is better power supply sensitivity.

At the same time the gate of M3 is also increased, causing it to source less current. This causes a drop in V_{biasn} (the same as V_{reg} because of the symmetry as discussed in Fig. 20.1 or Ex. 20.3). Figure 20.20 shows how the reference current, in Fig. 20.19, changes with *VDD* when the added amplifier has a gain of 10. We've lowered R to 5.5k to more accurately set the current and used, in the simulation, a voltage-controlled voltage source for the amplifier.

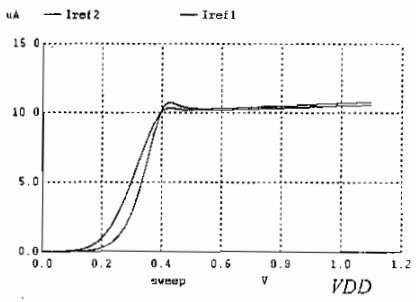


Figure 20.20 Improvement with the added amplifier.

Figure 20.21 shows a possible differential amplifier configuration. The PMOS devices form a current mirror while the NMOS devices will, when $V_{biasn} = V_{reg}$, simply mirror the current in M1. If $V_{biasn} \neq V_{reg}$ then the imbalance causes the amplifier output to swing up or down providing the desired action.

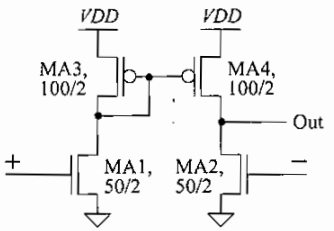
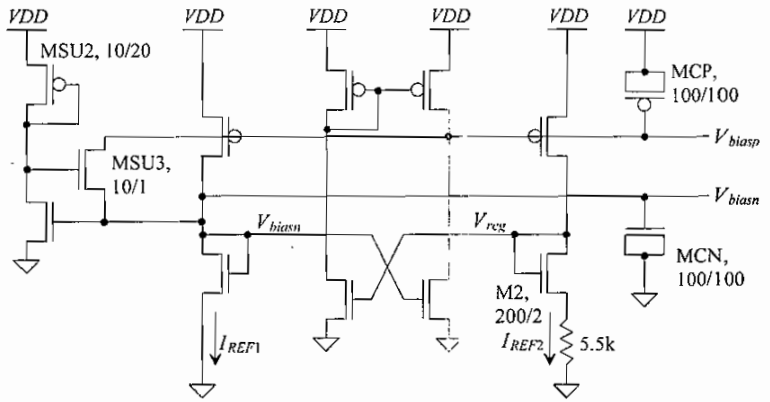


Figure 20.21 A possible implementation of the diff-amp in Fig. 20.19.

Figure 20.22 shows the improved current reference for use in a short-channel CMOS process (where devices have very low output resistances). The amplifier in Fig. 20.21 is placed in the reference, as indicated in Fig. 20.19. As with any feedback scheme, stability is of critical importance. As we'll see in the coming chapters, a feedback amplifier with only a single high-impedance node (meaning no diode-connected MOSFETs or MOSFET-source terminals are connected to the node) is straightforward to compensate (make stable). So that the reference has a single high-impedance node, that is V_{biasp}, we connected M2's gate to its drain (instead of to V_{biasn}). We get the same effect, as



All unlabeled NMOS are 50/2.
All unlabeled PMOS are 100/2.

Figure 20.22 Improved current reference for short-channel devices. See also Sec. 23.1.3 for more information.

seen in Fig. 20.19 but the gain around the loop is reduced. To make the reference stable, we add capacitors (MCP and MCN) to the circuit. The high-impedance node is the critical point for adding the compensation capacitor (that is MCP is the critical capacitance). If the reference drives a significant number of MOSFETs (so these driven MOSFETs provide the load capacitance), then MCP and MCN may be excluded from the design. Figure 20.23 shows how the reference currents change with *VDD*.

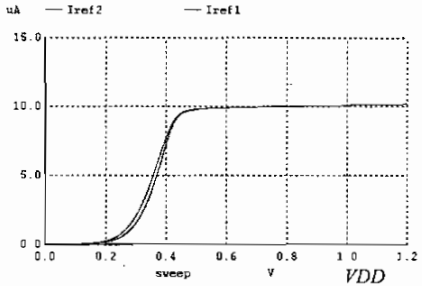
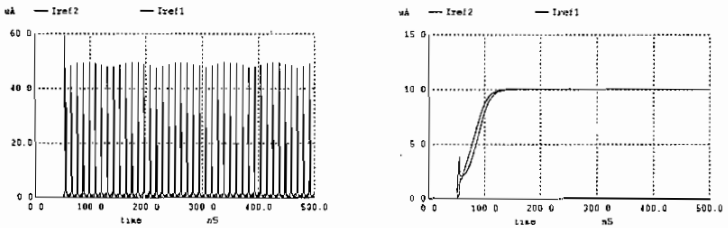


Figure 20.23 Variation of reference currents with VDD for the circuit in Fig. 20.22.

An Important Note

It's extremely important to understand that together with the benefits of better power supply sensitivity are the undesired traits of feedback, that is, the potential for an unstable circuit. For example, Fig. 20.24a shows what happens to the reference currents if we remove MCP and MCN in Fig. 20.22 and then apply a step voltage to *VDD* (that is, *VDD* steps from 0 to 1 V at 50 ns in the simulation). Clearly, the reference is not stable and the currents oscillate. In (b) we do the same thing but with MCP and MCN present. The response shows first-order behavior, and oscillations are not present. We might think we are done with the reference and move on to the other analog circuit designs. However, further characterization of the current reference using simulations is warranted. For example, what happens if *VDD* has a 50 mV squarewave signal coupled to it at 100 MHz? How does the reference behave with this signal (a squarewave *VDD* that oscillates between 1 and 1.05 V at 100 MHz)? In general, no analog circuit has good power supply noise rejection at high frequencies. To remedy this, the power supply is decoupled (a large capacitor is placed from *VDD* to ground to remove high-frequency noise).



(a) MCP and MCN not present (b) MCP and MCN present

Figure 20.24 What happens when VDD is pulsed from 0 to 1 at 50 ns.

20.1.5 Temperature Behavior

While we've been focusing on how the reference current changes with *VDD*, it's also important to know how the current changes with temperature. Looking at the basic current mirror in Fig. 20.11a, note that if I_{REF} is a constant, independent of temperature, then the output current will also be independent of temperature. That's not to say that the MOSFET characteristics aren't changing because they are; they change at the same rate. Because they change together, the current mirror relationship is still valid and M2 mirrors the current in M1. Of course, if M1 is located at a different physical location than M2 and each is heated differently, a mismatch in the currents will result. *The point is that the temperature behavior of the reference current determines the temperature behavior of all mirrored currents.*

Using the short-channel CMOS parameters we know, from Ch. 9 (Fig. 9.30b), that the change in V_{THN} with temperature is −0.6 mV/C° (for an NMOS device). For the PMOS device we can vary temperature and look at the change in V_{THP}. Using the mirror in Fig. 20.11 and looking at the V_{SG} at different temperatures results in the plots seen in Fig. 20.25. We see that it also varies at a rate of −0.6 mV/C°.

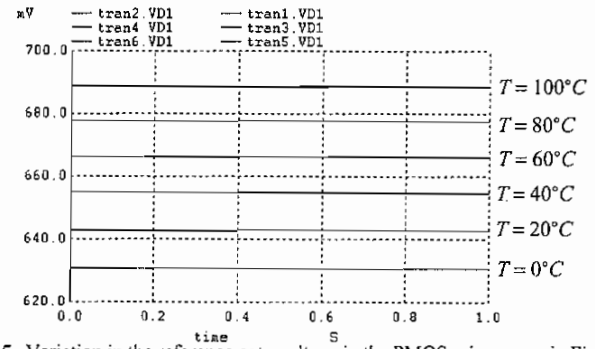


Figure 20.25 Variation in the reference gate voltage in the PMOS mirror seen in Fig. 20.11.

Resistor-MOSFET Reference Circuit

For a resistor-MOSFET reference circuit (Fig. 20.2), we can write (neglecting the MOSFET's finite output resistance)

$$I_{REF} = \frac{VDD - V_{GS}}{R} \tag{20.26}$$

The change in I_{REF} with temperature is

$$\frac{\partial I_{REF}}{\partial T} = -\frac{VDD - V_{GS}}{R^2} \cdot \frac{\partial R}{\partial T} - \frac{1}{R} \cdot \frac{\partial V_{GS}}{\partial T} \tag{20.27}$$

or

$$\frac{\partial I_{REF}}{\partial T} = -I_{REF} \cdot \frac{1}{R}\frac{\partial R}{\partial T} - \frac{1}{R} \cdot \frac{\partial V_{GS}}{\partial T} \tag{20.28}$$

We know that to write the reference current as a function of temperature we use

$$I_{REF}(T) = I_{REF}(T_0) \cdot (1 + TCI_{REF} \cdot (T - T_0)) \tag{20.29}$$

where the temperature coefficient of the reference current is

$$TCI_{REF} = \frac{1}{I_{REF}} \cdot \frac{\partial I_{REF}}{\partial T} \tag{20.30}$$

Rewriting Eq. (20.28) using Eq. (20.26) yields

$$TCI_{REF} = \frac{1}{I_{REF}} \cdot \frac{\partial I_{REF}}{\partial T} = -\frac{1}{R}\frac{\partial R}{\partial T} - \frac{1}{VDD - V_{GS}}\frac{\partial V_{GS}}{\partial T} \tag{20.31}$$

Noting, again, that the temperature coefficient is not a constant but rather changes with temperature (even though this isn't indicated in Eq. (20.29)) just like a small-signal parameter, e.g., g_m, changes with the DC operating point.

Example 20.4

Determine the temperature behavior of the reference current in the mirror seen in Fig. 20.10. Verify the answer with SPICE.

If the temperature coefficient of the resistor, $TCR = \frac{1}{R}\frac{\partial R}{\partial T} = 2000\ ppm/C°$ (= 0.002) and we assume after looking at Fig. 9.30 that $\frac{\partial V_{GS}}{\partial T} \approx \frac{\partial V_{THN}}{\partial T} \approx -0.6\ mV/C°$ then, knowing the parameters used in Table 9.2 were the basis for the design of the current mirror in Fig. 20.10,

$$TCI_{REF} = -0.002/C° - \frac{1}{1 - 0.35} \cdot (-0.6\ mV/C°) \approx -1,000\ ppm/C°$$

and so

$$I_{REF}(T) = 10 \cdot (1 - 0.001 \cdot (T - 27))\ \mu A$$

At 100 C° the reference current is 9.27 μA and at 0 C° it's 10.27 μA (of course, the reference current is approximately 10 μA at 27 C°). Figure 20.26 shows the SPICE simulation results. ■

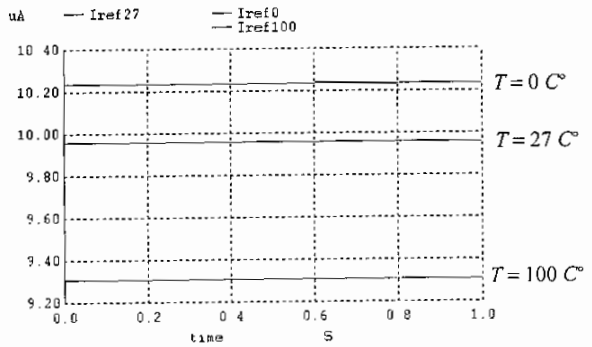


Figure 20.26 Example 20.4 showing the temperature behavior of the reference current in Fig. 20.10.

MOSFET-Only Reference Circuit

For the MOSFET-only bias circuit, Fig. 20.12, we can write the reference current as (see Eq. [20.16])

$$I_{REF} = \frac{VDD^2 - 2VDD \cdot (V_{THN} + V_{THP}) + (V_{THN} + V_{THP})^2}{\left(\sqrt{\frac{2L_3}{KP_p \cdot W_3}} + \sqrt{\frac{2L_1}{KP_n \cdot W_1}}\right)^2} \tag{20.32}$$

Knowing $KP_n = M \cdot KP_p$, where M is 3 for our long-channel process and 2 for our short-channel process (see Tables 9.1 and 9.2), we can write

$$I_{REF} = KP_n \cdot \frac{VDD^2 - 2VDD \cdot (V_{THN} + V_{THP}) + (V_{THN} + V_{THP})^2}{\left(\sqrt{\frac{M \cdot 2L_3}{W_3}} + \sqrt{\frac{2L_1}{W_1}}\right)^2} \tag{20.33}$$

or

$$\frac{\partial I_{REF}}{\partial T} = \frac{\partial KP_n}{\partial T} \cdot \frac{VDD^2 - 2VDD \cdot (V_{THN} + V_{THP}) + (V_{THN} + V_{THP})^2}{\left(\sqrt{\frac{M \cdot 2L_3}{W_3}} + \sqrt{\frac{2L_1}{W_1}}\right)^2} +$$

$$\frac{KP_n}{\left(\sqrt{\frac{M \cdot 2L_3}{W_3}} + \sqrt{\frac{2L_1}{W_1}}\right)^2} \cdot \left(-2VDD \cdot \left(\frac{\partial V_{THN}}{\partial T} + \frac{\partial V_{THP}}{\partial T}\right) + 2(V_{THN} + V_{THP})\left(\frac{\partial V_{THN}}{\partial T} + \frac{\partial V_{THP}}{\partial T}\right)\right) \tag{20.34}$$

Dividing both sides by Eq. (20.33) gives

$$\frac{1}{I_{REF}}\frac{\partial I_{REF}}{\partial T} = \frac{1}{KP_n}\frac{\partial KP_n}{\partial T} + \frac{2\left(\frac{\partial V_{THN}}{\partial T} + \frac{\partial V_{THP}}{\partial T}\right)(V_{THN} + V_{THP} - VDD)}{VDD^2 - 2VDD \cdot (V_{THN} + V_{THP}) + (V_{THN} + V_{THP})^2} \tag{20.35}$$

noting, from Eq. (9.52), that the first term is simply −1.5/T.

Example 20.5

Determine the temperature behavior of the reference current in the MOSFET-only bias circuit seen in Fig. 20.13.

We note that both references in Fig. 20.13 have the same temperature behavior since Eq. (20.35) doesn't show a width or length dependence (there is a dependence in the reference current though). Figure 20.13 used the long-channel MOSFET paramcters from Table 9.1 where

$$V_{THN} = 0.8,\ V_{THP} = 0.9,\ \frac{\partial V_{THN}}{\partial T} = -1\ mV/C°,\ \frac{\partial V_{THP}}{\partial T} = -1.4\ mV/C°$$

and VDD = 5 V with I_{REF} = 13.5 μA (measured in the simulation that generated Fig. 20.13) at T = 300 K°. Plugging the numbers into Eq. (20.35) gives

$$TCI_{REF} = \frac{1}{I_{REF}}\frac{\partial I_{REF}}{\partial T} = -\frac{1.5}{300} + \frac{2(-0.0024)(-3.3)}{25 - 17 + 2.89} = -0.005 + 0.00146 \approx -3,500\ ppm/C°$$

and so $I_{REF}(T) = 13.5(1 - 0.0035(T - T_0))\ \mu A$. ■

Temperature Behavior of the Beta-Multiplier

To determine the temperature behavior of the Beta-multiplier, we take the derivative of Eq. (20.23) with respect to temperature

$$\frac{\partial I_{REF}}{\partial T} = \frac{-4}{R^3 KP_n \cdot \frac{W_1}{L_1}}\left(1-\frac{1}{\sqrt{K}}\right)^2 \cdot \frac{\partial R}{\partial T} - \frac{2}{R^2 KP_n^2 \cdot \frac{W_1}{L_1}}\left(1-\frac{1}{\sqrt{K}}\right)^2 \frac{\partial KP_n}{\partial T} \quad (20.36)$$

Dividing both sides by I_{REF} (Eq. [20.23]) gives

$$TCI_{REF} = \frac{1}{I_{REF}} \cdot \frac{\partial I_{REF}}{\partial T} = -2\left(\frac{1}{R}\frac{\partial R}{\partial T}\right) - \frac{1}{KP_n}\frac{\partial KP_n}{\partial T} \quad (20.37)$$

Example 20.6

Determine the temperature behavior of the Beta-multiplier seen in Fig. 20.22.

If the temperature coefficient of the resistor, $TCR = \frac{1}{R}\frac{\partial R}{\partial T} = 2000\ ppm/C°$ then

$$TCI_{REF} = -0.004 + \frac{1.5}{300} = 1000\ ppm/C°$$

and so

$$I_{REF}(T) = 10 \cdot (1 + 0.001 \cdot (T-27))\ \mu A$$

The positive TC of the resistor subtracts from the negative TC of the mobility to stabilize the current reference. If the mobility change with temperature is complicated by velocity saturation effects, then simulations become invaluable to determine the actual temperature performance of the circuit. ■

Voltage Reference Using the Beta-Multiplier

Let's try designing the Beta-multiplier where the gate voltage of M1 (see Figs. 20.15 or 20.22) is a constant independent of temperature. In this type of design we aren't using this voltage directly to bias anything so we'll change the label from $V_{bias n}$ to V_{REF}. Using Eqs. (20.22) and (20.23) gives

$$V_{REF} = V_{GS1} = \frac{2}{R \cdot KP_n \cdot \frac{W}{L}}\left(1-\frac{1}{\sqrt{K}}\right) + V_{THN} \quad (20.38)$$

Taking the derivative with respect to temperature gives

$$\frac{\partial V_{REF}}{\partial T} = \frac{\partial V_{THN}}{\partial T} - \frac{2}{R \cdot KP_n \cdot \frac{W}{L}}\left(1-\frac{1}{\sqrt{K}}\right) \cdot \left(\frac{1}{R}\frac{\partial R}{\partial T} + \frac{1}{KP_n} \cdot \frac{\partial KP_n}{\partial T}\right) \quad (20.39)$$

If we want the change with temperature to be zero, $\partial V_{REF}/\partial T = 0$, then we can select the resistor based on

$$R = \frac{2}{\frac{\partial V_{THN}}{\partial T} \cdot KP_n \cdot \frac{W}{L}}\left(1-\frac{1}{\sqrt{K}}\right) \cdot \left(\frac{1}{R}\frac{\partial R}{\partial T} + \frac{1}{KP_n} \cdot \frac{\partial KP_n}{\partial T}\right) \quad (20.40)$$

Using the parameters from Table 9.1 with $K = 4$ and a resistor temperature coefficient of 0.002 gives a resistor value of 5 kΩ. Figure 20.27 shows the SPICE simulation results. The variation is approximately 60 mV/100 C° or 600 μV/C°.

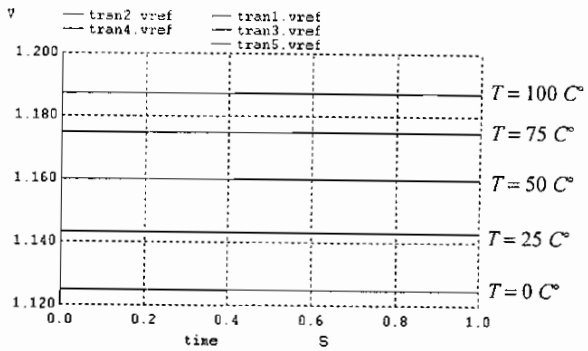


Figure 20.27 Temperature performance of a voltage reference using the Beta-multiplier.

20.1.6 Biasing in the Subthreshold Region

In some CMOS designs (for example, low-power designs) the MOSFETs may be operated in the weak or subthreshold regions. If a resistor is used to bias a current mirror, as seen in Fig. 20.10, its value can be very large. For example, if we use the short-channel process with $VDD = 1$ V and a V_{SG} of approximately V_{THP} (= 280 mV) with an I_D of 10 nA, we need a resistor with a value of

$$I_{REF} = \frac{VDD - V_{SG}}{R} \rightarrow R = 72\ M\Omega\ ! \quad (20.41)$$

Clearly, this isn't practical. To bias circuits for subthreshold operation, let's use the Beta-multiplier. From Eq. (9.17) we can write

$$V_{GS1} = nV_T \cdot \ln\left(\frac{I_{REF} \cdot L_1}{I_{D0} \cdot W_1}\right) + V_{THN} \quad (20.42)$$

and

$$V_{GS2} = nV_T \cdot \ln\left(\frac{I_{REF} \cdot L_1}{I_{D0} \cdot K \cdot W_1}\right) + V_{THN} \quad (20.43)$$

Knowing

$$I_{REF} = \frac{V_{GS1} - V_{GS2}}{R} \quad (20.44)$$

we get

$$I_{REF} = \frac{n \cdot V_T}{R} \cdot \ln K \quad (20.45)$$

or

$$R = \frac{n \cdot V_T}{I_{REF}} \cdot \ln K \quad (20.46)$$

Using the values above, that is, $K = 4$ and an I_{REF} of 10 nA, gives an R (assuming $n = 1$) of approximately 3.5 MΩ. Still large but much better than 72 MΩ.

20.2 Cascoding the Current Mirror

In this section we discuss increasing the output resistance of a current mirror so that it behaves more ideally. Figure 20.28 shows the problem with using a single MOSFET for a current source. Instead of being a constant, the output current, I_O , increases as the voltage across the current source, V_O , increases. We model this increase using the MOSFET's output resistance, r_o (see Table 9.2). If we can hold the drain-source voltage of the MOSFET constant, then the current doesn't vary. However, this requires a fixed V_O. To avoid this, we'll add circuitry in between the current-mirrored MOSFET and V_O.

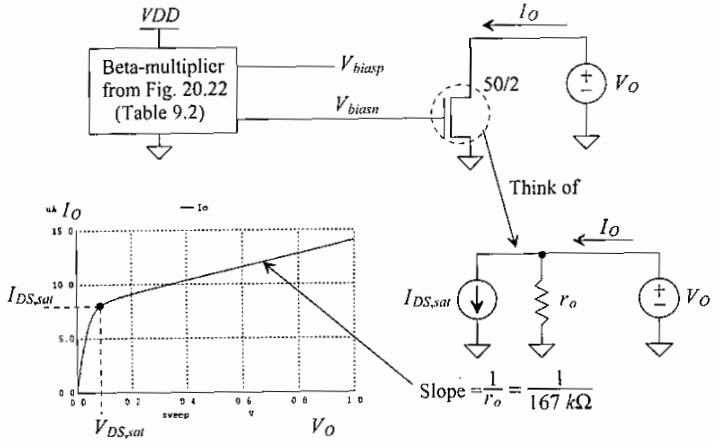


Figure 20.28 How the finite output resistance of the MOSFET affects the output current.

20.2.1 The Simple Cascode

The simple cascode current mirror made with M1–M4 is seen in Fig. 20.29. The name "cascode" is a vestige from the days of vacuum tubes when a common-cathode amplifier was cascaded (in series with) a common-grid amplifier. Drawing an analogy with MOSFETs, the cathode of a tube is equivalent to the source of a MOSFET. The grid of a tube corresponds to the gate of a MOSFET and the anode to the drain of the MOSFET.

It's important to realize that I_O is still determined by the gate-source voltages of M1 and M2. Changing the sizes of M3 and M4 simply changes the drain-source voltages of M1/M2, affecting the matching of their drain currents, as indicated by Eq. (20.10). Again, this (the gate-source voltages of M1/M2 determine the currents) is important to understand since, in the next several pages, we present different modifications to this basic cascode circuit. These modifications attempt to hold the drain source voltages of M1/M2 more constant to increase the current mirror's output resistance (make I_O change less with changes in V_O).

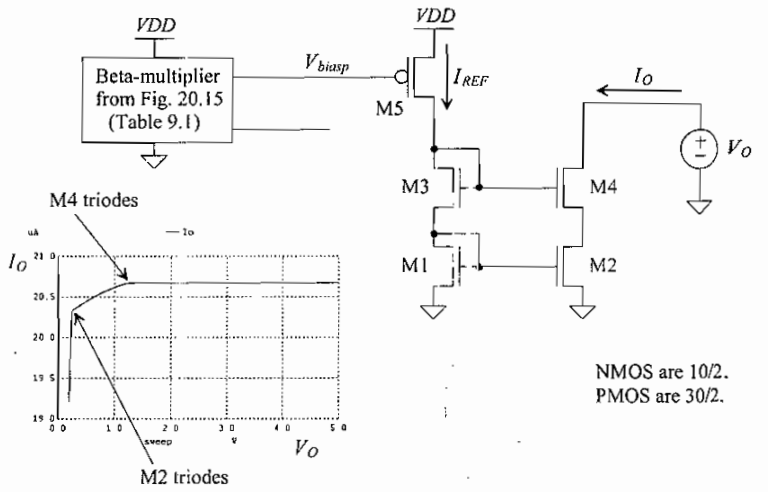


Figure 20.29 Biasing of the cascode current source and its operation.

DC Operation

The voltage on the gate of M4 in Fig. 20.29 is (remembering the actual values for these specific biasing conditions is given in Table 9.1)

$$2V_{GS} = 2(V_{DS,sat} + V_{THN}) \tag{20.47}$$

The voltage on the drain of M2, assuming that it is operating in the saturation region, is V_{GS}. The minimum voltage across the current source is then

$$V_{O,\min} = V_{DS,sat4} + V_{GS} = 2V_{DS,sat} + V_{THN} \tag{20.48}$$

Plugging in the numbers from Table 9.1, we get $V_{O,\min} = 1.3\ V$. Reviewing Fig. 20.29, we see that M4 starts to move into the triode region at this voltage.

To keep both M2 and M4 operating in the saturation region, we need only a $V_{DS,sat}$ across each one. In the next section we'll use this fact to design a *low-voltage cascode* current mirror.

Cascode Output Resistance

To estimate the output resistance of the cascode current mirror, let's apply a test voltage to the simplified cascode schematic seen in Fig. 20.30. We treat M3 and M4 as DC bias sources (AC grounds) and assume that the DC voltage on the drain of M4 is large enough to ensure that M2 and M3 are operating in the saturation. Further we draw the output resistances external to the devices. Note that M2's gate source voltage is zero and its drain is $-v_{gs4}$. The resistance seen looking into the drain of M4 is

$$R_o = \frac{v_T}{i_T} \tag{20.49}$$

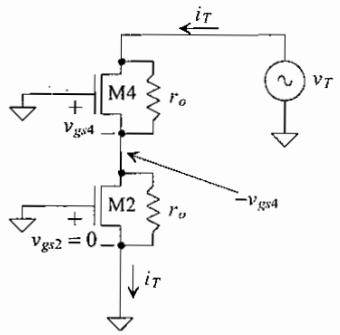


Figure 20.30 AC circuit used to determine the output resistance of a cascode current source.

Looking at Fig. 20.30, we can write

$$-v_{gs4} = i_T \cdot r_o \tag{20.50}$$

The current through M4 is then

$$i_T = g_m v_{gs4} + \frac{v_T - (-v_{gs4})}{r_o} \tag{20.51}$$

Substituting (20.50) into (20.51) gives

$$i_T = g_m(-i_T \cdot r_o) + \frac{v_T}{r_o} - i_T \tag{20.52}$$

Solving for the output resistance

$$R_o = (2 + g_m r_o) r_o \approx g_m r_o^2 \tag{20.53}$$

This result should be remembered because it will be referred to often. Using the numbers from Tables 9.1 and 9.2, we can tabulate the output resistance of cascode structures as seen in Table 20.1.

Table 20.1 Output resistances for cascode structures.

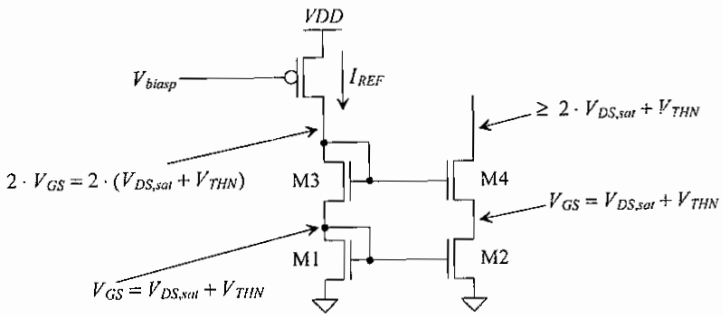
Cascode type	Long-channel process, Table 9.1, R_o	Short-channel process, Table 9.2, R_o
PMOS	2.4 GΩ	16.6 MΩ
NMOS	3.75 GΩ	4.2 MΩ

Notice the large difference (almost three orders of magnitude) between the output resistance of the short- and the long-channel devices. Again, this is the reason we need to use special circuit techniques when designing in short-channel processes.

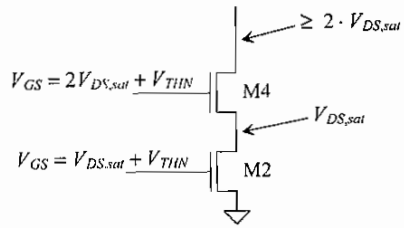
Finally, note that the output resistance can be determined from a plot like the one seen in Fig. 20.29 using, in WinSPICE, plot 1/deriv(lo). As seen in Fig. 20.28, the output resistance is the reciprocal of the derivative of the output current.

20.2.2 Low-Voltage (Wide-Swing) Cascode

Figure 20.31a shows the regularly biased cascode structure. The key point of this figure is that the drain of M2 is not at the minimum voltage required to keep it in saturation, $V_{DS,sat}$, but rather it is held at a threshold voltage above this minimum. In (b) the cascode is biased for lowest voltage operation. What this means is that the drain of M4, V_o, can go to the minimum possible voltage that keeps M2/M4 in saturation, that is, $2V_{DS,sat}$. Again, it's important to notice that the gate-source voltages of M1/M2 are what set the currents in the mirror.



(a) Regular cascode structure



(b) Low-voltage (aka wide-swing) structure

Figure 20.31 DC voltages in (a) a cascode current mirror and in (b) a low-voltage cascode.

To generate the gate voltage of M2, we simply use a diode-connected MOSFET as seen in almost every schematic in this chapter. From Eq. (20.1), for example, we can write

$$I_{REF} = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{GS} - V_{THN})^2 \tag{20.54}$$

neglecting channel-length modulation. To generate the bias voltage for M4, we can adjust the width and length of a MOSFET, MWS, as seen in Fig. 20.32. We can write, knowing $V_{DS,sat} = V_{GS} - V_{THN}$

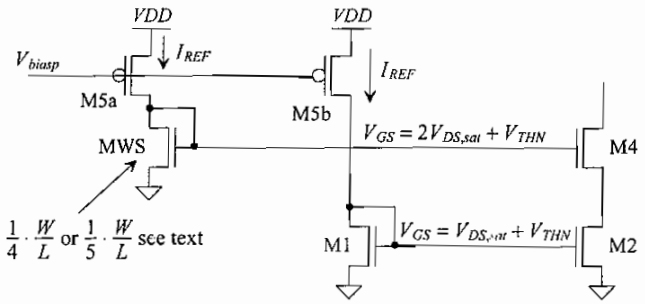


Figure 20.32 Generating a bias voltage for M4.

$$I_{REF} = \frac{KP_n}{2} \cdot \frac{W_{MWS}}{L_{MWS}} (2(V_{GS} - V_{THN}) + V_{THN} - V_{THN})^2 \qquad (20.55)$$

or

$$I_{REF} = \frac{KP_n}{2} \cdot \frac{W_{MWS}}{L_{MWS}} \cdot 4(V_{GS} - V_{THN})^2 \qquad (20.56)$$

Using our device sizes from either Tables 9.1 or 9.2, we can make the length of MWS four times the length of the other MOSFETs or

$$\frac{W}{L} = \frac{W_{MWS}}{L_{MWS}} \cdot 4 \qquad (20.57)$$

If we use the same widths, then

$$L_{MWS} = 4 \cdot L \qquad (20.58)$$

If Eq. (20.58) is valid, then Eq. (20.56) is equal to Eq. (20.54).

Note that, as seen in Fig. 20.31b, we are biasing M2 right on the edge of the triode region when we use Eq. (20.58) to size MWS. In many situations we may want to move M2 further into the saturation region. Intuitively, we would think that by increasing the length of MWS we increase its effective resistance and so the voltage drop across it goes up. If we increase MWS's length further, the gate voltage of M4 goes up and thus so does the drain voltage of M2. M2 is moved further away from the edge of the triode region into the saturation region. The minimum voltage across the current source increases. It's important to understand that it's OK for M2 to be biased away from the triode region by a small amount.

Example 20.7

Regenerate Fig. 20.29 using a wide-swing cascode structure.

The simulation results are seen in Fig. 20.33. It appears that the output resistance is really large until $V_O < V_{DS,sat}$. However, if we zoom in around $2V_{DS,sat}$ (= 500 mV here), we would see that the slope increases and M4 triodes. ■

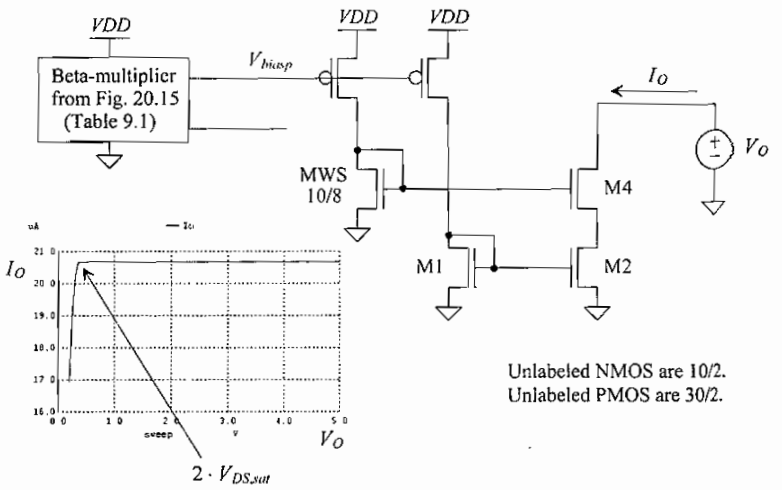


Figure 20.33 Wide-swing cascode current source in the long-channel process.

An Important Practical Note

In general we won't design a wide-swing current sink by setting the length to four times the length used in the rest of the design. As just mentioned, this biases M2 right on the edge of the triode region (point A) in Fig. 20.34. Stealing a current from the drain of M2 will move M2 into the triode region, point B. If we bias M2 further into the saturation region, point C, then more current must be removed before M2 triodes and the output resistance of the current source decreases. When we design folded-cascode amplifiers later, we will steal or add current in this way. For general *long-channel design,* we'll use

$$\frac{W_{MWS}}{L_{MWS}} = \frac{1}{5} \cdot \frac{W}{L} \qquad (20.59)$$

or using the same widths then

$$L_{MWS} = 5 \cdot L \qquad (20.60)$$

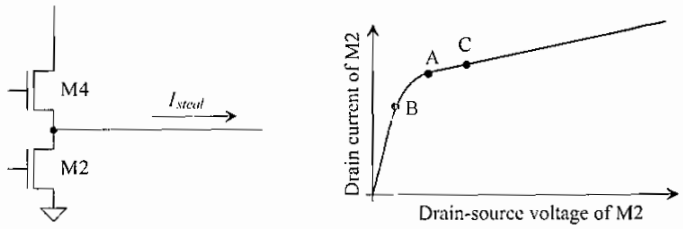


Figure 20.34 Showing how stealing current from M2 moves it into the triode region.

Layout Concerns

When we go to layout the long L device seen in Fig. 20.32, we might simply layout a single MOSFET with the appropriate length. However, the threshold voltage can vary significantly with the length of the device. Figure 20.35 shows a method where MOSFETs connected in series with the same widths and their gates tied together behave like a single MOSFET with the sum of the individual MOSFET's lengths. Because each device is identical, changes in the threshold voltage shouldn't affect the biasing circuit.

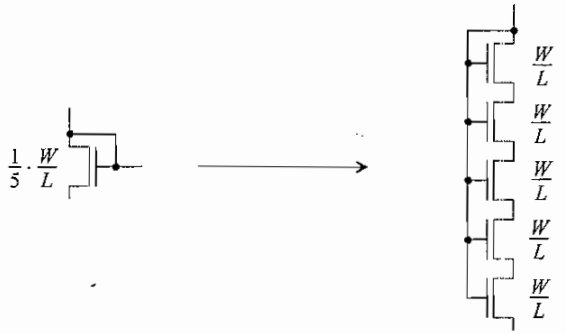


Figure 20.35 Using equal length devices to implement a long L MOSFET.

20.2.3 Wide-Swing, Short-Channel Design

When designing a wide-swing current mirror in a short-channel CMOS process, we know

$$V_{DS,sat} \neq V_{GS} - V_{THN} = V_{ovn} \tag{20.61}$$

Further, as seen in Fig. 9.32 in Ch. 9, the output resistance of a short-channel MOSFET depends on the drain-to-source voltage. Figure 20.36 shows a wide-swing current mirror designed in the short-channel CMOS process (see Table 9.2). Note that the current is approximately 4 µA when it should be, as seen in Fig. 20.23, 10 µA. Further, if we were to measure the output resistance, we would get approximately 600k, which is considerably less than the 4 MΩ seen in Table 20.1. As seen in Fig. 9.32, the output resistance at $V_{DS,sat}$ isn't 166k as indicated in Table 9.2 but much less. To increase the output resistance, let's attempt to bias M2 deeper into the saturation region. Let's do this by using a larger value of voltage on the gate of M4. For our short-channel design, let's try

$$\frac{W_{MWS}}{L_{MWS}} = \frac{1}{25}\cdot\frac{W}{L} \tag{20.62}$$

or a five times smaller ratio than what we used for the long-channel devices (Eq. [20.59]). The simulation results for the circuit in Fig. 20.36 are seen in Fig. 20.37 when MWS is sized 10/10. Notice that our output current is closer to 10 µA. Studying Fig. 20.36 and remembering our constant theme in this chapter that good matching requires both equal V_{GS} and V_{DS}, we see a concern. M1's drain-to-source voltage isn't the same as

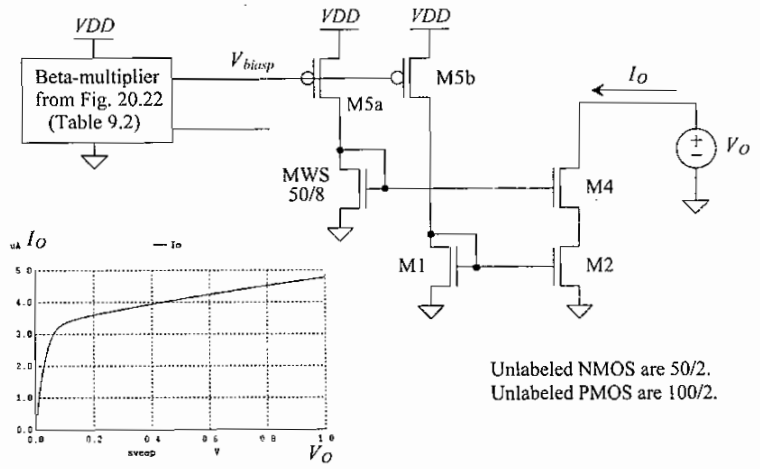
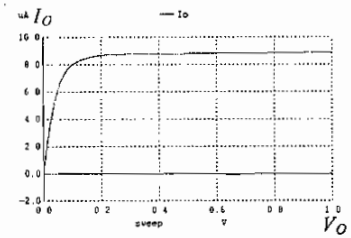


Figure 20.36 Wide-swing cascode current source in the short-channel process (bad).

M2's. By making the gate voltage of M4 larger, we act to equalize the two drain-to-source voltages (we increase the drain voltage of M2) causing the output current to approach the current that flows in M1 (10 µA). Let's attempt to make the current mirrors more symmetrical in an effort to equilibrate the drain-to-source voltages of M1 and M2.

Figure 20.38 shows the addition of a MOSFET, M3, to lower M1's V_{DS} so that it matches M2's V_{DS}. The 10 µA current through M5b can now be mirrored accurately. M1's gate voltage increases until M1 (and thus M2) can sink the current supplied by M5b with the smaller V_{DS}. *For all wide-swing current mirrors the topology seen in Fig. 20.38 using M3 should be used.* Note also, in Fig. 20.38, the larger voltage needed across the current source to keep M2/M4 saturated. The drains of M1 and M2 are at 150 mV or considerably above $V_{DS,sat}$ (which is 50 mV from Fig. 9.32). M3 is moving close to the triode region. Equation (20.62) can be modified to move M3 away from triode by using a larger W/L device (say 1/10); however, the output resistance will decrease (again as seen in Fig. 9.32, the decrease in V_{DS} results in smaller output resistance, r_o).



MWS is 10/10 in Fig. 20.36.

Figure 20.37 Increasing the W/L of MWS to 1/25 the other W/Ls.

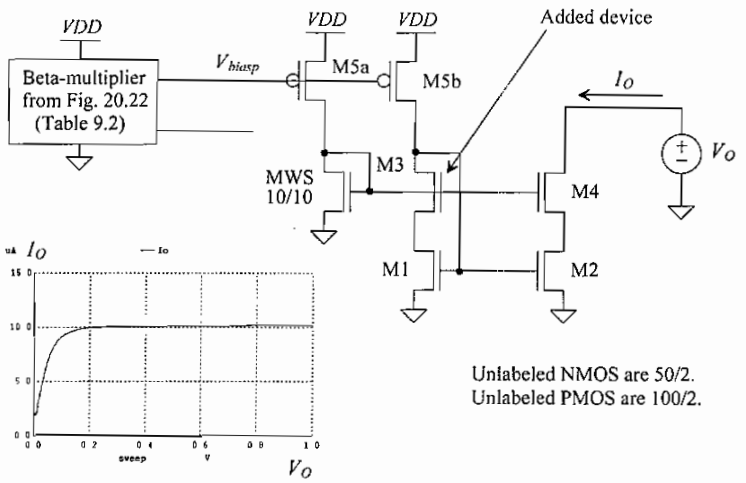


Figure 20.38 Wide-swing cascode current source in the short-channel process (good). Notice the drain-to-source voltages of M1 and M2 are the same.

Example 20.8

Regenerate the plot seen in Fig. 20.38 if MWS is sized using

$$\frac{W_{MWS}}{L_{MWS}} = \frac{1}{10} \cdot \frac{W}{L} \tag{20.63}$$

The simulation results are seen in Fig. 20.39 where MWS is sized 10/4. The output resistance is roughly four times smaller than what we got in Fig. 20.38 using Eq. (20.62). M3 is biased further away from the triode region. The point here is that there isn't any single equation that governs the selection of MWS size for all processes. Design trade-offs must be made.

Note that an interesting simulation to run at this point is to verify that the current mirror still functions correctly with small *VDD* (as seen in Fig. 20.23 with *VDD* down to 0.5 V). The output voltage, V_O, can still go to 1 V. ■

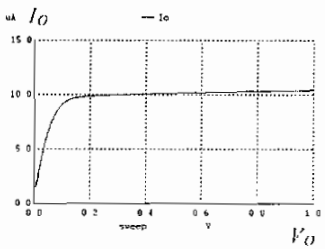


Figure 20.39 Resimulating the circuit in Fig. 20.39 using a 10/4 device for MWS.

20.2.4 Regulated Drain Current Mirror

Consider adding an amplifier to the basic wide-swing current mirror as seen in Fig. 20.40. The purpose of the amplifier is to regulate or hold the drain of M2 at a fixed potential (in this case, the drain potential of M1). If we can hold the drain potential of M2 perfectly fixed, then M2's drain current won't vary and the output resistance of the current mirror will be infinite. In the next section, we'll discuss general biasing circuits. We'll use the drain voltage of M1 as a bias voltage simply for regulating the drain in current mirrors.

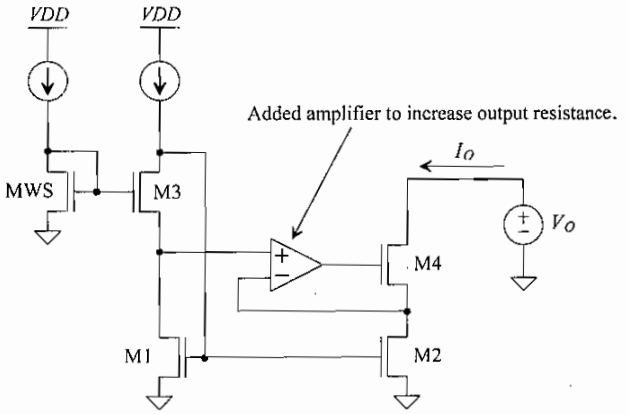


Figure 20.40 Regulating the drain of M2 using an amplifier.

We can estimate the enhancement in the output resistance of the current mirror by assuming the output of the added amplifier is related to its inputs using

$$v_{out} = A \cdot (v_+ - v_-) \tag{20.64}$$

where A is the gain of the added amplifier. Using Fig. 20.41, we can write, knowing the drain and gate of M1 are DC bias voltages (AC grounds),

$$v_{gs4} = -i_T r_o (A+1) \tag{20.65}$$

and

$$i_T = g_m(-i_T r_o(A+1)) + \frac{v_T - i_T r_o}{r_o} \tag{20.66}$$

and finally,

$$R_o = \frac{v_T}{i_T} = 2r_o + g_m r_o^2 (A+1) \approx g_m r_o^2 \cdot A \tag{20.67}$$

When compared to Eq. (20.53), the output resistance is increased by the gain of the added amplifier. This result can be very useful (and we will use it often) in practical circuit design.

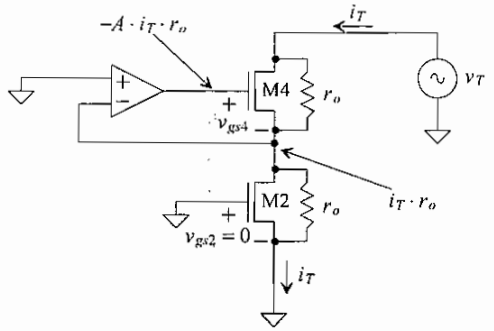


Figure 20.41 Determining the output impedance of a regulated drain current mirror.

A practical example of a current mirror that uses this technique is seen in Fig. 20.42. The drains of M1 and M2 are held at a V_{GS} by MA1 and MA2. If the drain potential of M2, for example, starts to decrease, MA2 starts shutting off, causing the gate voltage of M4 to increase, and pulling the drain potential of M2 back up. MA1 and MA2 are used to ensure that the circuit is symmetrical (M1 and M2 have the same V_{DS}). The drawback of this topology, as seen in the simulation results in the figure, is that the minimum voltage across the current mirror is $V_{DS,sat} + V_{GS}$ (just like the basic cascode, see Eq. [20.48]). The gain of the added amplifier (A) is, as we'll see in the next chapter, $g_m r_o/2$. We can estimate the output resistance as $g_m^2 r_o^3/2$ or 52 MΩ. Note that M3 is operating in the triode region.

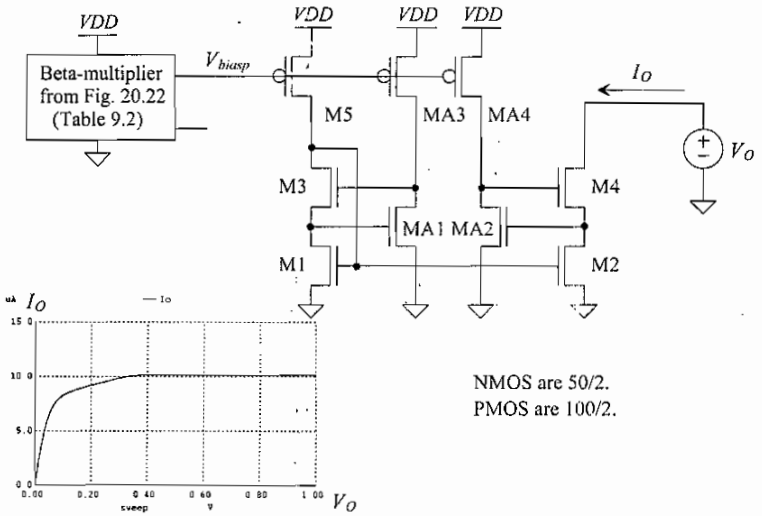


Figure 20.42 Using amplifiers to regulate the drain potentials in a current mirror.

20.3 Biasing Circuits

In this section we provide biasing circuit examples for the chapters that follow based on the data in Tables 9.1 and 9.2.

20.3.1 Long-Channel Biasing Circuits

Figure 20.43 shows a general biasing circuit for the long-channel CMOS process based on the sizes and currents given in Table 9.1. The Beta-multiplier, self-referenced circuit from Fig. 20.15 is used for biasing wide-swing current mirrors (V_{bias1} to V_{bias4}). V_{high} and V_{low} will be used later with amplifiers for regulating the drain of a MOSFET, as seen in Fig. 20.40. V_{ncas} and V_{pcas} are used for a circuit called a "floating current source" (which is discussed later). Perhaps the best way to see how this circuit can be used is to give a couple of examples.

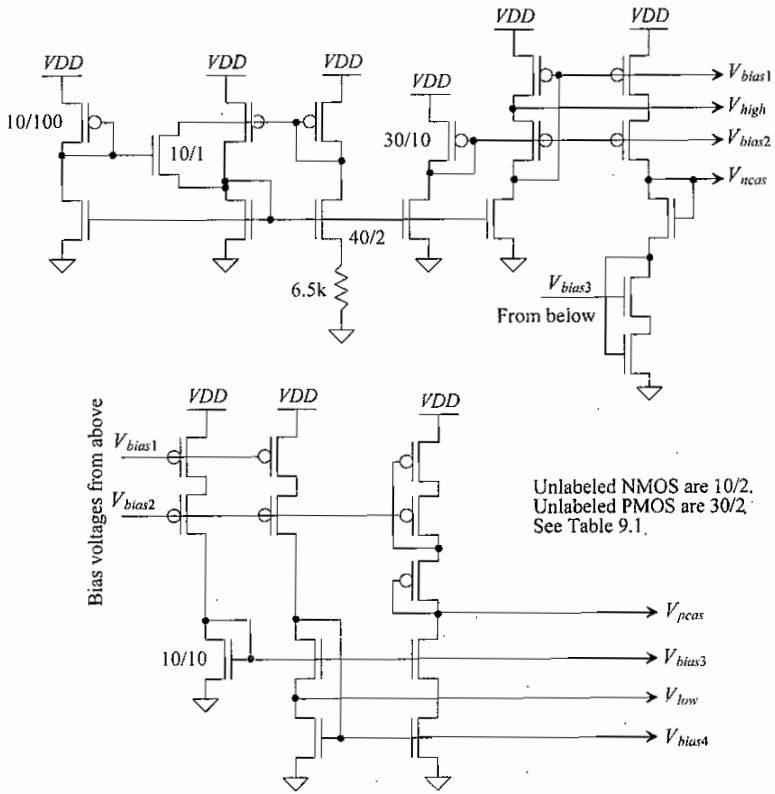
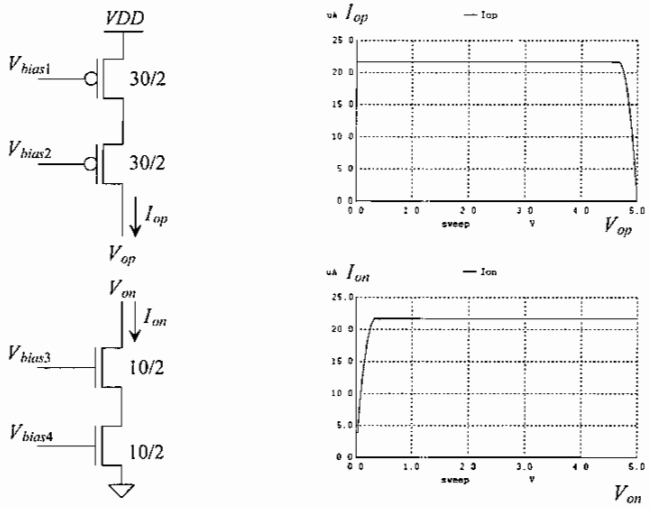


Figure 20.43 General biasing circuit for long-channel CMOS design using the data in Table 9.1

Basic Cascode Biasing

Figure 20.44 shows how the bias voltages generated in Fig. 20.43 can be used to bias both PMOS and NMOS cascode current sources. As indicated in Table 9.1 and Fig. 20.31b, the minimum voltage across the current sources is $2V_{DS,sat}$ or 500 mV. The current that flows in the current sources is nominally 20 µA.



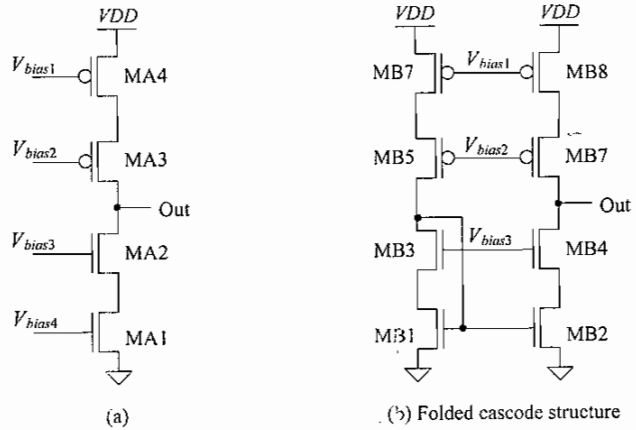
Bias voltages come from Fig. 20.43 (long-channel parameters in Table 9.1).

Figure 20.44 How cascode currents are biased and how they operate.

The Folded-Cascode Structure

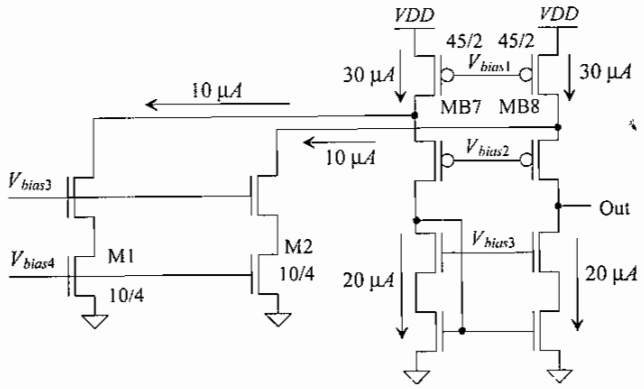
Figure 20.45a shows a structure where an NMOS current source is connected to a PMOS current source. In this topology there is no way we can guarantee that the current flowing in the PMOS transistors is precisely equal to the current flowing in the NMOS transistors. This will cause the output of the circuit to move towards either *VDD* or ground (depending on which cascode structure is sourcing the most current). In (b) we fold the cascode structure over and diode-connect the folded NMOS devices (we could, just as well, have diode-connected the PMOS devices instead). Now, the gate potentials of MB1 and MB2 are set by the current that is sourced from MB5 and MB7. If the transistors are perfectly matched, the output voltage will equal the gate voltage of MB1 (the drain voltage of MB3) because of the circuit's symmetry. (We also saw this in Fig. 20.1.)

In Figure 20.46 we add some MOSFETs to steal 10 µA from MB7 and MB8. To make sure that the currents are equal, we size up MB7 and MB8 so that they source 30 µA. Note how we doubled the length of M1 and M2 so that they sink 10 µA from the same bias voltages.



Bias voltages come from Fig. 20.43 (long-channel parameters in Table 9.1). All NMOS are 10/2, while all PMOS are 30/2.

Figure 20.45 Using a folded cascode structure to make sure that the current sourced by the PMOS equals the current sourced by the NMOS.



Bias voltages come from Fig. 20.43 (long-channel parameters in Table 9.1). All unlabeled NMOS are 10/2, while all unlabeled PMOS are 30/2.

Figure 20.46 Stealing current from the folded cascode structure.

20.3.2 Short-Channel Biasing Circuits

Figure 20.47 shows a biasing circuit for a general analog design short-channel process. It uses the Beta-multiplier, self-referenced bias circuit from Fig. 20.22. As indicated in Sec. 20.1.4, the critical capacitance for stability is MCP. Notice that we are using the PMOS devices in the Beta-multiplier to bias the current mirrors, whereas in Fig. 20.43 we used the NMOS devices. We picked the PMOS here simply to increase the capacitance on V_{biasp} and so to further stabilize the circuit. Figure 20.48 shows the operation of the cascode current mirrors biased with the circuit in Fig. 20.47. It's important to realize that the minimum voltage across the current source is considerably above $2V_{DS,sat}$. Again, this was a design choice (see Eqs. [20.62] and [20.63]) to increase output resistance.

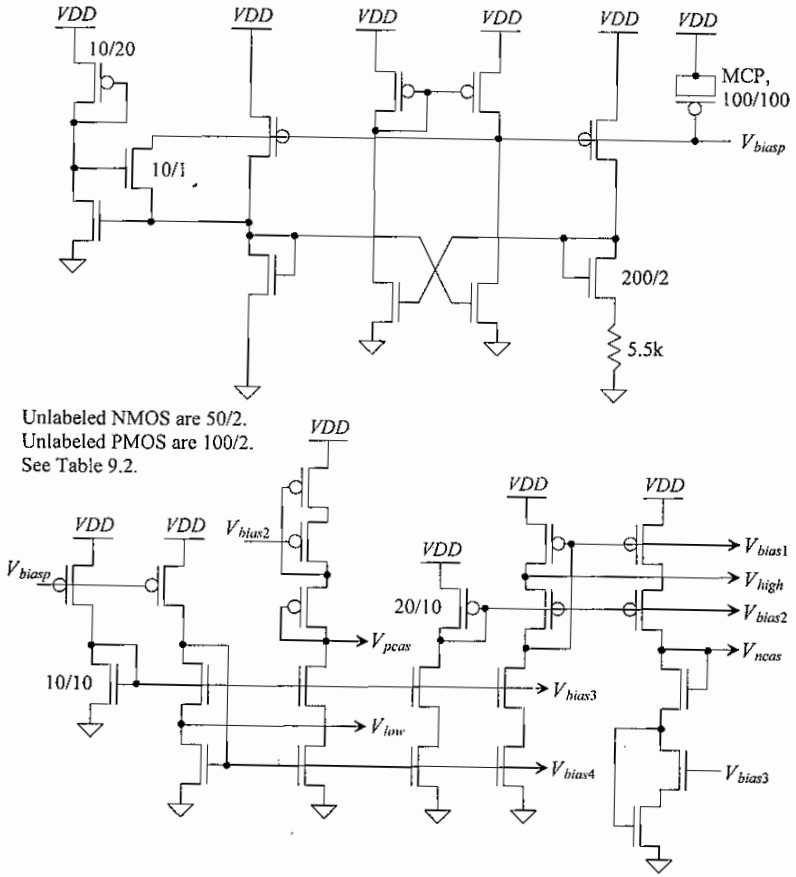
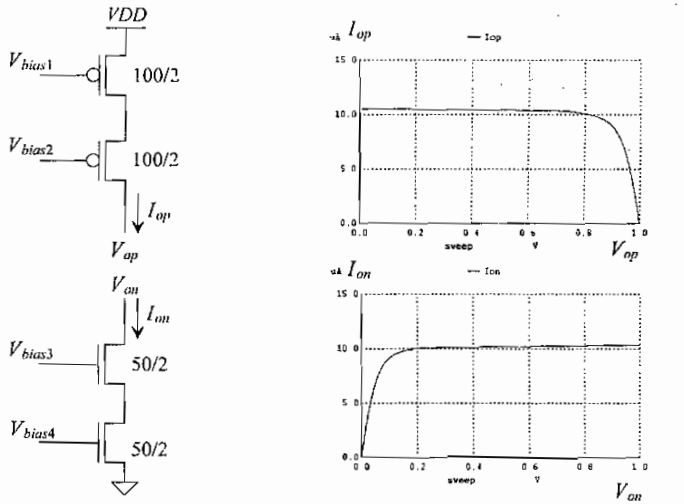


Figure 20.47 General biasing circuit for short-channel design using the data in Table 9.2.



Bias voltages come from Fig. 20.47 (short-channel parameters in Table 9.2).

Figure 20.48 Cascode current sources operating in a short-channel process.

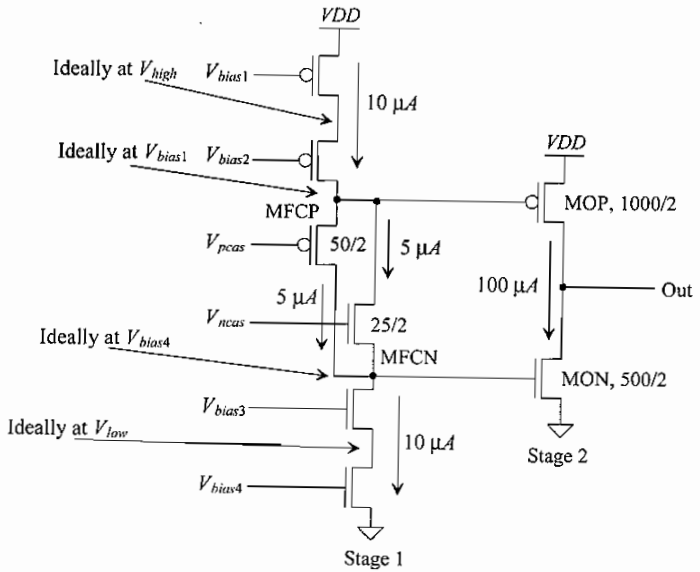
Floating Current Sources

In Fig. 20.49 we add two MOSFETs, MFCP and MFCN, to the cascode structure to form a *floating current source*. The addition of MFCN, for example, allows the voltage across the PMOS cascode structure to become V_{bias1} (assuming that matching between the bias circuit and this circuit is perfect). The voltage across the NMOS cascode becomes V_{bias2}. This can be very useful for biasing the next stage in an amplifier. For example, if we connect an NMOS (MON in the figure) device to the source of MFCN, then, because this potential is ideally V_{bias4}, we treat the MOSFET as if it were biased with V_{bias4}. Since the width of MON in the figure is ten times the widths of the other NMOS, the current in the output stage will be 100 μA. Figure 20.50 shows the simulation results with currents that flow in stage 1 and stage 2 of the circuit.

Note that the circuit in Fig. 20.49 uses all of the bias voltages except for V_{low} and V_{high}. These voltages are used with circuits that employ drain regulation, as discussed in Sec. 20.2.4.

20.3.3 A Final Comment

The bias circuits in Figs. 20.43 and 20.47 will be used in the next several chapters to provide biasing for amplifier design examples. It's important to remember our constant theme when we design current mirrors, that is, if we can set both the gate-source and the drain-source voltages of a MOSFET, we set the drain current. Similarly if we can set the drain current and gate-source voltage, then we set the drain-source voltage.



Bias voltages come from Fig. 20.47 (short-channel parameters in Table 9.2). Unlabeled NMOS are 50/2, while unlabeled PMOS are 100/2.

Figure 20.49 Biasing with a floating current source.

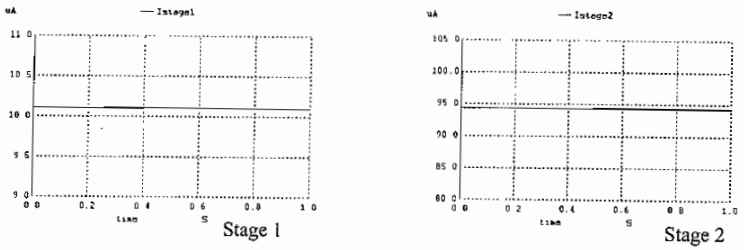


Figure 20.50 The simulated currents that flow in stages 1 and 2 of the circuit in Fig. 20.49.

ADDITIONAL READING

[1] S. J. Lovett, M. Welten, A. Mathewson, and B. Mason, "Optimizing MOS transistor mismatch," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 147–150, January 1998.

[2] K. de Langen and J. H. Huijsing, "Compact low-voltage power-efficient operational amplifier cells for VLSI," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 1482–1496, October 1998.

[3] P. Larsson, "Parasitic resistance in an MOS transistor used as on-chip decoupling capacitance," *IEEE Journal of Solid-State Circuits*, vol. 32, pp. 574–576, April 1997.

[4] C. Yu and R. L. Geiger, "An automatic offset compensation scheme with ping-pong control for CMOS operational amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 601–610, May 1994.

[5] R. G. Eschauzier, R. Hogervorst, and J. H. Huijsing, "A programmable 1.5 V CMOS class-AB operational amplifier with hybrid nested Miller compensation for 120 dB gain and 6 MHz UGF," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 1497–1504, December 1994.

[6] R. Hogervorst, J. P. Tero, R. G. H. Eschauzier, and J. H. Huijsing, "A Compact Power-Efficient 3 V CMOS Rail-to-Rail Input/Output Operational Amplifier for VLSI Cell Libraries," *IEEE Journal of Solid State Circuits*, vol. 29, pp. 1505–1513, December 1994.

[7] E. Säckinger and W. Guggenbühl, "A High-Swing, High-Impedance MOS Cascode Circuit," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 289–298, February 1990.

[8] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, pp. 1433–1439, October 1989.

[9] K. Lakshmikumar, R. A. Hadaway, and M. A. Copeland, "Characterisation and Modeling of Mismatch in MOS Transistors for Precision Analog Design," *IEEE Journal of Solid-State Circuits*, vol. 21, pp. 1057–1066, December 1986.

[10] J. Shyu, F. Krummenacher, and G. C. Temes, "Random error effects in matched MOS capacitors and current sources," *IEEE Journal of Solid-State Circuits*, vol. 19, pp. 948–956, December 1984.

[11] P. R. Gray and R. G. Meyer, "MOS operational amplifier design - A tutorial overview," *IEEE Journal of Solid-State Circuits*, vol. 17, pp. 969–982, December 1982.

[12] J. L. McCreary, "Matching properties, and voltage and temperature dependence of MOS capacitors," *IEEE Journal of Solid-State Circuits*, vol. 16, pp. 608–616, December 1981.

[13] W. Steinhagen and W. L. Engl, "Design of integrated analog CMOS circuits - A multichannel telemetry transmitter," *IEEE Journal of Solid-State Circuits*, vol. 13, pp. 799–805, December 1978.

[14] E. Vittoz and J. Fellrath, "CMOS analog integrated circuits based on weak inversion operations," *IEEE Journal of Solid-State Circuits*, vol. 12, pp. 224–231, June 1977.

PROBLEMS

20.1 Using the CMOS long-channel process (1 μm), determine the current flowing in the circuit seen in Fig. 20.51. Verify your answer with SPICE.

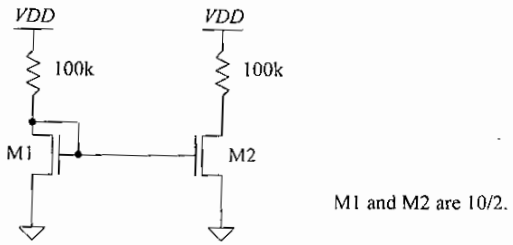


Figure 20.51 Current mirror used in Problem 20.1.

20.2 Repeat problem 20.1 for the circuit in Fig. 20.52. Can M1 and M2 be replaced with a single MOSFET? If so how and what size? If not why?

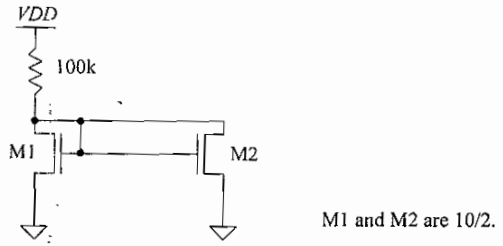


Figure 20.52 Circuit used in Problem 20.2.

20.3 Show the SPICE simulations for the PMOS devices in Ex. 20.1.

20.4 A threshold voltage mismatch in SPICE can be modeled by adding a small voltage source in series with the gate of one of the MOSFETs, as seen in Fig. 20.53. Show, using SPICE, that Eq. (20.8) is valid.

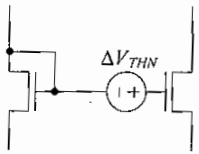


Figure 20.53 Modeling an offset voltage in SPICE (or for hand calculations).

20.5 Show, using simulations and hand-calculations, that by using a larger value of $V_{DS,sat}$ when designing bias circuits, the MOSFETs enter the triode region earlier.

20.6 In Ex. 20.2, how does the gate voltage of M1/M2 change as *VDD* is decreased? How does the V_{SG} change? Use SPICE to verify your answers.

20.7 For the same bias current used in Table 9.1 (nominally 20 μA), regenerate Fig. 20.13 if minimum-length MOSFETs are used in the current mirror. Show the hand calculations leading to the selection of the MOSFET sizes.

20.8 Suppose that the bias circuit used to generate Fig. 20.16 shows a reference current change with *VDD*, as seen in Fig. 20.54. Knowing that the reference current should be constant with changes in *VDD*, what is wrong with the reference? (What important portion of the reference is causing the problem?)

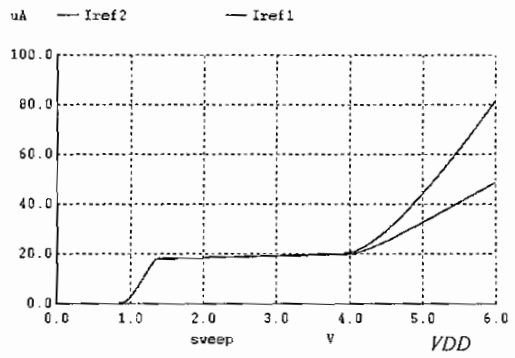


Figure 20.54 Problems with reference current increasing with VDD.

20.9 Using hand calculations, estimate the value of the voltage across *R* for the current reference in Fig. 20.15.

20.10 Using SPICE, determine if the reference circuit seen in Fig. 20.22 can become unstable with changes in *VDD*. Show that reducing the size of MCP and MCN affects the stability of the circuit.

20.11 Estimate the temperature behavior of the gate voltage of M1 in Fig. 20.10.

20.12 Design a voltage reference using the Beta-multiplier in the short-channel process and the data from Table 9.2. Generate a figure similar to Fig. 20.27 to show the temperature performance of your design.

20.13 *K* can be used to minimize *R* in Eq. (20.46) for a fixed-reference current. If M1/M2 experience a threshold voltage mismatch (see problem 20.4), discuss and show with simulations how much ΔV_{THN} can be tolerated and the error in the reference current that results.

20.14 Using an AC test voltage, determine the output resistance of the MOSFET current mirror seen in Fig. 20.28.

20.15 Based on the data in Table 9.1, what are the voltages on the drain, gate, and source terminals of M1–M4 in Fig. 20.29, assuming that the devices are operating in the saturation region? Compare your hand calculations to simulations.

20.16 If the MOSFET in Fig. 20.55 is operating in the saturation region, determine the small-signal resistance looking into its drain.

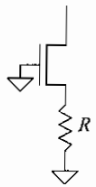


Figure 20.55 Resistance looking into the drain of a MOSFET with a source resistance. See Problem 20.16.

20.17 Suppose that the wide-swing current mirror seen in Fig. 20.38 is implemented in the long-channel CMOS process (Table 9.1). Estimate the size of MWS when M3 triodes. Verify your answer with simulations.

20.18 Label the voltages, based on the values in Table 9.1, in the schematic of the general biasing circuits seen in Fig. 20.43. Compare the tabulated values of V_{GS}, V_{DS}, etc., to the simulated values.

20.19 How would the current in Fig. 20.44 change if the width of the device connected to V_{bias2} were doubled? Were halved? Why? Explain what is going on in the circuit.

20.20 Should the voltage labeled "Out" in Fig. 20.49 be at a specific value? Why or why not?

Chapter
21

Amplifiers

In this chapter we turn our attention towards amplifiers. Single-stage amplifiers are used in virtually every op-amp design. By replacing a passive load resistor with a MOSFET transistor (called an *active load*), significant amounts of chip area can be saved. An active load can also produce higher values of resistance when compared with a passive resistor, resulting in higher gains.

Several types of active loads are studied in this chapter. The *gate-drain load* consists of a MOSFET with the gate and drain shorted and provides large bandwidths and low-output impedance at the cost of reduced gain. The *current source load* amplifier has a higher gain and output impedance at the expense of lower bandwidth. Current source load amplifiers are preferred when external feedback is applied to the amplifier to set the gain. This chapter explores basic single-stage amplifiers with active loads, as well as the trade-offs associated with each. The cascode amplifier is examined in detail along with several configurations of output stages, including the push-pull amplifier.

21.1 Gate-Drain-Connected Loads

A gate-drain-connected MOSFET (load) is half of a current mirror as seen in Fig. 20.1 in the last chapter. Here we move the discussion from the DC operating conditions and biasing presented in the last chapter to AC small-signal analysis.

21.1.1 Common-Source (CS) Amplifiers

A gate-drain-connected MOSFET (operating with a nonzero drain current) can be thought of as a resistor of value $1/g_m$ (see Fig. 9.18 and Eq. [9.25]). The four possible common- source (CS) amplifiers using gate-drain-connected loads are seen in Fig. 21.1. In each configuration, M1 and M2 are assumed to be biased in the saturation region. Notice how the source of the amplifying MOSFET (i.e., not the load or gate-drain MOSFET) is common to both the input and the output in each configuration.

Examine the circuit shown in Fig. 21.1a. Since many MOSFET amplifiers are analyzed in this chapter and the next several chapters using small-signal analysis, an intuitive analysis approach will be used. This approach allows the designer to quickly analyze the gain of a circuit. To analyze the gain of the circuit in Fig. 21.1, begin by

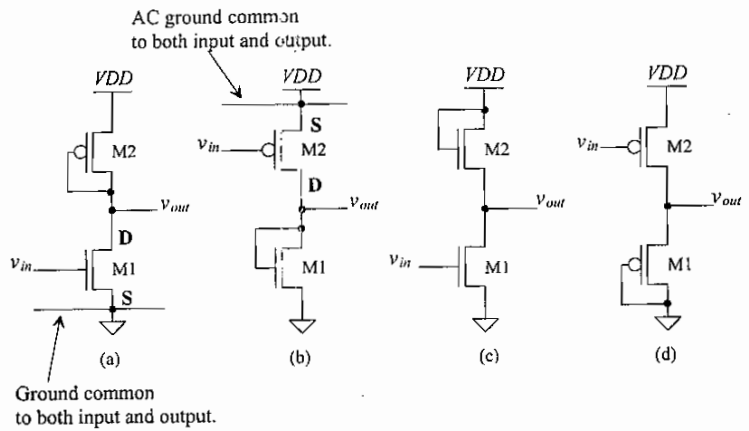


Figure 21.1 Four possible configurations of common-source amplifiers with gate-drain loads.

replacing M2 with a resistor of value $\frac{1}{g_{m2}}$ and replacing M1 with a current source of value $g_{m1} \cdot v_{in}$, provided $\frac{1}{g_{m2}} << r_{o1} || r_{o2}$. The equivalent circuit is shown in Fig. 21.2. Note that there is no body effect in either MOSFET and that only the low-frequency model is shown. Again, as seen in Ch. 9, a gate-drain-connected MOSFET has a small-signal resistance of value $\frac{1}{g_m}$. The small-signal gain of the **common-source amplifier** is given by

$$\frac{v_{out}}{v_{in}} = \frac{-i_d \cdot \frac{1}{g_{m2}}}{i_d \cdot \frac{1}{g_{m1}}} = -\frac{\frac{1}{g_{m2}}}{\frac{1}{g_{m1}}} = -\frac{\text{resistance in the drain}}{\text{resistance in the source}} = -\frac{g_{m1}}{g_{m2}} \qquad (21.1)$$

This result is very important in the intuitive analysis. It states that the small-signal gain of a common-source amplifier is simply the resistance in the drain of M1 divided by the resistance in the source (the resistance looking into the source of M1 added to any resistance from the source of M1 to ground). The actual value of the resistance in the

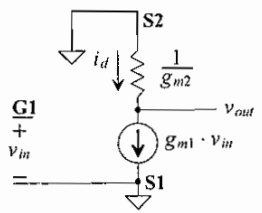


Figure 21.2 Simplified circuit of Fig. 21.1a.

drain, as will be soon seen, is $\frac{1}{g_{m2}} || r_{o1} || r_{o2}$. In most cases, however, r_o will be much greater than $\frac{1}{g_m}$, and the approximation is an accurate one.

Example 21.1
Determine the small-signal AC gain of the circuit shown in Fig. 21.3 using the intuitive method presented in Eq. (21.1).

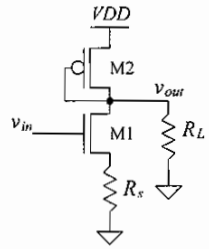


Figure 21.3 Amplifier analyzed in Ex. 21.1.

The small-signal gain of this circuit is given by

$$A_v = \frac{v_{out}}{v_{in}} = -\frac{\text{resistance in the drain}}{\text{resistance in source}}$$

The resistance in the drain is the parallel combination of all resistances connected to the drain of M1, given by

$$\text{Resistance in the drain} = \frac{1}{g_{m2}} || R_L \text{ for } r_{o1} || r_{o2} >> \frac{1}{g_{m2}}$$

The resistance in the source is the sum of the resistance looking into the source of M1 ($1/g_{m1}$) and the resistance connected from the source of M1 to ground. For the present problem, this resistance is given by

$$R_{\text{in the source}} = \frac{1}{g_{m1}} + R_s$$

The voltage gain is then given by

$$A_v = -\frac{\frac{1}{g_{m2}} || R_L}{\frac{1}{g_{m1}} + R_s} \text{ which for } R_L \to \infty \text{ and } R_s \to 0 \text{ reduces to } -\frac{g_{m1}}{g_{m2}} \blacksquare$$

We can determine the exact gain of the amplifier of Fig. 21.1a using the full small-signal model shown in Fig. 21.4. Using KCL at the output of the amplifier gives

$$g_{m1} v_{in} + \frac{v_{out}}{r_{o2} || r_{o1}} = -g_{m2} v_{out} \qquad (21.2)$$

or

$$A_v = \frac{v_{out}}{v_{in}} = -\frac{g_{m1}}{g_{m2} + \frac{1}{r_{o1} || r_{o2}}} = -\frac{\frac{1}{g_{m2}} || r_{o1} || r_{o2}}{\frac{1}{g_{m1}}} = -\frac{\text{resistance in the drain}}{\text{resistance in the source}}$$

(21.3)

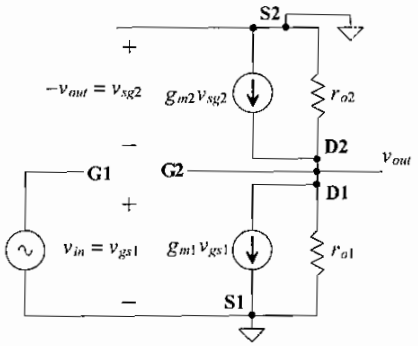


Figure 21.4 Small-signal model of the amplifier shown in Fig. 21.1(a).

When $\frac{1}{g_{m2}} << r_{o1} || r_{o2}$, this reduces to

$$A_v = -\frac{\frac{1}{g_{m2}}}{\frac{1}{g_{m1}}} = -\frac{g_{m1}}{g_{m2}} \qquad (21.4)$$

which is the same form as Eq. (21.1).

Miller's Theorem

Consider the inverting amplifier seen in Fig. 21.5a with a gain of $-A_v$ and a feedback capacitance between the output and the input of the amplifier. The current supplied by v_{in} or v_{out} to the capacitor is

$$i = \frac{v_{in} - v_{out}}{1/j\omega C_F} = j\omega C_F \cdot (1 + |A_v|) \cdot v_{in} = j\omega C_F \cdot \left(1 + \frac{1}{|A_v|}\right) \cdot v_{out} \qquad (21.5)$$

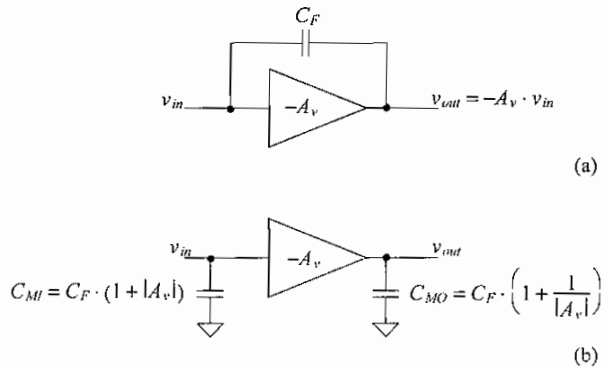


Figure 21.5 The derivation of Miller's theorem.

In Fig. 21.5b we replace the feedback capacitor with capacitors on the input and the output of the amplifier to ground. These (adjusted size) capacitors require the same currents from the input source and the output of the amplifier. Note that this substitution, Eq. (21.5), is often called *Miller's theorem*, and the feedback capacitor, C_F, is called the *Miller capacitance*.

Notice that the effective capacitance on the input of the amplifier is multiplied by the gain of the amplifier. For big gains, $|-A_v|$, this means that the input capacitance of the amplifier can be large and can result in slow circuits. Intuitively, this can be understood by realizing that the effective voltage across C_F is $A_v + 1$ times larger than the input signal itself. This large voltage drop across C_F (one side going up by v_{in} while the other side goes down by $A_v \cdot v_{in}$) results in a large displacement current, which makes the capacitor appear larger than it really is.

Frequency Response

Now consider the frequency response of the CS amplifier of Fig. 21.1a. The high-frequency (meaning that the device capacitances are included in the circuit) equivalent circuit is shown in Fig. 21.6a. The gate-drain capacitance of M2 isn't drawn because the gate and drain of M2 are shorted together. A source resistance is added to the circuit to model the effects of the driving source impedance. Miller's theorem is used to break C_{gd1} into two parts: a capacitance at the gate of M1 to ground and a capacitance from the drain of M1 to ground, as seen in Fig. 21.6b (note that,the low-frequency gain of the amplifier in Fig. 21.1a is $-g_{m1}/g_{m2}$). Two *RC* time constants exist in this circuit: one on the input of the circuit and one on the output of the circuit. The input time constant is given by

$$\tau_{in} = R_s(C_{Mf} + C_{gs1}) \qquad (21.6)$$

where the Miller capacitance at the input, C_{Mf}, is

$$C_{Mf} = C_{gd1}(1 + \frac{g_{m1}}{g_{m2}}) \qquad (21.7)$$

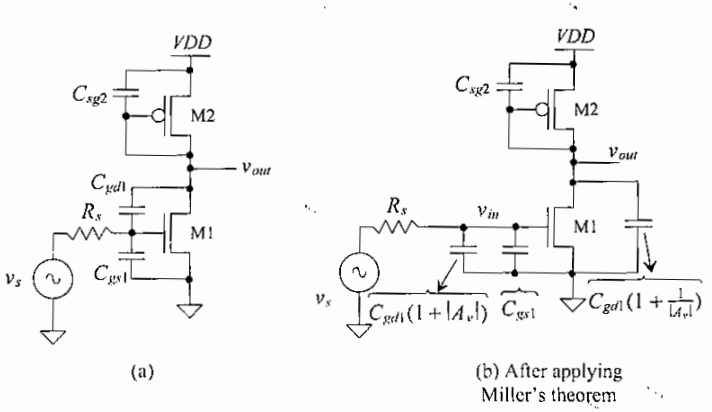


Figure 21.6 Frequency response of the CS amplifier with gate-drain-connected load.

The output time constant is found in a similar way and is given by

$$\tau_{out} = \frac{1}{g_{m2}} \cdot (C_{sg2} + C_{MO}) \tag{21.8}$$

The Miller capacitance which appears at the output, C_{MO}, now becomes

$$C_{MO} = C_{gd1}(1 + \frac{g_{m2}}{g_{m1}}) \tag{21.9}$$

The frequency response of the amplifier in Fig. 21.1a (Fig. 21.6a) is given by

$$A_v(f) = \frac{-\frac{g_{m1}}{g_{m2}}}{\left(1 + j\frac{f}{f_{in}}\right)\left(1 + j\frac{f}{f_{out}}\right)} \tag{21.10}$$

where the pole associated with the amplifier's input is located at

$$f_{in} = \frac{1}{2\pi\tau_{in}} \tag{21.11}$$

and the pole associated with the amplifier's output is located at

$$f_{out} = \frac{1}{2\pi\tau_{out}} \tag{21.12}$$

The Right-Half Plane Zero

It should be noted that when using Miller's theorem a zero is neglected. If we take a look at Fig. 21.5a, we see that at high frequencies C_F shorts the amplifier's output to its input. Our model in Fig. 21.5b doesn't show this shorting. What this means for the amplifier in Fig. 21.1a or its model in Fig. 21.6a is that C_{gd1} shorts the output to the input at high frequencies.

To characterize this high-frequency effect, where we can't use Miller's theorem, consider the circuit seen in Fig. 21.7. In this figure we model the amplifier in Fig. 21.6 (or Fig. 21.1) with a small-signal resistance on the output of the circuit called $R_o = \frac{1}{g_{m2}} || r_{o2} || r_{o1} \approx \frac{1}{g_{m2}}$ and a capacitance on the output, without including C_{gd1}, of $C_o = C_{sg2}$. We don't concern ourselves with the input time constant but rather just look at the output of the circuit. Summing the currents on the output of the model results in

$$\frac{v_{out} - v_{in}}{1/j\omega C_{gd1}} + \frac{v_{out}}{R_o || 1/j\omega C_o} + g_{m1} \cdot v_{in} = 0 \tag{21.13}$$

Solving for v_{out}/v_{in} gives

$$\frac{v_{out}}{v_{in}} = -g_{m1}R_o \cdot \frac{1 - j\omega\frac{C_{gd1}}{g_{m1}}}{1 + j\omega(C_{gd1} + C_o) \cdot R_o} \tag{21.14}$$

The pole should be recognized as what we wrote in Eq. (21.12) if the gain of the amplifier is small ($A_v = -g_{m1}/g_{m2} \approx 0$). In the numerator we see a right-half plane zero at

$$f_z = \frac{g_{m1}}{2\pi C_{gd1}} \tag{21.15}$$

A zero in the right-half plane (RHP) results in the same magnitude response as a zero in the left-half plane (LHP). However, the phase response is different. A zero in the RHP has the same influence on the phase of a system as a pole in the LHP. Understanding this

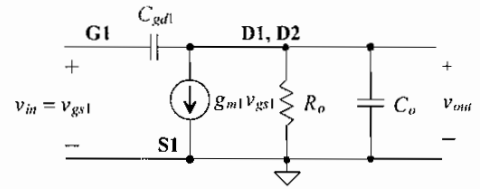


Figure 21.7 Small-signal model used to calculate the RHP zero for amplifiers in Fig. 22.1a–d.

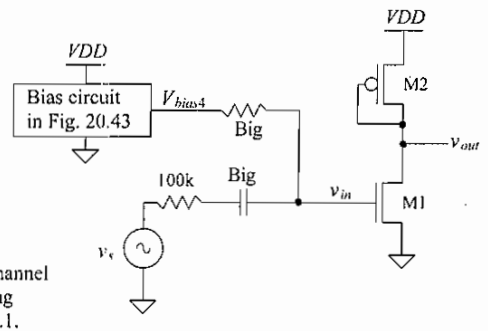
point is important. For the inverting amplifiers (which mean that at low frequencies their phase shift is 180°) in Fig. 21.1, for example, the input feeding directly to the output because of the RHP zero, without the inversion, can result in instability when the amplifier is used with feedback. The output can feed back and add to the input (positive feedback). The result is that the output of the amplifier will grow without limit (and, ultimately, the output of the amplifier will oscillate back and forth between some voltages). The magnitude response of the RHP zero is usually ignored because its value typically places it well beyond the pole frequency. However, in CMOS op-amp design, the phase shift of this RHP zero can, and will, result in stability and/or settling time issues.

Example 21.2

Determine the gain, magnitude, and phase shift of the amplifier shown in Fig. 21.8. Using an AC SPICE simulation, verify your hand calculations.

The bias circuit is used to bias the amplifier at the operating point seen in Table 9.1, that is, at a drain current of 20 μA. The big resistor and capacitor set the DC bias point but don't affect the AC operation of the circuit.

$$A_v = -\frac{\frac{1}{g_{m2}}}{\frac{1}{g_{m1}}} = -\frac{g_{m1}}{g_{m2}} = -\frac{150\ \mu A/V}{150\ \mu A/V} = -1$$



Use the long-channel sizes and biasing seen in Table 9.1.

Figure 21.8 Amplifier used in Ex. 21.2.

From Table 9.1, $C_{gs1} = 23.3\ fF$ and $C_{gd1} = 2\ fF$. The Miller input capacitance, because the gain of the amplifier is −1, is then 4 *f*F. The input pole is located at

$$f_{in} = \frac{1}{2\pi \cdot R_s(C_{M1} + C_{gs1})} = \frac{1}{2\pi \cdot 100k \cdot (27.3\,fF)} = 58\ MHz \qquad (21.16)$$

From Table 9.1, $C_{sg2} = 70\ fF$. The Miller output capacitance is, again, 4 *f*F. The output pole is located at

$$f_{out} = \frac{1}{2\pi \cdot \frac{1}{g_{m2}} \cdot (C_{MO} + C_{sg2})} = \frac{1}{2\pi \cdot 6.5k \cdot (74\,fF)} = 331\ MHz \qquad (21.17)$$

The zero is located at

$$f_z = \frac{g_{m1}}{2\pi C_{gd1}} = \frac{150\ \mu A/V}{2\pi \cdot 2\,fF} = 11.9\ GHz$$

The transfer function of the amplifier is then

$$A_v(f) = \frac{v_{out}(f)}{v_s(f)} = -\frac{\left(1 - j\frac{f}{11.9\ GHz}\right)}{\left(1 + j\frac{f}{58\ MHz}\right)\left(1 + j\frac{f}{331\ MHz}\right)}$$

Figure 21.9 shows the AC analysis simulation results for the amplifier of Fig. 21.8. What would happen if we used v_{in} instead of v_s? We wouldn't see the input

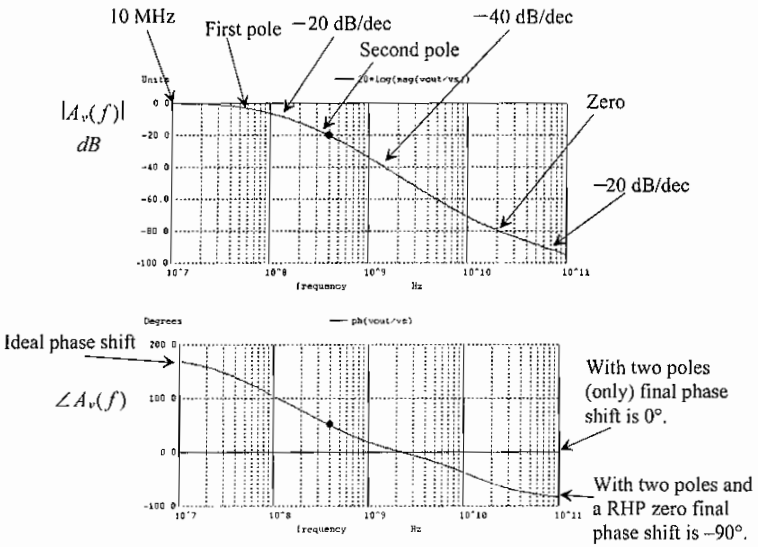


Figure 21.9 Frequency response of the amplifier in Fig. 21.8.

pole's effects on the frequency response. In the magnitude response we see that right after the first pole the gain decreases by a factor of 10 for every increase in frequency by 10 (equivalent to saying −20 dB/decade). We could also say that the gain decreases by 2 for every increase in frequency by 2 (equivalent to saying −6 dB/octave). After the second pole, the gain decreases by 100 for every increase in frequency by 10. The zero causes the frequency response to go back to −20 dB/decade. (The output and input of M1 are shorted together so that all we see are the effects of R_s charging the input capacitance of the amplifier.) It is useful to look at, and understand, the behavior of the amplifier from v_{in} to v_{out} in the simulation.

The effects of the RHP zero occur one decade before f_z. This *excess phase* shift is the problem we discussed earlier. It makes the amplifier appear not like an inverting amplifier (phase shift of −180 degrees) but rather like a noninverting amplifier sooner (at lower frequencies).

Note that magnitude of voltage gains in the neighborhood of 1 are common when using gate-drain-connected loads. At this point it may appear that this amplifier is almost worthless; however, we will see that when this amplifier is used with additional circuitry (e.g., a current source load for a differential pair), it can be very useful. ■

Example 21.3

Determine the magnitude and phase of the time-domain signal v_{out}, at 400 MHz, for the amplifier in Ex. 21.2 based on the frequency plots seen in Fig. 21.9. Verify the results using SPICE.

The magnitude response at 400 MHz is

$$\left|\frac{v_{out}}{v_s}\right| = -20\ dB = 0.1$$

and the phase response is

$$\angle\frac{v_{out}}{v_{in}} = 50° = -310°$$

The positive value means that the output is leading, or occurring earlier in time, than the input. Remembering that phase shift is simply an indication of a time delay at a particular frequency, we can write

$$50° = 360 \cdot \frac{t_d}{T} = 360 \cdot t_d \cdot 400\ MHz \rightarrow t_d \approx 350\ ps$$

for a period of 2.5 ns. We'll use an input voltage of 1 mV to ensure small-signal operation. In the simulation that generated Fig. 21.9 our input AC signal was 1 V. Again, remember that SPICE replaces the active devices with linear models in an AC simulation so any value of amplitude can be used for the input signals. Using 1 V in an AC simulation is convenient because of the ratios often used when calculating gains. Figure 21.10 shows the SPICE simulation results. Notice how the peak amplitude of the input is 1 mV while the peak amplitude of the output (remembering the DC output voltage is $VDD - V_{SG}$ or, from Table 9.1, approximately 3.85 V) is 0.1 mV (−20 db down from the low-frequency value). In a transient simulation we have to wait some time before the circuit settles to a

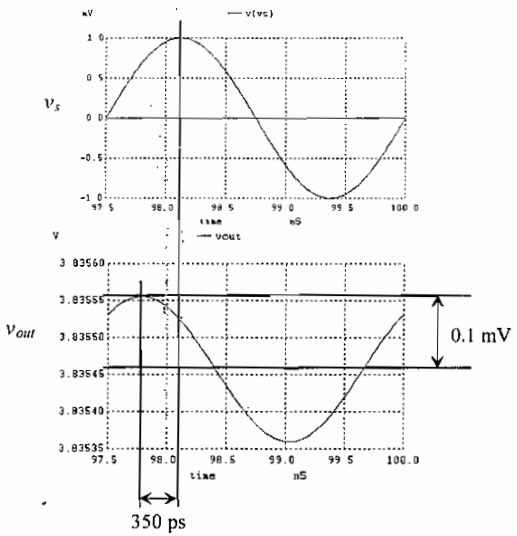


Figure 21.10 Time-domain behavior of the amplifier in Fig. 21.8 at 400 MHz.

stable operating point. In Fig. 21.10 we've waited 97.5 ns before looking at the output voltage. This gives the bias circuit, for example, time to turn on and generate the bias voltage needed in the amplifier (the value of the big capacitor was also reduced in this transient simulation so that its charging wouldn't increase the simulation time.) ■

A Common-Source Current Amplifier

The previous discussion was centered around CS voltage amplifiers. Next consider the circuit shown in Fig. 21.11. This amplifier can be thought of as a transimpedance amplifier, that is, voltage out and current in. The ideal input impedance of an amplifier with a current input is zero (all of the input current flows into the amplifier with this ideal input impedance). Note that the biasing of the amplifier is controlled by the current mirror formed between M3 and M1 (drawn in a different way to ensure that the reader recognizes the topology when it is seen).

The input impedance of the amplifier in Fig. 21.11 is given by

$$R_{in} = \frac{1}{g_{m3}} || r_{o3} || r_{o4} \approx \frac{1}{g_{m3}} \tag{21.18}$$

The input current, i_{in}, is determined by

$$i_{in} = g_{m3} v_{in} = g_{m3} v_{gs3} = g_{m3} v_{gs1} \tag{21.19}$$

The current through M1, and thus M2, is given by

$$i_d = g_{m1} v_{in} = \frac{g_{m1}}{g_{m3}} \cdot i_{in} \tag{21.20}$$

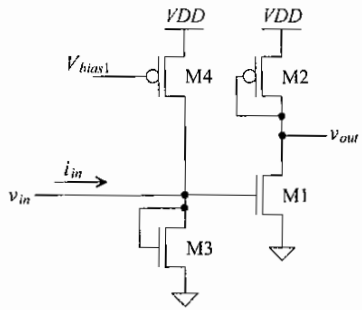


Figure 21.11 A transimpedance amplifier.

This equation can be rewritten as

$$\frac{i_d}{i_{in}} = \sqrt{\frac{\beta_1}{\beta_3}} = \sqrt{\frac{W_1 L_3}{W_3 L_1}} \tag{21.21}$$

That is, we can get a current gain simply by adjusting the size of M1 and M3. The current in M2 can then be mirrored out to a load. The transresistance gain of this configuration is given by

$$A_R = \frac{v_{out}}{i_{in}} = -\frac{i_d \frac{1}{g_{m2}}}{i_d \cdot \frac{g_{m3}}{g_{m1}}} = -\frac{g_{m1}}{g_{m2} g_{m3}} \tag{21.22}$$

Note that adding M3 and thus the resistor of value $\frac{1}{g_{m3}}$ to ground lowers the input time constant and increases the bandwidth of the amplifier.

Common-Source Amplifier with Source Degeneration

The amplifier in Fig. 21.3 is an example of a CS amplifier with source degeneration. The gain of the amplifier is reduced because the effective transconductance of the amplifier becomes

$$\frac{1}{g_{m,eff}} = \frac{1}{g_m} + R_s \tag{21.23}$$

as seen in Ex. 21.1. Adding the source resistance is useful, especially if $R_s >> 1/g_m$, to precisely set the transconductance of an amplifier. Note that we didn't, in Ex. 21.1, include the resistance looking into the drain of M1 when we wrote the "resistance in the drain." *Let's review, or again verify, how to determine the small-signal, low-frequency resistances looking into the drain and source of a MOSFET.* Using this information will greatly aid in our intuitive analysis of CMOS circuits.

Consider the test circuit shown in Fig. 21.12. Here the DC voltages ensure that the MOSFET is operating in the saturation region. We are concerned with the AC current that flows through v_t. This test voltage divided by i_d gives us the small-signal AC resistance looking into the drain of the MOSFET. In other words, this is the resistance connected to ground at the drain node.

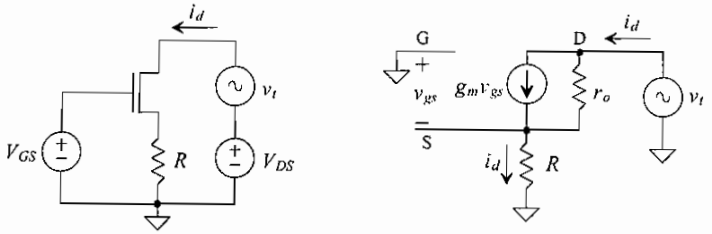


Figure 21.12 Circuit used to determine the resistance looking into the drain of a MOSFET.

Neglecting the body effect, we can write the test voltage as

$$v_t = (i_d - g_m v_{gs}) r_o + i_d R \tag{21.24}$$

where (noting $v_g = 0$ and $v_s = i_d R$)

$$v_{gs} = - i_d R \tag{21.25}$$

The resistance looking into the drain of the MOSFET is then given by

$$R_o = r_d = \frac{v_t}{i_d} = (1 + g_m R) r_o + R \approx (1 + g_m R) r_o \tag{21.26}$$

keeping in mind that $r_o \approx \frac{1}{\lambda I_D}$. For a cascoded MOSFET topology ($R = r_o$), this is approximately, as seen in Eq. (20.53), $g_m r_o^2$. For a *triple cascode*, three MOSFETs in series, ($R = g_m r_o^2$), the output resistance is

$$R_o \approx g_m^2 r_o^3 \tag{21.27}$$

noting that the output resistance goes up with each added MOSFET by the open-circuit gain, $g_m r_o$.

Example 21.4

Repeat Ex. 21.1 without neglecting the loading effects of the output resistances of M1 and M2.

The resistance looking into the drain of M2 is $\frac{1}{g_{m2}} || r_{o2}$, while the resistance looking into the drain of M1 is $r_{o1}(1 + g_{m1} R_s)$. The exact gain of the circuit is

$$A_v = \frac{v_{out}}{v_{in}} = - \frac{\frac{1}{g_{m2}} || r_{o2} || [r_{o1}(1 + g_{m1} R_s)] || R_L}{R_s + \frac{1}{g_{m1}}}$$

or when $r_{o1} || r_{o2} >> \frac{1}{g_{m2}}$, this reduces to

$$A_V = \frac{\frac{1}{g_{m2}} || R_L}{R_s + \frac{1}{g_{m1}}} \quad \blacksquare$$

The resistance in the source of the MOSFET can be found using the circuit shown in Fig. 21.13. Neglecting the body effect and output resistance of the MOSFET gives

$$v_{in} = v_{gs} + i_d R \tag{21.28}$$

Since $g_m v_{gs} = i_d$, this equation can be written as

$$v_{in} = \frac{i_d}{g_m} + i_d R = i_d \left[\frac{1}{g_m} + R \right] \tag{21.29}$$

We can look at this in a simplified manner; that is, $\frac{1}{g_m}$ is the resistance looking into the source of a MOSFET, while R is the resistance connected from the source of the MOSFET to ground.

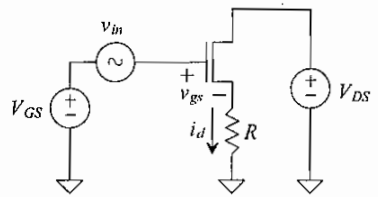


Figure 21.13 Circuit used to determine the resistance looking into the source of a MOSFET.

Noise Performance of the CS Amplifier with Gate-Drain Load

Figure 21.14 shows the CS amplifier of Fig. 21.1a with each MOSFET's model for noise power spectral density included (see Eq. [9.66]). Remembering that we only add the power from each noise source, we can write the output noise power spectral density as

$$V_{anoise}^2(f) = \left(\frac{1}{g_{m2}} \right)^2 \cdot \left(I_{M1}^2 + I_{M2}^2 \right) \tag{21.30}$$

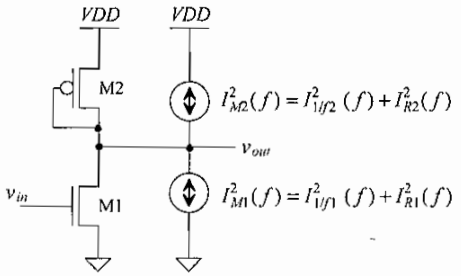


Figure 21.14 Noise modeling in a CS amplifier.

The input-referred noise power spectral density is then

$$V^2_{inoise}(f) = \frac{V^2_{onoise}(f)}{A_v^2} = \left(\frac{g_{m2}}{g_{m1}}\right)^2 \cdot \left(\frac{1}{g_{m2}}\right)^2 \cdot \left(I^2_{M1} + I^2_{M2}\right) \qquad (21.31)$$

Minimizing the input-referred noise is accomplished by making g_{m1} large. Using the expressions in Eq. (9.66), we also see that minimizing g_{m2} reduces the input-referred noise. This is equivalent to saying large A_v results in minimum input-referred noise.

21.1.2 The Source Follower (Common-Drain Amplifier)

Source-follower configurations using a gate-drain-connected load are shown in Fig. 21.15. The drain of the amplifying device, that is, the MOSFET *not* gate-drain-connected (the load), is common to both the input and the output, so this topology is often called a common-drain amplifier. Setting the bias current in this topology can be challenging. In the following we don't concern ourselves with this issue.

A source follower implemented in CMOS has an asymmetric drive capability; that is, the ability of the follower to source current is not equal to the ability to sink current for a given bias condition and AC input signal. The small-signal gain of the NMOS source follower shown in Fig. 21.15 is simply determined by a voltage divider between the resistance looking into the source of M2 ($1/g_{m2}$) with the resistance of the gate-drain-connected load, M1 ($1/g_{m1}$). The gain of the **source follower amplifier** is

$$A_v = \frac{v_{out}}{v_{in}} = \frac{\text{resistance connected to the source}}{\text{resistance connected to the source + resistance looking into the source}} \qquad (21.32)$$

or for the NMOS source follower in Fig. 21.15

$$v_{in} = \overbrace{v_{gs2}}^{i_d/g_{m2}} + \overbrace{v_{out}}^{i_d/g_{m1}} \text{ where } (v_{gs1} = v_{out}) \text{ and so } v_{out} = v_{in} \cdot \frac{\frac{1}{g_{m1}}}{\frac{1}{g_{m1}} + \frac{1}{g_{m2}}} \qquad (21.33)$$

The output resistance of the source follower shown in this figure is

$$R_{out} = \frac{1}{g_{m1}} || \frac{1}{g_{m2}} \qquad (21.34)$$

Note that M2 can only source current, while M1 can only sink current, and the gain of the source follower is always less than one.

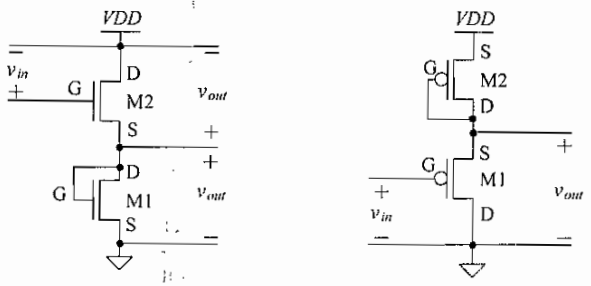


Figure 21.15 Source followers (common drain) using gate-drain loads.

21.1.3 Common Gate Amplifier

The common gate amplifier with gate-drain load is shown in Fig. 21.16. The input resistance of this amplifier is simply the resistance looking into the source of M1, or

$$R_{in} = \frac{1}{g_{m1}} \qquad (21.35)$$

The gain of this **common-gate amplifier** is

$$A_v = \frac{v_{out}}{v_{in}} = \frac{-i_d \cdot \frac{1}{g_{m2}}}{-v_{gs1}} = \frac{-i_d \cdot \frac{1}{g_{m2}}}{-i_d \cdot \frac{1}{g_{m1}}} = \frac{\text{resistance in the drain}}{\text{resistance in the source}} = \frac{\frac{1}{g_{m2}}}{\frac{1}{g_{m1}}} \qquad (21.36)$$

or noting the positive gain

$$A_v = \frac{g_{m1}}{g_{m2}} = \sqrt{\frac{KP_n}{KP_p} \cdot \frac{W_1 L_2}{W_2 L_1}} \qquad (21.37)$$

which is the same form as the gain for the CS amplifier with gate-drain load.

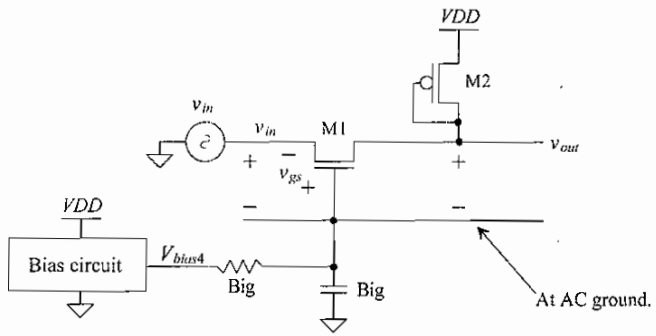


Figure 21.16 The common gate amplifier.

21.2 Current Source Loads

In the last section we talked about the gate-drain MOSFET part of a current mirror. In this section we talk about the current-source side of the mirror. The current source load provides an amplifier with the largest possible load resistance and thus the largest gain.

21.2.1 Common-Source Amplifier

Consider the CS amplifier with current source load shown in Fig. 21.17. The MOSFET M1 is the common-source component of the amplifier, while the MOSFET M2 is the current source load. The DC transfer characteristics are also seen in this figure. The amplifying portion of the curve occurs when the output is not close to *VDD* or ground (both M1 and M2 are saturated). The slope of the line when both transistors are saturated corresponds to the gain of the amplifier.

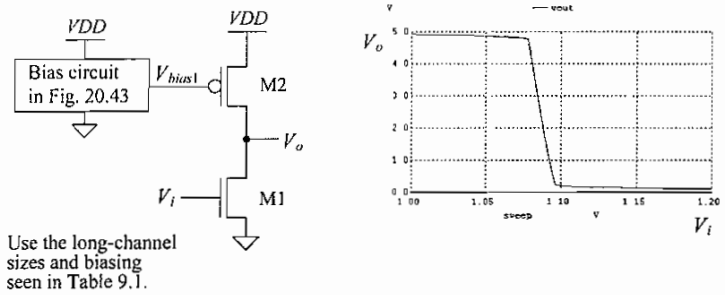


Figure 21.17 Common-source amplifier with current source load.

Class A Operation

We know from Table 9.1, and the bias circuit in Fig. 20.43, that M2 in Fig. 21.17 is biased to behave like a 20 μA current source. This means that if M1 shuts off, the maximum current we can supply to the output is 20 μA: this is important to understand. The CS amplifier in Fig. 21.17 is called a *class A* amplifier because, for proper operation, both MOSFETs are always conducting a current. In a *class B* amplifier, only one MOSFET is conducting a current at a given time. In *class AB* amplifiers (which we'll talk about later), both MOSFETs or a single MOSFET conduct a current at a given time.

Example 21.5

Estimate the maximum rate the load capacitor in Fig. 21.18 can be charged (estimate the slew rate across the load capacitor). Verify the estimate with SPICE.

The maximum current that the amplifier can source is 20 μA. The slew rate across the load is calculated using

$$I = C_L \cdot \frac{dV_{out}}{dt} \text{ or slew rate} = \frac{dV_{out}}{dt} = \frac{I}{C_L} \qquad (21.38)$$

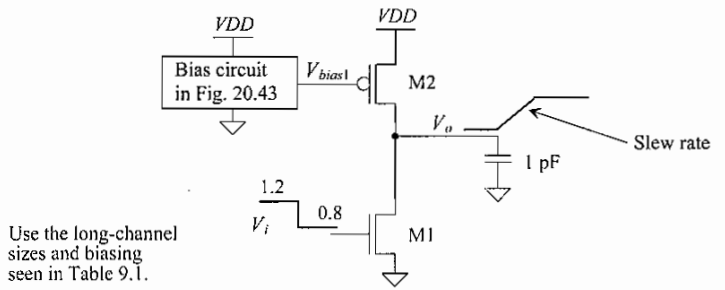


Figure 21.18 Slew rate limitations in a class A amplifier.

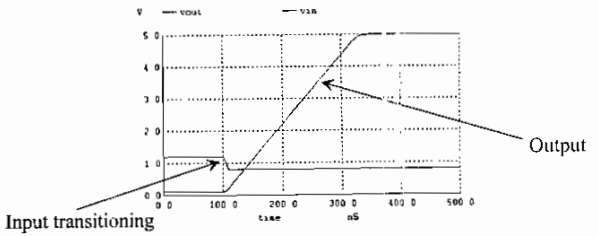


Figure 21.19 Verifying the results in Ex. 21.5

or

$$\frac{dV_{out}}{dt} = \frac{20\ \mu A}{1\ pF} = 20\ V/\mu s$$

For the output to transition 5 V requires 250 ns. ■

Small-Signal Gain

If we follow the intuitive method discussed in the last section, Eq. (21.1), the voltage gain of a common-source amplifier is the total parallel resistance at the drain of M1 divided by the resistance in the source of M1, or

$$A_v = \frac{v_o}{v_i} = -\frac{r_{o1} \| r_{o2}}{\frac{1}{g_{m1}}} = -g_{m1}(r_{o1} \| r_{o2}) = -\frac{g_{m1}}{g_{o1} + g_{o2}} \qquad (21.39)$$

where $r_{o1} = 1/g_{o1}$.

Open Circuit Gain

The gain of the amplifier can be increased by using a cascode current load in place of M2. The resistance looking into the drain of the cascode current source/load is much larger than the output resistance of M1. This situation is sometimes referred to as the *open circuit gain* of a common-source amplifier (see the specification in Tables 9.1 and 9.2). This specification indicates the maximum possible gain attainable with a single MOSFET (the MOSFET's load is its own output resistance r_o). The open circuit gain of M1 can be written as

$$\text{Open circuit gain} = -\frac{r_{o1}}{\frac{1}{g_{m1}}} = -g_{m1} r_{o1} = -\frac{\sqrt{2\beta_1 I_D}}{I_D \lambda_1} \qquad (21.40)$$

High-Impedance and Low-Impedance Nodes

In Fig. 21.17, the output node at the drains of M1 and M2 is termed a *high-impedance node*, that is, a node with only drain connections. The effective resistance at this node to ground is $r_{o1} \| r_{o2}$. A node connected to the source of a MOSFET or a MOSFET with its drain and gate connected is termed a *low-impedance node*. The small-signal resistance looking into the source of a MOSFET and the small-signal resistance of the gate-drain (diode)-connected MOSFET are both $1/g_m$. High-impedance nodes usually have a low-frequency pole associated with them that limits the speed of the amplifier.

Frequency Response

Figure 21.20 shows the AC frequency response circuit for the CS amplifier with current source load in Fig. 21.17. *VDD* in this figure is drawn as an AC ground. The source-gate capacitance of M2 isn't shown because both sides of it are at AC ground (DC voltages), so it doesn't affect the frequency behavior of the circuit. The pole associated with the input node of the circuit is

$$f_{in} = \frac{1}{2\pi(C_{gs1} + C_{gd1}(1 + |A_v|)) \cdot R_s} \tag{21.41}$$

while the pole associated with the output node of the circuit is located at

$$f_{out} = \frac{1}{2\pi\left(C_{dg2} + C_{gd1}\left(1 + \frac{1}{|A_v|}\right)\right) \cdot r_{o1} || r_{o2}} \tag{21.42}$$

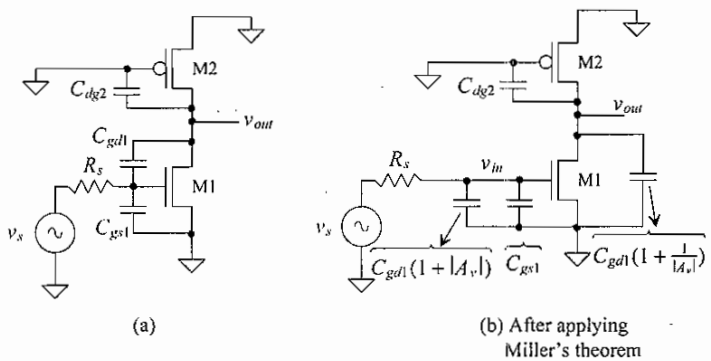
where $A_v = -g_{m1} \cdot r_{o1} || r_{o2}$ (see Eq. [21.39]). Using Eq. (21.42) to calculate the output pole, results in a *wrong estimate* when A_v is large. As we'll see in a moment, the fact that A_v decreases above f_{in} causes v_{in} to "flatten out" because the Miller capacitance, $C_{gd1}(1 + |A_v|)$, gets smaller. The effect of v_{in} ceasing to decrease causes the output pole to appear much larger than what is predicted by Eq. (21.42). This important phenomenon is called *pole splitting*.

The zero in the transfer function (see Eq. (21.15) is located at

$$f_z = \frac{g_{m1}}{2\pi C_{gd1}} \tag{21.43}$$

The transfer function of the amplifier is then estimated as

$$A_v(f) = -g_{m1} \cdot (r_{o1} || r_{o2}) \cdot \frac{\left(1 - j\frac{f}{f_z}\right)}{\left(1 + j\frac{f}{f_{in}}\right)\left(1 + j\frac{f}{f_{out}}\right)} \tag{21.44}$$



(a) (b) After applying Miller's theorem

Figure 21.20 Frequency response of the CS amplifier with current source load.

We can then write the magnitude as

$$|A_v(f)| = \frac{g_{m1} \cdot (r_{o1} || r_{o2}) \cdot \sqrt{1 + \left(\frac{f}{f_z}\right)^2}}{\left(\sqrt{1 + \left(\frac{f}{f_{in}}\right)^2}\right) \cdot \left(\sqrt{1 + \left(\frac{f}{f_{out}}\right)^2}\right)} \tag{21.45}$$

and the phase shift through the amplifier as

$$\angle A_v = 180 - \tan^{-1}\left(\frac{f}{f_z}\right) - \tan^{-1}\left(\frac{f}{f_{in}}\right) - \tan^{-1}\left(\frac{f}{f_{out}}\right) \tag{21.46}$$

Example 21.6

Determine the gain, magnitude, and phase shift of the amplifier shown in Fig. 21.21. Using an AC SPICE simulation, verify your hand calculations.

For the AC small-signal analysis, the big resistor appears like an open and the big capacitor a short. These components are added to ensure that M1 is biased to sink the current supplied by M2.

The low-frequency gain of the circuit is, from Eq. (21.39) and Table 9.1,

$$|A_v| = (150\ \mu A/V) \cdot (5M\Omega || 4M\Omega) = 333\ V/V \rightarrow 50\ dB$$

The input capacitance is

$$C_{in} = (C_{gs1} + C_{gd1}(1 + |A_v|)) = (23.3 + 2 \cdot 333)\,fF = 691\,fF\,! \tag{21.47}$$

The output capacitance is

$$C_{out} = \left(C_{dg2} + C_{gd1}\left(1 + \frac{1}{|A_v|}\right)\right) \approx 6 + 2\,fF = 8\,fF \tag{21.48}$$

Using Eqs. (21.41)–(21.43), we get f_{in} = 2.3 MHz, f_{out} = 9 MHz, and f_z = 11.9 GHz. Figure 21.22 shows the simulation results. As the discussion following Eq. (21.42) indicated, the estimate for f_{out} is wrong. To determine the correct value, let's discuss pole splitting (see Eq. [21.62]). ■

Use the long-channel sizes and biasing seen in Table 9.1.

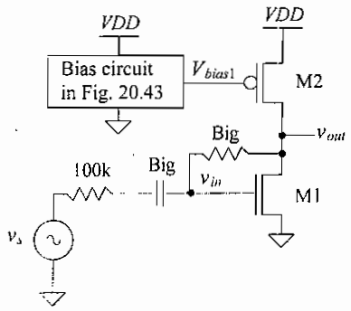


Figure 21.21 Amplifier used in Ex. 21.6.

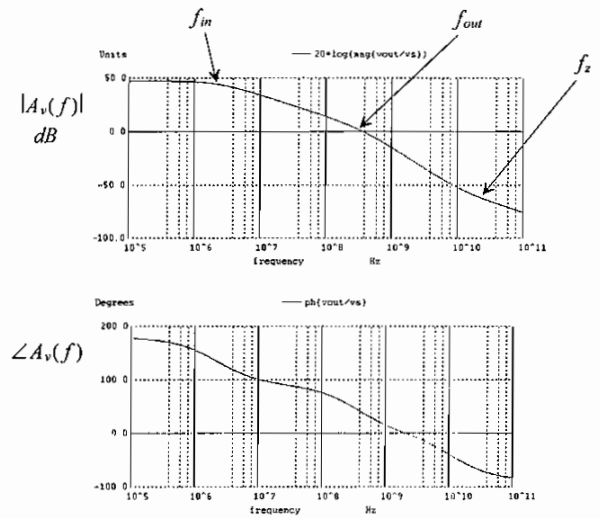


Figure 21.22 Frequency response of the amplifier in Fig. 21.21.

Pole Splitting

As discussed after Eq. (21.42), using Miller's theorem to calculate the location of the output pole when the gain of the amplifier is large results in error. Clearly, as just demonstrated in Ex. 21.6, the calculated value of f_{out} is 9 MHz while the SPICE simulation in Fig. 21.22 shows that f_{out} is approximately 200 MHz. The source of this discrepancy can be traced to the fact that above f_{in} the gain of the amplifier, A_v, decreases causing the effective input capacitance of the amplifier to decrease. As seen in Fig. 21.23, this causes the input voltage to "flatten out" instead of continuing to fall at –20 dB/decade. The result is the second, or output pole, appears to split from (leave) the first,

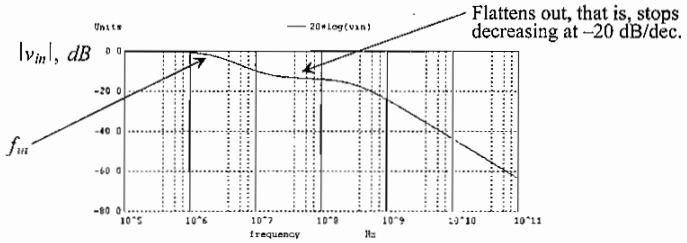


Figure 21.23 Showing how the amplifier's, in Fig. 21.21, input voltage flattens out as the gain decreases.

or input pole, and move to a higher frequency. Note that if we resimulate the amplifier in Fig. 21.21 with the source resistance set to zero ohms (so there isn't an input pole), pole splitting isn't present and Eq. (21.42) can be used to calculate the location of the output pole.

To characterize this effect, let's use the circuit in Fig. 21.7 with the addition of a source resistance, R_s, as seen in Fig. 21.24. In this figure, $R_o = r_{o1} \| r_{o2}$ and $C_o = C_{db2}$. From Eq. (21.14), we can write

$$\frac{v_{out}}{v_{in}} = -g_{m1} R_o \cdot \frac{1 - j\omega \frac{C_{gd1}}{g_{m1}}}{1 + j\omega (C_{gd1} + C_o) \cdot R_o} \tag{21.49}$$

At the gate of M1, we can write

$$\frac{v_{in} - v_s}{R_s} + \frac{v_{in}}{1/j\omega C_{gs1}} + \frac{v_{in} - v_{out}}{1/j\omega C_{gd1}} = 0 \tag{21.50}$$

or

$$v_{in} = \frac{\frac{v_s}{R_s} + v_{out} \cdot j\omega C_{gd1}}{\frac{1}{R_s} + j\omega C_{gs1} + j\omega C_{gd1}} \tag{21.51}$$

Substituting Eq. (21.51) into Eq. (21.49) gives

$$v_{out} = -g_{m1} R_o \cdot \frac{1 - j\omega \frac{C_{gd1}}{g_{m1}}}{1 + j\omega (C_{gd1} + C_o) \cdot R_o} \cdot \frac{\frac{v_s}{R_s} + v_{out} \cdot j\omega C_{gd1}}{\frac{1}{R_s} + j\omega C_{gs1} + j\omega C_{gd1}} \tag{21.52}$$

or, with $s = j\omega$,

$$\frac{v_{out}}{v_s} = \frac{-g_{m1} R_o \cdot \left(1 - s\frac{C_{gd1}}{g_{m1}}\right)}{s^2 [R_o R_s (C_{gd1} C_{gs1} + C_o C_{gs1} + C_o C_{gd1})] + s[(C_{gd1} + C_o) R_o + (C_{gs1} + C_{gd1}) R_s + C_{gd1} g_{m1} R_o R_s] + 1} \tag{21.53}$$

The zero is still, as indicated in Eqs. (21.15) and (21.43), located at

$$f_z = \frac{g_{m1}}{2\pi C_{gd1}} \tag{21.54}$$

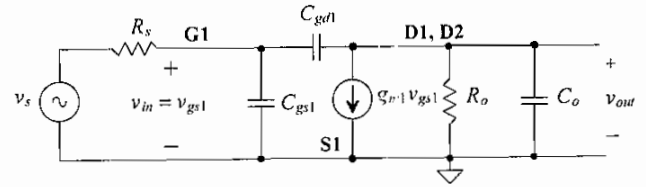


Figure 21.24 Circuit used to calculate the transfer function of the amplifier in Fig. 21.21.

At low frequencies the denominator is approximately (the coefficient of the s^2 term is small)

$$1 + j\omega \cdot [(C_{gd1} + C_o)R_o + (C_{gs1} + C_{gd1})R_s + C_{gd1}g_{m1}R_oR_s] \qquad (21.55)$$

and so the low-frequency pole, using $|A_v| = g_{m1}(r_{o1}||r_{o2}) = g_{m1}R_o$, $C_o = C_{dg2}$, is located at

$$f_1 \approx \frac{1}{2\pi[(C_{gd1} + C_{dg2}) \cdot r_{o1}||r_{o2} + (C_{gs1} + C_{gd1}(1 + |A_v|)) \cdot R_s]} \qquad (21.56)$$

If $C_{gd1}(1 + |A_v|)$ is much larger than the other capacitances,

$$f_1 \approx \frac{1}{2\pi C_{gd1}(1 + |A_v|) \cdot R_s} \qquad (21.57)$$

Notice the similarity to Eq. (21.41), where the pole is associated with the input of the amplifier. Further notice that if we had used this result in Ex. 21.6, the input pole would have changed very little. Also notice that if $R_s = 0$, then

$$f_1 \approx \frac{1}{2\pi(C_{gd1} + C_{dg2}) \cdot r_{o1}||r_{o2}} \qquad (21.58)$$

Notice the similarity to Eq. (21.42), where the pole is associated with the output of the amplifier. To determine the location of the second root of the denominator (the second pole location, f_2), let's factor Eq. (21.55) from the denominator of Eq. (21.53) or

$$[1 + s[(C_{gd1} + C_o)R_o + (C_{gs1} + C_{gd1})R_s + C_{gd1}g_{m1}R_oR_s]] \cdot \left(1 + \frac{s^2[R_oR_s(C_{gd1}C_{gs1} + C_oC_{gs1} + C_oC_{gd1})]}{s[(C_{gd1} + C_o)R_o + (C_{gs1} + C_{gd1})R_s + C_{gd1}g_{m1}R_oR_s] + 1}\right) \qquad (21.59)$$

where this equation is in the form

$$\left(1 + j \cdot \frac{f}{f_1}\right) \cdot \left(1 + j \cdot \frac{f}{f_2}\right) \qquad (21.60)$$

Note again that if $R_s = 0$, the second term in Eq. (21.59) goes to unity. Looking at the second term of Eq. (21.59), let's divide the numerator and denominator by sR_oR_s

$$1 + j \cdot \frac{2\pi f \cdot (C_{gd1}C_{gs1} + C_oC_{gs1} + C_oC_{gd1})}{(C_{gd1} + C_o)/R_s + (C_{gs1} + C_{gd1})/R_o + C_{gd1}g_{m1} + 1/sR_oR_s} \qquad (21.61)$$

In any practical MOSFET amplifier, $g_m >> 1/r_o$ (if not, then the magnitude of the open circuit gain, Eq. (21.40), is too small for the MOSFET to be of practical value as an amplifying device). Also, the R_s is assumed to be large (if not, then f_2 is not used), that is, the amplifier's transfer function has a single-pole response, where the pole is associated with the output of the amplifier as seen in Eq. (21.58). We can therefore write

$$f_2 \approx \frac{g_{m1}C_{gd1}}{2\pi \cdot (C_{gd1}C_{gs1} + C_oC_{gs1} + C_oC_{gd1})} \qquad (21.62)$$

Calculating the location of the second pole in Ex. 21.6 with this equation results in $f_2 =$ 240 MHz (which is very close to the simulated location).

Why the name pole splitting? If we increase the effective size of C_{gd1} by placing a capacitor, C_c, from the amplifier's input to its output (so that C_c is in parallel with C_{gd1}, that is, the effective value is $C_c + C_{gd1}$), then, as Eq. (21.56) shows, the low-frequency location of the pole, f_1, decreases. At the same time, increasing the effective size of C_{gd1} causes the location of the high-frequency pole f_2, as seen in Eq. (21.62), to increase. Thus the name *pole splitting*.

Pole Splitting Summary

Because of the ubiquity of pole splitting in CMOS amplifier design, let's summarize our discussion using a generic model, Fig. 21.25. In terms of the parameters seen in this figure the location of the zero is now

$$f_z = \frac{g_{m2}}{2\pi \cdot C_c} \qquad (21.63)$$

The first pole is located at

$$f_1 \approx \frac{1}{2\pi[(C_c + C_2) \cdot R_2 + (C_1 + C_c(1 + g_{m2}R_2)) \cdot R_1]} \qquad (21.64)$$

For large gain, $g_{m2}R_2$, we can simplify this to

$$f_1 \approx \frac{1}{2\pi g_{m2}R_2R_1C_c} \qquad (21.65)$$

The second pole is located at

$$f_2 \approx \frac{g_{m2}C_c}{2\pi \cdot (C_cC_1 + C_1C_2 + C_cC_2)} \qquad (21.66)$$

The transfer function is then written as

$$A_v(f) = \frac{v_{out}(f)}{v_s(f)} = g_{m1}R_1g_{m2}R_2 \cdot \frac{\left(1 - j \cdot \frac{f}{f_z}\right)}{\left(1 + j \cdot \frac{f}{f_1}\right) \cdot \left(1 + j \cdot \frac{f}{f_2}\right)} \qquad (21.67)$$

where $A_v = g_{m1}R_1g_{m2}R_2$ (the gain is now positive because of the defined direction of the current source in Fig. 21.25).

Again note that if C_c is increased, f_1 moves downwards and f_2 moves upwards (pole splitting). Also note that if the capacitor, C_c, is made large (quite common), then f_1 is much lower than the location of the other pole or the zero (the pole associated with f_1 is

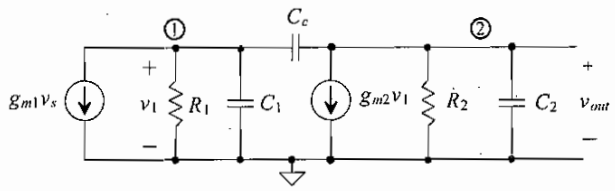


Figure 21.25 Generic model used to estimate bandwidth in a CMOS amplifier.

said to be the *dominant pole*). The amplifier's transfer function, with the help of Eq. (21.65), can be written as

$$A_v(f) = \frac{v_{out}(f)}{v_s(f)} \approx \frac{g_{m1}R_1 g_{m2}R_2}{\left(1 + j \cdot \frac{f}{f_1}\right)} = \frac{g_{m1}R_1 g_{m2}R_2}{(1 + j \cdot 2\pi f \cdot g_{m2}R_2R_1C_c)} \qquad (21.68)$$

This equation is in the form

$$A_v(f) = \frac{v_{out}(f)}{v_s(f)} = \frac{A_{DC}}{1 + j \cdot \frac{f}{f_{3dB}}} \qquad (21.69)$$

where

$$A_{DC} = g_{m1}R_1 g_{m2}R_2 \text{ and } f_{3dB} = \frac{1}{2\pi g_{m2}R_2R_1C_c} \qquad (21.70)$$

The magnitude and phase responses of a generic CMOS amplifier are seen in Fig. 21.26. Note that at frequencies much larger than the amplifier's 3 dB frequency, that is, $f/f_{3dB} >> 1$, Eq. (21.68) can be written as

$$A_v(f) = \left|\frac{v_{out}(f)}{v_s(f)}\right| \approx \frac{g_{m1}}{2\pi f \cdot C_c} \qquad (21.71)$$

To determine the frequency, f_{un}, where the transfer function is unity, we set Eq. (21.71) to one and solve to yield

$$f_{un} = \frac{g_{m1}}{2\pi C_c} \qquad (21.72)$$

We've derived some very useful equations, so let's apply them in some examples.

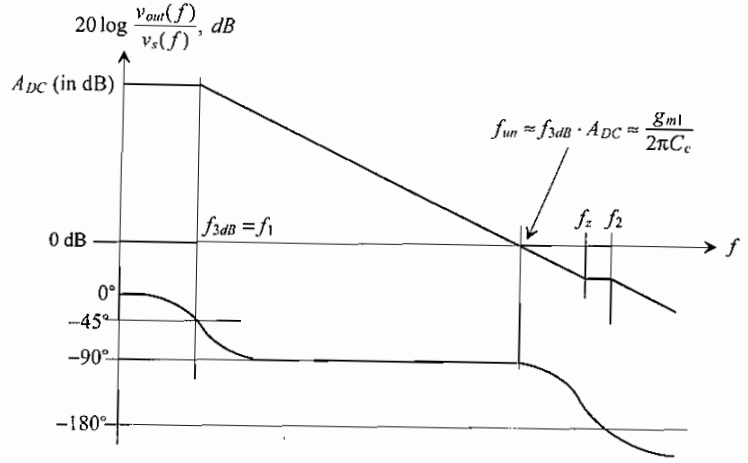


Figure 21.26 Magnitude and phase responses of a generic CMOS amplifier.

Example 21.7
Repeat Ex. 21.6 using the model in Fig. 21.25.

Using the model parameters in Fig. 21.25 and looking at Figs. 21.20a and 21.21, we can write (with the help of Table 9.1)

$$R_1 = R_s = 100k,\ C_1 = C_{gs1} = 23.3\,fF,\ C_c = C_{gd1} = 2\,fF,$$

$$C_2 = C_{db2} = 6\,fF,\ R_2 = r_{o1} \| r_{o2} = 2.22\ M\Omega$$

$$g_{m2} \text{ (in Fig. 21.25)} = g_{m1} \text{ (in Fig. 21.21)} = 150\ \mu A/V$$

$$g_{m1} \text{ (in Fig. 21.25)} = 1/R_s \text{ (in Fig. 21.21)} = 10\ \mu A/V$$

Using Eq. (21.67) or (21.70), we have $|A_v| = 333\ V/V \rightarrow 50\ dB$. Using Eq. (21.63), we get $f_z = 11.9\ GHz$; Eq. (21.65), we get $f_1 = 2.3\ MHz$; and finally using Eq. (21.66), we get $f_2 = 240\ MHz$. These calculations should be compared to the simulation results seen in Fig. 21.22. ■

Example 21.8
Repeat Ex. 21.7 if R_s is zero. Verify the hand calculations using SPICE.

Again we use the model seen in Fig. 21.25. From Ex. 21.7, $g_{m1}R_1$ is unity (and so R_s doesn't affect the gain). Further, the location of the zero doesn't change. Since R_s is zero ($R_1 = 0$ so the current source, $g_{m1}v_s$, and the resistor, R_1 in Fig. 21.25 are replaced with a voltage source, v_s), the location of f_2 is moved to an infinite frequency (as seen in Eq. [21.59]). Using Eq. (21.64), we get

$$f_1 \approx \frac{1}{2\pi(C_{gd1} + C_{db2}) \cdot r_{o1} \| r_{o2}} = \frac{1}{2\pi(8\,fF) \cdot 2.22M} = 9\ MHz$$

The simulation results are seen in Fig. 21.27. Note the small perturbation in the frequency response above f_1. The response is not perfectly decreasing at −20 dB/decade. This imperfection can be traced to the bias voltage, V_{bias1}, used on the gate of M2. It is not a perfect AC ground as was assumed in our hand calculations. By placing a capacitor on the gate of M2 to ground, the variation in V_{bias1} is reduced and the amplifier behaves as expected. This is an *important point*. It is often a good idea to "bypass" the bias voltages to AC ground to reduce their impedance in practical design. This also keeps circuits that are biased from the same bias circuit from interacting. ■

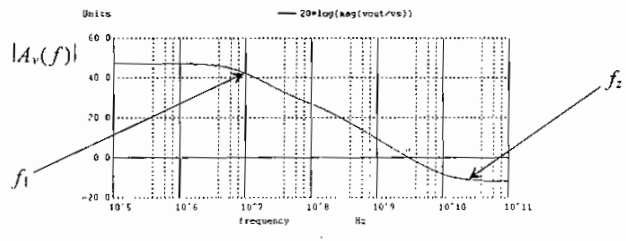


Figure 21.27 Simulation results for Ex. 21.8.

Example 21.9

Determine the frequency response of the amplifier seen in Fig. 21.28. Verify the hand calculations using simulations.

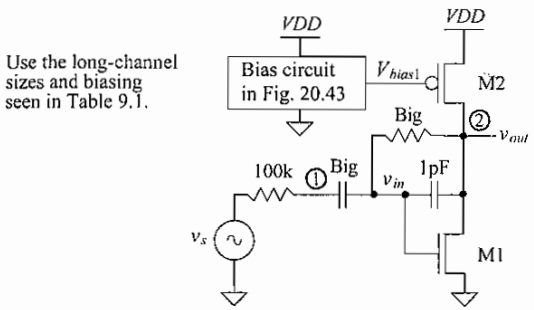


Figure 21.28 Amplifier used in Ex. 21.9.

The only thing that is different between this amplifier and the one in Ex. 21.7 is the value of C_c. In Fig. 21.28

$$C_c = 1\ pF + C_{gd1} \approx 1\ pF$$

We can then write, using Eq. (21.65)

$$f_1 \approx \frac{1}{2\pi \cdot (150\ \mu A/V) \cdot 100k \cdot 2.2MEG \cdot 1pF} = 4.8\ kHz$$

using Eq. (21.66)

$$f_2 \approx \frac{(150\ \mu A/V) \cdot 1p}{2\pi \cdot (1p \cdot 23.3f + 23.3f \cdot 6f + 1p \cdot 6f)} = 811\ MHz$$

At the risk of stating the obvious, these two poles are split apart by a great distance with the addition of the 1 pF capacitor. The zero, from Eq. (21.63), is located at

$$f_z = \frac{150\ \mu A/V}{2\pi \cdot 1p} = 23\ MHz$$

The unity-gain frequency is found using Eq. (21.72) with $R_s = 1/g_{m1}$.

$$f_{un} = \frac{1}{2\pi R_s C_c} = 1.59\ MHz$$

The simulation results are seen in Fig. 21.29. The value of the gain is 47 dB (we calculated 50 dB). The value of the first pole is approximately 6 kHz (we calculated 4.8 kHz). The small discrepancies are most likely the result of differences in the actual circuit parameters compared to the values we used in the formulas. ■

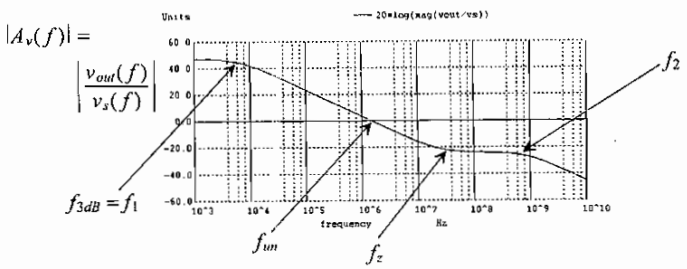


Figure 21.29 Frequency response of the amplifier in Fig. 21.28.

Example 21.10

Estimate the frequency response of the amplifier seen in Fig. 21.30. Verify the estimates with SPICE.

MA's gate and drain, at DC, are at the same potential. This biases MB to sink the right amount of current from MD.

Using the model in Fig. 21.25, we see that the source voltage, v_s, modulates the drain current in MA. We can therefore write

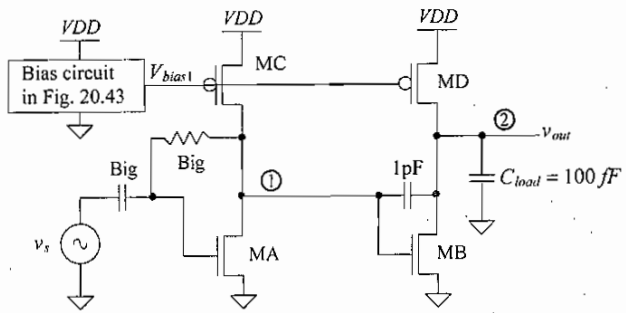
$$g_{mA} v_s = g_{m1} v_s \rightarrow g_{m1}\ (\text{in Fig. 21.25}) = g_{mA}\ (\text{in Fig. 21.30}) = 150\ \mu A/V$$

$$g_{m2}\ (\text{in Fig. 21.25}) = g_{mB}\ (\text{in Fig. 21.30}) = 150\ \mu A/V$$

$$R_1 = r_{oA} \| r_{oC} = \quad R_2 = r_{oB} \| r_{oD} = 2.22\ M\Omega$$

To determine the capacitance on nodes 1 and 2, we can use Fig. 21.31. At node 1

$$C_1 = C_{gdA} + C_{dgC} + C_{gsB} = 31.3\ fF$$



Use the long-channel sizes and biasing seen in Table 9.1.

Figure 21.30 Amplifier used in Ex. 21.10.

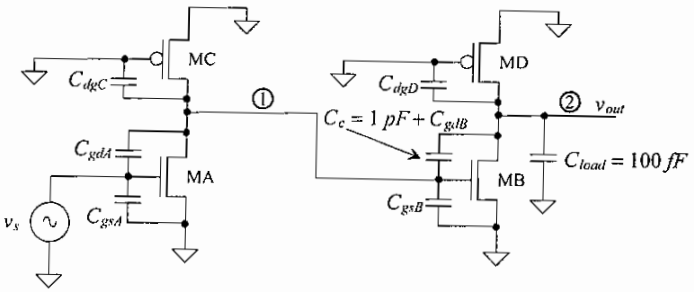


Figure 21.31 The capacitances for the amplifier in Fig. 21.30.

where C_{gdA} is the output Miller capacitance of MA ($C_{gdA} \approx C_{gdA}\left(1+\frac{1}{|A|}\right)$ when the gain of MA is large). At node 2

$$C_2 = C_{load} + C_{dgD} = 106\,fF$$

The value of C_c is

$$C_c = 1\,pF + C_{gdB} \approx 1\,pF \text{ (just the added capacitor)}$$

The gain of the topology is estimated as

$$A_{DC} = g_{m1}R_1 g_{m2}R_2 = (150\,\mu A/V)^2(2.22\,M\Omega)^2 = 110,889\,V/V \rightarrow 101\,dB$$

The poles and zero can be calculated using Eqs. (21.63), (21.65), and (21.66)

$$f_z = \frac{150\,\mu A/V}{2\pi \cdot 1.002\,pF} = 23\,MHz$$

$$f_1 \approx \frac{1}{2\pi \cdot (150\,\mu A/V)(2.22\,M\Omega)^2(1.002\,pF)} = 214\,Hz$$

$$f_2 \approx \frac{(150\,\mu A/V) \cdot 1.002}{2\pi \cdot (1.002 \cdot 0.0313 + 0.0313 \cdot 0.106 + 1.002 \cdot 0.106)\,pF} = 170\,MHz$$

The location of the unity gain frequency is calculated using Eq. (21.72) as 23 MHz. The simulation results are seen in Fig. 21.32. ■

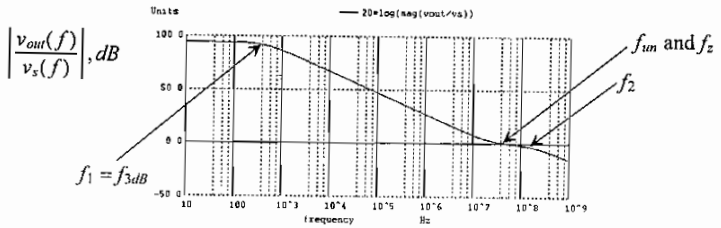


Figure 21.32 Frequency response of the amplifier in Fig. 21.30.

Canceling the RHP Zero

Notice in Ex. 21.10 that the unity-gain frequency (the frequency where the gain is one) is equal to the frequency of the zero. As mentioned earlier, the input of the amplifier feeds directly through C_c to the amplifier's output without the phase inversion. If the gain of the amplifier is larger than one (or even slightly less than one), then the lack of inversion, when the amplifier uses feedback, can result in an unstable amplifier. The amplifier's output signal feeds back and adds to its input signal without the inversion through the amplifier. To avoid this situation, we might add an amplifier in series with C_c that allows the output to feed back to the input through C_c but not vice versa, Fig. 21.33a. A simpler solution is to add a resistor in series with the capacitor, Fig. 21.33b, to attenuate the higher frequency signals (where the zero occurs) and push the zero out to a higher frequency. Adding the resistor moves the zero to a frequency (see Eq. [21.63])

$$f_z = \frac{1}{2\pi \cdot C_c \cdot \frac{1}{g_{m2}}} \xrightarrow{\text{With a resistor}} f_z = \frac{1}{2\pi \cdot C_c \cdot \left(\frac{1}{g_{m2}} - R_z\right)} \qquad (21.73)$$

If $R_z = 1/g_{m2}$, the zero disappears (is pushed to an infinite frequency). If $R_z > 1/g_{m2}$, the zero is pushed back into the LHP (phase shift is opposite from the poles). *Any practical design where pole splitting is used should include the zero-nulling resistor R_z.* Consider the following example.

Figure 21.33 Removing the zero in the amplifier's transfer function.

Example 21.11

Using a zero-nulling resistor, show that the location of the zero in the frequency response of the amplifier in Fig. 21.30 can be eliminated.

In this amplifier, $R_z = 1/g_{m2} = 6.5\,k\Omega$. By adding this resistor in series with the 1 pF capacitor, we get the frequency response seen in Fig. 21.34. Note that the zero is still seen but at a considerably higher frequency. ■

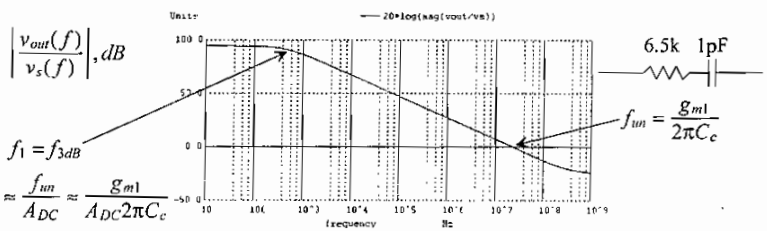


Figure 21.34 Pushing the zero in Fig. 21.32 to a higher frequency.

Noise Performance of the CS Amplifier with Current Source Load

To determine the noise performance of the CS amplifier with current source load, we can use the schematic in Fig. 21.14 with the gate of M2 at AC ground. The output noise power spectral density is then

$$V^2_{onoise}(f) = (r_{o1} || r_{o2})^2 \cdot \left(I^2_{M1} + I^2_{M2}\right) \tag{21.74}$$

The input-referred noise is then

$$V^2_{inoise}(f) = \frac{V^2_{onoise}(f)}{A^2_v} = \frac{1}{g^2_{m1}} \cdot \left(I^2_{M1} + I^2_{M2}\right) \tag{21.75}$$

showing once again that to minimize the input-referred noise we need to make the transconductance of M1 large (or, in other words, make A_v large).

21.2.2 The Cascode Amplifier

Consider a CS amplifier with current source load as seen Fig. 21.17 but implemented using the short-channel CMOS process. Using the parameters in Table 9.2, the gain of the topology, Eq. (21.39), is only 16.7 (it was 333 using the long-channel process). To boost the gain of the single-stage amplifier and to eliminate the Miller effect, consider the cascode amplifier seen in Fig. 21.35. The resistance looking into the drain of M3 is $g_{mp} \cdot r^2_{op}$ (see Table 20.1) with a value of 16.6 MΩ. The resistance looking into the drain of M2 is 4.2 MΩ (again from Table 20.1). The gain of the cascode amplifier is the resistance in the drain divided by the resistance in the source of the amplifying device (M1) or

$$\frac{v_{out}}{v_{in}} = -\frac{g_{mn}r^2_{on} || g_{mp}r^2_{op}}{1/g_{mn}} = -g_{mn} \cdot R_{ocas} \tag{21.76}$$

Using the bias circuit from Fig. 20.47 and the sizes in Table 9.2 (and thus the small-signal parameters in this table).

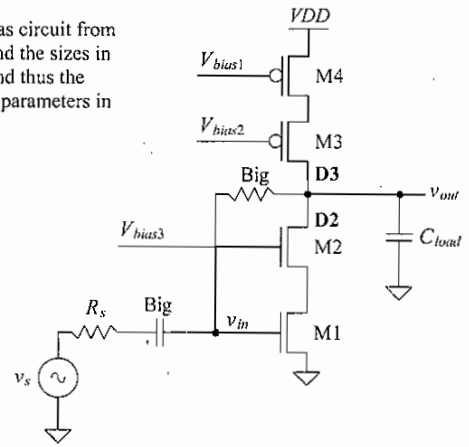


Figure 21.35 A cascode amplifier.

where

$$R_{ocas} = g_{mn}r^2_{on} || g_{mp}r^2_{op} \tag{21.77}$$

Using the parameters from Table 9.2, the gain of the cascode amplifier in Fig. 21.35 is, roughly, 500.

Frequency Response

The resistance looking into the source of M2 is $1/g_{m2}$ (= $1/g_{mn}$). The resistance in the drain of M1 is then approximately $1/g_{m2}$. The gain from the gate of M1 (the amplifier's input) to the drain of M1 is then

$$A_{v1} = \frac{v_{d1}}{v_{in}} = \frac{-v_{gs2}}{v_{gs1}} = \frac{-i_d/g_{m2}}{i_d/g_{m1}} = -\frac{1/g_{m2}}{1/g_{m1}} = -\frac{g_{m1}}{g_{m2}} = -1 \tag{21.78}$$

If we calculate the Miller capacitance, Eq. (21.5), we see that the low gain of M1 eliminates the large loading (from the Miller capacitance) on the input of the amplifier. For this reason, the cascode connection is often used in high-frequency design. Consider the following.

Example 21.12

Estimate the frequency response of the cascode amplifier in Fig. 21.35 if R_s is 100k and $C_L = 100\,fF$. Verify the hand calculations with simulations.

The input time constant is calculated as

$$\tau_{in} = R_s \cdot (C_{gs1} + (1 + |A_{v1}|) \cdot C_{gd1})$$

Using the values from Table 9.2

$$\tau_{in} = 100k \cdot (4.17\,fF + 2 \cdot 1.56\,fF) = 729\,ps$$

The pole associated with this input time constant is then

$$f_{in} = \frac{1}{2\pi\tau_{in}} = 218\,MHz$$

Since the load capacitance is large, $C_{load} >> C_{gd2} + C_{dg3}$, we can write

$$\tau_{out} = R_{ocas} \cdot (C_{load} + C_{gd2} + C_{dg3}) \approx g_{mn}r^2_{on} || g_{mp}r^2_{op} \cdot C_{load} = 335\,ns$$

and so the pole associated with the output node is located at

$$f_{out} = \frac{1}{2\pi\tau_{out}} = 475\,kHz$$

Again, the gain of the topology is 500 (54 dB). The simulation results are seen in Fig. 21.36. The magnitude response is

$$\left|\frac{v_{out}}{v_s}\right| = \frac{500}{\left(1 + j \cdot \frac{f}{475kHz}\right)\left(1 + j \cdot \frac{f}{218\,MHz}\right)}$$

and the phase response (in degrees) is

$$\angle\frac{v_{out}}{v_{in}} = 180 - \tan^{-1}\frac{f}{475\,kHz} - \tan^{-1}\frac{f}{218\,MHz}$$ ■

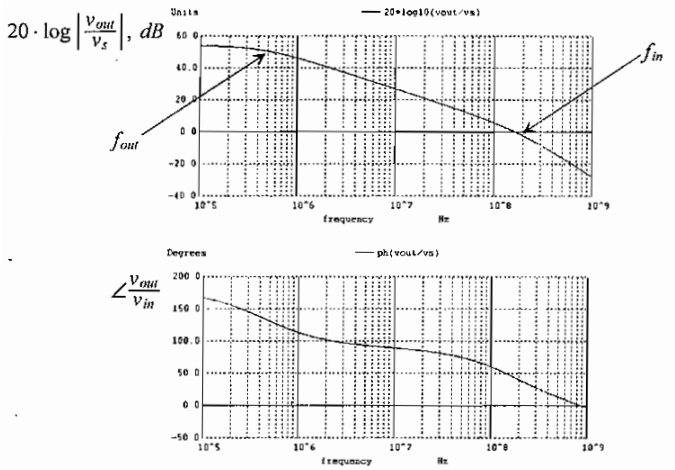


Figure 21.36 The simulation results for Ex. 21.12.

Class A Operation

The cascode amplifier in Fig. 21.35 is another example of a class A amplifier. When the input of the amplifier goes sufficiently negative, M1 shuts off. M3 and M4 are a current source and so the maximum rate that the load capacitance can be charged is given by Eq. (21.38). For the amplifier in Fig. 21.35, this is 10 μA/100 *fF* or 100 mV/ns.

Noise Performance of the Cascode Amplifier

The cascode's output noise power spectral density is given by

$$V^2_{onoise}(f) = \left(g_{mn}r^2_{on} \| g_{mp}r^2_{op}\right)^2 \left(I^2_{M1} + I^2_{M2} + I^2_{M3} + I^2_{M4}\right) \quad (21.79)$$

The input-referred noise power is then ($g_{m1} = g_{mn}$)

$$V^2_{inoise}(f) = \frac{V^2_{onoise}(f)}{A^2_v} = \frac{\left(I^2_{M1} + I^2_{M2} + I^2_{M3} + I^2_{M4}\right)}{(g_{m1})^2} \quad (21.80)$$

Again, maximizing the transconductance of the amplifying device (equivalent to saying maximizing the gain of the amplifier) reduces the input-referred noise.

Operation as a Transimpedance Amplifier

Figure 21.37 shows a transimpedance amplifier (current input and voltage output). Note that the AC voltages between the gates and sources of M1 and M4 are zero (both the gates and the sources are at AC ground). The gate-source voltage of M2 is

$$i_{in} = g_{m2}v_{gs2} = g_{m2}(-v_{ds1}) \quad (21.81)$$

See Table 9.2 and Fig. 20.47

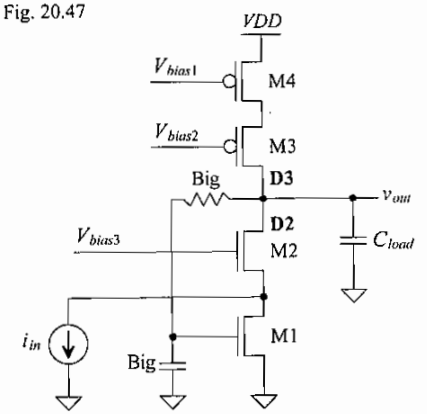


Figure 21.37 A transimpedance amplifier.

since the gate of M2 is at AC ground. The drain current that flows through M2 is i_{in}. The output voltage is

$$v_{out} = \left(g_{mn}r^2_{on} \| g_{mp}r^2_{op}\right) \cdot (-i_{in}) \quad (21.82)$$

or the gain is

$$\frac{v_{out}}{i_{in}} = -g_{mn}r^2_{on} \| g_{mp}r^2_{op} = -R_{ocas} \quad (21.83)$$

The input resistance of the transimpedance amplifier is

$$R_{in} = \frac{1}{g_{m2}} = \frac{1}{g_{mn}} \quad (21.84)$$

The ideal input resistance of a transimpedance amplifier (or any current input amplifier) is zero ohms.

21.2.3 The Common-Gate Amplifier

M2 in Fig. 21.37 is an example of a common-gate (CG) amplifier. We can redraw this circuit, as seen in Fig. 21.38 with a voltage source input, to show how the gate of the amplifying device, M2, is common to both the input and the output of the amplifier. Though the gate of M2 is at a DC voltage of V_{bias3}, we think of it as being at AC ground. M1 is simply an ordinary current source while M3 and M4 are a cascode current source load. The input resistance of this amplifier is given by Eq. (21.84), and the output resistance is given by Eq. (21.77). The voltage gain can be calculated by first writing

$$v_{in} = -v_{gs2} = -i_d/g_{m2} = i_{in}/g_{mn} \quad (21.85)$$

See Table 9.2 and Fig. 20.47

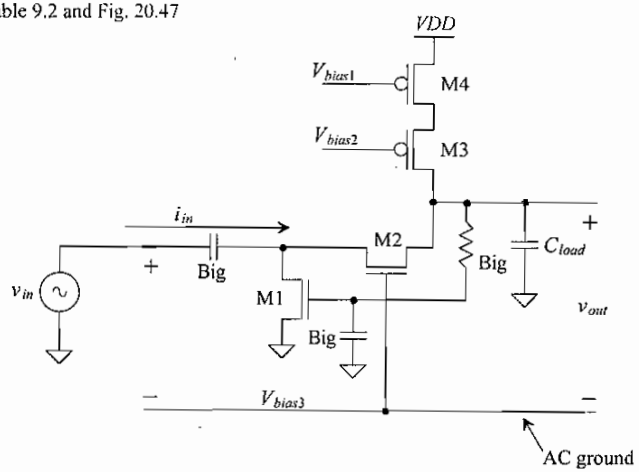


Figure 21.38 A common-gate amplifier.

where the drain current of M2 is $-i_{in}$. The output voltage is given by Eq. (21.82) and so the gain is

$$A_v = \frac{v_{out}}{v_{in}} = \frac{-i_d\left(g_{mn}r_{on}^2 \| g_{mp}r_{op}^2\right)}{-i_d/g_{m2}} = g_{mn} \cdot R_{ocas} \qquad (21.86)$$

21.2.4 The Source Follower (Common-Drain Amplifier)

The source follower (SF) with current source load is seen in Fig. 21.39. Looking at the NMOS SF, we can write, for AC small signals,

$$v_{in} = v_{gs2} + v_{out} \qquad (21.87)$$

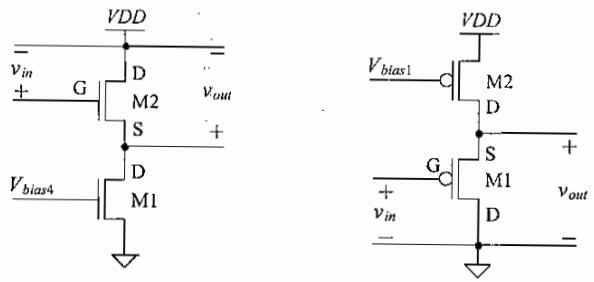


Figure 21.39 Source followers (common drain) using current source loads.

Knowing the resistance looking into the drain of M1 is r_o and the current that flows in M1 is i_d, we can write

$$v_{out} = i_d \cdot r_o \text{ and } v_{gs2} = \frac{i_d}{g_{m2}} \qquad (21.88)$$

Solving for the gain, we get

$$A_v = \frac{v_{out}}{v_{in}} = \frac{r_o}{r_o + 1/g_{m2}} = \frac{g_{m2}r_o}{g_{m2}r_o + 1} \qquad (21.89)$$

If we use a current source for the load with a very large small-signal output resistance, the gain goes to one. (Limitations concerning the body effect will be discussed in a moment.) Looking at Eq. (21.89), we see that the gain is a voltage divider between the resistance looking into the source of a MOSFET ($1/g_m$) and the output resistance of the current source (here using the simple, single MOSFET r_o) as seen in Eq. (21.32).

Note that the maximum current the NMOS SF can sink is limited by the size of the current flowing in the current source load M1. The SF is a class A amplifier where, when discharging a load capacitance, the current through M1 limits the rate at which the capacitance can discharge (as indicated in Eq. [21.38]). M2 only sources current and M1 only sinks current. When the SF is sourcing a current, M2 provides the current to both the amplifier's output and to M1. When the SF is sinking a current, the current in M2 decreases while the current in M1 is constant (resulting in a net sinking of current on the amplifier's output). Finally, note that the maximum output voltage of an NMOS SF occurs when the gate of M2 goes to *VDD*. The maximum output voltage then goes to $VDD-V_{GS2}$. The minimum output voltage is set by the minimum voltage across the current source load (to keep the MOSFETs in the saturation region).

Body Effect and Gain

We might think, after looking at Eq. (21.89), that by using a cascode current source load for the source follower we can get an AC small-signal gain very close to unity. However, the body effect ultimately limits the gain of the amplifier. Figure 21.40 shows an SF with an ideal current source load. If we use the AC small-signal model seen in the figure where the output resistance, r_o, is infinite, we see that the AC output voltage, v_{out}, equals the AC source to bulk potential v_{sb}. Further, we can write

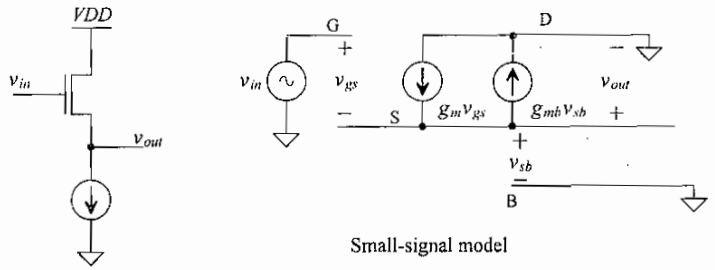


Figure 21.40 SF gain with body effect

$$v_{gs} = v_{in} - v_{out} \tag{21.90}$$

and, summing the currents at the output node,

$$g_m v_{gs} = g_{mb} v_{sb} = g_{mb} v_{out} \tag{21.91}$$

Solving for the gain of the SF with body effect and knowing $g_{mb} = \eta \cdot g_m$ (Eq. (9.28)), we get

$$\frac{v_{out}}{v_{in}} = \frac{g_m}{g_m + g_{mb}} = \frac{1}{1+\eta} \tag{21.92}$$

If $\eta = 0.25$, the gain is 0.8. The obvious way of eliminating the body effect (and thus move the gain closer to one) is to put the common-drain device in its own well, Fig. 21.41.

See Table 9.2 and Fig. 20.47

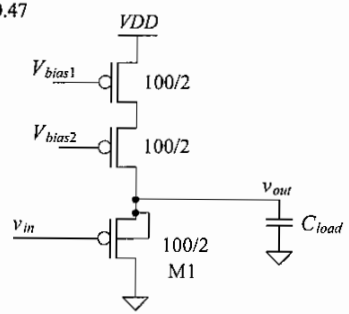


Figure 21.41 PMOS SF without body effect.

Level Shifting

One of the important uses of an SF is to provide a DC level shift to an input voltage. For example, M1 in Fig. 21.41 will remain in saturation as long as

$$v_{SD} = v_{OUT} \geq v_{SG} - V_{THP} = \overbrace{v_{OUT} - v_{IN}}^{v_{SG}} - V_{THP} \tag{21.93}$$

or

$$v_{IN} \geq -V_{THP} \tag{21.94}$$

As long as the input voltage is greater than $-V_{THP}$, M1 operates in the saturation region. This result is practically important. Note the output and input of the SF in Fig. 21.41 are related by

$$v_{OUT} = v_{IN} + V_{SG} \tag{21.95}$$

Using the parameters in Table 9.2 where *VDD* is 1 *V*, V_{SG} = 350 mV, and V_{THP} = 250 mV, the input voltage is shifted upwards, on the output of the amplifier, by 350 mV. The input signal can go down to –250 mV before M1 triodes. The minimum voltage across the current source, as seen in Fig. 20.48, is approximately 150 mV. This means that the range of the input signal is from *VDD* – 150 mV – 350 mV (= 600 mV = $V_{in,max}$) down to – 250 mV. Simulation results are seen in Fig. 21.42.

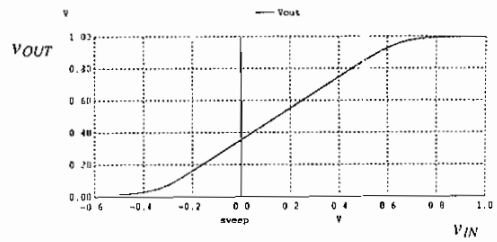


Figure 21.42 How the SF in Fig. 21.41 can shift negative input voltages upwards. This circuit is very useful when input signals are centered around ground.

Input Capacitance

Consider the partial schematic of an SF seen in Fig. 21.43. The capacitance on the input can be determined by looking at how much charge is supplied by the input source for an input voltage change Δv_{IN}

$$Q_{IN} = \Delta v_{IN} \cdot C_{dg} + (\Delta v_{IN} - A_v \cdot \Delta v_{IN}) \cdot C_{sg} \tag{21.96}$$

where $A_v \leq 1$. The input capacitance of an SF is then

$$C_{IN} = \frac{Q_{IN}}{\Delta v_{IN}} = C_{dg} + C_{sg}(1 - A_v) \approx C_{dg} \tag{21.97}$$

As the gain, A_v, of the SF approaches one, the input capacitance approaches the drain-gate capacitance of the MOSFET. In other words the source-gate capacitance doesn't affect the input capacitance (unlike the CS amplifier, which can have a very large input capacitance due to the Miller effect; see Ex. 21.6). Intuitively we can understand why C_{sg} doesn't affect the input capacitance by realizing that the displacement current through it is zero when the gate and source potentials move at the same rate. When the input voltage change and the output voltage changes are equal, the current through C_{sg} is zero. The SF is often used on the input of amplifiers that must have low input capacitance (like in charge-amplification applications). Unfortunately, the noise performance of the SF, because it doesn't have gain, is poorer than the CS or CG amplifiers.

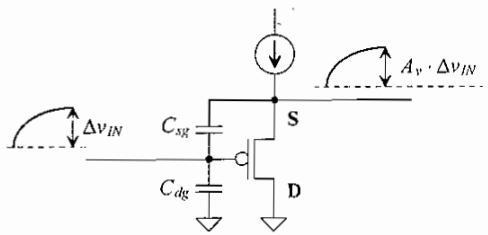


Figure 21.43 Input capacitance of a SF.

Noise Performance of the SF Amplifier

The SF amplifier with MOSFET noise sources is seen in Fig. 21.44. The noise current from M3 flows through M2 to the output node. The resistance on the output node (the output resistance of the SF) is the resistance looking into the source of M1 in parallel with the resistance looking into the drain of M2 or

$$R_{oSF} = \frac{1}{g_{m1}} || R_o \approx \frac{1}{g_{m1}} \tag{21.98}$$

The output noise power spectral density is

$$V^2_{onoise}(f) = \frac{I^2_{M1}(f) + I^2_{M2}(f) + I^2_{M3}(f)}{g^2_{m1}} \tag{21.99}$$

Noting that the SF's gain is close to one, the input-referred noise is nearly equal to the output noise PSD. Again, by using a large value of g_{m1}, we can minimize both the input-referred and output noise.

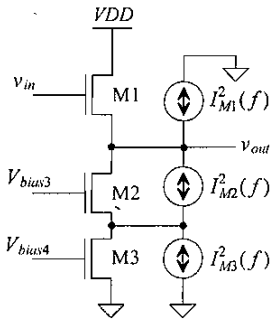


Figure 21.44 Noise model of the source follower amplifier.

Frequency Behavior

Using Eq. (21.97), we see that if we drove the SF with an input whose source resistance was R_s, then we would have a pole associated with the input at

$$f_{in} = \frac{1}{2\pi \cdot R_s C_{dg}} \tag{21.100}$$

a very high frequency. On the output of the SF driving a capacitive load, we can write, using Eq. (21.98),

$$f_{out} = \frac{1}{2\pi \cdot R_{oSF} C_{load}} = \frac{g_{m1}}{2\pi C_{load}} \tag{21.101}$$

However, both Eqs. (21.100) and (21.101) assume that the transconductance doesn't vary with frequency. At very high frequencies, the input and output of the SF are shorted together through the gate-source capacitance. To calculate the variation of the transconductance with frequency, consider the circuit seen in Fig. 21.45. We can write

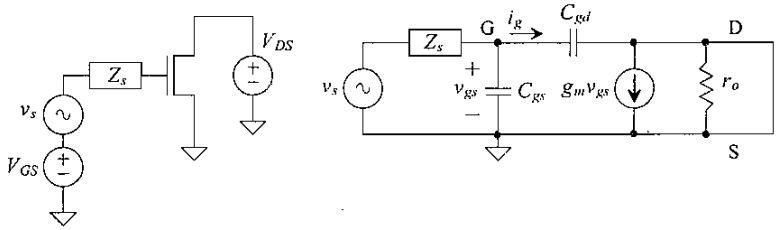


Figure 21.45 Determining the variation of the transconductance with frequency.

$$v_{gs}(f) = v_s \cdot \frac{\frac{1}{j\omega \cdot (C_{gs}+C_{gd})}}{\frac{1}{j\omega \cdot (C_{gs}+C_{gd})} + Z_s} = \frac{v_s}{1 + Z_s \cdot j\omega \cdot (C_{gs} + C_{gd})} \tag{21.102}$$

and

$$i_d(f) = g_m \cdot v_{gs}(f) = \frac{g_m v_s}{1 + Z_s \cdot j\omega \cdot (C_{gs} + C_{gd})} \tag{21.103}$$

The effective transconductance as a function of frequency is then

$$g_m(f) = \frac{g_m}{1 + j\omega \cdot Z_s(C_{gs} + C_{gd})} \tag{21.104}$$

The impedance looking into the source of a MOSFET as a function of frequency is then

$$R_{\text{into source}} = \frac{1}{g_m(f)} = \frac{1}{g_m} \cdot (1 + j\omega \cdot Z_s(C_{gs} + C_{gd})) \tag{21.105}$$

We note that at low frequencies the impedance is resistive and has a value of $1/g_m$. However, at higher frequencies and with a resistive source impedance, that is, $Z_s = R_s$, the impedance looking into the source appears to be the series connection of a resistor with a value of $1/g_m$ and an inductor of value

$$L_s = \frac{R_s(C_{gs} + C_{gd})}{g_m} \tag{21.106}$$

An SF driving a capacitive load can exhibit ringing because of the effective RLC circuit formed by its output impedance driving a load capacitance. If the impedance driving an SF is inductive (as we would have in the cascade of two SFs), the output impedance can become negative. For example, if $Z_s = j\omega L$, then substituting into Eq. (21.105), we get

$$R_{\text{into source}} = \frac{1}{g_m} - \omega^2 \cdot L \cdot (C_{gs} + C_{gd}) \tag{21.107}$$

A battery or voltage source is an example of a circuit with a negative resistance (any source of power has a negative resistance). At higher frequencies the resistance looking into the source of the MOSFET becomes negative. This means that the circuit will oscillate or simply have a poor step response (the voltage across the capacitive load will ring). Often, to implement a microwave frequency oscillator using a MOSFET, an inductor is added from the gate of the MOSFET to AC ground to create the negative resistance.

SF as an Output Buffer

One of the common uses of an SF is as an output buffer. Connecting a resistive load to a high-impedance node kills the gain of a single-stage CMOS amplifier. A buffer amplifier is often inserted between the high-impedance node and the load resistance to keep the gain high. Consider the following example.

Example 21.13

Suppose that the cascode amplifier in Fig. 21.35, with R_s = 100k, must drive a 1 pF capacitor in parallel with a 10 kΩ resistor. Design an SF buffer to ensure that the 10 kΩ resistor doesn't kill the gain of the amplifier. Estimate the frequency response of the amplifier. Verify your design with SPICE simulations.

Looking at Eq. (21.76) and the associated discussion, we see that the output resistance of the cascode amplifier is in the megaohms region. Connecting a 10 kΩ load resistor on the amplifier's output would cause the cascode gain to drop from 500 to less than 1. The DC output voltage of the cascode amplifier in Fig. 21.35 is the V_{GS} of M1 (which, from Table 9.2 is 350 mV). If we use an NMOS SF, this DC voltage isn't enough to ensure that all of the MOSFETs in the SF operate in the saturation region. Looking at the NMOS SF in Fig. 21.39, a voltage of 350 mV on the gate of M2 would be just enough to turn it on but leave little voltage to drop across the current source M1. We'll use a PMOS source follower in this design. The schematic of the design is seen in Fig. 21.46. The current source (M5 and M6) in the SF must drive 1 V across the resistor of 10 kΩ (100 μA), so we've bumped up their size. Knowing, from Table 9.2, that when using the bias circuit in Fig. 20.47 a 100/2 PMOS conducts 10 μA of current, we increased the widths of M5–M7 so that they conduct 125 μA of current with the same bias voltages (the current source, M5/M6, sources 125 μA of current).

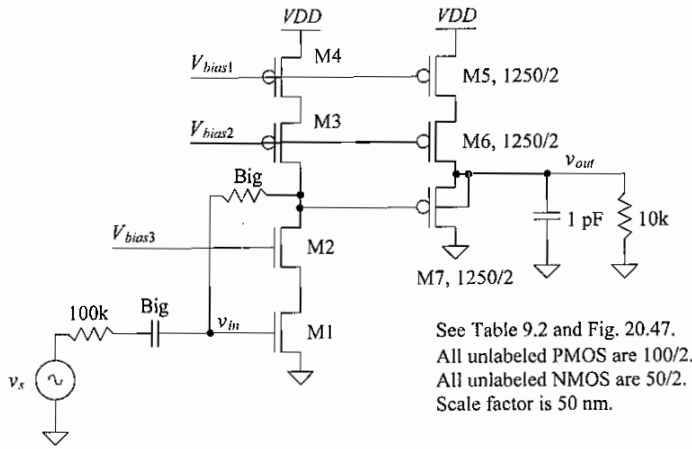


Figure 21.46 A cascode amplifier with SF output buffer.

The location of the input pole, f_{in}, is still (from Ex. 21.12) located at 218 MHz. The SF won't have much affect on the overall frequency response (because of its low input capacitance and small output resistance). However, now the cascode amplifier's load capacitance changes. Using the result in Eq. (21.97) and the data from Table 9.2, we can estimate the capacitance on the output of the cascode amplifier as the gate-drain capacitance of M7 ($3.7\,fF \cdot 12.5 \approx 50\,fF$). The factor of 12.5 comes from the increase in the PMOS's width by 12.5. Again, using the results from Ex. 21.12, the output pole, f_{out}, (output of the cascode amplifier) is at approximately 1 MHz (double what it was in Ex. 21.12).

The gain of the cascode amplifier is approximately 500. The transconductance of the wider PMOS devices is estimated using (see Eq. [9.22])

$$g_{m,wide} = \sqrt{W} \cdot g_{m,Table\ 9.2} = 530\ \mu A/V \qquad (21.108)$$

The gain of the SF driving a resistive load, see Eq. (21.89), is

$$A_{v,SF} = \frac{R_L}{R_L + 1/g_{m,wide}} = \frac{10k}{10k + 1.88k} = 0.84 \qquad (21.109)$$

The overall gain from the input of the cascode amplifier to the output of the SF is then estimated as $500 \cdot 0.84 = 421 \rightarrow 52.5\ dB$. The simulation results are seen in Fig. 21.47. Note that the bandwidth of this amplifier can be increased by reducing the width of M7. This reduces the input capacitance of the SF output amplifier and thus pushes the pole location out to a higher frequency. ■

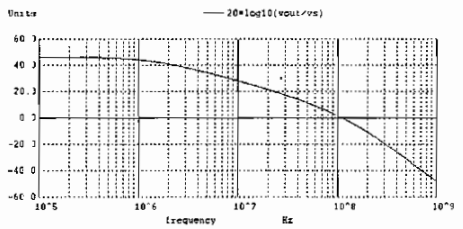


Figure 21.47 Simulating the operation of the amplifier in Fig. 21.46.

A Class AB Output Buffer Using SFs

In order to drive the 10k resistor in the previous example, we had to increase the size of the current source used in the SF. The practical problem with this approach is that as the load resistor gets small, the current source must source a significant current. The current source is sourcing this current all of the time (class A operation) either to the load or, if the load capacitance is being discharged, to the common-drain MOSFET (M7 in Fig. 21.46). What we need, for better drive capability, is for one side of the output buffer to shut off if the other side is supplying a large amount of current. For example, if, in Fig. 21.46, M7 starts to pull a large amount of current from the load, we would want M5 and M6 to turn off so that power wasn't wasted (class AB operation).

A class AB output buffer using SFs is seen in Fig. 21.48. The output buffer is comprised of M1–M4. M5 and M6 form a common-source amplifier with a current source load. When the circuit is in steady state, the AC input, v_{in}, is zero and the gate of M6 is at a DC potential of V_{bias1}. The current through M3 and M4 is mirrored by M1 and M2. This sets the DC current in M1/M2 to a known value (important). As v_{in} goes up, M6 shuts off. However, the current in M5 is constant, so the gates of M3 and M4 move towards ground. This turns M1 on and shuts M2 off (thus the class AB action). Similarly if v_{in} decreases, M6 turns on and pulls the gates of M3 and M4 up. The result is that M2 turns on and M1 shuts off.

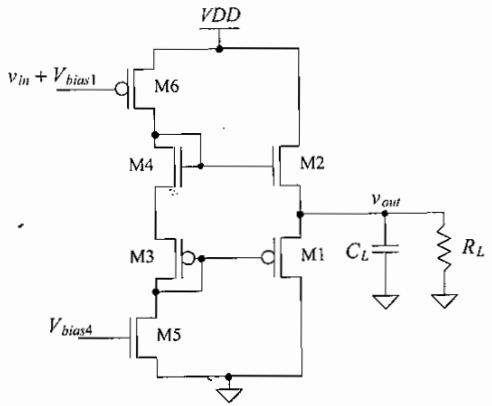


Figure 21.48 Class AB output buffer (M1–M4) using source followers M1 and M2.

The *practical problem* with this buffer is that the output can't swing very close to the power supply voltages (rails). The lowest potential we can get on the gate of M1 is ground, and the highest potential we can get on the gate of M2 is *VDD*. Knowing that M1 has to have a source-to-gate voltage greater than V_{THP} and M2 has to have a gate-source voltage greater than V_{THN}, we can write

$$VDD - V_{THN} \geq v_{out} \geq V_{THP} \tag{21.110}$$

For our short-channel CMOS process (neglecting body effect which will make things worse), the range of output voltages is half the power supply voltage. For example, with *VDD* = 1 V, our output voltage may swing up to (roughly) 750 mV and down to 250 mV. Losing half of the supply voltage often makes this output buffer impractical.

21.3 The Push-Pull Amplifier

What is needed for an output buffer is a topology like an inverter where the output can swing from rail (*VDD*) to rail (ground). Figure 21.49 shows a possible topology with biasing circuitry. The current source, I_{bias}, is a floating current source as seen in Fig. 20.49 of the last chapter. In Fig. 21.49, with zero AC input current, i_{in}, M1 and M2 mirror

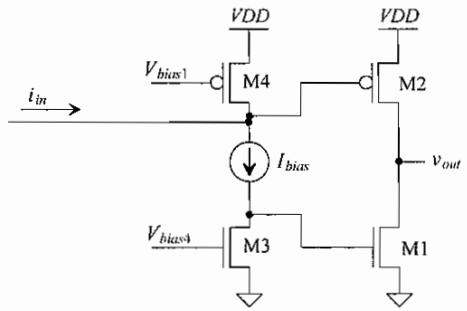


Figure 21.49 Class AB amplifier using an inverter output structure (push-pull).

the current in M3 and M4 (I_{bias}). This precisely sets the current in the output stage (again, important). A positive AC input current causes both the gates of M1 and M2 to go up (shutting M2 off and turning M1 on). M1 and M2 are pushing or pulling a current to/from the output (and so this topology is often called a *push-pull amplifier*). Because the output can swing very close to ground and *VDD* before M1 and M2 triode or shut off, this topology is very useful in modern CMOS output buffer design.

Note that the AC input current can be connected to the gate of M2, as shown in Fig. 21.49, or the gate of M1 (or both).

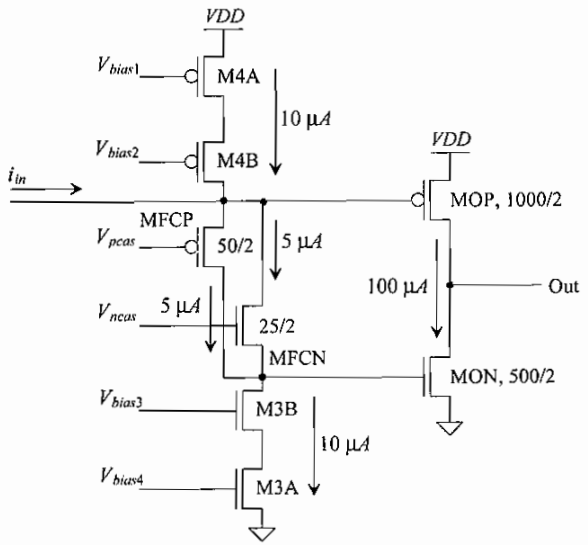
21.3.1 DC Operation and Biasing

The biasing of the push-pull amplifier can be accomplished, as seen in Fig. 21.50 (see also Fig. 20.49). MFCP and MFCN form the floating current source. M3 (A and B) with M4 (A and B) are cascode current sources. If the input current is positive, the gate of MOP is charged up and thus MOP shuts off. At the same time, MFCP turns on (more), causing the gate of MON to go up, turning it on further. If the input current is negative, MOP turns on and MON shuts off.

Figure 21.51 shows the amplifier in Fig. 21.50 with a voltage input (V_{bias4} replaced with an input voltage, V_{in}). M3 and M4 now form a common-source amplifier with a current source load. Shown in Fig. 21.51 is a plot of V_{out} for varying V_{in} with no load. The slope of the curve in this figure is the gain. To show the slope (the gain) as a function of V_{in}, we can take the derivative of the output voltage (also shown in the figure). The gain of this topology (without a load) is roughly 5,000 V/V. If we place a 1k resistor on the output of the amplifier, the gain drops to 1,500 V/V. Using a 100-ohm resistor results in a gain of only 150. The push-pull amplifier, MON/MOP, has a gain less than one when the load resistor is only 100 ohms. If the amplifier did need to drive such a heavy load (small resistance), it would be a good idea to increase the widths of MON and MOP.

Power Conversion Efficiency

The power conversion efficiency (PCE) is defined as the ratio of the power supplied to the load to the power delivered to the amplifier and load by a power supply. In other words, a perfectly efficient amplifier doesn't dissipate any power; rather, all of the power



Bias voltages come from Fig. 20.47 (short-channel parameters in Table 9.2). Unlabeled NMOS are 50/2, while unlabeled PMOS are 100/2.

Figure 21.50 Biasing the push-pull amplifier.

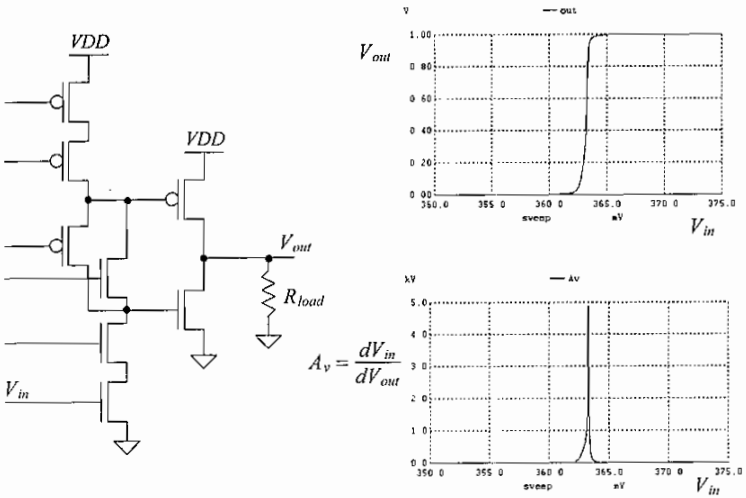


Figure 21.51 Simulating the amplifier in Fig. 21.50 with a voltage input.

from the power supply (*VDD*) is supplied to the load. The PCE is usually written as a percentage

$$\%\ \text{PCE} = \frac{\text{Load power, } P_{load}}{\text{Supply power, } P_{supply}} \times 100\ \% \qquad (21.111)$$

For example, the source follower seen in Fig. 21.41 (a class A amplifier) pulls a fixed current of I_{bias} (= 10 μA for the sizes seen in this figure and Table 9.2). The power supplied (pulled or delivered) from the power supply is

$$P_{supply} = VDD \cdot I_{bias} \qquad (21.112)$$

If the SF is driving a resistive load, R_{load}, the peak value (under ideal conditions) of a sinewave voltage applied to this resistor is *VDD*/2 (the sinewave can go up or down by *VDD*/2). The RMS value of this sinewave is then $VDD/\left(2\sqrt{2}\right)$ and the power supplied to the load is then

$$P_{load} = \frac{\left(VDD/\left(2\sqrt{2}\right)\right)^2}{R_L} \qquad (21.113)$$

The current source, I_{bias}, when the output voltage goes to *VDD*, drives the load resistor directly (M1 in Fig. 21.41 shuts off), that is

$$VDD = I_{bias} \cdot R_L \qquad (21.114)$$

This is the best efficiency because no power is wasted in the amplifier (M1 is off and all of the supply power is delivered to the load). Calculating the PCE of the SF gives

$$\%\ \text{PCE} = \frac{1}{VDD \cdot I_{bias}} \cdot \frac{\left(VDD/\left(2\sqrt{2}\right)\right)^2}{VDD/I_{bias}} \times 100\ \% \qquad (21.115)$$

So the PCE % is, at best, 12.5%. This is the ideal efficiency of a class A amplifier. If the output voltage swing is reduced, the PCE goes down as well.

For the class AB amplifier, the PCE can approach 100%. (Typical values for general designs with low distortion are approximately 75%.) If the current supplied to the load is much larger than the current burned in the amplifier, the PCE is large (much larger than the ideal 12.5% of the class A amplifier). Consider the following example.

Example 21.14

Suppose the amplifier in Figs. 21.50 and 21.51 drives a 1 kΩ load. Using SPICE plot the current supplied to the amplifier and load, the current supplied to only the load, and the current supplied to only the amplifier as a function of the input voltage, V_{in}. Using the results estimate the PCE.

The simulation results are seen in Fig. 21.52. The current supplied to the amplifier includes the bias circuit current. Note how, when the output is 1 V (= *VDD*), the current supplied to the load is 1 mA. The total current supplied by *VDD* is approximately 1.2 mA. At the risk of stating the obvious, the push-pull amplifier is much more power-efficient than the SF (and has a wider output swing).

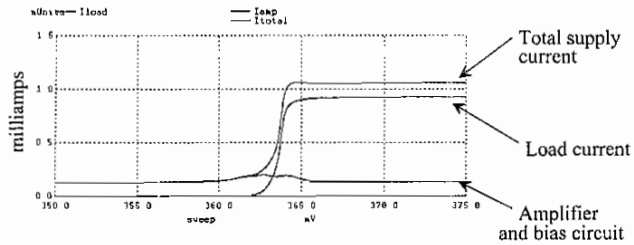


Figure 21.52 Supply, load, and total currents when the amplifier in Fig. 21.51 drives a 1k load resistor.

The PCE efficiency can be estimated by writing the power dissipated by the load

$$P_{load} = \left(\frac{VDD}{2\sqrt{2}\,R_{load}}\right)^2 = \left(\frac{1}{2\sqrt{2}\,1k}\right)^2 = 353\ \mu W$$

The power supplied by *VDD* (assuming the amplifier and bias circuit pull a constant 200 μA of current) is then

$$P_{supply} = P_{load} + VDD \cdot 200\ \mu A = 553\ \mu W$$

The PCE is then

$$\%\ \text{PCE} = \frac{353}{553} \times 100\% = 64\%$$

Note that the bias circuit pulls 140 μA. If we recalculate the PCE without the bias circuit current included, we get

$$\%\ \text{PCE} = \frac{353}{413} \times 100\% = 85\% \quad \blacksquare$$

21.3.2 Small-Signal Analysis

The simplified schematic of the push-pull amplifier is seen in Fig. 21.53. The resistance on the output of the amplifier (the drains of MOP and MON) is $r_{op}||r_{on}||R_{load} \approx R_{load}$. The drain current of MON is $g_{mon}v_{in}$, and the drain current of MOP is $g_{mop}v_{in}$. The output voltage is then the sum of these two currents or

$$v_{out} = -(g_{mon} + g_{mop}) \cdot v_{in} \cdot R_L \qquad (21.116)$$

Note the "resistance in the source" in this amplifier is the parallel combination of the resistance looking into the sources of MON and MOP (both sources are connected to AC ground). We can rewrite Eq. (21.116) as

$$A_{v,push\text{-}pull} = \frac{v_{out}}{v_{in}} = -\frac{R_L}{\frac{1}{g_{mon}}||\frac{1}{g_{mop}}} \qquad (21.117)$$

If the load resistance is 1k and the sum of the transconductances (using wider devices than what is indicated in Tables 9.1 or 9.2, as seen in Fig. 21.50) is 1 mA/V, then the gain of the push-pull amplifier is −1 V/V. Note that if the load resistance changes, the gain of the output stage push-pull amplifier changes as well.

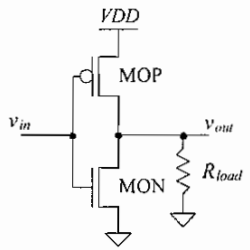


Figure 21.53 Small-signal analysis of the push-pull amplifier.

The gain of the CS amplifier with the cascode load portion of the amplifier in Figs. 21.50 and 21.51 (M3 and M4) is given by Eq. (21.76). The overall small-signal gain of the topology in Fig. 21.51 is given by the product of the two gains or

$$A_v = g_{mn} \cdot \left(g_{mn}r_{on}^2||g_{mp}r_{op}^2\right) \cdot (g_{mon} + g_{mop}) \cdot R_L \qquad (21.118)$$

We might wonder how the floating current source, MFCP and MFCN, affects the gain. If we look at Fig. 21.50, we see that one or the other has a source connection to the drains of either M3 (gate of MON) or M4 (gate of MOP). Further we might think that this source connection would load the gates with a $1/g_m$ small-signal resistance. However, because MFCP and MFCN form a feedback loop, their addition doesn't reduce the output resistance of the CS amplifier (M3 and M4). Consider the following.

Figure 21.54 shows a test circuit used for determining the small-signal resistance that the PMOS current source (M4) sees on its output. To determine this resistance, we apply a test voltage and look at the current that flows. Looking at the figure and noting the current that flows in $r_{on}||r_{op}$ is $(i_t - i_{dn} - i_{dp})$, we can write

$$v_t = (i_t - i_{dn} - i_{dp}) \cdot r_{on}||r_{op} + i_t \cdot R_{meas} \qquad (21.119)$$

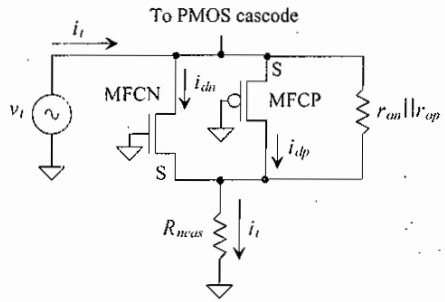


Figure 21.54 Determining the loading affects of the floating current source..

where R_{ncas} is the output resistance of the cascode stack. The gate-source AC voltage of MFCN is $-i_t R_{ncas}$ and so

$$i_{dn} = g_{mn} \cdot (-i_t R_{ncas}) \tag{21.120}$$

Similarly, the drain current through MFCP is

$$i_{dp} = g_{mp} \cdot v_t \tag{21.121}$$

Substituting these equations into Eq. (21.119) gives

$$v_t = (i_t + g_{mn} \cdot i_t R_{ncas} - g_{mp} \cdot v_t) \cdot r_{on} || r_{op} + i_t \cdot R_{ncas} \tag{21.122}$$

or

$$\overbrace{v_t \cdot (1 + g_{mp} \cdot r_{on} || r_{op})}^{\approx g_{mp} \cdot r_{on} || r_{op}} = i_t \cdot \overbrace{(1 + g_{mn} R_{ncas} \cdot r_{on} || r_{op} + R_{ncas})}^{\approx g_{mn} \cdot r_{on} || r_{op} \cdot R_{ncas}} \tag{21.123}$$

The resistance the PMOS cascode sees is then

$$\frac{v_t}{i_t} \approx R_{ncas} \tag{21.124}$$

In other words, the floating current source doesn't load the cascode structure.

21.3.3 Distortion

Small-signal analysis works remarkably well for all of the internal nodes in an op-amp. However, the output buffer must, ideally, swing from rail-to-rail. To illustrate the problem with this, consider the basic gain of a CS amplifier with current source load, Eq. (21.39),

$$|A_v| = g_{m1} \cdot (r_{o1} || r_{o2}) = \frac{\sqrt{2\beta_1 (I_D + i_d)}}{2(\lambda_n + \lambda_p)(I_D + i_d)} \approx \frac{\sqrt{2\beta_1}}{2(\lambda_n + \lambda_p)\sqrt{(I_D + i_d)}} \tag{21.125}$$

Normally, the AC component of the drain current, i_d, is assumed to be much less than the DC component of the drain current, I_D, and the amplifier gain is essentially constant (small-signal approximation). If the AC component is comparable to the DC component, noticeable distortion results. The voltage gain for large inputs depends on the input signal amplitude.

Characterizing an amplifier begins by applying a pure single-tone sinusoid of the form

$$V_{in}(t) = V_p \sin 2\pi f \cdot t \tag{21.126}$$

to the input of the amplifier. The output of the amplifier is a series of tones at an integer multiple of the input tone given by

$$V_{out}(t) = a_1 V_p \sin(\omega t) + a_2 V_p \sin(2\omega t) + \cdots + a_n V_p \sin(n\omega t) \tag{21.127}$$

The magnitude of the fundamental or wanted signal is $a_1 V_p$. Ideally, a_2 through a_n are zero, and the amplifier is free of distortion. The n^{th} term harmonic distortion is given by

$$HD_n = \frac{a_n}{a_1}, \text{ for } n > 1 \tag{21.128}$$

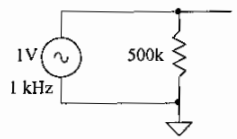
The *total harmonic distortion* (THD) is given by

$$THD = \sqrt{\frac{a_2^2 + a_3^2 + a_4^2 + \cdots + a_n^2}{a_1^2}} \tag{21.129}$$

Again, output buffers, amplifiers used to drive a large load capacitance or low resistance, are examples of amplifiers where low THD is important. If the output amplifier is biased so that the DC component of the drain current is large compared to the transient or time-varying current, the buffer will dissipate too much power for most applications. Therefore, in almost all situations, the biasing current in an output buffer is comparable, or even smaller for class B operation, than the time-varying current. We reduce the distortion by employing feedback around the amplifier. If the open-loop gain of an op-amp varies from 1,000 to 10,000 depending on the amplitude of the input signal, then the feedback around the amplifier reduces the gain sensitivity. In fact, it is nearly impossible to design a linear output amplifier with low distortion without using feedback.

Modeling Distortion with SPICE

SPICE can be used to simulate distortion using a transient analysis and the .FOUR (Fourier) statement. The general form of this statement is .FOUR FREQ OV1 <OV2 OV1 . . . where FREQ is the frequency of the fundamental and OV1 . . . are the outputs of the circuit (the voltage or current outputs for which SPICE will calculate distortion). As a simple example, consider the circuit and netlist shown in Fig. 21.55. The input sine wave must have at least one full period for the .FOUR statement to calculate distortion. In the case where there's more than one period, SPICE uses the output over the last full period. Also, the maximum transient step size should be less than the period of the input divided by 100. For the example of Fig. 21.55 with a 1 kHz input (a period of 1 ms), the maximum print size should be 10 µs.




```
*** SPICE Fourier Analysis Example ***
.four 1k Vinout
R1 Vinout 0 500k
Vinout Vinout 0 DC 0 AC 0 0 SIN(0 1 1khz 0 0)
.tran 10u 2m 0 10u
.end
```

Figure 21.55 Simple circuit to demonstrate the use of the .FOUR statement.

The resulting simulation output follows.

```
Fourier analysis for vinout:
 No. Harmonics: 10, THD: 3.17012e-06 %, Gridsize: 200, Interpolation Degree: 1

Harmonic Frequency    Magnitude     Phase         Norm. Mag     Norm. Phase
0  0.000000e+00  -5.60197e-09  0.000000e+00  0.000000e+00  0.000000e+00
1  1.000000e+03   9.996317e-01 -9.39744e-06  1.000000e+00  0.000000e+00
2  2.000000e+03   1.120393e-08 -8.64000e+01  1.120806e-08 -8.64000e+01
3  3.000000e+03   1.120393e-08 -8.46000e+01  1.120806e-08 -8.46000e+01
4  4.000000e+03   1.120394e-08 -8.28000e+01  1.120806e-08 -8.28000e+01
5  5.000000e+03   1.120394e-08 -8.10000e+01  1.120806e-08 -8.10000e+01
6  6.000000e+03   1.120394e-08 -7.92000e+01  1.120806e-08 -7.92000e+01
```

```
7    7.000000e+03   1.120393e-08   -7.74000e+01   1.120806e-08   -7.74000e+01
8    8.000000e+03   1.120393e-08   -7.56000e+01   1.120806e-08   -7.56000e+01
9    9.000000e+03   1.120393e-08   -7.38000e+01   1.120806e-08   -7.38000e+01
```

Notice that, as we would expect, the output of this circuit doesn't show any harmonic distortion. SPICE calculates the magnitude and phase of the first nine harmonics and DC. Also note that SPICE automatically calculates the THD for a circuit.

As a more practical example, consider the push-pull amplifier shown in Fig. 21.56. The SPICE netlist for simulating the distortion performance of this output amplifier is shown below. Note that to avoid a long start-up transient we started the big capacitors at the DC bias voltages (V_{bias1} for the PMOS and V_{bias4} for the NMOS). When the input sinewave has an amplitude of 2 mV, the THD is 0.6%. When the input drops to 100 μV (small-signal approximation is more valid), the THD drops to 0.046%. If the input amplitude increases to 10 mV (so the output of the push-pull amplifier swings close to the rails), the THD increases to 9.2%.

```
*** Figure 21.56 CMOS: Circuit Design, Layout, and Simulation ***

.option scale=50n
.tran 10n 2u UIC
.four 1MEG Vout
VDD   VDD   0      DC    1
Vin   Vin   0      DC    0      sin 0 2m 1MEG
RL    out   0      1k
Rbigp Vbias1 vgp   1G
Cbigp vgp   vin    1 IC=0.643
Rbign Vout  vgn    1G
Cbign Vgn   vin    1 IC=0.362
Xbias VDD Vbias1 Vbias2 Vbias3 Vbias4 Vhigh Vlow Vncas Vpcas bias
MON   Vout  vgn    0      0      NMOS L=2 W=500
MOP   Vout  vgp    VDD    VDD    PMOS L=2 W=1000
(subcircuit and SPICE models not shown)
```

Note that another useful SPICE tool to evaluate the distortion introduced into a signal by a circuit is a Discrete Fourier Transform (DFT). The spec command is used in SPICE (spectral analysis) to determine the output spectrum of an amplifier. Using this command is covered in depth in the second volume of this book entitled, *CMOS Mixed Signal Circuit Design*.

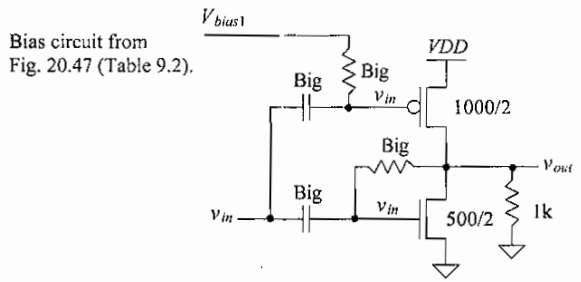


Figure 21.56 Simulating the distortion in a push-pull amplifier.

ADDITIONAL READING

[1] K. N. Leung, P. K. T. Mok, W. Ki, and J. K. O. Sin, "Three-stage large capacitive load amplifier with damping-factor-control frequency compensation," *IEEE Journal of Solid-State Circuits,* vol. 35, pp. 221–230, February 2000.

[2] A. B. Dowlatabadi, "A robust, load-insensitive pad driver," *IEEE Journal of Solid-State Circuits,* vol. 35, pp. 660–665, April 2000.

[3] B. Razavi, "CMOS technology characterization for analog and RF design," *IEEE Journal of Solid-State Circuits,* vol. 34, pp. 268–276, March 1999.

[4] K. de Langen and J. H. Huijsing, "Compact low-voltage power-efficient operational amplifier cells for VLSI," *IEEE Journal of Solid-State Circuits,* vol. 33, pp. 1482–1496, October 1998.

[5] P. E. Allen, B. J. Blalock, and G. A. Rincon, "A 1V CMOS Opamp using Bulk-Driven MOSFETs," *IEEE International Solid-State Circuits Conference,* vol. 38, pp. 192–193, February 1995.

[6] R. G. Eschauzier, R. Hogervorst, and J. H. Huijsing, "A programmable 1.5 V CMOS class-AB operational amplifier with hybrid nested Miller compensation for 120 dB gain and 6 MHz UGF," *IEEE Journal of Solid-State Circuits,* vol. 29, pp. 1497–1504, December 1994.

[7] R. Hogervorst, J. P. Tero, R. G. H. Eschauzier, and J. H. Huijsing, "A Compact Power-Efficient 3 V CMOS Rail-to-Rail Input/Output Operational Amplifier for VLSI Cell Libraries," *IEEE Journal of Solid State Circuits,* vol. 29, pp. 1505–1513, December 1994.

[8] A. A. Abidi, "On the operation of cascode gain stages," *IEEE Journal of Solid-State Circuits,* vol. 23, pp. 1434–1437, December 1988.

[9] S. L. Wong and C. A. T. Salama, "An efficient CMOS buffer for driving large capacitive loads," *IEEE Journal of Solid-State Circuits,* vol. 21, pp. 464–469, June 1986.

[10] D. B. Ribner and M. A. Copeland, "Design techniques for cascoded CMOS op amps with improved PSRR and common-mode input range," *IEEE Journal of Solid-State Circuits,* vol. 19, pp. 919–925, December 1984.

[11] V. R. Saari, "Low-power high-drive CMOS operational amplifiers," *IEEE Journal of Solid-State Circuits,* vol. 18, pp. 121–127, February 1983.

[12] B. K. Ahuja, "An Improved Frequency Compensation Technique for CMOS Operational Amplifiers," *IEEE Journal of Solid-State Circuits,* vol. 18, pp. 629–633, December 1983.

[13] P. R. Gray and R. G. Meyer, "MOS operational amplifier design - A tutorial overview," *IEEE Journal of Solid-State Circuits,* vol. 17, pp. 969–982, December 1982.

[14] B. J. Hosticka, "Dynamic CMOS amplifiers," *IEEE Journal of Solid-State Circuits,* vol. 15, pp. 887–894, October 1980.

PROBLEMS

21.1 Using simulations, show that the small-signal resistance of a gate-drain-connected PMOS device, Fig. 21.57, behaves like a resistor with a value of $1/g_m$.

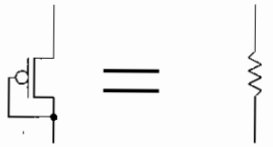


Figure 21.57 The small-signal behavior of a diode-connected MOSFET.

21.2 Estimate the frequency response of the circuit seen in Fig. 21.58. Verify your hand calculations using SPICE.

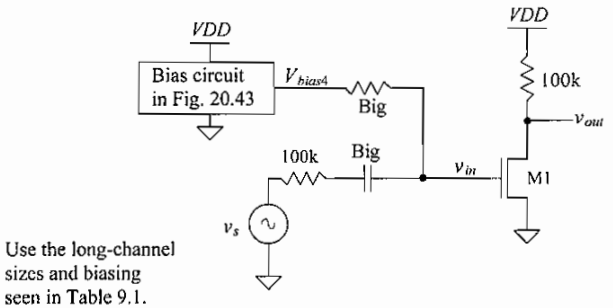


Figure 21.58 Amplifier used in Problem 21.2.

21.3 Repeat problem 21.2 if the amplifier drives a 100 *fF* load capacitance.

21.4 For the circuit in Fig. 21.12, estimate the effective transconductance of the circuit, g_{meff}, that relates the MOSFET drain current to the AC input voltage, v_{in}. Verify your solution with SPICE.

21.5 Simulate the operation of the common-gate amplifier in Fig. 21.16 using the bias circuit in Fig. 20.43 and the device sizes seen in Table 9.1. Compare the simulation results to hand calculations.

21.6 Estimate the transfer function of the amplifier in Ex. 21.6 if it drives a 100 fF load. Verify your hand calculations using simulations.

21.7 Determine the frequency response of the amplifier seen in Fig. 21.59. Verify your hand calculations using simulations.

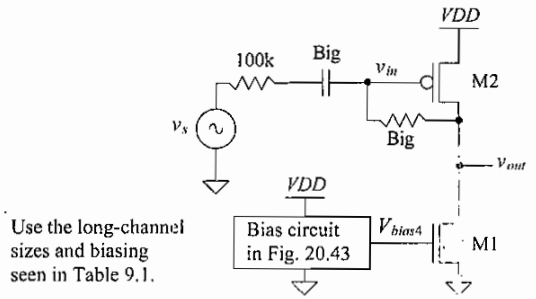


Figure 21.59 Amplifier for Problem 21.7.

21.8 Repeat Ex. 21.9 using the biasing circuit from Fig. 20.47 and the sizes in Table 9.2.

21.9 Repeat Ex. 21.10 using the biasing circuit from Fig. 20.47 and the sizes in Table 9.2.

21.10 Repeat Ex. 21.12 using the biasing circuit from Fig. 20.43 and the sizes in Table 9.1.

21.11 Show, using SPICE simulations and an ideal current source of 10 μA (device size from Table 9.2), the difference in the voltage gains with and without body effect for the circuit seen in Fig. 21.60.

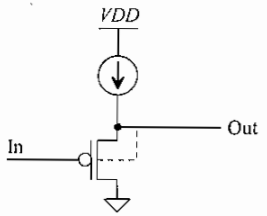


Figure 21.60 Gain of an SF with and without body effect for Problem 21.11.

21.12 Does the SF in Fig. 21.46 exhibit slew-rate limitations if it needs to discharge the 1-pF load capacitance quickly? Why or why not? Verify your answer using simulations.

21.13 Simulate the operation of the class AB output buffer in Fig. 21.48 using the bias circuit from Fig. 20.47 and the sizes in Table 9.2. If the buffer is driving a load resistor of 10k, plot v_{in} against v_{out}. What is the linear output range?

21.14 In the class AB output buffer in Fig. 21.50 or 21.51, a load resistor connected to ground causes the PMOS device to conduct more current then the NMOS. Why? Resimulate the buffer in Fig. 21.51 if it drives a 1k resistor and MON is reduced in size to 50/2. Is this a better design? Why or why not?

21.15 Using simulations, show the problem of using an inverter output buffer without a floating current source as seen in Fig. 21.53. (The quiescent current that flows in the MOSFETs is huge and not accurately controlled as it is in Fig. 21.50.)

21.16 Is the distortion the output buffer in Fig. 21.56 introduces into a signal a function of the load resistance? Verify your answer with simulations and show some time-domain waveforms with and without distortion.

Chapter
22

Differential Amplifiers

In the last chapter big resistors and capacitors were used to bias the circuits to the correct operating point, as seen in Fig. 21.21. The DC operating voltage on the gate of M1 (in Figs. 21.17 or 21.21) is extremely important when biasing the amplifier. If it's not at precisely the correct value, then the current sourced by M2 won't equal the current in M1 when both MOSFETs are operating in the saturation region.

The differential amplifier (*diff-amp*) is used on the input of an amplifier to allow input voltages to move around so that biasing of the gain stages isn't affected (that is, so it isn't a function of the input voltage). The diff-amp is a fundamental building block in CMOS analog integrated circuit design, and an understanding of its operation and design is extremely important. In this chapter we discuss three basic types of differential amplifiers: the source-coupled pair, the source cross-coupled pair, and the current differential amplifier.

22.1 The Source-Coupled Pair

The source-coupled pair comprised of M1 and M2 is shown in Fig. 22.1. When M1 and M2 are used in this configuration they are sometimes called a *diff-pair*.

22.1.1 DC Operation

The diff-pair in Fig. 22.1 is biased with a current source so that

$$I_{SS} = i_{D1} + i_{D2} \tag{22.1}$$

If we label the input voltages at the gates of M1 and M2 as v_{I1} and v_{I2}, we can write the input difference as

$$v_{DI} = v_{I1} - v_{I2} = v_{GS1} - v_{GS2} \tag{22.2}$$

or in terms of the AC and DC components of the differential input voltage, v_{DI},

$$v_{DI} = V_{GS1} + v_{gs1} - V_{GS2} - v_{gs2} \tag{22.3}$$

When the gate potentials of M1 and M2 are equal, then (assuming both are operating in the saturation region)

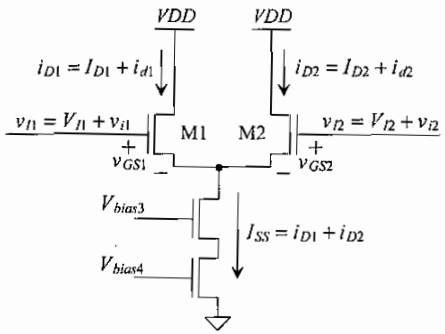


Figure 22.1 A basic NMOS source-connected pair made using the diff-pair M1 and M2.

$$I_{D1} = I_{D2} = \frac{I_{SS}}{2} \tag{22.4}$$

Maximum and Minimum Differential Input Voltage

Since we know that a saturated MOSFET follows the relation

$$i_D = \frac{\beta_n}{2}(v_{GS} - V_{THN})^2 \tag{22.5}$$

the difference in the input voltages may be written as

$$v_{DI} = \sqrt{\frac{2}{\beta_n}}\left(\sqrt{i_{D1}} - \sqrt{i_{D2}}\right) \tag{22.6}$$

The maximum difference in the input voltages, v_{DIMAX} (maximum differential input voltage), is found by setting i_{D1} to I_{SS} (M1 conducting all of the tail bias current) and i_{D2} to 0 (M2 off)

$$v_{DIMAX} < v_{I1} - v_{I2} = \sqrt{\frac{2 \cdot L \cdot I_{SS}}{KP_n \cdot W}} \tag{22.7}$$

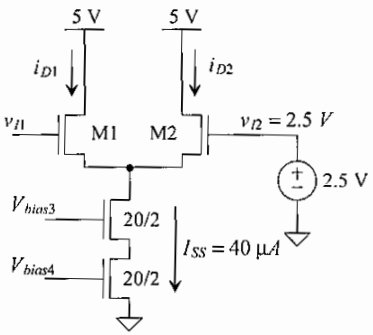
The minimum differential input voltage, v_{DIMIN}, is found by setting i_{D2} to I_{SS} and i_{D1} to 0

$$v_{DIMIN} = -v_{DIMAX} = -(v_{I1} - v_{I2}) = -\sqrt{\frac{2 \cdot L \cdot I_{SS}}{KP_n \cdot W}} \tag{22.8}$$

Example 22.1

Estimate the maximum and minimum voltage on the gate of M1 in Fig. 22.2 that ensures that neither M1 or M2 shut off. Verify your hand calculations with SPICE simulations.

Note that the diff-amp tail current, I_{SS}, is 40 μA (the widths of the MOSFETs were doubled from the sizes indicated in Table 9.1). When $v_{I1} = v_{I2} = 2.5$ V, the differential input voltage, v_{DI}, is 0 and the current flowing in M1 and M2 is 20 μA ($= i_{D1} = i_{D2}$).



Parameters from Table 9.1
Bias circuit from Fig. 20.43

Figure 22.2 Diff-amp used in Ex. 22.1.

Using Eq. (22.7), we can estimate the maximum voltage on the gate of M1 as

$$v_{I1MAX} = \sqrt{\frac{2 \cdot L \cdot I_{SS}}{KP_n \cdot W}} + v_{I2} = \sqrt{\frac{2 \cdot 2 \cdot 40}{120 \cdot 10}} + 2.5 = 2.865\ V$$

or v_{I1} is 365 mV above v_{I2}. The minimum voltage allowed on the gate of M1 (to keep some current flowing in M1) is then 365 mV below v_{I2} or 2.135 V. The simulation results are seen in Fig. 22.3. Notice how, when the inputs are equal (both are 2.5 V), the currents in M1 and M2 are equal and approximately 20 μA ($= I_{SS}/2$). ■

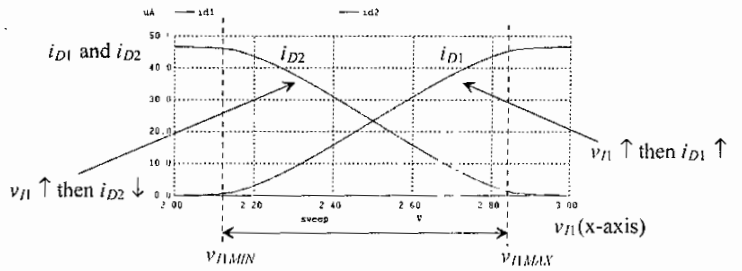


Figure 22.3 Simulating the operation of the diff-amp in Fig. 22.2.

Maximum and Minimum Common-Mode Input Voltage

When the diff-amp is used on the input of an op-amp, the inputs are forced, via feedback around the op-amp, to the same values (or very nearly the same values). This value (or more precisely the average of the two inputs) is called the *common-mode voltage*. Figure 22.4 shows the diff-amp with both inputs tied together. We're interested in the maximum and minimum voltage that will keep both M1 and M2 operating in the saturation region

(that is, not off or in the triode region). We'll call the maximum common-mode voltage, v_{CMMAX} and the minimum common-mode voltage, v_{CMMIN}. Note that when the common-mode voltage (the potential on the gates of M1 and M2) is too high, both M1 and M2 triode (behave like resistors). When the common-mode voltage is too low, M1 and M2 shut off.

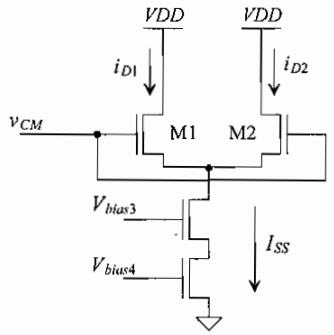


Figure 22.4 Diff-amp used to calculate the minimum and maximum diff-amp input voltages.

To determine the maximum input voltage, we know that for M1 and M2 to remain in the saturation region

$$V_{DS} \geq V_{GS} - V_{THN} \rightarrow V_D \geq V_G - V_{THN} \tag{22.9}$$

Because $V_D = VDD$, we can write

$$V_{CMMAX} = VDD + V_{THN} \tag{22.10}$$

The input common-mode voltage can actually be higher than *VDD* before M1/M2 move into the triode region.

For the minimum input voltage, we can write

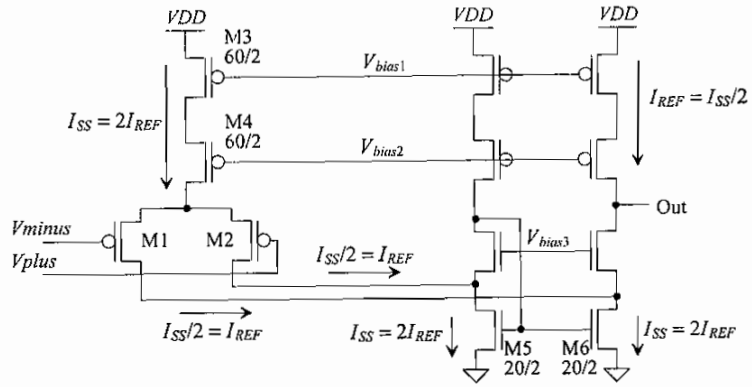
$$V_{CMMIN} = V_{GS1,2} + 2 \cdot V_{DS,sat} \tag{22.11}$$

where the minimum voltage across the current source is assumed to be $2V_{DS,sat}$. For the long-channel process, from Table 9.1, the minimum input voltage for an NMOS diff-amp is 1.55 V.

Example 22.2

Figure 22.5 shows a folded cascode amplifier (see Fig. 20.45). Assuming that all MOSFETs are operating in the saturation region, estimate the minimum and maximum input voltage of the amplifier.

Note how the widths of M5–M6 are doubled to sink the additional current from the diff-amp. It's important to understand how the sizes of the MOSFETs are adjusted in the current mirrors so that the currents sum correctly. The currents, as seen in Fig. 22.5, are labeled assuming *Vplus* = *Vminus*.



Parameters from Table 9.1
Unlabeled PMOS are 30/2
Unlabeled NMOS are 10/2
Bias circuit from Fig. 20.43

Figure 22.5 Folded cascode amplifier.

The maximum allowable input common-mode voltage is determined by

$$V_{CMMAX} = VDD - V_{SG} - 2V_{SD,sat} = 5 - 1.15 - 0.5 = 3.35\ V$$

The minimum input common-mode voltage is determined by noting the drain voltage of M5 and M6 (or M1 and M2) is a $V_{DS,sat}$

$$V_{SD} \geq V_{SG} - V_{THP} \rightarrow V_D \leq V_G + V_{THP}$$

and so

$$V_{CMMIN} = V_D - V_{THP} = 0.25 - 0.9 = -0.65$$

In other words, the input common-mode voltage can actually be below ground and the amplifier will still function correctly. This result is *very useful*. The folded- cascode amplifier with PMOS diff-amp can be used when the input voltage swings around ground. Note, however, that the output voltage of the amplifier will not swing below ground but is, rather, limited to $2V_{DS,sat}$ above ground and $2V_{SD,sat}$ below *VDD* if the MOSFETs are to remain in saturation (the high-gain region of operation). ■

Current Mirror Load

The previous uses of the diff-amp have exploited the fact that the output of the diff-amp is a current controlled by a differential input voltage, v_{Dl}. In some applications, it is desirable to have a diff-amp with a voltage output. Towards this goal, consider the diff-amp with a current mirror load, as seen in Fig. 22.6. An imbalance in the drain currents of M1 and M2 causes the output of the diff-amp to swing either towards *VDD* or

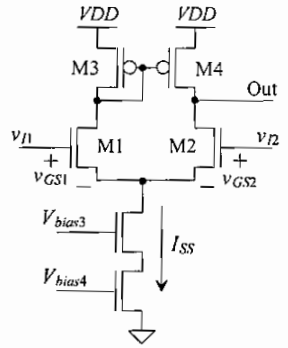


Figure 22.6 Diff-amp with a current mirror load.

ground. The minimum input common mode voltage is, once again, given by Eq. 22.11. The maximum input common mode voltage is determined knowing that the drain voltage of M2 is the same as the drain voltage of M1 (when both diff-amp inputs are the same potential), that is, $VDD-V_{SG}$ of the PMOS. We can therefore write

$$V_{DS} \geq V_{GS} - V_{THN} \rightarrow V_D \geq V_G - V_{THN} \rightarrow V_{CMMAX} = VDD - V_{SG} + V_{THN} \quad (22.12)$$

For the parameters in Table 9.1, $V_{CMMAX} = 5 - 1.15 + 0.8 = 4.65\ V$.

The voltage output swing can be determined by noting that the maximum voltage output is limited by keeping M4 in saturation. Therefore,

$$V_{OUTMAX} = VDD - V_{SD,sat} \quad (22.13)$$

The minimum output voltage is determined by the voltage on the gate of M2 (M2 must remain in saturation).

$$V_D \geq V_G - V_{THN} \rightarrow V_{OUTMIN} = V_{I2} - V_{THN} \quad (22.14)$$

Example 22.3

Suppose the diff-amp in Fig. 22.6 is implemented with the sizes and bias currents seen in Table 9.1 (I_{SS} = 40 μA). If the gate of M2 is held at 4 V, estimate the diff-amp's output swing. Verify the answers with SPICE.

From Eq. (22.13), the output can swing up to 4.75 V before M4 triodes. From Eq. (22.14), the output can swing down to 4–0.8 or 3.2 V before M2 triodes. The simulation results are seen in Fig. 22.7. Looking at the results, we see the output goes down to approximately 2.8 V or 400 mV less than what we predicted. This can be attributed to the body effect in M1 and M2. With body effect, the threshold voltage is 1.2 V (instead of 0.8 V). ■

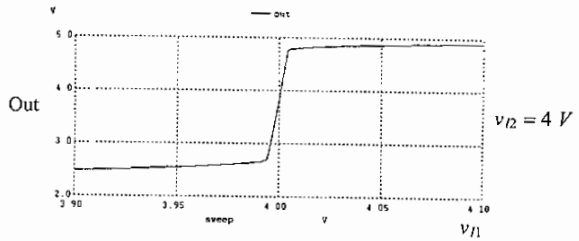


Figure 22.7 Simulation results for Ex. 22.3

Biasing from the Current Mirror Load

Consider the connection of the common-source amplifier, M7, to the output of the diff-amp in Fig. 22.8. When the inputs to the diff-amp are at the same potential, the currents that flow in M3 and M4 are equal (= $I_{SS}/2$). We know from Ch. 20 that the drain of M4 is then at the same potential as its gate. This means, for biasing purposes, that the gate of M7 can be treated as if it were tied to the gate of M3 (M4).

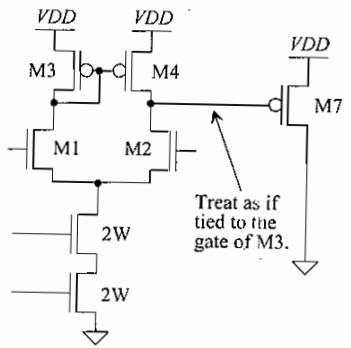


Figure 22.8 Using a diff-amp to bias the next stage amplifier.

Minimum Power Supply Voltage

Looking at the diff-amp in Fig. 22.6, we can estimate the minimum allowable power supply voltage, VDD_{min}, as

$$VDD_{min} = V_{SG3} + \overbrace{V_{DS,sat1}}^{\text{minimum voltage across M1}} + \overbrace{2V_{DS,sat}}^{\text{minimum voltage across Is}} \quad (22.15)$$

Looking at this equation, we see that the V_{SG} of M3 is the "weak link" in the minimum power supply voltage. Using the parameters from Table 9.2 (the short-channel devices with a VDD of 1 V), we get a VDD_{min} of 500 mV. We can lower this value by using a single MOSFET to bias the diff-amp (instead of a cascode circuit).

22.1.2 AC Operation

To describe the AC small-signal operation of the diff-amp in Fig. 22.1, consider the AC schematic seen in Fig. 22.9. We've replaced the current source with an open and the DC supply (*VDD*) with a short (to ground). In the following discussion, it's important to remember that a negative AC current simply means that the overall current (AC + DC) is decreasing (see Ex. 9.5 in Ch. 9). We can write

$$v_{di} = v_{i1} - v_{i2} = v_{gs1} - v_{gs2} = \frac{i_{d1}}{g_m} - \frac{i_{d2}}{g_m} \tag{22.16}$$

Reviewing Fig. 22.9, we see that

$$i_{d1} = -i_{d2} = i_d \tag{22.17}$$

and thus

$$v_{gs1} = -v_{gs2} = \frac{v_{di}}{2} \tag{22.18}$$

so finally

$$g_m v_{gs1} = g_m(-v_{gs2}) = i_d = g_m \cdot \frac{v_{di}}{2} \tag{22.19}$$

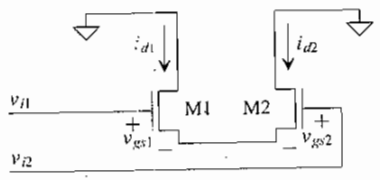


Figure 22.9 AC circuit for a diff-amp.

Example 22.4
Estimate the AC component of the drains of M1 and M2 for the circuit in Fig. 22.2 if $v_{i1} = 2.5 + 1mV \cdot \sin(2\pi \cdot 1kHz \cdot t)$. Verify your hand calculations using SPICE.

From Table 9.1 we know the g_m of the NMOS devices used in the diff-pair is 150 $\mu A/V^2$. The AC component of v_{i2} is zero so we know

$$v_{gs1} = -v_{gs2} = 0.5\ mV$$

and thus

$$i_d = g_m v_{gs} = (150 \times 10^{-6})(500 \times 10^{-6}) = 75\ nA$$

This is the AC component of the drain currents. When i_{d1} = 75 nA, then i_{d2} = –75 nA. The overall drain currents are written as

$$i_{D1} = 20\ \mu A + (75\ nA) \cdot \sin(2\pi 1k \cdot t) \text{ and } i_{D2} = 20\ \mu A - (75\ nA) \cdot \sin(2\pi 1k \cdot t)$$

The simulation results are seen in Fig. 22.10. ■

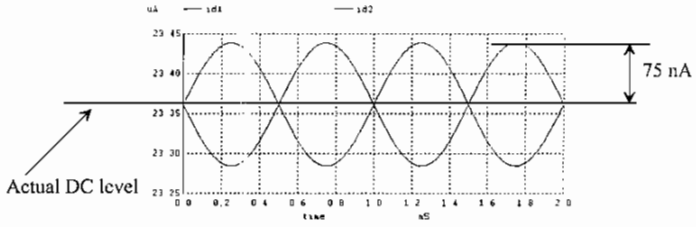


Figure 22.10 Simulation results for Ex. 22.10.

AC Gain with a Current Mirror Load

To determine the AC gain of the diff-amp with a current mirror load, Fig. 22.6, consider the small signal model seen in Fig. 22.11. We've replaced the diode-connected MOSFET, M3, with a $1/g_{m3}$ resistor. Also the resistance looking into the drain of M4 (r_{o4}) and the resistance looking into the drain of M2 (r_{o2}) are drawn explicitly. We assumed that the small-signal resistance looking into the drain of M2 is simply r_{o2} (for all practical design cases this is a good assumption). Note how, because of the current mirror action, the AC current flowing in r_{o4} is i_{d1}. The current in M3 is mirrored by M4. The output voltage can be written as

$$v_{out} = (i_{d1} - i_{d2}) \cdot (r_{o2} || r_{o4}) \tag{22.20}$$

Knowing $i_{d1} = -i_{d2} = i_d$

$$v_{out} = 2i_d \cdot (r_{o2} || r_{o4}) \tag{22.21}$$

The differential mode gain is then written with the help of Eq. (22.19) as

$$A_d = \frac{v_{out}}{v_{di}} = \frac{v_{out}}{v_{i1} - v_{i2}} = g_m \cdot (r_{o2} || r_{o4}) \tag{22.22}$$

Notice that as the voltage on the gate of M1 increases, the current in M1 (and M3/M4) increases. This causes the output voltage to increase. When the gate potential of M2 increases, so does its drain current, causing the output voltage to decrease. Sometimes the gate potential of M1 is called the *noninverting* input and the gate potential of M2 is called the *inverting* input.

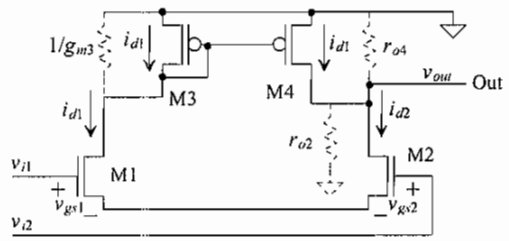


Figure 22.11 AC circuit of the diff-amp with a current-source load.

Example 22.5
Determine the output voltage for the diff-amp seen in Fig. 22.12. Verify your hand calculations using SPICE.

The tail current of the diff-amp conducts 20 μA so M1 and M2, when the inputs are equal, each conduct 10 μA. Using the parameters from Table 9.2 and Eq. (22.22), the gain of the diff-amp is

$$A_d = g_m \cdot (r_{on} || r_{op}) = (150 \times 10^{-6}) \cdot \left(\frac{167 \cdot 333}{167 + 333} \times 10^3 \right) = 16.7\ V/V$$

This means that the 1 mV AC input will appear as a 16.7 mV AC output. The DC level on the output is $VDD - V_{SG} = 650\ mV$. The output voltage is then

$$v_{out}(t) = 0.65 + (16.7\ mV) \cdot \sin(2\pi 10 MHz \cdot t)$$

The simulation results are seen in Fig. 22.13. ■

Bias circuit from Fig. 20.47.
See Table 9.2.

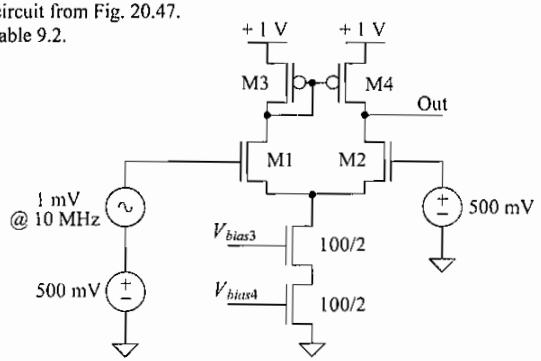


Figure 22.12 Circuit used in Ex. 22.5.

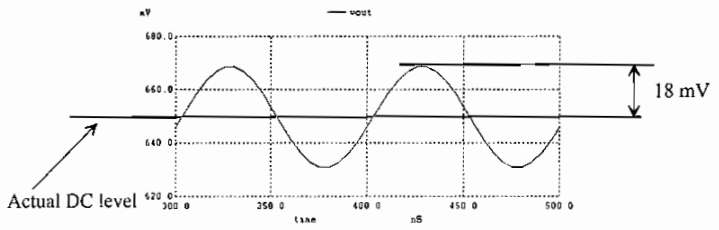


Figure 22.13 The output of the circuit in Fig. 22.12. Note that the transient analysis doesn't show the beginning of the simulation where the reference circuit is starting up.

Example 22.6
Estimate the f_{3dB} of the diff-amp in Ex. 22.5 if it drives a load capacitance of 1 pF. Verify your hand calculations with SPICE.

The resistance on the output node is $r_{o2} || r_{o4} = 111\ k\Omega$; therefore, since this is the *high-impedance node*, we can estimate the circuit's 3-dB frequency in the circuit as

$$f_{3dB} = \frac{1}{2\pi \cdot 111k \cdot 1\ pF} = 1.4\ MHz$$

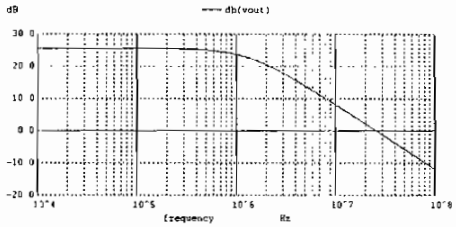


Figure 22.14 The AC response of the diff-amp in Fig. 22.12 when driving a 1 pF load capacitance.

The simulation results are seen in Fig. 22.14. ■

22.1.3 Common-Mode Rejection Ratio

An important aspect of the differential amplifier is its ability to reject a common signal applied to both inputs. Often, in analog systems, signals are transmitted differentially, and the ability of an amplifier to reject coupled noise into each line is very desirable. Consider the amplifier shown in Fig. 22.15. The bias current at the sources of M1/M2 has been replaced with its small-signal output resistance. The common-mode signal (the "noise" labeled in the figure) on both inputs is, ideally, rejected by the diff-amp. The diff-amp's output voltage doesn't vary. The noise causes the source potentials of M1/M2 to vary (the voltage across the tail current source).

If we apply an AC signal, v_c, to the gates of M1/M2, equivalent to saying that a common signal is applied to the input of the differential amplifier, we can calculate the common-mode gain. We begin by writing the AC small-signal, common-mode input voltage, v_c, as

$$v_c = v_{gs1,2} + 2 i_d R_o \qquad (22.23)$$

The MOSFETs M1 and M2 each source a current, i_d, through the output resistance of the current source, R_o (hence the factor of two in this equation). This equation may be rewritten as

$$v_c = i_d \left(\frac{1}{g_m} + 2R_o \right) \approx i_d 2R_o \qquad (22.24)$$

The output voltage, because of the symmetry of the circuit, is given by

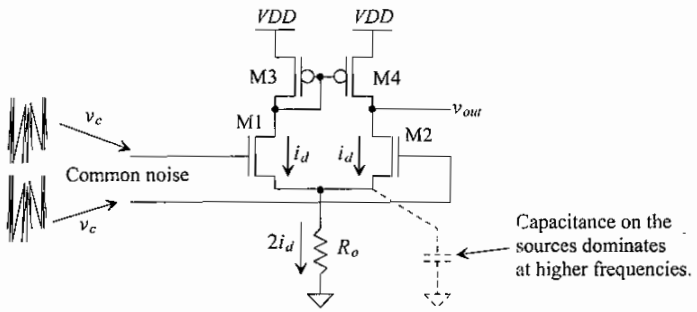


Figure 22.15 Calculating the CMRR of a diff-amp.

$$v_{out} = -i_d \cdot \frac{1}{g_{m3}} = -i_d \cdot \frac{1}{g_{m4}} \tag{22.25}$$

The common-mode gain is then

$$A_c = \frac{v_{out}}{v_c} = \frac{-1/g_{m3,4}}{2R_o} = -\frac{1}{2g_{m3,4}R_o} \tag{22.26}$$

Notice that by increasing the output resistance of the biasing current source, the common-mode gain drops. (This is good because we don't want any common signal to be amplified by the diff-amp.) The common-mode rejection ratio (*CMRR*), in dB, of a diff-amp with a current mirror load is given by

$$CMRR = 20 \cdot \log\left|\frac{A_d}{A_c}\right| = 20 \cdot \log\left[g_{m1,2}(r_{o2}||r_{o4}) \cdot 2g_{m3,4}R_o\right] \tag{22.27}$$

The larger the *CMRR*, the better the performance of the diff-amp. For the diff-amp in Fig. 22.12, Eq. (22.27) evaluates to 86 dB. As the frequency of the input common-mode signal, v_c, increases, the *CMRR* goes down. This is the result of the impedance of the capacitance to ground on the sources of the diff-pair becoming much smaller than R_o.

Example 22.7

Simulate the *CMRR* of the diff-amp in Fig. 22.12. Show that the *CMRR* falls with increasing frequency.

As mentioned above, the (low-frequency) calculated *CMRR* is 86 dB. To simulate the *CMRR*, we perform AC simulations on two of the diff-amps in Fig. 22.12 so we can simultaneously determine A_d and A_c. The ratio of these gains is the *CMRR*. Figure 22.16 shows the simulation results. The common-mode gain, A_c, depends on the DC common-mode voltage. The results seen in Fig. 22.16 are with a common-mode (DC) voltage of 700 mV. Dropping the common-mode voltage down to 500 mV causes the low-frequency *CMRR* to drop down to 50 dB. The reason for this is that, with higher DC common-mode voltages, the voltage across the tail current source is higher, so its output resistance is larger. ■

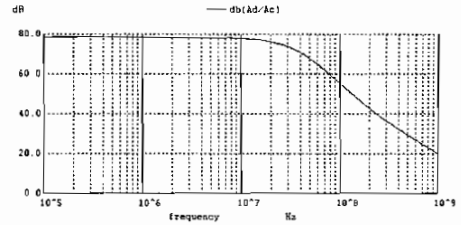


Figure 22.16 The simulated CMRR for the diff-amp in Fig. 22.12.

Input-Referred Offset from Finite CMRR

It's important we understand, in Eq. (22.26), why large R_o reduces A_c. Looking at Fig. 22.15, notice that as v_c varies so too will the voltage across the biasing current source (represented in this figure as R_o). As seen in Ch. 20, Fig. 20.4, variations in the voltage across a current source changes the output of the current source (unless its output resistance is infinite). This variation in current causes the current in M1 and M2 to change with variations in v_c. Changes in the drain current of M1/M2 (and thus M3/M4) causes the output voltage to change. Figure 22.17 shows how the output voltage changes as a function of the DC common-mode voltage. Understanding this circuit is *very important*. If we wanted to hold the diff-amp output voltage constant while varying the common mode voltage, we would have to apply a small voltage difference between the gates of M1 and M2 (*an offset voltage*). We'll revisit this important topic when we discuss op-amp *CMRR*.

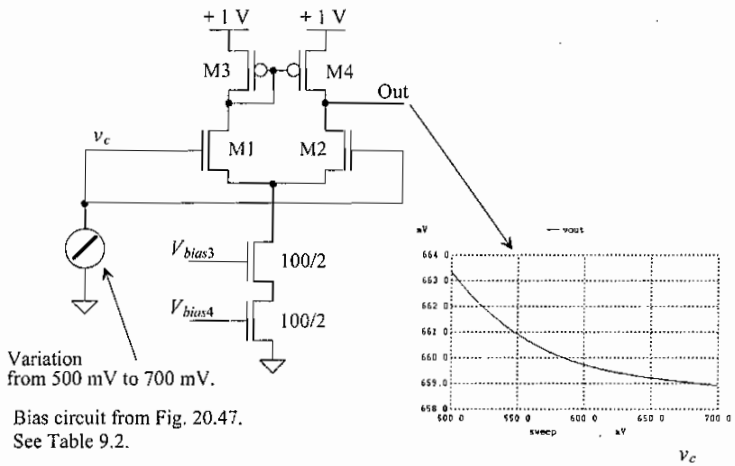
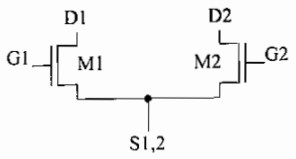
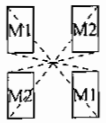


Figure 22.17 How variation in common-mode voltage affects the output voltage.

22.1.4 Matching Considerations

The fact that performance depends on the matching of devices is another important aspect of differential amplifiers. Layout techniques can be used to minimize the first-order effects of mismatch due to oxide gradients and other process variations. Figure 22.18 shows how the diff-pair would be laid out in a *common-centroid* configuration (see Ch. 5). Common-centroid layout, as the name implies, constructs two devices symmetrically about a common center in the layout. This allows the two devices to cancel process gradients in both the x and y directions and exposes both devices to heat sources in an identical manner. Note how we've attempted to match the parasitic capacitances of the pair (on the gates and drains) and how we've minimized the capacitance at the sources (to keep the *CMRR* high at higher frequencies).





Common-centroid layout

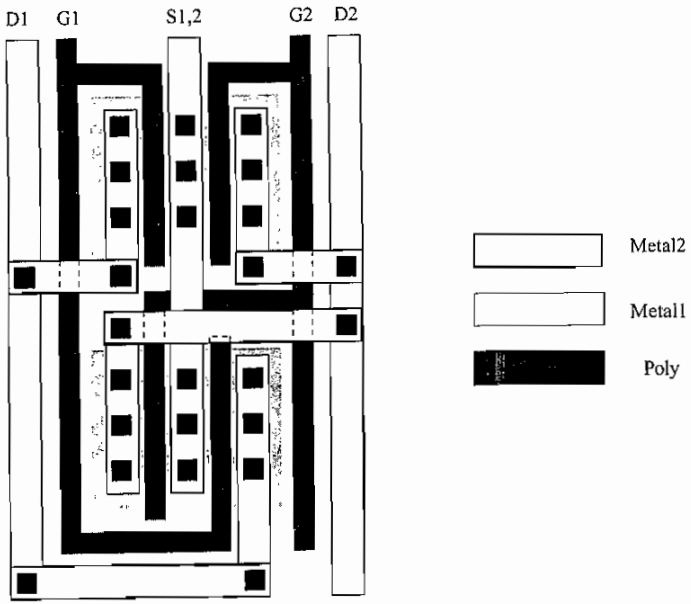


Figure 22.18 Common-centroid layout of a diff-pair.

The input-referred offset voltage of a diff-pair, resulting from mismatches in threshold voltage, geometries, and load resistances, can be determined with the help of Fig. 22.19. When the diff-pair is offset-free, connecting the diff-amp's inputs together causes the drain currents and drain voltages of M1 and M2 to be equal ($V_o = 0$). When the load resistors or M1 and M2 are mismatched, V_o is not zero. (We have mismatch and thus V_o is offset from its ideal value.). To drive V_o to zero, we apply a voltage difference between the gates of M1 and M2 (an input-referred offset).

$$V_{OS} = V_{GS1} - V_{GS2} = V_{THN1} + \sqrt{\frac{2I_{D1}}{\beta_1}} - V_{THN2} - \sqrt{\frac{2I_{D2}}{\beta_2}} \qquad (22.28)$$

We can define the differences and averages of the threshold voltage, load resistance, and geometry as ΔV_{THN} (V_{THN1}–V_{THN2}), V_{THN} (average of V_{THN1} and V_{THN2}), ΔR_L (R_{L1}–R_{L2}), R_L (average of R_{L1} and R_{L2}), $\Delta(W/L)$ [(W_1/L_1)–(W_2/L_2)] and (W/L) [average of (W_1/L_1) and (W_2/L_2)]. Thus, we can make the appropriate substitutions into Eq. (22.28), requiring $I_{D1}R_{L1} = I_{D2}R_{L2}$, with the result:

$$V_{OS} = \Delta V_{THN} + \frac{V_{GS} - V_{THN}}{2} \cdot \left[\frac{-\Delta R_L}{R_L} - \frac{\Delta(W/L)}{(W/L)} \right] \qquad (22.29)$$

The threshold voltage mismatch must be reduced using layout techniques. Mismatches resulting from unequal geometry and differences in the load resistance can be reduced by designing with a small V_{GS} (V_{GS} close to V_{THN}). This should be compared with Eq. (20.8), which shows that using a small V_{GS} in a current mirror results in a large mirrored current difference.

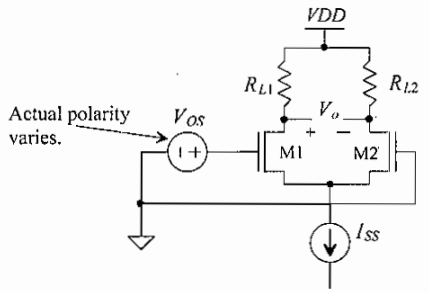


Figure 22.19 Determining diff-amp input offset voltage.

Input-Referred Offset with a Current Mirror Load

Suppose the diff-amp with current mirror load, Fig. 22.12, has an offset (the devices aren't perfectly matched). What this means is that the drain potential of M4 (M2) is not equal to the drain potential of M3 (M1 and most likely the drain currents in M1 and M2 aren't equal as well). The input-referred offset would then be the voltage difference on the gates of M1 and M2 needed to drive the output voltage to the correct value. Using Eq. (22.22), we can write the input-referred offset in terms of the difference-mode gain as

$$|V_{OS}| = \frac{V_{o,ideal}}{|A_d|} = \frac{V_{o,ideal}}{g_m \cdot (r_{o2} || r_{o4})} \tag{22.30}$$

which indicates that to reduce the input-referred offset voltage we simply need to design the diff-amp with large gain. This is the same result as that of Eq. (22.29), if we pause and think about it. To reduce the offset, we said we design with a small V_{GS}. If we hold the diff-amp's bias current constant and reduce V_{GS} (by increasing the widths of M1 and M2), we are increasing g_m.

Finally, although not clearly indicated by Eq. (22.30) as it is in Eq. (22.29), a mismatch in the diff-pair's threshold voltage can only be reduced by layout techniques (e.g., using common-centroid layout). Using large devices (larger-width MOSFETs), reduces variation in device characteristics and leads to better matching.

22.1.5 Noise Performance

The diff-amp with noise sources is seen in Fig. 22.20. The noise from M5 and M6 feeds equally into M1 and M2 and thus, because of the current mirror action in M3 and M4, ideally doesn't affect the output voltage. Therefore, we will ignore M5 and M6 in the following discussion.

The diff-amp's output noise power spectral density (PSD) is given by

$$V_{onoise}^2(f) = \left(I_{M1}^2(f) + I_{M2}^2(f) + I_{M3}^2(f) + I_{M4}^2(f)\right) \cdot (r_{o2} || r_{o4})^2 \tag{22.31}$$

The input-referred noise PSD is then

$$V_{inoise}^2(f) = \frac{V_{inoise}^2(f)}{A_d^2} \tag{22.32}$$

Substituting in the difference-mode gain of the diff-amp, Eq. (22.22), we get

$$V_{inoise}^2(f) = \frac{I_{M1}^2(f) + I_{M2}^2(f) + I_{M3}^2(f) + I_{M4}^2(f)}{g_m^2} \tag{22.33}$$

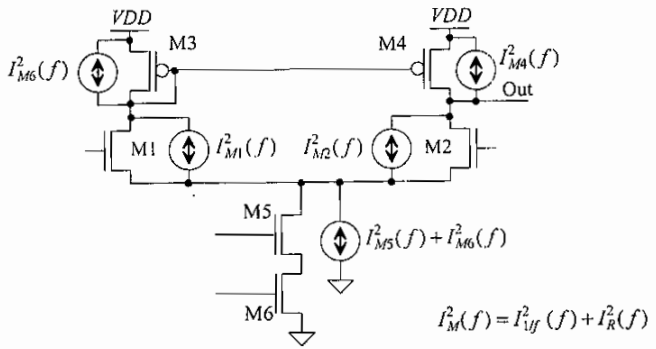


Figure 22.20 Noise model for the diff-amp.

By maximizing the transconductance, g_m, of the diff-pair, we can reduce the input-referred noise. This statement alone is a little misleading. As we saw in Eqs. (9.63)–(9.65), increasing the drain current increases the noise PSD. So to reduce the input-referred noise, we would want to maximize the transconductance by increasing the diff-pair's width (make M1 and M2 wide), not by increasing I_{SS}.

22.1.6 Slew-Rate Limitations

Like the class A amplifiers studied in the last chapter, the basic diff-pair also exhibits slew-rate limitations. As seen in Fig. 22.21, when either M1 or M2 turns off, the total current available to charge a load capacitance is I_{SS}. The slew-rate is then (see also Eq. (21.38))

$$\text{slew rate} = \frac{dV_{out}}{dt} = \frac{I_{SS}}{C_L} \tag{22.34}$$

For high-speed, we need a large bias current. However, this causes excess power dissipation. Once again, to get high-speed (large current drive capability) with low quiescent power dissipation, let's look at a class AB configuration.

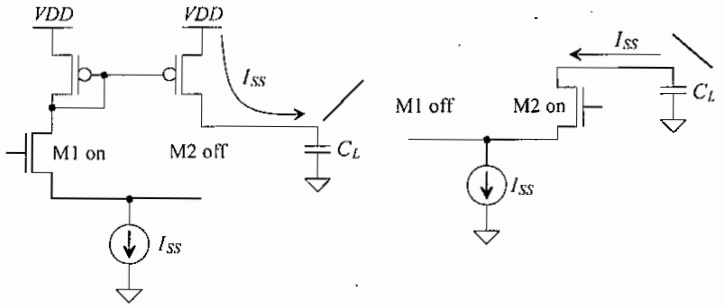


Figure 22.21 Slew-rate limitations in diff-amps.

22.2 The Source Cross-Coupled Pair

The source cross-coupled pair is shown in Fig. 22.22. This configuration is very useful in practical circuit design because it can eliminate slew-rate limitations. A diff-amp built with the source cross-coupled pair can operate in the class AB mode with significant output drive current. Assuming that both inputs to the pair are connected to a (the same) voltage within the pair's common-mode input range and that all NMOS are sized the same and that all PMOS are sized the same, a current I_{SS} flows in all the MOSFETs in the circuit. Note that M11, M21, M31, and M41 are simply behaving as biasing batteries. Their gate-source voltages are mirrored by M1–M4 to set the DC operating current. Using the parameters in Table 9.1 with the gates of M1 and M2 (v_{I1} and v_{I2}) at 3.5 V and I_{SS} of 20 μA, the gate of M3 is at a constant potential that we will label v_{I2}–V_{bias} (or V_{bias} = $V_{GS21} + V_{SG31}$ = 2.2 V neglecting body effect).

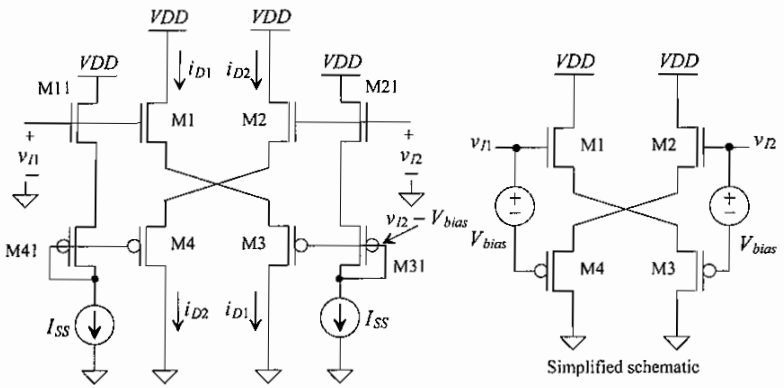


Figure 22.22 Source cross-coupled differential amplifier with NMOS input devices.

Operation of the Diff-Amp

An important characteristic of the source cross-coupled differential amplifier is that as v_{DI} is increased, i_{D1} continues to increase and i_{D2} shuts off, while the opposite is true when v_{DI} decreases. In other words, the amplifier is operating class AB where neither of the output currents is zero as long as their magnitudes remain less than I_{SS}. Looking at the simplified schematic in Fig. 22.22, if v_{I1} increases, then M1 turns on. Because the gate potential of M3 is constant, it turns on as well. In other words, an increase in v_{I1} causes the gate-source voltages of both M1 and M3 to increase (i_{D1} increases). At the same time, the gate potential of M4 increases. This cause both M4 and M2 to shut off (i_{D2} shuts off).

Example 22.8

Simulate the operation of the diff-amp in Fig. 22.22 using the parameters from Table 9.1.

The simulation results are seen in Fig. 22.23. Note that when both inputs are at 3.5 V, the output currents are equal. This figure should be compared to Fig. 22.3.

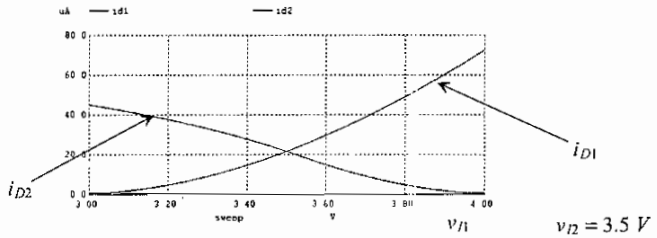


Figure 22.23 The output currents of the diff-amp in Fig. 22.22.

Note how in Fig. 22.3. the output currents flatten out above a maximum input voltage. Here the current continues to increase. ■

After looking at the results of this example, we might wonder: why use such a complicated diff-amp if the output current is only three times more than what we get with the basic diff-amp discussed in the first section? We can always just scale up the current in a basic diff-amp to get more drive. The benefits of the source cross-coupled diff-amp are seen in special-purpose, *low-power* design. In low-power design, the gate source voltages are close to the threshold voltages (and so low power equates to low speed, see Eq. [9.36]). Consider the amplifier and simulation results seen in Fig. 22.24. Here we've increased the widths of the amplifying devices and reduced their bias current (by sizing up the lengths of the biasing current sources as seen in the figure). When the two inputs are the same voltage, the quiescent current that flows in the diff-amp is a few microamps. However, if one input gets a little larger than the other input, the drain-current increases dramatically. In a low-power design, this topology would be useful to quickly charge a capacitor while burning little quiescent power.

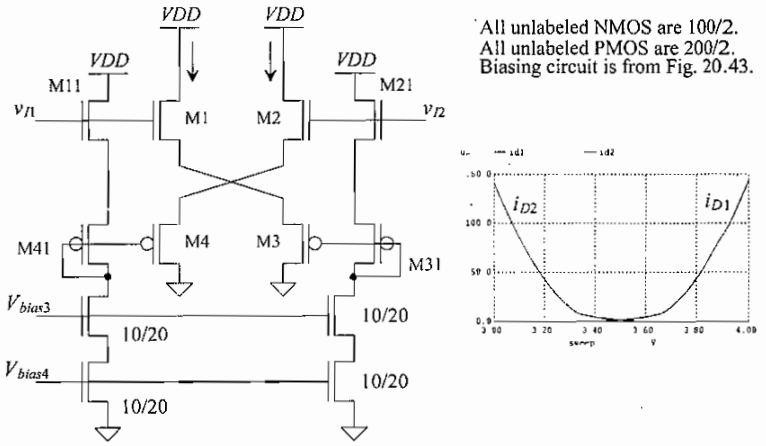


Figure 22.24 Lowering the power in the source cross-coupled diff-amp.

Input Signal Range

The drawback of the source cross-coupled diff-amp is the limited input range. The PMOS version of the source cross-coupled differential amplifier is shown in Fig. 22.25. Normally the version (PMOS or NMOS) of this differential amplifier depends on the input signal range. The NMOS version has the best positive input signal range, while the PMOS version has the best negative input signal range. For centering the input signal range the topology seen in Fig. 22.26 can be used. Again, both an NMOS V_{GS} and a PMOS V_{SG} are used to bias the cross-coupled diff-amp. Note that the biasing devices

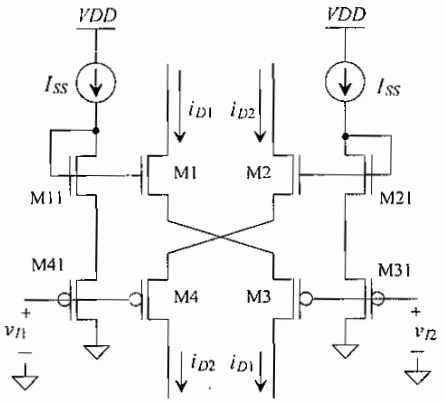


Figure 22.25 Source cross-coupled differential amplifier, PMOS.

(M11, M21, M31, and M41) experience body effect differently than M1–M4. This causes the currents in M1–M4 to be less than the current in the biasing devices. If possible, place M1–M4 and the biasing devices in their own wells to eliminate the body effect. Note that when very low currents are used, the constant current sources driving the gates of M1–M4 can cause slew-rate limitations. To eliminate this problem, capacitors can be placed between the gates of M1/M4 and M2/M3 to make the biasing appear more battery-like.

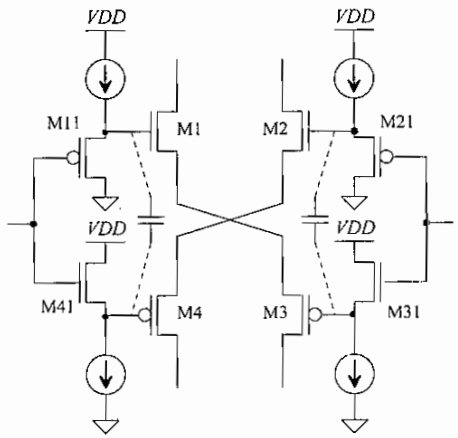


Figure 22.26 Biasing scheme used to shift input common-mode range. Note that the body effect affects the biasing, causing the current in M1–M4 to decrease.

22.2.1 Current Source Load

The source cross-coupled differential amplifier with active loads is shown in Fig. 22.27. Note that we could have also added a pair of active loads in series with the drains of M1 and M4 to obtain a differential output amplifier (two complementary amplifier outputs). Superposition can be used to find the AC small-signal gain from the output back to each input. First, determine the small-signal gain, $\frac{v_o}{v_{i1}}$, with v_{i2} at AC ground. Neglecting the bulk effect, we can see that the voltage gain from the gate of M1 to the source of M1 is simply a common drain amplifier of the form

$$\frac{v_{s1}}{v_{i1}} = \frac{g_{m1}R_{Leq}}{1+g_{m1}R_{Leq}} \text{ V/V} \tag{22.35}$$

where R_{Leq} is simply the load seen by the source of M1. This can be found by using a test source (as seen in Fig. 22.28) to determine the effective impedance seen looking into the source of M3. This results in

$$R_{Leq} = \frac{v_t}{i_t} = \frac{1+\frac{1}{g_{m3}r_{o3}}}{g_{m3}+\frac{1}{r_{o3}}} \approx \frac{1}{g_{m3}} \tag{22.36}$$

Therefore, the voltage gain from v_{i1} to the source of M1 becomes

$$\frac{v_{s1}}{v_{i1}} = \frac{g_{m1}\frac{1}{g_{m3}}}{1+g_{m1}\frac{1}{g_{m3}}} \tag{22.37}$$

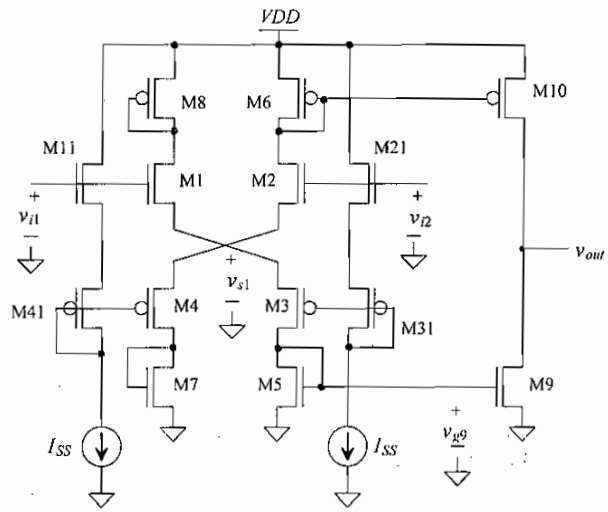


Figure 22.27 Source cross-coupled differential amplifier, with current source loads.

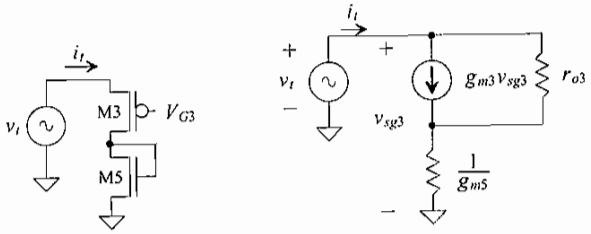


Figure 22.28 (a) Simplified schematic and (b) small-signal model.

The voltage gain from v_{s1} to v_{g9} is the same form as a common gate amplifier. This gain is

$$\frac{v_{g9}}{v_{s1}} = g_{m3}\frac{1}{g_{m5}} \tag{22.38}$$

And finally, the gain from the gate of M9 to the output is simply $-g_{m9}(r_{o9}||r_{o10})$, and

$$\frac{v_o}{v_{i1}} = -\frac{g_{m1}\frac{1}{g_{m5}}}{1+g_{m1}\frac{1}{gm3}}g_{m9}(r_{o9}||r_{o10}) \text{ for } v_{i2}=0 \tag{22.39}$$

Using the same type of analysis, we find that the expression from v_{i2} to the output is

$$\frac{v_o}{v_{i2}} = \frac{g_{m2}\frac{1}{g_{m6}}}{1+g_{m2}\frac{1}{gm4}}g_{m10}(r_{o9}||r_{o10}) \text{ for } v_{i1}=0 \tag{22.40}$$

Superposition can now be used to determine the total effect of each input on the output. Therefore, if M1 and M2 are matched, as are M3 and M4, and if M5 and M6 are designed so that their g_ms are equal as are M7 and M8 (respectively), then the differential gain can be expressed as

$$v_o = (v_{i2}-v_{i1})\frac{2\cdot g_{m1,2}\frac{1}{g_{m5,6}}}{1+g_{m1,2}\frac{1}{gm3,4}}g_{m9,10}(r_{o9}||r_{o10}) \Rightarrow \frac{v_o}{v_{di}} = \frac{2\cdot g_{m1,2}\frac{1}{g_{m5,6}}}{1+g_{m1,2}\frac{1}{gm3,4}}g_{m9,10}(r_{o9}||r_{o10}) \tag{22.41}$$

Input Signal Range

The positive-input, common-mode range (CMR) is determined in exactly the same manner as in the source coupled diff-amp of the first section. The positive CMR is given by

$$v_{CMMAX} \approx VDD - \sqrt{\frac{2I_{SS}}{\beta_6}} = VDD - V_{DS,sat} \tag{22.42}$$

while the negative CMR is given by

$$v_{CMMIN} = \overbrace{\sqrt{\frac{2I_{SS}}{\beta_1}}}^{V_{GS1}} + V_{THN} + \overbrace{\sqrt{\frac{2I_{SS}}{\beta_3}}}^{V_{DS3}} + \overbrace{\sqrt{\frac{2I_{SS}}{\beta_5}}}^{V_{GS5}} + V_{THN} \tag{22.43}$$

An alternative current source (current mirror) load configuration is shown in Fig 22.29. The gain of this configuration is given by

$$A_d = \frac{v_{out}}{v_{i1}-v_{i2}} = 2\cdot\frac{r_{o2}||r_{o4}}{\frac{1}{g_{m1}}+\frac{1}{g_{m3}}} \tag{22.44}$$

assuming $g_{m1} = g_{m2}$ and $g_{m3} = g_{m4}$.

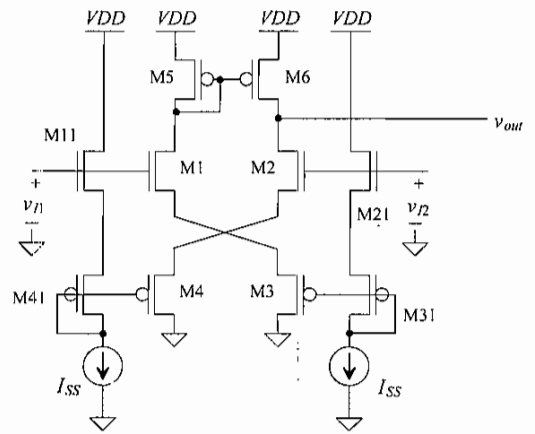


Figure 22.29 Alternative current source load configuration.

22.3 Cascode Loads (The Telescopic Diff-Amp)

The differential amplifier with current source load of Fig. 22.6 gave a voltage gain of $g_m(r_{o2}||r_{o4})$. For many applications, this gain may be too low. Using a cascode current source load in place of M3/M4, results in a gain of approximately $g_m r_{o2}$ (the output resistance of the load becomes much larger than the resistance looking into the drain of M2). This modest improvement in gain can be increased by also cascoding M1 and M2. Figure 22.30 shows the resulting circuit configuration (sometimes called a *telescopic diff-amp*). The MOSFET M6 is selected so that its V_{GS} keeps M1/M2 and MC1/MC2 in the saturation region. The gain of this configuration is given by

$$A_d = g_{m1}\cdot(R_{into\,DC2}||R_{into\,DC4}) \tag{22.45}$$

The resistance looking into the drain of MC2, assuming M2 and MC2 are the same size, is given by

$$R_{into\,DC2} \approx g_{m2}\cdot r_{o2}^2 \tag{22.46}$$

and the resistance looking into the drain of MC4, again assuming MC4 and M4 are the same size, is given by

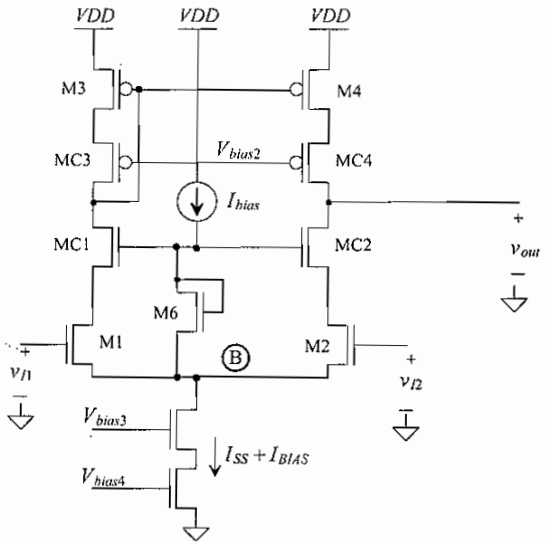


Figure 22.30 Cascode differential amplifier.

$$R_{into\ DC4} \approx g_{m4} \cdot r_{o4}^2 \tag{22.47}$$

The gain of the cascode differential amplifier may be written as

$$A_d = g_{m1}\left(g_{m2}r_{o2}^2 \| g_{m4}r_{o4}^2\right) \tag{22.48}$$

The main drawback of using the cascode differential amplifier is the reduction in positive common-mode range. In the negative direction, V_{CMMIN} is identical to the source-coupled diff-amp discussed in Sec. 22.1. The positive common-mode signal is limited by the additional circuitry required by the cascode loads. Notice that V_{CMMAX} in this case is limited by the amount of voltage necessary to keep M1, MC1, MC3, and M3 in their respective saturation regions.

An interesting issue concerning this circuit is the DC feedback through M6. Because the current through M6 is constant, V_{GS6} will be constant. Normally, an increase in the common-mode voltage on the gates of M1 and M2 causes their drain voltages to decrease. However, because the currents through M1 and M2 are constant, the common source node, marked as node B in Fig. 22.30, increases with the increasing input voltage. This causes the gate potential of MC1 and MC2 to increase. The result is that the drain voltages of M1 and M2 go up (since the gate-source voltages of MC1 and MC2 are constant). This feedback action attempts to keep M1 and M2 from going into nonsaturation. The maximum common-mode voltage, V_{CMMAX}, will be limited by the output voltage and by MC1/MC2 going into nonsaturation.

Example 22.9

Determine the minimum and maximum input common-mode voltage for the diff-amp seen in Fig. 22.31. Note that the voltage across M6 is $V_{GS}+V_{DS,sat}$, where V_{GS} is the gate-source voltage of the other NMOS devices in this figure (see Fig. 20.32 and the associated discussion).

Parameters from Table 9.1
Unlabeled PMOS are 30/2.
Unlabeled NMOS are 10/2.
Bias circuit from Fig. 20.43.

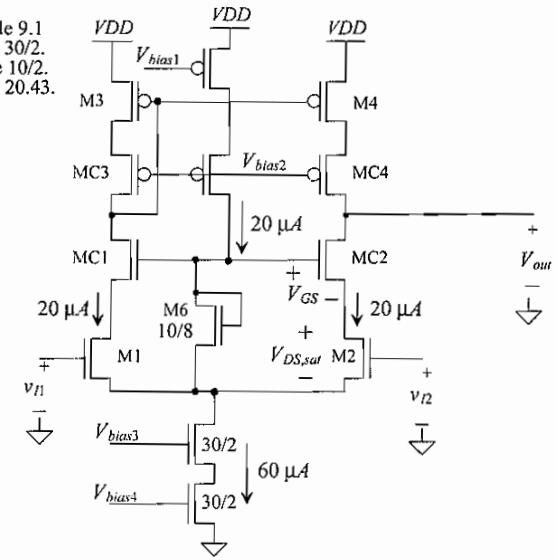


Figure 22.31 Cascode differential amplifier.

The minimum input common-mode voltage is given by

$$V_{CMMIN} = V_{GS1,2} + 2V_{DS,sat} = 1.05 + 0.5 = 1.55\ V$$

As mentioned a moment ago, MC1 and MC2 must be kept in the saturation region. If we assume that the drains of MC1 and MC2 are at $VDD - V_{SG}$ (= 3.85 V), then when MC1 and MC2 are on the verge of trioding (their drain-source voltages are $V_{DS,sat}$), the drains of M1 and M2 are at

$$V_{D1,2} = VDD - V_{SG} - V_{DS,sat} = 3.65\ V$$

We know for an NMOS device (M1 or M2) to be in saturation

$$V_D \geq V_G - V_{THN} \text{ and so } V_{D1,2} \geq V_{CMMAX} - V_{THN}$$

giving a $V_{CMMAX} = 4.45\ V$. In terms of an arbitrary output voltage, we can write

$$V_{CMMAX} = V_{out} - V_{DS,sat} + V_{THN}$$

so if V_{out} is 2.45 V, then $V_{CMMAX} = 3\ V$. ■

22.4 Wide-Swing Differential Amplifiers

An increasingly important concern with the advent of low-voltage design is the need for improved common-mode range. A technique for extending the allowable input swing of a diff-amp is to use two complementary diff-amp stages in parallel, as shown in Fig. 22.32. To understand the operation of this circuit, consider the case when the DC component of the input signal, v_{in}, is such that both diff-amps are on (and the AC component is small compared to the DC). The current through the diff-pairs M1/M2 and M9/M10 is I, while the current through the summing MOSFETs M4 and M12 is $2I$. If M5 is the same size as M4 and if M7 is the same size as M12, then a current $2I$ flows in the output transistors. The small-signal voltage gain, assuming $g_{m1} = g_{m2}$ and $g_{m9} = g_{m10}$, is given by

$$A_v = (g_{m1} + g_{m9})[r_{o7}(2I) \| r_{o5}(2I)] = \frac{(g_{m1} + g_{m9})}{\lambda_7 2I + \lambda_5 2I} = \frac{\sqrt{2\beta_1 I} + \sqrt{2\beta_9 I}}{2I(\lambda_7 + \lambda_5)} \qquad (22.49)$$

If the input is such that the p-channel diff-amp is on and the n-channel diff-amp is off, then a current I flows in the summing MOSFETs M4 and M12 and zero current flows in M1, M2, M3, and M6. The current that flows in M5 and M7 is now I. The small-signal voltage gain is

$$A_v = g_{m9}[r_{o7}(I) \| r_{o5}(I)] = \frac{\sqrt{2\beta_9 I}}{I(\lambda_7 + \lambda_5)} \qquad (22.50)$$

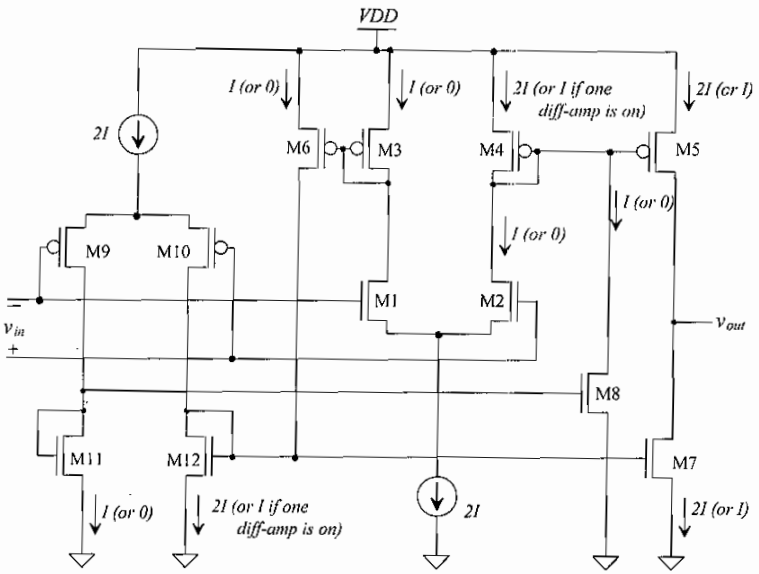


Figure 22.32 Two parallel differential amplifiers used to increase input swing.

The small-signal gain when the p-channel diff-amp is off and the n-channel diff-amp on is given by

$$A_v = g_{m1}[r_{o7}(I) \| r_{o5}(I)] = \frac{\sqrt{2\beta_1 I}}{I(\lambda_7 + \lambda_5)} \qquad (22.51)$$

It is desirable to transition smoothly from having a single diff-amp on to having both diff-amps on. If we require

$$\beta_1 = \beta_9 = \beta \Rightarrow G_m = \sqrt{2\beta I} \qquad (22.52)$$

then Eqs. (22.49) through (22.51) can be rewritten as

$$A_v = G_m \cdot [r_{o7}(I) \| r_{o5}(I)] \qquad (22.53)$$

For low distortion, a constant gain is important. Also, for a stable amplifier, it is important that the amplifier can be compensated to ensure stability. An amplifier whose gain depends on the input signal amplitude may be more challenging to compensate for.

22.4.1 Current Differential Amplifier

Another wide-swing, differential amplifier is the current differencing amplifier. The current diff-amp is shown schematically in Fig. 22.33. In the following discussion, let's assume that M1 through M4 are the same size. If both i_1 and i_2 are zero, then a current I_{SS} flows in all MOSFETs in the circuit. Now assume that i_1 is increased above zero, but less than I_{SS}. This causes the current in both M1 and M2 to increase. As a result, the drain current in M3 decreases to keep $i_{D2} + i_{D3} = 2I_{SS}$. The decrease in M3 drain current is mirrored in M4, forcing the current i_1 out of the diff-amp. A similar argument can be made for increasing i_2; that is, it causes the output of the differential amplifier to sink a current equal to i_2. The sizes of M1–M4 can be ratioed to give the diff-amp a gain or to scale the input currents.

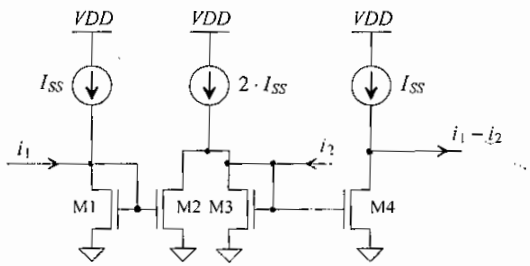


Figure 22.33 Current differential amplifier.

The input impedance of the current differential amplifier is simply the small-signal resistance of a diode-connected MOSFET, or

$$R_{in} = \frac{1}{g_m} \qquad (22.54)$$

This configuration finds applications in both low-power and high-speed circuit design.

22.4.2 Constant Transconductance Diff-Amp

A rail-to-rail differential amplifier made using n- and p-channel diff-amps is shown in Fig. 22.34. It is desirable, for good distortion and proper compensation, that the overall transconductance of the diff-amp remain constant independent of the region of operation, that is, operation with both n- and p-channel diff-amps on or only a single diff-amp on. A constant g_m is guaranteed over the input range if

$$g_m = g_{mn} + g_{mp} = \sqrt{2\beta_n I_n} + \sqrt{2\beta_p I_p} = \text{constant} \qquad (22.55)$$

where g_{mn} and g_{mp} are the transconductances of the n- and p-channel diff-amps and g_m is the overall transconductance of the input stage. Since β_n and β_p are constant and can be made equal, Eq. (22.55) can be rewritten as (assuming both diff-amps are operating),

$$\sqrt{I_n} + \sqrt{I_p} = \text{constant} \qquad (22.56)$$

This equation always holds if both differential amplifiers are on. The problem with nonconstant transconductance occurs if only one diff-amp is on. If, for example, the common-mode input (common input voltage on the + and – inputs of the differential amplifier) is large enough to shut the p-channel diff-amp off, then $I_p = 0$ and the transconductance of the overall input diff-amp changes.

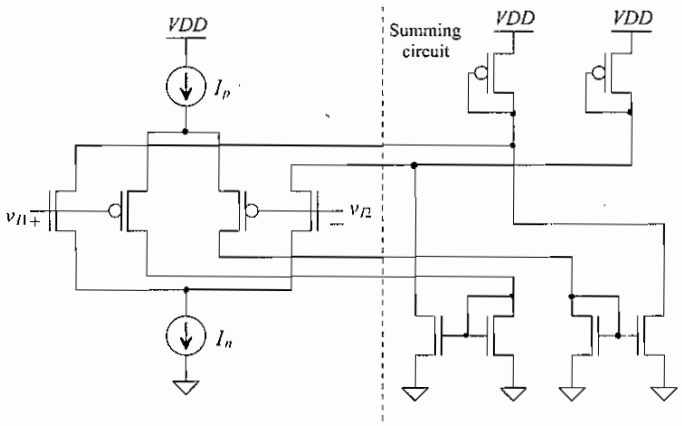


Figure 22.34 Rail-to-rail differential amplifier.

The solution to the problem of non-constant g_m begins by making

$$I_n = I_p = I_o \qquad (22.57)$$

When both diff-amps are on (keeping in mind that we have already set the requirement that $\beta_n = \beta_p$), Eq. (22.56) reduces to

$$2\sqrt{I_o} = \text{constant} \qquad (22.58)$$

If we add a current of $3I_o$ to I_n (or to I_p) when the p-channel diff-pair (n-channel diff-pair) is off, the transconductance of the pair is constant, or

$$\overbrace{2\sqrt{I_o}}^{\text{both on}} = \overbrace{\sqrt{3I_o + I_n}}^{\text{n diff-amp on}} = \overbrace{\sqrt{3I_o + I_p}}^{\text{p diff-amp on}} \text{ if } I_o = I_p = I_n \qquad (22.59)$$

An example of a constant-g_m, rail-to-rail input stage is shown in Fig. 22.35. The summing circuit of Fig. 22.34 is not shown in this figure. MOSFETs M1–M4 make up the p- and n-channel diff-pairs, while MP1 and MN1 source the constant current I_o when both diff-pairs are on. (MP1 and MN1 represent the constant current sources shown in Fig. 22.28.) When both diff-pairs are on, the current out of each current diff-amp is approximately 0. (The drain current of M6 and M7 is approximately 0.) If the common-mode input voltage becomes large enough to shut the p-channel diff-pair off, then MOSFETs MS1 and MS2 are off as well. This causes the current in M5 to become I_o. The current in M6 mirrors the current in M5. Since M6 is three times larger than M5, the current in M6 becomes $3I_o$.

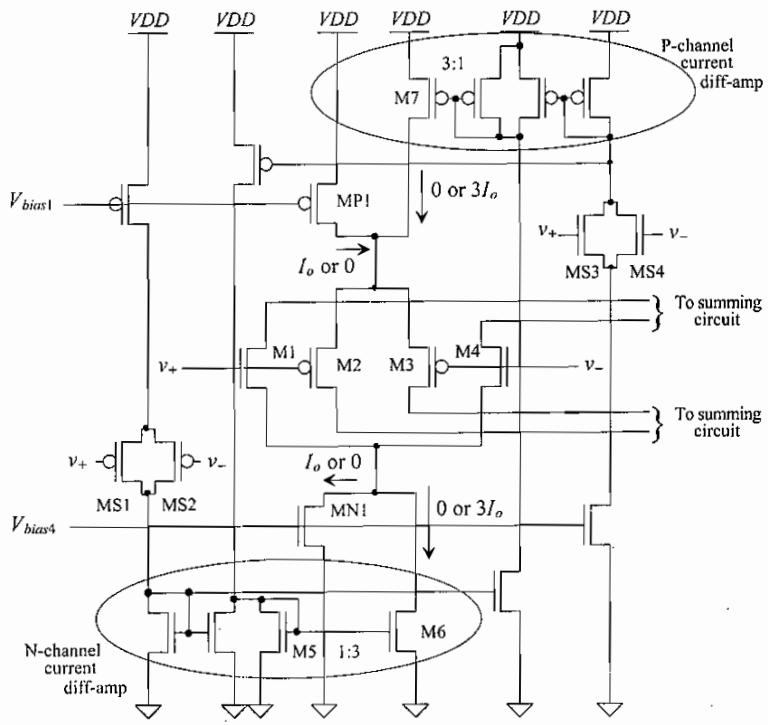


Figure 22.35 A wide-swing diff-amp with constant gm.

Discussion

In general, the constant-g_m diff-amp is used in an op-amp to prevent overcompensation of the op-amp and to avoid distortion when the op-amp is used with large signals where the diff-amps, on the input of the op-amp, are turning on and off with variations in the input signal. Several practical problems exist with these uses of the constant-g_m diff-amp. Because the value of g_m may vary with shifts in the process, we may still overcompensate an op-amp design. Using the constant-g_m stage to avoid overcompensation is useful, but since we may overcompensate (or undercompensate) the op-amp with process variations, the added complexity and power draw may not justify the additional circuitry. Using the constant-g_m stage to avoid distortion is based on keeping the unity gain frequency of an op-amp, f_{un}, a constant independent of the common-mode input voltage. In practice, mismatches (threshold voltage and geometry) in the input diff-pair and changes in DC biasing conditions, resulting from the diff-amps turning off and on, can cause distortion. In some cases, this distortion can be worse than the distortion resulting from the nonconstant f_{un} (which we get with the nonconstant g_m diff-amp). The distortion resulting from mismatches can be modeled as an offset-voltage that is dependent on the input common-mode voltage. Since the DC currents sourced or sunk from a constant-g_m diff-amp are not constant, the DC operating point of the circuit summing the currents changes with the input signal's amplitude. The result is a change in the low-frequency gain and added distortion.

ADDITIONAL READING

[1] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*, 2nd ed., Oxford University Press, 2002. ISBN 0-19-511644-5.

[2] P. R. Gray, P. J. Hurst, and S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits,* 4th ed., Wiley, 2001. ISBN 0-47-132168-0.

[3] R. Castello and P. R. Gray, "A High-Performance Micropower Switched-Capacitor Filter," *IEEE Journal of Solid State Circuits,* vol. SC-20, no. 6, pp. 1122–1132, December, 1985.

[4] E. Seevinck and R. F. Wassenaar, "A Versatile CMOS Linear Transconductor/Square-Law Function Circuit," *IEEE Journal of Solid State Circuits,* vol. SC-22, no. 3, pp. 366–377, June 1987.

[5] M. Steyaert and W. Sansen, "A High-Dynamic-Range CMOS Op Amp with Low-Distortion Output Structure," *IEEE Journal of Solid State Circuits,* vol. SC-22, no. 6, pp. 1204–1207, December 1987.

[6] T. S. Fiez, H. C. Yang, J. J. Yang, C. Yu, and D. J. Allstot, "A Family of High-Swing CMOS Operational Amplifiers," *IEEE Journal of Solid State Circuits,* vol. 22, no. 6, pp. 1683–1687, December 1989.

[7] J. H. Botma, R. F. Wassenaar, and R. J. Wiegerink, "A Low-Voltage CMOS Op Amp with a Rail-to-Rail Constant-g_m Input Stage and a Class AB Rail-to-Rail Output Stage," *Proceedings of the 1993 IEEE ISCAS,* p. 1314.

[8] A. L. Coban, P. E. Allen, and X. Shi, "Low-Voltage Analog IC Design in CMOS Technology," *IEEE Transactions on Circuits and Systems,* vol. 42, no. 11, November 1995.

PROBLEMS

22.1 Determine the drain current of M1 as a function of the input voltage, $v_{I1} - v_{I2}$, for the diff-amp shown in Fig. 22.36. Neglect body effect. What does your derivation give for i_{D1} when v_{I1} is much larger than v_{I2}?

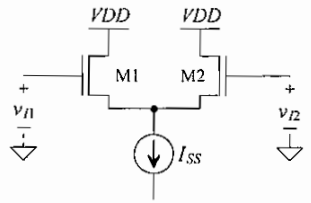


Figure 22.36 Diff-amp used in Problem 22.1

22.2 Repeat Ex. 22.1 if the widths of M1 and M2 are increased from 10 to 100. Determine the transconductance of the diff-amp. Write i_{o2} as a product of g_m (= $g_{m1} = g_{m2}$) and v_{I1} (with v_{I2} = AC ground), v_{I2} (with v_{I1} = AC ground), and $v_{I1} - v_{I2}$.

22.3 Determine the maximum and minimum common mode voltages for the PMOS version of the diff-amp seen in Fig. 22.4.

22.4 Determine the AC currents flowing in the circuit of Fig. 22.5 if the gate of M1 is grounded and the gate of M2 is an AC signal of 1 mV. Verify your answers with an AC SPICE simulation.

22.5 Determine the small-signal gain and the input common-mode-range (CMR) for the diff-amps shown in Fig. 22.37. Verify your answers with SPICE.

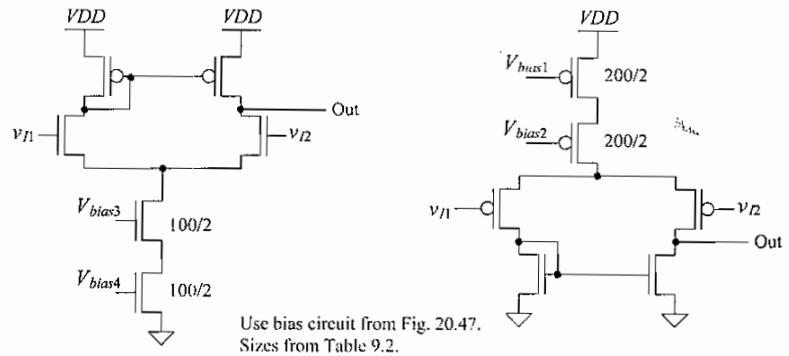


Figure 22.37 Diff-amps for Problem 22.5.

22.6 Show that the capacitance on the sources of M1/M2 in Ex. 22.6 causes the *CMRR* to roll off quicker with increasing frequency. (Add a capacitance to the sources of M1/M2 in a simulation and show that *CMRR* decreases at a lower frequency.)

22.7 Estimate the slew-rate limitations in charging and discharging a 1 pF capacitor tied to the outputs of the diff-amps shown in Fig. 22.37. Verify with SPICE.

22.8 For the n-channel diff-pair shown in Fig. 22.38, show that the following relationships are valid if the body effect is included in the analysis of the transconductance.

$$i_{d1} = \frac{g_m}{2}\left[v_{i1}\left(2 - \frac{g_m}{g_m + g_{mb}}\right) - v_{i2} \cdot \frac{g_m}{g_m + g_{mb}} \right] - \frac{g_m \cdot g_{mb}}{g_m + g_{mb}} \cdot \frac{[v_{i1} + v_{i2}]}{2}$$

and

$$i_{d2} = \frac{g_m}{2}\left[v_{i2}\left(2 - \frac{g_m}{g_m + g_{mb}}\right) - v_{i1} \cdot \frac{g_m}{g_m + g_{mb}} \right] - \frac{g_m \cdot g_{mb}}{g_m + g_{mb}} \cdot \frac{[v_{i1} + v_{i2}]}{2}$$

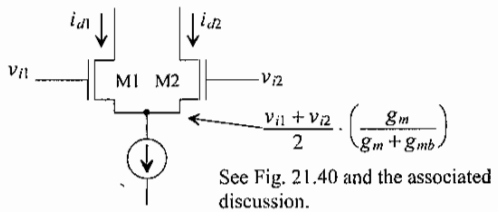


Figure 22.38 How body effect changes the AC behavior of the diff-amp.

22.9 The diff-amp configuration shown in Fig. 22.39 is useful in situations where a truly differential output signal is needed. Determine the following: (a) the transconductance of the diff-amp, (b) the AC small-signal drain currents of all MOSFETs in terms of the input voltages and g_{mn} (the transconductance of an n-channel MOSFET), and (c) the small-signal voltage gain, $(v_{O+} - v_{O-})/(v_{I+} - v_{I-})$. Verify your answers with SPICE.

22.10 Using the diff-amp topology in Fig. 22.24 and a current mirror load, show how quickly (or slowly) the diff-amp can drive a 1 pF load. Does this diff-amp exhibit slew-rate limitations? Verify your answers with SPICE.

22.11 Using the topology seen in Fig. 22.26 with the long-channel process (Table 9.1) without body effect (all MOSFET's bodies tied to their respective sources), show that the currents in M1–M4 match the biasing currents used in the source followers. Show how the currents become mismatched when the bodies of the NMOS are tied to ground and the bodies of PMOS are tied to *VDD*.

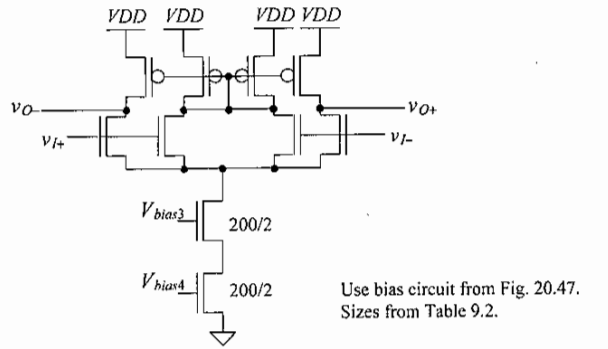


Figure 22.39 Fully differential diff-amp used in Problem 22.9.

22.12 Simulate the operation of the diff-amp seen in Fig. 22.31. Use a common-mode voltage of 2.5 V and an AC input voltage of 100 μV at 1 kHz. Compare the simulation results to your hand calculations.

22.13 Using SPICE with current sources for inputs, show the operation of the diff-amp in Fig. 22.33.

22.14 The circuit seen in Fig. 22.40 is an another example of a circuit that sums the currents from NMOS and PMOS diff-amps (so the input common-mode range can extend from beyond the power supply rails). Describe and simulate the operation of this circuit using the short-channel parameters in Table 9.2 and the bias circuit from Fig. 20.47.

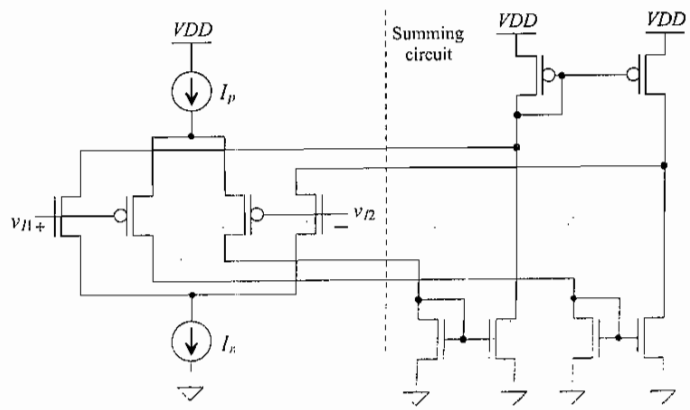


Figure 22.40 Summing circuit for Problem 22.14.

22.15 Using symbolic analysis, determine the small-signal gain of the self-biased diff-amp seen in Fig. 22.41 (see also Fig. 18.17).

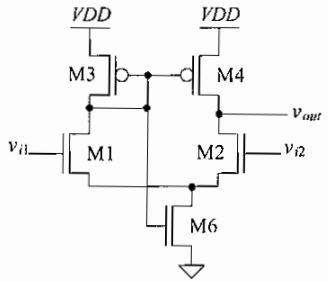


Figure 22.41 A self-biased NMOS diff-amp.

Chapter
23

Voltage References

A general-use (ideal) voltage reference is a circuit used to generate a fixed voltage, V_{REF}, that is independent of the power supply voltage VDD (where $V_{REF} < VDD$), temperature, and process variations. In other words, the ideal reference voltage is independent of PVT. In some cases, we want to design a reference that varies with temperature. For example, if V_{REF} increases with temperature, Fig. 23.1a, we say that the reference voltage is *proportional to absolute temperature* or PTAT. If the reference voltage decreases with increasing temperature, Fig. 23.1b, the reference is said to be *complementary to absolute temperature* or CTAT. The PTAT and CTAT references can be used to design a voltage reference that changes very little with temperature called a *bandgap* reference. Unfortunately, the generation of PTAT and CTAT reference voltages requires using parasitic diodes. In a CMOS process, the electrical characteristics of the parasitic pn junctions are not monitored and controlled (like, say, the threshold voltage) during manufacturing. Therefore, if possible, the implementation of a reference voltage using MOSFET-resistor circuits is desirable.

This chapter is split into two sections. The first section covers the design of voltage references using MOSFETs and resistors, while the second section covers the design of voltage references using parasitic diodes. The first section also offers some overview material common to all voltage references.

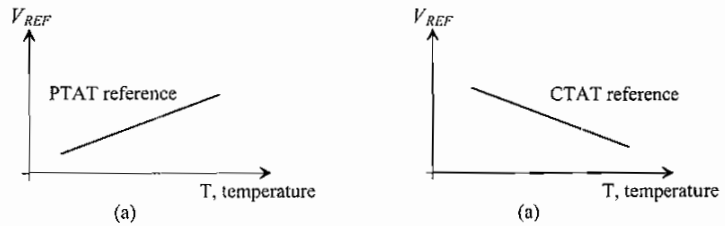


Figure 23.1 (a) PTAT and (b) CTAT voltage references.

23.1 MOSFET-Resistor Voltage References

We can derive reference voltages from the power supplies using the resistor and the MOSFET, as seen in Fig. 23.2. The voltage divider formed with two resistors has the advantage of simplicity, temperature insensitivity (as was shown in Ch. 7), and process insensitivity; that is, changes in the sheet resistance have no effect on the voltage division. The main problem with this circuit is that in order to reduce the power dissipation (i.e., the current through the resistors), the resistors must be made large. Since large resistors require a large area on the die, this voltage divider may not be practical in many cases. One situation where we will use this simple voltage divider is in generating a voltage halfway between *VDD* and ground, *VDD*/2, (sometimes called the *common-mode* voltage of an analog circuit or system).

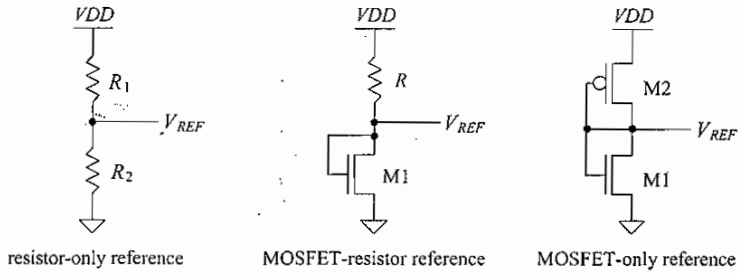


Figure 23.2 Voltage dividers implemented in CMOS.

The voltage divider formed between the resistor and the MOSFET can be recognized as the same circuit we used for a bias in the current mirror back in Ch. 20 (see Fig. 20.2 and Eqs. (20.26)–(20.31)). The final reference, a voltage divider between NMOS and PMOS devices, has the advantage that the layout can be small (see Fig. 20.13 and Eqs. (20.12)–(20.19)). In the following two subsections, we analyze the behavior of these last two voltage dividers.

23.1.1 The Resistor-MOSFET Divider

The reference voltage used in the resistor-MOSFET divider is equal to the V_{GS} of the MOSFET. We can write for this circuit

$$I_D = \frac{VDD - V_{REF}}{R} = \frac{\beta_1}{2}(V_{REF} - V_{THN})^2 \qquad (23.1)$$

or

$$V_{REF} = V_{THN} + \sqrt{\frac{2I_D}{\beta_1}} = V_{THN} + \sqrt{\frac{2(VDD - V_{REF})}{R \cdot \beta_1}} \qquad (23.2)$$

If V_{REF} is designed so that it is close to V_{THN}, then the reference voltage will be insensitive to changes in *VDD* and its temperature behavior will follow the threshold voltage (see

Eqs. (9.43)–(9.48)). However, for the general case, the temperature coefficient of the resistor-MOSFET voltage divider is determined using

$$TCV_{REF} = \frac{1}{V_{REF}} \cdot \frac{\partial V_{REF}}{\partial T} \qquad (23.3)$$

and if we assume $VDD >> V_{REF}$, then

$$TCV_{REF} = \frac{1}{V_{REF}}\left[V_{THN} \cdot TCV_{THN} - \frac{1}{2}\sqrt{\frac{2L_1}{W_1} \cdot \frac{VDD}{R \cdot KP(T)}} \cdot \left[\frac{1}{R}\frac{\partial R}{\partial T} - \frac{1.5}{T}\right]\right] \qquad (23.4)$$

Example 23.1

Estimate the temperature performance of the resistor-MOSFET voltage references seen in Fig. 23.3. Assume that the temperature coefficient of the resistor is 2,000 *ppm/C*, the long-channel process is used where $KP_n = 120\ \mu A/V^2$, and the nominal *VDD* is 5 V. Use simulations to verify the answers. Also show how the reference voltages change with *VDD*.

Figure 23.3 Resistor-MOSFET references used in Ex. 23.1

For the reference made using the 1MEG resistor, Fig. 23.3a, the current that flows in the circuit is

$$I = \frac{VDD - V_{REF}}{10^6} = \frac{KP_n}{2} \cdot \frac{W}{L}(V_{REF} - V_{THN})^2$$

which can be solved to determine I is around 4 μA and

$$V_{REF} = V_{GS} \approx 900\ mV$$

From Sec. 9.1.3 the rate the threshold voltage changes with temperature is

$$\frac{\partial V_{THN}}{\partial T} \approx -1\ mV/C^\circ \approx \frac{\partial V_{REF}}{\partial T}$$

The temperature coefficient of the reference voltage is

$$TCV_{REF} = \frac{1}{V_{REF}}\frac{\partial V_{REF}}{\partial T} = \frac{-0.001}{0.9} = -1,111\ ppm/C^\circ \qquad (23.5)$$

and thus

$$V_{REF}(T) = V_{REF} \cdot (1 + TCV_{REF}(T - T_0)) = 0.9 \cdot (1 - 0.00111(T - 25))) \qquad (23.6)$$

where we assume that the threshold voltage was measured at 25 °C. The simulation results are seen in Fig. 23.4. Note that at higher temperatures the threshold voltage decreases and thus so does V_{REF}. Further note that a change in temperature of 25 °C corresponds to a change in the reference voltage of 31.25 mV (= 25 · 0.00125).

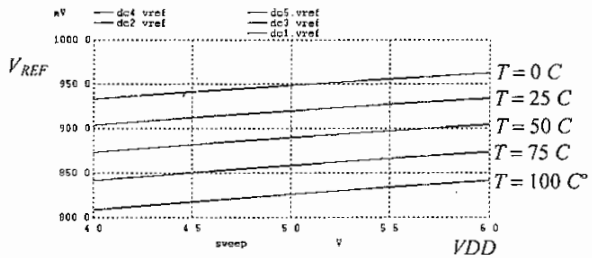


Figure 23.4 How the reference voltage changes with VDD and temperature for the reference in Fig. 23.3a.

To estimate the temperature behavior of the circuit in Fig. 23.3b, we must first determine the value of V_{REF}. Using Eq. (23.2), we can write

$$V_{REF} = V_{THN} + \sqrt{\frac{2(VDD - V_{REF})}{R \cdot \beta_1}} = 0.8 + \sqrt{\frac{2(5 - V_{REF})}{10k \cdot \frac{10}{2} \cdot 120\mu A/V^2}}$$

After a few iterations, we can determine $V_{REF} \approx 1.85\ V$. The temperature coefficient is calculated using Eq. (23.4)

$$TCV_{REF} = \frac{1}{1.85} \cdot \left[-1\ mV/C^\circ - \frac{1}{2}\sqrt{\frac{2 \cdot 2}{10}} \cdot \frac{5}{10k \cdot 120\ \mu A/V^2} \cdot \left[0.002 - \frac{1.5}{300} \right] \right]$$

which evaluates to $TCV_{REF} = 500\ ppm/C$. The simulation results are seen in Fig. 23.5. Note how, when comparing Figs. 23.4 and 23.5, one reference circuit performs well with regard to temperature (23.3b), while the other reference circuit (23.3a) performs well with regard to *VDD* variations. ■

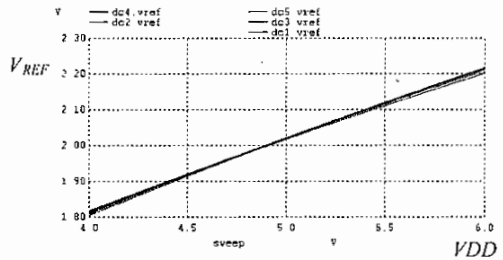


Figure 23.5 How the reference voltage changes with VDD and temperature for the reference in Fig. 23.3b.

A modification to the basic resistor-MOSFET divider is shown in Fig. 23.6. The reference voltage in this circuit is given by

$$V_{REF} = V_{GS}\left(\frac{R_1}{R_2} + 1\right) \tag{23.7}$$

noting the ratio of the resistors (and so their temperature behavior or sheet resistance shift with process variations) doesn't affect the reference voltage. When V_{GS} is designed to be approximately V_{THN}, the resulting reference circuit is said to be a threshold-voltage multiplier reference circuit.

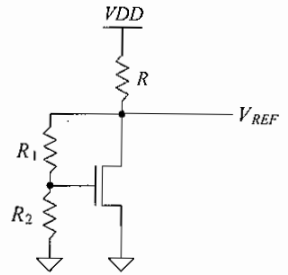


Figure 23.6 Modification of the resistor-MOSFET voltage divider.

23.1.2 The MOSFET-Only Voltage Divider

The MOSFET-only voltage divider shown in Fig. 23.2 generates a reference voltage that is equal to the voltage on the gates of the MOSFETs with respect to ground. Since $I_{D1} = I_{D2}$, we can write

$$\frac{\beta_1}{2}(V_{REF} - VSS - V_{THN})^2 = \frac{\beta_2}{2}(VDD - V_{REF} - V_{THP})^2 \tag{23.8}.$$

or the reference voltage is given by

$$V_{REF} = \frac{VDD - V_{THP} + \sqrt{\frac{1}{\beta_2}} \cdot V_{THN}}{\sqrt{\frac{\beta_1}{\beta_2}} + 1} \tag{23.9}$$

or knowing the desired reference voltage and the power supply voltage

$$\frac{\beta_1}{\beta_2} = \left[\frac{VDD - V_{REF} - V_{THP}}{V_{REF} - V_{THN}} \right]^2 \tag{23.10}$$

The temperature dependence of the MOSFET-only voltage divider, assuming the temperature dependence of the ratio of the transconductance parameters, $\frac{\beta_1}{\beta_2}$, is negligible, is given by

$$TCV_{REF} = \frac{1}{V_{REF}} \cdot \frac{\partial V_{REF}}{\partial T} = \frac{1}{V_{REF}} \cdot \frac{1}{\sqrt{\frac{\beta_1}{\beta_2}} + 1} \cdot \left[\frac{\partial(-V_{THP})}{\partial T} + \sqrt{\frac{\beta_1}{\beta_2}} \frac{\partial V_{THN}}{\partial T} \right] \tag{23.11}$$

From Ch. 9, we know for the long-channel devices

$$\frac{\partial V_{THN}}{\partial T} = -1\ mV/C^{\circ} \tag{23.12}$$

and

$$-\frac{\partial V_{THP}}{\partial T} = 1.4\ mV/C^{\circ} \tag{23.13}$$

To achieve $TCV_{REF} = 0$. requires

$$-\frac{\partial V_{THP}}{\partial T} = -\sqrt{\frac{\beta_1}{\beta_2}} \cdot \frac{\partial V_{THN}}{\partial T} \Rightarrow 1.4\ mV/C^{\circ} = \sqrt{\frac{\beta_1}{\beta_2}} \cdot 1\ mV/C^{\circ} \tag{23.14}$$

or

$$\sqrt{\frac{\beta_1}{\beta_2}} = 1.4 \tag{23.15}$$

Zero temperature coefficient, to a first order, can be met by satisfying this equation. However, this ratio is most often set by the desired V_{REF}.

23.1.3 Self-Biased Voltage References

We've already presented a voltage reference back in Ch. 20 using the beta-multiplier reference (BMR) (see Figs. 20.15, 20.19, and 20.22). Consider the BMR seen in Fig. 23.7. The added amplifier forces the drain/gates of M1 and M2 to the same potential. Because the gates and sources of M3/M4 are at the same potential, the same current is forced through each side of the reference. *This is an important common theme for all of the self-biased references discussed in this chapter.* We'll discuss this in greater detail in a moment. Before moving on to this topic, let's discuss why we connect the inverting and noninverting inputs to the added amplifier in the way seen in Fig. 23.7. (Why is M2 connected to the + amplifier input instead of the – input?) For the amplifier to be stable, we want the amount of signal fed back and subtracted from the input to be larger than the

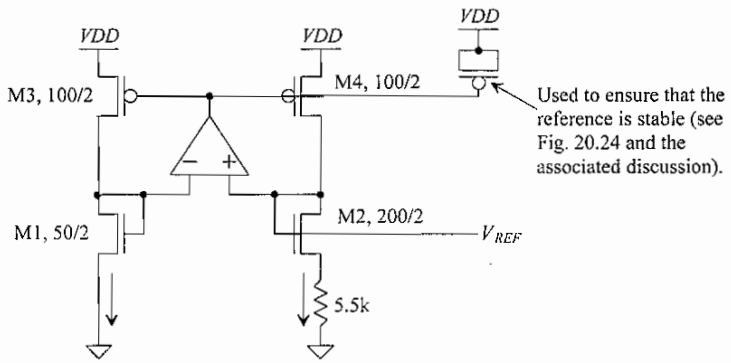


Figure 23.7 BMR from Figs. 20.19 and 20.22.

amount of signal fed back and added to the input. Because M3 and M4 are *inverting* common-source amplifiers, we want the signal fed back to the + input of the amplifier to be larger than the signal fed back to the – input of the amplifier. Since the (AC) currents flowing in M3 and M4 are equal and the small-signal resistance of M2 + R is larger than the small-signal resistance of M1, we connect M2 to the + amplifier input.

Forcing the Same Current through Each Side of the Reference

The simplest method to force the same current through each side of a reference is to use a simple current mirror, Fig. 23.8a. As we saw in Fig. 20.15, the low-output resistance found in short-channel devices makes the resulting reference very sensitive to changes in *VDD*. By cascoding the current mirror, Fig. 23.8b, the currents can be made more equal and the sensitivity to *VDD* can be reduced. The problem with using a cascode current mirror is that the minimum allowable *VDD* increases. Consider the following.

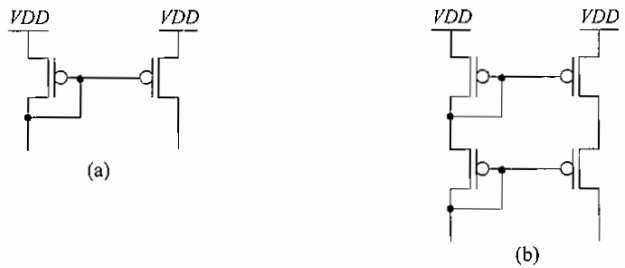


Figure 23.8 Using a current mirror to force the same current through each side of a reference.

Example 23.2

Resimulate the BMR in Fig. 20.18 using a cascode current mirror for the PMOS devices as seen in Fig. 23.9.

Using the cascode current mirror, we expect the currents through each branch of the reference to be very nearly equal. As seen in the simulation results in Fig. 23.10, the currents *are* equal. However, they are not constant and independent of changes in *VDD*. Similarly, if we were to use V_{biasn} as a reference voltage (see Sec. 20.1.5), then V_{biasn} would show a large sensitivity to changes in *VDD*. The problem with cascoding only the PMOS devices is that while the currents through each branch are equal (but not constant), the variation with *VDD* is still present. The voltages at the drains of M1 and M2 change with *VDD*. To reduce this voltage variation, we might consider cascoding the NMOS devices as well (see Fig. 23.11). The PMOS devices are still used to force the same current through each side of the reference, while now the NMOS cascode stack is used to keep the voltages across M1 and M2 constant with changes in *VDD*. As the simulation results, Fig. 23.12, show, the currents do stabilize but at a voltage 20% higher than *VDD*. Clearly, cascoding devices won't be useful in a short-channel CMOS process (and so *we'll stick with using the added amplifier to both set the currents*

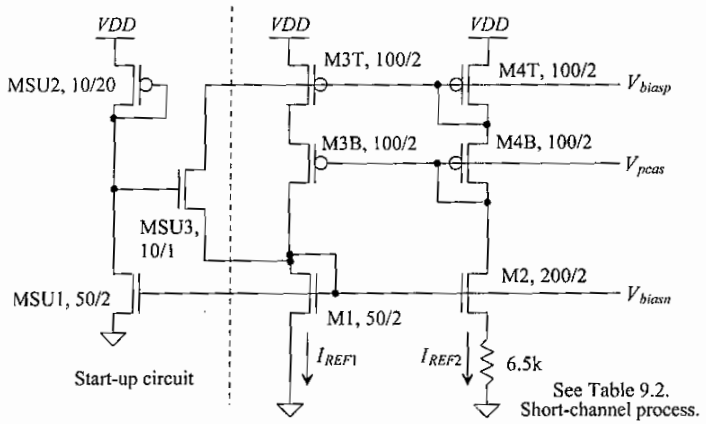


Figure 23.9 Cascoding the PMOS devices in the BMR circuit.

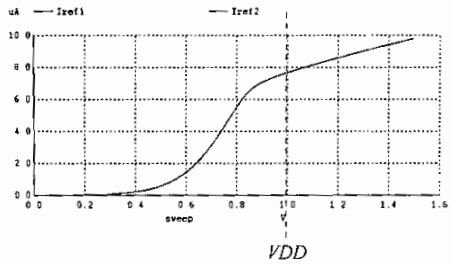


Figure 23.10 Simulating the behavior of the currents in the BMR of Fig. 23.9.

and hold the voltage across M1 and M2 constant in the remainder of the ***short-channel CMOS*** *designs presented in this chapter*). Note that cascoding devices can be useful when designing in long-channel CMOS processes. In these processes, the threshold voltage is a smaller percentage of the power supply voltage. For example, in our short-channel process, Table 9.2, this percentage is 28%. In the long-channel process, Table 9.1, it is 16%. Also note that using NMOS cascodes alone won't make the reference more tolerant to changes in *VDD*. The NMOS cascodes will keep the voltages across M1 and M2 constant (but not equal). Since the currents through each branch aren't equal (because the PMOS devices aren't cascoded), we still get significant sensitivity to *VDD*.

Finally, we need to, again (see Fig. 20.15 and the associated discussion), mention the importance of the start-up circuit. The start-up circuit is often overlooked and is often a cause of problems in practical designs. We include a start-up circuit in every self-biased reference to avoid the situation where zero

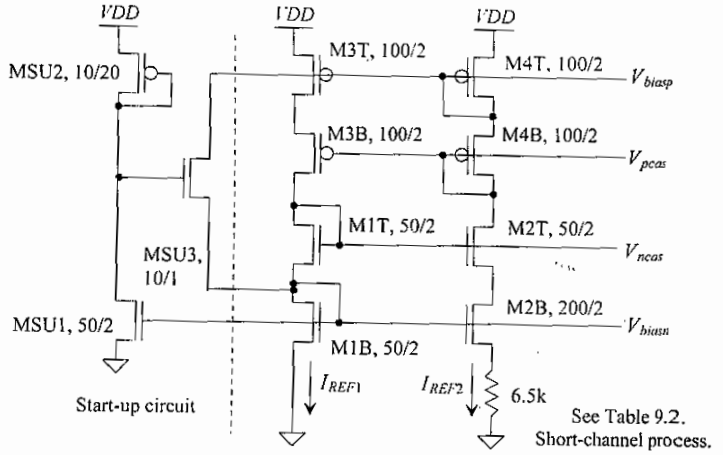


Figure 23.11 Cascoding both NMOS and PMOS devices in the BMR circuit.

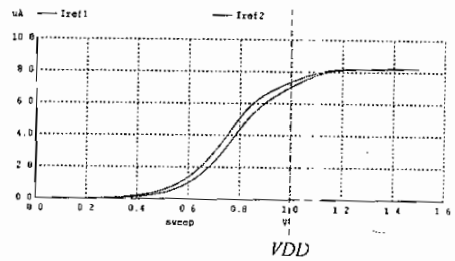


Figure 23.12 Simulating the behavior of the currents in the BMR of Fig. 23.11.

current flows in the reference. It's important to ensure that the start-up circuit doesn't affect normal operation or draw too much current from *VDD*. For the start-up circuit seen in Figs. 23.9 and 23.11, we use the V_{biasn} to set the current drawn by the bias circuit. In normal operation, MSU3 should be off (it should have a negative V_{GS}). ■

Example 23.3

Using the topology seen in Fig. 20.22, design a voltage reference with, ideally, zero temperature coefficient. Simulate the design to determine the reference's sensitivity to changes in temperature and *VDD*. Comment on the repeatability of the V_{REF} (with process variations from one process run to the next) and methods (and concerns) with trimming V_{REF}. Assume $TCR = 0.002$ (2,000 ppm/C).

We begin by writing the long-channel equation, Eq. (20.40), for the resistance needed for zero temperature coefficient (ZTC)

$$R = \frac{2}{\frac{\partial V_{THN}}{\partial T} \cdot KP_n \cdot \frac{W}{L}} \left(1 - \frac{1}{\sqrt{K}}\right) \cdot \left(\frac{1}{R}\frac{\partial R}{\partial T} + \frac{1}{KP_n} \cdot \frac{\partial KP_n}{\partial T}\right) \quad (23.16)$$

While this equation won't directly apply for the design in a short-channel process, we can use it, with simulations, to help point us to the factors that influence the temperature behavior of the reference. Substituting in the appropriate numbers with $K = 4$, we get

$$R = \frac{1}{(-0.0006) \cdot KP_n \cdot \frac{W}{L}} \cdot \left((0.002) - \frac{1.5}{300}\right) \quad (23.17)$$

or

$$R = \frac{5}{KP_n \cdot \frac{W}{L}} \quad (23.18)$$

To move the reference towards a ZTC, let's set the resistor to nominally 5.5k (the same value we used before, see Fig. 20.22 and the associated discussion) and adjust the widths and lengths of the NMOS devices until the simulations show good temperature behavior (keeping in mind that we must keep the W/L of M2 K times the W/L of M1). The resulting circuit is seen in Fig. 23.13. Simulation results are seen in Fig. 23.14. (Note that the currents and gate-source voltages have nothing to do with the values listed in Table 9.2.) The reference voltage is nominally 500 mV (= *VDD*/2). The reference, according to simulations, moves 40 mV over a 100 °C temperature change (–400 μV/C or a TC of –800 ppm/C, not that great when compared to the references described later). Further, the reference is fairly insensitive to variations in *VDD* once *VDD* gets above 600 mV.

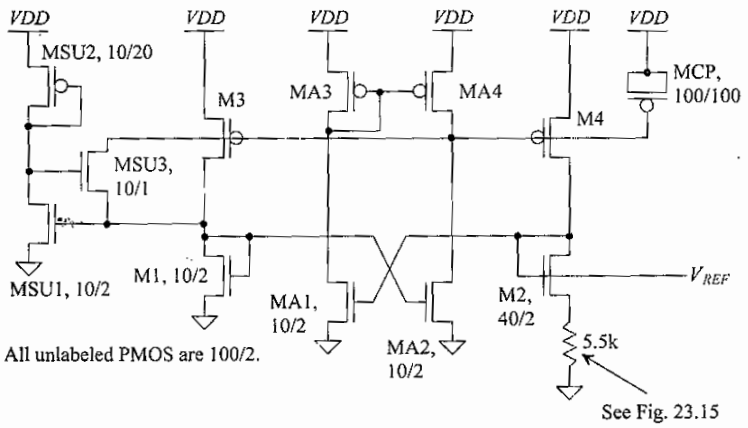


Figure 23.13 Voltage reference using the beta-multiplier.

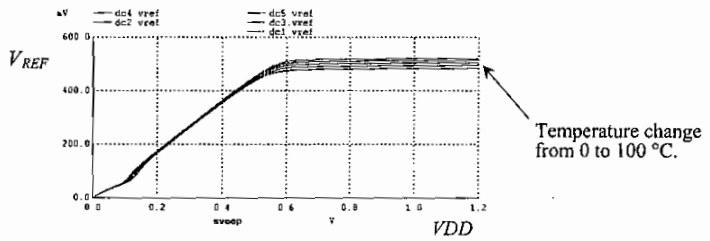


Figure 23.14 Simulating the operation of the reference in Fig. 23.13.

A couple of practical notes are needed at this point. While we've used simulations to zero in on a design with decent temperature behavior and a reference voltage at *VDD*/2, actually fabricating this reference would result in a V_{REF} different from the one seen in Fig. 23.14 (in most fabrication runs). The MOSFET characteristics and the sheet resistance vary with each process run.

To adjust the reference voltage to *VDD*/2, the resistor must be trimmed using fuses, Fig. 23.15. It can be shown that the temperature behavior doesn't vary significantly with small changes in V_{REF} when trimming. As seen in Eq. (20.38), lowering the resistor value causes V_{REF} to increase, while increasing the resistor value causes V_{REF} to decrease. The fuses in Fig. 23.15 short across the resistor until they are blown. To trim the resistor, we start blowing the resistors (electrically or with a laser). With each blown fuse we add (nominally) 200 Ω in series with the nominally 4k resistor. If the processes' sheet resistance increases by 20% (so that the resistors are now 4.8k and 240 Ω), then we only need to blow three fuses (neglecting the change in the MOSFET characteristics) to trim the resistor to 5.5k. If the sheet resistance decreases by 20%, then the resistors are 3.2k and 160 Ω. Fifteen fuses would need to be blown to trim the resistor to 5.5k.

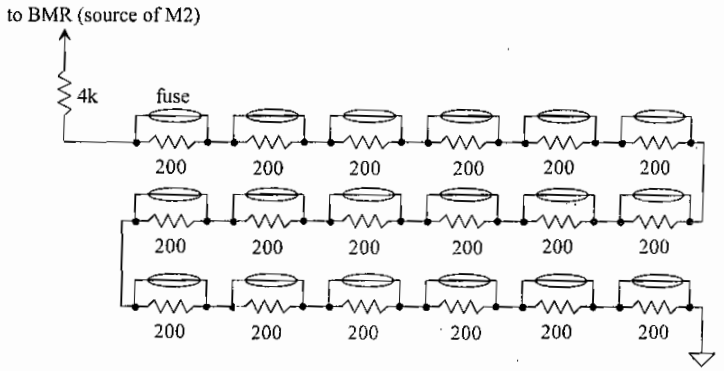
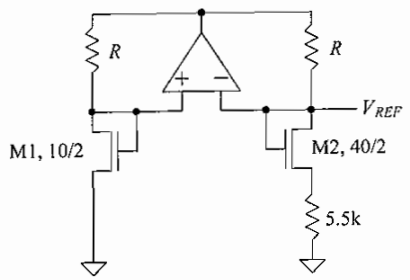


Figure 23.15 Trimming a resistor with fuses.

Finally, it's important to remember that stability is of great importance when using feedback. As discussed in Sec. 20.1.4, the feedback loop is made stable by adding a capacitance on the output of the added amplifier. The bigger this capacitance, the more stable the reference. We stabilize the reference in Fig. 23.13 with the addition of MCP. ■

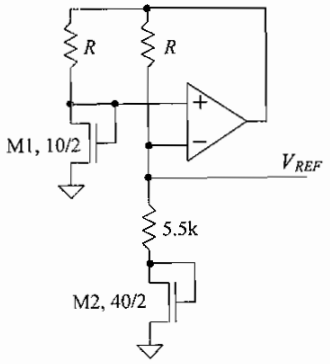
An Alternate Topology

Instead of using PMOS devices in the top of each branch of the reference, we might try holding the voltage across resistors constant to force the current through each branch of the reference to be the same. Figure 23.16 shows the idea drawn two different ways. In (a) the amplifier tries to hold both of its inputs at the same potential. This causes the voltage across each R to be the same and thus each branch of the reference has the same current. Note now we've tied M2 and the resistor (a resistance larger than M1 alone) to the – amplifier input. This is because the inverting PMOS common-source amplifiers (M3 and M4) are no longer in the feedback signal path; we need to ensure that the largest signal is fed back to the inverting amplifier terminal. Note also, in (b), that the series order of the resistor and MOSFET is not important. In fact, having both sources connected to ground eliminates the body-effect mismatch and may result in better temperature performance. We won't use these topologies in any of the MOSFET-resistor topologies in this chapter because the amplifier has to drive a resistive load (which, as we saw in Chs. 22 and 23, kills the amplifier's gain). A two-stage op-amp, discussed in the next chapter, can be used. However, then we have to be concerned with compensating the op-amp. Finally, note that a start-up circuit is still needed with these topologies. The start-up circuit would leak current into the node connected to the + amplifier input to start the reference circuit up (and then shut off after the reference turns on). Connecting the start-up circuit to the – amplifier input would move the amplifier output in the wrong direction and the reference would never start up (noting in Figs. 23.7 and 23.13 that we do connect the start-up circuit to the – input because M3 and M4 are inverting amplifiers).



(a) Holding the voltage across resistors to force the same current through each branch of the reference.

Note start-up circuit not shown (see text).



(b) Redrawing (a) and switching the places of the resistor and M2.

Figure 23.16 Other references useful in MOSFET-resistor circuits.

23.2 Parasitic Diode-Based References

By using a junction diode, the variability encountered in the references of the last section that used a MOSFET's threshold voltage can be reduced. One practical implementation of a parasitic diode available in CMOS is seen in Fig. 23.17. A diode is formed between the p+ implant and the n-well. However, in a practical circuit the parasitic vertical PNP bipolar device formed by the p+, n-well, and p-substrate causes current to be injected into the substrate. Good guard rings are used to surround the p+ in both the n-well and the p-substrate to collect this current. This ensures that the injected carriers are collected and not causing substrate current to flow in other portions of the chip. Because the substrate is connected to ground, the diode's cathode (K) must be tied to ground. We might try to use an n+ implant directly in the substrate for diode formation. However, assuming the substrate (which forms the anode of the resulting diode) is grounded, we would have to apply a negative voltage to the n+ (K) to forward-bias the diode. One other practical parasitic diode available in CMOS is made using the lateral PNP device seen in Fig. 23.18. In this device, the p+ forms the emitter/collector of the resulting device (the n-well is the base). However, the vertical PNP (and the resulting substrate current) is still present. A significant portion of the lateral PNP's emitter current will flow into the substrate. In the following example, we'll assume vertical PNP (diodes) are used.

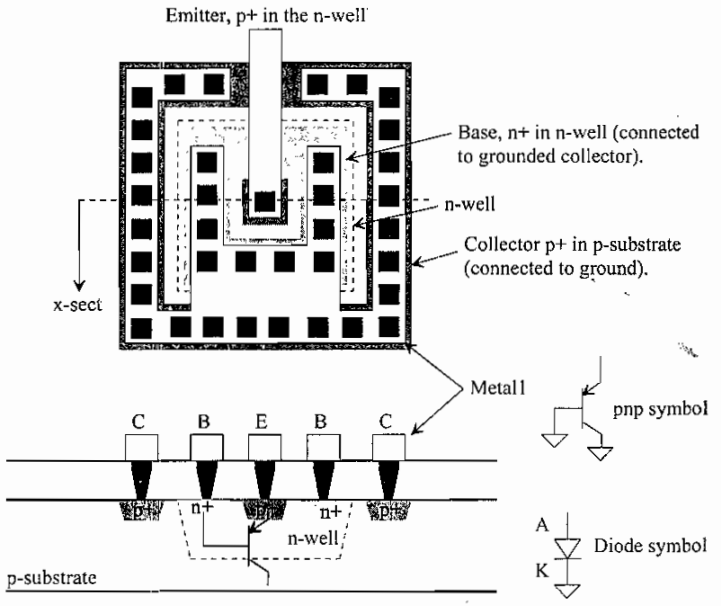


Figure 23.17 A vertical parasitic pnp bipolar junction transistor used as a diode.

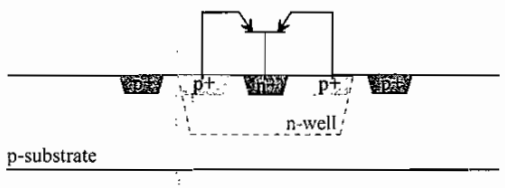


Figure 23.18 A lateral parasitic bipolar junction transistor.

Diode Behavior

It's important to realize that the parasitic diode's behavior must be thoroughly characterized in a particular CMOS process to determine its voltage-current (and temperature) characteristics. For the sake of illustrating design procedures, considerations, and examples in this chapter, we can develop a general diode model. The current through a forward-biased diode is given by

$$I_D = I_S \cdot e^{V_D/n \cdot V_T} \qquad (23.19)$$

where V_D is the voltage across the diode, n is the emission coefficient (used to shape the diode's current-voltage curve to fit experimental data), V_T is the thermal voltage (kT/q or 26 mV at room temperature), and I_S is the diode's scale current. Here, we'll use a value of 10^{-18} A for the scale current and an n of 1. The SPICE statement corresponding to the schematic seen in Fig. 23.19 is

```
D1    1    0    PNPDIODE
D2    2    0    PNPDIODE    4
```

where the 4 indicates 4 diodes are connected in parallel. We may use, in schematics drawn in later in the chapter, the term K to indicate K diodes are connected in parallel. The model statement for the diode, using the parameters seen above, is

```
.MODEL      PNPDIODE    D      is=1e-18      n=1
```

Figure 23.20 shows how the voltage across the diode changes if the current through the diode is held constant and the temperature is changed. Using our model (n =1 and I_S =

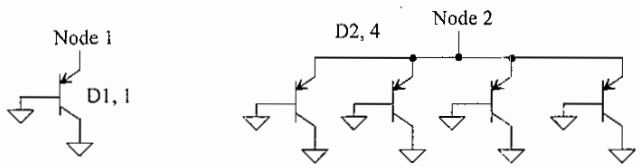


Figure 23.19 A schematic of diodes.

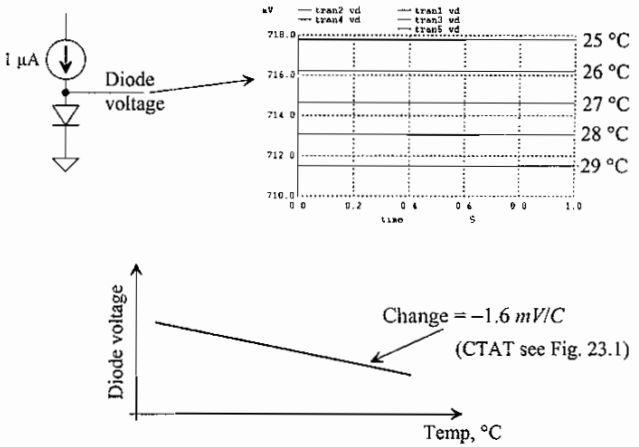


Figure 23.20 Change in diode voltage with temperature.

10^{-18}), we can estimate the change in the diode's voltage with temperature, from the simulation data, as

$$\frac{dV_D}{dT} = -1.6\ mV/C \qquad (23.20)$$

noting that this number is very dependent on the DC current through the diode. (Also note this voltage change is complementary to absolute temperature, CTAT.) Increasing the bias current in Fig. 23.20 to 10 µA, causes the diode's voltage drop to increase (of course) and its voltage change with temperature to become −1.4 *mV/C*.

The Bandgap Energy of Silicon

The silicon bandgap energy as a function of temperature is given by

$$E_g(T) = 1.16 - (702 \times 10^{-6}) \cdot \frac{T^2}{T + 1108}\ (\text{eV}) \qquad (23.21)$$

At room temperature, the bandgap of silicon is approximately 1.1 eV (where 1 electron-volt, eV, is 1.6×10^{-19} J). Looking at Eq. (23.21), notice that the bandgap energy decreases with increasing temperature. If we put a constant current through a diode and increase the temperature, the barrier height between the n and p sides of the diode decreases and thus so does the diode's voltage drop (and that is why the diode's voltage change with temperature in Fig. 23.20 is negative).

In the next two sections we'll talk about *BandGap References* (BGR). BGR's combine the CTAT behavior of the bandgap of silicon (the diode's forward voltage drop) with the PTAT behavior of the thermal voltage (the thermal voltage, kT/q , increases with increasing temperature see Fig. 23.1) to form a voltage reference (a BGR) that doesn't vary (much) with temperature.

Lower Voltage Reference Design

While, at the time of this writing, power supply voltages are well above the forward turn-on voltage of the pn junction diode (parasitic PNP device), it is clear that the near future will bring *VDD* voltages below 0.7 V. To implement a reference in these lower supply voltage processes, consider the Schottky diode layout of Fig. 23.21. In this layout, we connect the metal (the anode) directly to the n-well (that is, without the p+) to form the Schottky diode. To connect the metal directly to the substrate, we need an opening in the oxide. To form this opening, we use the "active" layer without the select layer (without the implant). It's important that no select layer surrounds this Schottky contact.

The turn-on voltage of a Schottky contact is considerably lower than a standard pn junction. In Fig. 23.21 we draw the turn on voltage as 300 mV. In real silicon this voltage will depend highly on the doping of the n-well (which, as we saw in Ch. 6, scales up as process dimensions shrink). The design discussions and procedures developed in the next two sections can be applied to Schottky diode-based references. Again, though, thorough characterization of the parasitic diode is required prior to designing the reference. One concern, among others, is the repeatability of the diode's current-voltage characteristics from one process run to the next. We don't discuss Schottky diode-based references any further in this chapter, but, rather, leave it to the refereed literature.

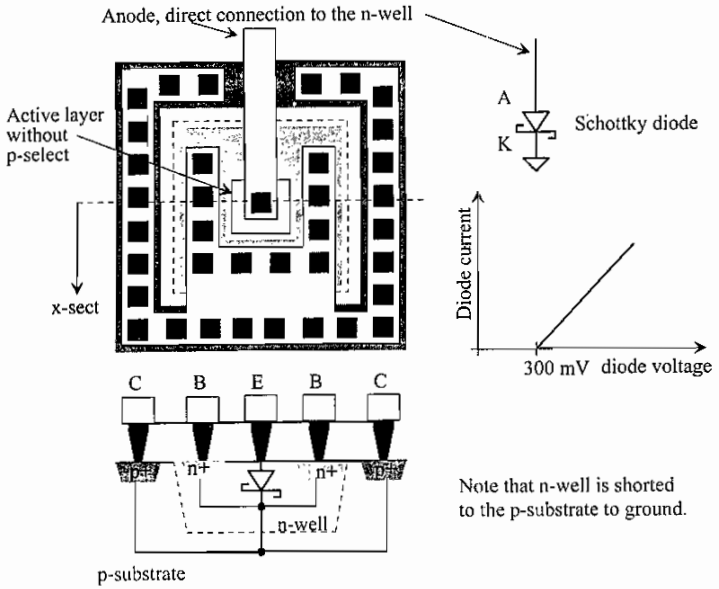


Figure 23.21 Connecting metal directly to n-well to make a Schottky diode.

23.2.1 Long-Channel BGR Design

In this section we present some long-channel designs for bandgap references (BGR) using the cascode structure seen in Fig. 23.11. We will not show the **(required)** start-up circuit in the following schematics.

Diode-Referenced Self-Biasing (CTAT)

A diode-referenced, self-biasing circuit is seen in Fig. 23.22. The cascode structures force the same current through each branch of the reference. The voltage across the resistor now equals the forward voltage drop of the diode. The current in the circuit (the current flowing in each branch of the reference) is then

$$I_{REF} = \frac{V_D}{R} = I_S e^{V_D/nV_T} \tag{23.22}$$

Solving for a resistor to set the reference current level gives

$$R = \frac{nV_T}{I} \ln \frac{I}{I_S} \tag{23.23}$$

To determine the reference current's variation with temperature, we can write

$$\frac{\partial I_{REF}}{\partial T} = \frac{\partial}{\partial T}\left(\frac{V_D}{R}\right) = \overbrace{\frac{1}{R} \cdot \frac{\partial V_D}{\partial T}}^{\text{change in } V_D \text{ with } T} - \overbrace{\frac{V_D}{R^2}\frac{\partial R}{\partial T}}^{\text{TC of the resistor}} \tag{23.24}$$

Remembering that the overall current change with temperature is written as

$$I_{REF}(T) = I_{REF}(T_0) \cdot (1 + TCI_{REF}(T - T_0)) \tag{23.25}$$

Start-up circuit not shown.
NMOS are 10/2.
PMOS are 30/2.
Scale factor is 1 micron.

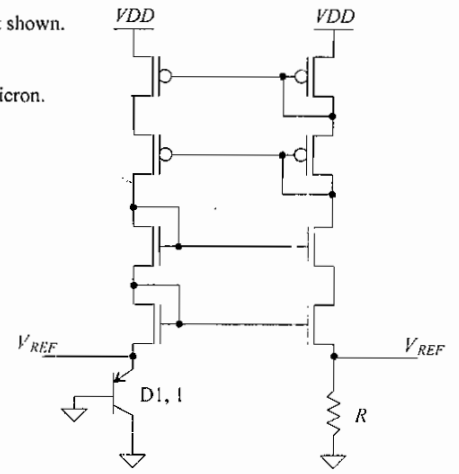


Figure 23.22 Diode-referenced, self-biasing circuit.

where

$$TCI_{REF} = \frac{1}{I_{REF}} \cdot \frac{\partial I_{REF}}{\partial T} \tag{23.26}$$

we can, with the help of Eq. (23.24), write

$$\frac{1}{I_{REF}} \frac{\partial I_{REF}}{\partial T} = \frac{1}{V_D} \frac{\partial V_D}{\partial T} - \frac{1}{R} \frac{\partial R}{\partial T} \tag{23.27}$$

The behavior of the diode-referenced, self-biased circuit shows a large negative coefficient. If the TC of the resistor is 0.002 and the change in the diode voltage with T (Eq. (23.20)) is −0.0016 (or if the diode voltage, V_D, is nominally 0.7, the first term in Eq. (23.27) is −0.0023), then the overall TC of the current is, roughly, −0.004 or −4,000 ppm/C. If we were to use the voltage across the diode (or resistor) as a voltage reference, the fact that the current through the diode is decreasing with increasing temperature (causing the diode's voltage to drop) and the bandgap of silicon is decreasing with increasing temperature, causes V_D to drop (relatively quickly) with increasing temperature (CTAT). Consider the following example.

Example 23.4

Simulate the operation of the reference in Fig. 23.22 where the voltage across the diode is used as a reference voltage. Use a nominal reference current of 1 μA.

Using Eq. (23.22), we can set R to 700k. The simulation results are seen in Fig. 23.23. The reference voltage change is essentially set by the decrease in the diode voltage with increasing temperature, that is, −1.6 mV/C. At the risk of stating the obvious, this isn't a very good voltage reference. ■

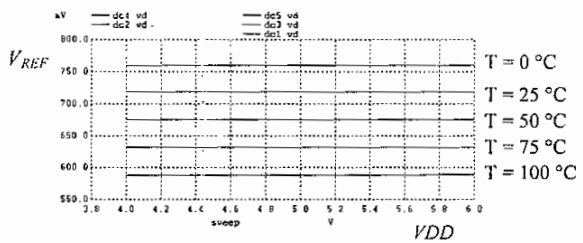


Figure 23.23 Temperature behavior of the diode-referenced circuit of Fig. 23.22. See Ex. 23.4.

Thermal Voltage-Referenced Self-Biasing (PTAT)

A thermal voltage, V_T, referenced, self-biasing circuit is seen in Fig. 23.24. In this configuration the voltage across D1 must equal the voltage across D2 and the resistor

$$V_{D1} = V_{D2} + I_{D2}R \tag{23.28}$$

Notice that D2 must be larger than D1 if the current flowing in the reference is to be nonzero. The larger D2 will drop a smaller voltage than D1 for the same current through each branch. As indicated in Eq. (23.28), the difference in the diode voltages is dropped

Start-up circuit not shown.
NMOS are 10/2.
PMOS are 30/2.
Scale factor is 1 micron.

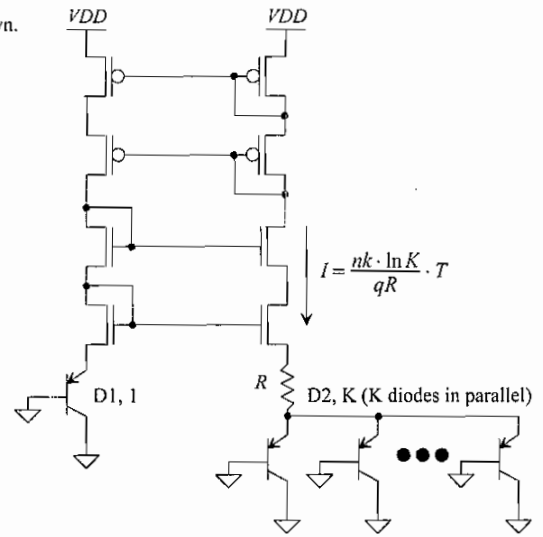


Figure 23.24 Thermal voltage-referenced, self-biasing circuit.

across R. Alternatively, we could size the diodes the same and increase the current flowing in the left branch of the reference by making the MOSFETs K times wider. The current in the left branch would be K times more than the current in the right branch of the reference.

We know

$$I_{D1} = I_S e^{V_{D1}/nV_T} \rightarrow V_{D1} = nV_T \cdot \ln \frac{I_{D1}}{I_S} \tag{23.29}$$

and

$$I_{D2} = K \cdot I_S e^{V_{D2}/nV_T} \rightarrow V_{D2} = nV_T \cdot \ln \frac{I_{D2}}{K \cdot I_S} \tag{23.30}$$

Knowing, because of the cascode structures, $I_{D1} = I_{D2} = I$, we can write

$$R = \frac{nV_T \cdot \ln K}{I} \text{ or } I = \frac{nk \cdot \ln K}{qR} \cdot T \tag{23.31}$$

Notice that the current is proportional to absolute temperature (PTAT). To get a voltage reference that is also PTAT, consider the reference seen in Fig. 23.25. Here the reference voltage is given by

$$V_{REF} = I \cdot L \cdot R = \frac{nk \cdot L \cdot \ln K}{q} \cdot T \tag{23.32}$$

Notice how the temperature behavior of the resistor falls out of the equation.

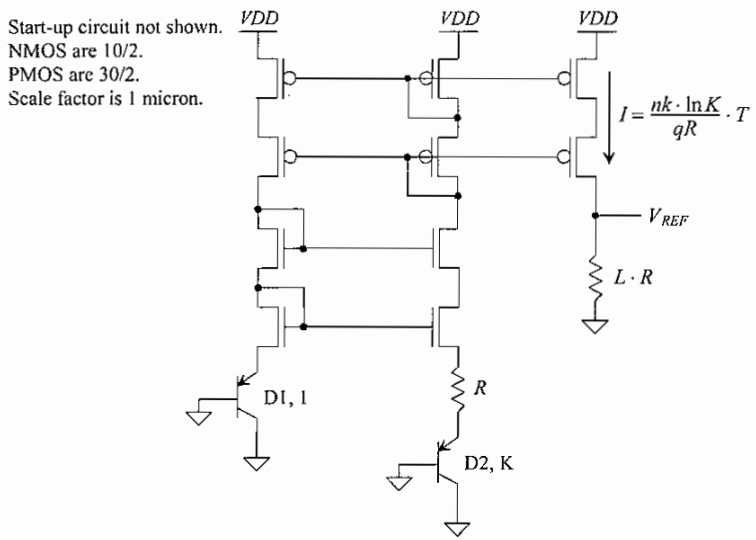


Figure 23.25 PTAT voltage reference based on the thermal voltage-referenced, self-biased circuit.

Example 23.5

Design a nominally 2.500 V voltage reference (at room temperature of 27 °C or 300 °K) PTAT voltage reference. Use the topology seen in Fig. 23.25 with a bias current of 1 μA. Simulate the operation of the design.

Let's set K (the number of diodes connected in parallel) to 8. We do this since it is easy to remember ln 8 = 2. Solving for R using Eq. (23.31) gives

$$R = \frac{1 \cdot 0.026 \cdot 2}{10^{-6}} = 52k$$

The voltage drop across the resistor is $1\,\mu A \cdot 52k = 52\,mV$. Using Eq. (23.32), we can solve for L as 48 (so $L \cdot R = 2.5\,M\Omega$). The simulation results are seen in Fig. 23.26. At room temperature, the reference voltage is roughly 2.5 V. Note how, as temperature increases, the reference voltage (PTAT) increases. Using Eq. (23.32), we can estimate the reference voltage's change with temperature as

$$\frac{\partial V_{REF}}{\partial T} = \frac{nk \cdot L \cdot \ln K}{q} \qquad (23.33)$$

Using the numbers from this example with $k = 1.38 \times 10^{-23} J/K$ and $q = 1.6 \times 10^{-19} C$ gives

$$\frac{\partial V_{REF}}{\partial T} = \frac{1 \cdot 1.38 \cdot 10^{-23} \cdot 48 \cdot 2}{1.6 \times 10^{-19}} = 8.26\,mV/C$$

For every 25 °C increase in temperature, we expect V_{REF} to go up by 206 mV. ■

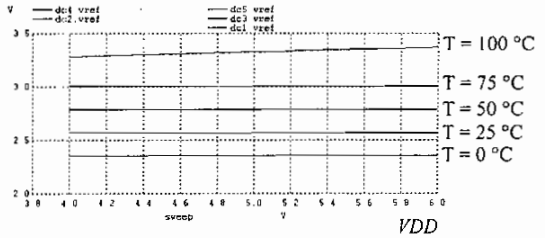


Figure 23.26 A PTAT voltage reference. See Ex. 23.5.

Bandgap Reference Design

The bandgap reference (BGR) is formed using CTAT and PTAT references. When designed properly, the TC of the BGR can be very small. Figure 23.27 shows a schematic of a BGR. The PTAT current generated from the circuit in Fig. 23.24 is driven into a resistive load (to give a PTAT voltage drop, see Eq. (23.32)) and a diode (to give a CTAT voltage drop, see Eq. (23.20)). The reference voltage is then the sum of the PTAT and CTAT voltage drops or

$$V_{REF} = V_{D3} + I \cdot L \cdot R = V_{D3} + L \cdot n \cdot \ln K \cdot V_T \qquad (23.34)$$

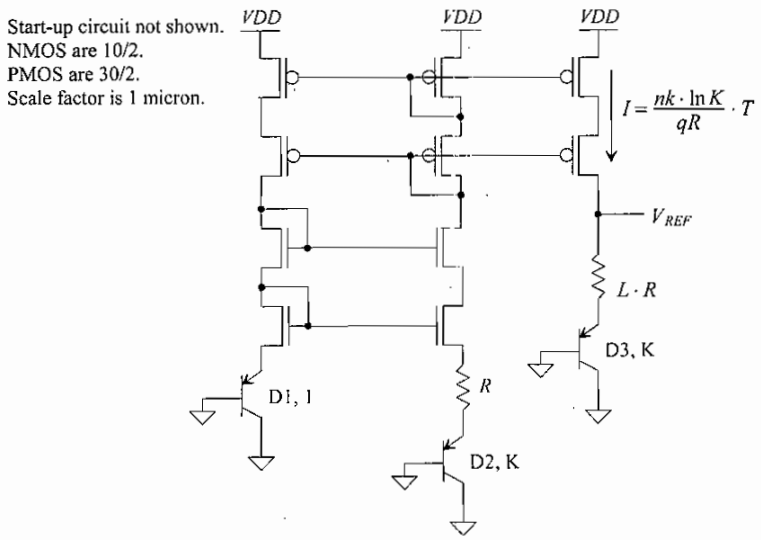


Figure 23.27 A bandgap reference circuit.

The change in the reference voltage with temperature is

$$\frac{\partial V_{REF}}{\partial T} = \overbrace{\frac{\partial V_{D3}}{\partial T}}^{-1.6\ mV/C} + L \cdot n \cdot \ln K \cdot \overbrace{\frac{\partial V_T}{\partial T}}^{0.085\ mV/C} \qquad (23.35)$$

where the change in the thermal voltage with temperature was taken from Eq. (9.44). To determine the value of L that causes the change in V_{REF} with temperature to go to zero, we set Eq. (23.35) to zero and solve

$$L = \frac{1.6}{n \cdot \ln K \cdot 0.085} \qquad (23.36)$$

If $K = 8$ and $n = 1$, then $L = 9.41$. Using Eq. (23.34) with these values, gives a V_{REF} of 1.2V. Figure 23.28 shows the simulation results using these values and the schematic seen in Fig. 23.27. At 100 °C the cascode structure doesn't appear to be holding the currents in each branch equal (and so the reference voltage isn't behaving properly). At the remaining temperatures, the change in the reference voltage with temperature is roughly 1.5 mV per 75 °C or 20 µV/C. The TC of the reference, Eq. (23.3), is then 16.7 ppm/C. Note that if a precision voltage is required, trimming the $L \cdot R$ resistor (see Fig. 23.15 and the associated discussion) is required.

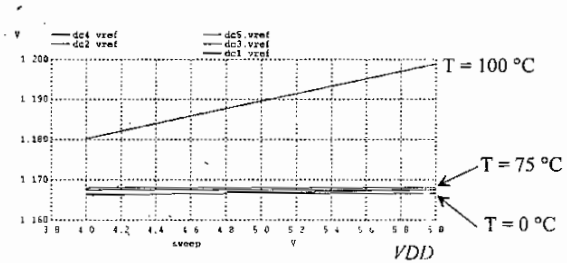
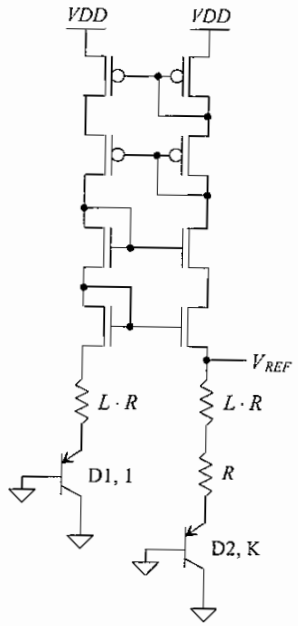


Figure 23.28 A PTAT voltage reference. See Ex. 23.5.

Alternative BGR Topologies

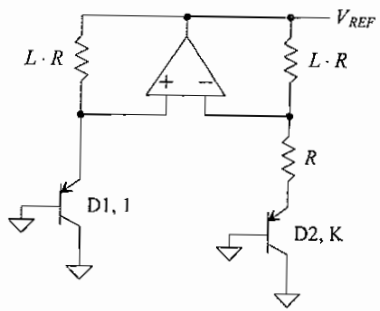
We might wonder if there is a way to simplify the BGR of Fig. 23.27 to reduce power and layout area. Figure 23.29a shows one possibility. In this circuit we've combined the three branches in Fig. 23.27 into two branches. The same voltage, as in Fig. 23.25, is dropped across the single resistor R. The $L \cdot R$ resistors are a common element to both branches. The current is still PTAT, while the reference voltage at the top of the $L \cdot R$ resistor is still a bandgap voltage. The drawback of this topology is the fact that the minimum value of *VDD* for proper reference operation increases.

Figure 23.29b shows a topology based on the configuration, using resistors, seen in Fig. 23.16. As with all of the topologies in this chapter (again), the point of the added amplifier is to force the same current through each side of the reference. The benefit of this topology over the topology in (a) is that *VDD* can move lower before it affects V_{REF}. The drawback of this topology is that the amplifier must be capable of driving a resistive load (which we'll see in the next chapter requires two gain stages for the amplifier's overall gain to be reasonably high). In (c) PMOS devices are added to isolate the

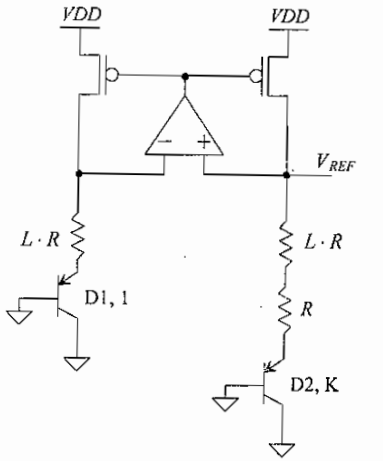


(a) Alternative BGR using cascodes.

Note: start-up circuits (required) not shown.



(b) Using an amplifier to force the same current through each branch of the reference, see Fig. 23.16.



(c) Using PMOS devices to isolate the amplifier from the resistive loading.

Figure 23.29 Alternative BGR topologies.

amplifier's output from the resistors. This makes using a diff-amp for the added amplifier possible. The currents are equal in (c) because the source-gate voltages of the PMOS devices are equal. Note how the branch containing D2 is a higher resistance (than the branch containing D1) and so it (the D2 branch) is always connected to an inverting point in the feedback loop. As seen in (c) the addition of the PMOS devices (which are inverting) means that we need to switch the inverting and noninverting amplifier inputs from (b). Finally, note that a start-up circuit is required for all three of these references.

23.2.2 Short-Channel BGR Design

The bandgap reference voltage of 1.2 V developed in the last section is greater than *VDD* (= 1 V) in our short-channel process. To develop a BGR for short-channel processes, consider the schematic seen in Fig. 23.30. The diodes D1 and D2 together with the resistor, *R*, form a PTAT current generator, as seen in Fig. 23.24. To provide a CTAT current to sum with the PTAT current, consider the addition of the $L \cdot R$ resistors, as seen in the schematic. As temperature increases, the diode voltage decreases, causing the current through the $L \cdot R$ resistors to decrease (CTAT). We know that the current due to the PTAT portion of the circuit, see Eq. (23.31), is

$$I_{PTAT} = \frac{nV_T \cdot \ln K}{R} \tag{23.37}$$

The current through the added resistors (the CTAT portion of the circuit) is

$$I_{CTAT} = \frac{V_{D1}}{L \cdot R} \tag{23.38}$$

The total current is driven through the $N \cdot R$ to generate the reference voltage

$$V_{REF} = nV_T \cdot N \cdot \ln K + \frac{N}{L} \cdot V_{D1} \tag{23.39}$$

The temperature behavior of the BGR is

$$\frac{\partial V_{REF}}{\partial T} = n \cdot N \cdot \ln K \cdot \overbrace{\frac{\partial V_T}{\partial T}}^{0.085\ mV/C} + \frac{N}{L} \cdot \overbrace{\frac{\partial V_{D1}}{\partial T}}^{-1.6\ mV/C} \tag{23.40}$$

For zero TC, we get an *L* of

$$L = \frac{1.6}{n \cdot \ln K \cdot 0.085} \tag{23.41}$$

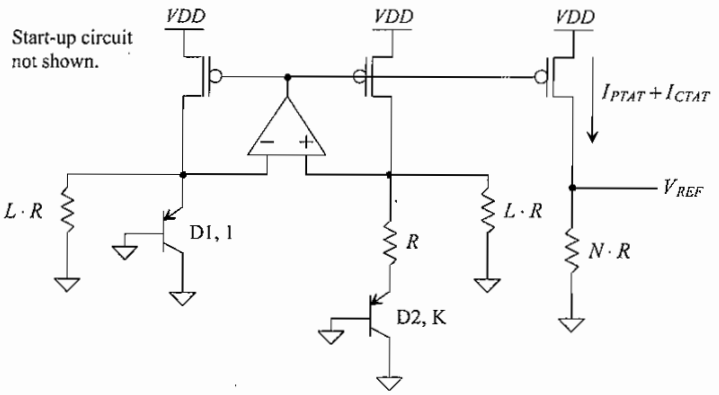
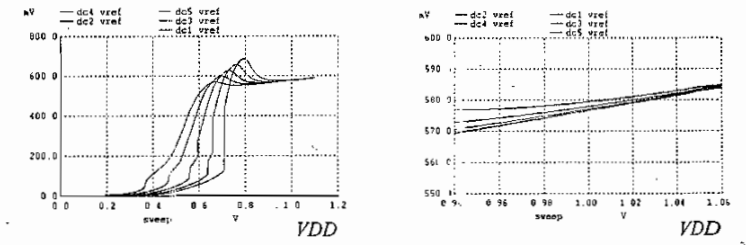


Figure 23.30 Lower voltage BGR.

Using a *K* of 8 results in, again, an *L* of 9.41. To get a particular reference voltage, we use Eq. (23.39) to determine *N*. For example, if we want a reference voltage of 500 mV (half of *VDD*), we get an *N* of (using a *K* of 8)

$$N = \frac{V_{REF}}{nV_T \cdot \ln K + \frac{V_{D1}}{L}} = \frac{0.5}{0.052 + \frac{0.7}{9.41}} = 3.91 \tag{23.42}$$

Figure 23.31 shows some simulation results using these numbers (the schematic of the full design is seen in Fig. 23.32). In (a) the reference turns on at a *VDD* of approximately 900 mV. The temperature behavior of the reference is seen in (b). Notice that the reference voltage is higher than what we designed for. In all practical situations, the $N \cdot R$ resistor (the 205k resistor in Fig. 23.32) would need trimming (see Fig. 23.15) to set the reference voltage to a precise value.



(a) The BGR turns on at 900 mV. (b) Close-up temperature behavior.

Figure 23.31 Simulating the behavior of the reference in Fig. 23.30.

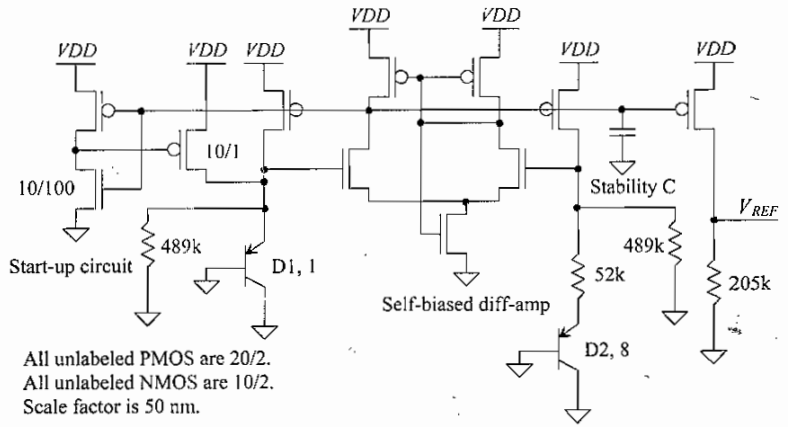


Figure 23.32 Lower voltage BGR used to generate the simulation data seen in Fig. 23.31.

The Added Amplifier

In Fig. 23.32 we used a self-biased amplifier (see Fig. 18.17) to hold the voltage across D1 to the same value as the voltage across the 52k resistor and D2. This amplifier is critical for good BGR performance. For example, notice how the reference voltage in Fig. 23.31b varies with changes in *VDD*. By using a higher-gain amplifier, we can reduce this variation. While we'll discuss op-amp design in detail in the next chapter, one concern is the added amplifier's input common-mode voltage. Because the forward voltage drop of the diodes is approximately 0.7 V (a significant percentage of *VDD*), the performance (gain) of the amplifier may drop if any of the MOSFETs in the diff-amp triode or shut off. To lower the input common-mode voltage, consider the topology of Fig. 23.33. Here we set

$$R_1 + R_2 = L \cdot R \tag{23.43}$$

The amplifier sets the voltages across each R_2 to the same value and so the current flowing in R_1 and R_2 is the same as in Fig. 23.30, provided Eq. (23.43) is satisfied. The big benefit of using the topology is that the DC input common-mode voltage of the added amplifier can be reduced.

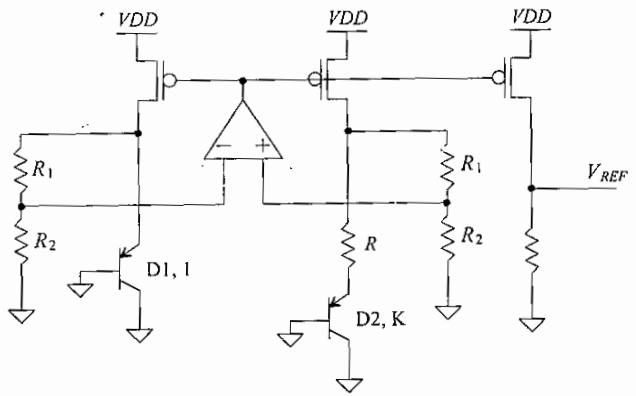


Figure 23.33 Lowering the input common-mode range of the added amplifier.

Lower Voltage Operation

Fundamentally, the limitation on how low *VDD* can be in the BGR reference of Fig. 23.30 (or 23.33) is the forward voltage drop of the diodes. As the dimensions of CMOS technology continue to scale downwards, so does *VDD*. While the MOSFET-resistor references, such as the beta-multiplier reference (BMR), discussed in the first section of this chapter may provide a solution to many reference circuit needs at lower *VDD*s, it is unlikely that they will replace the BGR in all applications. Even though both the BMR and BGR require trimming to set V_{REF}, the temperature performance of the BGR is, in general, better than that of the BMR. What may happen is that the pn junction used in the BGR is replaced by a junction with a lower built-in potential (and thus lower forward turn-on voltage) like the Schottky diode (see Fig. 23.21).

ADDITIONAL READING

[1] E. Vittoz and J. Fellrath, "CMOS Analog Integrated Circuits Based on Weak Inversion Operation," *IEEE Journal of Solid-State Circuits*, vol. SC-12, no. 3, June 1977, pp. 224–231. Covers the BMR design operating in the weak inversion region.

[2] K. E. Kuijk, "A Precision Reference Voltage Source," *IEEE Journal of Solid-State Circuits*, vol. SC-8, no. 3, June 1973, pp. 222–226. Development of the BGR seen in Fig. 23.29b.

[3] G. Tzanateas, C. A. T. Salama, and Y. P. Tsividis, "A CMOS Bandgap Voltage Reference," *IEEE Journal of Solid-State Circuits*, vol. SC-13, no. 3, June 1979, pp. 655–657. Good paper discussing BGR reference design.

[4] H. Banba, H. Shiga, A. Umezawa, T. Miyaba, T. Tanzawa, S. Atsumi, and K. Sakui, "A CMOS Bandgap Reference Circuit with Sub-1-V Operation," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, May 1999, pp. 670–674. Presents the design of the lower voltage BGR seen in Fig. 23.30.

[5] K. N. Leung and P. K. T. Mok, "A Sub-1-V 15-ppm/C CMOS Bandgap Voltage Reference Without Requiring Low Threshold Voltage Device," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 4, April 2002, pp. 526–530. Develops the scheme seen in Fig. 23.33 for lowering the input common-mode range of the added amplifier.

PROBLEMS

23.1 Using the MOSFET-only reference seen in Fig. 23.2, design a nominally 500 mV reference in the short-channel CMOS process. Using simulations, characterize the sensitivity of the reference voltage to changes in *VDD* and temperature.

23.2 Use the long-channel CMOS process and the topology seen in Fig. 23.6 to design a voltage reference of $3V_{THN}$. Simulate your design and show the *VDD* sensitivity and temperature behavior of the reference. Do TCs of R_1 and R_2 affect the performance of the reference? Why or why not?

23.3 Suppose it was desired, in Fig. 23.7 (see also Fig. 20.22), to make M1 and M2 the same size. However, to increase the gate source voltage of M1, relative to M2, the width of M3 is increased by *K*. How do the equations governing the operation of the BMR change? How does the current flowing in the BMR change?

23.4 Verify that if the PMOS devices in Fig. 23.11 are not cascoded (that is, they have only NMOS cascodes), the currents in each branch will not be equal and there will be significant sensitivity to *VDD* (a sensitivity similar to what is seen in Fig. 23.10).

23.5 In a CMOS process, several of the layers including poly, n+, p+, and n-well can be used for resistor formation. Each of these layers has a different temperature coefficient (TC). For the BMR that generated Fig. 23.14, use simulations to determine the optimum resistor TC.

23.6 Derive the equations that govern the operation of the reference in Fig. 23.16b.

23.7 Show why n+ directly in the p-substrate cannot be used as a diode in a CMOS process.

23.8 Generate a diode model that produces a forward voltage drop of 700 mV when driven with 1 μA and has a change with temperature, dV_D/dT, near room temperature, of −2 mV/C. Use simulations to verify your model meets the requirements.

23.9 Generate a SPICE model for the Schottky diode seen in Fig. 23.21. Assume that the series resistance of the diode is 1 kΩ.

23.10 Estimate, using hand calculations, the minimum allowable *VDD* for the reference of Ex. 23.4. What are the PMOS and NMOS gate-source and drain-source voltages when the reference current is 1 μA? Note that the parameters in Table 9.1 have nothing to do with the operating conditions in this question. Verify your answers using SPICE.

23.11 Show that K forward-biased diodes in parallel behave like a single diode with a scale current of $K \cdot I_S$, as assumed in Eq. (23.30).

23.12 Suggest a reference design that would output a voltage of $n \cdot V_T$.

23.13 Determine whether the performance of the BGR of Fig. 23.32 can be enhanced by using the topology of Fig. 23.33. Use simulations to verify your answer.

Chapter

24

Operational Amplifiers I

The operational amplifier (op-amp) is a fundamental building block in analog integrated circuit design. A block diagram of the two-stage op-amp with output buffer is shown in Fig. 24.1. The first stage of an op-amp is a differential amplifier. This is followed by another gain stage, such as a common source stage, and finally by an output buffer. If the op-amp is intended to drive a small purely capacitive load, which is the case in many switched capacitor or data conversion applications, the output buffer is not used. If the op-amp is used to drive a resistive load or a large capacitive load (or a combination of both), the output buffer is used.

Design of the op-amp consists of determining the specifications, selecting device sizes and biasing conditions, compensating the op-amp for stability, simulating and characterizing the op-amp A_{OL} (open-loop gain), CMR (common-mode range on the input), CMRR (common-mode rejection ratio), PSRR (power supply rejection ratio), output voltage range, current sourcing/sinking capability, and power dissipation.

We'll start this chapter off with a very simple two-stage op-amp (without an output buffer). By pointing out the weaknesses with this two-stage topology, we'll set the stage for developing practical op-amps in the rest of the chapter.

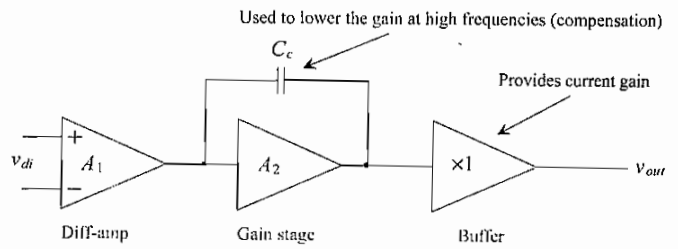


Figure 24.1 Block diagram of two-stage op-amp with output buffer.

24.1 The Two-Stage Op-Amp

Figure 24.2 shows the basic two stage op-amp made using an NMOS diff-amp and a PMOS common-source amplifier (M7). As seen in Fig. 22.8 M7 is biased to have the same current as M3 and M4 (10 µA from Table 9.2). Note also the addition of the compensating network consisting of a compensation capacitor, C_c, (*Miller compensation*) and a zero-nulling resistor R_z (see Figs. 21.25 and 21.33 along with the associated discussion). Because the op-amp doesn't have an output buffer, it is limited to driving capacitive loads and very large resistances (comparable to the output resistance of a MOSFET, that is, megaohms).

Low-Frequency, Open Loop Gain, A_{OLDC}

The low-frequency, open loop gain of the op-amp is calculated as the product of each stage gain, that is,

$$A_{OLDC} = A_1 \cdot A_2 = \overbrace{g_{mn} \cdot (r_{on} \| r_{op})}^{A_1 = \text{diff-amp's gain}} \cdot \overbrace{g_{mp} \cdot r_{op}}^{A_2,\ \text{M7's gain}} \qquad (24.1)$$

where the output resistance of the cascode current source load, M8, is assumed to be much larger than the M7 output resistance, r_{op}. Using the values from Table 9.2, we get an A_{OLDC} of 832 *V/V*.

Input Common-Mode Range

The minimum input common-mode voltage is given by Eq. (22.11) or 450 mV. The maximum input common-mode voltage is given by Eq. (22.12) or 900 mV. This means, for proper operation of our two-stage op-amp, the input voltages (v_p and v_m) should fall within the range of 450 to 900 mV. If they go outside this range, the op-amp gain drops, and it is likely that the circuit employing the op-amp will not function properly. Figure 24.3 shows a SPICE DC sweep where the *inverting* input (v_m) is held at 500 mV and the *noninverting* input (v_p) is swept from 495 to 505 mV. The slope of this transfer curve is the DC open-loop gain of the op-amp, A_{OLDC}.

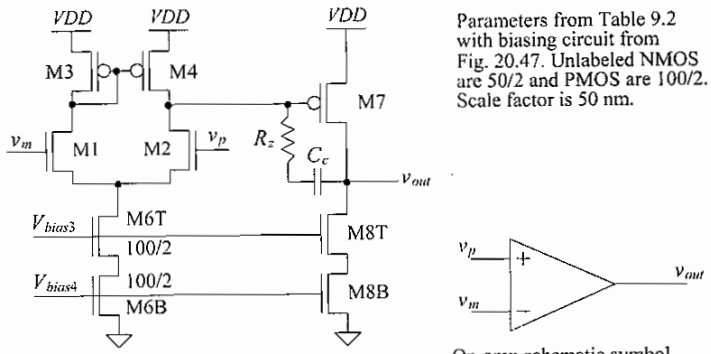


Figure 24.2 Basic two-stage op-amp.

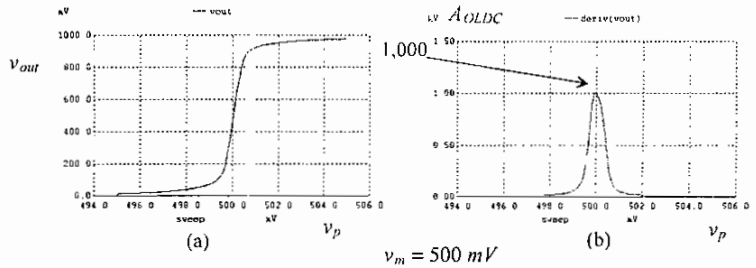


Figure 24.3 (a) DC transfer curves for the op-amp in Fig. 24.2 and (b) its gain (the derivation of (a)).

Power Dissipation

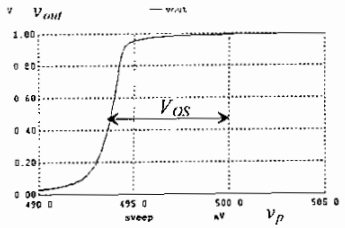
To determine the power dissipated by the op-amp, we sum the currents supplied by the constant current sources and multiply the result by *VDD*. For the op-amp in Fig. 24.2, the current through M6 is 20 µA and the current through M8 is 10 µA. The total power dissipation is then 30 µW (*VDD* = 1 *V*).

Output Swing and Current Sourcing/Sinking Capability

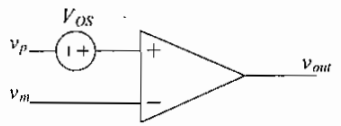
The maximum output swing (for the op-amp in Fig. 24.2) is limited by M7 going into the triode region. If we must keep at least 100 mV across M7, then the maximum output voltage is 900 mV. When M8 goes into triode or roughly 100 mV (see Fig. 20.48), the minimum output voltage is set. As seen in Fig. 24.3a, the high-gain (or large slope) region falls between v_{out} of 100 and 900 mV. Note that the maximum amount of current that this op-amp can sink is limited by the constant current sink M8 or 10 µA. The op-amp can source considerably more than 10 µA by pulling the gate of M7 downwards. Because this topology can source a considerable amount of current, it is useful as a voltage regulator (because the op-amp is always only sourcing current in a voltage regulator application).

Offsets

We've talked in great length (earlier in the book) about random offsets and how to design and layout circuits to minimize their effects. Another type of offset (that is not random) is termed a *systematic offset*. When we sized the MOSFETs in Fig. 24.2, for example, we made sure that M7 was sized to source 10 µA and M8 was sized to sink 10 µA. What would happen if we sized M7 to source 100 µA of current instead (changed its size from 100/2 to 1000/2)? Because M8 is a constant bias of 10 µA, M7 would move into the triode region until it was sourcing the 10 µA of current that M8 wants to sink. The output voltage would be very close to *VDD* with M7 in the triode region. Effectively we would get a shift or offset in the transfer curves seen in Fig. 24.3a (see Fig. 24.4a). To model this output voltage shift, we can refer it back to the op-amp input as an input-referred offset voltage, Fig. 24.4b. Note that unlike a random offset, which may be positive or negative in value, a systematic offset will always be a known polarity. Finally, note that while we chose to model the offset in series with the noninverting input terminal, in Fig. 24.4, we could just as easily have placed it in series with the inverting op-amp terminal (with a change in polarity).



(a)



(b) How the offset is modeled

$v_m = 500\ mV$

Figure 24.4 Showing how increasing M7's width to 1000 in Fig. 24.2 causes an input-referred (systematic) offset voltage.

Compensating the Op-Amp

An important step in op-amp design is the design of the compensation network. The op-amp takes the difference between the inverting and noninverting input terminal voltages and, ideally, multiplies the result by a very large number (the gain). A block diagram representation of an op-amp is seen in Fig. 24.5. The open-loop gain as a function of frequency is labeled $A_{OL}(f)$.

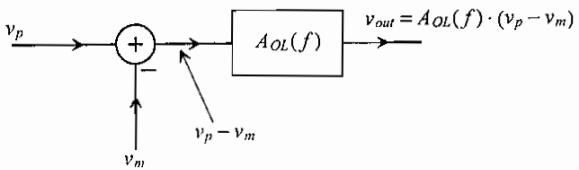


Figure 24.5 Block level diagram of an op-amp.

In all practical situations, the op-amp is used with feedback, Fig. 24.6. While our op-amp in Fig. 24.2 cannot drive a resistive load (unless it is megaohms, as a resistor on the op-amp output will kill the gain of the second stage), we'll still use this figure (Fig. 24.6) to illustrate the concept of feedback and compensation. We can write

$$v_{out} = A_{OL}(f) \cdot (v_{in} - v_f) \tag{24.2}$$

and

$$v_f = v_{out} \cdot \frac{R_2}{R_1 + R_2} \tag{24.3}$$

The amount of the output that is fed back is often called the feedback factor β or

$$\beta = \frac{R_2}{R_1 + R_2} \tag{24.4}$$

Substituting Eqs. (24.3) and (24.4) into Eq. (24.2) and solving for the closed loop gain gives

$$A_{CL}(f) = \frac{v_{out}}{v_{in}} = \frac{A_{OL}(f)}{1 + \beta \cdot A_{OL}(f)} \tag{24.5}$$

If we take $A_{OL}(f) \to \infty$, then the closed loop gain becomes

$$A_{CL}(f) \to \frac{1}{\beta} = 1 + \frac{R_1}{R_2} \tag{24.6}$$

We have several important points that we need to discuss. To begin, notice that in Eq. (24.5) if

$$\beta \cdot A_{OL}(f) = -1 \tag{24.7}$$

or more precisely

$$|\beta \cdot A_{OL}(f)| = 1 \text{ and } \angle \beta \cdot A_{OL}(f) = \pm 180° \tag{24.8}$$

the closed loop gain blows up (the feedback amplifier becomes unstable). The worst case situation (the largest value of β) occurs when all of the output is fed back to the op-amp input. The voltage follower, Fig. 24.7, is an example of this situation. *To determine the stability of an op-amp, we'll look at the open loop gain when the feedback factor is one, that is,*

$$|A_{OL}(f)| = 1 \text{ and } \angle A_{OL}(f) = 180° \tag{24.9}$$

We'll discuss this in greater detail in a moment. Notice that the larger the closed loop gain, the smaller the value of β (the less output signal we feed back) and the more likely the op-amp circuit, with feedback, will be stable. *This is important.* While feedback helps to desensitize an amplifier's gain to variations in an op-amp's A_{OL}, the drawback is stability. In a high-performance op-amp that will never operate in a unity-follower configuration, we can get an enhancement in speed by reducing the amount of compensation (the amount of the reduction guided by Eq. (24.8) with the actual value of β used).

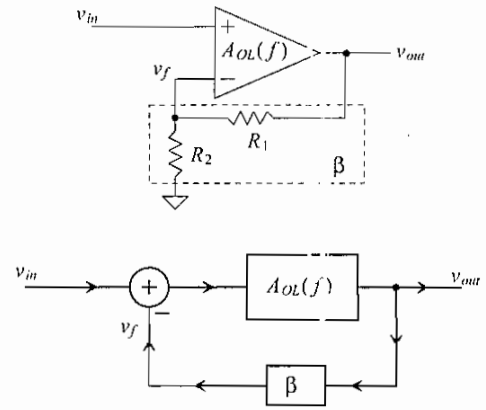


Figure 24.6 An example of feedback in an op-amp.

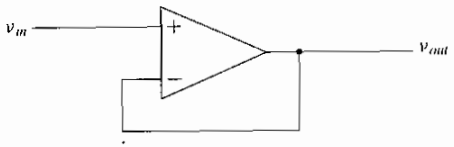


Figure 24.7 Voltage follower configuration, an example of a closed-loop amplifier with unity feedback factor.

To estimate the open-loop frequency response of the op-amp, let's use the generic model seen in Fig. 21.25. Figure 24.8 shows the location of nodes 1 and 2 on our two-stage op-amp. We've included a load capacitance in this figure. With the help of Table 9.2 we can write

$$R_1 = r_{on} || r_{op} = 111\ k\Omega$$

$$R_2 = r_{op} || R_{ocasn} \approx r_{op} = 333\ k\Omega$$

$$g_{m1} = g_{mn} = 150\ \mu A/V \text{ (diff-amp)}$$

$$g_{m2} = g_{mp} = 150\ \mu A/V \text{ (common-source)}$$

$$C_1 = C_{dg4} + C_{gd2} + C_{gs7} = 13.6\ fF$$

$$C_2 = C_L + C_{gd8} \approx C_L + 1.56\ fF$$

Let's calculate the open-loop response with a load and compensation capacitance of 100 fF, that is, $C_L = C_c = 100\ fF$. The pole associated with node 1, from Eq. (21.65), is

$$f_1 \approx \frac{1}{2\pi g_{m2} R_1 R_2 C_c} = 287\ kHz$$

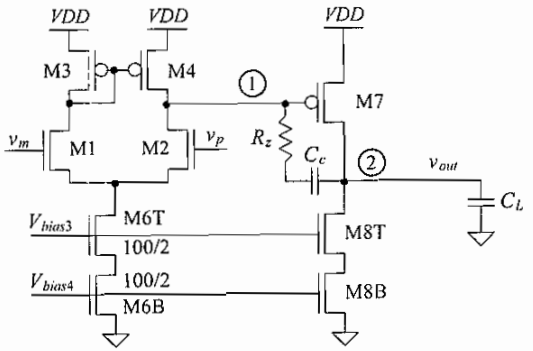


Figure 24.8 Calculating the frequency response of the op-amp.

The location of the pole associated with the output node (node 2) is, from Eq. (21.66),

$$f_2 = \frac{g_{m2} C_c}{2\pi(C_c C_1 + C_1 C_2 + C_c C_2)} = 210\ MHz$$

The location of the zero, f_z in Eq. (21.63), is at 240 MHz (very near the second pole).

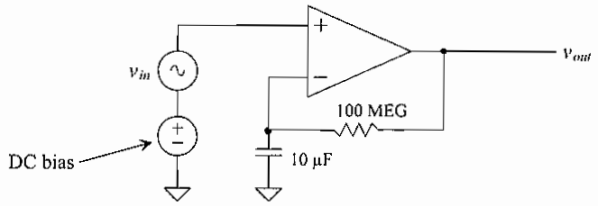


Figure 24.9 Circuit configuration used to simulate open-loop frequency response.

To simulate the open-loop response of the op-amp, we can use the configuration seen in Fig. 24.9. The feedback resistor and capacitor form a time constant so large that for all intents and purposes none of the AC output voltage is fed back to the inverting input. However, the DC bias level is fed back so that the op-amp biases up correctly (all MOSFETs are operating in the saturation region). With a DC bias voltage of 500 mV, Fig. 24.10 shows the open loop responses of the op-amp in Fig. 24.8 with 100 fF compensation capacitance and load capacitance. The simulated values are close to the

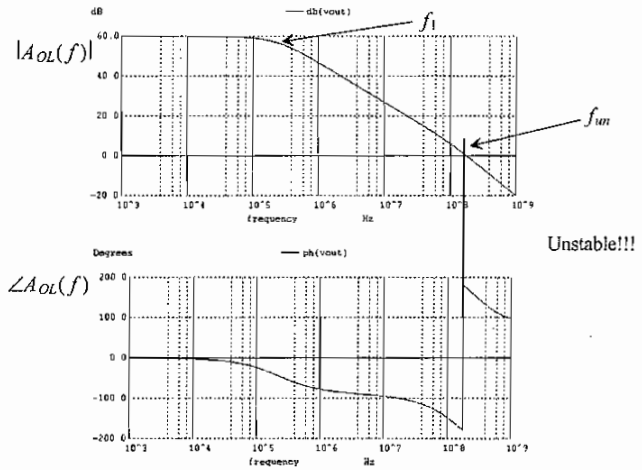


Figure 24.10 The open-loop frequency response of the op-amp in Fig. 24.8 with a 100 fF compensation capacitance and load capacitance.

hand-calculated values given above. However, from the criteria given in Eq. (24.9), the op-amp will clearly be unstable. *We want the open loop gain of the op-amp to be much less than one when the phase shift is 180°.* Looking at Fig. 24.10, we effectively want to shift the lower frequency pole (f_1) downwards in frequency. At the same time, we want to move the higher frequency pole, f_2 , higher in frequency (we want to *split the poles*). Remember this (pole-splitting) was the point of adding the compensation capacitor to an amplifier back in Ch. 21. Let's use Eq. (21.72) to select C_c. Looking at Fig. 24.10, we see that the phase shift is −100° at 10 MHz, so let's set the unity gain frequency to this value (in an attempt to make sure that the open loop gain is well under one when the phase shift is 180°).

$$f_{un} = \frac{g_{m1}}{2\pi C_c} = \frac{150\ \mu A/V}{2\pi \cdot C_c} = 10\ MHz \rightarrow C_c = 2.4\ pF \qquad (24.10)$$

The simulation results are seen Fig. 24.11. The unity-gain frequency, f_{un} , is close to 10 MHz. However, while the lower frequency moved downwards and the higher frequency pole moved upwards, the zero moved down to unity-gain frequency, see Eq. (21.63). This causes the open-loop gain to hover around unity instead of continuing to decrease (this is bad). The stability and step response of the op-amp will be degraded because of the zero. To illustrate this, consider the simulation results seen in Fig. 24.12. Notice that, in a transient simulation, we have to wait some time to let the bias circuit start up. Here, in this simulation, we've waited 500 ns before applying the input signal. Next, notice that our step input signal has a small amplitude change (5 mV). If we apply a larger step, the small-signal analysis used to derive pole-splitting is no longer valid (and we'll see the slew-rate limitations). Clearly the output of the op-amp doesn't behave well with the zero present. To eliminate the zero, we add the zero-nulling resistor, as seen in Fig. 21.33b. Figure 24.11 is regenerated in Fig. 24.13 with the addition of a zero-nulling resistor ($1/g_{mn} = 6.5$k). Figure 24.14 shows the well-behaved step response.

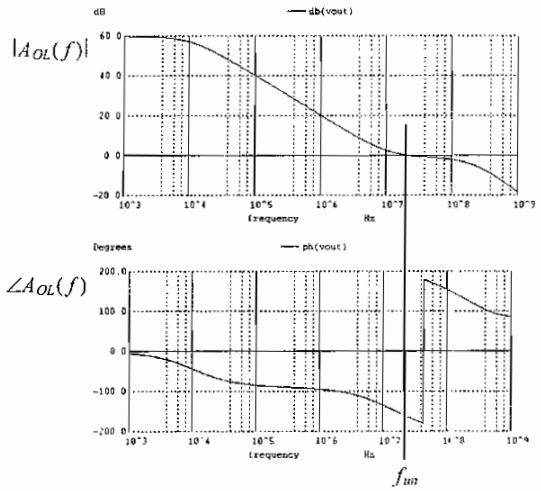


Figure 24.11 Increasing the compensation capacitor's value to 2.4 pF.

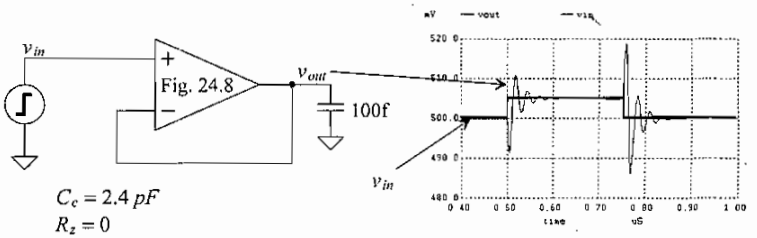


Figure 24.12 The poor step response of the op-amp with the zero present.

Gain and Phase Margins

In the previous discussions we assumed a load of 100 fF. If the load capacitance varies, the stability can (will) be affected. Further, with temperature, process, and power supply variations, the stability of the op-amp can change. In order to specify "how stable" the op-amp is at a given set of operating conditions, the parameters gain margin (GM) and phase margin (PM) are used. To determine an op-amp's PM, we look at the phase shift when the open-loop gain is unity. The amount of phase shift away from 180° is the PM of the op-amp. As seen in Fig. 24.13, the phase shift when the gain is unity is –90°. Taking the difference between this value and 180° gives a PM of 90°. To calculate the GM, we look at the difference between the open-loop gain and unity when the phase of the op-amp is ±180°. For the op-amp response in Fig. 24.13, the GM is approximately 25 dB. Note that as seen in Fig. 24.14, it is nice to have a PM of 90° because the step response of the op-amp has a first-order response (like an RC circuit).

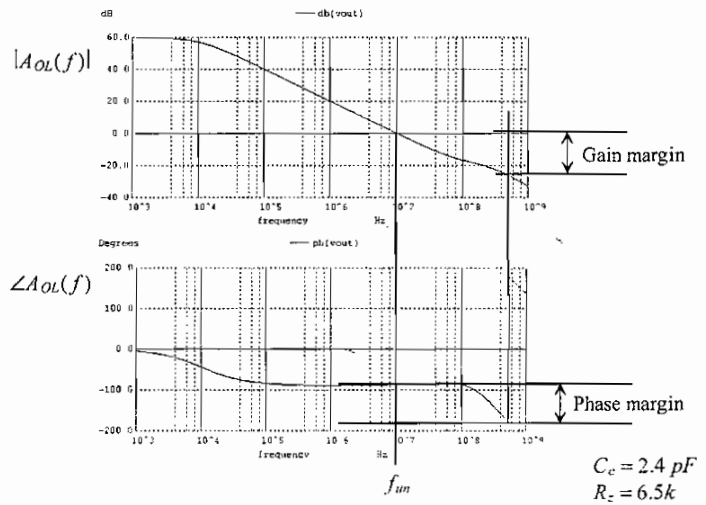


Figure 24.13 Adding a zero nulling resistor to the op-amp in Fig. 24.8.

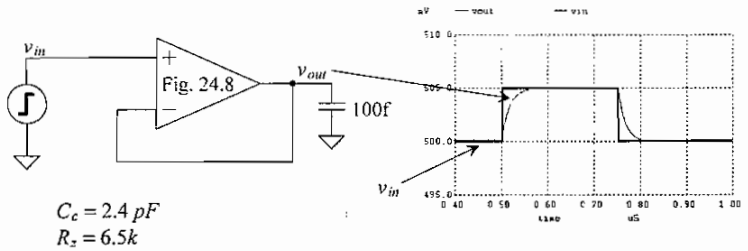


Figure 24.14 Good step response of the op-amp with the zero absent.

Removing the Zero

As the previous discussion illustrated, the RHP zero can raise issues concerning stability and settling time (this is important). As seen in Eq. (21.73) and the associated discussion, R_z can be added to eliminate ($R_z = 1/g_{m1}$) or move the zero into the LHP ($R_z > 1/g_{m1}$). When the zero is moved to the LHP, the phase response of the zero adds to the overall phase response, increasing the PM (this is called *lead compensation*). The practical problem with using R_z is that setting its value to a precise number (say $1/g_{m1}$) is challenging with shifts in process, temperature, or voltage. One solution to this problem is to replace the resistor with a MOSFET operating in the triode region, Fig. 24.15. Mz behaves like a resistor with a value of $1/g_{m1}$. Ideally, the source-gate voltage of M7 is the same as the V_{SG} of MP1. It then follows that the source-gate potential of MP2 equals the source-gate potential of Mz. The channel resistance of Mz is then, from Eq. (9.16), set to $1/g_{m1}$. The practical issues with this method are the wasted power dissipated by the additional circuitry and the fact that if the output swing becomes large (especially at higher frequencies where C_c has a small impedance), Mz can move out of the triode region, which can affect the large-signal behavior of the op-amp.

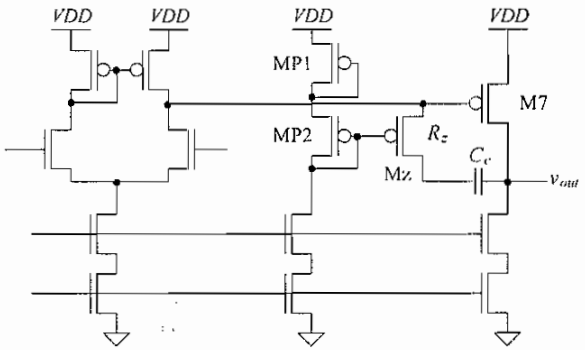


Figure 24.15 Making the zero-nulling resistor process independent.

The other method to remove the RHP zero seen in Fig. 21.33a is to add an amplifier with a gain of +1 in series with the compensation capacitor. Figure 24.16 shows the idea. A source follower allows the output signal to feed back through the compensation capacitor (so that the pole-splitting effect is still present). However, the root cause of the RHP zero, namely, C_c shorting the input of the second stage (the output of the diff-amp) to the output of the second stage (the output of the op-amp) at higher frequencies is removed. The concerns with this topology are, again, power dissipation and, perhaps more importantly, large signals. If the overall output voltage swings too low, it causes the source-follower to shut off.

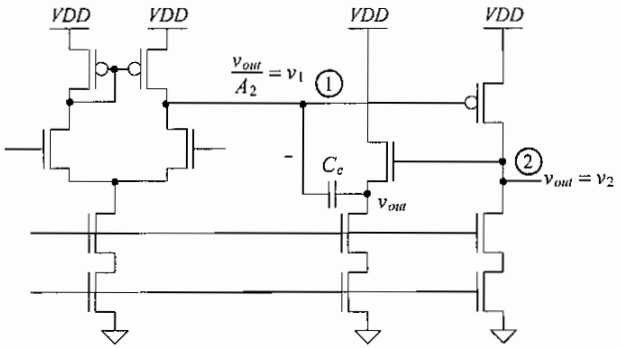


Figure 24.16 Using an amplifier to eliminate forward signal feedthrough via the compensation capacitor.

Compensation for High-Speed Operation

Notice, in Fig. 24.16 (or in any other of the topologies we've discussed up to this point), that the current fed back through C_c is

$$i_{Cc} = \frac{v_{out} - \frac{v_{out}}{A_2}}{1/j\omega C_c} \tag{24.11}$$

If the second-stage gain, A_2 ($g_{m2}R_2 = g_{m7}r_{o7}$) is reasonably large, then this equation can be approximated using

$$i_{Cc} \approx \frac{v_{out}}{1/j\omega C_c} \tag{24.12}$$

If we can feed back this current *indirectly* to the output of the diff-amp, we can still compensate the op-amp (and have pole-splitting). Further, if we do it correctly, we avoid connecting the compensation capacitor directly to the output of the diff-amp and thus avoid the RHP zero. Towards this goal, consider the modified op-amp schematic seen in Fig. 24.17. The added MOSFETs form a common-gate amplifier. (Note that here we are assuming $v_p \ll v_{out}$, so the source-follower action of MCG is negligible. MCG is connected to v_p to set its gate at the DC voltage of the input.) The current i_{Cc} is fed back through the common-gate MOSFET, MCG, to node 1 (the output of the diff-amp).

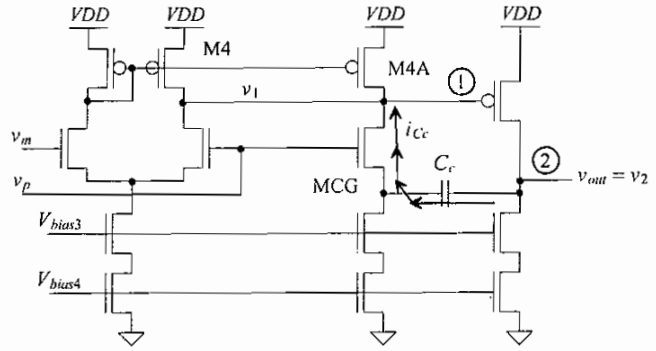


Figure 24.17 Feeding back a current indirectly to avoid the RHP zero.

To determine the frequency response of this amplifier, consider the model seen in Fig. 24.18 (modified from Fig. 21.25). Summing the currents at node 1 gives

$$-g_{m1}v_s + \frac{v_1}{R_1 \| \frac{1}{j\omega C_1}} - \overbrace{\frac{v_{out}}{1/j\omega C_c + 1/g_{mcg}}}^{i_{Cc}} = 0 \qquad (24.13)$$

where the resistance looking into the source of MCG is $1/g_{mcg}$. Solving for v_1 gives

$$v_1 = \frac{g_{m1}R_1}{1+j\omega R_1 C_1} \cdot \left(\frac{j\omega\frac{C_c}{g_{m1}}}{1+j\omega\frac{C_c}{g_{mcg}}} \cdot v_{out} + v_s \right) \qquad (24.14)$$

For the output node (node 2), we can write (noting that now C_2 includes both C_c and C_L)

$$v_{out} = -g_{m2}v_1 \cdot \left(\frac{R_2}{1+j\omega R_2 C_2} \right) \qquad (24.15)$$

Plugging Eq. (24.14) into this equation gives

$$v_{out} = -g_{m1}R_1 g_{m2}R_2 \cdot \frac{\frac{j\omega\frac{C_c}{g_{m1}}}{1+j\omega\frac{C_c}{g_{mcg}}} \cdot v_{out} + v_s}{(1+j\omega R_1 C_1)(1+j\omega R_2 C_2)} \qquad (24.16)$$

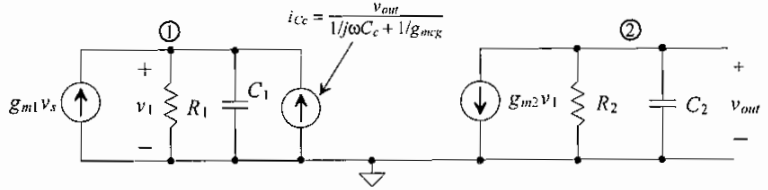


Figure 24.18 Model used to estimate bandwidth when indirect feedback current is used.

Letting

$$K = \frac{-g_{m1}R_1 g_{m2}R_2}{(1+j\omega R_1 C_1)(1+j\omega R_2 C_2)} \qquad (24.17)$$

we can rewrite Eq. (24.16) as

$$v_{out} = \frac{j\omega\frac{C_c}{g_{m1}}}{1+j\omega\frac{C_c}{g_{mcg}}} \cdot K \cdot v_{out} + K \cdot v_s \qquad (24.18)$$

and thus

$$\frac{v_{out}}{v_s} = \frac{K \cdot \left(1+j\omega\frac{C_c}{g_{mcg}}\right)}{1-j\omega\frac{C_c K}{g_{m1}}} \qquad (24.19)$$

Resubstituting in K gives

$$\frac{v_{out}}{v_s} = \frac{-g_{m1}R_1 g_{m2}R_2\left(1+j\omega\frac{C_c}{g_{mcg}}\right)}{(1+j\omega R_1 C_1)(1+j\omega R_2 C_2) + j\omega\frac{C_c}{g_{m1}} \cdot g_{m1}R_1 g_{m2}R_2} \qquad (24.20)$$

or, with $s = j\omega$, we can write

$$\frac{v_{out}}{v_s} = \frac{-g_{m1}R_1 g_{m2}R_2\left(1+s\frac{C_c}{g_{mcg}}\right)}{s^2 \cdot (R_1 C_1 R_2 C_2) + s \cdot (R_1 C_1 + R_2 C_2 + R_1 g_{m2} R_2 C_c) + 1} \qquad (24.21)$$

Notice there is a LHP zero at

$$f_z = \frac{g_{mcg}}{2\pi C_c} \qquad (24.22)$$

Since this zero is in the LHP, it will add to the phase response and enhance the speed of the op-amp. Intuitively, we can think that at high speeds the phase shift through C_c will cause the output signal to feed back and add to the signal at node 1. This *positive feedback* enhances the speed of the op-amp. To determine the location of the second pole, let's assume $R_1 g_{m2} R_2 C_c >> R_1 C_1$ or $R_2 C_2$ or $R_1 C_1 R_2 C_2$ so

$$s_{1,2} \approx \frac{-R_1 g_{m2} R_2 C_c \pm R_1 g_{m2} R_2 C_c}{2(R_1 C_1 R_2 C_2)} \qquad (24.23)$$

Our approximation won't tell us the location of the lower frequency pole (it's given by Eq. (21.65)). The location of the second pole is

$$f_2 \approx \frac{g_{m2}C_c}{2\pi \cdot C_1 C_2} \approx \frac{g_{m2}C_c}{2\pi \cdot C_1(C_L + C_c)} \qquad (24.24)$$

This result should be compared to Eq. (21.66). The location of the second pole is at a considerably higher frequency using this technique. *The result is that we can set the unity gain frequency to a higher value and still have a stable op-amp.* Further, the load capacitance (which is included in C_2) can be considerably larger for a given PM or GM. Again, Eq. (21.72) can be used to set the unity-gain frequency, f_{un}, assuming $f_2 \approx f_z$

$$f_{un} = \frac{g_{m1}}{2\pi C_c} \ (\approx f_z \text{ if } g_{m1} = g_{mcg}) \qquad (24.25)$$

Using the *indirect compensation* method seen in Fig. 24.17 in the op-amp of Fig. 24.8 with C_c = 240 *fF* (f_{un} = 100 MHz) gives the results seen in Fig. 24.19. The small-signal step response is seen in Fig. 24.20. This response should be compared to Figs. (24.12) and (24.14).

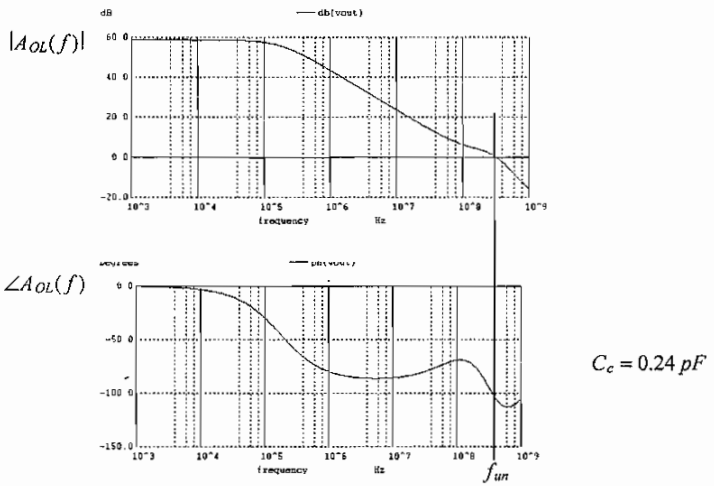


Figure 24.19 Simulating the op-amp of Fig. 24.8 using the indirect compensation scheme seen in Fig. 24.17 with a compensation capacitor of 240 fF.

The indirect feedback of the current through the compensation capacitor results in faster op-amp circuits and less layout area (the compensation capacitor generally dominates the layout area of an op-amp). However, the added circuit in Fig. 24.17 dissipates more power. To eliminate the additional power dissipation, consider the op-amp topology seen in Fig. 24.21. Here we've used the fact that a 100/2 PMOS device can be laid out as two 100/1 PMOS in series (see also Fig. 20.35). The current through the compensation capacitor is fed back through M4B to the output of the diff-amp. Note how we've also split M7 into two devices. It's important, for small offsets, to try to match both the drain-source and the gate-source voltages of the MOSFETs used in the op-amp. The step-response of the op-amp is seen in Fig. 24.22.

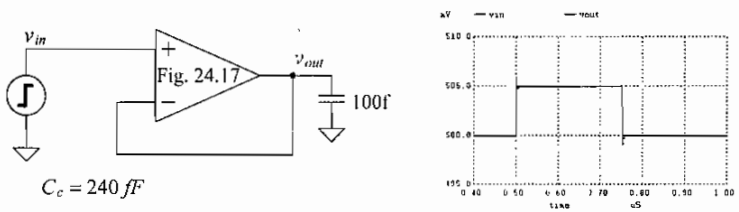


Figure 24.20 The step response of the op-amp in Fig. 24.17 with frequency response seen in Fig. 24.19.

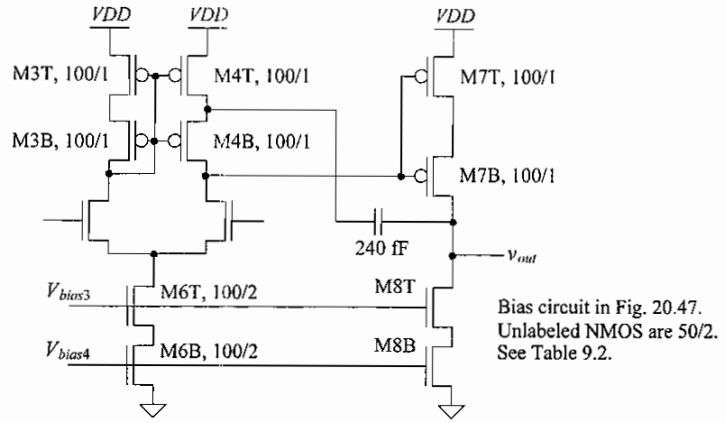


Figure 24.21 Implementing indirect feedback compensation without additional power dissipation in a two-stage op-amp.

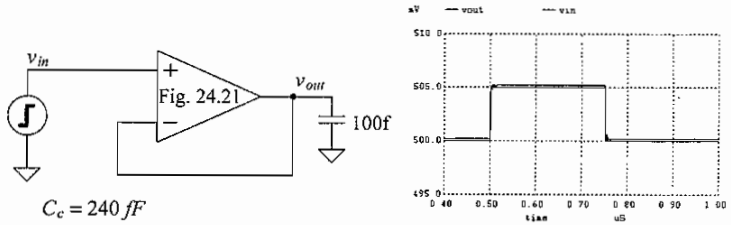


Figure 24.22 The step response of the op-amp in Fig. 24.21 driving 100 fF.

Slew-Rate Limitations

When we talked about step response, we limited the amplitude of the signals to small steps (5 mV) to avoid slew-rate limitations. Let's take the op-amp from Fig. 24.8 and put it in the configuration seen in Fig. 24.14, driving a 1 pF load capacitance. Further, let's increase the input pulse amplitude to (close to) the maximum allowable range, that is, from 500 mV to 900 mV (limited by the op-amp's input common-mode range). The simulated results are seen in Fig. 24.23. Referring to Fig. 24.8, notice that when the op-amp's noninverting input terminal (v_p) is driven high, M2 turns on and pulls the gate of M7 down. Since there isn't a current source in series with M7, it can quickly charge the 1 pF load capacitance. The only large-signal limitation is the bias current of the diff-amp, I_{SS} (= 20 µA here), charging C_c (= 2.4 pF here). This limits the output rate of change since any variation in v_{out} causes a displacement current through C_c, which must be sourced/sunk by the diff-amp. This slew-rate limitation is calculated as 20 µA/2.4 pF or 8.3 mV/ns. The change in the output voltage is 400 mV, so we would require roughly 50 ns to change the voltage across C_c.

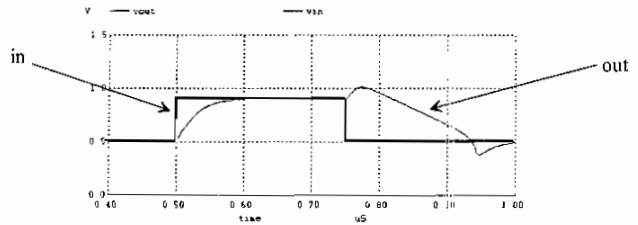


Figure 24.23 The large-signal (500 mV to 900 mV) performance of the op-amp in Fig. 24.8 with $C_c = 2.4$ pF and $R_z = 6.5$ k driving a 1pF load. The low-to-high settling time is roughly 100 ns, while the high-to-low settling time, which is slew-rate limited, is roughly 500 ns.

The more interesting behavior occurs when the input signal drops from 900 mV to 500 mV. This causes M2 to shut off, allowing all of the current from M4 to charge the gate of M7. The result is a positive pulse on the gate of M7 that feeds directly to the output through the compensation capacitor (this is bad, and it is another example of why it is desirable to use the indirect compensation method). This pulse accounts for the (unwanted) positive movement in the output voltage seen in Fig. 24.23 just after the input pulse transitions negative. Now the load capacitor and the compensation capacitor must both be discharged through the constant current source M8. The output slew-rate is estimated by how fast this constant current (here 10 µA) can discharge 3.4 pF (the sum of the compensation capacitor and the load capacitor). This rate is calculated as (roughly) 3 mV/ns. For the output to transition 400 mV requires 133 ns (this time will be longer because of the positive movement in the output voltage just after the input switches). Note that we might think that simply by sizing up the current conducting in M8 we can enhance the op-amp's slew-rate. This is true to the point where the diff-amp's current charging the compensation capacitor becomes the limiting factor.

Finally, note how the output voltage shoots past the final (desired) voltage level of 500 mV. This is because the gate of M7 is pulled to *VDD* to shut it off when slewing is taking place (both M4 and M7 are shut off because of this). When the inputs of the op-amp move to the same value (500 mV here), the gate of M7 must be pulled downwards to the quiescent value. The time it takes for this to happen results in the undershoot in the op-amp's output voltage seen in Fig. 24.23.

The same test setup used to generate the data in Fig. 24.23 is also used to generate the data in Fig. 24.24 except that here the op-amp in Fig. 24.21 is used. Note the compressed time scale. The settling time for the low-to-high transition is now 10 ns, while the high-to-low transition time is 60 ns. We would expect much faster settling because the compensation capacitor is now ten times smaller than the value used in the op-amp of Fig. 24.8. Again, the settling time is limited by how fast M8 can discharge the load capacitance and the compensation capacitance. This can be estimated as 10 µA/1.24 pF or 8 mV/ns. It takes 50 ns for the output to transition 400 mV. The size of M8 can be increased to meet a specific load capacitance and settling time requirement.

Because of the improved speed performance, smaller layout area, and (as we'll see) better power supply noise rejection, we'll use indirect compensation in place of Miller compensation in the remaining two-stage op-amps that we discuss in this book.

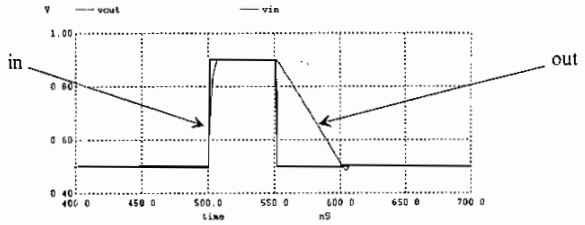


Figure 24.24 The large-signal (500 mV to 900 mV) performance of the op-amp in Fig. 24.21 with $C_c = 0.24$ pF driving a 1 pF load. The low-to-high settling time is roughly 10 ns, while the high-to-low settling time, which is slew-rate limited, is roughly 60 ns. Note the different time scale when compared to Fig. 24.23.

Common-Mode Rejection Ratio (CMRR)

The *CMRR* of an op-amp is calculated in the same way as the diff-amp in Sec. 22.1.3. The common-mode gain of the diff-amp is A_c. The common-mode gain of the op-amp is $A_c A_2$. The differential gain of the op-amp is $A_{OL}(f) = A_d \cdot A_2$ (where $A_d = A_1$). The *CMRR* of an op-amp in dB is given by

$$CMRR = 20 \cdot \log \left| \frac{A_{OL}(f)}{A_c \cdot A_2} \right| = 20 \cdot \log \left| \frac{A_d}{A_c} \right| \qquad (24.26)$$

which shows that the op-amp *CMRR* is determined by the differential stage. Simulating the *CMRR* of an op-amp can be accomplished with the circuits seen in Fig. 24.25. For the op-amps discussed so far in this chapter, the *CMRR* is approximately 50 dB (see Ex. 22.7, where the common-mode voltage is 500 mV). We can think of A_c as the open-loop gain of the op-amp for common-mode signals. If the applied differential voltage is zero, and we change the common-mode voltage by ΔV_c, then the output voltage will change by ΔV_o

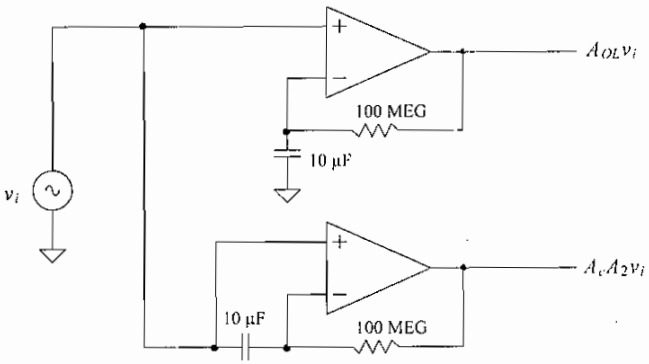


Figure 24.25 Circuit configuration used to simulate CMRR.

$= A_{cm} \cdot \Delta V_c$ (see Fig. 22.17). To compensate for the change in the output voltage, a nonzero input differential voltage develops on the input of the op-amp (an offset voltage that is a function of the common-mode voltage). This offset voltage can be estimated by

$$\Delta V_{OS} = \frac{\Delta V_o}{A_{OL}} = \frac{\Delta V_c \cdot A_c}{A_{OL}} = \frac{\Delta V_c}{CMRR} \qquad (24.27)$$

Knowing that 50 dB = 316, a change in the common-mode level on the inputs of the op-amp by 500 mV results in an input-referred offset voltage change of 500 mV/316 or 1.6 mV (if the gain of the diff-amp is 10, then its output voltage will change by 16 mV). This may not seem like such a big deal. However, notice that at higher frequencies, in Fig. 22.16, the *CMRR* falls off, indicating that the common-mode gain is increasing. This leads to distortion and forces the use of op-amp topologies in which common-mode voltage doesn't vary (the inverting op-amp topology, see Fig. 24.26).

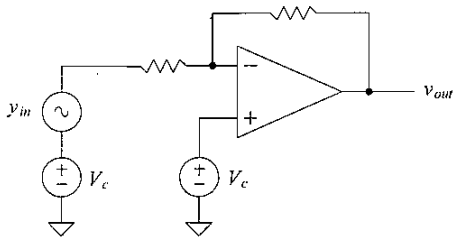


Figure 24.26 An inverting op-amp topology. The common mode voltage is held constant.

Power Supply Rejection Ratio (PSRR)

The power supply rejection ratio (PSRR) is a term used to describe how well an amplifier rejects noise or changes on the *VDD* and or ground power buses. This parameter can be extremely important in precision analog design. Consider the test setup shown in Fig. 24.27. The positive PSRR is defined by

$$PSRR^+ = \frac{A_{OL}(f)}{v_{out}/v^+} \qquad (24.28)$$

while the negative PSRR is defined by

$$PSRR^- = \frac{A_{OL}(f)}{v_{out}/v^-} \qquad (24.29)$$

Ideally, v_{out} doesn't vary with changes in *VDD* and ground (and so the *PSRR* is infinite).

To understand how variations in *VDD* or ground can feed to the output of the amplifier, consider the op-amp in Fig. 24.8. Variations in *VDD* feed through to the gate of M3. Because of the circuit's symmetry, this causes the drain of M4, and thus the gate of M7, to move around. However, the source of M7 is moving the same amount, causing the v_{sg} of M7 to remain constant. Again, because of the symmetry, the output voltage

moves at the same rate. Thus, noise on *VDD* feeds directly to the output of the op-amp. This is indicated in Fig. 24.27c, where v_{out} / v^+ is one. At higher frequencies (in the kHz range), the gate and drain of M7 get shorted together through the compensation capacitor. This, again, causes all of the noise from *VDD* to feed directly to the output of the amplifier (see netlists at cmosedu.com). The indirect compensation scheme used in the op-amp of Fig. 24.21 can be designed for larger f_{un} and, thus, has better *PSRR* at higher frequencies. Using the common-gate stage, Fig. 24.17, can result in an even higher *PSRR* by isolating the compensation capacitor from the power supplies. For this reason op-amp's that use a cascode-load diff-amp, Fig. 24.29, are very useful in practical design.

Noise on ground ideally won't affect the operation of the op-amp (in either of the topologies of Fig. 24.8 or 24.21). The noise appears as a voltage variation across the current sources. In reality, at low frequencies, some of the ground noise, v^-, appears at the output (this is especially true in nanometer processes that have low r_o). However, at higher frequencies, because of the compensation capacitance connected to the output of the op-amp, ground noise contributions to the output signal decrease.

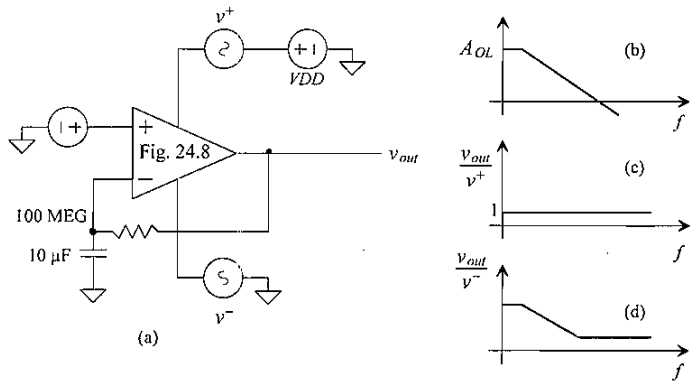


Figure 24.27 (a) Test setup to determine PSRR, (b) open-loop gain, (c) gain from AC signal on VDD to output, and (d) gain from AC signal on ground to output.

Increasing the Input Common-Mode Voltage Range

It may be useful in some situations to have an input common-mode voltage range that extends close to the power supply rails. For our op-amp in Fig. 24.21, the allowable input common-mode voltage range is from (roughly) 450 mV to 900 mV (basically half of *VDD*). Towards the goal of extending the input common-mode voltage range, consider the addition of a PMOS diff-amp to Fig. 24.21, as seen in Fig. 24.28. The circuitry on the right side of the schematic is the op-amp in Fig. 24.21 drawn a little differently. Here, we've drawn the current source of the diff-amp as two parallel current sources. In Fig. 20.21 the current source biasing the NMOS diff-amp used MOSFETs with 100/2 sizes. Here we supply the same current but use two 50/2-sized current sources. Also, we've bumped up the size of the PMOS devices (M3, M4, and M7) by two. This is to accommodate the extra current supplied by the added PMOS diff-amp (the left side of the schematic). We could keep the PMOS devices the same size as seen in Fig. 21.21 and the

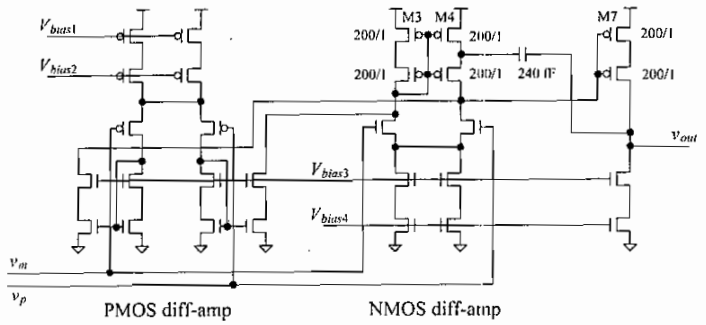


Figure 24.28 Two-stage op-amp of Fig. 24.21 with rail-to-rail input range.

circuit would still work fine (the V_{SG} voltages of the PMOS would simply be a little larger when both diff-amps are on). Two other notes: notice that the DC currents flowing in M3, M4, and M7 change depending on the input common-mode voltage. When the input common-mode voltage is large, the PMOS diff-pair is off; when it's small, the NMOS diff-pair is off. In the middle, both are conducting current. This means that the transconductance of the diff-amp will vary from g_{mn} to $g_{mn} + g_{mp}$ to g_{mp}. When both diff-amps are on (assuming $g_{mn} = g_{mp}$), the transconductance is twice as high as when a single diff-amp is on. This may require doubling the compensation capacitor. Finally, the change in the DC biasing conditions result in an offset that is a function of the input common-mode voltage.

Estimating Bandwidth in Op-Amp Circuits

When we finish designing an op-amp, its frequency response can be written using a single dominant-pole response (see Fig. 21.26 on page 680 with $A_{DC} = A_{OLDC}$) as

$$A_{OL}(f) = \frac{A_{OLDC}}{1 + j \cdot \frac{f}{f_{3dB}}} \tag{24.30}$$

For the op-amp response in Fig. 24.19, we can write

$$A(f) = \frac{1,000}{1 + j\frac{f}{10\,kHz}} \tag{24.31}$$

To estimate the bandwidth of a closed-loop op-amp circuit where $f >> f_{3dB}$, we can write Eq. (24.30) as

$$|A_{OL}(f)| \approx \frac{A_{OLDC}}{\frac{f}{f_{3dB}}} = \frac{A_{OLDC} \cdot f_{3dB}}{f} = \frac{f_{un}}{f} \tag{24.32}$$

noting the unity-gain frequency is calculated using

$$f_{un} = A_{OLDC} \cdot f_{3dB} \tag{24.33}$$

From this equation, we should see why the unity-gain frequency, f_{un}, is called the *gain-bandwidth product*. Note that for every increase in frequency by 10 (decade) above f_{3dB}, we get a decrease in A_{OL} by 10 (–20 dB). Alternatively, for every increase in frequency by 2 (octave) above f_{3dB}, we get a decrease in A_{OL} by 2 (– 6 dB).

The closed-loop bandwidth (which we'll call f_{3dBCL}) of the op-amp circuit cannot be larger than the bandwidth of the op-amp. Therefore, to estimate the bandwidth of a closed-loop op-amp circuit, assuming the op-amp is the limiting factor, we can write

$$A_{CL} \cdot f_{3dBCL} = f_{un} = \text{gain-bandwidth product} \tag{24.34}$$

If the op-amp circuit has a closed-loop gain of one (a follower configuration as seen in Fig. (24.7)), the bandwidth is f_{un}. The op-amp in Fig. 24.21 used in a follower configuration would have a bandwidth of (roughly) 100 MHz. If this op-amp were used in a closed-loop configuration with a gain of 10, then the bandwidth of the op-amp circuit would be estimated as 10 MHz.

24.2 An Op-Amp with Output Buffer

In the previous section, the low-frequency open loop gain of the op-amps in Figs. 24.8 or 24.21 was determined by a product of the diff-amp's gain, A_1, and the common-source's gain, A_2 or

$$A_{OLDC} = A_1 \cdot A_2 = \overbrace{g_{mn}(r_{o2} \| r_{o4})}^{A_1} \cdot \overbrace{g_{mp} r_{o7}}^{A_2} \tag{24.35}$$

which, from the simulations (see Fig. 24.19 for example) was 1,000. The small-signal resistance on the output of the op-amp, r_{o7}, is (from Table 9.2) 333 kΩ. If we were to connect a 10 kΩ resistor to the op-amp output, then A_{OLDC} would drop to 33. As seen in Fig. 24.1, we can add a buffer to the output of second stage to isolate it from a load resistance (or a large capacitance). We could use a source-follower (that has a voltage gain of 1), as seen in Figs. 21.46 or 21.48. However, this addition limits the output swing of the op-amp (which may be OK in some situations). Next we could try using a push-pull topology, as seen in Fig. 21.49. However, the voltage gain of the push-pull amplifier is >1 (see Eq. (21.116)) so if we were to include it in our basic op-amp of Fig. 24.21, we would have a three-stage op-amp (meaning all three stages have voltage gains greater than one). A three-stage op-amp can be challenging to compensate over production corners (and temperature). Also, as seen in Eq. (21.116), a small load resistor (kΩ) would still kill the push-pull amplifier's gain. To keep the gain high, let's try to use a first-stage topology with a large gain (increase A_1) while using a push-pull amplifier for the second stage. The gain of the second stage can still drop but, since the gain of the first stage is large, we still end up with a reasonable overall gain.

Towards the goal of increasing the first-stage gain, consider the op-amp schematic seen in Fig. 24.29. For the first stage, we've used the cascode diff-amp originally seen in Fig. 22.30. The gain of this diff-amp (now A_1) is 500 (from Tables 9.2, and 20.1 and Eq. (22.48)). The second stage of the op-amp is the topology seen in Fig. 21.50. Its gain depends on the load resistance it drives. If no load is connected to its output, then

$$A_2 = -\sqrt{10} \cdot (g_{mn} + g_{mp}) \cdot \frac{(r_{op} \| r_{on})}{10} = 10.1 \text{ V/V} \tag{24.36}$$

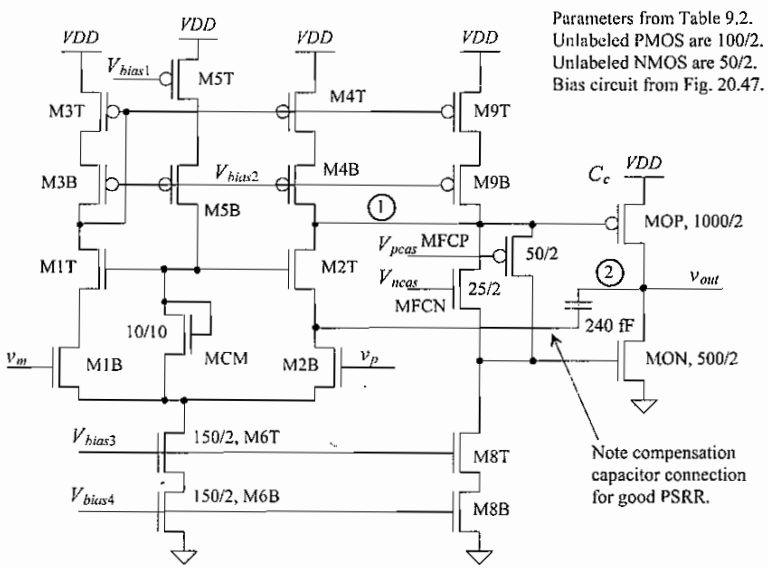


Figure 24.29 A CMOS op-amp with output buffer.

giving an overall low-frequency gain, A_{OLDC}, of 5,000 (= 74 dB). Note that the widths of MON and MOP are ten times wider than the values specified in Table 9.2 and so, as seen in Fig. 21.50, they conduct 100 μA of current. In Eq. (24.36) we multiplied the g_ms in Table 9.2 by the root of 10 $\left(g_{mon} = \sqrt{2KP_n \frac{10W}{L} \cdot 10\mu A} = \sqrt{10} \cdot g_{mn}\right)$ and divided the output resistances given in Table 9.2 by 10 $\left(r_{omon} = \frac{1}{\lambda 10 \cdot 10\mu A} = \frac{r_{on}}{10}\right)$.

Compensating the Op-Amp

We increased the gain of the first stage by increasing R_1 (the resistance at node 1 is now set by cascoded current sources). As seen in Eq. (21.65), this pushes the pole lower in frequency. As just described in Eq. (24.33), this decrease in f_1 (= f_{3dB}) and increase in low-frequency gain *doesn't have any effect* on the gain-bandwidth product (= f_{un}) of the op-amp. What this indicates, to a certain extent, is that we can still compensate the op-amp with a 240 fF capacitor as we did in the last section. To feed the indirect compensation current to node 1 we connect the compensation capacitor to the source of M2T, as seen in Fig. 24.29. This ensures good PSRR.

The open-loop response of the op-amp in Fig. 24.29 (without a load) is seen in Fig. 24.30. The phase-margin is roughly 70 degrees. The step response of the op-amp driving a 1k resistor and a 10 pF capacitor is seen in Fig. 24.31. The input signal is a pulse from 100 mV to 900 mV. Since the gain of the amplifier is −1 and the common-mode voltage is 500 mV, the output swings from 900 mV to 100 mV.

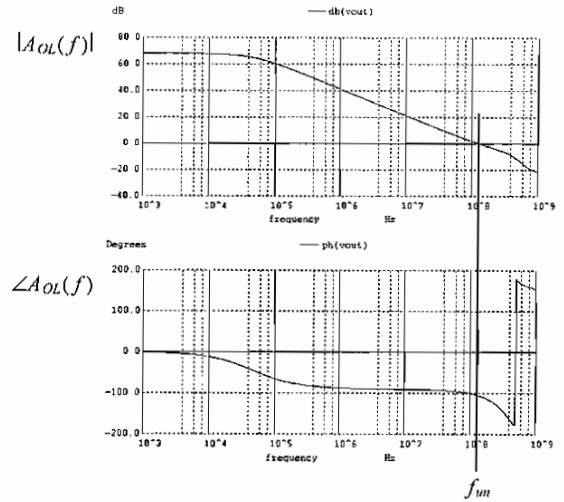


Figure 24.30 Open-loop response of the op-amp in Fig. 24.29.

This is a good point to remember and one of the fundamentals we presented in Ch. 9: that is, for high-speed design, we must use minimum length devices (see Eq. (9.59)). If we reduce all of the length of 2 (100 nm) devices in Fig. 24.29 to length of 1 (50 nm) devices, we can increase the speed of the op-amp. Figure 24.32 shows how the step response in Fig. 24.31 changes when we make this reduction in length. The big problem with using minimum channel lengths is the reduction in gain. Close inspection of Fig. 24.32 shows that the output is never quite pulled up to the correct voltage (it is either slightly below 900 mV or 100 mV). Simulating the AC open-loop response results in a gain of 44 dB (= 159). An output voltage of 900 mV would result in a difference on the inputs of the op-amp of 5.66 mV (0.9/159 = $v_p - v_m$).

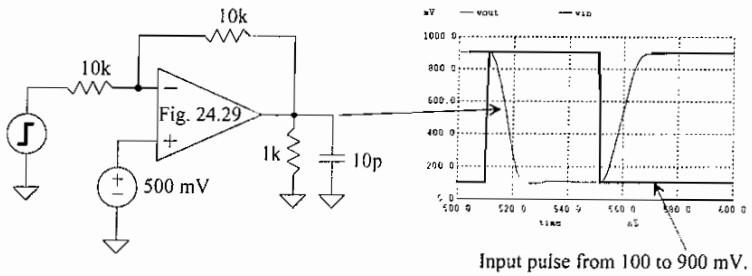


Figure 24.31 Step response of the op-amp in Fig. 24.29 driving 1k and 10 pF.

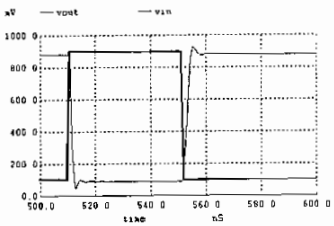


Figure 24.32 Step response of the amplifier in Fig. 24.29 when the lengths of the devices are reduced from 2 to 1 (from 100 nm to 50 nm). The load is still, as seen in Fig. 24.31, 1k and 10 pF.

24.3 The Operational Transconductance Amplifier (OTA)

The operational transconductance amplifier (OTA) can be defined as an amplifier where all nodes are low impedance except the input and output nodes. A simple example of an OTA is the diff-amp with current mirror load seen in Fig. 22.6. *An OTA without buffer can only drive capacitive loads.* A resistive load (unless the resistor is very large) will kill the gain of the OTA.

A sample OTA is shown in Fig. 24.33. Note that the basic op-amp in Fig. 24.2 is not considered an OTA because the drain of M4 is a high-impedance node and not the input or output of the amplifier. As seen in Fig. 24.33, except for the input and output nodes, all nodes have either a gate-drain-connected device or a source connected to them. The terminology "1:K" indicates that M4 and M5 can be sized K times wider (K > 1) than the other MOSFETs in the circuit. Assuming that $\beta_1 = \beta_2$, $\beta_{31} = \beta_{41}$, we observe that the current i_{d31} or i_{d41} is given by

$$-i_{d31} = i_{d41} = \frac{g_{mn}}{2}(v_p - v_m) = i_d \tag{24.37}$$

Furthermore, if $\beta_4 = K \cdot \beta_{41} = K \cdot \beta_{31} = K \cdot \beta_3$ and $K \cdot \beta_{51} = \beta_5$, then $i_{d4} = -i_{d5} = K \cdot i_{d41} = -K \cdot i_{d31}$. If the impedance of the capacitor is large compared to $r_{o4} \| r_{o5}$, then the output voltage of the OTA is given by

$$v_{out} = 2Ki_d(r_{o4} \| r_{o5}) \tag{24.38}$$

and the voltage gain is given by

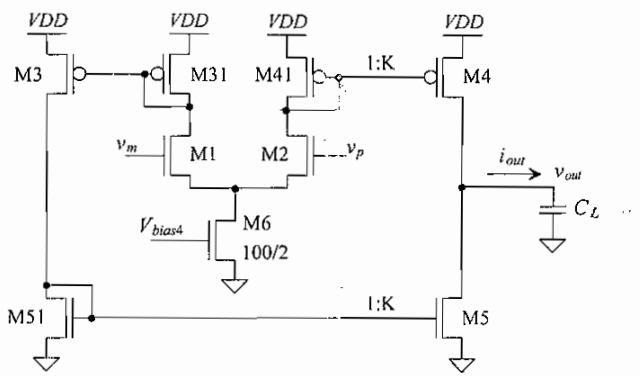
$$A_v = \frac{v_{out}}{v_p - v_m} = K \cdot g_m \cdot (r_{o4} \| r_{o5}) \tag{24.39}$$

noting that the noninverting input of the OTA is the gate of M2. While the derivation is interesting, as the name implies, we are interested in the transconductance of OTA. If the impedance of the capacitor is small compared to the OTA output resistance (at higher frequencies), we can write

$$i_{out} = i_{d4} - i_{d5} = 2Ki_d \tag{24.40}$$

The transconductance of the OTA is given by

$$g_{mOTA} = \frac{i_{out}}{v_p - v_m} = K \cdot g_m \tag{24.41}$$



Unless otherwise indicated, parameters from Table 9.2 with biasing circuit in Fig. 20.47.

Figure 24.33 Example of an operational transconductance amplifier (OTA).

Unity-Gain Frequency, f_{un}

To simulate the unity-gain frequency, f_{un}, (where the open loop gain is one) for the OTA, we can use the DC stability scheme seen in Fig. 24.34. We can write (for higher frequencies) the output voltage as

$$v_{out} = i_{out} \cdot \frac{1}{j\omega \cdot C_L} = g_{mn}v_{in} \cdot \frac{1}{j\omega \cdot C_L} \tag{24.42}$$

or

$$\frac{v_{out}}{v_{in}} = \frac{g_{mn}}{j\omega C_L} \rightarrow \left|\frac{v_{out}}{v_{in}}\right| = \frac{g_{mn}}{2\pi f \cdot C_L} \tag{24.43}$$

As before, we define the unity-gain frequency as the point where the open-loop gain is unity

$$\frac{g_{mn}}{2\pi f_{un} \cdot C_L} = 1 \rightarrow f_{un} = \frac{g_{mn}}{2\pi C_L} \tag{24.44}$$

For the OTA in Fig. 24.33 driving a 1 pF load capacitance, we can estimate the unity-gain frequency (with $K = 1$) as

$$f_{un} = \frac{150\ \mu A/V}{2\pi \cdot 1\,pF} = 24\ MHz$$

The simulation results are seen in Fig. 24.34. The location of the pole is estimated using

$$f_{3dB} = \frac{1}{2\pi(r_{o4} \| r_{o5})C_L} \tag{24.45}$$

For the OTA in Fig. 24.33, f_{3dB} = 1.4 MHz using the values in Table 9.2. The low-frequency gain is $g_m(r_{o4} \| r_{o5})$ = 16.65 V/V (= 24.4 dB).

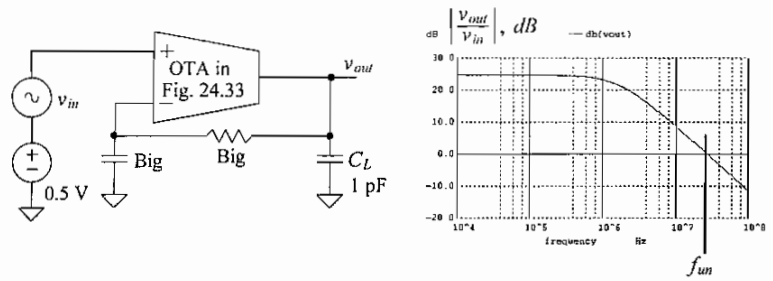


Figure 24.34 Simulating the open-loop response of the OTA in Fig. 24.33.

Note that the maximum value of i_{out} is KI_{SS} (the drain current of M6 multiplied by the increase in widths, K, in the current mirrors M41/M4 and M51/M5). This is important to understand because using an OTA for the first stage of an op-amp results in the same slew-rate limitations of I_{SS} driving C_c that we had when using a diff-amp.

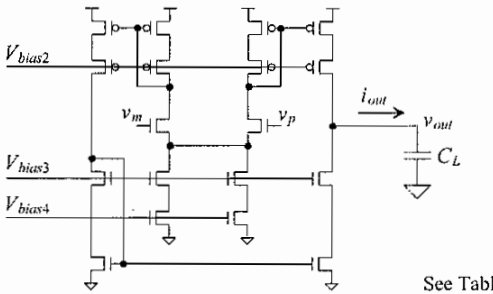
Increasing the OTA Output Resistance

The ideal OTA has infinite output resistance. All of i_{out} flows in the external capacitive load and none flows in the OTA's own output resistance (which is $r_{o4}||r_{o5}$ for the OTA in Fig. 24.33). Towards increasing the OTA output resistance, consider the configuration seen in Fig. 24.35. This topology is based on the one seen in Fig. 24.33 except that now we've cascoded the current mirrors. Note how we've drawn the diff-amp's tail current source as four MOSFETs instead of two with twice the width (this is the way we would lay it out too). The low-frequency gain of the OTA is now increased to

$$A_v = g_{mn} \cdot (R_{ocasn}||R_{ocasp}) \tag{24.46}$$

The 3-dB frequency is now

$$f_{3dB} = \frac{1}{2\pi(R_{ocasn}||R_{ocasp}) \cdot C_L} \tag{24.47}$$



See Table 9.2 and Fig. 20.47.

Figure 24.35 A cascode OTA circuit (higher output resistance).

As discussed before (Sec. 24.2 in the discussion concerning compensating the op-amp), when we get a decrease in the 3-dB frequency and an increase in the low-frequency gain, the effects cancel and the gain-bandwidth product (f_{un}) remains constant. The unity-gain frequency is still given by Eq. (24.44). For the OTA in Fig. 24.35, we can estimate the low-frequency gain, using Tables 9.2 and 20.1, as

$$A_v = (150) \cdot (16.6||4.2) = 500\ (54\ dB)$$

and the 3-dB frequency, driving a 1 pF load, is

$$f_{3dB} = \frac{1}{2\pi(16.6||4.2 \times 10^6)(1\ pF)} = 47\ kHz$$

Simulation results are seen in Fig. 24.36. The PM is roughly 85°.

An Important Note

Note, in Eq. (24.44), that increasing the load capacitance, decreases the unity-gain frequency, making the OTA more stable. Unlike the two-stage op-amps discussed in the first two sections of this chapter, where the op-amp can become unstable with a large load capacitance, the OTA only becomes more stable with large load capacitance. Also, the PM approaches 90° as the load capacitance increases. The step response of the OTA has a first-order shape (just as in an RC circuit). In many on-chip applications, the load is purely capacitive and so the OTA works great as an "op-amp" in a closed-loop configuration (thus we'll often call OTAs op-amps). Note that adding a second stage to the basic OTA forms a (two-stage) op-amp like those discussed earlier. We can use the methods already presented to compensate the resulting structure (but, again, driving a load capacitance that's too large with these two-stage structures can result in instability).

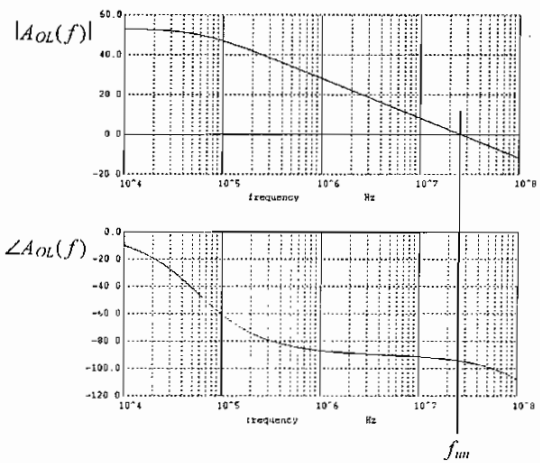


Figure 24.36 The open-loop gain and phase response for the OTA in Fig. 24.35.

OTA with an Output Buffer (An Op-Amp)

If we use the OTA in Fig. 24.35 (or Fig. 24.33) to drive a 1 pF load, we're limited to charging/discharging the capacitor at a rate of

$$\frac{I_{SS}}{C_L} = \frac{dv_{out}}{dt} = \frac{20\ \mu A}{1\ pF} = \frac{20\ mV}{ns} \qquad (24.48)$$

For the OTA's output to change from ground to *VDD* (1 V) requires (just due to slewing) 50 ns. This length of time (the slew-rate limited portion of the settling time) may be fine for many applications. However, for high-speed design, we want faster settling times. Looking at Eq. (24.48), we see that we can decrease the settling time by increasing I_{SS}. However, as discussed in Ch. 9, this causes the gain of the circuit to decrease. To avoid the reduction in gain, we can make a corresponding increase in the size of the devices to keep the overdrive voltage constant. But this is doing nothing more than effectively reducing the size of the load capacitor (the load capacitance becomes comparable in size to the parasitic transistor capacitances).

To speed up the circuit, we may add an output stage, Fig. 24.37. The added stage is a common-source amplifier (with a negative gain indicating **we swap the inverting and noninverting terminals** of the op-amp or else we waste a lot of time). While M7 can turn on and charge a load capacitor quickly, the current source, M8, still limits the rate of load capacitance discharge. For this reason, we've bumped up the current in the output stage to 100 µA. The open-loop simulations driving a 1-pF load are seen in Fig. 24.38. The PM is 45° (too low if the op-amp will be used in a unity follower configuration). Figure 24.38 shows the step response of the op-amp.

We should compare this figure to the response we saw in Fig. 24.31. The op-amp in Fig. 24.31 is driving a heavier load but still has cleaner settling behavior. Further, a comparison of the current pulled from *VDD* reveals, under the same loading and operating conditions, that the op-amp in Fig. 24.37 pulls more current (see Fig. 24.39). The main difference in these two op-amps is the output buffer used. Let's consider adding a class AB buffer to the OTA in Fig. 24.35 (like the one used in the op-amp of Fig. 24.29). Since we are only driving a 1 pF load, we'll leave the output devices, MON

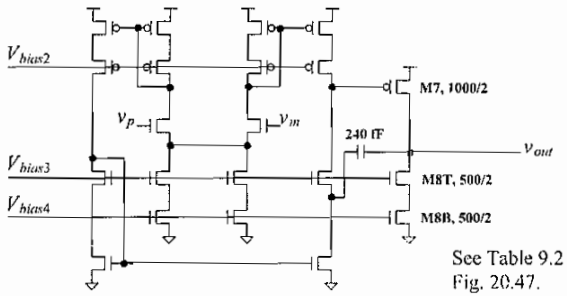


Figure 24.37 A cascode OTA circuit with common-source output buffer (an op-amp). Note the inverting and noninverting terminals are swapped from Fig. 24.35.

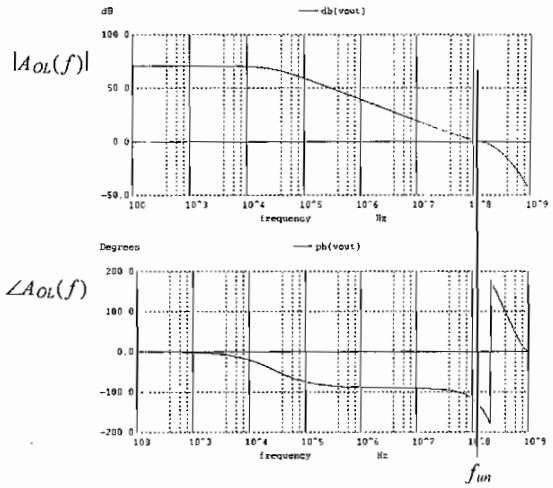
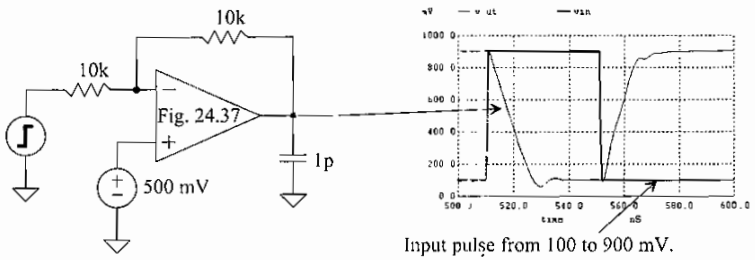


Figure 24.38 Open-loop response of the op-amp in Fig. 24.37 driving a 1-pF load.



Current from VDD. Includes the bias circuit's (Fig. 20.47) current (140 uA).

Figure 24.39 Step response of the op-amp in Fig. 24.37 driving a 1 pF.

and MOP, at the same size as the other MOSFETs in the circuit (PMOS are 100/2 and NMOS are 50/2). The resulting op-amp is seen in Fig. 24.40. The step response (in the same configuration used to generate Fig. 24.39) is seen in Fig. 24.41. Notice that the current pulled from *VDD* by the op-amp in Fig. 24.40 is 100 µA less than the current pulled by the op-amp in Fig. 24.37. This lower current is the result of using a class AB output stage.

The performance of the op-amps in Figs. 24.37 and 24.40, with regard to high-speed operation and settling time, won't be quite as good as the topology in Fig. 24.29 (assuming the same biasing conditions, device sizes, and overdrive voltages). To understand why, look at the path, in Fig. 24.40, that the noninverting op-amp input, v_p, takes to get to the op-amp output, v_{out} (that is, M1 - M31 - M3 - M51 - M5 - MON). The delay through this path is longer than the delay, in Fig. 24.29, through M2 - MOP. Further, for high speed, M5 of the op-amp in Fig. 24.40 must turn on (above its quiescent current level) to pull the gate of MON down. For the op-amp in Fig. 24.29, the fact that both the gate of M9T and MOP are driven (in opposite directions) by the diff-amp allows us to use a simple pull-down current (M8) without losing speed. The issues with the cascode diff-amp-based, op-amp topology of Fig. 24.29 are the more limited input common-mode range and the limited voltage swing of node 1. Towards getting better input common-mode voltage range and output swing as well as improved high-speed operation, let's discuss the folded-cascode OTA (see Fig. 22.5).

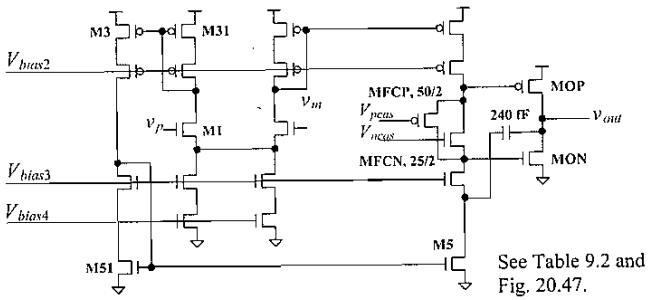
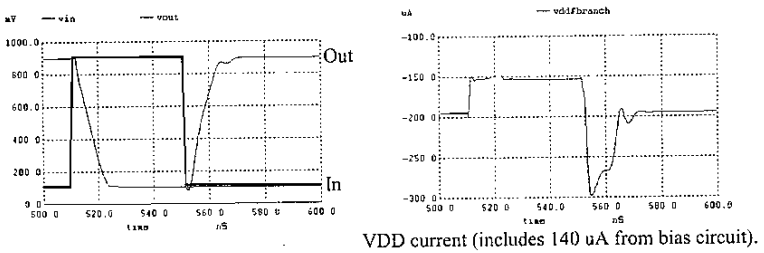


Figure 24.40 Using a class AB output stage with the OTA.



VDD current (includes 140 uA from bias circuit).

Figure 24.41 The step response and current in the topologies seen in Fig. 24.39 using the op-amp in Fig. 24.40.

The Folded-Cascode OTA and Op-Amp

An example of an NMOS diff-amp-based folded cascode OTA is seen in Fig. 24.42. It is assumed that the reader understands the biasing and operation of the circuits seen in Figs. 20.45, 20.46, and 22.5. To avoid labeling MOSFETs with twice the width, we've drawn the schematic in Fig. 24.42 with MOSFETs in parallel in the branches that supply or sink more current (for example, the diff-amp's tail current). All MOSFETs in Fig. 24.42 conduct 10 µA of current under equilibrium conditions. Before moving on, it may be a good idea to pause and make sure that the reader understands how the currents sum in this OTA.

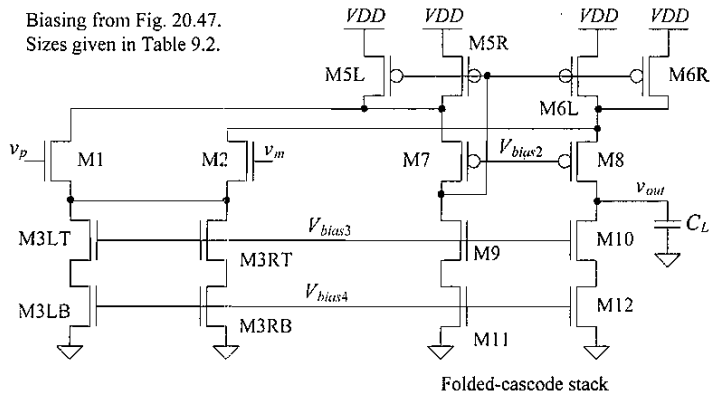


Figure 24.42 A folded-cascode OTA.

To qualitatively describe the operation of the OTA, let's look at what happens when v_p goes high (above v_m). This causes M1 to turn on and M2 to shut off. When M1 turns on, it pulls the drain of M5 down, shutting M7 off. As M7 shuts off, M9 and M11 pull the gate of M5 down. With the gate of M5 being pulled down, the current in M6 increases. At the same time, because the current in M2 is decreasing, the current in M8 is increasing and the output voltage increases. The maximum current the OTA output can supply (out of the OTA to the load) is in the neighborhood of 20 µA, while the maximum current the OTA can sink (into the OTA) is 10 µA (set by the current sink M10/M11).

The same AC equations presented at the beginning of this section also apply to this OTA. For example, the AC drain currents of M1 and M2 can be written as

$$i_{d1} = -i_{d2} = g_{mn} v_{gs1} = -g_{mn} v_{gs2} \tag{24.49}$$

which, again, indicates $v_{gs1} = -v_{gs2}$. However, we know

$$v_p - v_m = v_{gs1} + (-v_{gs2}) \tag{24.50}$$

so

$$i_{d1} = i_{d5} = i_{d6} = (v_p - v_m) \cdot \frac{g_m}{2} \tag{24.51}$$

noting that the AC current through M9/M11 (and thus M7) is ideally zero. The current through M8 is then

$$i_{d8} = i_{d6} - i_{d2} = 2 \cdot i_{d1} \tag{24.52}$$

and thus the output voltage (again, the AC current in M10/M11 is ideally zero) is

$$v_{out} = i_{d8} \cdot (R_{ocasn} || R_{ocasp}) \tag{24.53}$$

and thus the gain is

$$A_v = \frac{v_{out}}{v_p - v_m} = g_{mn} \cdot (R_{ocasn} || R_{ocasp}) \tag{24.54}$$

This result should be compared to Eq. (24.46). Equation (24.47) gives the 3-dB frequency of the OTA's open-loop response, while Eq. (24.44) gives the unity-gain frequency. Figure 24.43 shows the open-loop response of the folded-cascode OTA driving a 1 pF load capacitance.

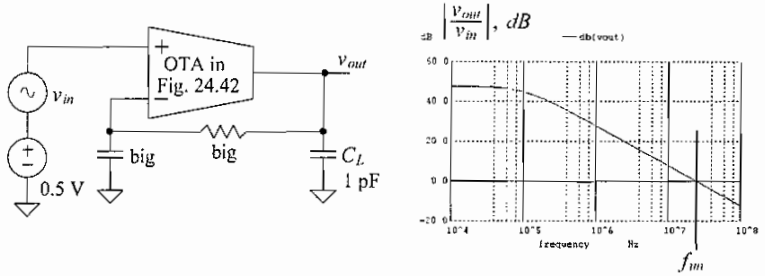


Figure 24.43 Simulating the open-loop response of the OTA in Fig. 24.42.

Figure 24.44 shows the schematic of a folded-cascode op-amp using a class AB output buffer. We've added floating current sources to equalize the voltages across M9/M10 (to minimize the input-referred offset). In the open-loop response of the op-amp (see Fig. 24.45), the load of the op-amp is a 1 pF capacitor. The step response is seen in Fig. 24.46, using the topology of Fig. 24.39 (gain of −1 driving 1 pF and the 10k feedback resistor). Reviewing the previous op-amp responses, we don't see much difference in performance. We can't seem to push the gain-bandwidth product, f_{un}, much above 100 MHz. Again, the only way to increase the speed is to reduce the channel length and/or increase the overdrive voltage.

Figure 24.47 shows what happens if we reduce the lengths of the MOSFETs used in the op-amp in Fig. 24.44 from 2 (100 nm) to 1 (50 nm). The gain-bandwidth product jumps to 400 MHz. However, the low-frequency gain drops to approximately 50 dB. The step response is clean, and the settling time is approximately 5 ns. While we can increase the speed further by redesigning the bias circuit and increasing the currents and overdrive voltage, let's, instead, concentrate on increasing the low-frequency gain of the op-amp. This won't increase the gain-bandwidth product of the op-amp, but it's necessary for precision amplification. To *enhance the gain,* we'll try to increase the OTA output resistance (R_1 or the resistance on node 1) by regulating the drain in the cascode current sources. We've already discussed this technique (see Fig. 20.40), but we haven't applied it to amplifiers. Before leaving this section, let's present one more example.

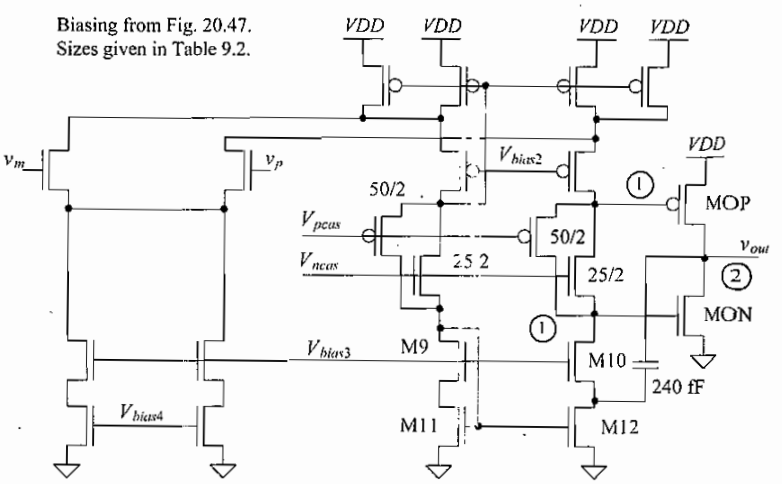


Figure 24.44 Folded-cascode op-amp with class AB output buffer.

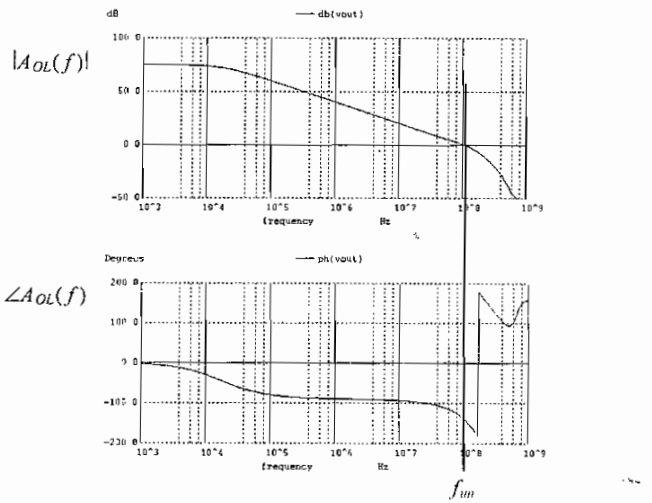


Figure 24.45 Open-loop response of the op-amp in Fig. 24.44 driving a 1-pF load.

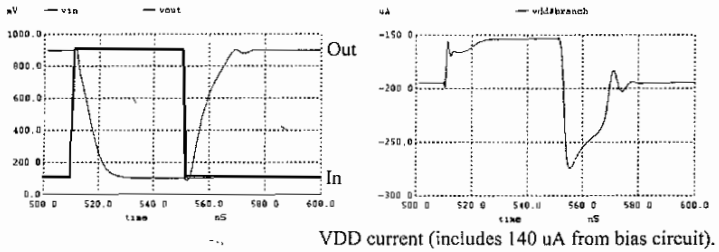


Figure 24.46 The step response and current in the topologies seen in Fig. 24.39 using the op-amp in Fig. 24.44.

Figure 24.48 shows a wide-swing op-amp based on the topology seen in Fig. 24.44. To get wide-swing operation (which means that the input common-mode voltage extends beyond the power supply rails and the op-amp still functions), we added a PMOS diff-amp stage. Further, to sink the additional current supplied by the PMOS diff-amp when it's on, we add two NMOS devices at the bottom of the folded-cascode stack. When, for example, the PMOS diff-amp shuts off with the input common-mode voltage moving towards *VDD*, the added two devices don't really affect the circuit's operation. The only result is that the V_{GS} of the bottom NMOS devices is slightly smaller when the PMOS diff-amp shuts off. A similar conclusion can be drawn when the NMOS diff-amp shuts off with the common-mode voltage moving towards ground. Note that when both diff-amps are on, the first-stage transconductance becomes, $g_{mn} + g_{mp}$. This means the gain-bandwidth product, f_{un}, moves to $(g_{mn}+g_{mp})/2\pi C_c$. The value of the compensation

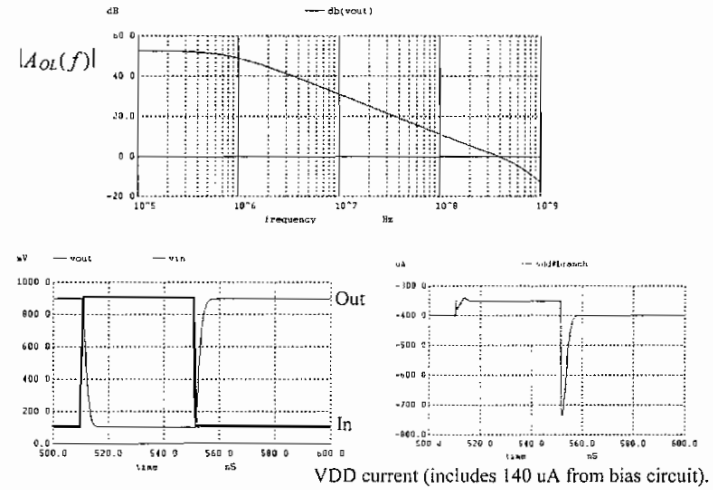


Figure 24.47 Reducing the length of the MOSFETs used in the op-amp in Fig. 24.44 from 2 to 1 and resimulating open-loop and step responses.

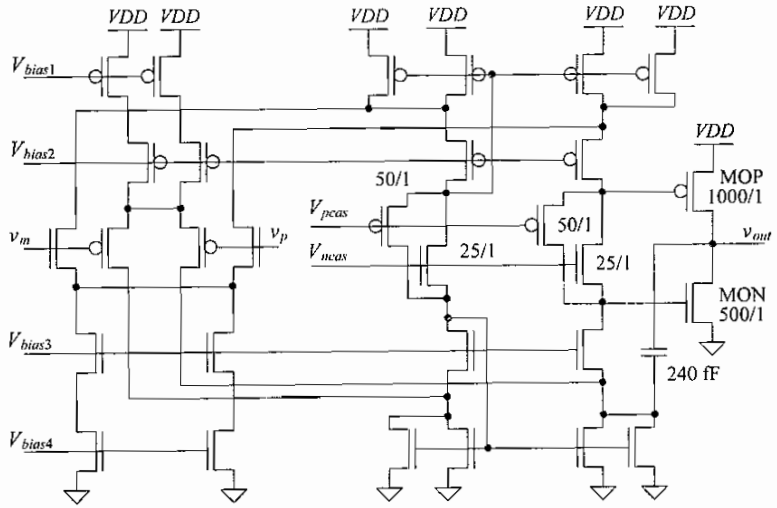


Figure 24.48 An op-amp with an input common-mode range that extends beyond the power supply rails and that can drive heavy loads.

capacitor may need to be increased. When only one diff-amp is on, the transconductance of the other diff-amp goes to zero, resulting in a smaller gain-bandwidth product. The issue here is that the gain of the op-amp is changing with variations in the input common-mode voltage. This, as mentioned before, can lead to distortion and is the reason that the inverting topology, as seen in Fig. 24.39, is preferred (noting that the input common-mode voltage is fixed when using the inverting topology).

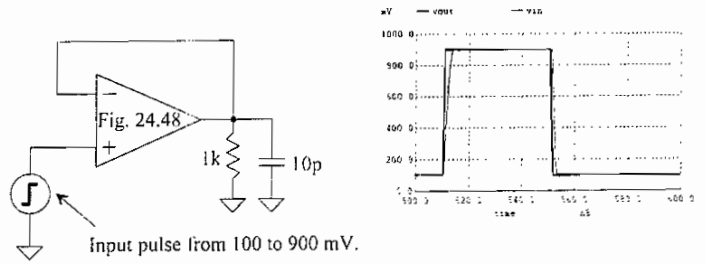


Figure 24.49 Step response of the op-amp in Fig. 24.48 driving 1k and 10 pF. Note that the op-amp is in the follower configuration and that the input common-mode voltage is changing.

Finally, in some applications, an op-amp that can supply significant amounts of current, say 100 mA, may be needed. To get such a large current, we may need to increase the widths of the output MOSFETs in the push-pull amplifier to, for example, 10,000 and 5,000 (for the PMOS and NMOS devices, respectively). The problem with this is that the quiescent current that flows in these devices, using the op-amp configuration in Fig. 24.44 as an example, is 1 mA. To reduce this current, we can increase the lengths of the floating current sources used in the folded-cascode section. This increases their gate-source voltage drops and moves the gates of the push-pull MOSFETs towards the power supply rails (shutting them off). Care must be exercised when doing this because shutting off the output MOSFETs can result in moving g_{m2} towards zero, causing, as indicated in Eq. (24.24), f_2 to move down in frequency, thus making the op-amp unstable.

24.4 Gain-Enhancement

While using minimum channel lengths (L = 1) results in the fastest op-amps, the open-loop gain, A_{OLDC} , is considerably reduced. Using gain-enhancement (GE) techniques (regulating the drain node in a cascode structure, see Fig. 20.40), we boost the low-frequency gain by increasing the cascode output resistance. However, the gain-bandwidth product, f_{un} , is still $g_m/2\pi C_c$. (We get the speed by using a smaller compensation capacitor and/or a larger diff-amp transconductance.) Note, in Fig. 24.47 for example, that our gain-bandwidth product went up by the increase in g_m with the reduction in the channel length. Reviewing Eqs. (24.30)–(24.34), we see that if A_{OLDC} goes up, f_{3dB} goes down (and f_{un} is a constant).

In the following discussion, we only concern ourselves with two-stage op-amps in which we are using GE to compensate for the low gain caused by our minimum channel lengths. While GE can be used in an OTA, the fact that the speed of the OTA is limited by the capacitance it is driving keeps them from being as fast as the two-stage op-amps we've discussed (unless, of course, the load capacitance is small).

As seen in Fig. 20.40, we need an additional amplifier to help regulate the drain of the cascode device. Consider the two differential amplifiers (OTAs) seen in Fig. 24.50. The source-followers on the inputs of the diff-amps are used to allow for the amplification of signals near ground (for the P amplifier) or *VDD* (for the N amplifier). The gains of the diff-amps are

$$A_P = g_{mp} \cdot (r_{on} || r_{op}) \text{ and } A_N = g_{mn} \cdot (r_{on} || r_{op}) \tag{24.55}$$

To simplify the equations, we'll assume

$$A_{GE} = A_P = A_N \tag{24.56}$$

The low-frequency gain of the op-amp in Fig. 24.44 is

$$A_{OLDC} = \overbrace{g_{mn} \cdot (R_{ocasn} || R_{ocasp})}^{\text{diff-amp gain}} \cdot \overbrace{(g_{mp} + g_{mn}) \cdot (r_{on} || r_{op})}^{\text{push-pull output stage}} \tag{24.57}$$

Using Eq. (20.67), we can write the open-loop gain *with gain-enhancement* as

$$A_{OLDC,GE} = A_{OLDC} \cdot A_{GE} \tag{24.58}$$

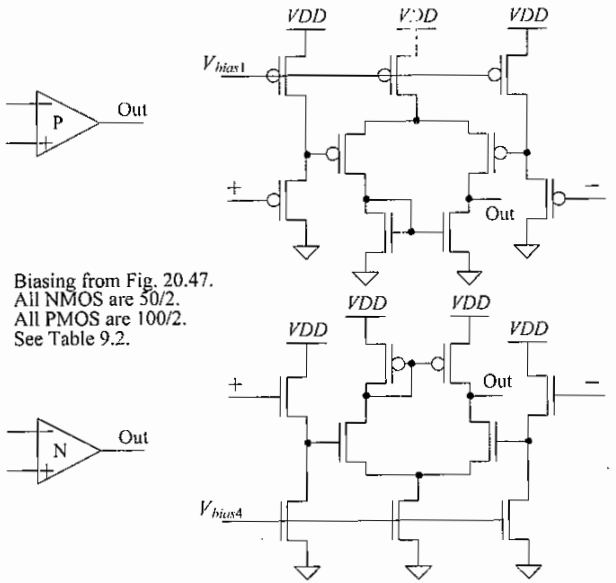


Figure 24.50 Diff-amps with source-follower level shifters for use in GE.

The open-loop gain is increased by the gain of the added amplifier. Figure 24.51 shows the op-amp of Fig. 24.44 with GE.

The simulation results in Fig. 24.47 were generated using the op-amp of Fig. 24.44 with the lengths reduced from 2 to 1. The low-frequency, open-loop gain was 53 dB. In Fig. 24.51 we added GE and used the resulting op-amp to generate the simulation results seen in Fig. 24.52 (with the GE compensation capacitors, C_{cGE} , set to zero). The low-frequency, open-loop gain is now 74 dB. The step response is seen as well. The current pulled from *VDD* is now 60 µA higher than the op-amp without GE. By replacing the diff-amps in the N and P amplifiers in Fig. 24.50 with folded-cascode OTAs (like the one seen in Fig. 24.42), we can get a greater increase in open loop gain. However, the stability of the regulated-drain feedback loop becomes even more important.

Bandwidth of the Added GE Amplifiers

Let's model the frequency response of the added amplifiers in Fig. 24.50 by

$$A_{GE}(f) = \frac{A_{GEDC}}{1 + j\frac{f}{f_{3dBGE}}} \tag{24.59}$$

Using this equation with Eqs. (24.30) and (24.58), we can write

$$A_{OLGE}(f) = \frac{A_{OLDC}}{1 + j \cdot \frac{f}{f_{-3dB}}} \cdot \overbrace{\frac{A_{GEDC}}{1 + j \cdot \frac{f}{f_{-3dBGE}}}}^{A_{GE}(f)} \tag{24.60}$$

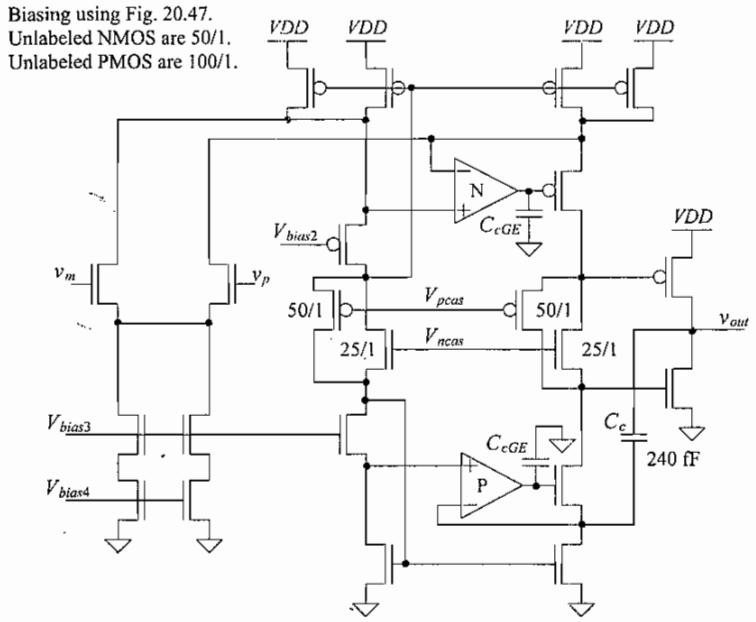


Figure 24.51 Folded-cascode op-amp with class AB output buffer and gain-enhancement.

However, the pole ($f_1 = f_{3dB}$) on the output of the folded-cascode OTA can be written, see Eqs. (20.67) and (24.47), with GE as

$$f_{3dB} \approx \frac{1}{2\pi(R_{ocasn} \| R_{ocasp}) \cdot A_{GE}(f) \cdot C_c} \qquad (24.61)$$

assuming the current through C_c is larger than the other currents, the output of the OTA may have to charge. If we substitute Eq. (24.61) into Eq. (24.60) and look at frequencies much larger than f_{3dB} (which from Fig. 24.52 is only a 100 kHz or so), we see that the frequency response of the added GE amplifiers cancels out of $A_{OLGE}(f)$. What this means is that the bandwidth of the added amplifiers doesn't need to be wide. As long as the bandwidths are larger than the f_{3dB} of the op-amp, the GE works as desired. Also, notice that when we substitute Eq. (24.61) into Eq. (24.60), the resulting transfer function shows a *doublet* (a pole and zero at the same frequency). If the pole and zero aren't at exactly the same frequency, then the settling time of the amplifier can be affected by the change in phase shift (noting that in Eq. (20.67) we are approximating the output resistance of the cascode structure). The output resistance of the cascode structures is decreasing with increasing frequency, while, at the same time, the decrease in the output resistance results in an increase in the circuit bandwidth (causing the gain-bandwidth product to be a constant).

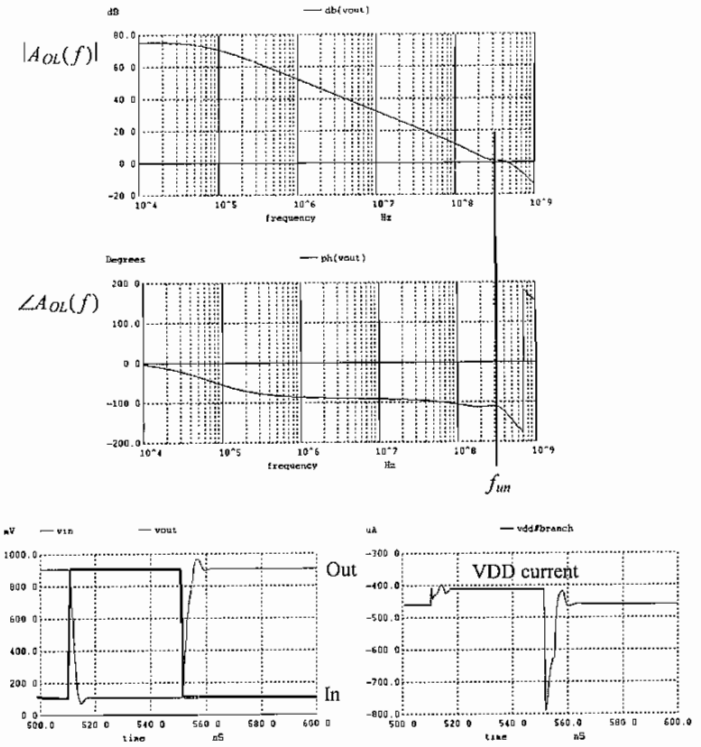


Figure 24.52 Operation of the op-amp in Fig. 24.51 driving a 1-pF load.

Compensating the Added GE Amplifiers

When we generated the simulations in Fig. 24.52, we didn't have compensation capacitors on the output of the added GE OTAs (N and P); that is, C_{cGE} were zero. The problem with this is that the resulting feedback loops aren't necessarily stable. As seen in Fig. 24.52, the step response shows some overshoot (indicating a low phase margin). Also, the frequency response isn't as monotonic as we would like (–20 dB/dec) around f_{un}. As with any feedback structure, compensation is important. *So we need to add capacitors to the outputs of the GE OTAs.* Figure 24.53 shows the response of the op-amp in Fig. 24.51 when C_{cGE}s are set to 240 fF (the GE OTAs then have f_{un} of 100 MHz, as calculated using Eq. (24.44)). The frequency response is more monotonic and the step response doesn't show overshoot and ringing. The gain-bandwidth product of the op-amp is 400 MHz. Note, it may be useful to regenerate Fig. 24.53, using larger values of C_{cGE}, to prove to oneself that indeed the bandwidth of the added GE amplifiers isn't critical. Also, it may be useful to replace the basic diff-amps with higher-gain OTAs to show that the low-frequency gain of the op-amp does indeed increase as Eq. (24.58) shows.

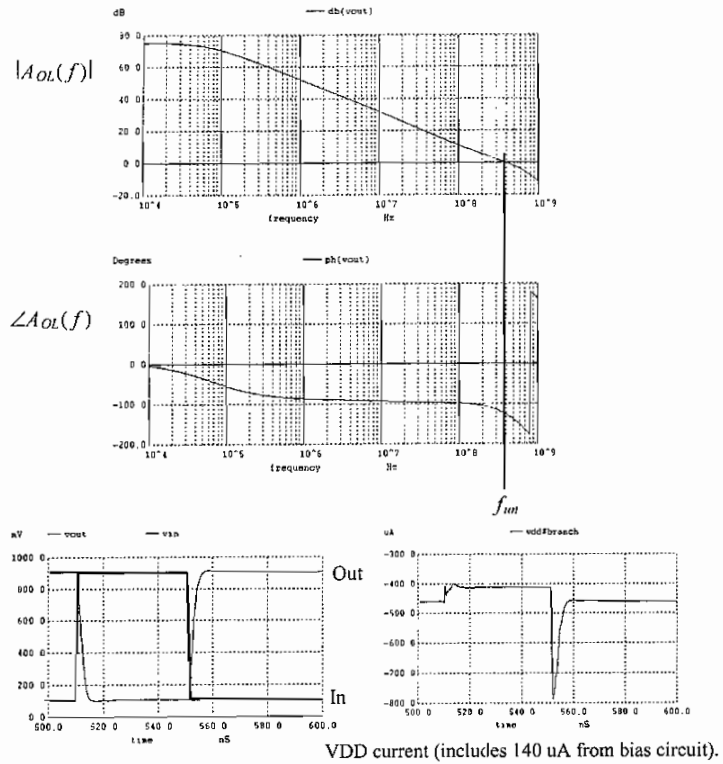


Figure 24.53 Operation of the op-amp in Fig. 24.51 driving a 1-pF load with GE compensation capacitors of 240 fF.

24.5 Some Examples and Discussions

A Voltage Regulator

One of the applications of an op-amp is in regulating a voltage on-chip. Figure 24.54 shows the basic topology. A voltage reference is used with the op-amp to generate a regulated voltage, V_{REG}. If the voltage reference is stable with temperature, the fact that the V_{REG} is a function of a ratio of resistors (so process or temperature changes in the resistance value don't affect the ratio) and the variation in the op-amp's open loop gain is desensitized using feedback makes the regulated voltage stable with process and temperature changes. The ideal (meaning that the op-amp has infinite open-loop gain) regulated voltage is

$$V_{REG} = V_{REF} \cdot \left(1 + \frac{R_A}{R_B}\right) \tag{24.62}$$

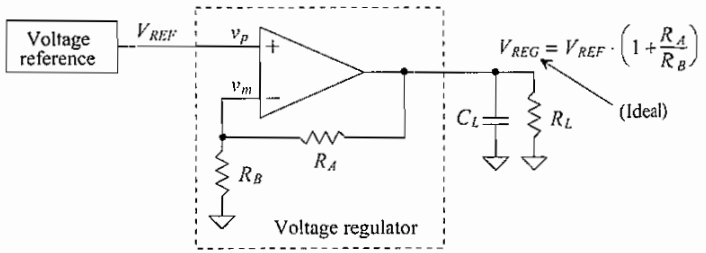


Figure 24.54 Schematic of a voltage regulator.

If the op-amp's open-loop gain is finite, then we can write

$$V_{REG} = A_{OL} \cdot (v_p - v_m) \tag{24.63}$$

and

$$v_m = V_{REG} \cdot \frac{R_B}{R_A + R_B} \text{ and } v_p = V_{REF} \tag{24.64}$$

Solving for the actual regulated voltage, gives

$$V_{REG} = V_{REF} \cdot \frac{1}{\frac{1}{A_{OL}} + \frac{R_B}{R_A + R_B}} \tag{24.65}$$

noting that when A_{OL} is infinite (or very large) this equation simplifies to Eq. (24.62). Equation (24.65) can be very revealing. Let's say that V_{REF} is 500 mV, R_B is an open, and R_A is a short (the op-amp is a voltage follower). Ideally, V_{REG} is also 500 mV. If we use a diff-amp with an open-loop gain of 20, then the actual regulated voltage is

$$V_{REG} = 0.5 \cdot \frac{1}{1.05} = 476\ mV \tag{24.66}$$

This simple example illustrates *why open-loop gain is important*. When op-amps are used in data converters (discussed in Chs. 28 and 29), they must amplify signals to within mV, which requires op-amp open-loop gains in the thousands.

Figure 24.55 shows a basic regulator topology using the two-stage op-amp. The regulator only sources current and so we've made the PMOS device, on the output, very large. One concern with using the large PMOS device is the op-amp offset (which can be considerable, see Fig. 24.4, for example). Another concern occurs when simulating. In some simulators both NRD and NRS (number of squares of implant area in the source or drain of a MOSFET) defaults to one. If the sheet resistance, rsh, is 50 ohms, then the resistance in series with the MOSFET is 100 ohms (limiting the current the MOSFET can supply). For large MOSFETs, it is generally a good idea to set NRD and NRS to zero. An example of a (large) SPICE MOSFET statement is

```
Mbig   vout   vn1   VDD   VDD   PMOS  L=1   W=10000  NRD=0   NRS=0
```

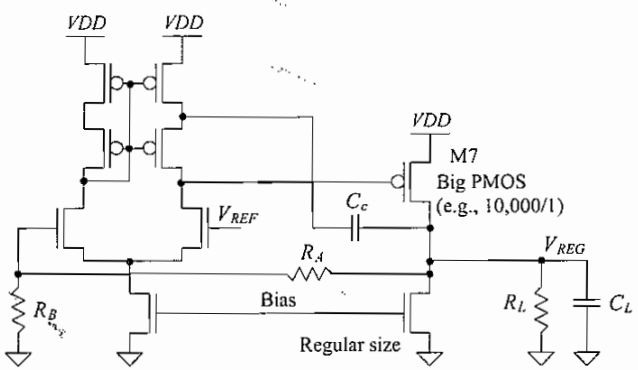


Figure 24.55 Schematic of a voltage regulator.

The point of making M7 big was to ensure that the regulator could supply a significant amount of current. However, notice that the gate of M7 cannot swing any lower (assuming that all MOSFETs remain in saturation) than

$$V_{G7,\min} = V_{REF} - V_{THN} \tag{24.67}$$

For example, if V_{REF} is 500 mV and V_{THN} is 250 mV, then the minimum voltage on the gate of M7 is 250 mV (25% of *VDD*). Ideally, we would like the gate of M7 to be able to swing to ground, fully turning it on. Towards this goal, consider the modified voltage regulator seen in Fig. 24.56. In this figure, we've used the OTA from Fig. 24.33 for the first stage of the op-amp. Now, when M5 turns on and M4 shuts off, the gate of M7 can get yanked down towards ground, allowing M7 to turn fully on.

In any practical regulator, a bypass capacitor (a big capacitor placed from V_{REG} to ground) is used to supply charge to the load for fast current transients. The required bandwidth of the regulator is reduced since the bypass capacitor (part of C_L in Figs. 24.55 and 24.56) takes care of the fast transients. It is desirable to have a large C_L for "filtering" the load's fast current needs. However, from our discussions earlier, this (large load capacitance) makes compensating the op-amp more challenging. In all of our compensation schemes, the pole associated with the output of the first stage was at a frequency much lower than the pole associated with the output stage (this was one of the reasons we used pole-splitting). To isolate the load from the second-stage gain, we might insert an NMOS source-follower (SF) between the load and the output of the second stage. The problem with this is that we won't be able to turn the SF on enough to supply significant amounts of current. Next, we might try replacing the common-source amplifier (M7) with a class AB stage (like the topology seen in Fig. 24.29). Here, again, the output won't be able to supply as much current as what we get with a full *VDD* across M7's V_{SG}. (Note that one of the reasons for using a voltage regulator is to hold V_{REG} constant even if *VDD* approaches V_{REG}.) If we must use a large load capacitor to supply charge for the fast variations in the load current and if the topology in Fig. 24.56 is used, then *we must try to compensate the op-amp with the load capacitance.*

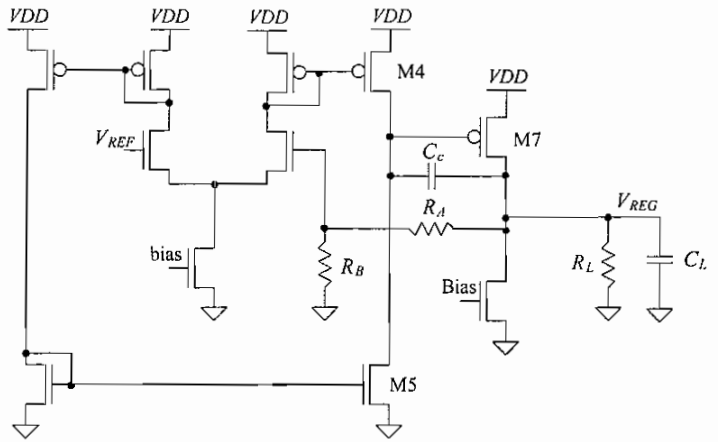


Figure 24.56 Schematic of a voltage regulator where the gate of M7 can be pulled all the way to ground.

If we are compensating the op-amp (the voltage regulator) using the load capacitance, then we've got to make sure that there is a minimum value of C_L. Further, increasing the load capacitance should make the regulator more stable, slowing the response of the op-amp. The reduction in the speed of the op-amp is offset by the fact that a larger load capacitance can supply more charge to a fast current transient before the op-amp must respond.

Why use a compensation capacitor, C_c, if we are compensating with the load capacitance? The compensation capacitor pushes the pole associated with the output node to a higher frequency (exactly what we don't want to do). We still use the capacitor (but with a smaller value) for large-signal reasons. If V_{REG} drops suddenly, the decrease in voltage is fed back directly to the gate of M7 through C_c. This turns M7 (quickly) on and allows it to pull V_{REG} back up (bypassing the slower feedback action of the op-amp). Note then that C_c is *not used* for compensating the op-amp (and actually makes the regulator more unstable). Further note that increasing the gain of the op-amp by making the first-stage gain larger (by, say, using a cascode diff-amp to increase the output resistance of the diff-amp, R_1) also hurts the regulator's stability. (By using a larger R_1, the pole associated with the output of the first stage is moved lower in frequency.)

To estimate the location of the output pole, assuming that it is the dominant pole in the system, let's calculate the closed-loop output resistance of the op-amp $R_{out,CL}$ (the open- loop output resistance is R_2). We know that the low-frequency, open-loop gain of the op-amp is

$$A_{OLDC} = \frac{v_{out}}{v_p - v_m} = g_{m1}R_1 g_{m2}R_2 \tag{24.68}$$

If we replace the load in Fig. 24.54 with a test voltage, v_t, and note that V_{REF} (= v_p = 0) is AC ground, then with the help of Fig. 24.57 we can write

$$i_t = \frac{v_t}{R_2} + \frac{v_t}{R_A + R_B} + g_{m1}R_1 g_{m2} \cdot \frac{v_t}{A_{CL}} \qquad (24.69)$$

If we assume that the current through the feedback path ($R_A + R_B$) is small, then the output resistance of the regulator is

$$R_{out,CL} \approx \frac{v_t}{i_t} = \frac{1}{\left(\frac{1}{R_2} + \frac{g_{m1}R_1 g_{m2}}{A_{CL}}\right)} \approx \frac{A_{CL}}{A_{OLDC}} \cdot R_2 \qquad (24.70)$$

Our output pole is located at

$$f_2 = \frac{1}{2\pi R_{out,CL} \cdot C_L} = \frac{A_{OLDC}}{2\pi R_2 C_L \cdot A_{CL}} \qquad (24.71)$$

For this pole to compensate the op-amp, we want to use a low value of open-loop gain, A_{OLDC}. Again, this means that we don't want a large value of resistance on node 1 (the output of the diff-amp). The problem with this is that the lower the open-loop gain, the more difficult it is to regulate the output voltage, as indicated in Eq. (24.66).

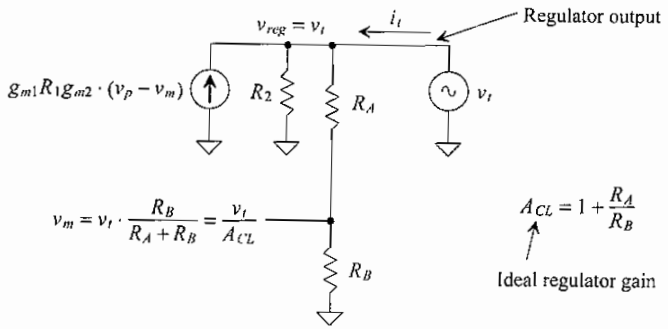


Figure 24.57 Determining the closed-loop output resistance of a two-stage op-amp.

To get an idea for the minimum size load capacitance required for a stable op-amp, let's use the parameters from Table 9.2. To begin, note that the more current the regulator supplies to a load (the more current flowing through M7), the smaller the value of R_2. This (decreasing R_2) means that the regulator becomes more likely to be unstable (f_2 increases) as it supplies more current. However, at the same time, the open-loop gain, A_{OLDC}, drops and the ability of the op-amp to regulate the output declines.

Let's assume M7's r_o and g_m are the values given in Table 9.2. Because M7 is so wide, this means (using these values for r_o and g_m) that it is conducting very little current. The value of f_1 can be estimated (assuming the $C_{gs7} >> C_c$ and the width of M7 is 10,000 or 100 times larger than the widths of the PMOS in Table 9.2) as

$$f_1 \approx \frac{1}{2\pi R_1 C_{gs7}} = \frac{1}{2\pi \cdot (167k||333k) \cdot 100 \cdot 8.34f} \approx 1.7\ MHz \qquad (24.72)$$

The DC open-loop gain of the op-amp is calculated using Eq. (24.68) as 277 (no load). If we use the op-amp in the follower configuration where R_B is infinite and R_A is zero (A_{CL} is 1), then (knowing $R_2 = 167k||333k = 111k$)

$$f_2 = \frac{A_{OLDC}}{2\pi R_2 C_L \cdot A_{CL}} = \frac{277}{2\pi(111k) \cdot C_L} \qquad (24.73)$$

Knowing we want $f_2 << f_1$, $A_{OLDC} = 277$, and $f_1 = 1.7$ MHz, we can set the unity-gain frequency to 1 MHz (making sure that it is less than f_1). If we want a –20 dB/decade roll-off above f_2 (now the low-frequency pole) then, as seen in Eq. (24.33), we can write

$$f_{un} = A_{OLDC} \cdot f_2 = \frac{A_{OLDC}^2}{2\pi R_2 C_{Lmin} \cdot A_{CL}} = 1\ MHz \qquad (24.74)$$

Solving for the minimum value of C_{Lmin} using these equations gives 110 nF. Clearly, this value is too large if we want the regulator to be completely on-chip. Rewriting Eq. (24.74)

$$C_{Lmin} = \frac{(g_{m1}R_1 g_{m2})^2 R_2}{2\pi \cdot f_{un} \cdot A_{CL}} \qquad (24.75)$$

Note that linearly increasing the current through M7 causes R_2 to decrease linearly (assuming that M7's output resistance dominates R_2) and g_{m2}^2 to increase linearly (see Eqs. (9.6) and (9.22)). The result is that C_{Lmin} doesn't change. However, notice what happens if we include R_L in the calculations. If $R_L << r_{o7}$, then we can write

$$\frac{C_{Lmin}}{R_{Lmax}} = \frac{(g_{m1}R_1 g_{m2})^2}{2\pi \cdot f_{un} \cdot A_{CL}} \qquad (24.76)$$

If we select $C_{Lmax} = 1000\ pF$, then $R_{Lmax} = 1\ k\Omega$. We use the variable R_{Lmax} to indicate that this is the maximum value of load resistance possible for a stable regulator (we want the regulator to always supply at least V_{REG}/R_{Lmax} current). If the regulator's output voltage is 500 mV, it must be supplying at least 2 mA of current. This resistor keeps the open-loop gain low when the regulator is supplying small amounts of current. We might think, after looking at Eq. (24.73), that by supplying more current (R_L or R_2 decreasing further), the location of f_2 increases and the regulator moves towards instability. However, for further decreases in the load resistance, the current supplied by M7 increases and thus so does its g_m. What happens is this: we get the canceling of effects. As the current supplied to the output increases linearly with a linear decrease in R_L, the value of g_m^2 increases linearly and the effects cancel.

A final comment: because this design is different from the previous op-amp designs, thorough characterization using simulations is needed. Steps in the load current as well as a wide range of load capacitances and resistances should be simulated to ensure that the regulator remains stable under all possible *VDDs*, temperatures, and process shifts.

Bad Output Stage Design

In this section we'll discuss some problems with biasing in Class AB output stages. To begin, consider the op-amp design seen in Fig. 24.58. This topology is a diff-amp driving an inverter. The problem with this design is that the current flowing in the push-pull output stage (an inverter) isn't set by a bias circuit. Because of this, the current may be

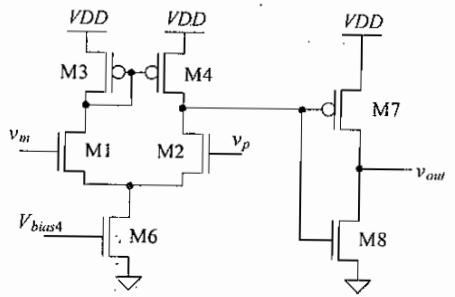


Figure 24.58 Bad op-amp design.

significant and vary greatly with both temperature and process shifts. Further, notice that when the inputs to the op-amp are equal, the drain of M4 is at the same potential as the drain of M3 (the currents in M3 and M4 are equal). This, as we've already discussed, can be used to set the bias current in M7 (we treat M7, for biasing purposes, as if its gate were tied to the gates of M3 and M4). If we call the gate potential of M7 (and thus the gate potential of M8) $VDD - V_{SG}$, then for M8 to be in saturation

$$V_{out} \geq VDD - V_{SG} - V_{THN} \qquad (24.77)$$

If VDD is 1 V, $V_{SG} = 350$ mV, and $V_{THN} = 250$ mV, then V_{out} must be greater than 400 mV for M8 to be saturated. What this means is that for wide output swings, M8 will move into triode and the push-pull output amplifier gain will drop (killing the overall gain of the op-amp). What we need to do is drop the gate potential of M8 down so that its drain can swing to a lower voltage without it trioding. Towards this goal, consider adding a source-follower level shifter to the output of the op-amp, as seen in Fig. 24.59. Now the output voltage in the amplifier can swing lower without M8 trioding. However, we still aren't controlling the current in the output stage. By not controlling this current, as we've already discussed, we produce an op-amp with poor systematic, input-referred offset voltage. If, for example, we have an input-referred offset of 10 mV and the diff-amp gain is 20, then the change in the output voltage on the gate of M7 is 200 mV. This can cause

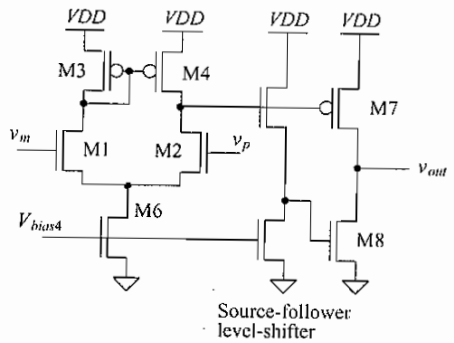


Figure 24.59 Another bad op-amp design.

the output current flowing in M7/M8 to be hundreds of μA above the current in the diff-amp.

We might try connecting the gate of M8 to the gates of M3/M4. When the noninverting input voltage, v_p, goes high, M2 turns on and causes the gate potential of M7 to drop (turning M7 further on). At the same time, the gate potential of M3 increases, causing M8 to also turn further on. The result is that M7 and M8 both turn on and fight each other for control of v_{out}. For proper operation, the gates of the output push-pull stage should move in the same direction.

To precisely control the current in the output stage, consider the topology seen in Fig. 24.60. Since, for biasing purposes, we treat the drain of M4 as if it's at the same potential as the gate of M3, the currents in M7, M9, and M10 are equal when the op-amp inputs are at the same potential. We also treat M8 as if its gate were tied to the gate of M11. The result is that the quiescent currents in M7 and M8 are equal and set by the biasing of M6. We seem to have solved the problem with biasing the class AB output stage.

However, a different problem arises. We now have three high-impedance nodes: the output of the diff-amp (a pole we've called f_1), the output of the op-amp (a pole we've called f_2), and, now, the gate of M8. If we try to make the node at the gate of M7 the dominant node, then the path that the op-amp inputs sees through M3 to M9 to the gate of M8 (the other high-impedance node) isn't affected. If we add compensation capacitors from the op-amp output to both the gates of M7 and M8, the current fed back isn't necessarily balanced. This imbalance may only be noticed when the op-amp's large signal step response is simulated. (It is generally not seen in an AC simulation.) The result is that the op-amp's settling time and stability become marginal in some situations. Having said this, the topology can still be useful in some situations (e.g., low power or low VDD).

In an attempt to improve or simplify the op-amp in Fig. 24.60, we might try to make M12 diode connected and remove M10/M11 from the circuit. With this change, however, we don't get class AB action (M9 mirrors the current in M3, which flows in the diode-connected M12 and is mirrored by M8). Finally, again, why not eliminate M9 and M12 followed by connecting the gate of M8 to the diode-connected M11? The gate potentials of M8 and M7 would then move in opposite directions, causing M7 to fight with M8 for control of the output voltage.

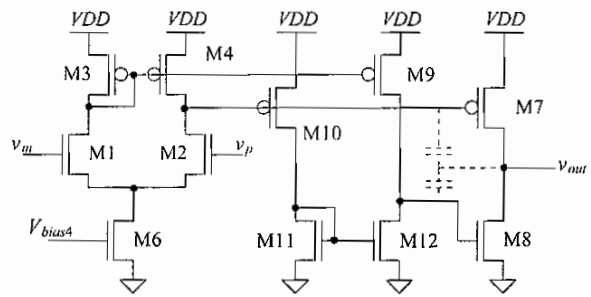


Figure 24.60 Controlling the current in a class AB output stage.

Three-Stage Op-Amp Design

In some applications we need an op-amp that can operate with a low power-supply voltage or with minimal power from *VDD*. The bias circuit we developed in Fig. 20.47 can pull more current from *VDD* than the op-amp it biases. (So, if possible, use the same bias circuit for multiple op-amps, if power and layout area are of concern.) If we look at the heart of this bias circuit (Fig. 20.47) and its operation, the Beta-multiplier in Figs. 20.22 and 20.23, we see that *VDD* can be decreased down to 500 mV with the bias currents relatively unaffected. The problem in using a *VDD* which is half of the normal *VDD* (= 1 V) is that our class AB output stage (Fig. 24.44 for example) fails to operate. For proper operation of the class AB output stage using the floating current sources (biased with V_{ncas} and V_{pcas}), we need a *VDD* > 800 mV (perhaps higher over process corners and temperature). Also, as already discussed, if we connect a resistive load to the output of the two-stage op-amp in Fig. 24.2, the gain is reduced.

Towards keeping a large gain and lowering the power supply voltage, consider the three-stage op-amp in Fig. 24.61. This design is a cascade of two diff-amps followed by a common-source amplifier. If we connect a resistive load to the output of the common- source stage, the overall op-amp gain remains relatively high due to the cascaded gain of the diff-amps. The concern with this topology (an op-amp with more than two stages), as alluded to earlier, is compensation. We still compensate the op-amp so that the pole associated with node 1 is dominant. The nesting of the compensation capacitors within the op-amp in Fig. 24.61 is sometimes called *nested Miller compensation*. We'll avoid the use of this label here since we aren't strictly using Miller compensation (Fig. 24.8 among others) but rather the indirect compensation method discussed earlier (see Fig. 24.18 and the associated discussion). Note how the current through C_{c1} feeds back to node 1 through the diode-connected load of the input diff-amp. This is necessary to ensure that the signal through C_{c1} adds with the signal fed back directly to node 1 through C_{c2}.

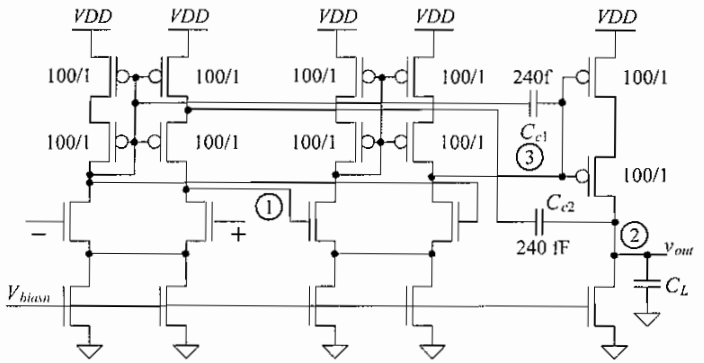


Figure 24.61 A three-stage op-amp.

Because of pole-splitting, the location of the pole at node 3 (the output of the second diff-amp) is pushed out to, see Eq. (24.24),

$$f_3 = \frac{g_{mn} \cdot C_{c1}}{2\pi \cdot C_1(C_{L2} + C_{c1})} \approx \frac{g_{mn}}{2\pi C_1} \tag{24.78}$$

a large frequency. Here we are assuming that the transconductance of the second diff-amp is g_{mn} and that C_{c1} is much larger than the input capacitance of common-source amplifier (which is the second diff-amp's C_{L2}). The location of the pole on the output of the amplifier is still given by

$$f_2 = \frac{g_{mp} \cdot C_{c2}}{2\pi \cdot C_1(C_L + C_{c2})} \tag{24.79}$$

where g_{mp} is the transconductance of the common-source amplifier. Note that we aren't discussing the LHP zeroes whose locations are specified using Eq. (24.22). We'll discuss these zeroes in a moment. We'll compensate, as usual, the op-amp so that the unity-gain frequency is less than the frequencies of the poles at f_2 and f_3.

The low-frequency, open-loop gain of the op-amp is given by

$$A_{OLDC} = \overbrace{g_{mn}(r_{on}||r_{op})}^{\text{first diff-amp's gain, } A_1} \cdot \overbrace{g_{mn}(r_{on}||r_{op})}^{\text{second diff-amp's gain, } A_2} \cdot \overbrace{g_{mp}(r_{on}||r_{op})}^{\text{common-source gain, } A_3} \tag{24.80}$$

At frequencies above the op-amp's f_{3dB} and below the gain-bandwidth product (the unity-gain frequency, f_{un}), the gain of the first-stage is rolling off. For all intents and purposes, the AC signal on the output of the first diff-amp is considerably smaller than the AC outputs of the second or third stages. In this situation the current fed back to node 1, knowing that the output voltage of the second diff-amp (at node 3) is v_{out}/A_3 , is

$$i_{Cctot} \approx \frac{v_{out}}{1/j\omega C_{c2}} + \frac{v_{out}/A_3}{1/j\omega C_{c1}} \tag{24.81}$$

The output current of the diff-amp is

$$i_{out1} = g_{mn}(v_p - v_m) \tag{24.82}$$

Because this current must equal the current fed back to node 1, we can equate Eqs. (24.81) and (24.82) to write

$$\frac{v_{out}}{v_p - v_m} = \frac{g_{mn}}{j\omega(C_{c2} + C_{c1}/A_3)} \tag{24.83}$$

The unity-gain frequency is then

$$f_{un} = \frac{g_{mn}}{2\pi(C_{c2} + C_{c1}/A_3)} \approx \frac{g_{mn}}{2\pi C_{c2}} \tag{24.84}$$

noting that the unity-gain frequency is a function of the third-stage gain. Here, we'll set C_{c2} to 240 fF to get a unity-gain frequency of 100 MHz. We still need C_{c1} to push node 3's pole to a higher frequency (pole splitting). We'll also set C_{c1} to 240 fF.

Figure 24.62 shows the frequency response of the op-amp in Fig. 24.61 without driving a load capacitance. As designed for using Eq. (24.84), the unity-gain frequency is 100 MHz. Also, as specified in Eqs. (24.22) and (24.25), we have zeroes around the unity-gain frequency. It appears from the frequency response in Fig. 24.62 that the zeroes occur above the unity-gain frequency and cause the open-loop response to flatten out.

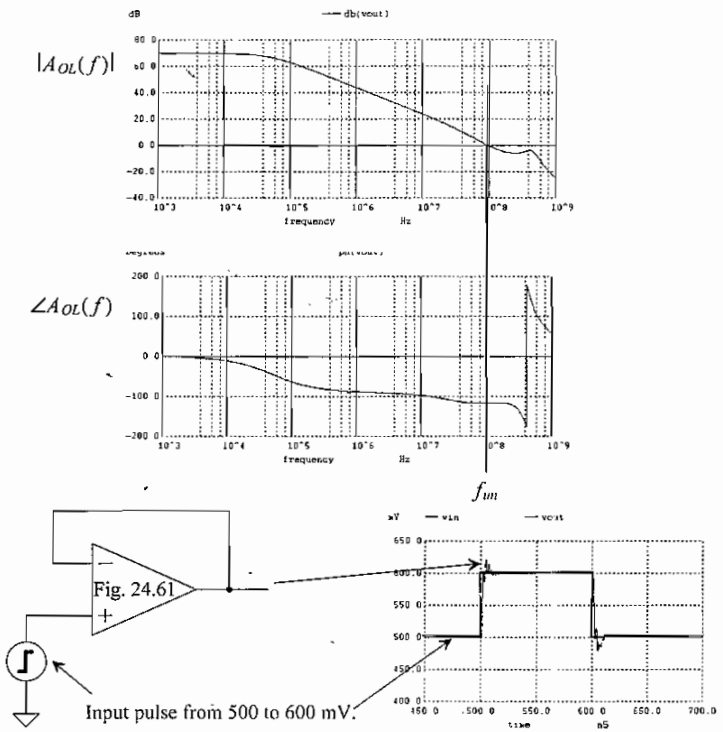


Figure 24.62 Unloaded AC and step responses for the op-amp in Fig. 24.61.

What this indicates is that one of the zeroes is canceling one of the higher-frequency poles. Note that the PM is 45°. The step response seen in Fig. 24.62 shows the characteristic ringing associated with a PM of 45°.

If we drive a load capacitance, then we expect the third-stage gain, A_3, to start to decrease at a lower frequency (it starts to decrease sooner with increasing frequency). Looking at Eq. (24.84), we then expect f_{un} to drop in value. Figure 24.63 shows the simulation where the op-amp in Fig. 24.61 is driving a 1 pF load. As expected, the addition of the 1 pF load causes f_{un} to drop in value. The PM is approximately 25°. With such a low PM, we expect the output to show significant ringing (and, as seen in Fig. 24.63, it does). To increase the PM, we can increase the value of the compensation capacitor. We can increase C_{c2} to, say, 2400 pF, for a unity gain frequency of (roughly) 10 MHz. We don't necessarily need to increase C_{c1} to decrease f_{un}, as seen in Eq. (24.84).

Finally, why did we choose the topology seen in Fig. 24.61 for our three-stage op-amp example? The main reason for using this topology comes from biasing the

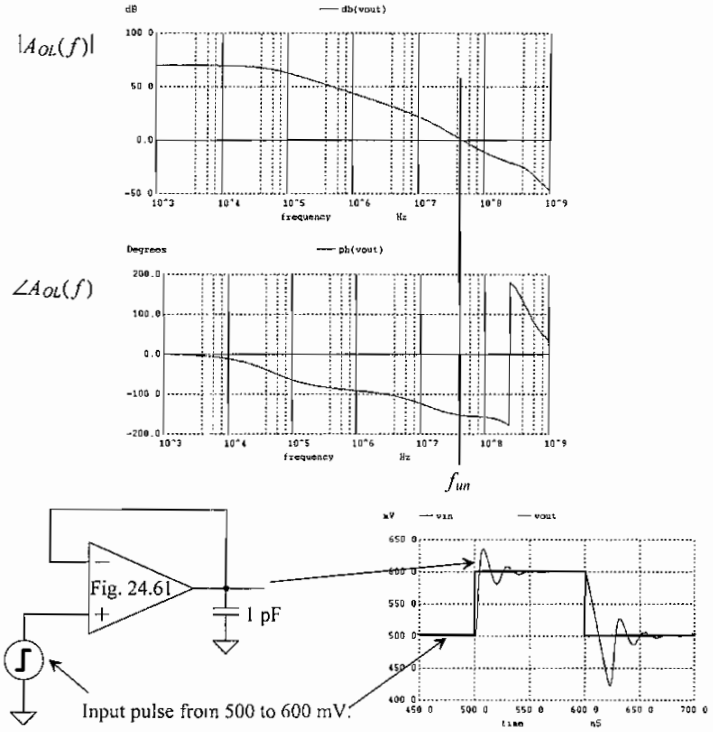


Figure 24.63 AC and step responses for the op-amp in Fig. 24.61 driving 1 pF.

common-source stage. If we were to replace the second diff-amp with a common-source stage (so our last two stages in the op-amp are common-source amplifiers), we wouldn't have a known second stage output voltage to bias the final common-source stage. The exact output voltage of a common-source amplifier, without feedback to set it, is unknown (the two drains of the NMOS and PMOS will either float up or float down, depending on which device is sourcing or sinking the most current). By using the diff-amp stages, as seen in Fig. 22.8, we ensure that each stage is biased with a known current. We don't use a diff-amp for the final stage of the op-amp because of the diff-amp's limited output swing.

Another benefit of using two diff-amps on the input of the op-amp is the increase in the *CMRR*. The overall common-mode gain is the product of each diff-amp's common-mode gain. If a single diff-amp has a *CMRR* of 80 dB, then the cascade of two diff-amps results in a *CMRR* of 160 dB.

ADDITIONAL READING

[1] R. Harjani, R. Heineke, and F. Wang, "An integrated low-voltage class AB CMOS OTA," *IEEE Journal of Solid-State Circuits,* vol. 34, pp. 134–142, February 1999.

[2] L. Moldovan and H. H. Li, "A rail-to-rail, constant gain, buffered op-amp for real time video applications," *IEEE Journal of Solid-State Circuits,* vol. 32, pp. 169–176, February 1997.

[3] F. You, S. H. K. Embabi, and E. Sánchez-Sinencio, "A multistage amplifier topology with nested Gm-C compensation for low-voltage application," *IEEE International Solid-State Circuits Conference,* vol. XL, pp. 348–349, February 1997.

[4] W. S. Wu, W. J. Helms, J. A. Kuhn, and B. E. Byrkett, "Digital-compatible high-performance operational amplifier with rail-to-rail input and output ranges," *IEEE Journal of Solid-State Circuits,* vol. 29, pp. 63–66, January 1994.

[5] R. Hogervorst, J. P. Tero, R. G. H. Eschauzier, and J. H. Huijsing, "A Compact Power-Efficient 3 V CMOS Rail-to-Rail Input/Output Operational Amplifier for VLSI Cell Libraries," *IEEE Journal of Solid State Circuits,* vol. 29, pp. 1505–1513, December 1994.

[6] J. Ramírez-Angulo and E. Sánchez-Sinencio, "Active compensation of operational transconductance amplifier filters using partial positive feedback," *IEEE Journal of Solid-State Circuits,* vol. 25, pp. 1024–1028, August 1990.

[7] K. Bult and G. J. G. M. Geelen, "A Fast-Settling CMOS Op-Amp for SC Circuits with 90-dB DC Gain," *IEEE Journal of Solid State Circuits,* vol. 25, pp. 1379–1384, December 1990.

[8] S. M. Mallya and J. H. Nevin, "Design procedures for a fully differential folded-cascode CMOS operational amplifier," *IEEE Journal of Solid-State Circuits,* vol. 24, pp. 1737–1740, December 1989.

[9] M. Banu, J. M. Khoury, and Y. Tsividis, "Fully Differential Operational Amplifiers with Accurate Output Balancing," *IEEE Journal of Solid State Circuits,* vol. 23, No. 6, pp. 1410–1414, December 1988.

[10] B. K. Ahuja, "An Improved Frequency Compensation Technique for CMOS Operational Amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 18, pp. 629–633, December 1983.

[11] P. R. Gray and R. G. Meyer, "MOS Operational Amplifier Design: A Tutorial Overview," *IEEE Journal of Solid-State Circuits,* vol. 17, pp. 969–982, December 1982.

[12] B. Y. Kamath, R. G. Meyer, and P. R. Gray, "Relationship Between Frequency Response and Settling Time of Operational Amplifiers," *IEEE Journal of Solid State Circuits,* vol. SC-9, pp. 347–352, December 1974.

PROBLEMS

24.1 Suggest, and verify with simulations, a method for reducing the minimum input common-mode voltage of the op-amp in Fig. 24.2.

24.2 Redesign the bias circuit for the op-amp in Fig. 24.2 for minimum power. Compare the power dissipation of your new op-amp design (actually the bias circuit) to the design in Fig. 24.2. Using your redesign, generate the plots seen in Fig. 24.3.

24.3 Show, using simulations, how a 1% mismatch in the widths of M1 and M2 in the op-amp of Fig. 24.2 affect the op-amp's input-referred offset voltage. Compare this offset to the offset caused by a 1% mismatch in the widths of M3 and M4. Quantitatively explain why one is worse than the other.

24.4 Simulate the use of the "zero-nulling" circuit in Fig. 24.15 in the op-amp of Fig. 24.8. Show AC, operating-point, and transient (step) operation of the resulting op-amp. Verify with the .op analysis that the gate of MP1 is at the same potential as the gate of M7 in quiescent conditions.

24.5 When we derived Eq. (24.24), we lumped C_c into C_2 and neglected the effects of MCG. A more accurate model for the indirect compensation, assuming $C_L >>$ the output capacitance of the amplifier, is seen in Fig. 24.64. Using this model, estimate the location of the output pole.

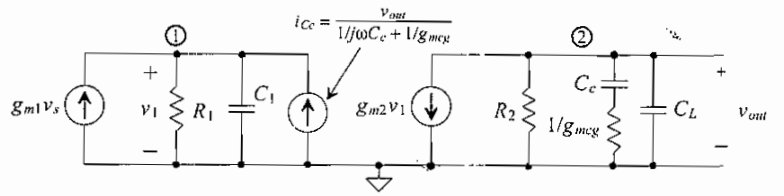


Figure 24.64 Model used to estimate bandwidth when indirect feedback current. See Problem 24.5.

24.6 Regenerate Fig. 24.19 using a 2.4 pF compensation capacitor. On the resulting simulation output, label the location f_1, f_2, f_3, and f_{un}. How do the simulated results compare to the hand calculated values?

24.7 For the op-amp in Fig. 24.21, determine the *CMRR* using hand calculations. Verify your hand calculations using simulations. How does the *CMRR* change based on the DC common-mode voltage?

24.8 Simulate the *PSRR*s for the op-amp in Fig. 24.8 (with an R_z of 6.5k and a C_c of 2.4 pF) and compare the results to the op-amp in Fig. 24.21 when C_c is set to (also) 2.4 pF (so each op-amp has the same gain-bandwidth).

24.9 Simulate the operation of the op-amp in Fig. 24.28. Show the open-loop frequency response of the op-amp. What is the op-amp's PM? Show the op-amp's step response when it is put into a follower configuration driving a 100 fF load with an input step in voltage from 100 mV to 900 mV.

24.10 The op-amp seen in Fig. 24.29 has a gain-bandwidth product (f_{un}) of about 100 MHz. Suppose that this op-amp is used in the amplifier seen in Fig. 24.65 (gain of −5). Estimate the amplifier's closed loop bandwidth (where the output of the amplifier is −3 dB down from its low-frequency value) using Eq. (24.34). Verify your results using simulations (transient analysis). What are the maximum and minimum voltages allowable for an input sinewave if the output voltage of the amplifier must lie between 100 and 900 mV?

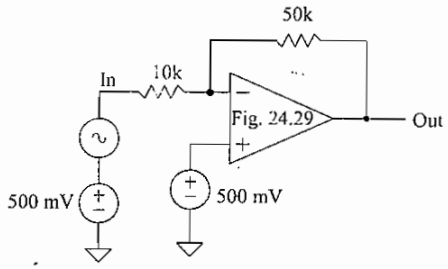


Figure 24.65 Amplifier used for Problem 24.10.

24.11 Suppose it is decided to eliminate the 500 mV common-mode voltage in the amplifier seen in Fig. 24.65 and use ground, as seen in Fig. 24.66. Knowing that the input voltage can only fall between ground and *VDD*, what is the problem one will encounter?

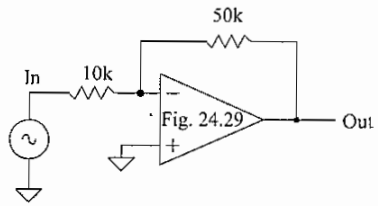


Figure 24.66 Amplifier used for Problem 24.11. What's wrong with this topology?

24.12 To limit the current flowing in MOP or MON in the op-amp of Fig. 24.29 (to protect the op-amp from destruction if its output is shorted to ground, for example), we may add 100 Ω resistors, as seen in Fig. 24.67. What is the maximum amount of current this modified op-amp can source/sink? How is the closed-loop output resistance of the op-amp affected. (Hint: see Eq. (24.70).) Simulate the operation of the op-amp using the topology seen in Fig. 24.31. Is there any noticeable difference between the simulation output in Fig. 24.31 and the output with the 100 Ω resistors present?

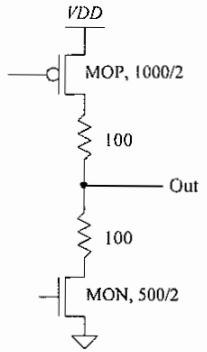


Figure 24.67 Adding resistors to the output of the op-amp in Fig. 24.29 for short circuit protection. See Problem 24.12.

24.13 Resimulate the OTA in Fig. 24.33 driving a 1 pF load (to determine f_{un}) if K = 10. How do the simulation results compare to the hand calculations using Eq. (24.41)? Estimate the parasitic poles associated with the gates of M4 and M5 (for example, the pole associated with the gate of M4 is $\approx 1/g_{m41} \cdot C_{sg4}$). Are these poles comparable to f_{un}?

24.14 Using the OTA in Fig. 24.35, design a lowpass filter with a 3 dB frequency of 1 MHz.

24.15 Suppose M8T in the op-amp of Fig. 24.37 is removed and replaced with a short from the output to the drain of M8B. How will the gain be affected? Verify your answer with SPICE.

24.16 Suppose, to simulate the open-loop gain of an OTA, the big resistor and capacitor used in Fig. 24.43 are removed and the inverting input is connected to 500 mV. Will this work? Why or why not? What happens if the OTA doesn't have an offset voltage? Will it work then?

24.17 Why is the noninverting topology (Fig. 24.49) inherently faster than the inverting topology (Fig. 24.39). What are the feedback factors, β, for each topology. Use the op-amp in Fig. 24.48 to compare the settling times for a +1 and a −1 amplifier driving 10 pF.

24.18 Suppose, to reduce power, the lengths of the current sources used in the amplifiers seen in Fig. 24.50 are increased from 2 to 10. Will the AC performance of the op-amp used to generate Fig. 24.53 change with this modification? Why or why not? Verify your answer using a SPICE simulation. Compare the currents used in the modified op-amp (with the lower power GE diff-amps) to the current seen in Fig. 24.53.

24.19 To increase the gain of the op-amp in Fig. 24.51, we may replace the GE diff-amps with folded-cascode OTAs. Will we need source-follower level-shifters

in the new design? Regenerate the simulation data seen in Fig. 24.53 using the folded-cascode OTAs.

24.20 Suppose an op-amp is to be used to amplify 250 mV to 500 mV with an error less than 1 mV. Estimate the minimum required op-amp open loop gain.

24.21 Design a voltage regulator to supply at least 50 mA of current at 500 mV with a *VDD* as low as 600 mV. Assume that a 500 mV voltage reference is available and that the load capacitance is, minimum, 1,000 pF. How does the design respond to a load current pulse from 0 to 50 mA? Use SPICE to verify your design.

24.22 Using the nominal sizes from Table 9.2 and the bias circuit in Fig. 20.47, simulate, using an .op analysis, the operation of the op-amp in Fig. 24.58 in the configuration seen in Fig. 24.9. What is the current flowing in M7 and M8 when *VDD* is 1 V? is 1.2 V?

24.23 Repeat Problem 24.22 for the op-amp in Fig. 24.59.

24.24 Using the information from Table 9.2 and the bias circuit in Fig. 20.47, demonstrate the operation (AC and transient) of the op-amp in Fig. 24.60 driving a 100 fF load. Show that the bias current pulled from *VDD* is relatively constant for a *VDD* of 1 or 1.2 V (put the op-amp in the configuration seen in Fig. 24.9).

24.25 Replace the common-source output stage in the op-amp of Fig. 24.61 with a class AB output stage like the one seen in Fig. 24.60. Simulate the operation of the amplifier (AC and transient).

Chapter

25

Dynamic Analog Circuits

In Chapter 14 we discussed dynamic logic gates. Dynamic logic is useful for reducing power dissipation and the number of MOSFETs used to perform a given circuit operation (layout area). Dynamic analog circuits exploit the fact that information can be stored on a capacitor or gate capacitance of a MOSFET for a period of time. In this chapter, we discuss analog circuits such as sample and holds, current mirrors, amplifiers, and filters using dynamic techniques.

25.1 The MOSFET Switch

A fundamental component of any dynamic circuit (analog or digital) is the switch (Fig. 25.1). An important attribute of the switch, in CMOS, is that under DC conditions the gate of the MOSFET does not draw a current. Therefore, neglecting capacitances from the gate to the drain/source, we find that the gate control signal does not interfere with information being passed through the switch. Figure 25.2 shows the small-signal resistance of the switches of Fig. 25.1 plotted against input voltage. The benefits of using the CMOS transmission gate are seen from this figure, namely, lower overall resistance. Another benefit of using the CMOS TG is that it can pass a logic high or a logic low without a threshold voltage drop. The largest voltage that an NMOS switch can pass is $VDD - V_{THN}$, while the lowest voltage a PMOS switch can pass is V_{THP}.

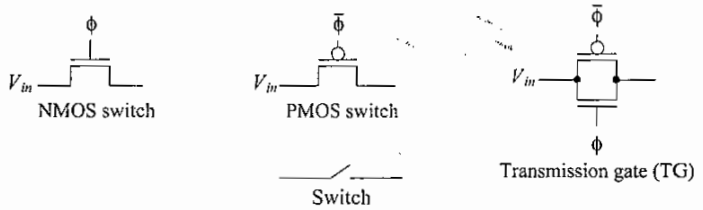


Figure 25.1 MOSFETs used as switches.

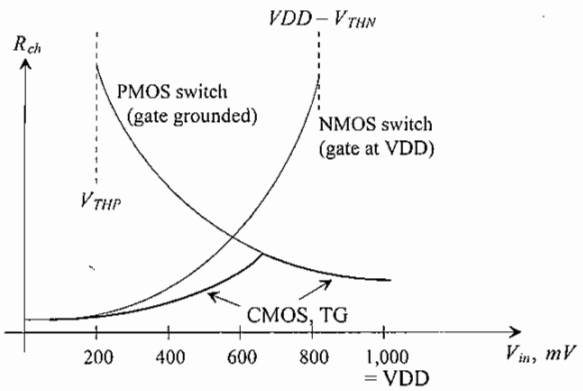


Figure 25.2 Small-signal on-resistance of MOSFET switches .

While MOS switches may offer substantial benefits, they are not without some detraction. Two nonideal effects typically associated with these switches may ultimately limit the use of MOS switches in some applications (particularly sampled-data circuits such as data converters). These two effects are known as *charge injection* and *clock feedthrough*.

Charge Injection

Charge injection can be understood with the help of Fig. 25.3. When the MOSFET switch is on and V_{DS} is small, the charge under the gate oxide resulting from the inverted channel is (from Ch. 6) Q'_{ch}. When the MOSFET turns off, this charge is injected onto the capacitor and into V_{in}. Because V_{in} is assumed to be a low-impedance, source-driven node, the injected charge has no effect on this node. However, the charge injected onto C_{load} results in a change in voltage across it. Note that we also have charge injection (in the opposite direction) when we turn the switch on. However, the fact that the input voltage is connected to C_{load} through the channel resistance makes this error unimportant (the voltage across C_{load} charges to V_{in} through the MOSFET's channel resistance).

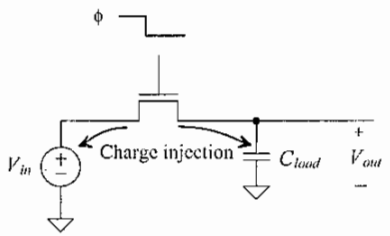


Figure 25.3 Simple configuration using an NMOS switch to show charge injection.

Although the charge injection mechanism is itself a complex one, many studies have sought to characterize and minimize its effects. It has been shown that if the clock signal turns off fast, the channel charge distributes fairly equally between the adjacent nodes. Thus, half of the channel charge is distributed onto C_{load} . From Ch. 6, the charge/unit area of an inverted channel can be approximated as

$$Q'_I(y) = C'_{ox} \cdot (V_{GS} - V_{THN}) \tag{25.1}$$

The total charge in the channel must then be multiplied by the area of the channel resulting in

$$Q_I(y) = C'_{ox} \cdot W \cdot L \cdot (V_{GS} - V_{THN}) \tag{25.2}$$

Therefore, the change in voltage across C_{load} (if an NMOS switch is used) is

$$\Delta V_{load} = -\frac{C'_{ox} \cdot W \cdot L \cdot (V_{GS} - V_{THN})}{2C_{load}} \tag{25.3}$$

which can be written as

$$\Delta V_{load} = -\frac{C'_{ox} \cdot W \cdot L \cdot (VDD - V_{in} - V_{THN})}{2C_{load}} \tag{25.4}$$

if it is assumed that the clock swings between *VDD* and ground. The threshold voltage, Eq. (6.19), can also be substituted into Eq. (25.4) to form

$$\Delta V_{load} = -\frac{C'_{ox} \cdot W \cdot L \cdot (VDD - V_{in} - [V_{THN0} + \gamma(\sqrt{|2V_{fp}| + V_{in}} - \sqrt{|2V_{fp}|})])}{2C_{load}} \tag{25.5}$$

Note that Eq. (25.5) illustrates the problem associated with charge injection. The change in voltage across C_{load} is nonlinear with respect to V_{in} due to the threshold voltage. Thus, it can be said that if the charge injection is signal-dependent, harmonic distortion results. In sampled-data systems, charge injection results in nonlinearity errors. In the case where the charge injection is signal-independent, a simple offset occurs, which is much easier to manage than harmonic distortion. These will be discussed in more detail in Chs. 28 and 29, but it should be obvious here that charge injection effects should be minimized as much as possible.

Capacitive Feedthrough

Consider the schematic of the NMOS switch shown in Fig. 25.4. Here the capacitances between the gate/drain and gate/source of the MOSFET are modeled with the assumption that the MOSFET is operating in the triode region. When the gate clock signal, ϕ, goes high, the clock signal feeds through the gate/drain and gate/source capacitances. However, as the switch turns on, the input signal, V_{in}, is connected to the load capacitor through the NMOS switch. The result is that C_{load} is charged to V_{in} and the capacitive feedthrough has no effect on the final value of V_{out} . However, now consider what happens when the clock signal makes the transition low, that is, the n-channel MOSFET turns off. A capacitive voltage divider exists between the gate-drain (source) capacitance and the load capacitance. As a result, a portion of the clock signal, ϕ, appears across C_{load} as

$$\Delta V_{load} = \frac{C_{overlap} \cdot VDD}{C_{overlap} + C_{load}} \tag{25.6}$$

where $C_{overlap}$ is the overlap capacitance value,

$$C_{overlap} = C'_{ox} \cdot W \cdot LD \tag{25.7}$$

and LD is the length of the gate that overlaps the drain/source.

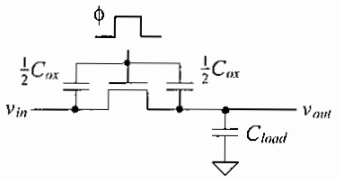


Figure 25.4 Illustration of capacitive feedthrough.

Reduction of Charge Injection and Clock Feedthrough

Many methods have been reported that reduce the effects of charge injection and capacitive feedthrough. One of the most widely used is the dummy switch, as seen in Fig. 25.5. Here, a switch, M2, with its drain and source shorted is placed in series with the desired switch M1. Notice that the clock signal controlling the dummy switch is the complement of the signal controlling M1, and in addition, should also be slightly delayed.

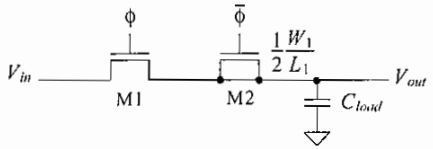


Figure 25.5 Dummy switch circuit used to minimize charge injection.

When M1 turns off, half of the channel charge is injected toward the dummy switch, thus explaining why the size of M2 is one-half that of M1. Although M2 is effectively shorted, a channel can still be induced by applying a voltage on the gate. Therefore, the charge injected by M1 is essentially matched by the charge induced by M2, and the overall charge injection is canceled. Note what happens when M2 turns off. It will inject half of its charge in both directions. However, because the drain and source are shorted and M1 is on, all of the charge from M2 will be injected into the low-impedance, voltage-driven source, which is also charging C_{load}. Therefore, M2's charge injection will not affect the value of voltage on C_{load}.

Another method for counteracting charge injection and clock feedthrough is to replace the switch with a CMOS transmission gate (TG). This results in lower changes in V_{out} because the complementary signals that are used will act to cancel each other.

However, this approach requires precise control on the complementary clocks (the clocks must be switched at exactly the same time) and assumes that the input signal, V_{in}, is small, since the symmetry of the turn-on and turn-off waveforms depend on the input signal.

Fully-differential circuit topologies are used to cancel these effects to a first order, as seen in Fig. 25.6. Since the nonideal charge injection and clock feedthrough effects appear as a common-mode signal to the amplifier, they will be reduced by the CMRR of the amplifier. However, the second-order effects resulting from the input signal amplitude dependence will ultimately limit the dynamic range of operation, neglecting coupled and inherent noise in the dynamic circuits. This subject will be discussed in more detail in the following sections and in the next chapter.

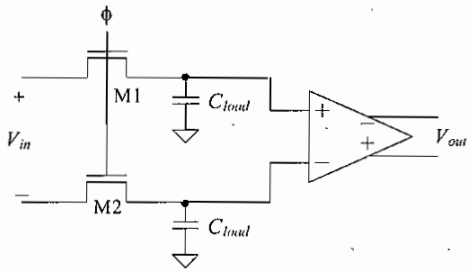


Figure 25.6 Using a fully-differential circuit to minimize charge injection and clock feedthrough.

kT/C Noise

In Ch. 8 we saw that the maximum RMS output noise generated from a simple *RC* circuit was $\sqrt{kT/C}$ (see Table 8.1). If we think of the MOSFET in Fig. 25.7 as a resistor (when the MOSFET is on), then we can add this RMS noise source in series with the output of the capacitor. The noise can be regarded as a sampled (random) voltage onto the capacitor each time the switch is turned on. The RMS noise generated, at room temperature, when using a 1 pF capacitor is 64 μV, while a 100 fF capacitor results in a noise voltage of 200 μV. In other words, the larger the capacitor, the smaller the noise voltage sampled on to the storage capacitor. For high-speed systems, it is desirable to use small capacitors since they take less time to charge. When designing a high-speed and low-noise circuit, trade-offs must be made when selecting the capacitor size.

Note, as mentioned above, the MOSFET is thought of as a resistor (and so *kT/C* noise in the circuit in Fig. 25.7 is from MOSFET thermal noise). If we review the noise mechanisms found in a MOSFET in Ch. 9, we also see that Flicker noise may be present. However, for a MOSFET's drain current to contain Flicker noise, the MOSFET must be conducting a DC current. Because the MOSFET in Fig. 25.7 doesn't conduct any DC current after C_H is charged, Flicker noise doesn't affect the sampled voltage.

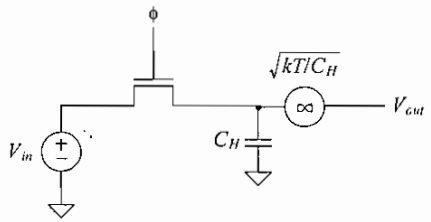


Figure 25.7 How kT/C noise adds to a sampled signal.

25.1.1 Sample-and-Hold Circuits

An important application of the switch is in the sample-and-hold circuit. The sample-and-hold circuit finds extensive use in data converter applications as a sampling gate. A variety of topologies exist, each with their own benefits. The simplest is shown in Fig. 25.8. A narrow pulse is applied to the gate of the MOSFET, enabling v_{in} to charge the hold capacitor, C_H. The width of the strobing gate pulse should allow the capacitor to fully charge before being removed. The op-amp simply acts as a unity gain buffer, isolating the hold capacitor from any external load. This circuit suffers from the clock feedthrough and charge injection problems mentioned in the previous discussion.

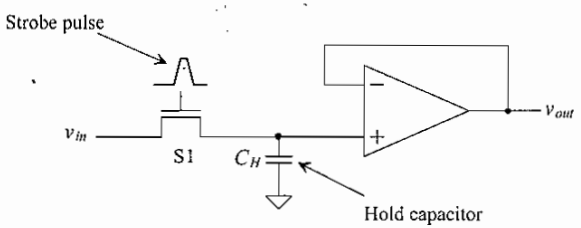


Figure 25.8 A basic sample-and-hold circuit (more correctly called a track-and-hold, see Fig. 28.5b).

A possible improvement in the basic S/H circuit is seen in Fig. 25.9. Here, two amplifiers buffer the input and the output. Notice that switch S_2 ensures that amplifier A1 is stable while in hold mode. If the switch were not present, A1 would be open loop during hold mode and would swing to one of the rails. During the next sample mode, it would then be slew-limited while going from the supply to the value of v_{in}. However, with the addition of S_2, the output of A1 tracks v_{in} even while in hold mode. The switch S_3 also disconnects A1 from the output during hold mode. This S/H has its disadvantages, however. The capacitor is still subjected to charge injection and clock-feedthrough problems. In addition, during sample mode, the circuit may become unstable since there are now two amplifiers in the single-loop feedback structure. Although compensation

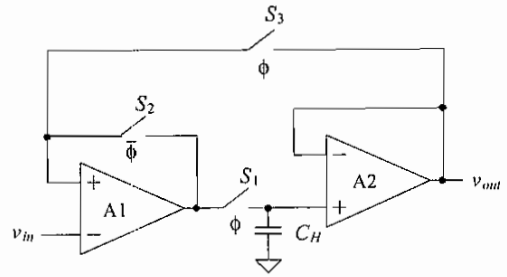


Figure 25.9 A closed-loop S/H circuit.

capacitors can be added to stabilize its performance, the size and placement of the capacitors depend solely on the type and characteristics of the op-amps.

Another S/H circuit is seen in Fig. 25.10. Here, a transconductance amplifier is used to charge the hold capacitor. A control signal turns the amplifier, A1, on or off digitally, thus eliminating the need for the switches S_2 and S_3 (from Fig. 25.9). Since CMOS op-amps are well suited for high-output impedance applications, this configuration would seem to be a popular one. However, the speed of this topology is dictated by the maximum current output of the transconductance amplifier and the size of the hold capacitor.

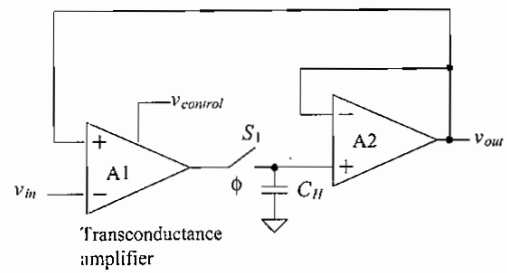


Figure 25.10 A closed-loop S/H circuit using a transconductance amplifier.

A third S/H circuit can be seen in Fig. 25.11. The advantage of this circuit may not be completely obvious and warrants further explanation. First, notice that the hold capacitor is actually in the feedback path of the amplifier, A2, with one side connected to the output of the amplifier and the other connected to a virtual ground. When switch S_1 turns off, any charge injected onto the hold capacitor results in a slight change in the output voltage. However, now that one side of the switch is at virtual ground, the change in voltage is no longer dependent on the threshold voltage of the switch itself. Therefore,

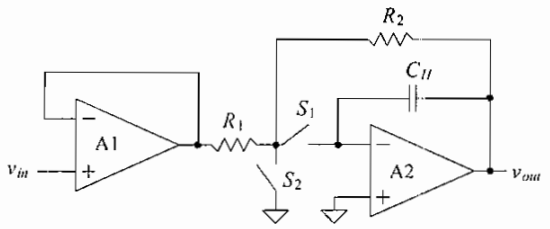


Figure 25.11 A closed-loop S/H circuit using a transconductance amplifier.

the charge injection will be independent of the input signal and will result as a simple offset at the output. An offset error is much easier to tolerate than a nonlinearity error, as will be seen in Chs. 28 and 29.

When sampling, S_1 is closed and S_2 is open, and the equivalent circuit is simply a low-pass filter with a buffered input. The overall transfer function becomes

$$\frac{v_{out}}{v_{in}} = -\frac{R_2}{R_1} \cdot \frac{1}{(sR_2C_H + 1)} \tag{25.8}$$

Therefore, this circuit performs a low-pass filter function while sampling. The buffer A1 can be eliminated when we desire a low-input impedance. Once hold mode commences, the output will stay constant at a value equal to v_{in}, while the switch S_2 isolates the input from the hold capacitor. One important issue to note here is that A2 will need to be a buffered CMOS amplifier because of the resistive load attached at v_{out} during hold mode. Notice also that during both sample mode and hold mode, there is only one op-amp in each feedback loop, so this S/H topology is much more stable than the closed-loop structure introduced in Fig. 25.10.

25.2 Fully-Differential Circuits

As we saw in Fig. 25.6 and the associated discussion, using a fully-differential topology (an op-amp with both differential inputs and outputs) can reduce the effects from imperfect switches. However, using a fully-differential topology requires the use of a *common-mode-feedback* (CMFB) circuit. In the following we present an overview of fully-differential op-amps and their application in a sample-and-hold circuit.

Gain

The differential output op-amp symbol is shown in Fig. 25.12. The open-loop gain of the op-amp is related to its inputs and outputs by

$$v_{out} = v_{op} - v_{om} = A_{OL} \cdot (v_p - v_m) \tag{25.9}$$

This should be compared to the single-ended output op-amp (the op-amp we have been discussing up to this point), which has a gain given by

$$v_{out} = v_{op} = A_{OL} \cdot (v_p - v_m) \tag{25.10}$$

If we ignore v_{om}, then the differential output op-amp behaves just like a single-ended op-amp. For linear applications, the op-amp is used with a feedback network, the inputs are related (assuming the open-loop gain, A_{OL} is large) by

$$v_p \approx v_m \tag{25.11}$$

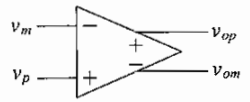


Figure 25.12 Differential output op-amp.

Common-Mode Feedback (CMFB)

Normally, the average of the op-amp outputs is called the common-mode output voltage. The two op-amp outputs swing around the common-mode voltage. If the largest op-amp output voltage is *VDD* and the minimum output voltage is ground, then the common-mode voltage is

$$V_{CM} = \frac{VDD}{2} \tag{25.12}$$

Figure 25.13a shows a simple differential output op-amp gain configuration (inverting or noninverting, depending on which output is used as the positive output). Because the input voltages to the circuit are equal, the output voltages of the op-amp should remain at V_{CM}. Due to the op-amp's high gain and the negative feedback, we always have $v_p \approx v_m$ (Eq. [25.11]). However, if $v_{op} = v_{om} = VDD$ or $v_{op} = v_{om}$ = ground or $v_{op} = v_{om}$ = anything, then $v_p = v_m$ and Eq. (25.11) is satisfied. *This is a problem* and the reason why we need a CMFB circuit. In Fig. 25.13b, we add a CMFB circuit to monitor the outputs of the op-amp and make adjustments in the op-amp (how these adjustments are made is discussed in the next chapter) to keep the two outputs balanced around V_{CM} ($VDD/2$).

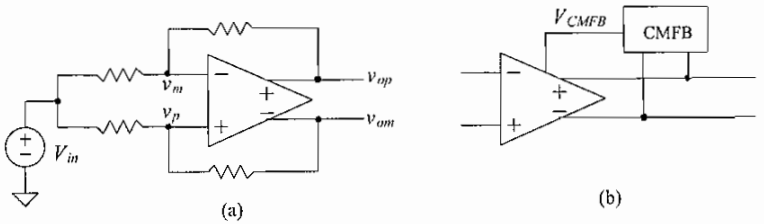


Figure 25.13 (a) Simple gain configuration using differential output op-amp and (b) the use of a common-mode feedback circuit to adjust the common-mode output voltage.

Coupled Noise Rejection

Differential topologies are required to minimize the effects of charge injection and capacitive feedthrough from switches. They also are used to help reduce the effects of coupled noise. Consider the cascade of differential output op-amps, with no feedback network shown, seen in Fig. 25.14. The stray capacitance between the interconnecting metal lines (the metal lines that carry the signals between op-amps) and the substrate or any other noise source are shown. If the metal lines are run close to one another, then the noise voltage will couple even amounts (ideally) of noise into each signal wire. Since the diff-amp, on the input of the op-amp, rejects common signals (signals that are present on both inputs), the coupled noise is not passed to the next op-amp in the string. Variations on the power supply rails are rejected as well. If the differential op-amp is symmetric, then changes on the power supply couple evenly into both outputs, having little effect on the difference, or desired signal, coming out of the op-amp. For these reasons, that is, good coupled noise rejection and PSRR, the differential op-amp is a necessity in any dynamic analog integrated circuit.

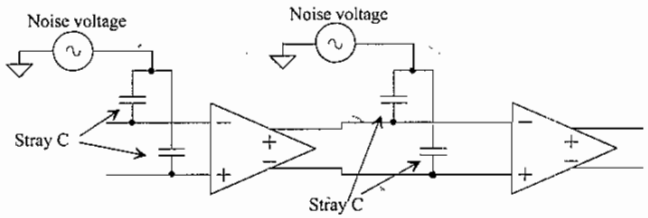


Figure 25.14 Differential output op-amps showing parasitic capacitance and noise.

Other Benefits of Fully-Differential Op-Amps

Another benefit of fully-differential circuits is a doubling in output voltage swing. For example, if *VDD* is 1 V, then the output of a single-ended op-amp can swing from 1 V to ground. In a fully-differential topology, both v_{op} and v_{om} can swing from 0 to 1. Using Eq. (25.9), we can write

$$v_{out,\max} = v_{op} - v_{om} = 1 - 0 = 1\ V \text{ and } v_{out,\min} = v_{op} - v_{om} = 0 - 1 = -1\ V \quad (25.13)$$

The output swing is 2*VDD* when using fully-differential topologies.

A further benefit of fully-differential topologies is the fact that the input common-mode voltage of the op-amp remains at V_{CM} (and so the op-amp's first stage diff-amp's common-mode voltage range requirements are easy to meet).

25.2.1 A Fully-Differential Sample-and-Hold

Figure 25.15 shows a fully-differential sample-and-hold circuit and the associated clock waveforms that eliminate clock feedthrough and charge injection to a first order. The switches in this figure are closed when their controlling clock signals are high. The basic operation can be understood by considering the state of the circuit at t_0. At this time, the

input signals charge the sampling capacitors. The bottom plates of the capacitors (poly1) are tied directly to the input signals, for reasons that will be explained below. The op-amp is operating in a unity-follower configuration in which both inputs of the op-amp are held at V_{CM}. At this particular instance in time, prior to t_1, the amplifier is said to be operating in the sample mode of operation.

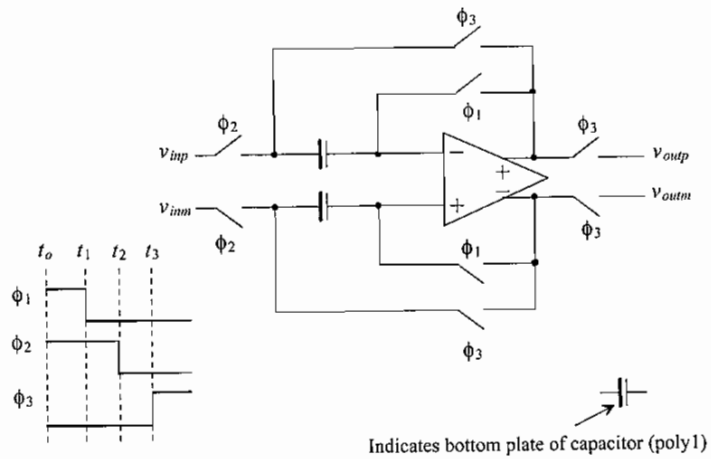


Figure 25.15 Sample-and-hold using differential topology.

At t_1, the ϕ_1 switches turn off. The resulting charge injection and clock feedthrough appear as a common-mode signal on the inputs of the op-amp and are ideally rejected. Since the top plates of the hold capacitors (the inputs to the op-amp) are always at V_{CM}, at this point in time the charge injection and clock feedthrough are independent of the input signals. This produces an increase in the dynamic range of the sample-and-hold (the minimum measurable input signal decreases). The voltage on the inputs of the op-amp (the top plate of the capacitor) between t_1 and t_2 is $V_{OFF1} + V_{CM}$, a constant voltage. Note that the op-amp is operating open loop at this time so the time between t_1 and t_3 should be short.

At t_2 the ϕ_2 switches turn off. At this point in time, the voltages on the bottom plates of the sampling (or hold) capacitors (poly1) are v_{inp} and v_{inm}. The voltages on the top plates of the capacitors (connected to the op-amp) are $V_{OFF1} + V_{OFF2} + V_{CM}$ (assuming that the storage capacitors are much larger than the input capacitance of the op-amp). The term V_{OFF2} is ideally a constant that results from the charge injection and capacitive feedthrough from the ϕ_2 switches turning off. The time between t_1 and t_2 should be short compared to variations in the input signals.

At time t_3 the ϕ_3 switches turn on and the op-amp behaves like a voltage follower; the circuit is said to be in the hold mode of operation. The charge injection and clock feedthrough resulting from the ϕ_3 switches turning on causes the top plate of the capacitor to become $V_{OFF1} + V_{OFF2} + V_{OFF3} + V_{CM}$, again assuming that the storage capacitors are much larger than the input capacitance of the op-amp. The outputs of the sample-and-hold are v_{inp} and v_{inm}, assuming infinite op-amp gain since these offsets appear as a common-mode voltage on the input of the op-amp. Note that the terms V_{OFF2} and V_{OFF3} are dependent on the input signals.

Connecting the Inputs to the Bottom (Poly1) Plate

The reason for connecting the input signals to the bottom plate of the capacitor can be explained with the help of Fig. 25.16. This figure is a simplified, single-ended version of Fig. 25.15 where the capacitance, C_p, is the parasitic capacitance from the bottom plate to the substrate. With regard to Fig. 25.16a, coupled noise from the substrate sees either the input voltage from the op-amp driving the sample-and-hold (in the sample mode) or the output voltage of the op-amp used in the sample-and-hold itself (in the hold mode). Since the op-amps in either mode directly set this voltage, the substrate noise has, ideally, little effect on the circuit's operation.

In Fig. 25.16b, coupled substrate noise feeds directly into the input of the op-amp and can thus drastically affect the output of the sample-and-hold. Another more subtle problem occurs in the circuit of Fig. 25.16b. When the circuit makes the transition to the hold mode at t_3, the output of the op-amp should quickly change to the voltage sampled on the input capacitors. The time it takes the output of the op-amp to change and settle to this final voltage is called the *settling time*. The parasitic capacitance on the input of the op-amp in (b) reduces the feedback factor from unity to $C_H/(C_p + C_H)$. This slows the settling time and causes a gain error in the circuit's transfer function. For these reasons, the parasitic capacitance on the top plate of the capacitor should be small. Nothing should be laid out over or in near proximity of poly2.

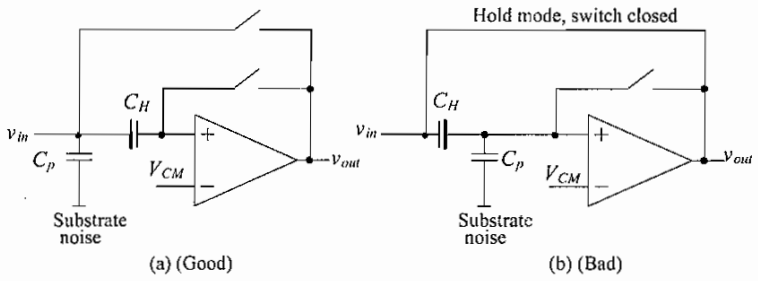


Figure 25.16 Explanation for connecting the bottom plate of the capacitor to the input.

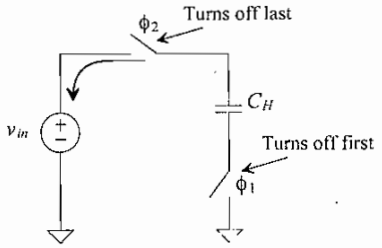


Figure 25.17 Bottom plate sampling.

Bottom Plate Sampling

Turning the ϕ_1 controlled switches off in Fig. 25.15 slightly before the ϕ_2 controlled switches is sometimes called *bottom plate sampling*. Figure 25.17 illustrates why. When the ϕ_1 switches turn off, the charge injected into C_H is a constant independent of the input signal amplitude. The resulting offset voltage across C_H is then constant. When the ϕ_2 switches turn off, the resulting charge injection then takes the path of least resistance, that is, into the input source v_{in}. The voltage across C_H is then, ideally, independent of the input signal voltage.

SPICE Simulation

Let's simulate the operation of the sample-and-hold in Fig. 25.15. To begin, let's use voltage-controlled voltage sources and a DC voltage to model the op-amp (again, we'll discuss differential output op-amps in the next chapter). The ideal fully-differential output op-amp SPICE model is seen in Fig. 25.18. The SPICE netlist statements used to model this circuit (we'll assume an open-loop gain of 10^6 and *VDD* of 1 *V*) are

```
E1    vop   vcm   vp    vm    1e6
E2    vcm   vom   vp    vm    1e6
Vcm   vcm   0     DC    500m
```

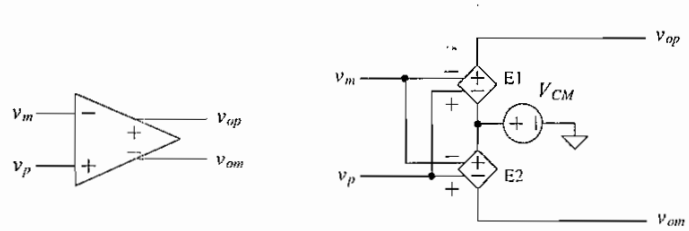


Figure 25.18 SPICE modeling a differential input/output op-amp with common-mode voltage.

For the switches we'll use NMOS devices as seen in the simulation schematic of Fig. 25.19. For the input signals, we've used a 5 MHz sinewave signal with a peak amplitude of 200 mV. Note that each of the inputs is referenced to the common-mode voltage, V_{CM} (which is 500 mV here).

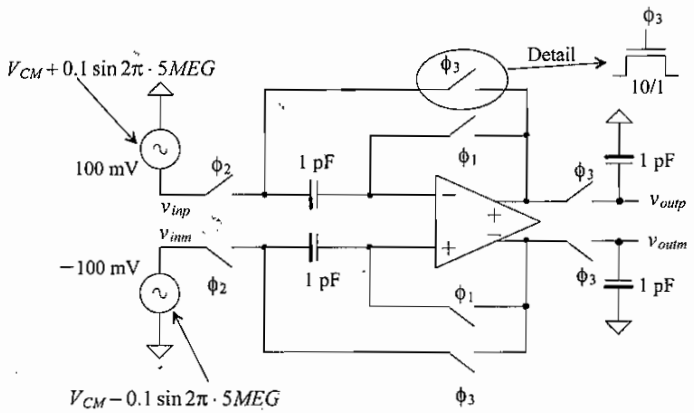


Figure 25.19 Simulating the operation of the fully differential sample-and-hold.

The simulation results are seen in Fig. 25.20. Let's first focus on the clock signals. Notice how ϕ_3 isn't high at the same time as ϕ_1 or ϕ_2. If all switches were "on" at the same time, the charge stored (or held) on the 1 pF capacitors at the output of the circuit would change (remember when the ϕ_1 switches close the op-amp is in the follower configuration with the inputs and outputs pulled to V_{CM}). We make sure to disconnect these output capacitors before putting the circuit into the sample mode. The output of the sample-and-hold seen in Fig. 25.20 doesn't exactly match the input signal. There is a one-clock cycle delay and we can see the effects of the finite clock frequency (finite sample points). Note, these topics are discussed in detail in Volume II of this book entitled *CMOS Mixed-Signal Circuit Design*.

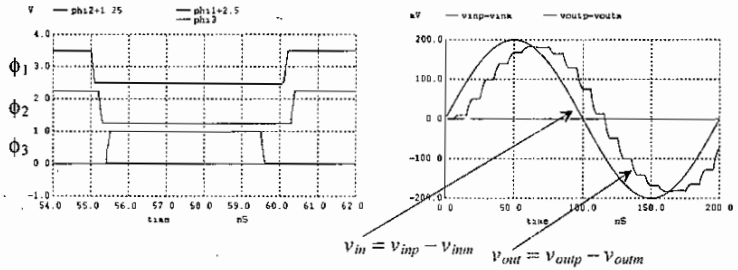


Figure 25.20 Simulating the operation of the sample-and-hold in Fig. 25.19.

25.3 Switched-Capacitor Circuits

Consider the circuit shown in Fig. 25.21a. This dynamic circuit, named a *switched-capacitor resistor*, is useful in simulating a large value resistor, generally >1 MΩ. The clock signals ϕ_1 and ϕ_2 form two phases of a non-overlapping clock signal with frequency f_{clk} and period T as seen in the figure. Let's begin by considering the case when S1 is closed. When ϕ_1 is high, the capacitor C is charged to v_1. The charge, q_1, stored on the capacitor during this interval, Fig. 25.21c, is

$$q_1 = Cv_1 \tag{25.14}$$

while if S2 is closed, the charge stored on the capacitor is

$$q_2 = Cv_2 \tag{25.15}$$

If v_1 and v_2 are not equal, keeping in mind that S1 and S2 cannot be closed at the same time due to the nonoverlapping clock signals, then a charge equal to the difference between q_1 and q_2 is transferred between v_1 and v_2 during each interval T. The difference in the charge is given by

$$q_1 - q_2 = C(v_1 - v_2) \tag{25.16}$$

If v_1 and v_2 vary slowly compared to f_{clk}, then the average current transferred in an interval T is given by

$$I_{avg} = \frac{C(v_1 - v_2)}{T} = \frac{v_1 - v_2}{R_{sc}} \tag{25.17}$$

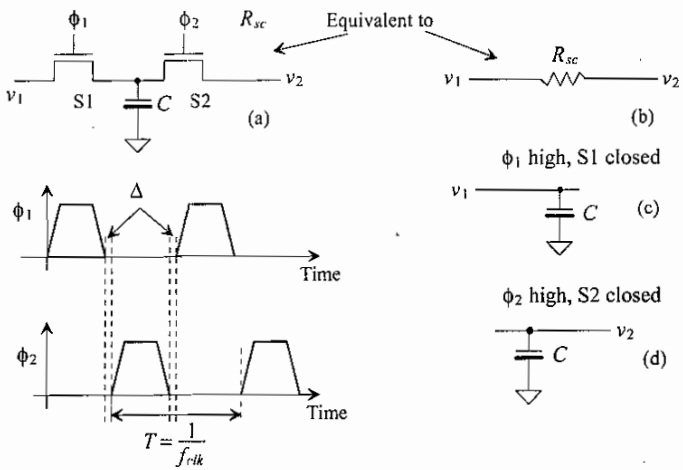


Figure 25.21 Switched-capacitor resistor (a) and associated waveforms and (b, c, d) the equivalent circuits.

The resistance of the switched-capacitor circuit is given by

$$R_{sc} = \frac{T}{C} = \frac{1}{C \cdot f_{clk}} \tag{25.18}$$

In general, the signals v_1 and v_2 should be bandlimited to a frequency at least ten times less than f_{clk} (more on this later). Note that we derived a similar result back in Ch. 17, see Eq. (17.29).

Example 25.1

Using switched-capacitor techniques, implement the circuit shown in Fig. 25.22a so that the product of RC is 1 ms, that is, the 3-dB frequency of $|v_{out}/v_{in}|$ is 159 Hz.

The switched-capacitor implementation of this circuit is shown in Fig. 25.22b. The product of RC may now be written in terms of Eq. (25.18) as

$$RC_2 = \frac{C_2}{C_1} \cdot \frac{1}{f_{clk}} \tag{25.19}$$

This result is important! The product RC_2 is determined by f_{clk}, which may be an accurate frequency derived from a crystal oscillator and the ratio of C_2 to C_1, which will be within 1% on a chip. This means that even if the values of the capacitors change by 20% from wafer to wafer, the ratio of the capacitors relative to one another, on the same wafer, will remain constant within 1%.

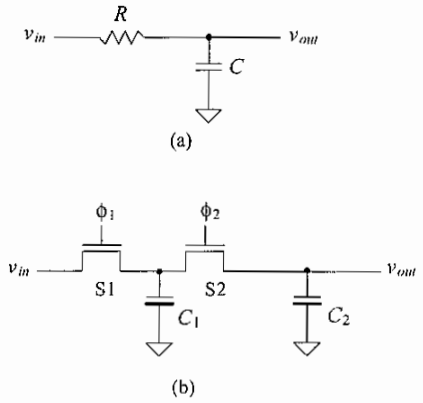


Figure 25.22 (a) Circuit used in Ex. 27.1 and (b) its implementation using a SC resistor.

It is also desirable to keep C_1 larger than the associated parasitics present in the circuit (e.g., depletion capacitances of the source/drain implants and the stray capacitances to substrate). For the present example, we will set C_1 to 1 pF. The selection of f_{clk} is usually determined by what is available. For the present design, a value of 100 kHz will be used. This selection assumes that the energy present in v_{in} at frequencies above 10 kHz is negligible. The value of C_2 is determined solving Eq. (25.19) and is 100 pF. Note that the value of the switched-capacitor

resistor is 10 MΩ. Implementing this resistor in a CMOS process using n-well with a sheet resistance of 1,000 ohms/square would require 10,000 squares! The resulting delay through the n-well resistor, because of the capacitance to substrate, may cause a significant phase error in the transfer function. ■

25.3.1 Switched-Capacitor Integrator

Because the switched-capacitor resistor of Fig. 25.21a is sensitive to parasitic capacitances, it finds little use, by itself, in analog switched-capacitor circuits. Consider the circuit of Fig. 25.23a. This circuit, a switched-capacitor integrator, is the heart of the circuits we will be discussing in the remainder of the section. The portion of the circuit consisting of switches S1 through S4 and C_f forms a switched-capacitor resistor with a value given by

$$R_{sc} = \frac{1}{C_f f_{clk}} \tag{25.20}$$

The equivalent continuous time circuit for the switched-capacitor integrator is shown in Fig. 25.23b. Notice that v_{in} is now negative. It may be helpful in the following discussion to remember that the combination of switches and C_f of the switched-capacitor integrator can be thought of as a simple resistor. The transfer function of the switched-capacitor integrator is given by

$$\frac{v_{out}}{v_{in}} = \frac{1/j\omega C_F}{R_{sc}} = \frac{1}{j\omega\left(\frac{C_F}{C_f} \cdot \frac{1}{f_{clk}}\right)} \tag{25.21}$$

Again, the ratio of capacitors is present, allowing the designer to precisely set the gain of the amplifier and the integration time constant.

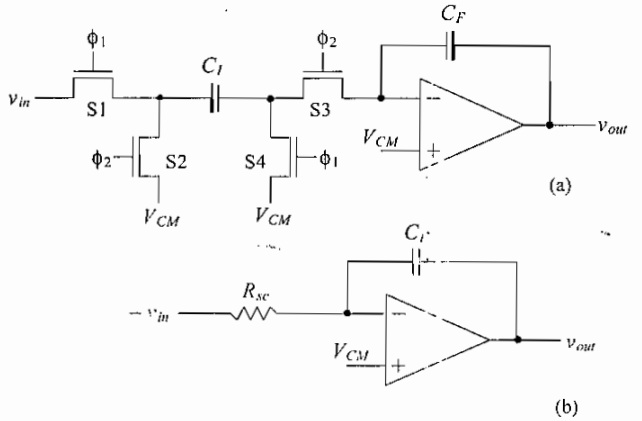


Figure 25.23 (a) A stray insensitive switched-capacitor integrator (noninverting) and (b) the equivalent continuous time circuit.

Parasitic Insensitive

The switched-capacitor integrator in Fig. 25.23 is not sensitive to parasitic or stray capacitances. This can be understood with the use of Fig. 25.24. To begin, if we realize that C_{p2} (the parasitic capacitance on the right side of C_I) is always connected to V_{CM} either through S4 or through the connection to the inverting input of the op-amp, then C_{p2} doesn't see a change in the charge stored on it. Next, the capacitance C_{p1} is charged to v_{in} when S1 is closed and then charged to V_{CM} when S2 closes. Since none of the charge stored on C_{p1} when S1 is closed is transferred to C_I, it does not affect the integrating function. A practical minimum for C_I is 100 fF set by kT/C noise (see Table 8.1).

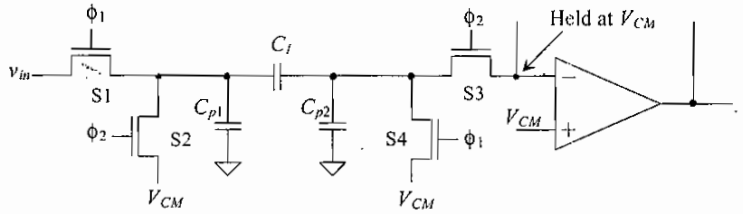


Figure 25.24 Parasitic capacitances associated with a switched-capacitor resistor.

Other Integrator Configurations

An inverting integrator configuration can be formed by simply swapping the clock signals used with S1 and S2 in Fig. 25.23 (the noninverting configuration). The gate of S1 is connected, for the inverting configuration, to ϕ_2 while the gate of S2 is connected to ϕ_1. The gain of this configuration is given by

$$\frac{v_{out}}{v_{in}} = -\frac{1}{j\omega\left(\frac{C_F}{C_I}\cdot\frac{1}{f_{clk}}\right)} \tag{25.22}$$

An example of a switched-capacitor integrator circuit that combines input signals is shown in Fig. 25.25a. Remembering that each switched-capacitor section can be thought of as a resistor, we note that the relationship between the inputs and output is

$$v_{out} = \frac{v_1}{j\omega\left(\frac{C_F}{C_1}\frac{1}{f_{clk}}\right)} + \frac{v_2}{j\omega\left(\frac{C_F}{C_2}\frac{1}{f_{clk}}\right)} - \frac{v_3}{j\omega\left(\frac{C_F}{C_3}\frac{1}{f_{clk}}\right)} \tag{25.23}$$

Figure 25.25b shows how redundant switches can be combined to reduce the number of devices used. Used alone, the basic integrator has the practical problem of integrating not only the input signal but also the offset voltage of the op-amp. In many applications, a reset switch or resistor is placed across the feedback capacitor (Fig. 25.25b). An example of a lossy integrator circuit useful in first-order filter design is shown in Fig. 25.26. The transfer function of this circuit is given by

$$\frac{v_{out}}{v_{in}} = \frac{R_4}{R_3}\left(\frac{1+j\omega R_3 C_1}{1+j\omega R_4 C_2}\right) = \frac{C_3}{C_4}\left(\frac{1+j\omega\left(\frac{C_1}{C_3}\cdot\frac{1}{f_{clk}}\right)}{1+j\omega\left(\frac{C_2}{C_4}\cdot\frac{1}{f_{clk}}\right)}\right) \tag{25.24}$$

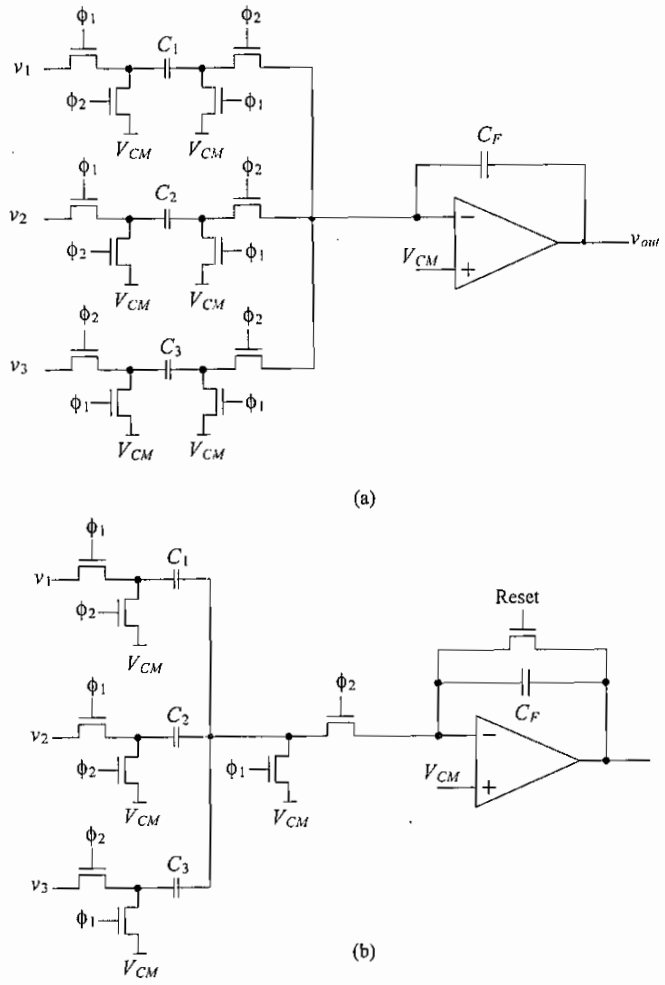


Figure 25.25 (a) Switched-capacitor implementation of a summing integrator and (b) practical implementation of the circuit combining switches and adding reset.

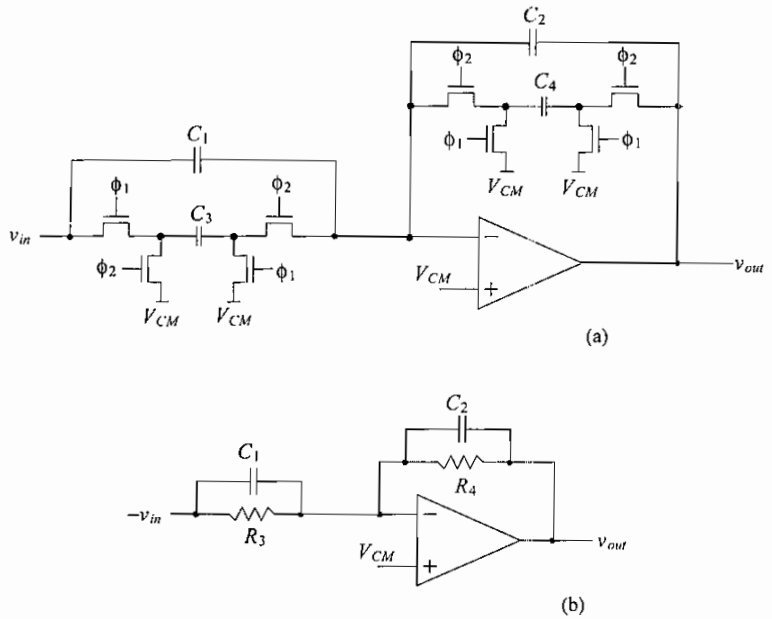


Figure 25.26 Lossy integrator (a) switched-capacitor implementation and (b) continuous time circuit.

For low-frequency input signals (low frequencies compared to the pole and zero given in Eq. (25.24), the gain of the lossy integrator is simply

$$\frac{v_{out}}{v_{in}} = \frac{C_3}{C_4} \tag{25.25}$$

which is again a precise number due to the ratio of the capacitors. Also note that the switched-capacitor resistor in the feedback loop is stray insensitive. The left side of C_4 is always connected to V_{CM}, while the right side is either connected to V_{CM} or to the output of the op-amp.

Note that we cannot eliminate the capacitor across the switched-capacitor resistor in the feedback path. If we were to do so, then the op-amp would be operating open-loop when ϕ_2 goes low. The outputs of the op-amp would then rail up at *VDD* or down at ground.

Example 25.2
Design a switched-capacitor filter with the transfer characteristics shown in Fig. 25.27.

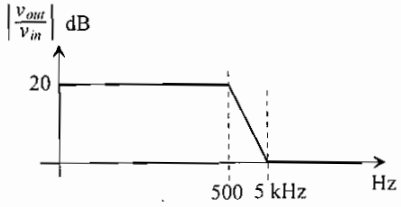


Figure 25.27 Filter characteristics for Ex. 25.2.

We can see that this transfer function has a pole at 500 Hz and a zero at 5 kHz. The lossy integrator of Fig. 25.26 will be used to realize this filter. The low-frequency gain of this circuit is 10 (20 dB). Using Eq. (25.25), we have

$$\frac{C_3}{C_4} = 10$$

while the pole and zero locations are given by

$$f_p = \frac{1}{2\pi\left(\frac{C_2}{C_4}\cdot\frac{1}{f_{clk}}\right)} = 500 \text{ and } f_z = \frac{1}{2\pi\left(\frac{C_1}{C_3}\cdot\frac{1}{f_{clk}}\right)} = 5 \text{ kHz}$$

If we set f_{clk} to 100 kHz and C_4 to 100 fF, then C_3 = 1.0 pF, C_2 = 3.2 pF, and C_1 = 3.2 pF. ■

Exact Frequency Response of a Switched-Capacitor Integrator

We will now develop an exact relationship between the switching frequency, f_{clk}, and the signal frequency ω. Referring to Fig. 25.28, we can write the output of the integrator as the sum of the previous output voltage, $v_{out(n)}$, at a time nT and the contribution from the current sample as

$$v_{out(n+1)} = v_{out(n)} + \frac{C_1}{C_F}\cdot v_{in(n)} \tag{25.26}$$

Since a delay in the time domain of T corresponds to a phase shift of ωT in the frequency domain, we can take the Fourier transform of this equation and get

$$e^{j\omega T}v_{out}(j\omega) = v_{out}(j\omega) + \frac{C_1}{C_F}\cdot v_{in}(j\omega) \tag{25.27}$$

Solving this equation for v_{out}/v_{in} gives

$$\frac{v_{out}}{v_{in}}(j\omega) = \frac{C_1}{C_F}\left(\frac{1}{e^{j\omega T}-1}\right) = \frac{C_1}{C_F}\left(\frac{e^{-j\omega T/2}}{e^{j\omega T/2}-e^{-j\omega T/2}}\right) = \frac{C_1}{C_F}\left[\frac{1}{z-1}\right] \tag{25.28}$$

where $z = e^{j\omega T}$. Remembering $f_{clk} = 1/T$ and $\omega = 2\pi f$, we get

$$\frac{v_{out}}{v_{in}}(j\omega) = \frac{1}{j\omega\left(\frac{C_F}{C_1}\cdot\frac{1}{f_{clk}}\right)}\left(\frac{\frac{\pi f}{f_{clk}}}{\sin\frac{\pi f}{f_{clk}}}\cdot e^{-j\pi f/f_{clk}}\right) \tag{25.29}$$

Ideally, the term on the right in parentheses is unity. This occurs when f is much less than f_{clk}. This equation describes how the magnitude and phase of the integrator are affected by finite f_{clk}.

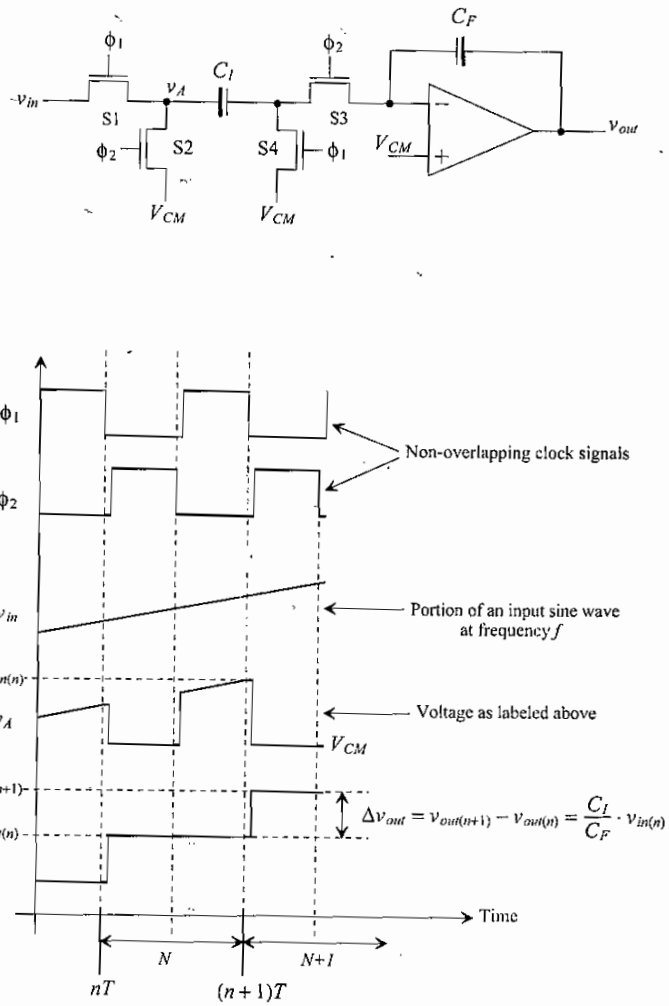


Figure 25.28 Switched-capacitor integrator used to determine the relationship between input frequency and switch clock frequency.

Capacitor Layout

An important step in the implementation of any switched-capacitor design is the layout of the capacitors. Normally, a unit-size capacitor is laid out and then replicated to the desired capacitance (as discussed in Ch. 6). For Ex. 25.2, the unit size capacitance would nominally be 100-fF (using the 1 µm process) or 10 *fF* (in the 50 nm process), as in Fig. 25.29. Note that here, in Fig. 25.29a, we are assuming that a circle can be accurately reproduced on the reticle and patterned on the wafer. In practice, effects such as the finite e-beam size (and the granularity of the grid) used to make the reticle can make this assumption questionable. Figure 25.29b shows a layout where we've tried to minimize the number of 90° corners in an effort to avoid pattern errors when etching poly2.

Again, the absolute value of the capacitors isn't important; rather, the important value is the ratio. A total of 32 of these unit-size capacitors (using a unit-size cell of 100 *fF*) would be used to achieve the larger nominally 3.2 pF capacitors in Ex. 25.2 (see Fig. 25.30). This approach (using unit elements to make the large capacitors) eliminates errors due to uneven patterning of poly to a first order. A p+ guard ring can be placed around the capacitor to help reduce coupled substrate noise. Substrate noise can also be reduced by laying the capacitor out over an n-well that is tied to *VDD*. Injected minority carriers are collected either by the p+ or the n-well (or a combination of both). If matching of the capacitors is critical, schemes that use a common-centroid layout can be used. Also, as was discussed in Ch. 20, dummy poly strips or capacitors can be placed around the array of capacitors so that the edge differences from underetching poly are eliminated. In both cases, what we are doing is trying to ensure that all capacitors see the same adjacent structures.

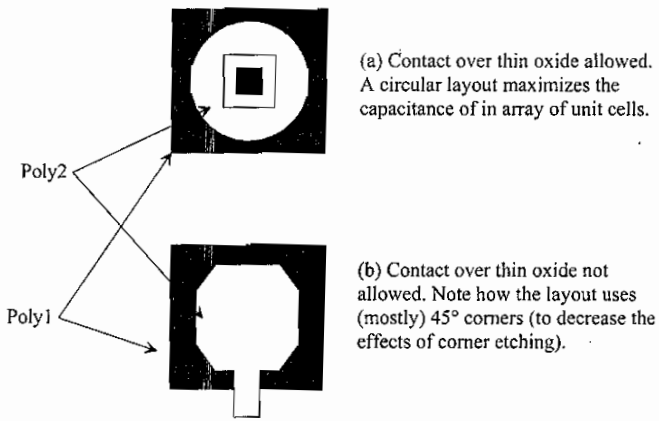


Figure 25.29 Layout of a unit cell capacitor.

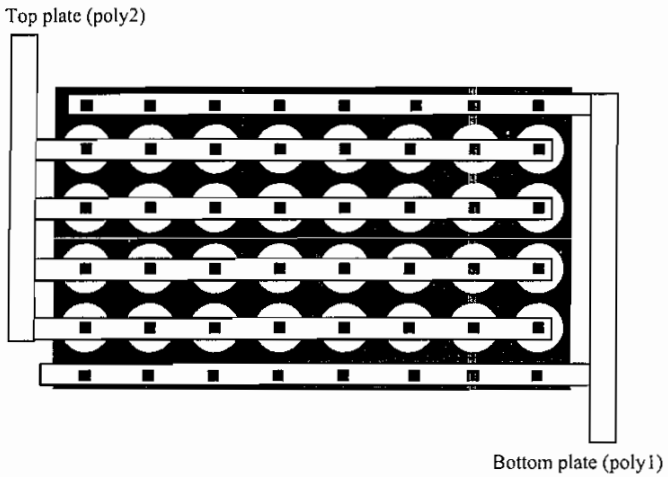


Figure 25.30 Layout of a 3.2 pF capacitor using a 100 fF unit cell.

Op-Amp Settling Time

Figure 25.31 shows an op-amp configuration in which the op-amp can source or sink current to a switched capacitor and a feedback capacitor. The time it takes to charge and discharge these capacitors is important because it directly affects the maximum switched capacitor clocking frequency, f_{clk}. The slew-rate limitations of the op-amp have been discussed in detail already. Let's now consider the limitations due to op-amp finite bandwidth. The closed-loop gain of the op-amp is given by (see Eq. [24.5])

$$A_{CL} = \frac{A_{OL}}{1 + A_{OL} \cdot \beta} \tag{25.30}$$

while the open-loop gain of an op-amp is given, with units of A/A, V/V, V/A, or A/V, by

$$A_{OL} = \frac{A_{OL}(0)}{1 + j\frac{f}{f_{3dB}}} \tag{25.31}$$

Combining these equations and assuming $1 >> 1/[\beta \cdot A_{OL}(0)]$, we get

$$A_{CL} = \frac{\frac{1}{\beta}}{1 + j\frac{f}{f_{un} \cdot \beta}} \tag{25.32}$$

where $f_{3dB} \cdot A_{OL}(0) = f_{un}$, where f_{un} may have units of Hz, Hz/Ω, or Hz·Ω ($f_u\beta$ is in Hz). The closed-loop gain reduces to a simple single-pole transfer function (see Eqs. [24.30]–[24.34]). The low-frequency gain of the circuit is $1/\beta$, while the product of f_{un} and β gives the circuit time constant of

$$\tau = \frac{1}{2\pi f_{un} \cdot \beta} \tag{25.33}$$

keeping in mind the unity-gain frequency of the op-amp, f_{un}, is a strong function of the load capacitance. For a step-input to the op-amp, a common occurrence in switched-capacitor circuits (Fig. 25.28), the output voltage of the op-amp, again neglecting slew-rate limitations, is given by

$$v_{out} = V_{outfinal}(1 - e^{-t/\tau}) \tag{25.34}$$

For the output voltage of the op-amp to settle to less than 1% of its final value requires 5τ. A "rule-of-thumb" estimate for the settling time is simply $1/(f_u \cdot \beta)$.

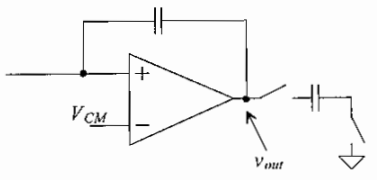


Figure 25.31 Charging and discharging a switched capacitor.

25.4 Circuits

This section presents several examples of dynamic analog circuits.

Reducing Offset Voltage of an Op-Amp

As seen in Fig. 24.4, the op-amp's offset voltage can be modeled by adding a DC voltage in series with the noninverting input of the op-amp, Fig. 25.32a. The basic idea behind eliminating the offset voltage is shown in Fig. 25.32b. A capacitor is charged to a voltage equal and opposite to the comparator offset voltage. The voltage across the capacitor is then added in series with the noninverting op-amp input to subtract away the op-amp's offset.

The dynamic analog circuit shown in Fig. 25.33 is used to implement this subtraction. For this method to be effective, *the op-amp must be stable* in the unity gain configuration. The clock signals ϕ_1 and ϕ_2 are the non-overlapping clock signals discussed earlier (see Fig. 25.28). The nonoverlapping clocks keep switches S1, S2, and S3 from being on at the same time as switches S4 and S5. Let's consider the case shown in Fig. 25.33b where ϕ_1 is high and ϕ_2 is low. The op-amp, via the negative feedback, tries to force its inverting input to V_{CM}. However, because of the offset, the inverting input is actually charged to $V_{OS} + V_{CM}$. Note that under these conditions the op-amp is removed from the inputs. The voltage across the capacitor is then V_{OS}. When ϕ_2 is high and ϕ_1 is low, Fig. 25.33c, the op-amp functions normally, assuming the storage capacitance C is much larger than the input capacitance of the op-amp.

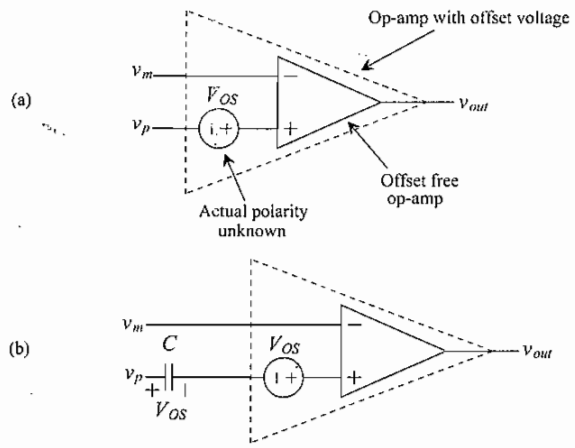


Figure 25.32 (a) Offset voltage of an op-amp modeled by a DC voltage source in series with the noninverting input of the op-amp and (b) using a capacitor to cancel the offset voltage.

Dynamic Comparator

Before we present a dynamic comparator, let's consider the *RC* switch circuit of Fig. 25.34a. When node A is connected to +1 V, node B is connected to ground. Consider what happens when the switches change positions, that is, when node A is connected to ground and the switch at node B is connected to an open. At the moment just after switching takes place, the potential at node B becomes –1 V. In other words, the voltage across the capacitor does not change instantaneously. If node A is connected back to +1 V and node B is connected back to ground a short time compared to the product of R and C, then the voltage across the capacitor remains +1 V.

A more useful circuit for CMOS is shown in Fig. 25.34b. If the switch across C_B is connected to ground when node A is connected to $V1$, then V_B, when the switches change positions, is given by

$$V_B = (V2 - V1) \cdot \frac{C_A}{C_A + C_B} \tag{25.35}$$

Figure 25.35 shows a dynamic comparator based on the inverter. When ϕ_1 is high, the voltage on the v_m input is connected to node A, while the voltage on node B is set via S3 so that the input and output voltages of the inverter are equal. (The inverter is operating as a linear amplifier where both M1 and M2 are in the saturation regions.) When ϕ_2 goes high (ϕ_1 is low since the clocks are nonoverlapping), the v_p input is connected to node A. If C_A is much larger than the input capacitance of the inverter (C_B), then the voltage change on the input of the inverter (V_B) is

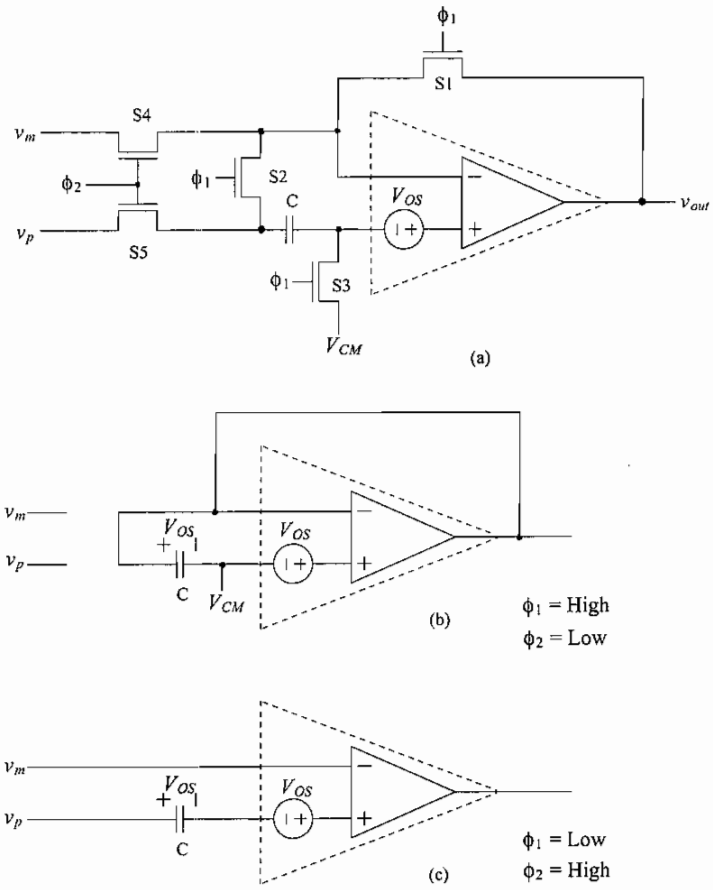


Figure 25.33 Dynamic reduction of the offset voltage.

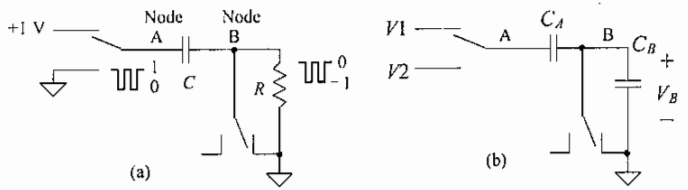


Figure 25.34 Circuits used to illustrate switching in dynamic circuits.

$$v_{in} = v_p - v_m \tag{25.36}$$

Provided the gain of the inverter is large, this change causes the inverter output to rail, that is, go to either *VDD* or ground. The output is then latched and available during ϕ_1. The gain of the comparator can be increased by using additional inverter stages.

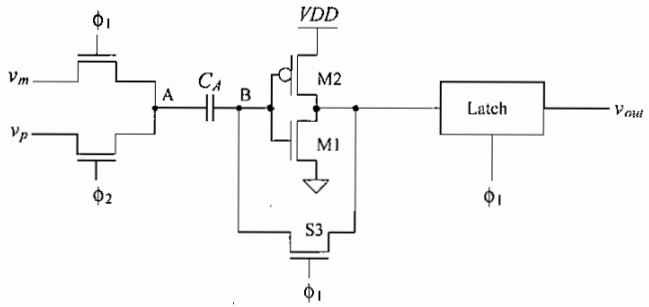


Figure 25.35 A dynamic comparator.

Another high-performance dynamic comparator configuration is based on the sense amplifier in Figs. 16.26 or 16.32 (or Fig. 16.35 so the final outputs only change on the clock's rising edge) that use positive feedback. The offset voltage of the overall comparator is reduced, using either input offset storage (IOS) or output offset storage (OOS) around the comparator preamp. Figure 25.36 shows the two types of offset cancellation techniques. In the IOS configuration in (a), the preamp must be stable in the unity feedback configuration. In the OOS configuration in (b) the MOSFETs in the preamp must remain in saturation when the offset voltage is stored on the capacitors. If a differential amplifier is used as the preamp, this condition is usually easily met.

The size of the storage capacitors is based on three important considerations: (1) preamp or latch input capacitance, (2) charge injection, and (3) kT/C noise. For the IOS scheme, the input storage capacitance must be much larger than the input capacitance of the preamp, so that the storage capacitors don't attenuate the input signals. For example, if the storage capacitors have the same capacitance value as the input capacitance of the preamp, then one-half of the input signals reaches the preamp. For the OOS scheme, the storage capacitors should be much larger than the input capacitance of the dynamic latch.

Dynamic Current Mirrors

Using dynamic techniques can reduce the effects of threshold voltage mismatches in current mirrors. Consider the circuit of Fig. 25.37. When ϕ_1 is high and ϕ_2 is low (again, these clock signals are nonoverlapping), switches S1 and S3 are on, while switch S2 is off. A current I_{ref} flows through M1, setting its gate-source voltage. This information [the gate-source voltage (actually the charge) of M1] is stored on C. When S2 closes with S1 and S3 off, a current I_{out} equal to I_{ref}, neglecting channel length modulation, flows. This circuit behaves like a current source when ϕ_2 is high and as an open when ϕ_2 is low. The circuit shown in Fig. 25.38 shows a dynamic current mirror that operates continuously.

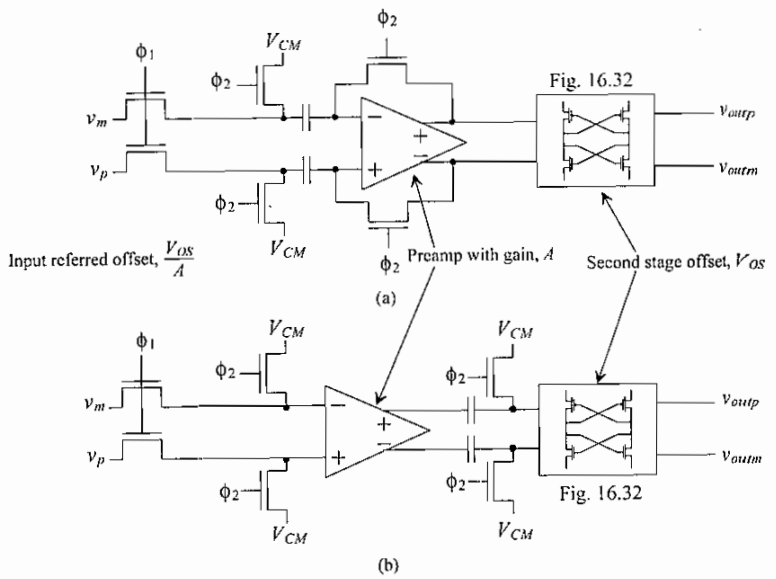


Figure 25.36 (a) Input offset storage (IOS) and (b) output offset storage (OOS).

When ϕ_1 is high, M2 sinks current, and when ϕ_2 is high, M1 sinks current. These circuits are useful in eliminating the mismatch effects and, thus, differences in the output currents, resulting from threshold voltage and transconductance parameter differences between devices. Since a single-reference current can be used to program the current in a string of current mirrors, only the finite output resistance of the mirrors causes current differences.

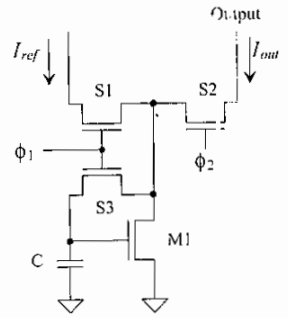


Figure 25.37 Dynamic biasing of a current mirror.

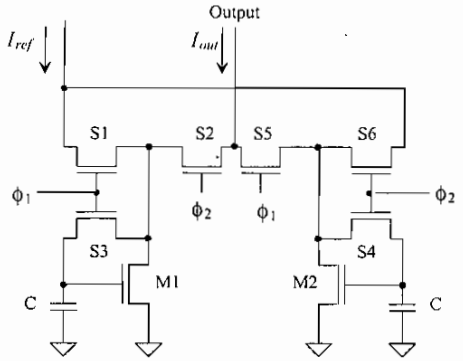


Figure 25.38 Dynamic current mirror that operates during both clock phases.

Dynamic Amplifiers

Figure 25.39 shows a dynamic amplifier. The circuit amplifies when ϕ is low and dynamically biases M1 and M2, and therefore does not amplify, when ϕ is high. If C1 and C2 are large compared to the input capacitance of M1 and M2, then the input AC signal, v_{in} , is applied to both gates. This biasing scheme makes the amplifier less sensitive to threshold and power supply variations. Other dynamic amplifier configurations exist, which have differential inputs and operate over both clock cycles.

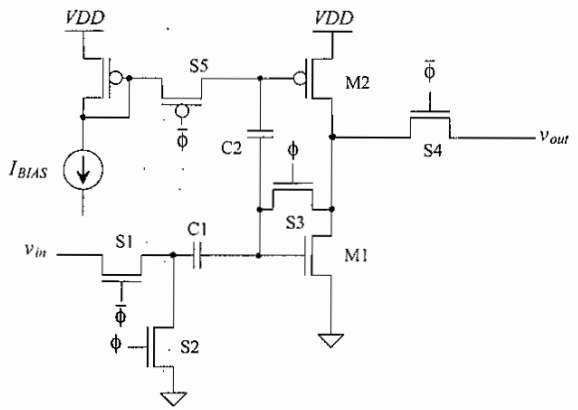


Figure 25.39 Dynamic amplifier used to reduce biasing sensitivity to power supply and threshold voltage.

ADDITIONAL READING

[1] 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, pp. 967–986, August 1983.

[2] J. Shieh, M. Patil, and B. Sheu, "Measurement and Analysis of Charge Injection in MOS Analog Switches," *IEEE Journal of Solid State Circuits*, vol. 22, no. 2, pp. 277–281, April 1987.

[3] G. Wegmann, E. Vittoz, and F. Rahali, "Charge Injection in Analog MOS Switches," *IEEE Journal of Solid State Circuits*, vol. 22, no. 6, pp. 1091–1097, December 1987.

[4] C. Eichenberger and W. Guggenbuhl, "On Charge Injection in Analog MOS Switches and Dummy Switch Compensation Techniques," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 2, pp. 256–264, February 1990.

[5] J. McCreary and P. R. Gray, "All MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part 1," *IEEE Journal of Solid State Circuits*, vol. 10, pp. 371–379, December 1975.

[6] P. W. Li, M. J. Chin, P. R. Gray, and R. Castello, "A Ratio-Independent Algorithmic Analog-to-Digital Conversion Technique," *IEEE Journal of Solid-State Circuits*, vol. SC-19, no. 6, pp. 828–836, December 1984.

[7] E. J. Kennedy, *Operational Amplifier Circuits: Theory and Applications*, Holt, Rinehart and Winston, New York, 1988.

[8] D. Johns and K. Martin, *Analog Integrated Circuit Design*, John Wiley and Sons, New York, 1997.

[9] A. B. Grebene, *Bipolar and MOS Integrated Circuit Design*, John Wiley and Sons, New York, 1984.

[10] R. W. Broderson, P. R. Gray, and D. A. Hodges, "MOS Switched-Capacitor Filters," *Proceedings of the IEEE*, vol. 67, no. 1, January 1979.

[11] K. Martin, "Improved Circuits for the Realization of Switched-Capacitor Filters," *IEEE Transactions on Circuits and Systems*, vol. CAS-25, no. 4, pp. 237–244, April 1980.

[12] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 2nd ed., John Wiley and Sons, 1984. ISBN 0-471-87493-0.

[13] R. Gregorian, K. W. Martin, and G. Temes, "Switched-Capacitor Circuit Design," *Proceedings of the IEEE*, vol. 71, no. 8, pp. 941–966, August 1983.

[14] D. J. Allstot, R. W. Broderson, and P. R. Gray, "MOS Switched-Capacitor Ladder Filters," *IEEE Journal of Solid-State Circuits*, vol. SC-13, no. 6, pp. 806–814, December 1978.

[15] R. Castello and P. R. Gray, "A High-Performance Micropower Switched -Capacitor Filter," *IEEE Journal of Solid-State Circuits*, vol. SC-20, no. 6, pp. 1122–1132, Dec. 1987.

[16] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*, 2nd ed., Oxford University Press, 2002. ISBN 0-19-511644-5.

[17] A. G. Dingwall and V. Zazzu, "An 8-MHz Subranging 8-bit A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. SC-20, no. 6, pp. 1138–1143, December 1985.

[18] B. Razavi and B. A. Wooley, "Design Techniques for High-Speed, High-Resolution Comparators," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 12, pp. 1916–1926, December 1992.

[19] S. Masuda, Y. Kitamura, S. Ohya, and M. Kikuchi, "CMOS Sampled Differential Push-Pull Cascode Operational Amplifier," *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 1211–1214, 1983.

PROBLEMS

In the following problems, where appropriate, use the short-channel CMOS process with a scale factor of 50 nm and a *VDD* of 1 V.

25.1 Using SPICE simulations, show the effects of clock feedthrough on the voltage across the load capacitor for the switch circuits shown in Fig. 25.40. How does this voltage change if the capacitor value is increased to 100fF?

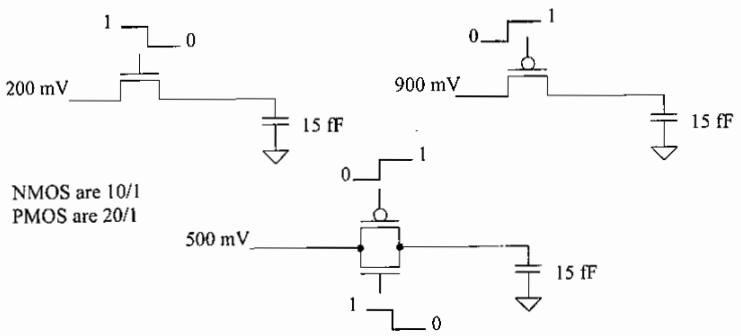


Figure 25.40 Circuits used in Problem 25.1 to show clock feedthrough.

25.2 Repeat problem 25.1 if dummy switches are used. Show schematics of how the dummy switches are added to the schematics.

25.3 Using a voltage-controlled voltage source for the op-amp (see Fig. 20.19 for example) with an open-loop gain of 10^6, use SPICE to show how the track-and-hold seen Fig. 25.8 operates with a sinewave input. What happens if the input sinewave's amplitude is above $VDD-V_{THN}$? Use a 100 MHz clock (strobe) pulse with a 50% duty cycle and an input sinewave frequency of 5 MHz. Note that the input sinewave should be centered around V_{CM} (= 500 mV).

25.4 Using the topology seen in Fig. 25.13a and the SPICE op-amp model in Fig. 25.18, show how both V_{in} and V_{CM} can be varied while the op-amp's inputs are equal, that is, $v_p \approx v_m$. Is the output common-mode voltage always at V_{CM} using the simple SPICE model for the fully-differential output op-amp in Fig. 25.18? Why or why not? Give an example supporting your answer.

25.5 Suppose, in Fig. 25.19, that instead of the two input sine waves being connected to ground they are tied (together) to a common mode signal (say a noise voltage). Show that a common mode signal (like a sine wave) won't change the circuits' output signals. The amplitude of the common-mode signal shouldn't be so large that the NMOS switches shut off.

25.6 Show that the switched-capacitor circuits shown in Fig. 25.41 behave like resistors, for $f<<f_{clk}$, with the resistor values shown.

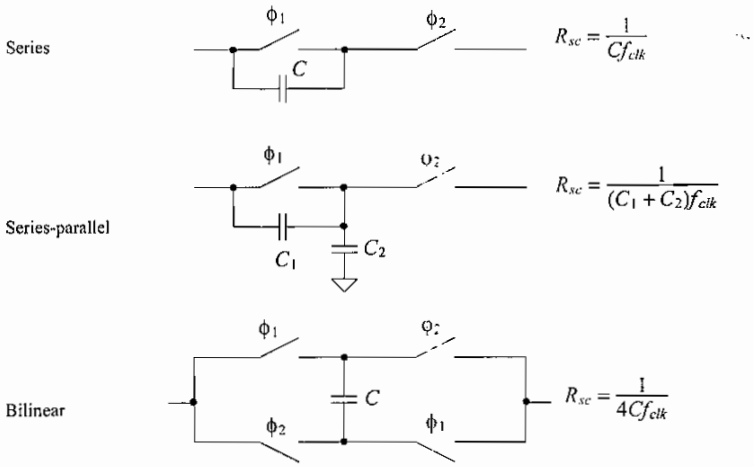


Figure 25.41 Alternative forms of switched-capacitor resistors.

25.7 Simulate the operation of the switched-capacitor resistor seen in Fig. 25.21. Plot the mean of the current flowing in the voltage sources v_1 or v_2 to show that the circuit actually behaves like a resistor. Comment on the selection of the bottom plate of the capacitor shown in Fig. 25.21a.

25.8 Comment on the selection of the bottom plate of C_F shown in Fig. 25.23.

25.9 Sketch the schematic, similar in form to Fig. 25.23, of the fully-differential switched-capacitor integrator made using a differential input/output op-amp. What is the transfer function of this topology?

25.10 Repeat Ex. 25.2 if the low-frequency gain is 40 dB and the zero is located at 50 kHz.

25.11 Using the results given in Eq. (25.29), plot the magnitude of v_{out}/v_{in} against f/f_{clk}. Comment on the resulting plot.

25.12 An important consideration in SC circuits is the slew-rate requirements of the op-amps used. In the derivation in Fig. 25.28, we assumed that a voltage source was connected to the input of the circuit. In reality, the input of the circuit is provided by an op-amp. When ϕ_1 goes high, in this figure, the capacitor C_I is charged to the input voltage v_{in} (= v_A). If C_I is 5 pF and f_{clk} is 100 kHz, estimate the minimum slew-rate requirements for the op-amp providing v_{in}.

25.13 Suppose that the op-amp in problem 25.12 is used with a feedback factor of 0.5. Estimate the minimum unity gain frequency, f_{un}, that the op-amp must possess.

25.14 Simulate the operation of the dynamic comparator shown in Fig. 25.35.

Chapter

26

Operational Amplifiers II

In the last chapter we saw that MOSFET switches cause charge injection and clock feedthrough in the circuits where they are used. To reduce the effects of these problems (and others), fully-differential op-amp topologies are used. As discussed in Sec. 25.2, the fully-differential output op-amp requires the design of a common-mode feedback (CMFB). The CMFB circuit keeps the op-amp's outputs balanced around a known voltage (generally the common-mode voltage of $V_{CM} = VDD/2$).

In this chapter we discuss the design of fully-differential output op-amps and CMFB circuits. Our discussion is centered around practical design where power, speed, offsets, and gain are (as usual) of importance. Throughout the chapter, the 50 nm process (with a *VDD* of 1 V) is used to illustrate the design techniques.

26.1 Biasing for Power and Speed

The biasing circuits we developed earlier were used for general analog design. In this chapter we want to design circuits that are used for very high speed with the least amount of power dissipation possible.

For high-speed design, we must use the minimum channel length (L = 1). However, using minimum channel lengths results in large mismatches between devices and low MOSFET output resistance (hence, why we used L = 2 for the designs presented earlier). The results are low gain and large input-referred offset voltages. Further, for low power design we want to use the lowest biasing currents possible. This (low biasing currents) is in direct conflict with high-speed design. As discussed in Ch. 9, the device speed figure-of-merit, FOM, was the transition frequency, f_T. Low biasing current is the same as low overdrive voltage. As seen in Eq. (9.55), using a low value of overdrive voltage results in slower circuits. Of course, if the overdrive voltage is too high, the MOSFET enters the triode region too soon. For general analog design, we set the overdrive voltage to 5% of *VDD*. For high-speed design, we might set the overdrive voltage to 10% of *VDD* or larger. *To minimize power and maximize speed, we will use minimum size devices.* For the NMOS, we'll use a 10/1. To match the drive, we'll use 20/1 for the PMOS devices. The question we need to answer having made these selections is: "Will these devices have the strength to drive a load capacitance quickly?"

26.1.1 Device Characteristics

Figure 26.1 shows the IV characteristics of our selected devices (an NMOS of 10/1 and a PMOS of 20/1). If we set the overdrive voltages to roughly 10% of *VDD* (here 100 mV), then knowing, from Table 9.2, the threshold voltages are 280 mV (typical), we can use gate-source voltages of, nominally, 400 mV with a corresponding drain current of 20 μA. If we want to increase the speed, then we must use a higher overdrive voltage (say bias the devices up at 50 μA).

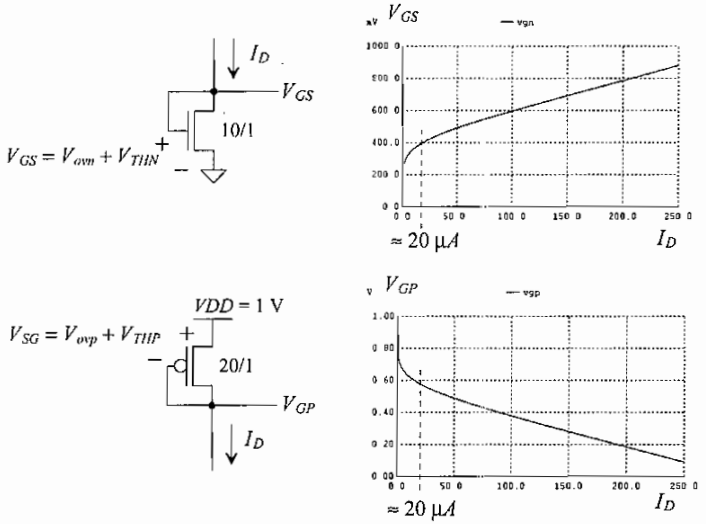


Figure 26.1 Gate-source voltages plotted against drain currents.

Next we need to determine if these devices will have the drive current needed to charge a specific size capacitor in the time required. Looking at Table 8.1 as a guide for selecting capacitor sizes, we see that using a 100 fF capacitor results in an RMS noise (because of the MOSFET switch thermal noise) of 200 μV. From Fig. 8.33 (the assumed PDF for thermal noise), the peak-to-peak value of this noise is (roughly) 1.2 mV. In this chapter *we'll use a load capacitance of 250 fF* for a peak-to-peak noise of 750 μV. Looking at the simulation results in Fig. 26.1, we see that a reasonable pulsed drain current estimate for the devices is 100 μA. The rate we can charge the load capacitor is then

$$\frac{dV_{out}}{dt} = \frac{I}{C_L} = \frac{100\,\mu A}{250\,fF} = 400\,mV/ns \qquad (26.1)$$

If our clock frequency, f_{clk}, is 100 MHz, then half a clock cycle is 5 ns and the device sizes and bias conditions will likely be adequate (remembering that *VDD* is 1 V and V_{CM} is 500 mV). However, if our clock frequency approaches 1 GHz (or the load capacitance increases), then we must increase the widths of the devices (to get more drive) and the overdrive voltages (to get more speed).

26.1.2 Biasing Circuit

When designing the bias circuits earlier, Fig. 20.47 for example, our goals were to provide biasing for general analog design. Here, in this chapter, our goal is to design a bias circuit for our op-amp designs with minimum power dissipation. When designing biasing circuits fully-differential topologies have some advantages over the single output op-amps presented in Ch. 24. For example, our earlier bias circuit provided biasing, V_{pcas} and V_{ncas}, for the floating current sources used in the class AB output stages (see Fig. 24.29 for example). These bias voltages won't be needed here.

Towards understanding this last statement, consider the two-stage op-amp seen in Fig. 26.2. Consider what happens if the noninverting op-amp input, v_p, increases relative to the inverting input, v_m. The drain voltage of M1R drops while the drain voltage of M1L rises. The decrease in the gate voltage of M6R causes it to turn on and the output to go high. At the same time, the increase in the drain voltage of M1L causes M4R to shut off and thus so does M7R (giving class AB operation). In simple terms, we can yank the gates of M4 or M6 down independent of the diff-amp's tail current.

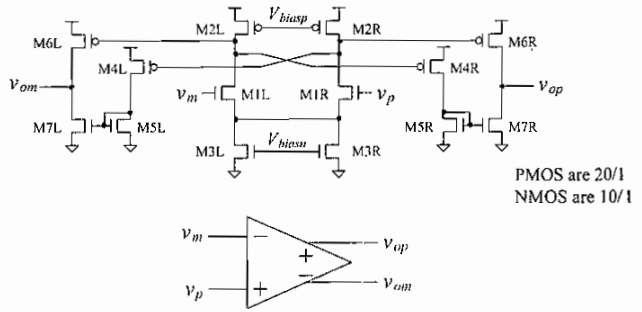


Figure 26.2 A two-stage fully-differential op-amp. Compensation and CMFB are not shown. Output stage operates class AB. See discussion in the next section concerning the output voltage of the diff-amp.

Layout of Differential Op-Amps

One of the common things we do in this chapter is draw schematics that are symmetrical. If we were to draw a line down the middle of the op-amp in Fig. 26.2, separating the left and right sides of the schematic, we could fold the left side of the schematic directly over onto the right side of the schematic and see a perfect match. For example, instead of drawing the diff-amp's tail current source (M3) as a single MOSFET with twice the width, we've drawn it as two MOSFETs in parallel, each having the same width. *Drawing schematics in this fashion is useful when doing layout.* We can fold the schematic in half and lay out like devices, e.g., M6L and M6R, directly next to each other. Further, then the outputs (and inputs) of the op-amp are laid out right next to each other.

Self-Biased Reference

Figure 26.3 shows the bias circuit we'll use in this chapter (see also Fig. 20.22 and the associated discussion). We used the length of 2 MOSFETs for the added amplifier to minimize power dissipation and boost the amplifier's gain. The current pulled from *VDD* is approximately 50 µA. Figure 26.4 shows the simulation results of current variations vary with changes in *VDD*. Notice how V_{biasn} is close to the 400 mV, and V_{biasp} is close to *VDD* – 400 mV (of course the absolute value of V_{biasp} varies with *VDD* but the ideal value of V_{SG} for the PMOS devices will be 400 mV). Note that, as discussed in Sec. 20.1.4, it is important to ensure that the reference is stable over all possible operating conditions and loads (connected to V_{biasn} and V_{biasp}). Notice that we aren't discussing how the reference currents vary with process shifts in the MOSFETs and the resistor. This is an important practical concern.

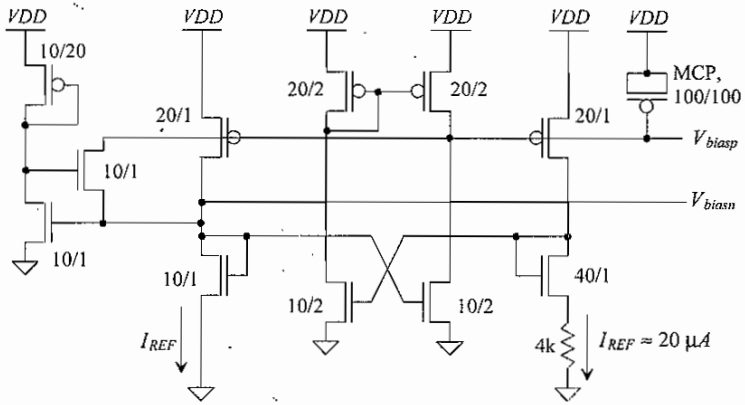


Figure 26.3 Biasing circuit used in this chapter. This bias circuit pulls approximately 50 microamps.

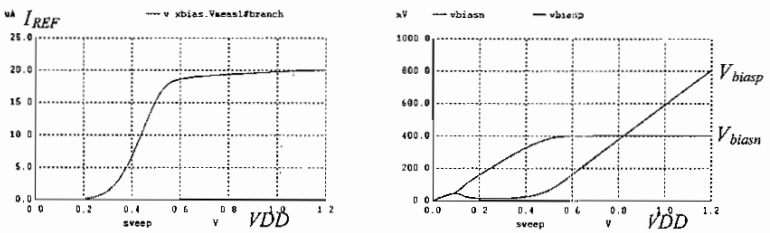


Figure 26.4 Simulating how the reference current changes with VDD.

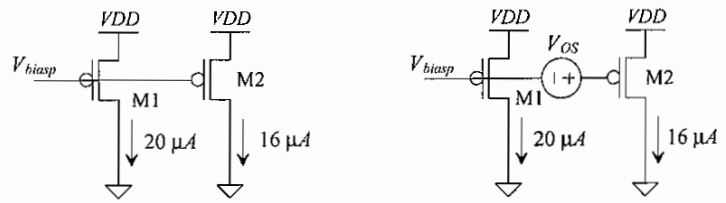
26.2 Basic Concepts

Before we discuss the design of op-amps, let's look at some basic concepts that will be useful when evaluating a specific op-amp topology.

Modeling Offset

In a real circuit, especially an analog circuit designed with minimum length devices, mismatches can be a significant factor in the selection of an op-amp topology. In a SPICE simulation, all of the MOSFETs are perfectly matched. We have to come up with a method, in SPICE, to determine an op-amp's sensitivity to offsets.

Consider the circuit in Fig. 26.5. In (a) the measured currents should be equal. Both MOSFETs V_{SG}s and V_{SD}s are equal (and they are the same size). However, for whatever reason (e.g., threshold voltage mismatch), there exists a mismatch between the two devices that causes a difference in their drain currents. We can model this mismatch in a SPICE simulation, as seen in (b), by adding a DC voltage source in series with the gate of M2. For a general design, we can insert these offset voltages at various points in the circuit and verify that the circuit still functions correctly. The next question that needs answering is: "What value of V_{OS} should be used?" While no absolute answer can be given here, **we'll use a V_{OS} of 50 mV** (5% of *VDD* or roughly 20% of the threshold voltage). The reader might feel that this value is way too high. However, if we can design circuits that function properly over the process, voltage, and temperature (PVT) variations with an offset this large, it is likely the op-amp will be difficult to "break."



(a) M1 and M2 are mismatched.

(b) M1 and M2 are perfectly matched (as in a SPICE simulation).

Figure 26.5 How we add an offset into the circuit to model mismatch.

A Diff-Amp

Figure 26.6a shows a basic diff-amp without an offset. This offset-free diff-amp has simulated output voltages of 700 mV. The diff-amp in (b) is simulated with an input-referred offset voltage of 50 mV. The polarity of the offset doesn't matter because it simply swaps the two output voltage values. In other words, we could have put the same polarity offset on the other input of the diff-amp and swapped the voltages on the output of the op-amp. We put the offset voltage on the op-amp's input because the input has the greatest effect on the diff-amp's output voltages. Note: we can think of the 50 mV as an input signal causing an output signal difference, from the ideal 700 mV, of 900 mV −550 mV or 350 mV (assuming linear operation). This indicates the gain of the diff-amp is only 7!

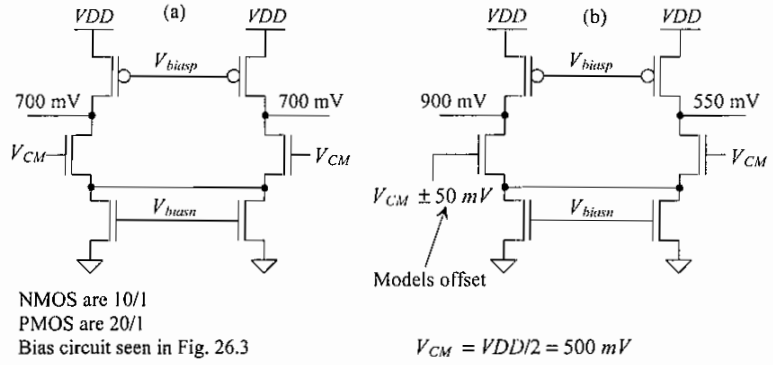


Figure 26.6 Comparing the diff-amp's output voltages with and without an offset.

The simulation results seen in Fig. 26.6 give some practical information that we need to consider. To begin, remember from Ch. 24 that we frequently use the diff-amp's output to bias the next stage (see Figs. 22.8 and 24.2). With a *VDD* of 1 V and a V_{SG} of nominally 400 mV, V_{biasp} is roughly 600 mV, as seen in Fig. 26.4. However, as seen in Fig. 26.6a, the diff-amp's output is 700 mV. The result, when the diff-amp is used with a second-stage and feedback, is an additional input-referred offset. It would be nice to know that the outputs of the diff-amp, in the ideal case, go to a known value (preferably a voltage that can be used to bias the next stage). The diff-amp in Fig. 26.6 is an example of two current sources, the PMOS biased with V_{biasp} and the NMOS biased with V_{biasn}, fighting each other for control of the output voltage. In general, we want to avoid this situation.

A Single Bias Input Diff-Amp

Figure 26.7 shows a diff-amp that generates its own bias reference for the PMOS devices. Notice that the two gate-drain-connected PMOS devices behave simply like a MOSFET with twice the width of the other two PMOS devices. Similarly, the four NMOS devices that are biased from V_{biasn} behave like a MOSFET with four times the width of the other NMOS devices. When the diff-amp's inputs are equal, the same current, *I*, flows in all of the MOSFETs. If the + diff-amp input is raised significantly above the – input, all of the bias tail current flows in the left two NMOS devices of the diff-amp (each will conduct 2*I*). The current flowing in the gate-drain-connected PMOS device, in either case, is the same, 2*I*, keeping the PMOS's gate voltage constant. The outputs of this diff-amp can be used to bias the next stage. Offsets, however, cause the diff-amp's outputs to vary from their ideal values. One drawback to using this diff-amp is that it dissipates twice the power of the diff-amp in Fig. 26.6. Another drawback is the larger input capacitance.

The Diff-Amp's Tail Current Source

Notice that we are not using a cascode tail current source to bias the diff-amps in this chapter, as we did in earlier chapters. We're avoiding the cascode structure because we'd need an additional bias voltage (resulting in more power dissipation). In a

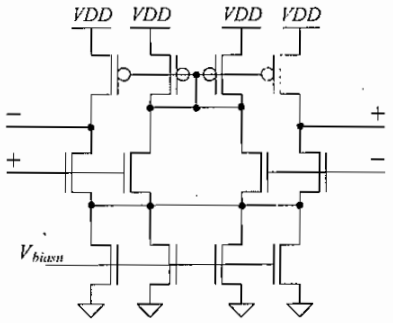


Figure 26.7 A fully-differential diff-amp that generates its own bias for the PMOS.

fully-differential op-amp topology (both the inputs and the outputs of the op-amp, with feedback, are double-ended signals swinging around V_{CM}), the input common-mode voltage of the op-amp is constant (= V_{CM}). The common-mode rejection ratio isn't as important when the input common-mode voltage of the op-amp doesn't vary.

Using a CMFB Amplifier

Another possible way to set the diff-amp's output voltages to a known value is seen in Fig. 26.8. An amplifier (called a common-mode feedback amplifier or CMFB amplifier) is used to amplify the difference between the average of the diff-amp's outputs and V_{biasp}. If the gain of the CMFB amplifier is large, then the average of the two outputs will be very close to V_{biasp}. Note that the CMFB amplifier's output signal, V_{CMFB}, is common, through M1L and M1R, to both outputs. Any variation in V_{CMFB} affects each output by the same amount. This is important because all we want the CMFB amplifier to do is make

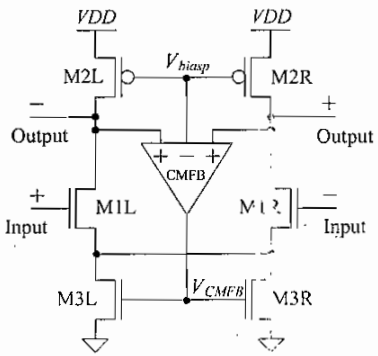


Figure 26.8 Using a common-mode feedback (CMFB) amplifier to set the output voltages.

sure that the diff-amp's outputs vary around V_{biasp}. The CMFB amplifier shouldn't affect the differential amplification in the diff-amp. When the diff-amp's outputs are equal (the inputs to the diff-amp are equal and neglecting offsets), they should be V_{biasp}.

Figure 26.9 shows one possible implementation of a CMFB amplifier. When the + inputs (noticing the PMOS connected to these inputs have half the width of the other PMOS) are equal to V_{biasp}, the currents that flow in M1 and M2 are 20 µA. This provides the proper mirroring action to the diff-amp (M3L, R) in Fig. 26.8. If the average of the + inputs moves above the – input, the current flowing in M2 decreases and the current in M1 increases. This causes V_{CMFB} to increase. The result is that M3L and M3R turn on more and pull both of the outputs in Fig. 26.8 down (until their average is equal to V_{biasp}). Several concerns exist with this topology. To begin, whenever we employ feedback, we must be concerned with stability. To stabilize the CMFB loop, we can add capacitors to the outputs of the diff-amp (inputs to the CMFB amplifier). We may be able to stabilize the CMFB loop by adding capacitors to the CMFB amplifier's outputs. However, since we will need compensation capacitors for compensating the differential action of the op-amp, we might as well use these capacitors for both loops. The other concern with the CMFB amplifier in Fig. 26.9 is the input common-mode range. When the CMFB amplifier's + inputs move significantly away from the – input, we don't get the correct balancing. This keeps large signal swings on the diff-amp's outputs in Fig. 26.8 from being balanced around V_{biasp}.

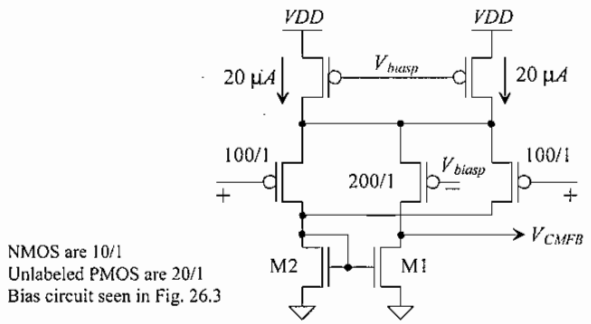
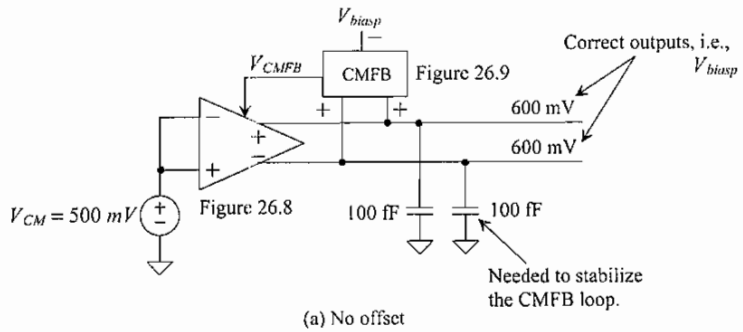
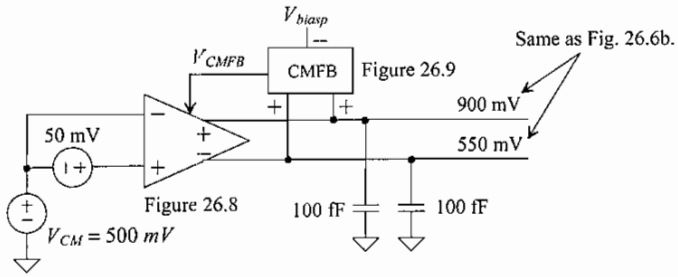


Figure 26.9 Implementation of the CMFB amplifier in Fig. 26.8.

Figure 26.10 shows simulation results using the CMFB amplifier in Fig. 26.9 with the amplifier in Fig. 26.8. The addition of the capacitors stabilizes the CMFB loop *(we need to discuss this further)*. In (a) we see the correct balancing. The CMFB loop adjusts the diff-amp's output voltages to V_{biasp}. In (b) we re-simulate with an offset and see that the CMFB circuit isn't balancing the outputs. If it were, we would see one output above V_{biasp} by some amount and the other output below V_{biasp} by the same amount. With these output voltages (the inputs to the CMFB amplifier), one of the PMOS devices on the input of the CMFB is off and we don't get proper amplifier action. *We must use a CMFB circuit that can balance the outputs over the entire range of diff-amp output voltages.*



(a) No offset



(b) With a 50 mV offset. Note how the CMFB isn't doing anything.

Figure 26.10 Simulating the operation of the CMFB circuit in Fig. 26.9.

Compensating the CMFB Loop

Consider the schematic seen in Fig. 26.11. The diff-amp behaves like an operational transconductance that can be compensated, as discussed in Sec. 24.3 (see Eq. [24.44]). The CMFB loop can be compensated in a similar fashion. The AC common-mode signal is represented in this schematic as v_c. If the gain of the CMFB amplifier is A_{cm}, then following the procedure leading to Eq. (24.44), we can write the unity-gain frequency of the CMFB loop as

$$f_{un,cm} = \frac{A_{cm} \cdot g_{mn}}{2\pi C_L} \tag{26.2}$$

If we want to compensate the CMFB loop with the same load capacitance used to compensate the differential forward signal path, then we must ensure that the gain of the CMFB amplifier is less than or equal to unity,

$$A_{cm} \leq 1 \tag{26.3}$$

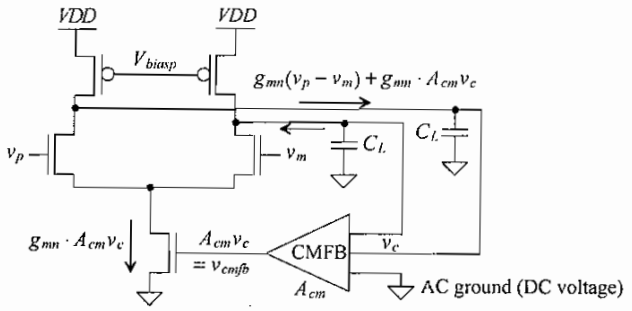


Figure 26.11 Schematic view of differential and CM feedback.

Reviewing the CMFB amplifier in Fig. 26.9 that has a current mirror active load, we see that A_{cm} is greater than 1. Removing the capacitors in Fig. 26.10 and resimulating will show that the CMFB loop is unstable. The added capacitors are relatively large and will overcompensate the differential signal path of the diff-amp. What we need to do is reduce the gain of the CMFB amplifier or reduce the CMFB loop's forward gain.

Towards reducing the gain, examine the CMFB amplifier seen in Fig. 26.12. This is the same amplifier topology seen in Fig. 26.9 except that here we've used a diode-connected load instead of a current mirror load (to reduce the gain). The schematic is drawn symmetrical around its center for ease of layout, as discussed earlier. Note that we used the common-mode voltage in this schematic, V_{CM}, as the voltage that the outputs of the amplifier will swing around (the more general case) rather than V_{biasp} as used in Fig. 26.10. Using this CMFB amplifier in the circuits of Fig. 26.10 (where the diff-amp of Fig. 26.8 has a low gain) won't precisely balance the outputs. The loop gain around the CMFB loop isn't large enough for proper operation. As we saw in Ch. 24, low first-stage gain can be remedied by using a telescopic input (cascode-load diff-amp) or a folded-cascode OTA. The other problem with this CMFB amplifier, again, is the limited allowable swing on the + inputs. As we've already seen, the input common-mode range limits the range of output voltages this CMFB can balance properly.

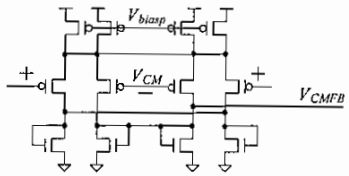


Figure 26.12 A CMFB amplifier with a gain of nominally unity.

Towards reducing the CMFB loop's forward gain, consider breaking the diff-amp's tail current up into parts, as seen in Fig. 26.13. The CMFB signal is applied to only one gate of the tail current. When compared to the topology in Fig. 26.8, the forward gain of the CMFB loop is halved. Further reduction can be implemented by adjusting the sizes of the transistors to further reduce the strength of the V_{CMFB} signal. *This is a common practical way of making CMFB loops stable.*

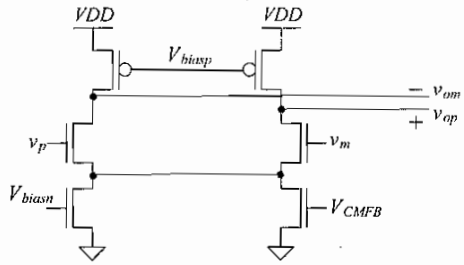
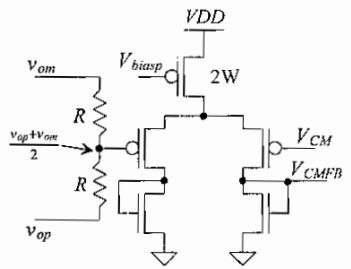


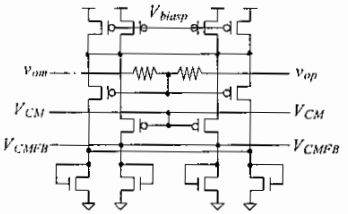
Figure 26.13 Reducing the forward gain of the CMFB loop.

Extending the CMFB Amplifier Input Range

The problem with the previous CMFB amplifier topologies based on a diff-amp is the diff-amp's limited input range. It's desirable to have a CMFB amplifier that functions over the entire range of possible amplifier output voltages. Towards this goal, consider the conceptual schematic in Fig. 26.14a. The resistors average the two outputs. This average is compared with the common-mode voltage (or a bias voltage as used in Fig. 26.8). Figure 26.14b shows the practical implementation of the amplifier for symmetry (at the cost of extra power dissipation). The practical problem with this topology is the



(a) Using resistors to average differential output signals.



(b) Symmetrical implementation of the CMFB circuit in (a).

Figure 26.14 Increasing the CMFB input signal s.

loading by the resistors. If we were to connect this CMFB amplifier in the circuit configuration of Fig. 26.8, the resistors, unless they are huge (>100k) would load the differential amplifier and lower its gain. This topology is used, most often, on the output of an op-amp that has output buffers (and can thus drive resistive loads).

When using resistors for averaging in high-speed applications, we may have some parasitic effects that should be considered. The output signals have to charge, through the averaging resistors, the input capacitance of the MOSFET, as seen in Fig. 26.15. To ensure that the balancing action works at high speeds, capacitors (shown dashed in the figure) can be added, shunting the resistors. These capacitors can be very important if the size of the resistors is increased to reduce their loading on the output of the amplifier.

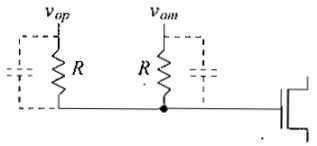


Figure 26.15 Adding parasitic capacitances across the resistors to compensate for the input capacitance of the MOSFET.

Dynamic CMFB

Figure 26.16 shows a switched-capacitor (SC) implementation of a CMFB circuit. The clocks, as in all SC circuits, are nonoverlapping (never high at the same time) clock signals, as seen in Fig. 25.28. The SC resistors are formed with the C_1 capacitors. The C_2 capacitors are used for the high-speed averaging just discussed (the dashed capacitors in Fig. 26.15). The SC resistors, as we'll see in a moment, perform both the averaging and the differencing needed in a CMFB amplifier. If the ϕ_2 controlled switches connected to amplifier's outputs, v_{op} and v_{om}, are transmission gates, the circuit can provide balancing from *VDD* to ground.

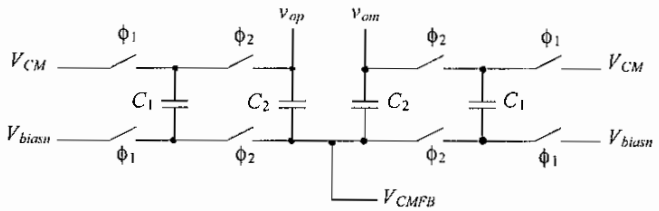


Figure 26.16 A switched-capacitor CMFB circuit.

To describe the operation of this circuit, consider the case when ϕ_1 is high. During this time, the total charge stored on each C_1 capacitor is

$$q_1 = 2 \cdot (V_{biasn} - V_{CM}) \cdot C_1 \qquad (26.4)$$

When the ϕ_1 switches shut off, the ϕ_2 switches turn on. The total charge on both C_1 capacitors is then

$$q_2 = (V_{CMFB} - v_{op}) \cdot C_1 + (V_{CMFB} - v_{om}) \cdot C_1 \qquad (26.5)$$

The change in V_{CMFB} is proportional to the difference in q_1 and q_2 or

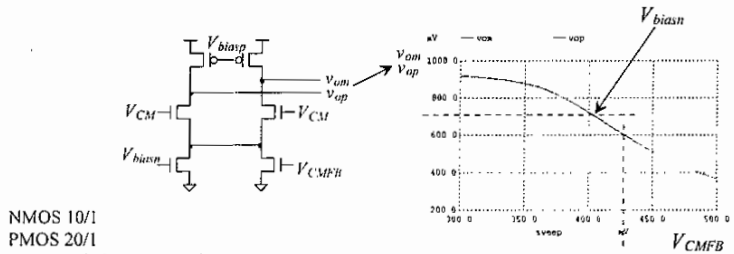
$$\Delta V_{CMFB} \cdot 2(C_1 + C_2) \propto (q_1 - q_2) \qquad (26.6)$$

This equation is important because it shows the CMFB voltage will continue to change until the two charges, q_1 and q_2 , are equal. Note that if v_{op} and v_{om} are balanced around V_{CM} , their net contributions, when ϕ_2 goes high, to V_{CMFB} are zero. Looking at the difference in the charges, we get

$$q_1 - q_2 = 2C_1\left(V_{biasn} - V_{CMFB} + \frac{v_{op} + v_{om}}{2} - V_{CM}\right) \qquad (26.7)$$

This equation is quite interesting. Ideally, V_{CMFB} is equal to V_{biasn} , and the average of the outputs is equal to the common-mode voltage. If the actual value of V_{CMFB} , for balanced outputs, is 10 mV offset from V_{biasn} , then the average of the outputs will be 10 mV offset from V_{CM}. The offsets can occur because of improper device sizing (under ideal conditions the currents don't sum correctly) or mismatches.

As an example of where this offset can come from, consider the amplifier seen in Fig. 26.17 (also in Figs. 26.6 and 26.13). As seen in Fig. 26.6a, when V_{CMFB} is V_{biasn} or roughly 400 mV, the outputs of the diff-amp are 700 mV. As seen in Fig. 26.10 and the associated discussions, it can be useful, for next stage biasing, if the outputs are set to V_{biasp} (600 mV). From Fig. 26.17, the value of V_{CMFB} at this output voltage is roughly 425 mV or a 25 mV offset.



NMOS 10/1
PMOS 20/1
Bias circuit from Fig. 26.3

Value of CM feedback voltage when the outputs are 600 mV.

Figure 26.17 Plotting the output voltages as a function of the CM feedback voltage.

26.3 Basic Op-Amp Design

Reviewing the data in Table 9.2, we see that the open circuit gains are 25 (NMOS) and 50 (PMOS). In this chapter we both increased the biasing current and reduced the channel length from the values used in Table 9.2. Each of these changes has the effect of reducing the MOSFET's open circuit gain. If, for example, the open circuit gains are both now 10, then a common-source amplifier with current source load will have a gain of 5. A cascode amplifier will have a gain of 25, and a two-stage op-amp using a cascoded first-stage a gain of only 125. In other words, we adjusted our biasing for high-speed operation but we are going to face some issues with getting large open-loop gain.

This is a good time to remember the useful simulation netlists that we've developed. Figure 26.18 shows the IV curves, output resistance, and transconductance for a 10/1 NMOS and a 20/1 PMOS based on Figs. 9.31 to 9.33. Notice in (a) that the NMOS's drain current at a V_{GS} of 400 mV and a V_{DS} of 100 mV is approximately 13 μA. The PMOS's drain current under the same conditions, (b), is closer to 9 μA. In (c) and

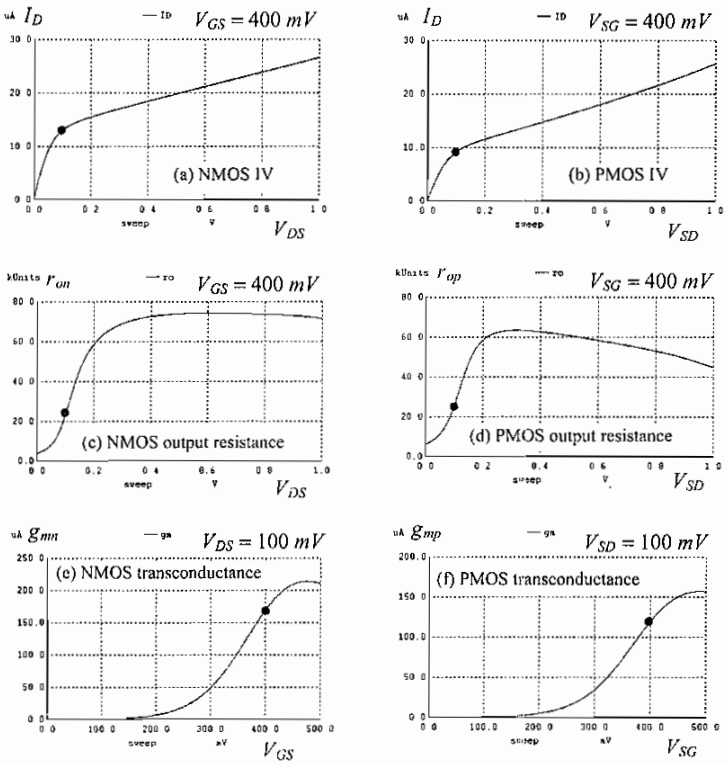


Figure 26.18 Characteristics of NMOS (10/1) and PMOS (20/1) devices.

(d), at a V_{DS} of 100 mV, the output resistance is only 25k. Using the transconductances in (e) and (f), the open circuit gains are, roughly, 4.375 (NMOS) and 3.125 (PMOS). In the actual circuits, the biasing points will vary (but in any case the gain of single stages will be low).

The Differential Amplifier

Figure 26.19 shows a cascode load diff-amp based on the topology seen in Fig. 26.7. Seen in the figure are typical values for the voltages in the circuit assuming gate-source voltages of 400 mV and, for the bottom two rows of NMOS devices, drain-source voltages of 100 mV. Notice how we used V_{biasn} to bias the second row of PMOS devices, which puts 200 mV across the drain-source voltages of the PMOS devices. This results in larger gain and reduced output swing. Diff-amp output swing is not an issue if this amplifier is used as the first-stage in a two-stage op-amp.

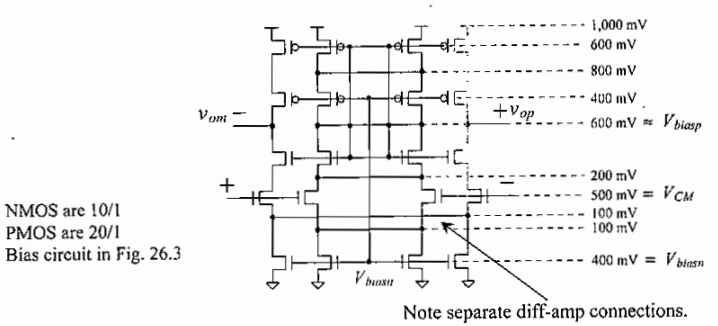


Figure 26.19 Fully-differential cascode diff-amp.

Figure 26.20 shows how the output voltages of the diff-amp in Fig. 26.19 vary with changes in V_{CM}. A 200 mV change in the common-mode voltage results in a, roughly, 50 mV change in the diff-amp's common-mode output voltages. Looking at Fig. 26.1, we see that the change in the drain current will be, again roughly, ± 5 μA around the quiescent value set with V_{CM} equal to 500 mV.

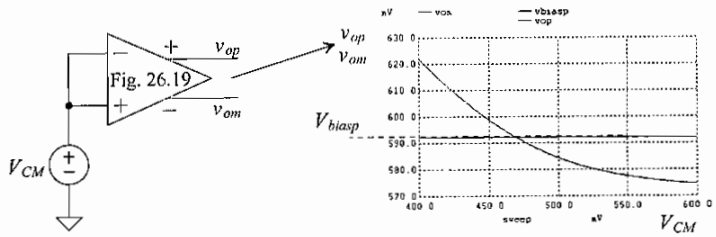


Figure 26.20 Varying the common-mode voltage and looking at the output.

Figure 26.21 shows the DC characteristics of the diff-amp. The gain is approximately 40. An important concern is how the mismatches in the MOSFETs used in the diff-amp affect the operation and biasing of the amplifier. Before discussing this issue, let's add the second stage and CMFB circuitry to form an op-amp. Note that, with a diff-amp gain of 40, and a second stage gain of 10, our op-amp's open loop gain will only have a value in the hundreds.

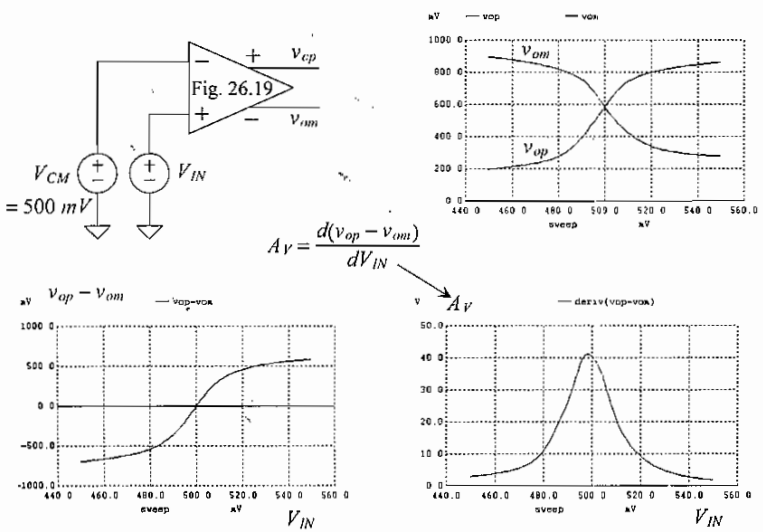
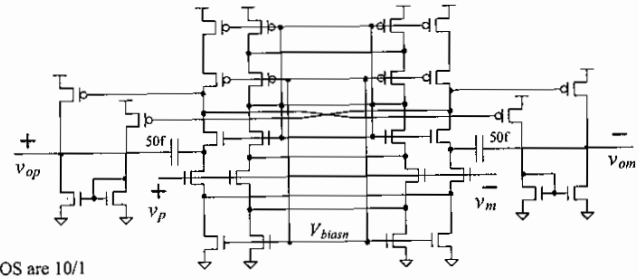


Figure 26.21 DC behavior and gain of the diff-amp in Fig. 26.19.

Adding a Second Stage (Making an Op-Amp)

Figure 26.22 shows a two-stage op-amp without CMFB circuit. The second stage of the op-amp operates class AB, as seen in Fig. 26.2, and the associated discussion. We've spent a considerable amount of time discussing how the output voltages of the diff-amp are approximately V_{biasp}. It should be clear after studying the op-amp in Fig. 26.22 why this is important. This voltage sets the quiescent current flowing in the output stages. There are eight vertical branches in this op-amp, so we can estimate the current pulled from *VDD* under quiescent conditions as 160 µA.

We used 50 fF capacitors for compensation in this op-amp. We can estimate the slew-rate limitations caused by the diff-amp driving the compensation capacitor, Eq. (22.34), as $20\ \mu A/50\ fF = 400\ mV/ns$. Using a class AB output stage, we don't have slew-rate limitations associated with driving a load capacitance from a constant current source. As discussed at the beginning of the chapter, the output MOSFET's drain currents can be pulsed to a value greater than 100 µA. When the op-amp is driving a 250 fF capacitive load, the speed limitations associated with charging the load capacitance are similar to the limitations we get when the diff-amp drives the compensation capacitor.



NMOS are 10/1
PMOS are 20/1
Bias circuit in Fig. 26.3

Figure 26.22 Basic two-stage op-amp without CMFB.

Figure 26.23 shows the DC characteristics of the op-amp in Fig. 26.22 where, once again, we've held the inverting op-amp input at the common-mode voltage and swept the voltage on the noninverting input. Notice how the outputs swing all the way from ground to *VDD* (= 1V). Further notice how the differential output voltage swings from −1 to +1 V (a doubling in the output swing as discussed in Sec. 25.2.) The op-amp's DC gain is approximately 500 without a DC load. Notice how the two op-amp outputs cross at 400 mV (not at V_{CM} where they should). We'll discuss the CMFB in a moment.

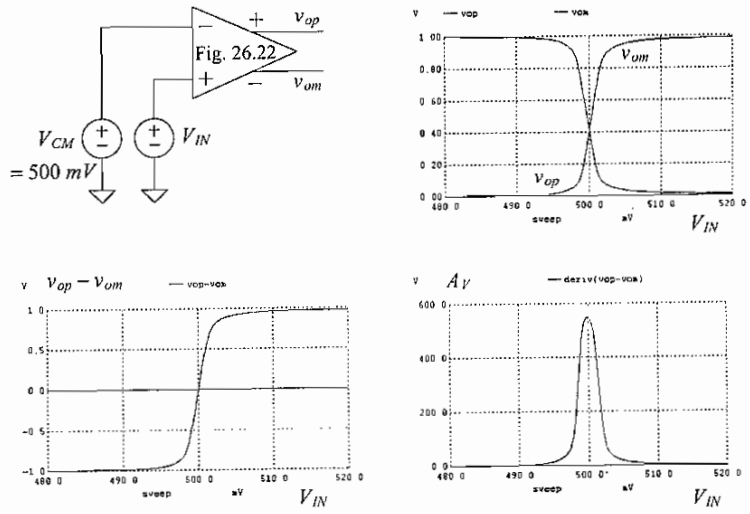


Figure 26.23 DC behavior and gain of the op-amp in Fig. 26.22.

Step Response

To determine both the stability of the op-amp in Fig. 26.22 and the settling time under certain loading conditions, consider the configuration seen in Fig. 26.24. We used relatively small resistors, 20k, in this circuit to reduce the RC time constant associated with the outputs of the op-amp charging the input capacitance of the op-amp. If, for example, the input capacitance of the op-amp is 25 fF (from parasitics and the MOSFETs used on the op-amp's input), then the RC time associated with charging this capacitance through a 20k resistor is 0.5 ns. As seen in Fig. 26.24, the settling time is approximately 2.5 ns so this RC time can have a significant effect on the settling time and stability of the circuit (important).

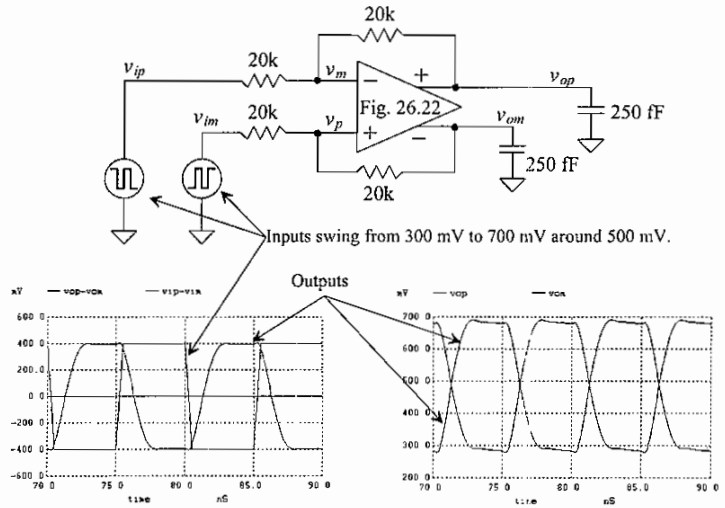


Figure 26.24 Step response of the op-amp in Fig. 26.22 driving 250 fF load capacitors and 20k feedback resistors.

Notice how, in Fig. 26.24, we used input signals that don't swing rail-to-rail. Since we don't have a CMFB circuit in the op-amp, the exact common-mode output voltage is an unknown. It may be 400 mV or it may be 600 mV. We get some help in setting the circuit's output common-mode level by using input signals with common-mode voltages of 500 mV and DC feedback. Looking at the simulation results in Fig. 26.24, we might get a false sense of not needing a CMFB circuit. To illustrate this is indeed a false sense, consider the sample-and-hold seen in Fig. 26.25 (see also Fig. 25.19). When the ϕ_1 switches are closed, the op-amp's inputs and outputs should be held to $V_{CM} \pm V_{OS}$ (the op-amp is placed in the follower configuration). As seen in the figure, the outputs during this time are driven close to 400 mV. On closer inspection, we see the output common- mode level is wandering downwards until eventually the op-amp shuts off.

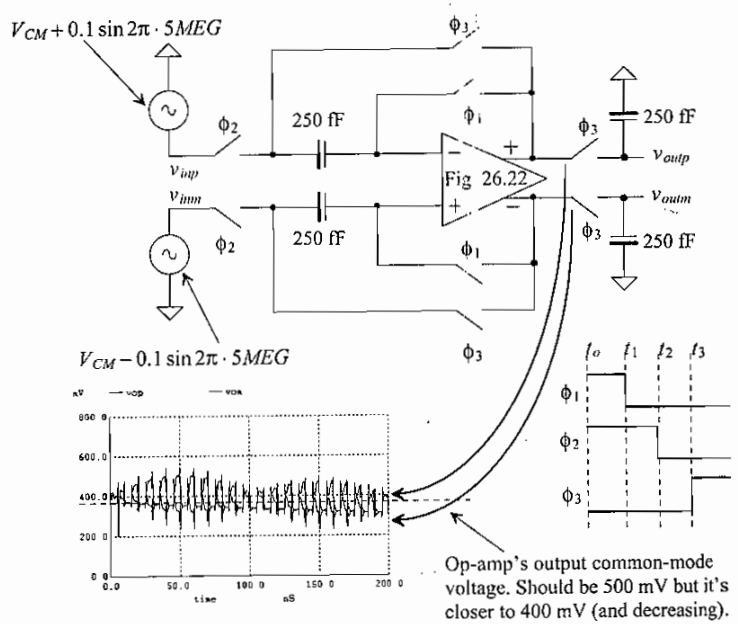


Figure 26.25 A sample-and-hold circuit. Notice how the output common-mode voltage is wandering.

Adding CMFB

Figure 26.26 shows how we can modify the basic op-amp to allow for a CMFB signal input. If the outputs of the op-amp (their average or common-mode voltage) are too high,

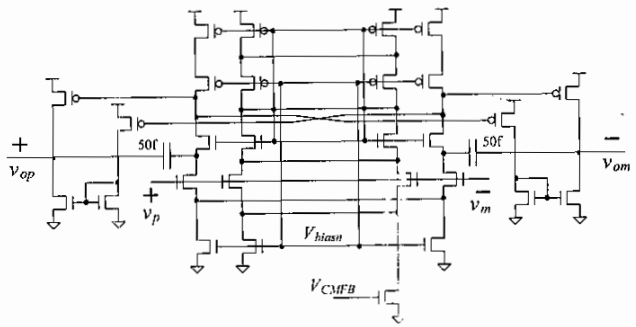


Figure 26.26 Modifying the op-amp for a CMFB input signal.

V_{CMFB} goes up. This causes the output voltage of the diff-amp to go up (increasing V_{CMFB} causes the bias current in the center MOSFETs of the diff-pair to increase). An increase in the diff-amp's output voltage lowers the quiescent current flowing in the output buffers, causing the output voltage to move downwards. This is a good time to remember one of the fundamentals from Ch. 20, namely, if two MOSFET gate-source voltages are equal and they have the same drain current, then their drain-source voltages must be equal. For the output buffer in Fig. 26.26, this means that as we reduce the drain currents flowing in the output buffer (by driving V_{CMFB} high), the gate-source voltages of the NMOS devices decrease. Because of the fundamental concept just mentioned, this causes the output voltages of the op-amp to decrease. Figure 26.27 shows the op-amp's output voltage change with V_{CMFB}. For stability concerns, it's of interest to determine the gain from the CMFB input to the outputs. From the simulation, the gain is approximately 25 (about 10 times less than the differential gain). The lower forward CMFB gain allows us to use a diff-amp with a current mirror load for the CMFB amplifier (discussed next).

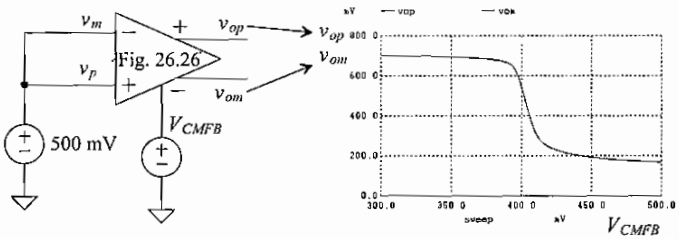


Figure 26.27 The CMFB input to output relationship. The gain is approximately 25 (considerably less than the forward differential gain).

CMFB Amplifier

Now that we know how our CMFB signal, V_{CMFB}, in the op-amp of Fig. 26.26 affects the output voltages and that the gain this signal sees is approximately 25 ($v_{op,n}/V_{CMFB}$), we need to discuss the CMFB amplifier. Consider the two diff-amps seen in Fig. 26.28. In the top diff-amp, gate-drain-connected loads are used. As seen in the simulations, the gain from the averaged input, V_{CMA}, to the diff-amp's output, V_{CMFB} is considerably less than 1 (and so the CMFB is guaranteed to be stable). Unfortunately, the CMFB amplifier's output voltage isn't high enough. (As seen in Fig. 26.27, we need approximately 400 mV.) The diff-amp's tail current can be increased in size (use at least two PMOS for the tail current) to increase the output voltage. However, the low gain, say around 0.3, combined with the CMFB gain through the op-amp, again around 25, means that the CMFB loop's overall gain is only around 7. This isn't large enough to precisely balance the outputs. Using the diff-amp with current mirror load, Fig. 26.28b, gives a gain of approximately 8. This combined with the op-amp's CMFB gain gives an overall CMFB loop gain of 200. This is less than the op-amp's differential gain (so we can use the same compensation capacitors to stabilize the CMFB loop), Fig. 26.23, and large enough to precisely balance the op-amp's outputs.

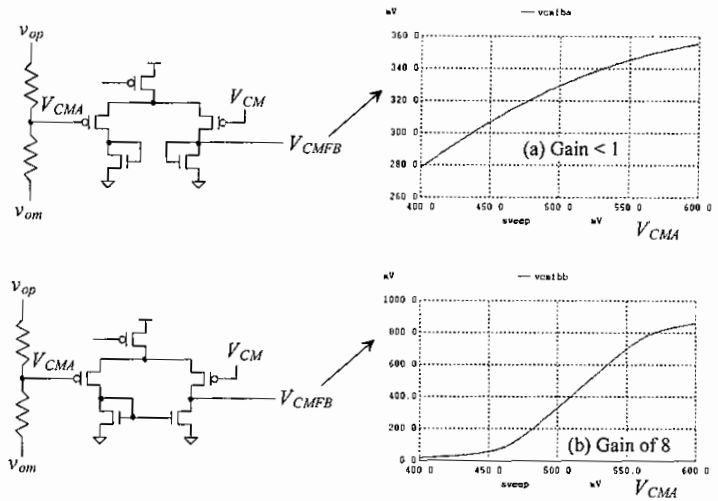


Figure 26.28 Gains of CMFB amplifiers.

The Two-Stage Op-Amp with CMFB

Figure 26.29 shows the complete schematic of the op-amp. Notice that we've doubled the width of the MOSFETs in the output buffer. The added loading from the CMFB averaging resistors will lower the open-loop gain. To compensate for this reduction, the

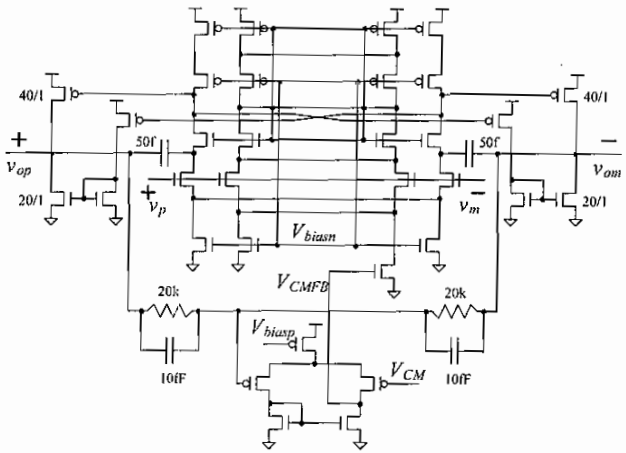


Figure 26.29 Complete schematic of op-amp with CMFB.

drive strength of the MOSFETs on the output of the op-amp was increased. Figure 26.30 shows the DC behavior of the op-amp when placed in the configuration seen in Fig. 26.23. Notice that the outputs cross at the ideal common-mode voltage of 500 mV. The gain is reduced because of the loading by the 20k resistors in the CMFB circuit (and so the gain can be increased by increasing the values of the resistors in the CMFB circuit). However, **we see a problem** with these simulation results. The outputs are only swinging from 200 to 800 mV (not from 0 to *VDD* as possible with a push-pull output stage). Further, we can estimate the current pulled from *VDD* for the op-amp in Fig. 26.29 by counting the number of branches in the op-amp (noting that the output stage counts twice because we've doubled the widths of these devices). Including the CMFB amplifier, there are 11 branches. If the bias current through each branch is 20 µA and the current pulled by the bias circuit is 50 µA, then we would expect the op-amp to pull 270 µA (perhaps even a little less because many of the MOSFETs are biased near the triode region). When we look at the simulation results that generated Fig. 26.30, we see that the current is closer to 500 µA (way off indicating, again, that we've got a problem).

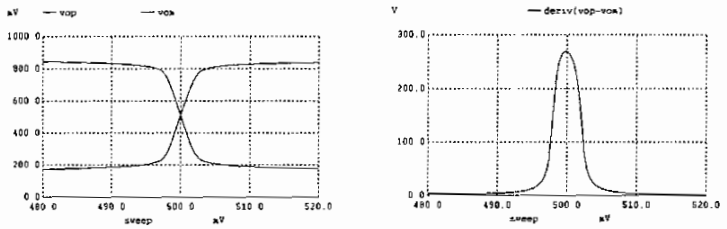


Figure 26.30 Simulating the operation of the op-amp in Fig. 26.29.

Origin of the Problem

The origin of the problem (that is, the op-amp drawing too much current and the output swing not reaching the rails) can be traced to the output buffer. We spent a considerable amount of time discussing how we want to set the outputs of the diff-amp to V_{biasp}. When we look at the simulations, we see the diff-amp's outputs are considerably less than the ideal 600 mV of V_{biasp}. To understand why, let's look at the output buffer in Fig. 26.31. Since the problems appear with the addition of the CMFB circuit, we assume that the two gates of the PMOS devices are moving at the same potential (tied together), that is, that the inputs to the buffer are moving with the diff-amp's output common-mode level. When the output of the diff-amp is V_{biasp} (600 mV), 20 µA flows in all of the MOSFETs. (To keep things simpler, we don't include the doubling in the widths used in the op-amp output devices.) The gate potentials of the NMOS are, roughly, V_{biasn} (400 mV). Because of the symmetry of the circuit, this means that the output is also at 400 mV. To increase the output voltage to 500 mV, we must drop the potential on the gates of the PMOS until the gate-drain-connected NMOS has a V_{GS} of 500 mV (again because of the symmetry). This increases the current (significantly) flowing in the output buffer, lowers its gain, and reduces the linear output swing. The overdrive voltages essentially change from the desired 100 mV when 20 µA of current flows to 200 mV when the output is driven to 500 mV.

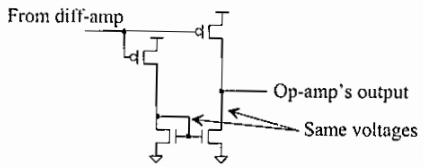


Figure 26.31 Output buffer used in the op-amp of Fig. 26.29.

Figure 26.32 shows one solution to this problem. We've added a device to cascode the output buffer's NMOS (the one connected to the output terminal). The speed shouldn't be affected by the addition of the device (which operates near or in the triode region). The added device allows the op-amp's output to swing more freely (but the current still won't be precisely set). The added device won't affect the output swing range of the op-amp. Figure 26.33 shows the op-amp with modified output buffer.

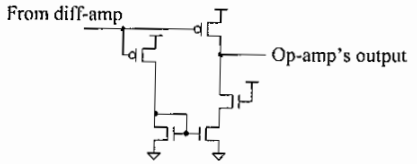


Figure 26.32 Adding a device to allow the output voltage to swing.

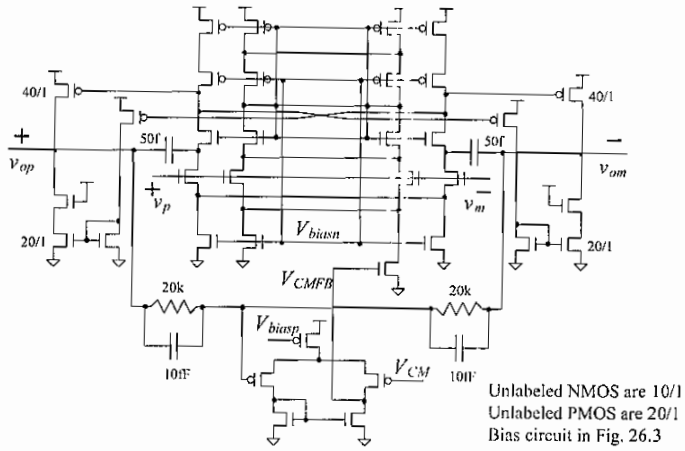


Figure 26.33 Op-amp with modified output buffer.

Simulation Results

Figures 26.34 to 26.36 show simulation results based on the circuit topologies in Figs. 26.23 to 26.25, respectively, using the op-amp in Fig. 26.33. In Fig. 26.34 we see that the outputs now swing close to the power supply rails (unlike what we saw in Fig. 26.30). Also, the gain is higher. Figure 26.35 shows the step response of the amplifier in the topology seen in Fig. 26.24. Finally, Fig. 26.36 shows the outputs of the sample-and-hold in Fig. 26.25 using the op-amp in Fig. 26.33.

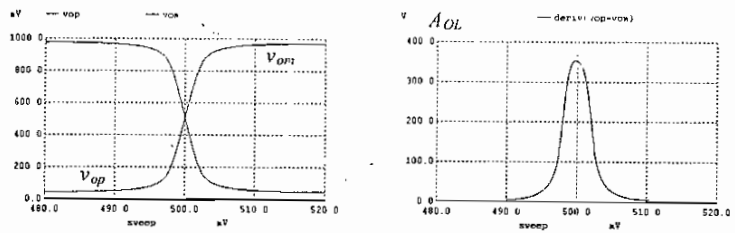


Figure 26.34 Resimulating the op-amp in Fig. 26.33 in the configuration seen in Fig. 26.23.

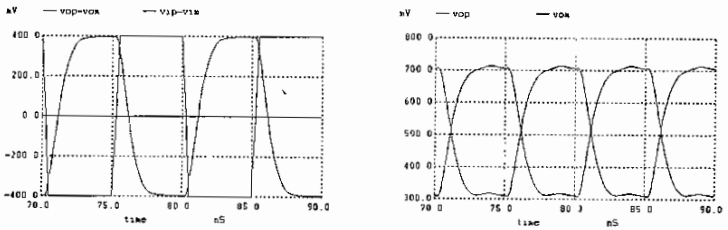


Figure 26.35 Regenerating the simulation results using the topology in Fig. 26.24 with the op-amp in Fig. 26.33.

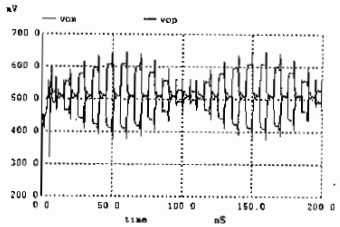


Figure 26.36 Using the op-amp in Fig. 26.33 in the sample-and-hold circuit of Fig. 26.25. Figure shows the outputs of the op-amp.

Using MOSFETs Operating in the Triode Region

For the output buffer in Fig. 26.32, we added a transistor in the drain portion of the circuit. We might wonder if we can accomplish better control of the current flowing in the output buffer by adding circuitry to the source side of the buffer. Towards answering this question, consider the portion of the output buffer seen in Fig. 26.37. If both outputs are at V_{CM}, the gate-source voltages of M2 and M3 are the same and so are the gate-source voltages of M1 and M4. If one output goes high and the other output goes low (and the outputs are centered around V_{CM}), then the net drain current of each M3 is constant. If one output goes below the threshold voltage of an NMOS device, then the balancing stops working. The practical problem of using triode-operating MOSFETs (in a CMFB circuit or any type of amplifying configuration) is that the gain through a MOSFET operating in the triode region is low (so it's difficult to provide control).

A portion of the output buffer.

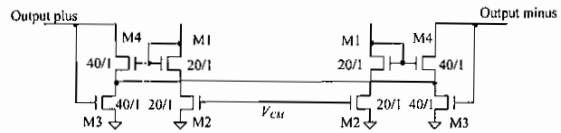


Figure 26.37 Using triode-operating MOSFETs to balance the outputs (bad).

Start-up Problems

Examine the circuit in Fig. 26.38 using the op-amp in Fig. 26.33. Because the op-amp's inputs are at 0 V, the diff-amps on the input of the op-amp are off. This causes the gates of the PMOS devices in the output buffer to be pulled to *VDD*. The circuit remains in this state and doesn't move the outputs or the inputs up to V_{CM}. To avoid this start-up problem, we need to ensure that there is some DC path to *VDD* or V_{CM} to "start-up" the op-amp.

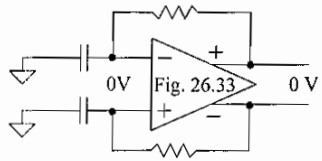


Figure 26.38 The op-amp in Fig. 26.33 won't turn on in this topology.

Lowering Input Capacitance

Consider connecting the gates of the NMOS diff-amp used for biasing to V_{CM} (see the bold line in Fig. 26.39). This reduces the input capacitance of the op-amp and shouldn't affect the normal operation as long as the common-mode voltage of the op-amp (the gates

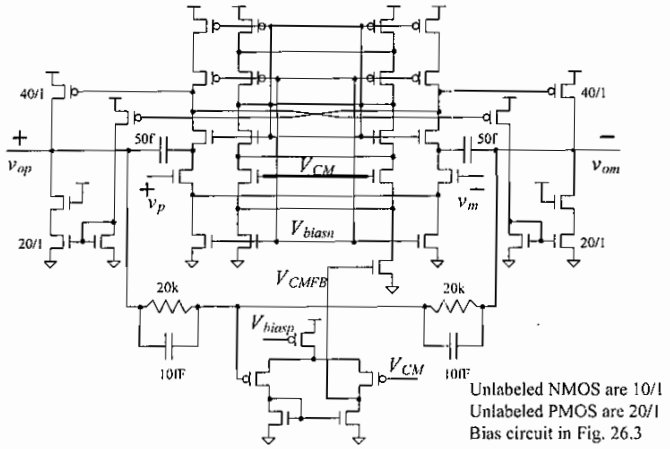


Figure 26.39 Connecting the bias circuit diff-amp's inputs to the common-mode voltage.

of the other diff-amp) is V_{CM}. Simulating the operation of the op-amp in Fig. 26.39 to generate the data in Figs. 26.34 to 26.36 (in the same topologies), we see no change in the simulation data (the netlists used to verify this statement can be found at cmosedu.com).

Looking at Fig. 26.39, we might now wonder why we need two biasing branches down the middle of the op-amp. Further, looking at the voltages in Fig. 26.19, we might wonder if we can make the op-amp more tolerant to changes in *VDD*. The way the op-amp is biased now the open-loop gain drops significantly if *VDD* drops to 900 mV.

Making the Op-Amp More Practical

As just mentioned, having two identical branches down the middle of our op-amp wastes power (although it is useful for a symmetrical layout). We can cut one of the branches out of the design and reduce the power dissipated by the op-amp. We might further wonder if using V_{biasn} for biasing the cascode PMOS current sources is such a good idea. Reviewing Fig. 26.4, we see that V_{biasn} essentially stays at 400 mV after the reference turns on (*VDD* gets above a certain value). In a practical op-amp, we want this voltage to decrease as *VDD* drops in an effort to keep the PMOS devices operating in the saturation region.

Examine the op-amp in Fig. 26.40. We've added a wide-swing bias circuit, see Fig. 20.38, and cut out one of the biasing branches used in the op-amp seen in Fig. 26.39. The power dissipation remains essentially the same as in the previous op-amp topologies. Note how we apply the CMFB to one side of the bias circuit (not to the 10/3 wide-swing bias branch). Further notice how we've reduced the gain of the CMFB loop by using two 10/2 MOSFETs. It may be a good idea to reduce the strength of the CMFB loop even further by increasing the lengths of the NMOS devices. For example, we might change these devices from 10/2 to 10/4. The ability to drive V_{CMFB} considerably above V_{biasn} eliminates the concern that the NMOS device can sink the current needed to bias the circuit at the correct point (and why the CMFB loop can so easily become unstable).

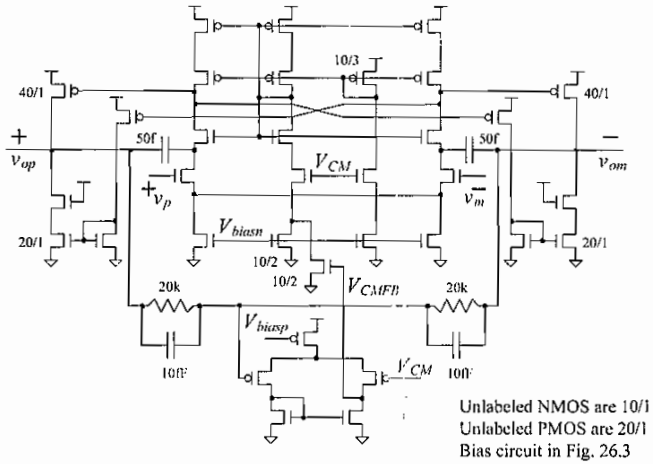


Figure 26.40 Making the op-amp more practical.

A more practical problem is the change in the MOSFET's drain currents with drain-source voltages. For example, looking at Figs. 26.18a and b, we see that it's possible for the NMOS's drain current to be twice the PMOS's drain current for the same gate-source voltages. This can cause op-amp failure in the CMFB circuit. If, for example, the current flowing in the 10/2 MOSFET connected to V_{biasn} is larger than the current sourced by the PMOS, the voltage V_{CMFB} goes to zero and the CMFB loop doesn't work properly. A good "rule-of-thumb" is for the fixed current flowing in the CMFB-controlled bias circuit to be 25–50% of the total expected current.

Increasing the Op-Amp's Open-Loop Gain

The op-amp that we've developed in this section has an open-loop gain in the hundreds. As we'll see in Ch. 29 (Eq. [29.59]), the open-loop gain of the op-amps used in a data converter has a direct effect on the maximum attainable resolution. Towards increasing the gain of the op-amp, let's use the gain-enhancement (GE) techniques presented in Sec. 24.4.

Figure 26.41 shows how we should *not* implement GE. An amplifier with fully-differential outputs, like the amplifier in Fig. 26.7, regulates the drains of the top PMOS devices. We know that this is bad because we would need a CMFB circuit to balance the outputs of the added amplifier.

Figure 26.42 shows how we can add GE to the op-amp developed in this section. Notice that GE is implemented with amplifiers having single-ended outputs. When designing the added amplifiers, as discussed in Sec. 24.4, the bandwidth isn't important for high-speed operation. The lengths of the MOSFET in the added amplifiers can be increased to reduce power and boost gain. An important concern is the added amplifier's input common-mode range.

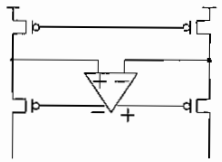


Figure 26.41 How not to implement GE in an op-amp.

The practical problem with the topology seen in Fig. 26.42 is the implementation of the CMFB. With the GE added to the op-amp we now have four additional feedback loops. Variations in V_{CMFB} affect all GE loops in addition to the CMFB through the op-amp. Making these loops stable becomes extremely challenging. What we need is to implement the CMFB without including the diff-amp and GE amplifiers. We can only do this by controlling the output buffer common-mode level, Fig. 26.43. This circuit includes the output buffer we used in Fig. 26.37. Now, however, we add an amplifier in series with the triode-operating MOSFETs to boost the CMFB gain (so that the outputs can be balanced around V_{CM}). The problem with this approach is that the CMFB isn't compensated using the same capacitors as the differential forward signal path (i.e., no Miller effect). We must be concerned with CMFB stability.

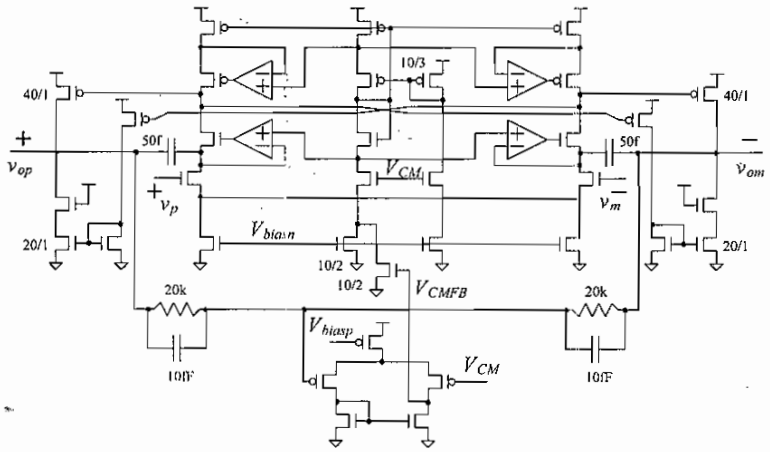


Figure 26.42 Adding gain-enhancement to the op-amp.

Let's discuss the op-amp design in Fig. 26.43. We begin by showing the DC behavior of the op-amp (to show that the outputs are indeed balanced). Figures 26.44a and (b) show the DC behavior and gain of the op-amp in Fig. 26.43 in the topology seen in Fig. 26.24. Again note that by increasing the values of the 20k resistors used to

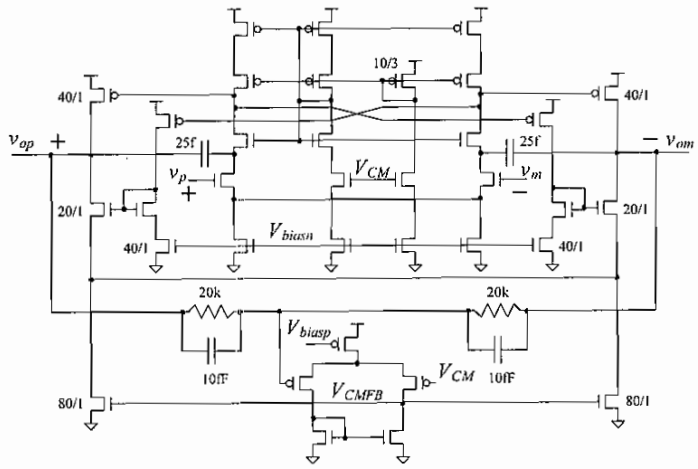
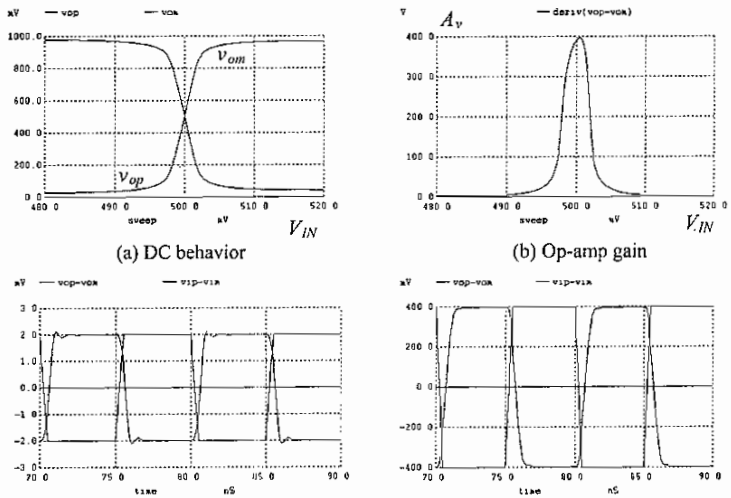


Figure 26.43 Providing CMFB through just the output buffer. Using an amplifier with triode-operating MOSFETs for CMFB (good).



(a) DC behavior

(b) Op-amp gain

(c) Small-signal step response

(d) Large-signal step response

Figure 26.44 Behavior of the op-amp in Fig. 26.43 (see text).

average the outputs we can increase the DC gain of the op-amp. Because the forward differential path compensation (now) doesn't include the CMFB path, we've reduced the compensation capacitors from 50 fF to 25 fF (there are practical issues with this change that we'll discuss next). Note that the widths of the added triode-operating MOSFETs are increased to ensure that they operate well within the triode region and don't significantly affect the output drive capability of the op-amp. Figures 26.44c and (d) show the small- and large-signal step responses (see Fig. 26.24). The settling times are approximately 1–2 ns. The current drawn from *VDD* (including the the bias circuit current) is approximately 250 μA (200 μA for the op-amp alone).

Offsets

We said in Sec. 26.2 that we want to add offsets of 50 mV in series with the gates of our MOSFETs to see if the op-amp "breaks." Consider the addition of an input-referred offset voltage in the schematic seen in Fig. 26.45. This offset is used to model the overall offset voltage of the op-amp (see Fig. 24.4). If we look at the effects of the offset on the circuitry in Fig. 26.43, we see that one of the NMOS devices in the diff-pair will have a higher overdrive voltage (larger g_m) than the other NMOS device. Since the unity-gain frequency of an op-amp is given by $g_m/2\pi C_c$, the effect is a shift in the unity-gain frequency (and potential instability). As seen in Fig. 26.45, using the 25 fF compensation capacitors destabilizes the op-amp (as indicated by the ringing). Increasing the compensation capacitor's value (back) to 50 fF makes the step response cleaner. We could argue that 50 mV is an unrealistically high offset voltage, so it's better to leave the capacitors at 25 fF (and this may be the case). However, in a practical CMOS process, the op-amp's characteristics shift with the process shifts (and temperature). It's better to overcompensate than to have an unstable op-amp.

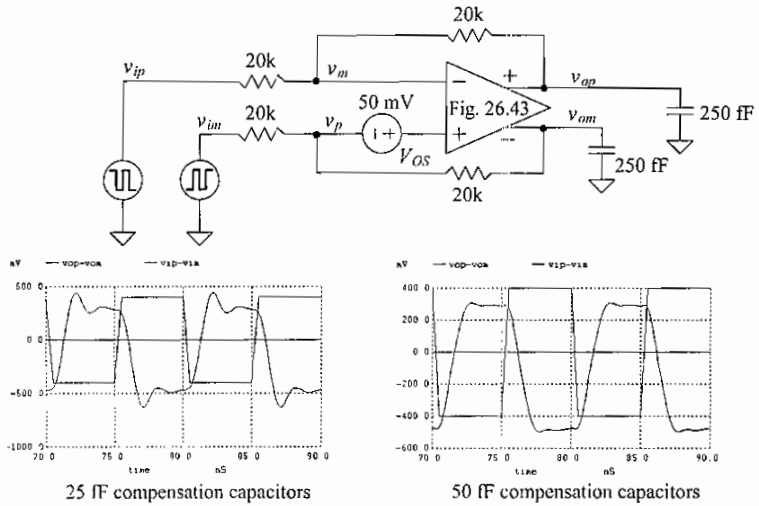


Figure 26.45 How an offset can affect the step response (compensation).

Op-Amp Offset Effects on Outputs

Notice, in Fig. 26.45, that an offset shifts the differential output signal by twice the offset voltage. The individual op-amp output voltages remain centered around V_{CM}. To describe this in more detail, consider the test setup seen in Fig. 26.46. We know the CMFB forces

$$\frac{v_{op}+v_{om}}{2}=V_{CM} \text{ or } v_{op}=2V_{CM}-v_{om} \tag{26.8}$$

Further, assuming large op-amp open-loop gain,

$$v_p+V_{OS}=v_{pos}\approx v_m \tag{26.9}$$

Equating currents, we can write

$$\frac{V_{CMI}-v_p}{R_{in}}=\frac{v_p-v_{om}}{R_f} \text{ or } \frac{V_{CMI}-v_m+V_{OS}}{R_{in}}=\frac{v_m-V_{OS}-v_{om}}{R_f} \tag{26.10}$$

and

$$\frac{V_{CMI}-v_m}{R_{in}}=\frac{v_m-v_{op}}{R_f} \tag{26.11}$$

Subtracting Eq. (26.11) from Eq. (26.10), we get

$$\frac{V_{OS}}{R_{in}}=\frac{-V_{OS}+v_{op}-v_{om}}{R_f} \tag{26.12}$$

or

$$v_{op}-v_{om}=V_{OS}\cdot\left(\frac{R_f}{R_{in}}+1\right) \tag{26.13}$$

The voltages on the outputs of the op-amp are then

$$v_{op}=\frac{V_{OS}}{2}\cdot\left(1+\frac{R_f}{R_{in}}\right)+V_{CM} \text{ and } v_{om}=V_{CM}-\frac{V_{OS}}{2}\cdot\left(1+\frac{R_f}{R_{in}}\right) \tag{26.14}$$

The op-amp's input voltages are

$$v_m\approx v_p+V_{OS}=\frac{V_{OS}}{2}+V_{CM}\cdot\left(\frac{R_{in}}{R_{in}+R_f}\right)+V_{CMI}\cdot\left(\frac{R_f}{R_{in}+R_f}\right) \tag{26.15}$$

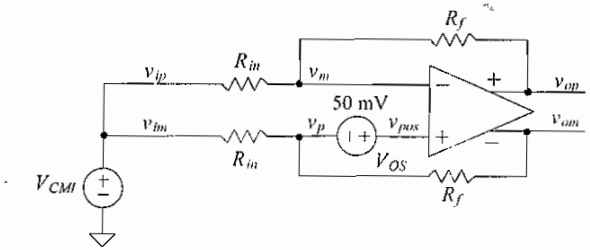


Figure 26.46 How an offset voltage causes an imbalance in the outputs.

If the input signal's common-mode voltage, V_{CMI}, is the same as the op-amp's common-voltage, V_{CM}, and V_{OS} is zero, then both op-amp inputs are held at V_{CM}. If $V_{CM} = V_{CMI}$ but the offset isn't zero, then $v_m = V_{OS}/2 + V_{CM}$ and $v_p = V_{CM} - V_{OS}/2$. Neglecting the offset voltage, if the input signal's common mode voltage isn't V_{CM}, then the op-amp's input common-mode voltage ($v_p = v_m$) will be different from the ideal value of V_{CM}. This can shut the op-amp off and cause undesirable behavior.

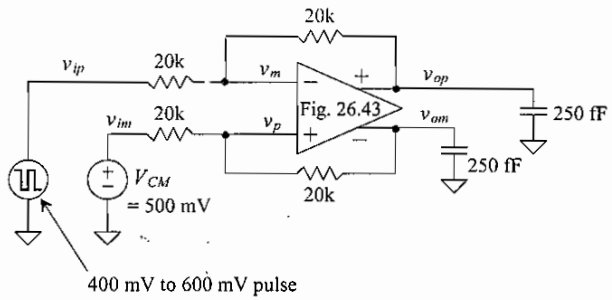


Figure 26.47 Problems with single-ended input signals.

Single-Ended to Differential Conversion

Even if the input signals are referenced around V_{CM} and the op-amp is offset free, we can still have problems. Consider the circuit in Fig. 26.47. In this circuit the input signal is single-ended. The other input to the op-amp is tied to V_{CM}. Even though the input is balanced around 500 mV, it is not a truly differential signal. When the input signal is up at 600 mV, the effective input common-mode voltage, V_{CMI}, is 550 mV (the average of the two inputs). When the input signal is at 400 mV, V_{CMI} is 450 mV. Figure 26.48 shows the simulation results using the circuit in Fig. 26.47. These results are very interesting. Consider what's happening between 70 and 75 ns. During this time, v_{ip} is 600 mV and v_{im} is 500 mV (and so $v_{ip} - v_{im} = 100\ mV$). The op-amp's inputs move above the ideal 500 mV, as seen in the figure. This has the effect of increasing the transconductance of the diff-amp (the diff-amp's tail current is operating near/in the triode region so the tail

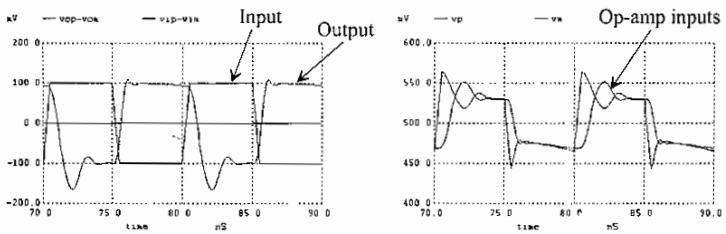


Figure 26.48 Simulating the operation of the circuit in Fig. 26.47.

current increases as the input common-mode voltage goes up). The increase in the diff-amp's g_m causes the unity gain frequency to increase and, with a fixed compensation capacitor, the phase margin to decrease (causing overshoot and ringing in the circuit's step response). Next consider what happens between 75 and 80 ns. The input signal switches to $400\ mV - 500\ mV = -100\ mV$ and the op-amp's inputs drop down to around 475 mV. This causes the diff-amp's g_m to decrease and the op-amp to slow down (become more stable). However, at such a low input common-mode voltage the gain of the op-amp will drop and the outputs may wander.

CMFB Settling Time

If we look at the voltage, V_{CMFB}, in the op-amps in Fig. 26.40 or 26.43, we may see that they aren't settling as quickly as the differential mode signals. Is this bad? As seen in Eq. (26.13), variations in V_{CMI} don't (ideally) affect the differential output signal. However, if the variations in V_{CMFB} and thus V_{CMI} are large, one of the output signals can saturate at close to *VDD* or ground (the op-amp's gain drops), affecting the differential signal path.

CMFB in the Output Buffer (Fig. 26.43) or the Diff-Amp (Fig. 26.40)?

In this section we've presented the design of op-amps using continuous-time CMFB. Various design trade-offs and topologies were presented. At this point a good question is: "Which CMFB scheme is better?" Controlling the output common-mode level through the output buffer (Fig. 26.43) is simple, easy to ensure stability, and fairly robust. However, consider what happens if the op-amp in Fig. 26.43 is used in the topology in Fig. 26.46 (without an offset). If V_{CMI} is ground, then, according to Eq. (26.15), the input voltages to the op-amp will move to 250 mV while the outputs of the op-amp remain at 500 mV. Thinking about this for a moment and neglecting the fact that an op-amp input common-mode voltage of 250 mV will shut the op-amp off, we see that the outputs of the op-amp must source a current back through the feedback resistors to the inputs. Further, anytime $V_{CMI} < V_{CM}$, the PMOS in the output buffer must source a DC current back to the inputs. Similarly, if $V_{CMI} > V_{CM}$, the NMOS in the output buffer must sink a current from the inputs of the op-amp. When using CMFB in the output buffers, we have no way of increasing the quiescent or DC current flowing in the output buffer to source/sink current from the input source (or to a DC load connected to ground). The output voltage of the diff-amp in the op-amp of Fig. 26.43 biases, or sets, the quiescent current in the output buffer. The CMFB scheme used in the output buffer simply adjusts the drive strength of the NMOS (in the output buffer) to set the op-amp's output common-mode level.

Using the CMFB scheme in Fig. 26.40, we can adjust the output voltage of the diff-amp (and ultimately the op-amp's output voltage) and thus the bias current in the output buffer (both the NMOS and the PMOS connected to the output buffer). However, if the output buffer must source/sink a significant amount of current, we can have problems with this topology too. For example, if the diff-amp's output voltage increases (turning off the PMOS in the output buffer and attempting to pull the op-amp's outputs down) while the NMOS devices connected to the outputs are sinking significant current, then it's possible that the outputs will get pulled upwards and the CMFB will fail (likely causing the gain of the op-amp to drop and the outputs to have limited swing).

To ensure the most robust op-amp design (tolerant to offsets and easy to compensate the CMFB loops), CMFB circuits can be placed around both op-amp stages. This method is used in our last op-amp design example discussed next.

26.4 Op-Amp Design Using Switched-Capacitor CMFB

In this section we turn our attention towards op-amp designs for switched-capacitor (SC) circuits. We'll develop an op-amp design based on the topologies discussed in the last section. In this section, however, we'll use SC CMFB instead of continuous-time CMFB. Again, we'll use the nanometer CMOS process with minimum lengths and (roughly) 100 mV overdrive voltages (for drain currents of 20 μA, see Fig. 26.18). We'll stick with two-stage designs because of the low open-circuit gains present when doing high-speed design.

Clock Signals

The SPICE listing for the clock signals used in the simulations in this section is seen below. Note that the 100 MHz phi1 and phi2 clocks are used with the NMOS transistors (phi1 and phi2 are not high at the same times), while the complements of these clock signals are used with the PMOS switches (not low at the same times).

```
*Clock Signals
Vphi1   phi1    0 DC 0 Pulse   0 1 0  200p  200p 4n 10n
Vphi1b  phi1b   0 DC 0 Pulse   1 0 0  200p  200p 4n 10n
Vphi2   phi2    0 DC 0 Pulse   0 1 5n 200p 200p 4n 10n
Vphi2b  phi2b   0 DC 0 Pulse   1 0 5n 200p 200p 4n 10n
R1      phi1    0       1MEG
R2      phi1b   0       1MEG
R3      phi2    0       1MEG
R4      phi2b   0       1MEG
```

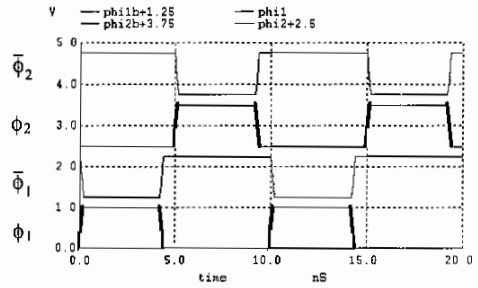


Figure 26.49 Generating nonoverlapping clocks for SC circuits.

Switched-Capacitor CMFB

Figure 26.50 shows the SC CMFB circuit from Fig. 26.16 along with a symbolic representation. We've selected 25 fF capacitors for the high-speed averaging capacitors. The kT/C noise associated with these capacitors is a common-mode signal and so it shouldn't affect the differential-mode signal. We don't want these capacitors to be too large because they will load the output of the amplifier. We made the switched-capacitors associated with the differencing and averaging 10 fF so that the changes in V_{CMFB} during one clock cycle won't be too large due to differences between the ideal V_{CMFB} and the actual V_{CMFB}. When the average of the outputs, v_{op} and v_{om}, is above their ideal value, the

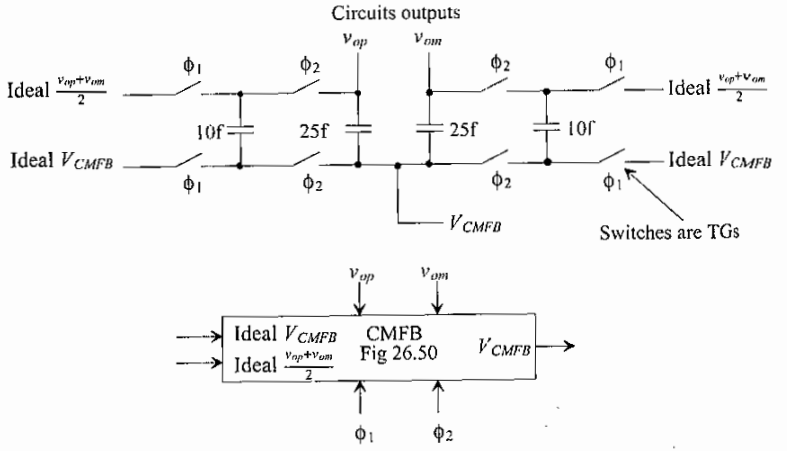


Figure 26.50 A switched-capacitor CMFB circuit (see Fig. 26.16). Switches implemented with transmission gates.

CMFB circuit's outputs moves upwards. It's important that this causes the average value of the outputs to move downwards. In other words, we've got to ensure that negative feedback is used in the CMFB loop.

Figure 26.51 shows some simulation results using the SC CMFB in Fig. 26.50. The ideal V_{CMFB} is 400 mV. The ideal average output, $\frac{v_{op}+v_{om}}{2}$, is 500 mV (V_{CM}). Prior to 150 ns, both v_{op} and v_{om} are 500 mV. At 150 ns, these two voltages jump (in the simulation) to 700 mV (they jump 200 mV away from the ideal average output of 500 mV). The voltage, V_{CMFB}, also jumps at this time. The key point to notice is that V_{CMFB} moves in the same direction as do changes in the output common-mode voltage. Knowing this is important when we are ensuring that our CMFB loop employs negative feedback.

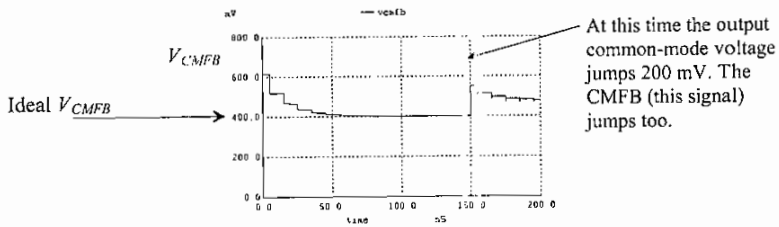


Figure 26.51 How the common-mode feedback signal increases to pull the outputs down.

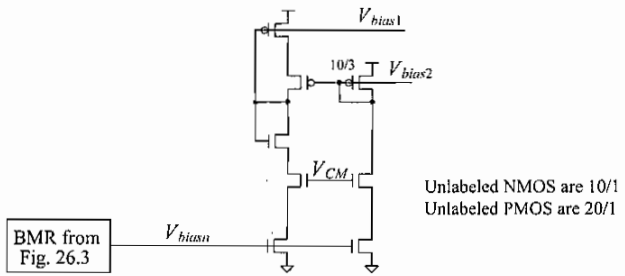


Figure 26.52 Biasing circuit for the op-amp developed in this section.

The Op-Amp's First Stage

As just mentioned, the output of the SC CMFB circuit, V_{CMFB}, moves in the same direction as movement in the average of the amplifier's outputs, v_{op} and v_{om} (the output common-mode level). Keeping this in mind, consider the bias circuit and diff-amp schematics seen in Figs. 26.52 and 26.53. We've separated the bias circuit out from the diff-amp, see Fig. 26.43, for a couple of reasons. To begin, an increase in the CMFB signal in Fig. 26.43 caused the outputs of the diff-amp to increase. As just mentioned, we want the diff-amp's outputs to move in the opposite direction of V_{CMFB}. Moving the CMFB-controlled MOSFET into the tail current of the diff-amp provides this control. Next, our bias circuit may now be shared with several op-amps. The benefit of sharing the bias circuit is a reduction in power dissipation.

Before adding the CMFB circuit, let's look at a SC circuit using an op-amp. Reviewing the sample and hold in Fig. 26.25, we see that when the ϕ_1 switches are closed, the op-amp's inputs and outputs are at (ideally) V_{CM}. The op-amp, during this

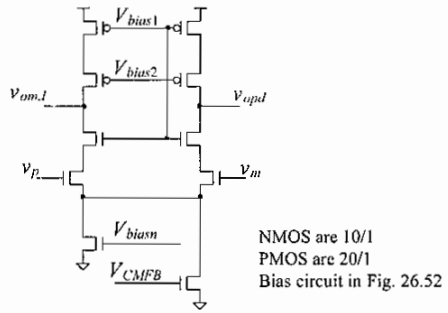


Figure 26.53 Diff-amp used with the bias circuit of Fig. 26.52.

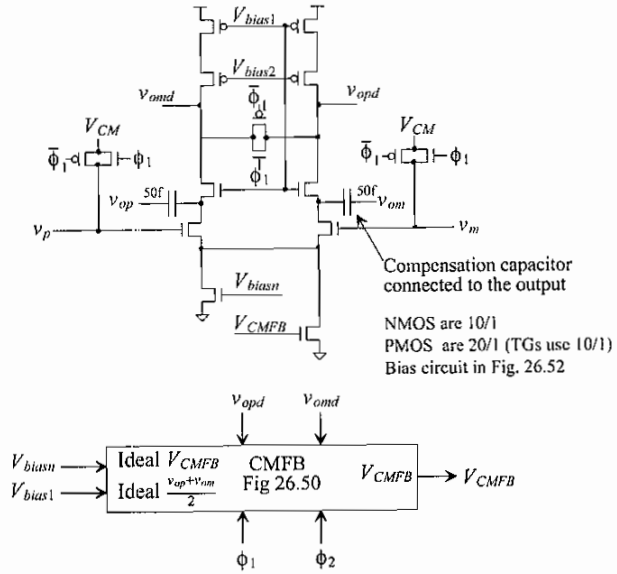


Figure 26.54 First-stage diff-amp with SC CMFB.

time, is in the unity-follower configuration. Instead of placing the op-amp in the follower configuration during this time, let's: 1) short the inputs to op-amp (Fig. 26.54) to V_{CM}, 2) short the outputs of the diff-amp (the inputs to the output buffer) together, 3) use a SC CMFB circuit around the diff-amp to ensure a balance condition (and to set the diff-amp's output voltage to V_{bias1}), 4) short the outputs of the op-amp together (so now both the output buffers inputs are shorted and outputs are shorted), and 5) use a SC CMFB around the output buffer to ensure balanced outputs. In other words, during ϕ_1, the differential inputs are shorted together. At the same time, the outputs of the op-amp are shorted together.

Figure 26.55 shows the simulated output of the diff-amp in Fig. 26.54. The SC CMFB circuit sets the outputs of the diff-amp at V_{bias1} or roughly 600 mV. By shorting the two diff-amp outputs together and connecting the inputs to V_{CM}, we are ensuring that even with horrible mismatch, the outputs of the diff-amp will be equal and can be used to bias the output buffer. If, for example, the diff-pair shows a 50 mV mismatch (V_{OS}) then, when ϕ_1 is high, the drain currents in each side of diff-amp are set differently (making this scheme very tolerant to offsets). When ϕ_1 goes low and the op-amp is placed in a feedback configuration, each of the diff-amp's inputs will move (in opposite directions) by $V_{OS}/2$.

Note that an important concern, as discussed in the last section, is the input common-mode voltage of the diff-amp. If, for example, the input common-mode voltage

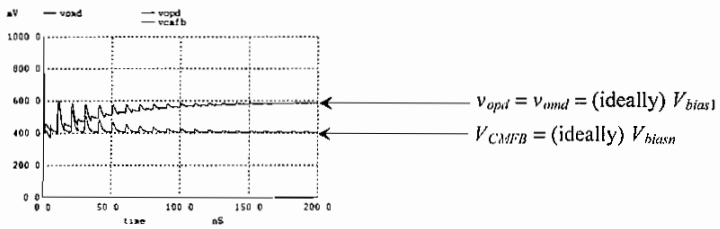


Figure 26.55 Simulating the operation of the circuit in Fig. 26.54.

drops from 500 mV, in Fig. 26.54, to 400 mV the NMOS devices will start shutting off. The op-amp will not settle or behave properly.

The Output Buffer

Figure 26.56 shows a schematic of the output buffer and SC CMFB stage. The nodes v_{omd} and v_{opd} are connected to the diff-amp outputs in Fig. 26.54. When ϕ_1 is high, the outputs of the buffer are shorted together. The SC CMFB is used to set the outputs to the common-mode voltage during this time. Figure 26.57 shows the simulation results for both stages of the op-amp (the op-amp is made with Figs. 26.54 and 26.56) with the inputs and outputs floating. When ϕ_1 is high, the inputs to the op-amp are connected to V_{CM}.

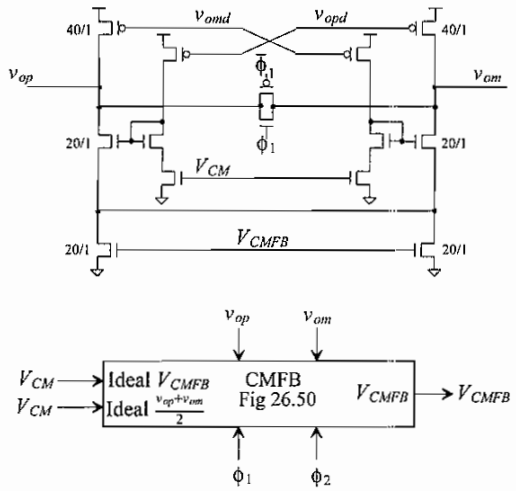


Figure 26.56 Output buffer and CMFB.

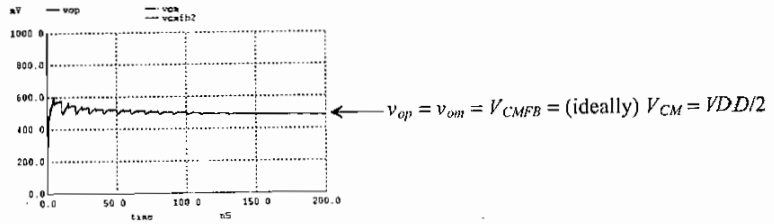


Figure 26.57 Simulating the operation of the op-amp (Figs. 26.54 and 26.56).

Is shorting the outputs of the buffer (or diff-amp) necessary? If the op-amp is operating open-loop (when ϕ_1 goes high), we don't want the outputs to float. By shorting the diff-amp and output buffer output terminals together, we ensure that they are set to a known value. If we were to use this op-amp in a SC integrator, we wouldn't short the inputs of the diff-amp to V_{cm}, the outputs of the diff-amp together, or the output buffer outputs together since there is always feedback around the op-amp. In other words, the same op-amp topology can be used in a SC integrator but without the three TGs seen in Figs. 26.54 and 26.56.

An Application of the Op-Amp

An application of the op-amp we've just developed is seen in Fig. 26.58. This is the sample and hold developed earlier (Figs. 25.15 and 26.25) except that now, during ϕ_1 (high) the inputs to the op-amp are shorted to V_{CM} (Fig. 26.54) and the outputs are shorted together (Fig. 26.56). When ϕ_2 goes high (ϕ_1 is low since the two clocks are nonoverlapping), the op-amp moves into the follower configuration and holds its outputs at the value of the inputs when the ϕ_1 switches shut off

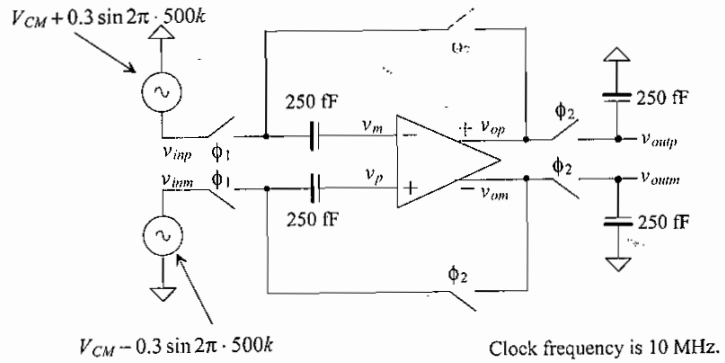
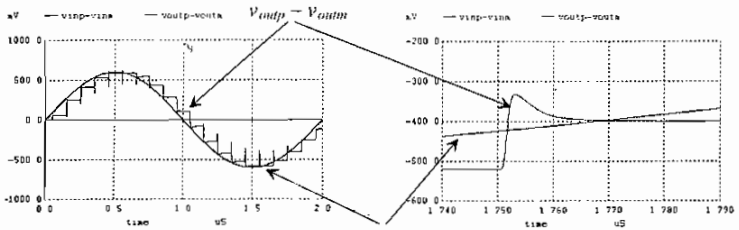
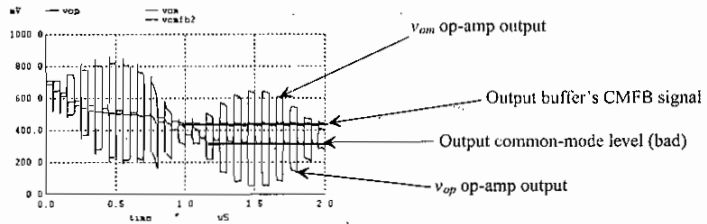


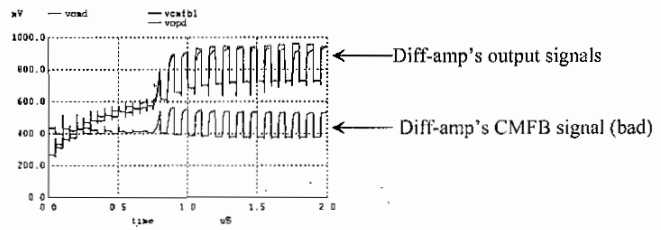
Figure 26.58 Simulating the operation of the op-amp formed with the diff-amp in Fig. 26.54 and buffer in Fig. 26.56.



(a) Circuits inputs and outputs (b) Slow settling time



(c) Op-amp's output and buffer's CMFB signal



(d) Diff-amp's outputs and CMFB signal

Figure 26.59 Simulating the operation of the circuit in Fig. 26.58 with the op-amp made with the diff-amp in Fig. 26.54 and the output buffer in Fig. 26.56.

Simulation Results

The simulation results are seen in Fig. 26.59. To ensure that we see the full settling behavior of the op-amp and any other issues or concerns, we use a slow clock (10 MHz) for the initial simulations. As we gain confidence that our op-amp is settling correctly and stable, we can increase the clock frequency and the input signal frequency to see how the sample and hold performs. The main thing that's different with this design compared to previous designs is that the op-amp isn't operating with feedback all of the time. When the op-amp does get put into a feedback configuration (ϕ_2 goes high), there will be a start-up time (e.g., the inputs of the op-amp will move away from V_{CM} because of the op-amp's offset).

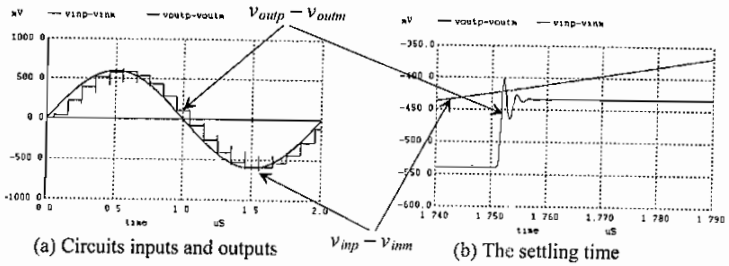
Returning to the simulation results in Fig. 26.59, we see, in (b), that the settling time is quite long (approximately 10 ns). Letting the simulation run for a longer period of time, we see that the settling time increases and, ultimately, that the op-amp stops functioning correctly. In (c) we see the problem: the outputs of the op-amp aren't balanced around V_{CM}. Further, movement downwards causes the common-mode voltage on the input of the diff-amp to drop. The result is that the input diff-amp starts to shut off. The voltage across the diff-amp's tail current source drops and makes it difficult to control the diff-amp's output common-mode level. As seen in (d), the diff-amp's CMFB signal is increasing in an attempt to increase the current flowing in the diff-amp and pull the diff-amp's output voltages downwards. However, because the common-mode voltage on the input of the diff-amp has decreased, the voltage across the tail currents is small (tens of mV). For example, if the input common-mode voltage drops from 500 mV (V_{CM}) to 450 mV and the V_{GS} of the input diff-amp is 400 mV, then only 50 mV is left to drop across the tail current sources (one biased with V_{biasn} and the other with V_{CMFB}). We've discussed this problem earlier.

We can do one of two things to make this decrease in input common-mode voltage less of a problem: 1) reduce the diff-amp's bias current or 2) increase the widths of the input diff-pair. In both cases we need to increase the voltage dropped across the tail current sources by decreasing the V_{GS} of the NMOS diff-pair (the inputs to the diff-amp are, ideally, at V_{CM}). If we decrease the bias current, the g_m of the diff-pair decreases and so does the speed of the op-amp (remembering $f_{un} = g_m/2\pi C_c$). By increasing the width of the diff-pair, we increase g_m and thus the unity-gain frequency of the op-amp. The issue with increasing the width of the diff-pair is that the overdrive voltage for these MOSFETs decreases and thus so does their f_T. If we keep the increase in the width to a modest level, the parasitic pole associated with these diff-pair MOSFETs shouldn't, in any significant way, affect the stability of the op-amp.

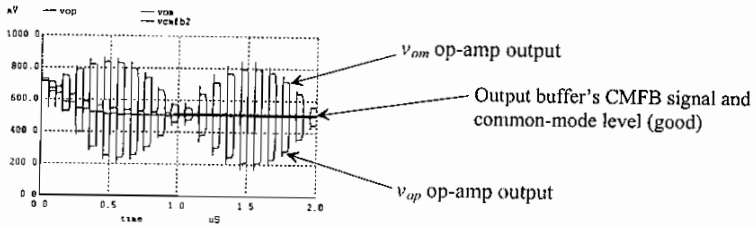
Increasing the Widths of the Diff-Pair

Figure 26.60 shows the operation of the op-amp made with the diff-amp in Fig. 26.54 and the output buffer in Fig. 26.56 if we increase the widths of the diff-pair from 10 to 30. We also change the MOSFETs in the biasing circuit Fig. 26.52 with gates to V_{CM} to widths of 30 to maintain symmetry (important for reducing the offset voltage). The settling time, as seen in (b), is approximately 4 ns. In (c) we see the outputs are swinging around the correct value (V_{CM}). In (d) the outputs of the diff-amp are swinging around V_{bias1} and the common-mode feedback voltage is relatively stable.

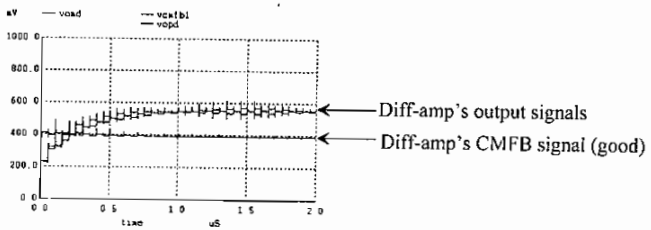
Reviewing the op-amp designs in the last section, we see that their performance may also be improved by a small increase in the widths of the diff-pair (with an increase in the compensation capacitors as needed). However, as just mentioned, we have to be careful to maintain symmetry. For example, if we take the op-amp in Fig. 26.43, increase the widths of the first stage diff-pair from 10 to 30, and simulate, we see that the outputs don't swing all the way up (or down) to the correct values. Further investigation reveals that the diff-pair moved into the triode region. To move the MOSFETs back into the saturation region, we must also increase the widths of the MOSFETs in the bias circuit (with gates at V_{CM}) from 10 to 30. This adjusts the bias voltages for the NMOS/PMOS diff-amp loads. By providing the CMFB back through the diff-amp (instead of through the output buffer), we can adjust the diff-amp's output voltage. Thus the op-amp is more tolerant to issues in the bias circuit.



(a) Circuits inputs and outputs (b) The settling time



(c) Op-amp's output and buffer's CMFB signal



(d) Diff-amp's outputs and CMFB signal

Figure 26.60 Regenerating the data in Fig. 26.59 after increasing the width of the input diff-pair to 30 (from 10).

A Final Note Concerning Biasing

In (most) of the designs presented in this chapter, we biased the gates of the NMOS cascode devices in the diff-amp's load at the same (ideally) voltage as the diff-amp's output. For example, in Fig. 26.53, the gate of the NMOS device is tied to V_{bias1}, while its drain (the output of the diff-amp) is also (ideally) at V_{bias1}. This selection is fine for general design. It minimizes power by avoiding an extra bias reference circuit. However, if the power supply voltage, *VDD*, starts to drop, the drain-source voltages across the diff-pair and tail current sources can drop to the point where the op-amp shuts off. Again, increasing the widths of the diff-pair helps with this concern. A more general solution is to bias the gates of the NMOS with a gate-drain-connected NMOS, as seen in Fig. 22.30. This allows the gate voltage of the NMOS to move to a higher voltage than its drain (allowing the op-amp to function with lower *VDD*). For example, the diff-amp in Fig. 26.53 has a DC output voltage of roughly 600 mV (= V_{bias1}). A more appropriate voltage

for the gate of the NMOS device is 800 mV. Remembering that the gate-source voltages of the NMOS diff-pair (Fig. 26.18) are roughly 400 mV puts the drains of the diff-pair at 400 mV. The result: *VDD* is more evenly divided across the devices in the diff-amp.

ADDITIONAL READING

[1] V. S. L. Cheung, H. C. Luong, M. Chan, and W. H. Ki, "A 1-V 3.5mW CMOS Switched-Opamp Quadrature IF Circuitry for Bluetooth Receivers," *Symp. VLSI Circuits Dig.* 16, pp. 140–143, June 2002.

[2] M. Keskin, U. Moon, and G. C. Temes, "A 1-V 10-MHz clock-rate 13-bit CMOS DS modulator using unity-gain-reset opamps," *IEEE Journal of Solid-State Circuits,* vol. 37, pp. 817–824, July 2002.

[3] V. S. L. Cheung, H. C. Luong, and W. H. Ki, "A 1V CMOS switched-opamp switched-capacitor pseudo-2-path filter," *IEEE International Solid-State Circuits Conference,* vol. 35, pp. 154–155, February 2000.

[4] I. E. Opris, L. D. Lewicki, and B. C. Wong, "A single-ended 12-bit 20 Msample/s self-calibrating pipeline A/D converter," *IEEE Journal of Solid-State Circuits,* vol. 33, pp. 1898–1903, December 1998.

[5] D. C. Thelen Jr. and D. D. Chu, "A low noise readout detector circuit for nanoampere sensor applications," *IEEE Journal of Solid-State Circuits,* vol. 32, pp. 337–348, March 1997.

[6] P. D. Walker and M. M. Green, "A tuneable pulse-shaping filter for use in a nuclear spectrometer system," *IEEE Journal of Solid-State Circuits,* vol. 31, pp. 850–855, June 1996.

[7] G. Caiulo, F. Maloberti, G. Palmisano, and S. Portaluri, "Video CMOS power buffer with extended linearity," *IEEE Journal of Solid-State Circuits,* vol. 28, pp. 845–848, July 1993.

[8] K. Nakamura and L. R. Carley, "An enhanced fully differential folded-cascode op amp," *IEEE Journal of Solid-State Circuits,* vol. 27, pp. 563–568, April 1992.

[9] R. Castello, G. Nicollini, and P. Monguzzi, "A high-linearity 50-Ω CMOS differential driver for ISDN applications," *IEEE Journal of Solid-State Circuits,* vol. 26, pp. 1809–1816, December 1991.

[10] S. M. Mallya and J. H. Nevin, "Design procedures for a fully differential folded-cascode CMOS operational amplifier," *IEEE Journal of Solid-State Circuits,* vol. 24, pp. 1737–1740, December 1989.

[11] M. Banu, J. M. Khoury, and Y. Tsividis, "Fully Differential Operational Amplifiers with Accurate Output Balancing," *IEEE Journal of Solid State Circuits,* vol. 23, No. 6, pp. 1410–1414, December 1988.

[12] R. Castello and P. R. Gray, "A high-performance micropower switched-capacitor filter," *IEEE Journal of Solid-State Circuits,* vol. 20, pp. 1122–1132, Dec. 1985.

[13] T. C. Choi, R. T. Kaneshiro, R. Broderson, and P. R. Gray, "High-Frequency CMOS Switched Capacitor Filters for Communication Applications," *IEEE Journal of Solid State Circuits,* vol. SC-18, pp. 652–664, December 1983.

[14] D. Senderowicz, S. F. Dreyer, J. H. Huggins, C. F. Rahim, and C. A. Laber, "A family of differential NMOS analog circuits for a PCM codec filter chip," *IEEE Journal of Solid-State Circuits,* vol. 17, pp. 1014–1023, December 1982.

PROBLEMS

Unless otherwise indicated, use the 50 nm CMOS process from the examples in this chapter, the biasing circuit in Fig. 26.3, and 10/1 NMOS and 20/1 PMOS.

26.1 Using simulations, determine the transition frequencies, f_T, for the NMOS and PMOS devices seen in Fig. 26.1 at the nominal operating conditions indicated in the figure. Show that by increasing the MOSFET's overdrive voltage, the f_Ts of the MOSFETs increases.

26.2 Simulate the operation of the two-stage op-amp in Fig. 26.2. Show that the quiescent current in the output buffers is considerably below the desired 20 μA. (Note that the inputs of the op-amp should be held at V_{CM} in the simulation to keep the diff-amp conducting current.) Does this affect the speed of the output buffer? Why or why not?

26.3 Plot reference current against resistor value for the BMR seen in Fig. 26.3. Use simulations to determine the I_{REF} for each value of resistance.

26.4 Comment on the benefits and/or concerns with the CMFB input connections seen in Fig. 26.61. Use simulations to support your answers. Note the similarity to the way CMFB was implemented in Fig. 26.17.

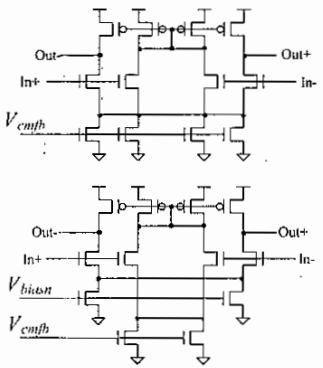


Figure 26.61 Circuits for Problem 26.4.

26.5 Verify, using simulations, that the circuit in Fig. 26.9 does indeed amplify the difference between V_{insp} and the average on the + inputs of the amplifier.

Comment on the operation of the circuit, making sure it is clear that the limitations, uses, and operation of the amplifier are understood.

26.6 Simulate the operation of the CMFB circuit in Fig. 26.16.

26.7 Is *VDD* divided evenly amongst the drain-source voltages of the MOSFETs in Fig. 26.19? Suggest a method to better divide *VDD* amongst the MOSFETs used in the diff-amp.

26.8 Suggest a simple method to speed up the step response of the op-amp in Fig. 26.22 in the circuit of Fig. 26.24. Verify the validity of your suggestion using SPICE simulations.

26.9 Suggest an alternative method to the one seen in Fig. 26.32 for controlling the output buffer's current. Using the modification, regenerate the results seen in Fig. 26.36.

26.10 Is the CMFB loop stable in the op-amp of Fig. 26.40? Use the large-signal test circuit seen in Fig. 26.24 (at a slower frequency) and simulations to look at the stability of this loop. Suggest, and verify with simulations, methods to improve the stability of the CMFB loop.

26.11 Repeat Problem 26.10 for the op-amp in Fig. 26.43.

26.12 As discussed at the end of the chapter and in Fig. 26.59 and the associated discussion, the input common-mode voltage which drops below V_{CM}, can shut off the op-amp's input diff-amp and cause problems. The diff-amp's NMOS devices have a nominal V_{GS} of 400 mV, leaving only 100 mV across the diff-amp's tail current. Show, using the op-amp in Fig. 26.39 in the configuration seen in Fig. 26.24 with an input common-mode voltage less than 500 mV (the input pulse waveforms average to a voltage less than V_{CM}), the resulting problems. Show that increasing the widths of the NMOS diff-pair (and the mirrored NMOS in the bias circuit with gates tied to V_{CM}) from 10 to 30 helps to increase the operating range (four MOSFET widths are increased from 10 to 30). What happens if only the diff-pair widths (two MOSFETs) are increased?

26.13 Repeat Problem 26.12 for the op-amp in Fig. 26.40. What happens if the input common-mode voltage becomes greater than 500 mV?

26.14 Repeat Problem 26.12 for the op-amp in Fig. 26.43. What happens if the input common-mode voltage becomes greater than 500 mV? Why does the op-amp in Fig. 26.40 perform better with variations in the input common-mode voltage than the op-amp in Fig. 26.43?

26.15 Design and simulate the operation of an op-amp using gain-enhancement (Fig. 26.42) and with an open-loop DC gain greater than 2,000, based on the topologies seen in Figs. 26.40 or 26.43. Simulate the operation of your design and generate outputs like those seen in Fig. 26.44.

26.16 Figure 26.62 shows an op-amp based on the topology seen in Fig. 26.40 but biased for lower *VDD* operation. Select the size of the added gate-drain connected MOSFET to allow for proper operation. Simulate the operation of the design showing the DC gain and large signal step responses (as in Fig. 26.34 and 26.35).

26.17 Repeat Problem 26.16 for the op-amp in Fig. 26.43.

26.18 Using the diff-amp in Fig. 26.63 in the configuration seen in Fig. 26.54 (with SC CMFB) and the output buffer in Fig. 26.56 (again with SC CMFB), regenerate the waveforms seen in Fig. 26.60.

26.19 A switched-capacitor integrator is an example of a circuit that uses an op-amp that can't have the outputs of its diff-amp or output buffer shorted when ϕ_1 goes high. Using the diff-amp in Fig. 26.63 (with SC CMFB) and the output buffer in Fig. 26.56 without the switches (again with SC CMFB), demonstrate the operation of a SC integrator. For examples of fully-differential SC circuits, see the second edition of this book entitled, *CMOS Mixed-Signal Circuit Design*.

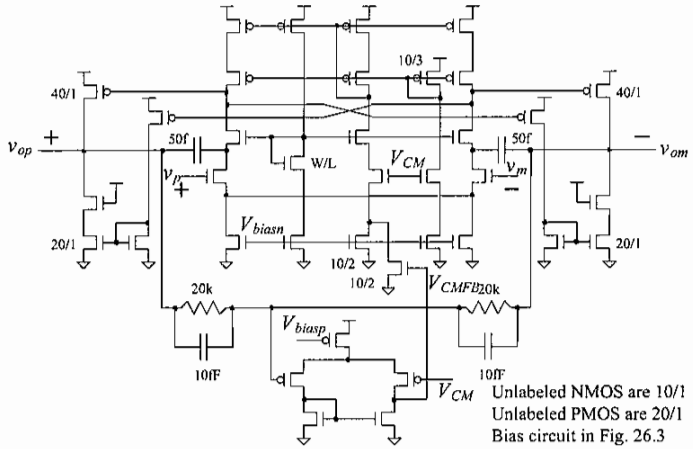


Figure 26.62 An op-amp with better biasing (but more power) for lower VDD operation.

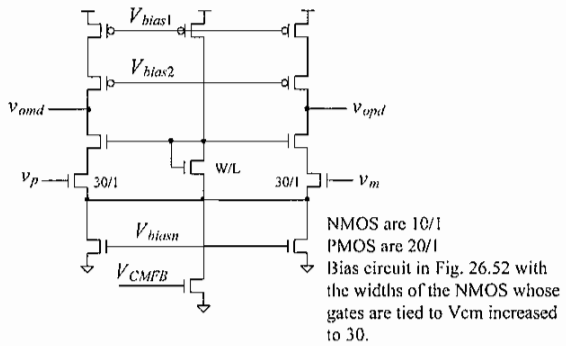


Figure 26.63 Modifying the diff-amp seen in Fig. 26.53 for wider swing operation.

Chapter

27

Nonlinear Analog Circuits

In this book we've studied digital circuits (two discrete amplitude levels) and linear analog circuits (circuits whose input signals are linearly related to the circuits' output signals). In this chapter, we discuss several circuits that are not purely analog or digital. We term these circuits *nonlinear analog circuits* (the inputs are not linearly related to the outputs). In particular, we discuss voltage comparator analysis and design, adaptive biasing, and analog multiplier design.

27.1 Basic CMOS Comparator Design

The schematic symbol and basic operation of a voltage comparator are shown in Fig. 27.1. The comparator can be thought of as a decision-making circuit. If the +, v_p, input of the comparator is at a greater potential than the –, v_m, input, the output of the comparator is a logic 1, whereas if the + input is at a potential less than the – input, the output of the comparator is at a logic 0. Although the basic op-amps developed in Ch. 24 can be used as a voltage comparator, in some less demanding low-frequency or speed applications, we do not consider the op-amp as a comparator. Instead, in this chapter, we discuss practical comparator design and analysis where propagation delay and sensitivity are important.

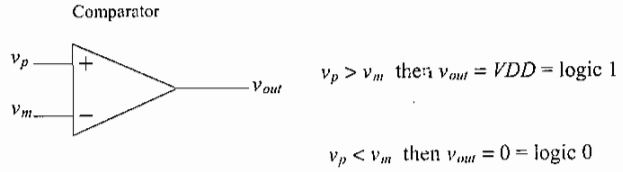


Figure 27.1 Comparator operation.

A block diagram of a high-performance comparator is shown in Fig. 27.2. The comparator consists of three stages: the input preamplifier, a positive feedback or decision stage, and an output buffer. The pre-amp stage (or stages) amplifies the input signal to improve the comparator sensitivity (i.e., increases the minimum input signal with which the comparator can make a decision) and isolates the input of the comparator from switching noise (often called kickback noise) coming from the positive feedback stage (*this is important*, see the discussion in Sec. 16.2.1). The positive feedback stage determines which of the input signals is larger. The output buffer amplifies this information and outputs a digital signal. Designing a comparator can begin with considering input common-mode range, power dissipation, propagation delay, and comparator gain.

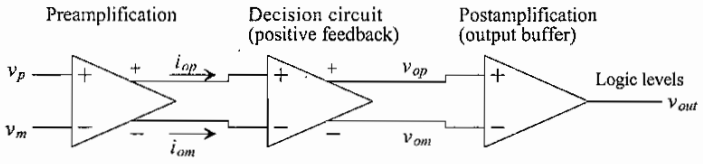


Figure 27.2 Block diagram of a voltage comparator.

Preamplification

For the preamplification (pre-amp) stage, we chose the circuit of Fig. 27.3. For this first section we'll use the long-channel CMOS process to illustrate the design procedures (since long-channel MOSFETs follow the square-law equations). This circuit is a differential amplifier with active loads. The sizes of M1 and M2 are set by considering the diff-amp transconductance, g_m, and the input capacitance. The transconductance sets the gain of the stage, while the input capacitance of the comparator is determined by the sizes of M1 and M2. Notice that there are no high-impedance nodes in this circuit, other than the input and output nodes. This is important to ensure high speed. Using the sizes given in the schematic, we can relate the input voltages to the output currents (noting i_{op} and i_{om} are the small-signal AC currents in the circuit) by

$$i_{op} = \frac{g_m}{2}(v_p - v_m) + \frac{I_{SS}}{2} = I_{SS} - i_{om} \tag{27.1}$$

Noting that if $v_p > v_m$, then i_{op} is positive i_{om} is negative ($i_{op} = -i_{om}$).

Decision Circuit

The decision circuit is the heart of the comparator and should be capable of discriminating mV-level signals. We should also be able to design the circuit with some hysteresis (see Ch. 18) for use in rejecting noise on a signal. The circuit that we use in the comparator under development is shown in Fig. 27.4. The circuit uses positive feedback from the cross-gate connection of M6 and M7 to increase the gain of the decision element.

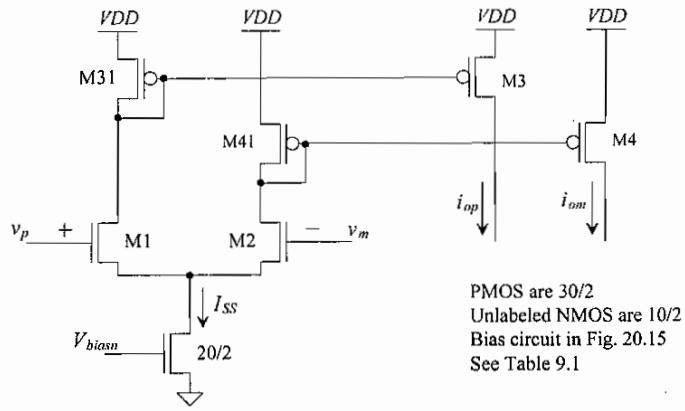


Figure 27.3 Preamplification stage of comparator.

Let's begin by assuming that i_{op} is much larger than i_{om} so that M5 and M7 are on and M6 and M8 are off. We also assume that $\beta_5 = \beta_8 = \beta_A$ and $\beta_6 = \beta_7 = \beta_B$. Under these circumstances, v_{om} is approximately 0 V and v_{op} is

$$v_{op} = \sqrt{\frac{2i_{op}}{\beta_A}} + V_{THN} \tag{27.2}$$

If we start to increase i_{om} and decrease i_{op}, switching starts to take place when the gate-source voltage of M8 is equal to V_{THN}. As we increase M8's V_{GS} beyond V_{THN} (by further increasing i_{om} with the corresponding decrease in i_{op}), M6 starts to take current away from M5. This decreases the drain-source voltage of M5/M6 and thus turns M7 off.

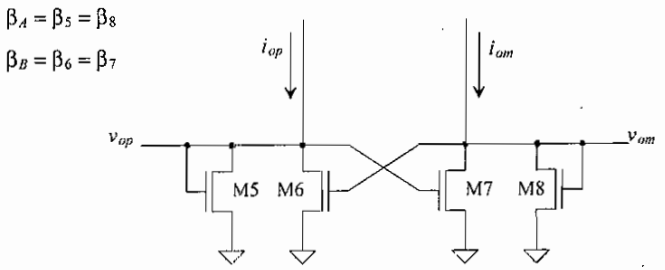


Figure 27.4 Positive feedback decision circuit.

When M8 is just about to turn on (M8's V_{GS} is approaching V_{THN} but the drain currents of M8 and M6 are still zero), the current flowing in M7 is

$$i_{om} = \frac{\beta_B}{2}(v_{op} - V_{THN})^2 \tag{27.3}$$

and the current flowing in M5 is

$$i_{op} = \frac{\beta_A}{2}(v_{op} - V_{THN})^2 \tag{27.4}$$

Noting that the current in M7 (at the switching point) mirrors the current in M5, we can write

$$i_{op} = \frac{\beta_A}{\beta_B} \cdot i_{om} \tag{27.5}$$

If $\beta_A = \beta_B$, then switching takes place when the currents, i_{op} and i_{om}, are equal. Unequal βs cause the comparator to exhibit hysteresis. Relating these equations to Eq. (27.1) yields the switching point voltages (review Ch. 18), or

$$V_{SPH} = v_p - v_m = \frac{I_{SS}}{g_m} \cdot \frac{\frac{\beta_B}{\beta_A} - 1}{\frac{\beta_B}{\beta_A} + 1} \text{ for } \beta_B \geq \beta_A \tag{27.6}$$

and

$$V_{SPL} = -V_{SPH} \tag{27.7}$$

Consider the following example.

Example 27.1

For the circuit shown in Fig. 27.5, estimate and simulate the switching point voltages for two designs: (1) $W_5 = W_6 = W_7 = W_8 = 10$ with $L = 1$ and (2) $W_5 = W_8 = 10$ and $W_6 = W_7 = 12$ with $L = 1$.

For the first case (using the data in Table 9.1)

$$\beta_A = \beta_B = 120\,\frac{\mu A}{V^2} \cdot \frac{10}{1} = 1.2\ mA/V^2$$

so that, as seen in Eq. (27.6) $V_{SPH} = V_{SPL} = 0$. In other words, the comparator does not exhibit hysteresis. Simulation results are shown in Fig. 27.6. The input v_m is set to 2.5 V, while the v_p input is swept from 2.48 to 2.52 V. Note that the amplitudes of v_{op} and v_{om} are limited.

For the second case

$$\beta_A = 120\,\frac{\mu A}{V^2} \cdot \frac{10}{1} \text{ and } \beta_B = 120\,\frac{\mu A}{V^2} \cdot \frac{12}{1}$$

or $\beta_B = 1.2\beta_A$. Using Eqs. (27.6) and (27.7) with $I_{SS} = 40$ μA and $g_m = 150$ μA/V (again, see Table 9.1), we get

$$V_{SPH} = -V_{SPL} = \frac{I_{SS}}{g_m} \cdot \frac{\frac{\beta_B}{\beta_A} - 1}{\frac{\beta_B}{\beta_A} + 1} = \frac{40}{150} \cdot \frac{1.2 - 1}{1.2 + 1} = 24\ mV$$

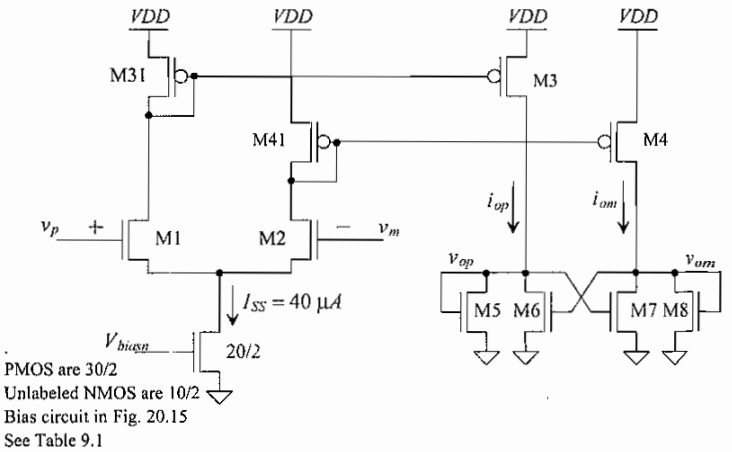
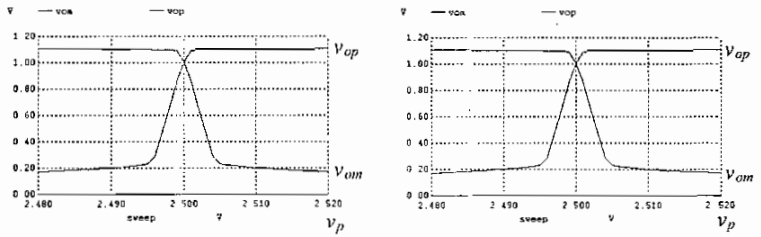


Figure 27.5 Schematic of the pre-amp and decision circuit used in Ex. 27.1.



(a) Minus input held at 2.5 V while the positive input is swept from 2.4 to 2.6 V.

(b) Minus input held at 2.5 V while the positive input is swept from 2.52 to 2.4.

Figure 27.6 The outputs of the decision circuit in Fig. 27.5 without hysteresis.

The simulation results are shown in Fig. 27.7. Figure 27.7a shows a sweep of v_p from 2.4 to 2.6 V, with v_m held at 2.5 V. Since V_{SPH} is 24 mV, the decision circuit switches states when v_p is 24 mV above v_m (i.e., when $v_p = 2.524$ V). The case of v_p being swept from 2.6 to 2.4 V is shown in part (b) of the figure. Switching occurs in this situation when v_p is approximately 2.476 V, or 24 mV, less than 2.5 V. ■

Output Buffer

The final component in our comparator design is the output buffer or post-amplifier. The main purpose of the output buffer is to convert the output of the decision circuit into a logic signal (i.e., 0 or *VDD*). The output buffer should accept a differential input signal

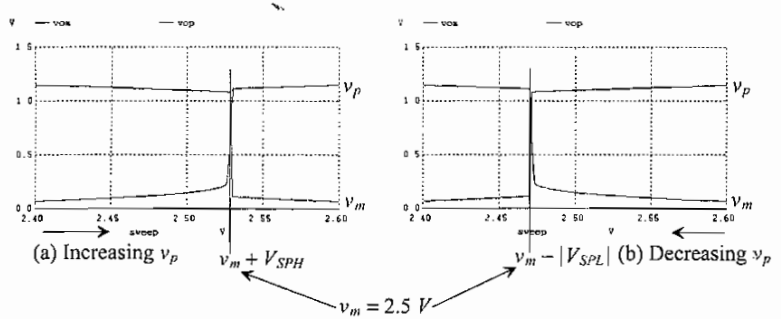


Figure 27.7 Simulating the pre-amp and decision circuit with hysteresis.

and not have slew-rate limitations. We might try to use a NAND SR latch as seen back in Fig. 16.35. However, with process shifts we might run into the situation where the switching points of the gates aren't centered in the middle of the limited swing of the decision circuit's outputs. For a general comparator design, we'll use a diff-amp to help regenerate the digital output signals. For a simple design for the output buffer, we can use the self-biased diff-amp seen back in Fig. 18.17. To move the decision circuit's output swing into the common-mode range of the diff-amp, we can add a gate-drain connected device as seen in Fig. 27.8. An inverter was added on the output of the amplifier as an additional gain stage and to isolate any load capacitance from the self-biasing differential amplifier. This comparator works very well. However, the current in the self-biased stage can be (relatively) huge. This can be a problem if power is a concern (as it is in the Flash ADC discussed in Ch. 29).

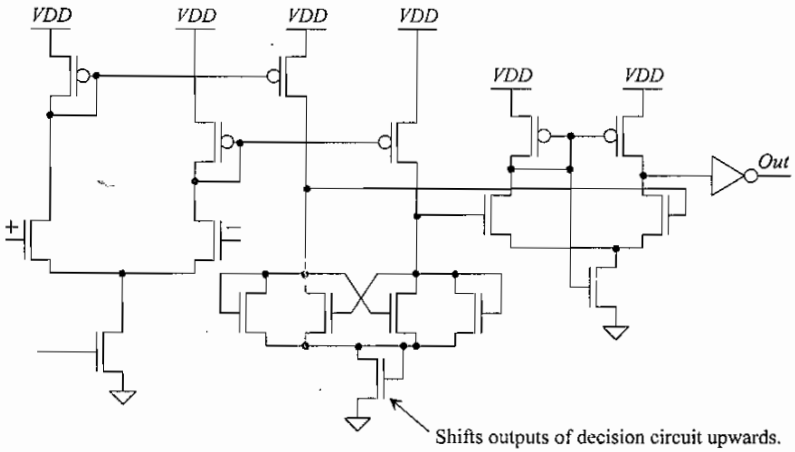


Figure 27.8 Using a self-biased diff-amp for the output buffer.

Figure 27.9 shows a comparator design with both a PMOS and NMOS diff-amp feeding the decision circuit (for input common-mode range beyond the power supply rails). The output buffer is a PMOS diff-amp driving an inverter. While this op-amp won't be as fast as the op-amp seen in Fig. 27.8, it is a good general purpose design. The current drawn from *VDD* will be considerably less than the design in Fig. 27.8. Note that the output currents of the PMOS diff-amp, on the input of the comparator, are added to the NMOS (current) outputs to ensure that current is always sourced to the decision circuit. We might be tempted to connect the PMOS diff-amp directly to the decision circuit. However, this would result in a larger minimum input common-mode voltage.

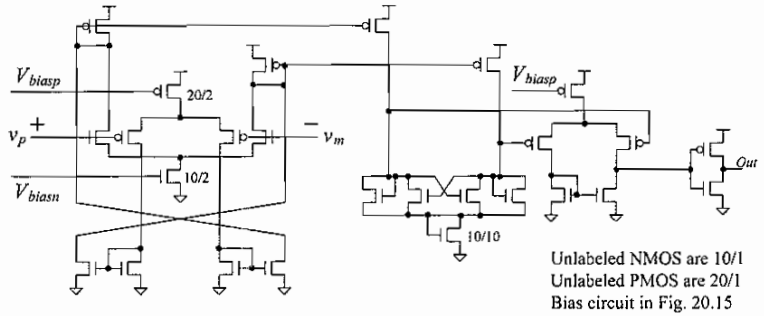


Figure 27.9 General-purpose comparator with rail-to-rail input common-mode range.

27.1.1 Characterizing the Comparator

Comparator DC Performance

Figure 27.10 shows the DC simulated performance of the comparator in Fig. 27.9. The positive comparator input is swept from 0 to *VDD* (= 5 V) while the negative input is stepped for each simulation from 0 to 5 in 500 mV increments (to show wide input common-mode range). Also seen in this figure is the comparator's current draw from *VDD*. Included in this current is the current supplied to the bias circuit.

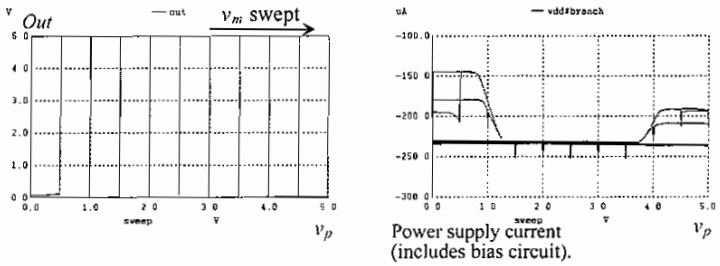


Figure 27.10 DC performance of the comparator in Fig. 27.9

If we take the derivative of these transfer curves, the gain of the comparator and thus the smallest difference that can be discriminated between v_p and v_m becomes known. We have to be careful here with the step size used in the DC simulation. If, for example, we were to use a step size of 1 mV, then the maximum gain we get from the simulation is 1,000. Figure 27.11 shows the expanded view of the comparator with the minus input held at 2.5 V while the positive input is swept from 2.499 to 2.501 V. The offset seen in the figure is a systematic offset (as discussed earlier). The gain of the comparator is, roughly, 175,000. Notice that the comparator was designed without hysteresis. However, in a practical comparator mismatch in the decision circuit results in hysteresis (and so if hysteresis isn't desired, the layout of the decision circuit is critical and may require common-centroid techniques).

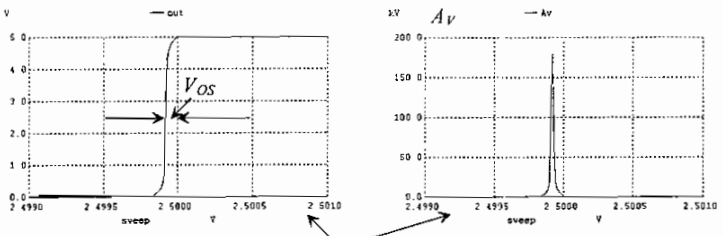


Figure 27.11 The gain of the comparator in Fig. 27.9

Transient Response

The transient response of a comparator can be significantly more difficult to characterize than the DC characteristics. Let's begin by considering v_m = 2.5 V DC, with the v_p input to the comparator a 10 ns wide pulse and an amplitude varying from 2.45 to 2.55 V. This is termed a narrow pulse with a 50 mV overdrive; we are driving the + input of the comparator 50 mV over the negative input. The simulation results for these inputs are shown in Fig. 27.12. If the pulse amplitude or width is reduced much beyond this, the comparator does not make a full transition. Notice how we only show simulation data after 120 ns. This was done to ensure the bias circuit had time to start-up and stabilize before changing the inputs of the comparator.

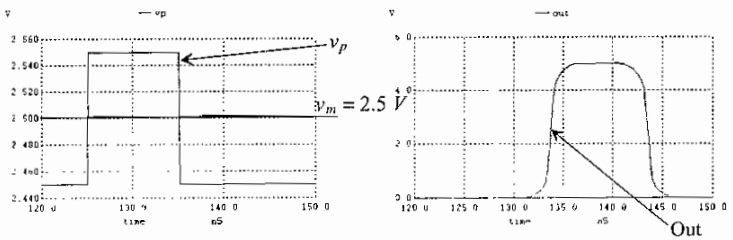


Figure 27.12 Transient response of the comparator in Fig. 27.9.

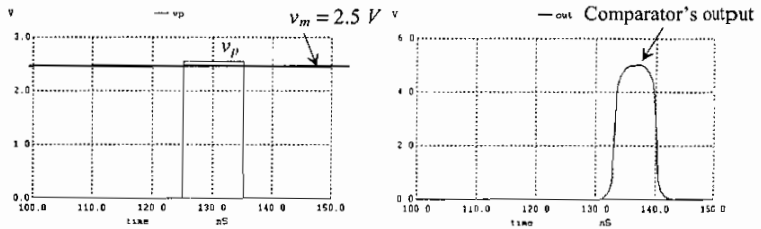


Figure 27.13 Transient response of the comparator in Fig. 27.9 with one side of the pre-amp initially shut off.

Although these results are interesting, they are not practically useful in some situations. In few cases will the comparator discriminate between signals similar to what is seen in Fig. 27.12. A better indication of comparator performance is to apply a signal that starts at a voltage where one side of the differential amplifier is cut off and to finish at a voltage slightly larger than the reference voltage (2.5 V in the above example), Fig. 27.13. In applying a 0 to 2.6 V pulse to the v_p input, the left side of the diff-amp is initially off and then on. The internal comparator nodes must be charged over a wider voltage range with a 0–2.6 V input when compared to the 2.49–2.51 V example of Fig. 27.12. In order to keep one side of the pre-amp from turning off, the circuit of Fig. 27.14 can be used. The added MOSFET ensures that the voltage between the drains of M31 and M41 don't deviate too much. This configuration is sometimes referred to as a clamped input stage. Note that the size of the added MOSFET should be as small as possible to avoid unneeded loading in the diff-amp (which slows down the comparator's performance).

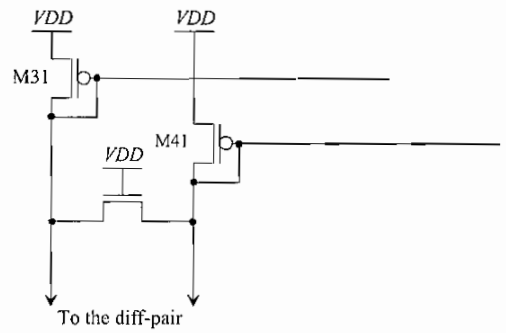


Figure 27.14 Adding a balancing resistor to the input stage of a comparator. See Fig. 27.5

Propagation Delay

Ideally, the propagation delay (the time difference between the input, v_p, crossing the reference voltage, v_m, and the output changing logic states) is zero. For the comparator of Fig. 27.9 with simulation results shown in Fig. 27.12, the delay of the comparator is approximately 9 ns (or less with an overdrive). This should be compared with the delay of an op-amp used as a comparator, which may be several hundred nanoseconds (because of the compensation capacitor). Another interesting fact about comparator design is that the delay of a comparator can be reduced by cascading gain stages. In other words, the delay of a single high-gain stage is in general longer than the delay of several low-gain stages. Improvements in comparator sensitivity and speed can be directly related to improvements in the preamplifier. A derivation (similar to that given in Ch. 11 for minimum delay through an inverter string driving a load capacitance, C_L) of the number of stages, N, needed in a preamplifier to reach minimum delay for a given load capacitance and input capacitance results in

$$N = \ln \frac{C_L}{C_{in}} \tag{27.8}$$

In the derivation of this equation, it was assumed that the driving resistance of any stage is $1/g_{mn}$ (the transconductance of the nth differential amplifier stage). In practice, Eq. (27.8) is of little use because one side of the differential amplifier can be off. For this reason (and others such as size and power draw), comparators with more than two or three pre-amp stages are not common.

Minimum Input Slew Rate

The last characteristic we discuss is the minimum input slew rate of a comparator. If the input signals to the comparator vary at a slow rate (e.g., a sine wave generated from the AC line), the output of the comparator may very well oscillate resulting in an output with a metastable state. If a comparator is to be used with slowly varying signals, or in a noisy environment, the decision circuit should have hysteresis. The minimum input slew rate is difficult to simulate with SPICE because of the slow and fast varying signals present at the same instant of time in the circuit. This same situation (metastability) can also occur if the input overdrive is small. The result, for this case, however, is an increase in the delay time of the comparator.

27.1.2 Clocked Comparators

We developed "sense amplifiers" back in Sec. 16.2.1. The sense amplifier can be used as a clocked comparator, that is a comparator whose outputs change on the rising (or falling) edge of a clock signal. Consider the clocked comparator in Fig. 27.15. This circuit is directly taken from Figs. 16.32 and 16.35. It works very well for signal differences that are tens of millivolts. When *clock* is low, the inputs to the NAND SR latch are pulled high. The outputs of the comparator don't change. When *clock* goes high, the two inputs are compared causing the output of the circuit to register which one is higher. As discussed in detail back in Ch. 16, the minimum input voltages for this comparator are above the threshold voltage of the NMOS devices. Further if the input signals get too large, MB1 and MB2 move deep into the triode region. The small differences in their channel resistances adversely affect the quality of the comparator's decisions. Let's modify this topology for better sensitivity, wider input signal swing, and better immunity to kickback.

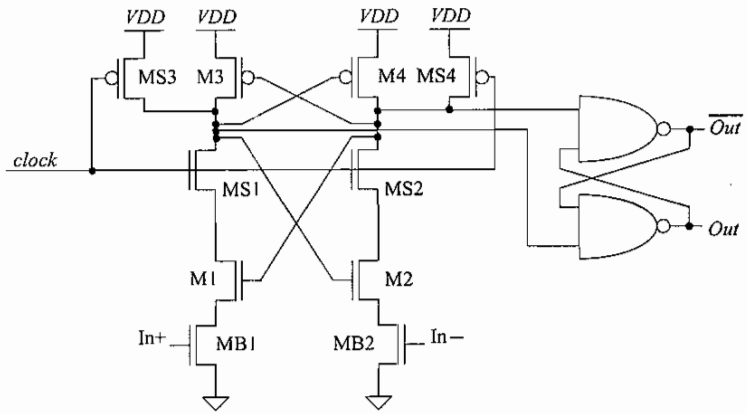


Figure 27.15 A clocked comparator based on Figs. 16.32 and 16.35.

Our modification is seen in Fig. 27.16. Instead of biasing the PMOS and NMOS diff-amps with a bias circuit, here, to be different, we simply use long-length MOSFETs. The bias currents aren't too critical in this application (like they are in an op-amp design). We can still use a bias circuit if controlling the bias current is, for some reason, important. We've removed the triode-operating MOSFETs MB1 and MB2 from the basic cross- coupled latch section. Now we are steering currents from one side of the latch to the other. The differences in the currents causes the latch to switch dependent on the input signals. Note that if either of the diff-amps isn't present in the comparator or if one is off because the comparator's input common-mode voltage is too high or too low, the comparator still works as desired. Only one diff-amp is needed to create the imbalance. Also note, again, that the SR latch is used to make the outputs of the circuit change on the rising edge of the clock signal. If the outputs of the comparator can go high, for a particular application, when the clock signal is low then the NAND gates can be removed.

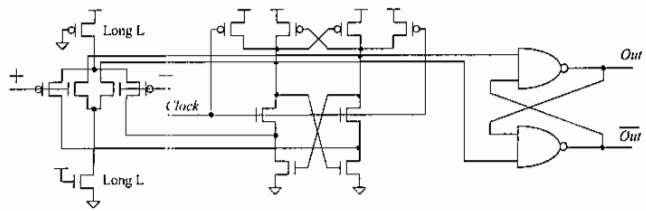


Figure 27.16 Wide-swing clocked comparator. Outputs change on the rising edge of the clock signal.

27.1.3 Input Buffers Revisited

When we discussed input buffers back in Ch. 18 we used a self-biased design for the highest speed (see Fig. 18.17 and the associated discussions). As discussed in Ch. 18, it's desirable to have symmetrical propagation delays independent of input slew-rate, amplitudes, or direction (high-to-low or low-to-high). Towards the more ideal input buffer, consider using two NMOS self-biased buffers in parallel, as seen in Fig. 27.17. Here, to maintain good symmetry, we've split the current sources in half and used one side to generate a common-mode feedback signal to balance the outputs. To reduce the power dissipation in the buffer, the triode-operating MOSFETs can have their lengths increased. Further, for rail-to-rail input common-mode range, we can place the PMOS version of this buffer in parallel with the NMOS version seen in Fig. 27.17, as we did in Fig. 18.23. This input buffer can be very useful for low-skew, high-speed design.

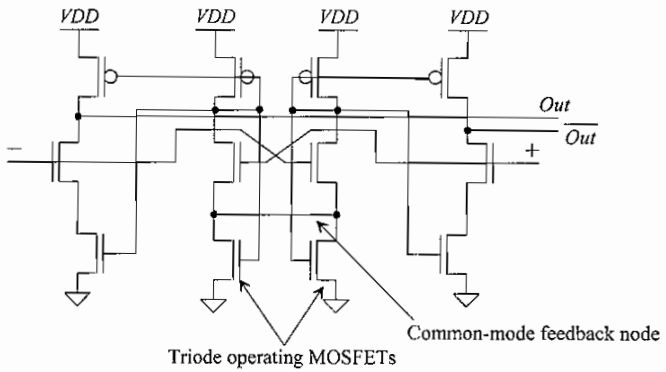


Figure 27.17 A fully-differential input buffer based on the topology discussed in Sec. 18.17.

27.2 Adaptive Biasing

Adaptive biasing can reduce power dissipation in an amplifier while at the same time increasing output current drive capability. Figure 27.18 can be used to help illustrate the idea. When v_{I1} and v_{I2} are equal, the current sources I_{SS1} and I_{SS2} are zero (an open). The diff-amp DC tail current is simply I_{SS}, the same as an ordinarily biased diff-amp. If v_{I1} becomes larger than v_{I2}, the current source I_{SS1} increases above zero, effectively increasing the diff-amp DC bias current. Similarly, if v_{I2} becomes larger than v_{I1}, the current source I_{SS2} increases above zero. The diff-amp output current is normally limited to I_{SS} when one side of the diff-amp shuts off. However, now that the maximum output current is limited to either $I_{SS} + I_{SS1}$ or $I_{SS} + I_{SS2}$. Power dissipation can be reduced using an adaptive bias, and slew-rate problems can be eliminated.

If $v_{I1} = v_{I2}$ then $I_{SS1} = I_{SS2} = 0$.

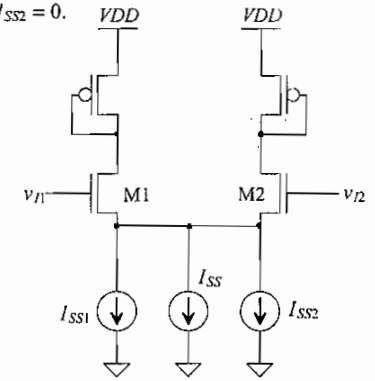


Figure 27.18 Adaptively biased diff-amp.

The current diff-amp of Fig. 27.19 can be used to implement the current source I_{SS1} or I_{SS2}. If the currents I_1 and I_2 are equal, then zero current flows in M3 and M4. Also, if I_2 is greater than I_1, zero current flows in M3 and M4. If I_1 is larger than I_2, the difference between these two currents $(I_1 - I_2)$ flows in M3. Since M4 is K times wider than M3, a current of $K(I_1 - I_2)$ flows in M4 (normally $K < 1$). Two of these diff-amps are needed to implement the adaptive biasing of the diff-amp of Fig. 27.18.

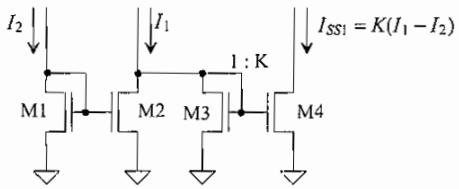


Figure 27.19 Current diff-amp used in adaptive biasing.

Figure 27.20 shows the implementation of adaptive biasing into the diff-amp of Fig. 27.18. P-channel MOSFETs are added adjacent to M3 and M4 to mirror the currents through M1 and M2 (I_1 and I_2). The maximum total current available through M1 occurs when M2 is off. Positive feedback exists through the loop M1, M3, M5–M7. Initially, when M2 shuts off, the current in M1 and M3 is I_{SS}. This is mirrored in M5 and M6, and thus I_{SS1} becomes $K \cdot I_{SS}$. At this particular instance in time, the tail current, which flows through M1, is now $I_{SS} + K \cdot I_{SS}$. However, provided the MOSFETs M1, M3–M7 remain in

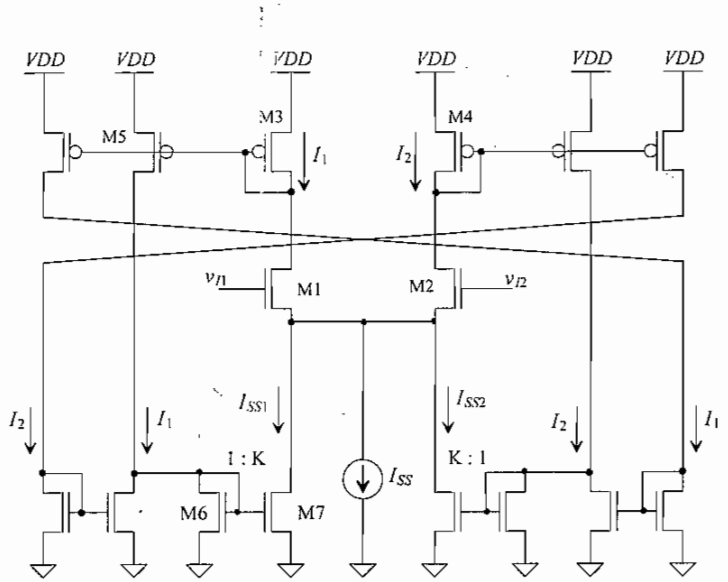


Figure 27.20 Adaptively biased diff-amp.

saturation, this current circles back around the positive feedback loop and increases by K. This continues, resulting in a final or total tail current of

$$I_{tot} = I_{SS} \cdot (1 + K + K^2 + K^3 + ...) \tag{27.9}$$

If $K < 1$, this geometric series can be written as

$$I_{tot} = \frac{I_{SS}}{1-K} \tag{27.10}$$

Setting $K = 0$ (MOSFET M7 doesn't exist) results in no adaptive biasing and a total tail current of I_{SS}. Setting $K = 1/2$ (M7 half the size of M6) results in a total available tail current of $2 \cdot I_{SS}$. Because the tail current limits the slew rate, when the diff-amp is driving a capacitive load, making K equal to one eliminates slew-rate limitations while at the same time not increasing static power dissipation. In practice, shutting off one side of the diff-pair, M1/M2, is difficult since the adaptive biasing has the effect of lowering the source potentials of the diff-pair, keeping both MOSFETs on. Adaptive biasing can be used in a comparator where M1 or M2 can shut off. However, the static power dissipation will be large; therefore, there is no benefit over the comparators discussed earlier in the chapter. When applying adaptive biasing to an OTA design, the value of K should be unity or less. [Using a K of 1 or 2 can still result in a finite I_{tot} since the MOSFETs have a finite output resistance and the MOSFETs in the diff-pair will not shut off, as was assumed in the derivation of Eq. (27.10)].

A final example of an adaptive voltage-follower amplifier is shown in Fig. 27.21. This amplifier can only source current to a load. If v_{in} and v_{out} are equal, the current that flows in M1 and M2 is $I_{SS} + I_{D6}$. If v_{in} is increased, the current in M1 and M3 increases. This causes the currents in M4–M6 to increase, effectively increasing the tail current of the diff-pair. The result is a large current available to drive the load. Note that M7 can be sized larger than the other MOSFETs to increase maximum output current.

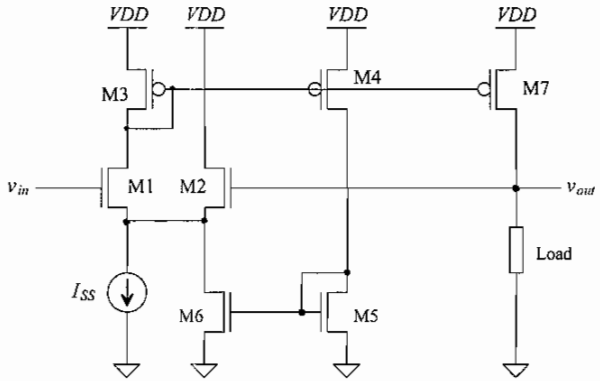


Figure 27.21 Adaptive voltage follower.

27.3 Analog Multipliers

Analog multipliers find extensive use in communication systems. Figure 27.22 shows the voltage characteristics of a four-quadrant multiplier. This multiplier is termed a four-quadrant multiplier because both inputs can be either positive or negative around a common-mode voltage, V_{CM}. The ideal output of the multiplier is related to the inputs by

$$v_{out} = K_m \cdot v_x v_y \tag{27.11}$$

where K_m is the multiplier gain with units of V^{-1}. In reality, imperfections exist in the multiplier gain, resulting in offsets and nonlinearities. The output of the multiplier can be written as

$$v_{out} = K_m(v_x + V_{OSx})(v_y + V_{OSy}) + V_{OSout} + v_x^n + v_y^m \tag{27.12}$$

where V_{OSx}, V_{OSy}, and V_{OSout} are the offset voltages associated with the x-, y-inputs, and the output, respectively. The terms v_x^n and v_y^m represent nonlinearities in the multiplier. Normally, these nonlinearities are specified in terms of the total harmonic-distortion or by specifying the maximum deviation in percentages between a straight line and the actual characteristic curves shown in Fig. 27.22 over some range of input voltages. Although many different techniques exist for implementing analog multipliers in CMOS, we concentrate on a technique useful in high- and low-frequency multiplication.

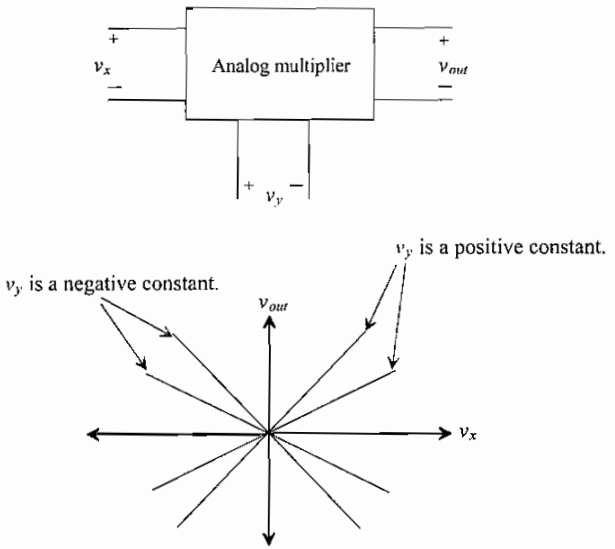


Figure 27.22 Operation of a four-quadrant analog multiplier.

27.3.1 The Multiplying Quad

A CMOS multiplier employing a multiplying quad (M1–M4) is shown in Fig. 27.23. The multiplying quad operates in the triode region, and thus MOSFETs M1–M4 can be thought of as resistors. For the moment we will not consider the biasing of the quad. The negative output voltage of the multiplier is given by

$$v_{om} = -R \cdot (i_{D1} + i_{D2}) \tag{27.13}$$

while the positive output voltage is

$$v_{op} = -R \cdot (i_{D3} + i_{D4}) \tag{27.14}$$

The output voltage of the multiplier is

$$v_{out} = v_{op} - v_{om} = R \cdot (i_{D1} + i_{D2} - i_{D3} - i_{D4}) \tag{27.15}$$

A simplified schematic of the multiplying quad with biasing is shown in Fig. 27.24. The op-amp inputs are at an AC virtual ground and at a DC voltage of V_{CM} (the op-amp output common-mode voltage). In order to minimize the DC input current on the x-axis inputs, the common-mode DC voltage on this input is set to V_{CM}. The DC biasing voltage on the y-input is set to a value large enough to keep the quad in triode. The input signals have been broken into two parts (e.g., $v_x/2$ and $-v_x/2$) to maintain generality. In practice, the minus inputs can be connected directly to the bias voltages at the cost of

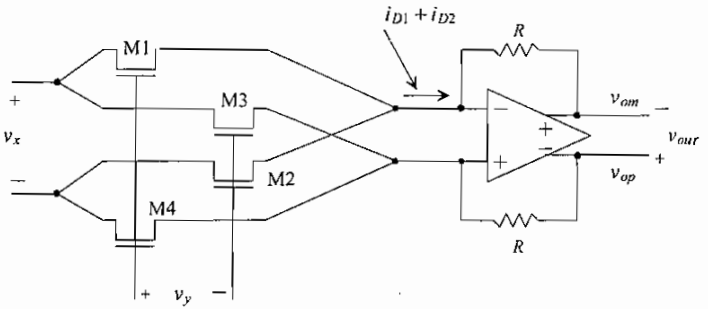


Figure 27.23 CMOS analog multiplier.

large-signal linearity. (The input is not truly differential in this situation.) However, as discussed earlier, a fully differential system has much better coupled noise immunity.

Using Eq. (9.12) and noticing that the DC gate-source voltage of all MOSFETs is the same, the drain currents can be written as

$$i_{D1} = \beta_1 \left[\left(V_{GS} + \frac{v_y}{2} - V_{THN1} \right) \left(\frac{v_x}{2} \right) - \frac{1}{2} \left(\frac{v_x}{2} \right)^2 \right] \tag{27.16}$$

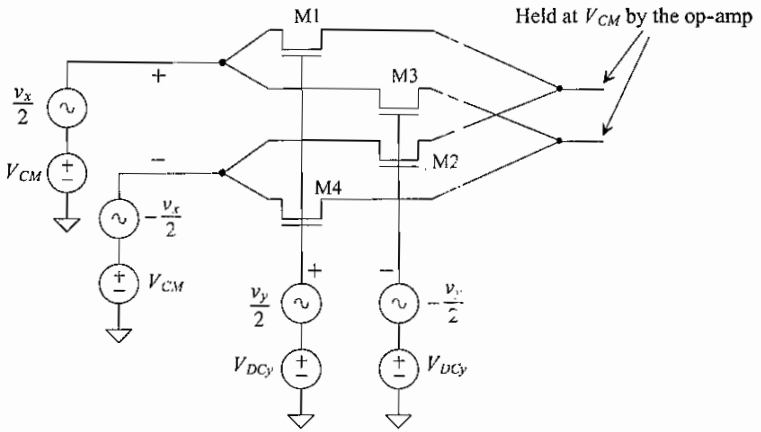


Figure 27.24 Biasing of the multiplying quad.

$$i_{D2} = \beta_2\left[\left(V_{GS} - \frac{v_y}{2} - V_{THN2}\right)\left(-\frac{v_x}{2}\right) - \frac{1}{2}\left(-\frac{v_x}{2}\right)^2\right] \quad (27.17)$$

$$i_{D3} = \beta_3\left[\left(V_{GS} - \frac{v_y}{2} - V_{THN3}\right)\left(\frac{v_x}{2}\right) - \frac{1}{2}\left(\frac{v_x}{2}\right)^2\right] \quad (27.18)$$

$$i_{D4} = \beta_4\left[\left(V_{GS} + \frac{v_y}{2} - V_{THN4}\right)\left(-\frac{v_x}{2}\right) - \frac{1}{2}\left(-\frac{v_x}{2}\right)^2\right] \quad (27.19)$$

We can design so that $\beta = \beta_1 = \beta_2 = \beta_3 = \beta_4$. We can use Eq. (27.15) together with Eqs. (27.16)–(27.19) to rewrite the output voltage of the multiplier as

$$v_{out} = R\beta \cdot \left(\frac{v_x}{2}\right)\left[\frac{v_y}{2} - V_{THN1} + \frac{v_y}{2} + V_{THN2} + \frac{v_y}{2} + V_{THN3} + \frac{v_y}{2} - V_{THN4}\right] \quad (27.20)$$

We can see that if $V_{THN1} = (V_{THN2}$ or $V_{THN3})$ and $V_{THN4} = (V_{THN3}$ or $V_{THN2})$, this equation can be rewritten as

$$v_{out} = R\beta \cdot v_x v_y \quad (27.21)$$

The source of a MOSFET (the terminal we label "source" depends on which way current flows in the MOSFET) in the multiplying quad is connected either to the op-amp or to the x inputs. When the sources of the MOSFETs are connected to the op-amp, all of the MOSFETs in the multiplying quad have the same threshold voltage. (Since the source of each MOSFET is tied to the same potential, the body effect changes each MOSFET's threshold voltage by the same amount.) If the positive x-input is sinking a current, then the sources of M1 and M3 are the "+" x-input and thus $V_{THN1} = V_{THN3}$. In any case, the threshold voltages of the MOSFETs cancel and Eq. (27.21) holds. Comparing Eqs. (27.21) and (27.11) results in defining the gain of this multiplier as

$$K_m = R \cdot \beta \quad (27.22)$$

Simulating the Operation of the Multiplier

Simulating the performance and understanding the analog multiplier operation is an important step in the design process. The design of a multiplier consists of designing the op-amp, selecting the sizes of the multiplying quad, and designing the biasing network. Because we covered the design of differential input/output op-amps in the last chapter, it is not covered here. In order to simulate the performance of a multiplier in SPICE without including the limitations of the op-amp, the simple model shown in Fig. 27.25 is used. The sum of the multiplying factors associated with the voltage-controlled voltage sources, E1 and E2, is the open-loop gain of the op-amp. A typical SPICE statement for these VCVS (voltage-controlled voltage source) where the op-amp open loop gain is 20,000 is

```
E1    Voplus 8 4 3   1E4
E2    8 Vominus 4 3  1E4
```

where the nodes correspond to those labeled in Fig. 27.25.

The next problem we encounter in simulating the operation of the multiplier is implementing the differential voltages (e.g., $\pm v_x/2$), in addition to the DC biasing voltages. The setup shown in Fig. 27.26 is used to implement the biasing and the differential voltage sources. The op-amp common-mode output voltage, V_{CM}, and the x

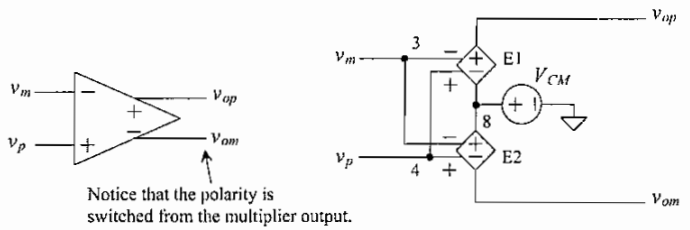


Figure 27.25 SPICE modeling a differential input/output op-amp with common-mode voltage.

input voltage are set to 1.5 V. The lower this voltage, the easier it is to bias the multiplying quad into the triode region. On the other hand, a reduction in the value of V_{CM} limits the op-amp output voltage swing and thus the multiplier output range. The size of the multiplying quad was set to 10/2. The larger the *W/L* ratio of the MOSFETs used in this quad, the easier it is to keep the quad in the triode region. On the other hand, using a large *W/L* increases the required input current. The channel length can be increased to ensure the device operates as a long-channel device and follows Eq. (9.12). Since the quad is part of the feedback around the op-amp, long-channel devices do not affect the

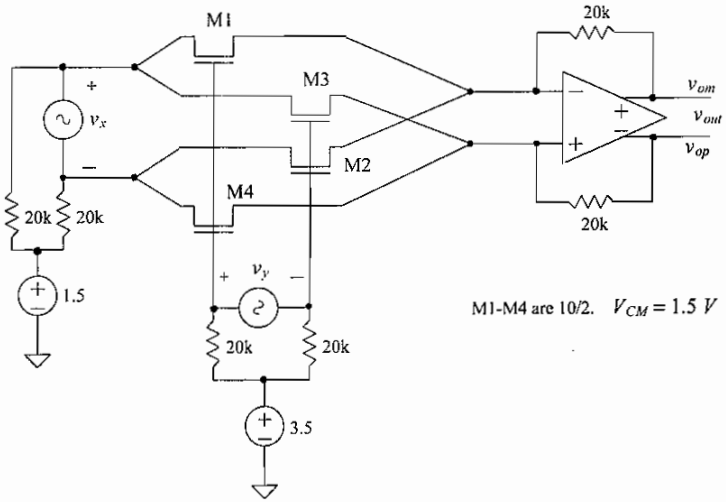


Figure 27.26 SPICE simulation schematic.

speed. The DC voltage at the y-inputs was set to 3.5 V as a compromise between keeping the multiplying quad in triode and the y-input voltage range. The gain of the multiplier in Fig. 27.26 is, from Eq. (27.22) and Table 6.2,

$$K_m = 20k \cdot 120\frac{\mu A}{V^2} \cdot \frac{10}{2} = 12\ V^{-1}$$

A DC sweep showing the operation of the multiplier is shown in Fig. 27.27. The x-input, v_x, was swept from −1 to +1, while at the same time the y-input was stepped from −1 to 1 V in 0.5 V increments. Keeping in mind that the output of the multiplier is $v_{op} - v_{om}$, we can understand the data presented in Fig. 27.27 by considering points A and B. At point A, the y-input is 1 V while the x-input is 0.5 V. The output voltage of the multiplier is the product of the multiplier gain and these two voltages (i.e., 12·1·0.25 = 3 V). The output voltage at point B is 12·0.5·(−0.3) = 1.8 V. Note that this figure was generated using an almost ideal op-amp. The characteristics do not show the limitations of the op-amp. In particular, the limited output swing.

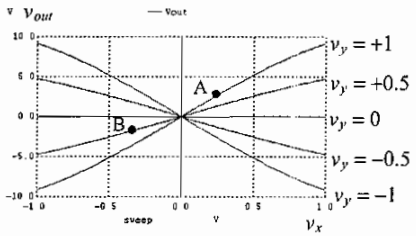


Figure 27.27 DC characteristics of the multiplier of Fig. 26.26.

27.3.2 Multiplier Design Using Squaring Circuits

An analog multiplier can be designed based on the difference between the sum of two voltages squared and the difference of two voltages squared, or

$$V_o = (V_1 + V_2)^2 - (V_1 - V_2)^2 = 4V_1V_2 \qquad (27.23)$$

The basic sum-squaring and difference-squaring circuits are shown in Fig. 27.28. MOSFETs M1 and M4 are source-followers, while MOSFETs M2 and M4 are called squaring MOSFETs. This circuit is designed so that $\beta_1 = \beta_4 = \beta_{14}$, $\beta_2 = \beta_3 = \beta_{23}$, and $\beta_{14} >> \beta_{23}$. This makes almost all of the DC bias currents, I_{S12} and I_{S34}, flow in the MOSFETs M1 and M4, respectively. The squaring current, I_{SQ}, assuming that zero current flows through the resistor when both inputs are zero volts (or whatever the common-mode voltage when a single supply is used), is given by

$$I_{SQ(a)} = \frac{\beta_{23}}{4}(V_1 + V_2)^2 \qquad (27.24)$$

Similarly, the squaring current in the difference-square circuit of Fig. 27.28b is given by

$$I_{SQ(b)} = \frac{\beta_{23}}{4}(V_1 - V_2)^2 \qquad (27.25)$$

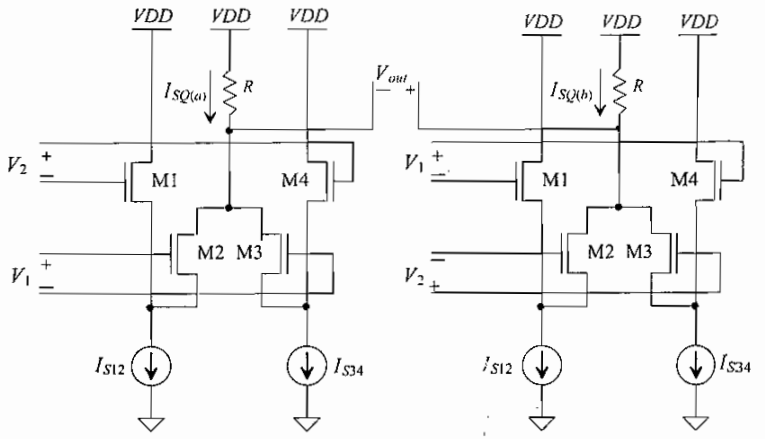


Figure 27.28 (a) Sum-squaring circuit and (b) difference squaring circuit.

The output voltage of the sum-square circuit is given by

$$V_{o-} = VDD - I_{SQ(a)}R \qquad (27.26)$$

while the output voltage of the difference-square circuit is given by

$$V_{o+} = VDD - I_{SQ(b)}R \qquad (27.27)$$

A multiplier is formed by taking the difference between these voltages. The output voltage of the multiplier of Fig. 27.27 is given by

$$V_{out} = V_{o+} - V_{o-} = R\frac{\beta_{23}}{4}\left[(V_1 + V_2)^2 - (V_1 - V_2)^2\right] \qquad (27.28)$$

or, using Eq. (27.23)

$$V_{out} = R\beta_{23} \cdot V_1V_2 \qquad (27.29)$$

The fundamental concern with using this type of multiplier is the fact that modern short-channel CMOS devices don't follow the square-law equations (Eq. [27.24]) we used to derive this result.

ADDITIONAL READING

[1] D. J. Allstot, "A Precision Variable-Supply CMOS Comparator," *IEEE Journal of Solid-State Circuits*, vol. SC-17, no. 6, pp. 1080–1087, December 1982.

[2] M. G. Degrauwe, J. Rijmenants, E. A. Vittoz, and H. J. DeMan, "Adaptive Biasing CMOS Amplifiers," *IEEE Journal of Solid-State Circuits*, vol. SC-17, no. 3, pp. 522–528, June 1982.

[3] E. A. Vittoz, "Micropower Techniques," Chapter 3 in J. E. Franca and Y. Tsividis (eds.) *Design of Analog-Digital VLSI Circuits for Telecommunications and Signal Processing*, 2nd ed., Prentice Hall, 1994. ISBN 0-13-203639-8.

[4] S. Soclof, *Applications of Analog Integrated Circuits*, Prentice Hall, 1985. ISBN 0-13-039173-5.

[5] M. Ismail, S-C. Huang, and S. Sakurai, "Continuous-Time Signal Processing," Chapter 3 in M. Ismail and T. Fiez (eds.), *Analog VLSI: Signal and Information Processing*, McGraw-Hill, 1994. ISBN 0-07-032386-0.

[6] B-S. Song, "CMOS RF Circuits for Data Communications Applications," *IEEE Journal of Solid-State Circuits*, vol. SC-21, no. 2, pp. 310–317, April 1986.

[7] J. Crols and M. S. J. Steyaert, "A 1.5 GHz Highly Linear CMOS Downconversion Mixer," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 7, pp. 736–742, July 1995.

[8] H-J. Song and C-K. Kim, "A MOS Four-Quadrant Analog Multiplier Using Simple Two-Input Squaring Circuits with Source Followers," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, pp. 841–848, June 1990.

PROBLEMS

27.1 Using the long-channel CMOS process, compare the performance (using simulations) of the comparator in Fig. 27.8 with the comparator in Fig. 27.9. Your comparison should include DC gain, systematic offset, delay, sensitivity, and power consumption.

27.2 Show, using simulations, how the addition of a balancing resistor in Fig. 27.14 can be used to improve the response seen in Fig. 27.13.

27.3 Simulate the operation of the comparator in Fig. 27.15 in the short-channel CMOS process. Determine the comparators sensitivity and the kickback noise.

27.4 Repeat problem 27.3 for the comparator in Fig. 27.16. Show that the input common-mode range of the comparator in Fig. 27.16 extends beyond the power supply rails.

27.5 Simulate the operation of the input buffer in Fig. 27.17 in the short-channel CMOS process. How sensitive is the buffer to input slew-rate? How symmetrical are the output rise and fall times? Suggest, and verify with simulations, a method to reduce the power consumed by the input buffer.

27.6 Design a low power clocked comparator for use with a Flash ADC (discussed in Ch. 29). Use the short-channel CMOS process and a clocking frequency of 250 MHz estimate the power dissipated by 256 of these comparators.

Chapter

28

Data Converter Fundamentals

Data converters (a circuit that changes analog signals to digital representations or vice-versa) play an important role in an ever-increasing digital world. As more products perform calculations in the digital or discrete time domain, more sophisticated data converters must translate the digital data to and from our inherently analog world. This chapter introduces concepts of data conversion and sampling which surround this useful circuit.

28.1 Analog Versus Discrete Time Signals

Analog-to-digital converters, also known as A/Ds or ADCs, convert analog signals to discrete time or digital signals. Digital-to-analog converters (D/As or DACs) perform the reverse operation. Figure 28.1 illustrates these two operations.To understand the functionality of these data converters, it would be wise first to compare the characteristics of analog versus digital signals.

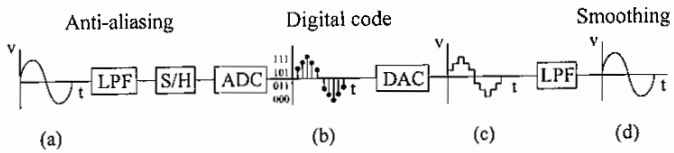


Figure 28.1 Signal characteristics caused by A/D and D/A conversion.

In Fig. 28.1 the original analog signal (a) is filtered by an anti-aliasing filter to remove any high-frequency components that may cause an effect known as aliasing (see Sec. 28.5). The signal is sampled and held and then converted into a digital signal (b). Next the DAC converts the digital signal back into an analog signal (c). Note that the output of the DAC is not as "smooth" as the original signal. A low-pass filter returns the analog signal back to its original form (plus phase shift introduced from the conversions)

after eliminating the higher order signal components caused by the conversion. This example illustrates the main differences between analog and digital signals. Whereas the analog signal in Fig. 28.1a is *continuous* and *infinite* valued, the digital signal in (b) is *discrete* with respect to time and *quantized*. The term *continuous-time signal* refers to a signal whose response with respect to time is uninterrupted. Simply stated, the signal has a continuous value for the entire segment of time for which the signal exists. By referring to the analog signal as infinite valued, we mean that the signal can possess any value between the parameters of the system. For example, in Fig. 28.1a, if the peak amplitude of the sine wave was +1V, then the analog signal can be any value between –1 and 1 V (such as 0.4758393848 V). Of course, measuring all the values between –1 and 1 V would require a piece of laboratory equipment with infinite precision.

The digital signal, on the other hand, is discrete with respect to time. This means that the signal is defined for only certain or discrete periods of time. A signal that is quantized can only have certain values (as opposed to an infinitely valued analog signal) for each discrete period. The signal illustrated in Fig. 28.1b illustrates these qualities.

28.2 Converting Analog Signals to Digital Signals

We have already established the differences between analog and digital signals. How is it possible to convert from an analog signal to a digital signal? An example will illustrate the process.

Where you live the temperature in the winter stays between 0° F and 50° F (Fig. 28.2a). Suppose you had a thermometer with only two readings, hot and cold, and you wanted to record the weather patterns and plot the results. The two quantization levels can be correlated with the actual temperature as follows:

If $0° F \leq T < 25° F$	Temperature is recorded as cold
If $25° F \leq T < 50° F$	Temperature is recorded as hot

You take a measurement every day at noon and plot the results after one week. From Fig. 28.2b, it is apparent that your discretized version of the weather is not an accurate representation of the actual weather.

Now suppose that you find another thermometer with four possible temperatures (hot, warm, cool, and cold) and you increase the number of readings to two per day. The result of this reading is seen in Fig. 28.3a. The quantization levels represent four equal bands of temperature as seen below:

If $0° F \leq T < 12.5° F$	Temperature is recorded as cold
If $12.5°F \leq T < 25° F$	Temperature is recorded as cool
If $25° F \leq T < 37.5° F$	Temperature is recorded as warm
If $37.5° F \leq T < 50° F$	Temperature is recorded as hot

Here, the digital version of the weather still looks nothing like the actual weather pattern, but the critical issues in digitizing an analog signal should be apparent. The actual weather pattern is the analog signal. It is continuous with respect to time, and its value can be between 0° F and 50° F (even 33.9638483920398439° F!). The accuracy of the digitized signal is dependent on two things: the number of samples taken and the

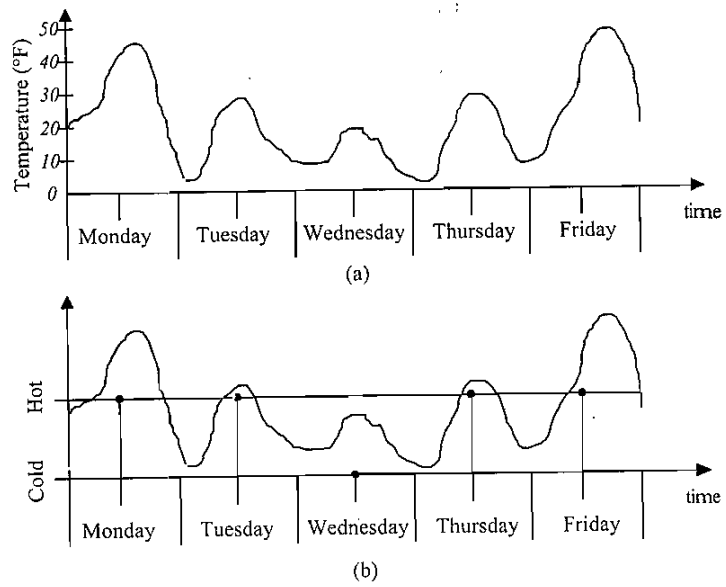


Figure 28.2 (a) An analog signal representing the temperature where you live and (b) a digital representation of the analog signal taking one sample per day with two quantization levels.

resolution, or number of quantization levels, of the converter. In our example, we need to increase both the number of samples and the resolution of thermometer.

Suppose that finally we obtain a thermometer with 25 temperature readings and that we take a reading eight times per day. Each of the 25 quantization levels now represents a 2° F band of temperature. From Fig. 28.3b, we can see that the digital version of the weather is approaching that of the actual analog signal. If we kept increasing both sampling time and resolution, the difference between the analog and the digital signals would become negligible. This brings up another critical issue: how many samples should one take in order to accurately represent the analog signal?

Suppose a sudden rainstorm swept through your town and caused a sharp decrease in temperature before returning to normal. If that storm had occurred between our sampling times, our experiment would not have shown the effects of the storm. Our sampling time was too slow to catch the change in the weather. If we had increased the number of samples, we would have recognized that something happened which caused the temperature to drop dramatically during that period.

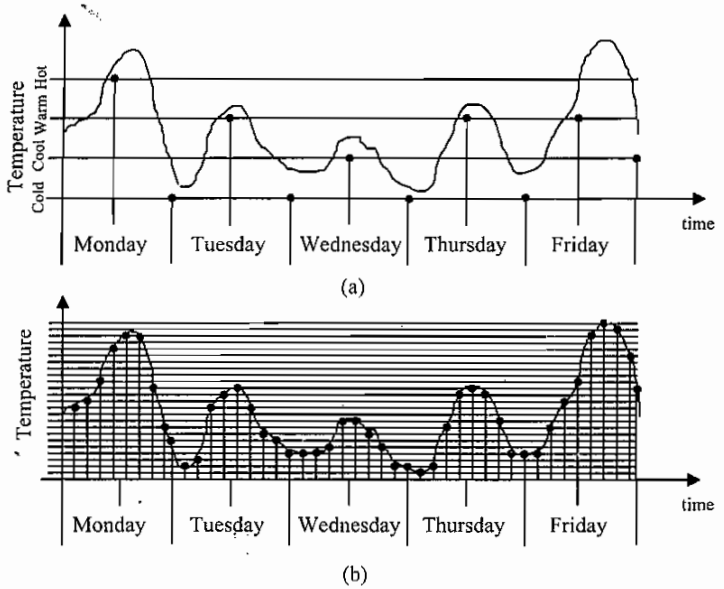


Figure 28.3 Digital representation of the temperature taking (a) two samples per day with four quantization levels and (b) nine samples per day with 25 quantization levels.

As it turns out, the *Nyquist Criterion* defines how fast the sampling rate needs to be to represent an analog signal accurately. This criterion requires that the sampling rate is at least two times the highest frequency contained in the analog signal. In our example, we need to know how quickly the weather can change and then take samples twice as fast as that value. The Nyquist Criterion can be described as

$$F_{sampling} = 2\,F_{MAX} \tag{28.1}$$

where $F_{sampling}$ is the sampling frequency required to accurately represent the analog signal and F_{MAX} is the highest frequency of the sampled signal.

How much resolution should we use to represent the analog signal accurately? There is no absolute criterion for this specification. Each application will have its own requirements. In our weather example, if we were only interested in following general trends, then the 25 quantization levels would more than suffice. However, if we were interested in keeping an accurate record of the temperature to within ±0.5° F, we would need to double the resolution to 50 quantization levels so that each quantization level would correspond to each degree ±0.5° F (Fig. 28.4).

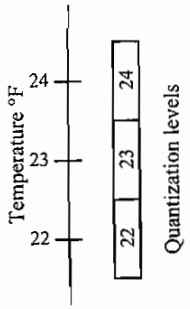


Figure 28.4 Quantization levels overlap actual temperature by ±½° F.

28.3 Sample-and-Hold (S/H) Characteristics

Sample-and-hold (S/H) circuits are critical in converting analog signals to digital signals. The behavior of the S/H is analogous to that of a camera. Its main function is to "take a picture" of the analog signal and hold its value until the ADC can process the information. It is important to characterize the S/H circuit when performing data conversion. Ignoring this component can result in serious error, for both speed and accuracy can be limited by the S/H. Ideally, the S/H circuit should have an output similar to that shown in Fig. 28.5a. Here, the analog signal is instantly captured and held until the next sampling period. However, a finite period of time is required for the sampling to occur. During the sampling period, the analog signal may continue to vary; thus, another type of circuit is called a track-and-hold, or T/H. Here, the analog signal is "tracked" during the time required to sample the signal, as seen in Fig. 28.5b. It can be seen that S/H circuits operate in both static (hold mode) and dynamic (sample mode) circumstances. Thus, characterization of the S/H will be discussed in the context of these two categories. Figure 28.6 presents a summary of the major errors associated with a S/H [1–5]. A discussion of each error follows.

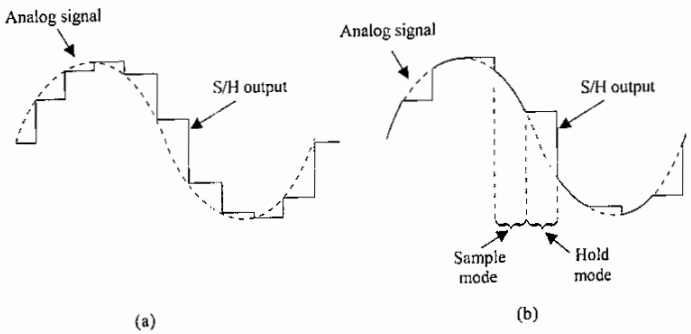


Figure 28.5 The output of (a) an ideal S/H circuit and (b) a track-and-hold (T/H).

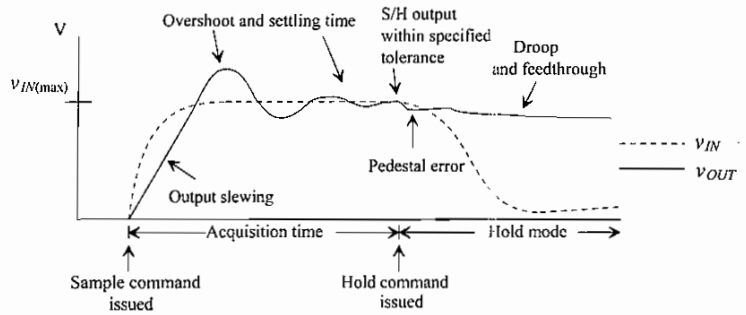


Figure 28.6 Typical errors associated with an S/H.

Sample Mode

Once the sampling command has been issued, the time required for the S/H to track the analog signal to within a specified tolerance is known as the *acquisition time*. In the worst-case scenario, the analog signal would vary from zero volts to its maximum value, $v_{IN(max)}$. And the worst-case acquisition time would correspond to the time required for the output to transition from zero to $v_{IN(max)}$. Since most S/H circuits use amplifiers as buffers (as seen in Fig. 28.7), it should be obvious that the acquisition will be a function of the amplifier's own specifications. For example, notice that if the input changes very quickly, then the output of the T/H could be limited by the amplifier's slew rate. The amplifier's stability is also extremely critical. If the amplifier is not compensated correctly, and the phase margin is too small, then a large *overshoot* will occur. A large overshoot requires a longer *settling time* for the S/H to settle within the specified tolerance. The error tolerance at the output of the S/H also depends on the amplifier's *offset*, *gain error* (ideally, the S/H should have a gain of 1) and *linearity* (the gain of the S/H should not vary over the input voltage range).

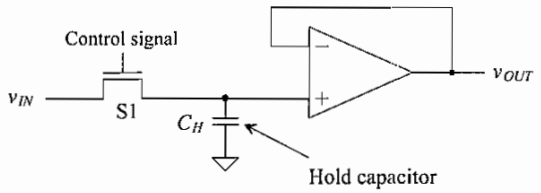


Figure 28.7 Track-and-hold circuit using an output buffer.

Hold Mode

Once the hold command is issued, the S/H faces other errors. Pedestal error occurs as a result of charge injection and clock feedthrough. Part of the charge built up in the channel of the switch is distributed onto the capacitor, thus slightly changing its voltage. Also, the clock couples onto the capacitor via overlap capacitance between the gate and the source or drain. Another error that occurs during the hold mode is called *droop*. This error is related to the leakage of current from the capacitor due to parasitic impedances and to the leakage through the reverse-biased diode formed by the drain of the switch. This diode leakage can be minimized by making the drain area as small as can be tolerated. Although the input impedance of the buffer amplifier is very large, the switch has a finite OFF impedance through which leakage can occur. Current can also leak through the substrate. The key to minimizing droop is increasing the value of the sampling capacitor. The trade-off, however, is increased time that's required to charge the capacitor to the value of the input signal.

Aperture Error

A transient effect that introduces error occurs between the sample and the hold modes. A finite amount of time, referred to as aperture time, is required to disconnect the capacitor from the analog input source. The aperture time actually varies slightly as a result of noise on the hold-control signal and the value of the input signal, since the switch will not turn off until the gate voltage becomes less than the value of the input voltage less one threshold voltage drop. This effect is called *aperture uncertainty* or *aperture jitter*. As a result, if a periodic signal were being sampled repeatedly at the same points, slight variations in the hold value would result, thus creating *sampling error*. Figure 28.8 illustrates this effect. Note that the amount of aperture error is directly related to the frequency of the signal and that the worst-case aperture error occurs at the zero crossing, where *dV/dt* is the greatest. This assumes that the S/H circuit is capable of sampling both positive and negative voltages (bipolar). The amount of error that can be tolerated is directly related to the resolution of the conversion. Aperture error will be discussed again in Sec. 28.5 as it relates to the error in an ADC.

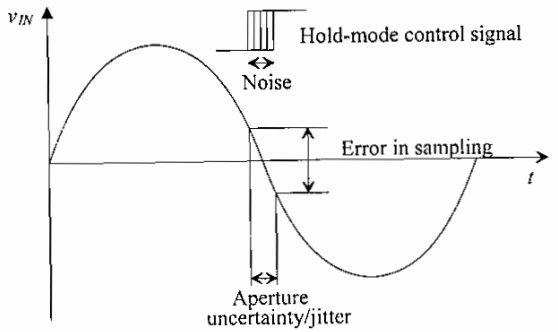


Figure 28.8 Aperture error.

Example 28.1
Find the maximum sampling error for a S/H circuit that is sampling a sinusoidal input signal that could be described as

$$v_{IN} = A \sin 2\pi ft$$

where A is 2 V and f = 100 kHz. Assume that the aperture uncertainty is equal to 0.5 ns.

The sampling error due to the aperture uncertainty can be thought of as a slew rate such that

$$\frac{dV}{dt} = \frac{d}{dt} A \sin 2\pi ft = 2\pi fA \cos 2\pi ft$$

with maximum slewing occurring when the cosine term is equal to 1. Therefore,

$$\frac{dV}{dt}(max) = 2\pi fA = (2\pi \cdot 100 \text{ kHz})(2 \text{ V})$$

and the maximum sampling error is

$$Maximum\ Sampling\ Error = dV(max) \quad \text{or}$$

$$(0.5 \times 10^{-9} \text{ s})(2\pi \cdot 100 \text{ kHz})(2 \text{ V}) = 0.628 \text{ mV} \blacksquare$$

28.4 Digital-to-Analog Converter (DAC) Specifications

Probably the most popular digital-to-analog converter application is the digital audio compact disc player. Here digital information stored on the CD is converted into music via a high-precision DAC. Many characteristics define a DAC's performance. Each characteristic will be discussed before we look at the basic architectures in Ch. 29. This "top-down" approach allows a smoother transition from the data converter characteristics to the actual architectures, since most data converters have similar performance limitations. A discussion of some of the basic definitions associated with DACs follows. It should be noted that DACs and ADCs can use either voltage or current as their analog signal. For purposes of describing specifications, it will be assumed that the analog signal is a voltage.

A block diagram of a DAC can be seen in Fig. 28.9. Here an N-bit digital word is mapped into a single analog voltage. Typically, the output of the DAC is a voltage that is some fraction of a reference voltage (or current), such that

$$v_{OUT} = FV_{REF} \tag{28.2}$$

where v_{OUT} is the analog voltage output, V_{REF} is the reference voltage, and F is the fraction defined by the input word, D, that is N bits wide. The number of input combinations represented by the input word D is related to the number of bits in the word by

$$\text{Number of input combinations} = 2^N \tag{28.3}$$

A 4-bit DAC has a total of 2^4 or 16 total input values. A converter with 4-bit resolution must be able to map a change in the analog output, which is equal to 1 part in 16. The maximum analog output voltage for any DAC is limited by the value of some reference voltage, V_{REF}. If the input is an N-bit word, then the value of the fraction, F, can be determined by,

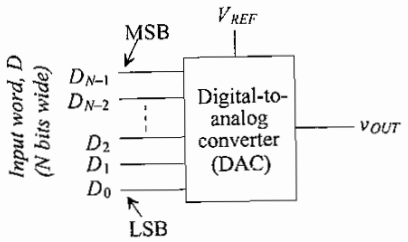


Figure 28.9 Block diagram of the digital-to-analog converter.

$$F = \frac{D}{2^N} \tag{28.4}$$

Therefore, if a 3-bit DAC is being used, the input, D, is $100 = 4_{10}$, and V_{REF} is 5 V, then the value of F is

$$F = \frac{100}{2^3} = \frac{4}{8} \tag{28.5}$$

and the analog voltage that appears at the output becomes,

$$v_{OUT} = \frac{4}{8}(5) = 2.5 \text{ V} \tag{28.6}$$

By plotting the input word, D, versus v_{OUT} as D is incremented from 000 to 111, the *transfer curve* seen in Fig. 28.10 would be generated. The y-axis has been normalized to V_{REF}; therefore, the graduated marks also represent F by Eq. (28.2). Some important characteristics need to be discussed here. First, notice that the transfer curve is not continuous. Since the input is a digital signal, which is inherently discrete, the input signal can only have eight values that must correspondingly produce eight output voltages. If a straight line connected each of the output values, the slope of the line would ideally be one increment/input code value. Also note that the maximum value of the output is 7/8. Since the case where D = 000 has to result in an analog voltage of 0 V, and a 3-bit DAC has eight possible analog output voltages, then the analog output will increase from 0 V to only 7/8 V_{REF}.

Again, using Eq. (28.2), this means that the maximum analog output that can be generated by the 3-bit DAC is

$$v_{OUT(max)} = \frac{7}{8} \cdot V_{REF} \tag{28.7}$$

This maximum analog output voltage that can be generated is known as *full-scale voltage*, V_{FS}, and can be generalized to any N-bit DAC as

$$V_{FS} = \frac{2^N - 1}{2^N} \cdot V_{REF} \tag{28.8}$$

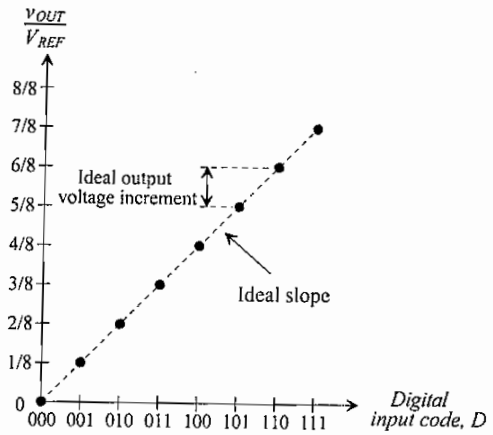


Figure 28.10 Ideal transfer curve for a 3-bit DAC.

The *least significant bit* (*LSB*) refers to the rightmost bit in the digital input word. The LSB defines the smallest possible change in the analog output voltage. The LSB will always be denoted as D_0. One LSB can be defined as

$$1\ LSB = \frac{V_{REF}}{2^N} \tag{28.9}$$

In the previous case of the 3-bit DAC, 1 LSB = 5/8 V, or 0.625 V. Generating an output in multiples of 0.625 V may not seem difficult, but as the number of bits increases, the voltage value of one LSB decreases for a fixed value of V_{REF}.

The *most significant bit* (*MSB*) refers to the leftmost bit of the digital word, *D*. In the previous example, $D = 100$ or $D_2D_1D_0$, with D_2 being the MSB. Generalizing to the *N*-bit DAC, the MSB would be denoted as D_{N-1}. (Since the LSB is denoted as bit 0, the MSB is denoted as *N*-1.) Note that when discussing DACs, the MSB causes the output to change by 1/2 V_{REF}.

When discussing data converters, the term *resolution* describes the smallest change in the analog output with respect to the value of the reference voltage, V_{REF}. This is slightly different from the definition of LSB in that resolution is typically given in terms of bits and represents the *number of unique output voltage levels*, i.e., 2^N.

Example 28.2

Find the resolution for a DAC if the output voltage is desired to change in 1 mV increments while using a reference voltage of 5 V.

The DAC must resolve

$$\frac{1\text{ mV}}{5\text{ V}} = 0.0002 \text{ or } 0.02\% \text{ adjustability}$$

Therefore, the *accuracy* required for 1 LSB change over a range of V_{REF} is

$$\frac{1\ LSB}{V_{REF}} = \frac{1}{2^N} = 0.0002 \tag{28.10}$$

and solving *N* for the resolution yields

$$N = \text{Log}_2\left(\frac{5\text{ V}}{1\text{ mV}}\right) = 12.29 \text{ bits}$$

which means that a 13-bit DAC will be needed to produce the accuracy capable of generating 1 mV changes in the output using a 5 V reference. ■

Example 28.3

Find the number of input combinations, values for 1 LSB, the percentage accuracy, and the full-scale voltage generated for a 3-bit, 8-bit, and 16-bit DAC, assuming that $V_{REF} = 5$ V.

Using Eqs. (28.3), (28.8), (28.9), and (28.10), we can generate the following information:

Resolution	Input combinations	1 LSB	% accuracy	V_{FS}
3	8	0.625 V	12.5	4.375 V
8	256	19.5 mV	0.391	4.985 V
16	65,536	76.29 μV	0.00153	4.9999 V

The value of 1 LSB for an 8-bit converter is 19.5 mV, while 1 LSB for a 16-bit converter is 76.3 μV (a factor of 256)! Increasing the resolution by 1 bit increases the accuracy by a factor of 2. The precision required to map the analog signal at high resolutions is very difficult to achieve. We will examine some of these issues as we examine the limitations of the data converter in Ch. 29.

Note that a data converter may have a resolution of 8 bits, where an LSB is 19.5 mV as above, while having a much higher accuracy. For example, we could require the 8-bit data converter above to have an accuracy of 0.1%. The higher accuracy results in a more ideal (linear) DAC. A typical specification for DAC accuracy is ±½ LSB for reasons discussed below. ■

Differential Nonlinearity

As seen in the ideal DAC in Fig. 28.10, each adjacent output increment should be exactly one-eighth. Since the y-axis is normalized, the values for the increment heights will be unitless. However, the increment heights can be easily converted to volts by multiplying the height by V_{REF}. This corresponds to the ideal increment corresponding to 0.625 V = 1 LSB (assuming $V_{REF} = 5$ V).

Nonideal components cause the analog increments to differ from their ideal values. The difference between the ideal and nonideal values is known as *differential nonlinearity*, or *DNL* and is defined as

DNL_n = Actual increment height of transition *n* – Ideal increment height (28.11)

where n is the number corresponding to the digital input transition. The DNL specification measures how well a DAC can generate uniform analog LSB multiples at its output.

Example 28.4

Determine the DNL for the 3-bit nonideal DAC whose transfer curve is shown in Fig. 28.11. Assume that $V_{REF} = 5$ V.

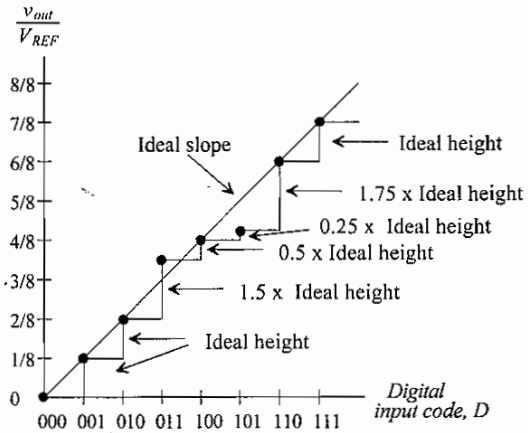


Figure 28.11 Example of differential nonlinearity for a 3-bit DAC.

The actual increment heights are labeled with respect to the ideal increment height, which is 1 LSB, or 1/8 of $\frac{v_{OUT}}{V_{REF}}$. Notice that there is no increment corresponding to 000, since it is desirable to have zero output voltage with a digital input code of 000. The increment height corresponding to 001, however, is equal to the corresponding height of the ideal case seen in Fig. 28.10; therefore, $DNL_1 = 0$. Similarly, DNL_2 is also zero since the increment associated with the transition at 010 is equal to the ideal height. Notice that the 011 increment, however, is not equal to the ideal curve but is 3/16, or 1.5 times the ideal height.

$$DNL_3 = 1.5 \text{ LSB} - 1 \text{ LSB} = 0.5 \text{ LSB}$$

Since we have already determined in Eq. (28.9) that for a 3-bit DAC, 1 LSB = 0.625 V, we can convert the DNL_3 to volts as well. Therefore, $DNL_3 = 0.5$ LSB = 0.3125 V. However, it is popular to refer to DNL in terms of LSBs. The remainder of the digital output codes can be characterized as follows:

$$DNL_4 = 0.5 \text{ LSB} - 1 \text{ LSB} = -0.5 \text{ LSB}$$

$$DNL_5 = 0.25 \text{ LSB} - 1 \text{ LSB} = -0.75 \text{ LSB}$$

$$DNL_6 = 1.75 \text{ LSB} - 1 \text{ LSB} = 0.75 \text{ LSB}$$

$$DNL_7 = 1 \text{ LSB} - 1 \text{ LSB} = 0$$

If we were to plot the value of DNL (in LSBs) versus the input digital code, Fig. 28.12 would result. The DNL for the entire converter used in this illustration is ±0.75 LSB since the overall error of the DAC is defined by its worst-case DNL. ■

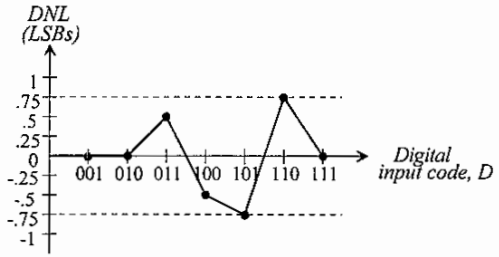


Figure 28.12 DNL curve for the nonideal 3-bit DAC.

Generally, a DAC will have less than ±½ LSB of DNL if it is to be N-bit accurate. A 5-bit DAC with 0.75 LSBs of DNL actually has the resolution of a 4-bit DAC. If the DNL for a DAC is less than –1 LSBs, then the DAC is said to be *nonmonotonic*, which means that the analog output voltage does not always increase as the digital input code is incremented. A DAC should always exhibit *monotonicity* if it is to function without error.

Integral Nonlinearity

Another important static characteristic of DACs is called *integral nonlinearity* (*INL*). Defined as the difference between the data converter output values and a reference straight line drawn through the first and last output values, INL defines the linearity of the overall transfer curve and can be described as

$$INL_n = \text{Output value for input code } n - \text{Output value of the reference line at that point} \tag{28.12}$$

An illustration of this measurement is presented in Fig. 28.13. It is assumed that all other errors due to offset and gain (these will be discussed shortly) are zero. An example follows shortly.

It is common practice to assume that a converter with N-bit resolution will have less than ±½ LSB of DNL and INL. The term, ½ *LSB*, is a common term that typically denotes the maximum error of a data converter (both DACs and ADCs). For example, a 13-bit DAC having greater than ±½ LSB of DNL or INL actually has the resolution of a 12-bit DAC. The value of ½ LSB in volts is simply

$$0.5\ LSB = \frac{V_{REF}}{2^{N+1}} \tag{28.13}$$

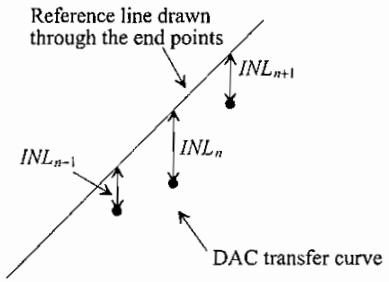


Figure 28.13 Measuring the INL for a DAC transfer curve.

Example 28.5

Determine the INL for the nonideal 3-bit DAC shown in Fig. 28.14. Assume that $V_{REF} = 5$ V.

First, a reference line is drawn through the first and last output values. The INL is zero for every code in which the output value lies on the reference line; therefore, $INL_2 = INL_4 = INL_6 = INL_7 = 0$. Only outputs corresponding to 001, 011, and 101 do not lie on the reference. Both the 001 and the 011 transitions occur ½ LSB higher than the straight-line values; therefore, $INL_1 = INL_3 = 0.5$ LSB. By the same reasoning, $INL_5 = -0.75$ LSB. Therefore, the INL for the DAC is considered to be its worst-case INL of +0.5 LSB and –0.75 LSB. The INL plot for the nonideal 3-bit DAC can be seen in Fig. 28.15. ■

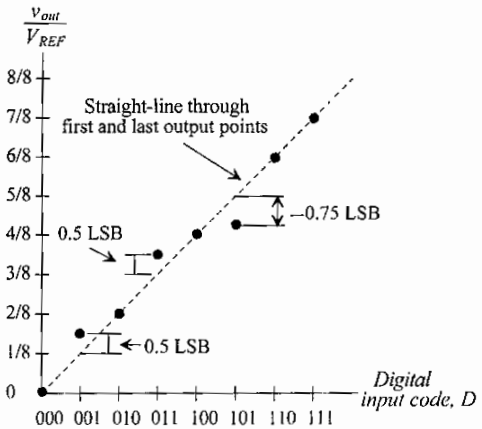


Figure 28.14 Example of integral nonlinearity for a DAC.

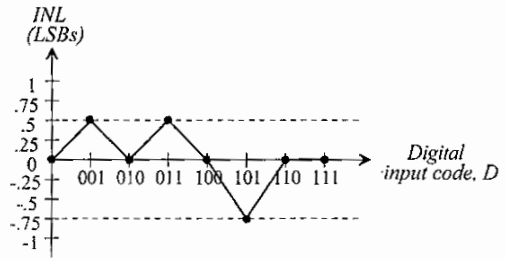


Figure 28.15 INL curve for the nonideal 3-bit DAC.

It should be noted that other methods are used to determine INL. One method compares the output values to the ideal reference line, regardless of the positions of the first and last output values. If the DAC has an offset voltage or gain error, this will be included in the INL determination. Usually, the offset and gain errors are determined as separate specifications.

Another method, described as the "best-fit" method, attempts to minimize the INL by constructing the reference line so that it passes as closely as possible to a majority of the output values. Although this method does minimize the INL error, it is a rather subjective method that is not as widely used as drawing the reference line through the first and last output values.

Offset

The analog output should be 0 V for $D = 0$. However, an offset exists if the analog output voltage is not equal to zero. This can be seen as a shift in the transfer curve as illustrated in Fig. 28.16. This specification is similar to the offset voltage for an operational amplifier except that it is not referred to the input.

Gain Error

A gain error exists if the slope of the best-fit line through the transfer curve is different from the slope of the best-fit line for the ideal case. For the DAC illustrated in Fig. 28.17, the gain error becomes

$$\text{Gain error} = \text{Ideal slope} - \text{Actual slope} \tag{28.14}$$

Latency

This specification defines the total time from the moment that the input digital word changes to the time the analog output value has settled to within a specified tolerance. Latency should not be confused with settling time, since latency includes the delay required to map the digital word to an analog value plus the settling time. It should be noted that settling time considerations are just as important for a DAC as they are for a S/H or an operational amplifier.

Signal-to-Noise Ratio (SNR)

Signal-to-noise (SNR) is defined as the ratio of the signal power to the noise at the analog output. In amplifier applications, this specification is typically measured using a

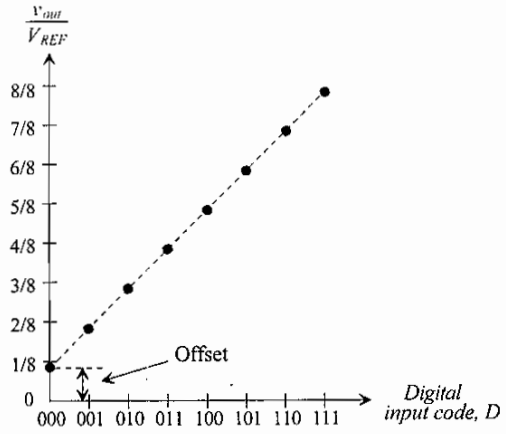


Figure 28.16 Illustration of offset error for a 3-bit DAC.

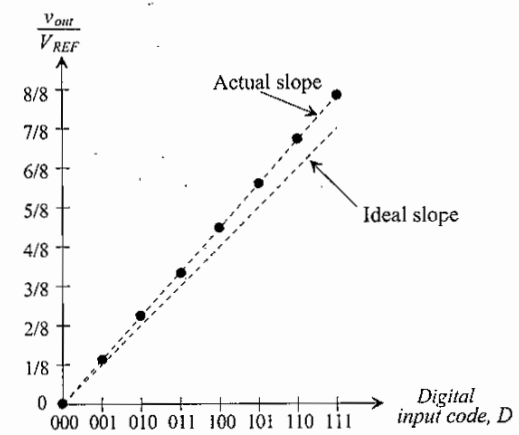


Figure 28.17 Illustration of gain error for a 3-bit DAC.

sinewave input. For the DAC, a "digital" sinewave is generated through instrumentation or through an A/D. The SNR can reveal the true resolution of a data converter as the effective number of bits can be quantified mathematically. A detailed derivation of the SNR is presented in Sec. 28.6, on the discussion of ADC specifications.

Dynamic Range

Dynamic range is defined as the ratio of the largest output signal over the smallest output signal. For both DACs and ADCs, the dynamic range is related to the resolution of the converter. For example, an N-bit DAC can produce a maximum output of $2^N - 1$ multiples of LSBs and a minimum value of 1 LSB. Therefore, the dynamic range in decibels is simply

$$DR = 20\text{Log}\left(\frac{2^N - 1}{1}\right) \approx 6.02 \cdot N \text{ dB} \qquad (28.15)$$

A 16-bit data converter has a dynamic range of 96.33 dB.

28.5 Analog-to-Digital Converter (ADC) Specifications

Many of the specifications that describe the ADC are similar to those that describe the DAC. However, there are subtle differences. Since the DAC is converting a discrete signal into an analog representation that is also limited by the resolution of the converter, a fixed number of inputs and outputs are generated. However, with the ADC, the input is an analog signal with an infinite number of values, which then has to be quantized into an N-bit digital word (Fig. 28.18). This process is much more difficult than the digital-to-analog process. In fact, many ADC architectures use a DAC as a critical component.

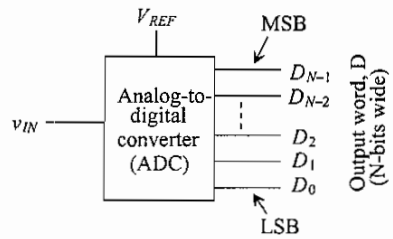


Figure 28.18 Block diagram of the analog-to-digital converter.

For example, in the previous discussion of DACs, it was determined that for a 16-bit DAC, the converter would need to generate output voltages in multiples of 76 μV. However, for the ADC, the converter needs to resolve differences in the analog signal of 76 μV. This means that the ADC must be able to detect changes in the input signal on the order of 1 part in 65,536! In contrast, the DAC had a finite number of input combinations (2^N). The ADC, however, has to "quantize" the infinite-valued analog signal into many segments so that

$$\text{Number of quantization levels} = 2^N \qquad (28.16)$$

This distinction is subtle but must be recognized to understand the differences between the two types of conversion.

Examine Fig. 28.19a. The digital output, D, of an ideal, 3-bit ADC is plotted versus the analog input, v_{IN}. Note the difference in the transfer curve for the ADC versus the DAC (Fig. 28.10). The y-axis is now the digital output, and the x-axis has been normalized to V_{REF}. Since the input signal is a continuous signal and the output is discrete, the transfer curve of the ADC resembles that of a staircase. Another fact to observe is that the 2^N quantization levels correspond to the digital output codes 0 to 7. Thus, the maximum output of the ADC will be 111 ($2^N - 1$), corresponding to the value for which $\frac{v_{IN}}{V_{REF}} \geq \frac{7}{8}$. Figure 28.19b corresponds to the error caused by the quantization.

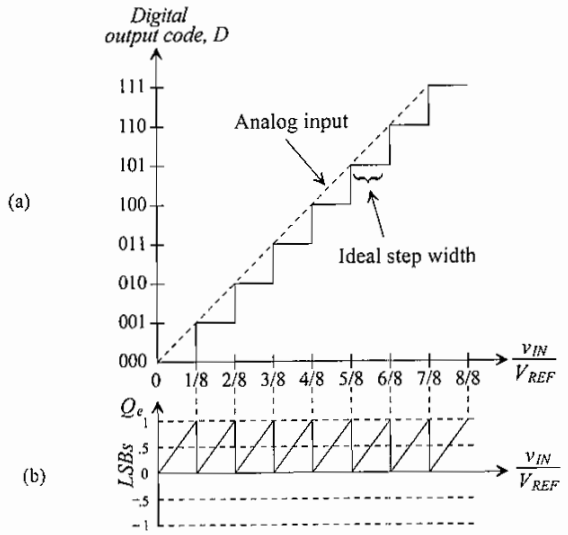


Figure 28.19 (a) Transfer curve for an ideal ADC and (b) its corresponding quantization error.

The value of 1 LSB for this ADC can be calculated using Eq. (28.9) and is the ideal step width (1/8) in Fig. 28.19 (versus the height for the DAC) multiplied by V_{REF}. Therefore, assuming that $V_{REF} = 5$ V,

$$1 \text{ LSB} = 0.625 \text{ V} \tag{28.17}$$

Quantization Error

Since the analog input is an infinite valued quantity and the output is a discrete value, an error will be produced as a result of the quantization. This error, known as *quantization error*, Q_e, is defined as the difference between the actual analog input and the value of the output (staircase) given in voltage. It is calculated as

$$Q_e = v_{IN} - V_{staircase} \tag{28.18}$$

where the value of the staircase output, $V_{staircase}$, can be calculated by

$$V_{staircase} = D \cdot \frac{V_{REF}}{2^N} = D \cdot V_{LSB} \tag{28.19}$$

where D is the value of the digital output code and V_{LSB} is the value of 1 LSB in volts, in this case 0.625 V. We can also easily convert the value of Q_e in units of LSBs. In Fig. 28.19a, Q_e can be generated by subtracting the value of the staircase from the dashed line. The result can be seen in Fig. 28.19b. A sawtooth waveform is formed centered about ½ LSBs. Ideally, the magnitude of Q_e will be no greater than one LSB and no less than 0. It would be advantageous if the quantization error were centered about zero so that the error would be at most ±½ LSBs (as opposed to +1LSB). This is easily achieved as seen in Fig. 28.20a and b. Here, the entire transfer curve is shifted to the left by ½ LSB, thus making the codes centered around the LSB increments on the x-axis. This drawing illustrates that at best, an ideal ADC will have quantization error of ±½ LSB.

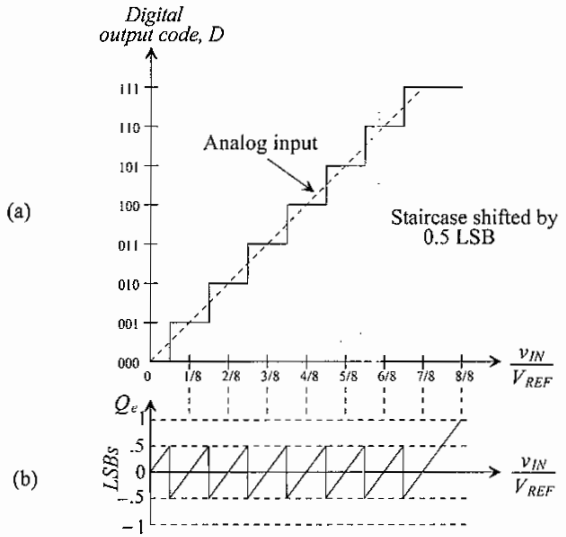


Figure 28.20 (a) Transfer curve for an ideal 3-bit ADC with (b) quantization error centered about zero.

In shifting this curve to the left, notice that the first code transition occurs when $\frac{v_{IN}}{V_{REF}} \geq \frac{1}{16}$. Therefore, the range of $\frac{v_{IN}}{V_{REF}}$ for the digital output corresponding to 000 is half as wide as the ideal step width. The last code transition occurs when $\frac{v_{IN}}{V_{REF}} \geq \frac{13}{16}$ (between 6/8 and 7/8). Note that the step width corresponding to this last code transition is 1.5 times larger than the ideal width and that the quantization error extends up to 1 LSB when $\frac{v_{IN}}{V_{REF}} = 1$. However, the converter would be considered to be out of range once $\frac{v_{IN}}{V_{REF}} \geq \frac{15}{16}$ (halfway between 7/8 and 8/8), so the problem is moot.

Differential Nonlinearity

Differential nonlinearity for an ADC is similar to that defined for a DAC. However, for the ADC, DNL is the difference between the actual code *width* of a nonideal converter and the ideal case. Figure 28.21 illustrates the transfer curve for a nonideal 3-bit ADC. The values for the DNL can be solved as follows:

$$\text{DNL} = \text{Actual step width} - \text{Ideal step width} \qquad (28.20)$$

Since the step widths can be converted to either volts for LSBs, DNL can be defined using either units. The value of the ideal step is 1/8. Converting to volts, this becomes

$$V_{idealstepwidth} = \tfrac{1}{8} \cdot V_{REF} = 0.625\ V = 1\ \text{LSB} \qquad (28.21)$$

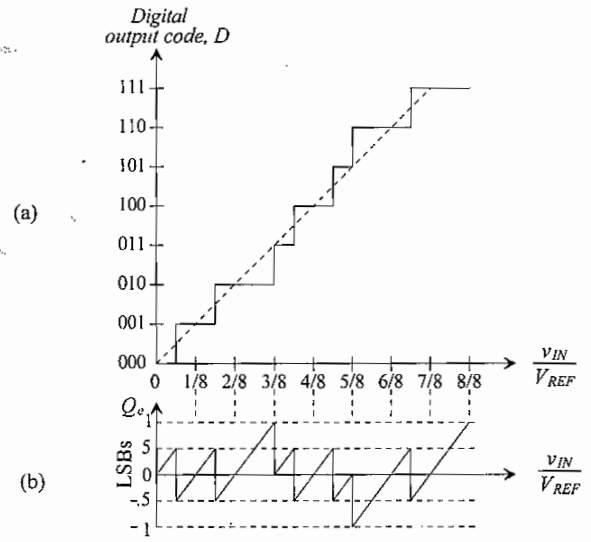


Figure 28.21 (a) Transfer curve for a nonideal 3-bit ADC used in Ex. 28.4 with (b) quantization error illustrating differential nonlinearity.

Example 28.6

Using Fig. 28.21a, calculate the differential nonlinearity of the 3-bit ADC. Assume that $V_{REF} = 5$ V. Draw the quantization error, Q_e, in units of LSBs.

The DNL of the converter can be calculated by examining the step width of each digital output code. Since the ideal step width of the 000 transition is ½ LSB, then $\text{DNL}_0 = 0$. Also note that the step widths associated with 001 and 011 are equal to 1 LSB; therefore, both DNL_1 and DNL_4 are zero. However, the remaining values code widths are not equal to the ideal value but can be calculated as

$\text{DNL}_2 = 1.5\ \text{LSB} - 1\ \text{LSB} = 0.5\ \text{LSB}$

$\text{DNL}_3 = 0.5\ \text{LSB} - 1\ \text{LSB} = -0.5\ \text{LSB}$

$\text{DNL}_5 = -0.5\ \text{LSB}$

$\text{DNL}_6 = 0.5\ \text{LSB}$

$\text{DNL}_7 = 0\ \text{LSB}$ (since the ideal step width is 1.5 LSB wide at this code transition)

The overall DNL for the converter used in this illustration is ±0.5 LSB. Note that the quantization error illustrated in Fig. 28.21b is directly related to the DNL. As DNL increases in either direction, the quantization error worsens. Each "tooth" in the quantization error waveform should ideally be the same size. ■

Missing Codes

It is of interest to note the consequences of having a DNL that is equal to –1 LSB. Figure 28.22 illustrates an ADC for which this is true. The total width of the step corresponding to 101 is completely missing; thus, the value of DNL_5 is –1 LSB. Any ADC possessing a DNL that is equal to –1 LSB is guaranteed to have a missing code. Notice that the step width corresponding to 010 is 2 LSBs and that the value for DNL_3 is +1 LSB. However, there is not a missing code corresponding to 011, since the step width of code 011 depends on the 100 transition. Therefore, an ADC having a DNL greater than +1 LSB is not guaranteed to have a missing code, though in all probability a missing code will occur.

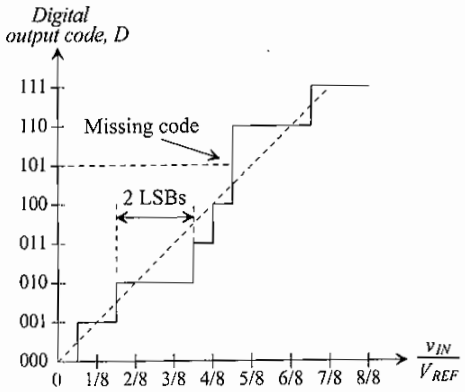


Figure 28.22 Transfer curve for a nonideal 3-bit ADC with a missing code.

Integral Nonlinearity

Integral nonlinearity (INL) is defined similarly to that for a DAC. Again, a "best-fit" straight line is drawn through the end points of the first and last code transition, with INL being defined as the difference between the data converter code transition points and the straight line with all other errors set to zero.

Example 28.7

Determine the INL for the ADC whose transfer curve is illustrated in Fig. 28.23a. Assume that V_{REF} = 5 V. Draw the quantization error, Q_e, in units of LSBs.

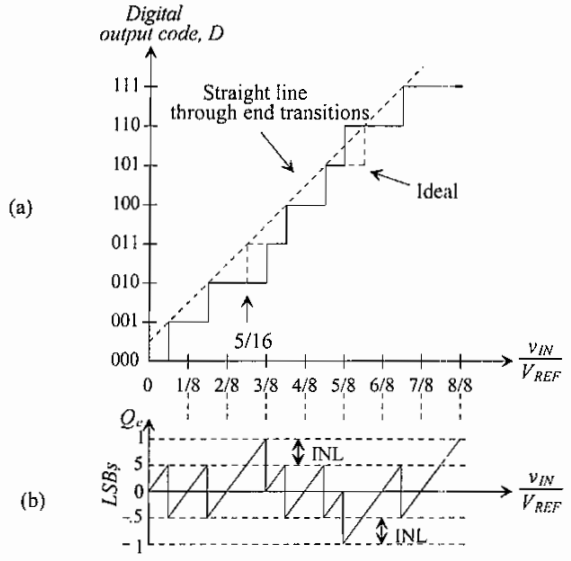


Figure 28.23 (a) Transfer curve of a nonideal 3-bit ADC and (b) its quantization error illustrating INL.

By inspection, it can be seen that all of the transition points occur on the best-fit line except for the transitions associated with code 011 and 110. Therefore,

$$INL_0 = INL_1 = INL_2 = INL_4 = INL_5 = INL_7 = 0$$

The INL corresponding to the remaining codes can be calculated as

$$INL_3 = 3/8 - 5/16 = 1/16 \text{ or } 0.5 \text{ LSB}$$

Similarly, INL_6 can be calculated in the same manner and is found to be –0.5 LSB. Thus, the overall INL for the converter is the maximum value of INL corresponding to ±0.5 LSB.

The INL can also be determined by inspecting the quantization error in Fig. 28.23b. Here, the INL will be the magnitude of the quantization error which lies outside the ±½ LSB band of Q_e. It can be seen that Q_e = 1 LSB, corresponding to the point at which INL = 0.5 LSB for digital output code 011, and that Q_e = –1 LSB at the output code corresponding to INL = –0.5 LSB for digital output code 110. ■

Offset and Gain Error

Offset and gain error are identical to the DAC case. *Offset error* occurs when there is a difference between the value of the first code transition and the ideal value of ½ LSBs. As seen in Fig. 28.24a, the offset error is a constant value. Note that the quantization error becomes ideal after the initial offset voltage is overcome. *Gain error* or *scale factor error*, seen in Fig. 28.24b, is the difference in the slope of a straight line drawn through the transfer characteristic and the slope of 1 of an ideal ADC. Causes of offset and gain error are discussed in Ch. 29, but it is important here to understand their overall effects on ADC transfer curves.

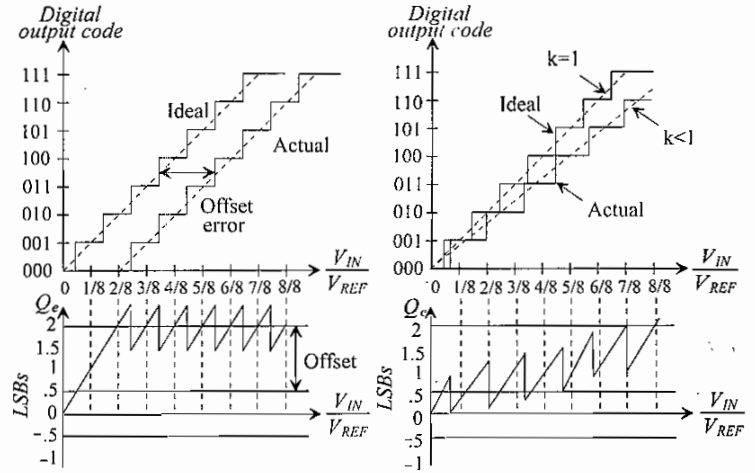


Figure 28.24 Transfer curve illustrating (a) offset error and (b) gain error.

So far, we have examined only the DC characteristics of an ADC. However, examining the dynamic aspects of the converter will lead to a whole new set of errors. Sampling is inherently a dynamic process since the accuracy of the sample is dependent on the speed of the analog signal. Many effects that occur during sampling limit the overall performance of the converter.

Aliasing

As mentioned earlier in the chapter, the Nyquist Criterion requires that a signal be sampled at least two times the highest frequency contained in the signal. What would happen if this criterion were ignored and the sampling rate was actually less than that amount? A phenomenon known as aliasing would occur.

Examine Fig. 28.25. Here, an analog signal is being sampled at a rate slower than the Nyquist Criterion requires. As a result, it appears that a totally different signal (see

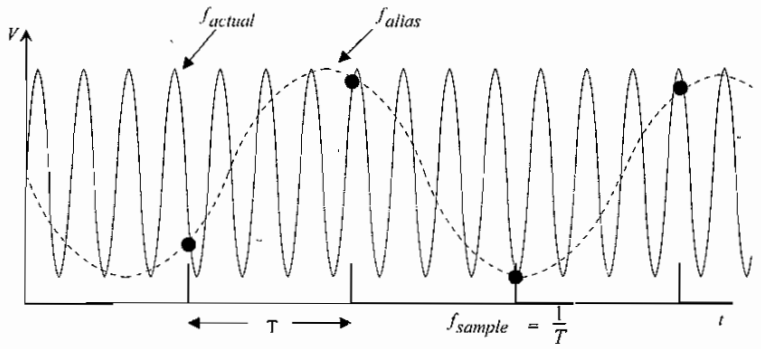


Figure 28.25 Aliasing caused by undersampling.

example dashed line) is being sampled. The different frequency signal is an "alias" of the original signal, and its frequency can be calculated using

$$f_{alias} = f_{actual} + k f_{sample} \quad (k = \ldots -2, -1, 0, 1, 2, 3\ldots) \qquad (28.22)$$

where f_{actual} is the frequency of the analog signal, f_{sample} is the sampling frequency, and f_{alias} is the frequency of the alias signal

Aliasing can be eliminated by both sampling at higher frequencies and by filtering the analog signal before sampling and removing any frequencies that are greater than one-half the sampling frequency. It is good practice to filter the analog signal before sampling to eliminate any unknown higher order frequency components or noise that could result in aliasing.

A frequency domain analysis may further illustrate the concepts of aliasing. Figure 28.26 shows the analog signal, the *sampling function* (represented by a unit impulse train) and the resulting sampled signal in both the time and frequency domains. The analog signal in Fig. 28.26a is represented as a simple band-limited signal with center frequency, f_o. This simply means that the signal is contained within the frequency range shown. In Fig. 28.26b, the sampling function is shown in both the time and frequency domain. The sampling function simply represents the action of sampling at discrete points in time. The frequency domain version of the sampling function is similar to its time domain counterpart, except that the x-axis is now represented as $f = 1/T$. Since each of the impulses has a value of 1, the resulting sampled signal shown in Fig. 28.26c is the impulse function multiplied by the amplitude of the analog signal at each discrete point in time. Remembering that multiplication in the time domain is equivalent to convolution in the frequency domain, we note that the frequency domain representation of the sampled signal reveals that the overall signal consists of multiple versions of the band-limited signal at multiples of the sampling frequency.

Note in Fig. 28.26b that as the sampling time increases, the sampling frequency decreases and the impulses in the frequency domain become more closely spaced. This results in 28.26d, which illustrates the aliasing as the multiple versions of the

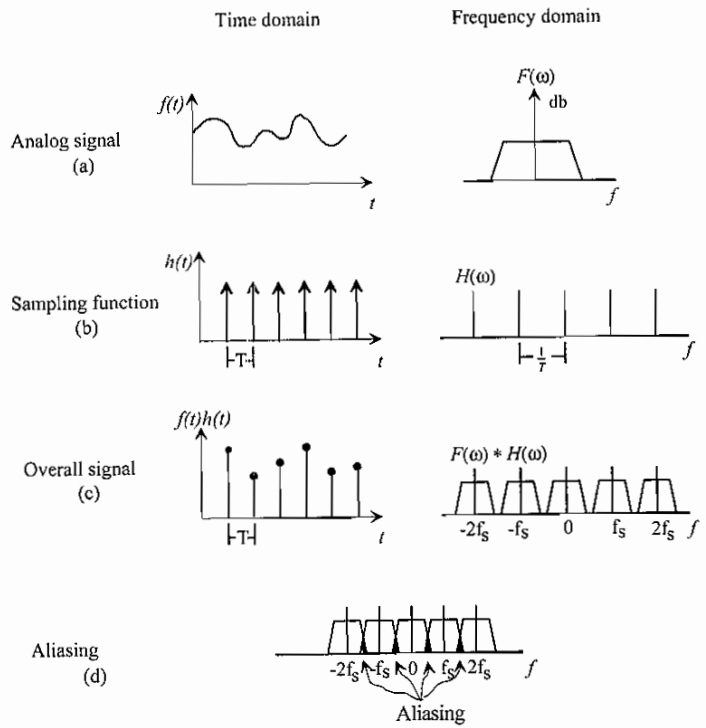


Figure 28.26 Illustration of aliasing in the time and frequency domain. (a) The analog signal; (b) the sampling function; (c) the overall signal; and (d) aliasing in the frequency domain.

band-limited signal begin to overlap. One could also filter the signal "post-sampling" and eliminate the frequencies for which overlap occurs. The point at which the spectra overlap is called the *folding frequency*.

As mentioned earlier, the solutions to aliasing are higher sampling frequency and filtering. Focusing on just one of the solutions may worsen the situation for several reasons. Some noise signals are wide band, which means that they have a large bandwidth. Attempting to increase only the sampling frequency to eliminate the aliasing effects of the noise would be a practical impossibility, not to mention a costly one. However, simply filtering the input signal and the sampled signal adds delays to the overall conversion and increases the expense of the circuit. It is best to use a combination of the two to minimize the problem most efficiently.

Signal-to-Noise Ratio

Signal-to-noise (SNR) ratios of ADCs represent the value of the largest RMS input signal into the converter over the RMS value of the noise. Typically given in dB, the expression for SNR is

$$SNR = 20\text{Log}(\frac{v_{in(max)}}{v_{noise}}) \tag{28.23}$$

If it is assumed that the input signal is a sinewave with a peak-to-peak value equal to the full-scale reference voltage of the converter, then the RMS value for $v_{in(max)}$ becomes

$$v_{in(max)} = \frac{V_{REF}}{2\sqrt{2}} = \frac{2^N(V_{LSB})}{2\sqrt{2}} \tag{28.24}$$

where V_{LSB} is the voltage value of 1 LSB. The value of the noise (if the data converter is considered to be ideal) will be equivalent to the RMS value of the error signal, Q_e (in volts), shown in Fig. 28.20b. The RMS value of Q_e can be calculated to be

$$Q_{e,RMS} = \left[\frac{1}{V_{LSB}}\int_{-0.5VLSB}^{0.5VLSB}(V_{LSB})^2 dV_{LSB}\right]^{0.5} = \frac{V_{LSB}}{\sqrt{12}} \tag{28.25}$$

Therefore, the SNR for the ideal ADC will be the ratio of these two RMS values,

$$SNR = 20\cdot\text{Log}\frac{\frac{2^N(V_{LSB})}{2\sqrt{2}}}{Q_{e,RMS}} \tag{28.26}$$

which can be written in terms of N as simply

$$SNR = 20N\text{Log}(2) + 20\text{Log}\sqrt{12} - 20\text{Log}(2\sqrt{2}) = 6.02N + 1.76 \tag{28.27}$$

Equation (28.27) is an important one relating SNR to the resolution of the ADC. For 16-bit data conversion, one must design a circuit that will have an SNR of (6.02)(16) + 1.76 = 98.08 dB! Equation (28.27) can also be used in calculating the *signal-to-noise plus distortion ratio*, also known as *SNDR*. Since the output data is digital, we cannot use a spectrum analyzer to calculate this ratio but must instead use a *Discrete Fourier Transform* (*DFT*) and examine the data in the digital domain.

Another useful application of Eq. (28.27) is the determination of effective number of bits given a system with a known SNR or SNDR. For example, if a 16-bit ADC yielded an SNDR of 88 dB, then the effective resolution of the converter would be

$$N = \frac{88 - 1.76}{6.02} = 14.32 \text{ bits} \tag{28.28}$$

and the ADC would be producing the resolution equivalent to that of a 14-bit converter.

Aperture Error

The aperture error described in Sec. 28.3 (S/H) should be related to the errors associated with the ADC. In the previous discussion, the aperture error resulted in sampling error (Fig. 28.8). However, now that ADC characteristics have been discussed, we can relate the sampling error to the ADC. Since we know that the maximum errors associated with an ADC are related to ½ LSB, we can assume that the maximum sampling error associated with the aperture uncertainty can be no larger than ½ LSB.

Example 28.8

Find the maximum resolution of an ADC which can use the S/H described in Ex. 28.1 while maintaining a sampling error less than ½ LSB.

Since it was determined that the maximum sampling error produced by the given aperture uncertainty was 0.628 mV, we can relate this value to the highest resolution of an ADC by assuming that 0.628 mV will be less than or equal ½ LSB. Therefore,

$$0.628 \text{ mV} \le .5\ LSB = \frac{V_{REF}}{2^{N+1}} = \frac{5}{2^{N+1}}$$

or

$$2^{N+1} \le 7961.8$$

which, solving for N (limited to an integer), yields a maximum resolution of 11 bits. ■

28.6 Mixed-Signal Layout Issues

Naturally, analog ICs are more sensitive to noise than digital ICs. For any analog design to be successful, careful attention must be paid to layout issues, particularly in a digital environment. Sensitive analog nodes must be protected and shielded from any potential noise sources. Grounding and power supply routing must also be considered when using digital and analog circuitry on the same substrate. Since a majority of ADCs use switches controlled by digital signals, separate routing channels must be provided for each type of signal.

Techniques used to increase the success of mixed-signal designs vary in complexity and priority. Strategies regarding the systemwide minimization of noise should always be considered foremost. A mixed-signal layout strategy can be modeled as seen in Fig. 28.27. The lowest issues are foundational and must be considered before each succeeding step. The successful mixed-signal design will always minimize the effect of the digital switching on the analog circuits.

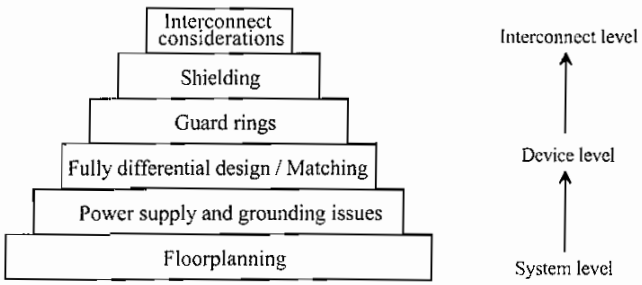


Figure 28.27 Mixed-signal layout strategy.

Floorplanning

The placement of sensitive analog components can greatly affect a circuit's performance. Many issues must be considered. In designing a mixed-signal system, strategies regarding the "floorplan" of the circuitry should be thoroughly analyzed well before the layout is to begin.

The analog circuitry should be categorized by the sensitivity of the analog signal to noise. For example, low-level signals or high-impedance nodes typically associated with input signals are considered to be sensitive nodes. These signals should be closely guarded and shielded, especially from digital output buffers. High-swing analog circuits such as comparators and output buffer amplifiers should be placed between the sensitive analog and the digital circuitry.

The digital circuitry should also be categorized by speed and function. Obviously, since digital output buffers are usually designed to drive capacitive loads at very high rates, they should be kept farthest from the sensitive analog signals. Next, the high and lower speed digital should be placed between the insensitive analog and the output buffers. An example of this type of strategy can be seen in Fig. 28.28 [6]. Notice that *the sensitive analog is as far away as possible from the digital output buffers* and that the least sensitive analog circuitry is next to the least offensive digital circuitry.

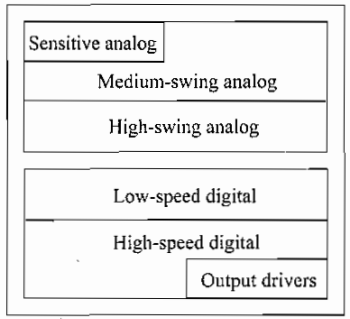


Figure 28.28 Example of a mixed-signal floorplan [6].

Power Supply and Grounding Issues

Whenever analog and digital circuit reside together on the same die, danger exists of injecting noise from the digital system to the sensitive analog circuitry through the power supply and ground connections. Much of the intercoupling can be minimized by carefully considering how power and ground are supplied to both the analog and digital circuits.

In Fig. 28.28a analog and digital circuitry share the same routing to a single pad for power and ground. The resistors, R_{l1} and R_{l2}, represent the small, nonnegligible resistance of the interconnect to the pad. The inductors, L_{B1} and L_{B2}, represent the inductance of the bonding wire which connects the pads to the pin on the lead frame.

Since digital circuitry is typified by high amounts of transient currents due to switching, a small amount of resistance associated with the interconnect can result in significant voltage spikes. Low-level analog signals are very sensitive to such interference, thus resulting in a contaminated analog system. Another significant voltage spike can occur due to the inductance of the bonding wire. Since the voltage across the inductor is proportional to the change in current through it, voltage spikes equating to hundreds of millivolts can result! Both of these voltage effects are true for both the power and ground connection.

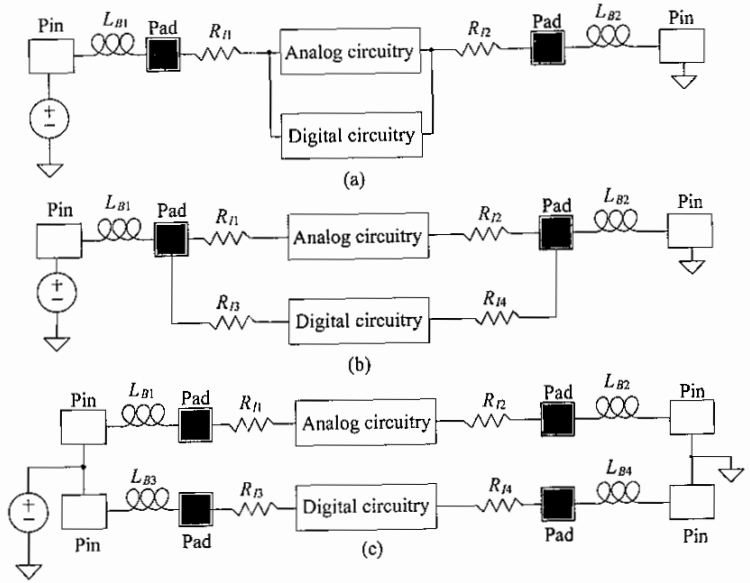


Figure 28.29 Power and ground connection examples, (a) poor noise immunity and (b) better noise immunity, and (c) using separate power and ground pins to achieve even better immunity.

One way to reduce the interference, seen in Fig. 28.29b, is to prohibit the analog and digital circuit from sharing the same interconnect. The routing for the supply and ground for both the analog and digital sections are provided separately. Although this eliminates the parasitic resistance due to the common interconnect, there is still a common inductance due to the bonding wire which causes interference.

Another method that minimizes interference even more than the previous case is seen in Fig. 28.29c. *By using separate pads and pins, the analog and the digital circuits are completely decoupled.* The current through the analog interconnect is much less abrupt than the digital; thus, the analog circuitry now has a "quiet" power and ground. However, this technique depends on whether extra pins and pads are available for this

use. The separate power supply and ground pins are then connected externally. *It is not wise to use two separate power supplies because if both types of circuits are not powered up simultaneously, latch-up could easily result.*

In cases presented in Fig. 28.29b and c, *the resistance associated with the analog connection to ground or supply can be reduced by making the power supply and ground bus as wide as feasible.* This reduces the overall resistance of the metal run, thus decreasing the voltage spikes that occur across the resistor. The inductor itself is impossible to eliminate, though it can be minimized with careful planning. Since the length of the bonding wire depends on the distance from the pad to the lead frame, *one could reduce the effect of the wire inductance by reserving pins closest to the die for sensitive connections such as analog supply and ground.* This, again, illustrates the importance of floorplanning.

Fully Differential Design

Fully differential operational amplifiers were discussed earlier, Fig. 28.30. The noise sources represent the noise from digital circuitry coupled through the parasitic stray capacitors. If equal amounts of noise are injected into the differential amplifiers, then the common-mode rejection inherent in the amplifiers will eliminate most or all of the noise. This, of course, depends on the symmetry of the amplifiers, meaning that matching the transistors in the amplifier becomes crucial. Therefore, *in a mixed-signal environment, layout techniques should be used to improve matching such as common-centroid and interdigitated techniques.*

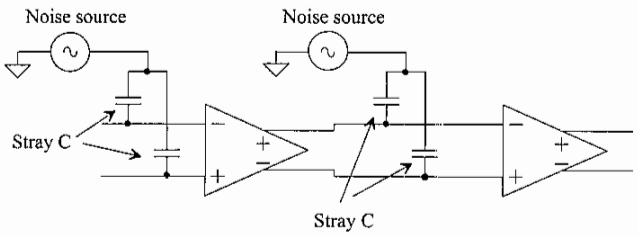


Figure 28.30 Differential output op-amps showing parasitic coupling to noise sources.

Guard Rings

Guard rings should be used wisely throughout a mixed-signal environment. Circuits that process sensitive signals should be placed in a separate well (if possible) with guard rings attached to the analog *VDD* supply. In the case of an n-well (only) process, the n-type devices outside the well should have guard rings attached to analog ground placed around them. Digital circuits should be placed in their own well with guard rings attached to digital *VDD*. Guard rings placed around the n-channel digital devices also help minimize the amount of noise transmitted from the digital devices.

Shielding

A number of techniques exist which can shield sensitive, low-level analog signals from noise resulting from digital switching. A shield can take the form of a layer tied to analog ground placed between two other layers, or it can be a barrier between two signals running in parallel.

If at all possible, one should avoid crossing sensitive analog signals, such as low-level analog input signals, with any digital signals. The parasitic capacitance coupling the two signal lines can be as much as a couple of fF, depending on the process. If it cannot be avoided, then attempt to carry the digital signal using the top layer of metal (such as metal2). If the analog signal is an input signal, then it will most likely be carried by the poly layer (or a lower level of metal). A strip of metal1 can be placed between the two layers and connected to analog ground (see Fig. 28.31).

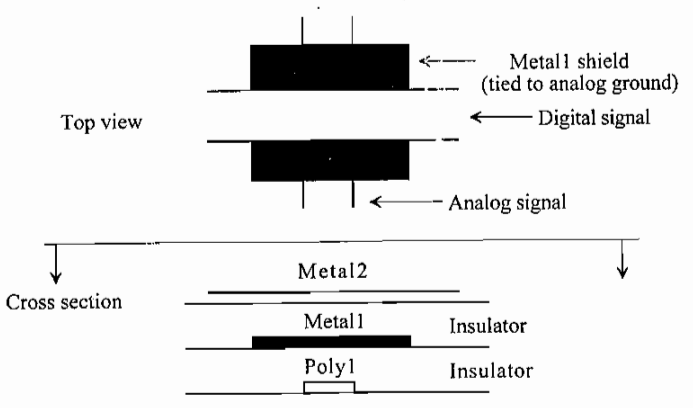


Figure 28.31 Shielding a sensitive analog signal from a digital signal crossover using a metal 1 shield layer.

Another situation that should be avoided is running interconnect containing sensitive analog signals parallel and adjacent to any interconnect carrying digital signals. Coupling occurs due to the parasitic capacitance between the lines. If this situation cannot be avoided, then an additional line connected to analog ground should be placed between the two signals, as seen in Fig. 28.32. This method can also be used to partition the analog and digital sections of the chip.

In addition, the n-well can be used as a bottom-plate shield to protect analog signals from substrate noise. Poly resistors (or capacitors) used for sensitive analog signals can be shielded by placing an n-well beneath the components and connecting the well to analog *VDD*.

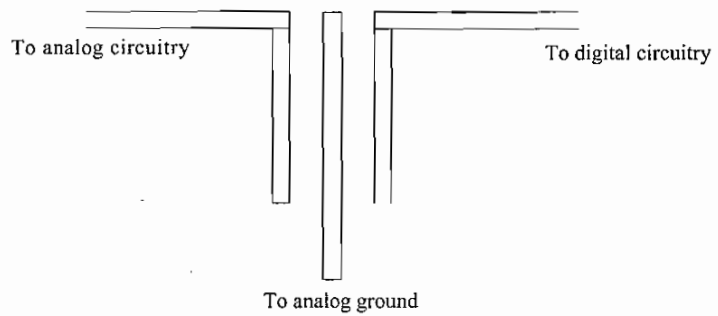


Figure 28.32 Using a dummy metal strip to provide shielding to two parallel signals.

Other Interconnect Considerations

Finally, some other layout strategies will incrementally improve the performance of the analog circuitry. However, if the previous strategies are not followed, these suggestions will be useless. *When routing the analog circuitry, minimize the lengths of current carrying paths.* This will simply reduce the amount of voltage drop across the path due to the metal1 or metal2 resistance. *Contacts should also be used very liberally whenever changing layers.* Not only does this minimize resistance in the path, but it also improves fabrication reliablity. *Avoid using poly to route current carrying signal paths.* Not only is the poly higher in resistance value, but also the additional contact resistance required to change layers will not be insignificant. If the poly is made wider to lower the resistance, additional parasitic capacitance will be added to the node. *Use poly to route only high-impedance gate nodes that carry virtually no current.*

REFERENCES

[1] M. J. Demler, *High-Speed Analog-to-Digital Conversion*, Academic Press, 1991.

[2] B. Razavi, *Principles of Data Conversion System Design*, IEEE Press, 1995.

[3] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Co., 1990.

[4] D. H. Sheingold, *Analog-Digital Conversion Handbook*, Prentice-Hall Publishing, 1986.

[5] S. K. Tewksbury, et al., "Terminology Related to the Performance of S/H, A/D and D/A Circuits," *IEEE Transactions on Circuits and Systems*, CAS-25, vol. CAS-25, pp. 419–426, July 1978.

[6] Y. Tsividis, *Mixed Analog-Digital VLSI Devices and Technology: An Introduction*, McGraw-Hill Publishing Co., 1996.

PROBLEMS

28.1 Determine the number of quantization levels needed if one wanted to make a digital thermometer that was capable of measuring temperatures to within 0.1 °C accuracy over a range from –50 °C to 150 °C. What resolution of ADC would be required?

28.2 Using the same thermometer as above, what sampling rate, in samples per second, would be required if the temperature displayed a frequency of $15°\cdot\sin(0.01\cdot 2\pi t)$?

28.3 Determine the maximum droop allowed in an S/H used in a 16-bit ADC assuming that all other aspects of both the S/H and ADC are ideal. Assume $V_{ref} = 5$ V.

28.4 An S/H circuit settles to within 1 percent of its final value at 5 μs. What is the maximum resolution and speed with which an ADC can use this data assuming that the ADC is ideal?

28.5 A digitally programmable signal generator uses a 14-bit DAC with a 10-volt reference to generate a DC output voltage. What is the smallest incremental change at the output that can occur? What is the DAC's full-scale value? What is its accuracy?

28.6 Determine the maximum DNL (in LSBs) for a 3-bit DAC, which has the following characteristics. Does the DAC have 3-bit accuracy? If not, what is the resolution of the DAC having this characteristic?

Digital Input	Voltage Output
000	0 V
001	0.625 V
010	1.5625 V
011	2.0 V
100	2.5 V
101	3.125 V
110	3.4375 V
111	4.375 V

28.7 Repeat Problem 28.6 calculating the INL (in LSBs).

28.8 A DAC has a reference voltage of 1,000 V, and its maximum INL measures 2.5 mV. What is the maximum resolution of the converter assuming that all the other characteristics of the converter are ideal?

28.9 Determine the INL and DNL for a DAC that has a transfer curve shown in Fig. 28.33.

28.10 A DAC has a full-scale voltage of 4.97 V using a 5 V reference, and its minimum output voltage is limited by the value of one LSB. Determine the resolution and dynamic range of the converter.

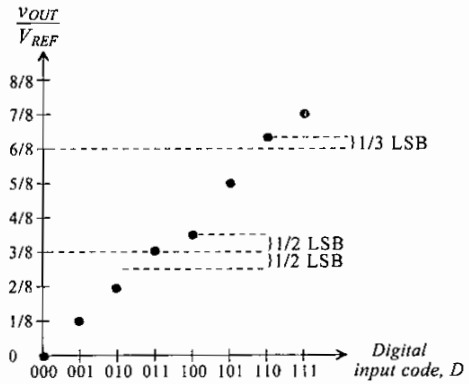


Figure 28.33 Transfer curves for Problem 28.9.

28.11 Prove that the RMS value of the quantization noise shown in Fig. 28.20b is as stated in Eq. (28.26).

28.12 An ADC has a stated SNR of 94 dB. Determine the effective number of bits of resolution of the converter.

28.13 Discuss the methods used to prevent aliasing and the advantages and disadvantages of each.

Chapter
29

Data Converter Architectures

Applications such as wireless communications and digital audio and video have created the need for cost-effective data converters that will achieve higher speed and resolution. The needs required by digital signal processors continually challenge analog designers to improve and develop new ADC and DAC architectures. There are many different types of architectures, each with unique characteristics and different limitations. This chapter presents a basic overview of the more popular data converter architectures and discusses the advantages and disadvantages of each along with their limitations.

Now that we have defined the operating characteristics of ADCs in Ch. 28, a more detailed examination of the basic architectures will be discussed using a top-down approach. Because many of the converters use op-amps, comparators, and resistor and capacitor arrays, the top-down approach will allow a broader discussion of the key component limitations in later sections.

29.1 DAC Architectures

A wide variety of DAC architectures exist, ranging from very simple to complex. Each, of course, has its own merits. Some use voltage division, whereas others employ current steering and even charge scaling to map the digital value into an analog quantity.

29.1.1 Digital Input Code

In many cases, the digital signal is not provided in binary code but is any one of a number of codes: binary, BCD, thermometer code, Gray code, sign-magnitude, two's complement, offset binary, and so on. (See Fig. 29.1 for a comparison of some of the more commonly used digital input codes.) For example, it may be desirable to allow only one bit to change value when changing from one code to the next. If that is the case, a Gray code will suffice. The thermometer code is used quite frequently and can also be seen in Fig. 29.1. Notice that it requires $2^N - 1$ bits to represent an N-bit word. The choice of code depends on the application, and the reader should be aware that many types of codes are available.

Decimal	Binary	Thermometer	Gray	Two's Complement
0	000	0000000	000	000
1	001	0000001	001	111
2	010	0000011	011	110
3	011	0000111	010	101
4	100	0001111	110	100
5	101	0011111	111	011
6	110	0111111	101	010
7	111	1111111	100	001

Figure 29.1 Comparison of digital input codes.

29.1.2 Resistor String

The most basic DAC is seen in Fig. 29.2a. Comprised of a simple resistor string of 2^N identical resistors and switches, the analog output is simply the voltage division of the resistors at the selected tap. Note that a N:2^N decoder is required to provide the 2^N signals controlling the switches. This architecture typically results in good accuracy, provided that no output current is required and that the values of the resistors are within the specified error tolerance of the converter. One big advantage of a resistor string is that the output is always guaranteed to be monotonic.

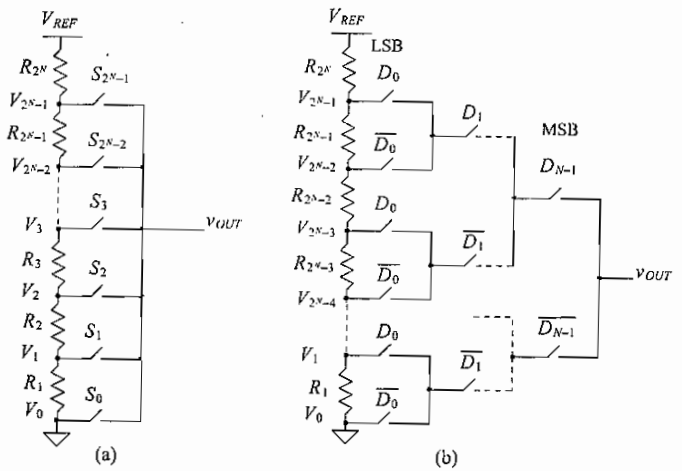


Figure 29.2 (a) A simple resistor-string DAC and (b) the use of a binary switch array to lower the output capacitance.

One problem with this converter is that the converter output is always connected to $2^N - 1$ switches that are off and one switch that is on. For larger resolutions, a large parasitic capacitance appears at the output node, resulting in slower conversion speeds. A better alternative for the resistor-string DAC is seen in Fig. 29.2b. Here, a binary switch array ensures that the output is connected to at most N switches that are on and N switches that are off, thus increasing conversion speed. The input to this switch array is a binary word since the decoding is inherent in the binary tree arrangement of the switches.

Another problem with the resistor-string DAC is the balance between area and power dissipation. An integrated version of this converter leads to a large chip area for higher bit resolutions because of the large number of passive components needed. Active resistors such as the n-well resistor can be used for low-resolution applications. However, as the resolution increases, the relative accuracy of the resistors becomes an important factor. Although the value of R could always be made small to minimize the chip area required, power dissipation would then become the critical issue as current flows through the resistor string at all times.

Example 29.1

Design a 3-bit resistor-string ladder using a binary switch array. Assume that V_{REF} = 5 V and that the maximum power dissipation of the converter is to be 5 mW (not including the power required by the digital logic). Determine the value of the analog voltage for each of the possible digital input codes.

The power dissipation will determine the current flowing through the resistor string by

$$I_{MAX} = \frac{5 \times 10^{-3}\ \text{W}}{5\ \text{V}} = 1\ \text{mA}$$

Since a 3-bit converter will have eight resistors, the value of R is

$$R = \frac{1}{8} \cdot \frac{5\ \text{V}}{1\ \text{mA}} = 625\ \Omega$$

The converter can be seen in Fig. 29.3. Examine the switch array if the input code is $D_2D_1D_0$ = 100 or 4_{10}. Since D_2 is high, the top switch will be closed and the lower switch, $\overline{D_2}$, will be open. In the row corresponding to D_1, since $D_1 = 0$, both of the switches marked $\overline{D_1}$ will be closed and the other two will be open. The LSB controls the largest number of switches; therefore, since D_0 is low, all of the $\overline{D_0}$ switches will be closed and all of the D_0 switches will be open. There should be only one path connecting a single tap on the resistor string to the output. This is bolded, with the resistor string tapped in the middle of the string. Therefore, $v_{OUT} = \frac{1}{2} V_{REF} = 2.5$ V. The remaining outputs can be seen in Fig. 29.4. ■

Mismatch Errors Related to the Resistor-String DAC

The accuracy of the resistor string is obviously related to matching between the resistors, which ultimately determines the INL and DNL for the entire DAC. Suppose that the i-th resistor, R_i, has a mismatch error associated with it so that

$$R_i = R + \Delta R_i \qquad (29.1)$$

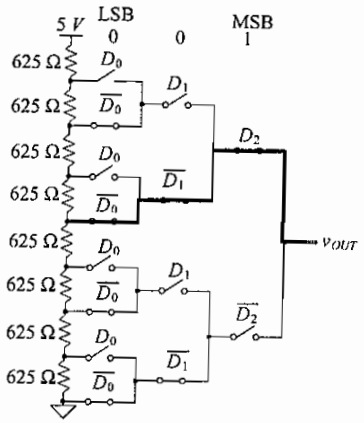


Figure 29.3 A 3-bit resistor-string DAC used in Ex. 29.1

where R is the ideal value of the resistor and ΔR_i is the mismatch error. Also suppose that the mismatches were symmetrical about the string so that the sum of all the mismatch terms were zero, or

$$\sum_{i=1}^{2^N} \Delta R_i = 0 \tag{29.2}$$

The value of the voltage at the tap associated with the i-th resistor should ideally be

$$V_{i,ideal} = \frac{(i)V_{REF}}{2^N}, \text{ for } i = 0, 1, 2, \ldots, 2^N - 1 \tag{29.3}$$

$D_2 D_1 D_0$	v_{OUT}
000	0
001	0.625
010	1.25
011	1.875
100	2.5
101	3.125
110	3.75
111	4.375

Figure 29.4 Output voltages generated from the 3-bit DAC in Ex. 29.1.

However, including the mismatch, the actual value of the i-th voltage is the sum of all of the resistances up to and including resistor i, divided by the sum of all of the resistances in the string. This can be represented by

$$V_i = V_{REF} \cdot \frac{\sum_{k=1}^{i} R_k}{\sum_{k=1}^{2^N} R_k} = V_{REF} \cdot \frac{\sum_{k=1}^{i} R + \Delta R_k}{2^N R} \tag{29.4}$$

The denominator does not include any mismatch error since it was assumed that the mismatches sum to zero as defined in Eq. (29.2). Notice that there is no resistor, R_0, corresponding to V_0 (see Fig. 29.2), and it is assumed that V_0 is ground. Equation (29.4) can be rewritten as

$$V_i = \frac{V_{REF}}{2^N R}\left[(i)R + \sum_{k=1}^{i} \Delta R_k\right] = \frac{(i)V_{REF}}{2^N} + \frac{V_{REF}}{2^N R}\sum_{k=1}^{i} \Delta R_k \tag{29.5}$$

or finally, the value of the voltage at the i-th tap is

$$V_i = V_{i,ideal} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} \tag{29.6}$$

Equation (29.6) is not of much importance by itself, but it can be used to help determine the nonlinearity errors.

Integral Nonlinearity of the Resistor-String DAC

Integral nonlinearity (INL) is defined as the difference between the actual and ideal switching points, or

$$INL = V_i - V_{i,ideal} \tag{29.7}$$

and plugging in Eqs. (29.6) and (29.3) into (29.7) yields,

$$INL = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} \tag{29.8}$$

Equation (29.8) is a general expression for the INL for a given resistor, R_i, and requires that the mismatch of all the resistances used in the summation are known. However, this equation does not illuminate how to determine the *worst-case* or maximum INL for a resistor string.

Intuitively, one would think that the worst-case INL would occur at the top of the resistor string (i=2^N) with all of the ΔR_k's at their maximum values. However, the previous derivation was performed with the assumption that the mismatches summed to zero. With this restriction, the maximum INL occurs at the midpoint of the string where $i = 2^{N-1}$, corresponding to the case where the MSB was a one and all other bits were zero. Another condition that will ensure a worst-case scenario is to consider the lower half resistors at their maximum positive mismatch value and the upper half resistors at their maximum negative mismatch value, or vice versa.

If the resistors on a string were known to have 2% matching, then ΔR_k would be constrained to the bounds of

$$-0.02R \le \Delta R_k \le 0.02R \tag{29.9}$$

and the worst-case INL (again, % matching = 0.02) using Eq. (29.8) would be,

$$|INL|_{max} = \frac{V_{REF}}{2^N}\cdot\sum_{k=1}^{2^{N-1}}\frac{\Delta R_k}{R} = \frac{V_{REF}}{2^N}\cdot\frac{2^{N-1}\cdot\Delta R_k}{R} = \frac{1}{2}\,LSB\cdot 2^N\cdot(\%\text{ matching}) = 0.01\,V_{REF} \tag{29.10}$$

which for $INL < 0.5\ LSB$ requires $1/2^N >$ (% matching). For 2% matching, the maximum number of bits, N, is then 5! For better than 0.2% matching $N = 9$ bits.

Because the worst-case analysis was performed, the maximum INL occurs at the middle of the string. We can improve this specification on paper by using the "best-fit" approach to measuring INL. In this case, the reference line is simply shifted up slightly (refer to Ch. 28) so that it no longer passes through the end points, but instead minimizes the INL.

Example 29.2

Determine the effective number of bits for a resistor-string DAC, which is assumed to be limited by the INL. The resistors are passive poly resistors with a known relative matching of 1%, and $V_{REF} = 5$ V.

Using Eq. (29.10), the maximum INL will be

$$|INL|_{max} = 0.005\cdot V_{REF} = 0.025\text{ V}$$

Since we know that this maximum INL should be equal to ½ LSB in the worst case,

$$\frac{1}{2}\,LSB = \frac{5}{2^{N+1}} = 0.025\text{ V}$$

and solving for N yields

$$N = \log_2\left(\frac{5}{0.025}\right) - 1 = 6.64\text{ bits}$$

This means that the resolution for a DAC containing a resistor string matched to within 1% will be, at most 6 bits. ■

Differential Nonlinearity of the Worst-Case Resistor-String DAC

Resistor-string matching is not as critical when determining the DNL. Remembering that the definition of DNL is simply the actual height of the stair-step in the DAC transfer curve minus the ideal step height, we can write this in terms of the voltages at the taps of adjacent resistors on the string. Using Eq. (29.5), we can express this as,

$$|V_i - V_{i-1}| = \left|\left[\frac{(i)V_{REF}}{2^N} + \frac{V_{REF}}{2^N}\cdot\sum_{k=1}^{i}\frac{\Delta R_k}{R}\right] - \left[\frac{(i-1)V_{REF}}{2^N} + \frac{V_{REF}}{2^N}\cdot\sum_{k=1}^{i-1}\frac{\Delta R_k}{R}\right]\right|$$

which can be simplified to

$$|V_i - V_{i-1}| = \left|\frac{V_{REF}}{2^N}\left(1 + \frac{\Delta R_i}{R}\right)\right| \tag{29.11}$$

The DNL can then be determined by subtracting the ideal step height from Eq. (29.11),

$$DNL_i = \left|\frac{V_{REF}}{2^N}\left(1 + \frac{\Delta R_i}{R}\right) - \frac{V_{REF}}{2^N}\right| = \left|\frac{V_{REF}}{2^N}\cdot\frac{\Delta R_i}{R}\right| \tag{29.12}$$

and the maximum DNL will occur at the value of i for which ΔR is at its maximum value. If it is assumed once again that the resistors are matched to within 2 percent, the worst-case DNL will be

$$DNL_{max} = \left|0.02\cdot\frac{V_{REF}}{2^N}\right| = 0.02\ LSB \tag{29.13}$$

which is well below the ½ LSB limit. The INL is obviously the limiting factor in determining the resolution of a resistor-string DAC as its maximum value is 2^N times larger than the DNL.

29.1.3 *R-2R* Ladder Networks

Another DAC architecture that incorporates fewer resistors is called the *R-2R* ladder network [1]. This configuration consists of a network of resistors alternating in value of R and $2R$. Figure 29.5 illustrates an N-bit *R-2R* ladder. Starting at the right end of the network, notice that the resistance looking to the right of any node to ground is $2R$. The digital input determines whether each resistor is switched to ground (noninverting input) or to the inverting input of the op-amp. Each node voltage is related to V_{REF}, by a binary-weighted relationship caused by the voltage division of the ladder network. The total current flowing from V_{REF} is constant, since the potential at the bottom of each switched resistor is always zero volts (either ground or virtual ground). Therefore, the node voltages remain constant for any value of the digital input.

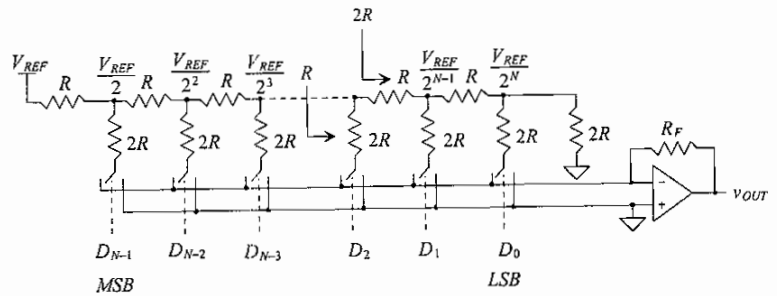


Figure 29.5 An *R-2R* digital-to-analog converter.

The output voltage, v_{OUT}, depends on currents flowing through the feedback resistor, R_F, such that

$$v_{OUT} = -i_{TOT}\cdot R_F \tag{29.14}$$

where i_{TOT} is the sum of the currents selected by the digital input by

$$i_{TOT} = \sum_{k=0}^{N-1} D_k\cdot\frac{V_{REF}}{2^{N-k}}\cdot\frac{1}{2R} \tag{29.15}$$

where D_k is the k-th bit of the input word with a value that is either a 1 or a 0.

This architecture, like the resistor-string architecture, requires matching to within the resolution of the converter. Therefore, the switch resistance must be negligible, or a small voltage drop will occur across each switch, resulting in an error. One way to eliminate this problem is to add dummy switches. Assume that the resistance of each switch connected to the $2R$ resistors is ΔR, as seen in Fig. 29.6. Dummy switches with one-half the resistance of the real switches are "hard-wired" so that they are always on and placed in series with each of the horizontal resistors. The total resistance of any horizontal branch, R', is

$$R' = R + \frac{\Delta R}{2} \tag{29.16}$$

The resistance of any vertical branch is $2R + \Delta R$, which is twice the value of the horizontal branch. Therefore, a $R' - 2R'$ relationship is maintained. Of course, a dummy switch equal to the switch size of a $2R$ switch will have to be placed in series with the terminating resistor as well.

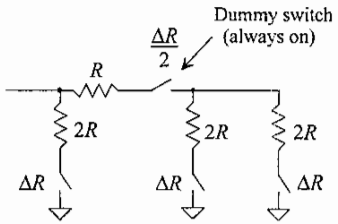


Figure 29.6 Use of dummy switches to offset switch resistance.

Example 29.3

Design a 3-bit DAC using an R-$2R$ architecture with R = 1 kΩ, R_F = 2 kΩ, and V_{REF} = 5 V. Assume that the resistances of the switches are negligible. Determine the value of i_{TOT} for each digital input and the corresponding output voltage, v_{OUT}.

Figure 29.7 shows the 3-bit DAC for a digital input of 001. The voltages at each node in the resistor network are labeled. For each switch, if the digital input bit is a zero, then the resistor is attached to the ground. If the bit is a one, then the resistor is attached to the virtual ground of the inverting input and current flows to the output of the op-amp. Therefore, for $D_2D_1D_0$ = 000, all of the switches are connected to ground, no current flows through the feedback resistor, and the output voltage, v_{OUT}, is zero.

When $D_2D_1D_0$ = 001, the rightmost resistor is switched to the op-amp inverting input and the other two resistors remain attached to ground. Therefore, the total current flowing through the feedback resistor is simply the current through the rightmost resistor, which is defined by Eq. (29.15) as

$$\frac{V_{REF}}{8} \cdot \frac{1}{2000} = 0.3126 \text{ mA}$$

and the output voltage, by Eq. (29.14) becomes,

$$v_{OUT} = -(0.3126 \text{ mA})(2000\ \Omega) = -0.625 \text{ V}$$

which is to be expected. The other values for the output voltage can be calculated using Eqs. (29.14) and (29.15) and are seen in Fig. 29.8. ■

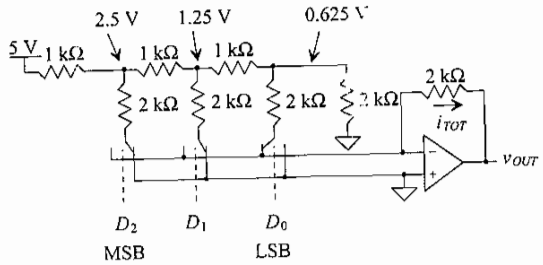


Figure 29.7 A 3-bit R-$2R$ digital-to-analog converter used in Ex. 29.3.

$D_2D_1D_0$	i_{TOT} (mA)	v_{OUT} (V)
000	0	0
001	0.3125	−0.625
010	0.625	−1.25
011	0.625 + 0.3125 = 0.9375	−1.875
100	1.25	−2.5
101	1.25 + 0.3125 = 1.5625	−3.125
110	1.25 + 0.625 = 1.875	−3.75
111	1.25 + 0.625 + 0.3125 = 2.1875	−4.375

Figure 29.8 Output voltages generated from the 3-bit DAC in Example 29.3.

29.1.4 Current Steering

In the previous section, a voltage was converted into a current, which then generated a voltage at the output. Another DAC method uses current throughout the conversion. Known as *current steering*, this type of DAC requires precision current sources that are summed in various fashions.

Figure 29.9 illustrates a generic current-steering DAC. This configuration requires a set of current sources, each having a unit value of current, I. Since there are no current sources generating i_{OUT} when all the digital inputs are zero, the MSB, D_{2^N-2}, is offset by two index positions instead of one. For example, for a 3-bit converter, seven current sources will be needed, labeled from D_0 to D_6. The binary signal controls whether or not the current sources are connected to either i_{OUT} or some other summing node (in this case ground). The output current, i_{OUT}, has the range of

$$0 \le i_{OUT} \le (2^N - 1) \cdot I \tag{29.17}$$

and can be any integer multiple of I in between. An interesting issue to note is the format of the digital code required to drive the switches. Since there are $2^N - 1$ current sources, the digital input will be in the form of a *thermometer code*. This code will be all ones from the LSB up to the value of the k-th bit, D_k, and all zeros above it. The point at which the input code changes from all ones to all zeros "floats" up or down and resembles the action of a thermometer, hence the name. Typically, a thermometer encoder is used to convert binary input data into a thermometer code.

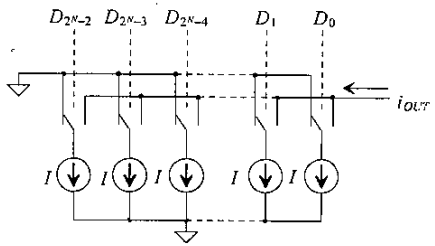


Figure 29.9 A generic current-steering DAC.

Another current-steering architecture is seen in Fig. 29.10. This architecture uses binary-weighted current sources, thus requiring only N current sources of various sizes versus $2^N - 1$ sources in the previous example. Since the current sources are binary weighted, the input code can be a simple binary number with no thermometer encoder needed.

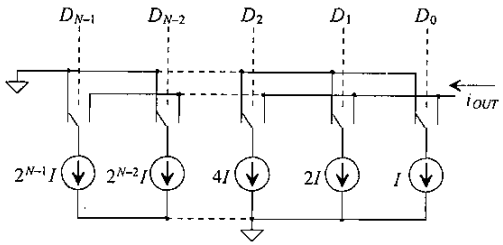


Figure 29.10 A current-steering DAC using binary-weighted current sources.

One advantage of the current-steering DACs is the high-current drive inherent in the system. Since no output buffers are necessary to drive resistive loads, these DACs are typically used in high-speed applications. Traditionally, high-speed current-steering DACs have been fabricated using bipolar technology. However, the ability to generate matched current mirrors makes CMOS an enticing alternative. Of course, the precision needed to generate high resolutions depends on how well the current sources can be matched or the degree to which they can be made binary weighted. For example, if a 13-bit DAC was designed using these architectures, there would have to be 8,191 current sources resident on the chip, not an insignificant amount. For the binary-weighted sources, only 13 current sources would be needed. Yet the size of the largest current source would have to be 4,096 or 2^{N-1} times larger than the smallest. Even if the unit current, I, was chosen to be 5 μA, the largest current source would be 20.48 mA!

Another problem associated with this architecture is the error due to the switching. Since the current sources are in parallel, if one of the current sources is switched off and another is switched on, a "glitch" could occur in the output if the timing was such that both of them were on or both were off for an instant. While this may not seem significant, if the converter is switching from 0111111 to 10000000, the output will spike toward ground and then back to the correct value if all the switches turn off for an instant. If the DAC is driving a resistive load and the output current is converted to a voltage, a substantial voltage spike will occur at the output.

Example 29.4

Construct a table showing the thermometer code necessary to generate the output shown in Fig. 29.11a for a 3-bit current-steering DAC using unit current sources.

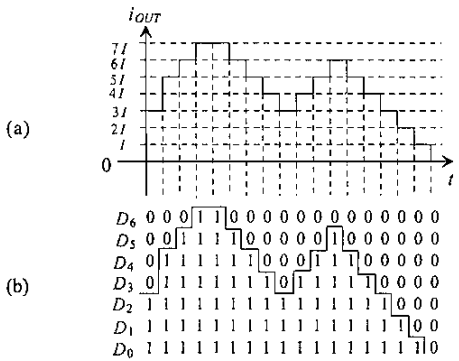


Figure 29.11 (a) Output of a 3-bit current-steering DAC and (b) the thermometer code input.

The thermometer code can be seen in Fig. 29.11b. When the code is all zeros, the output is 0 volts. Therefore, only 7 bits are needed to represent the 2^N or 8 states of a 3-bit DAC. Note how the interface between all ones and all zeros actually resembles the output signal itself. ■

Mismatch Errors Related to Current-Steering DACs

Analysis of the mismatch associated with the current sources is similar to the resistor string analysis. It is assumed that each current source in Fig. 29.9 is

$$I_k = I + \Delta I_k \text{ for } k = 1, 2, 3, \ldots, 2^N - 1 \tag{29.18}$$

where I is the ideal value of the current and ΔI_k is the error due to mismatch. If it is again assumed that the ΔI_K terms sum to zero and that one-half of the current sources contain the maximum positive mismatch, ΔI_{max}, and the other half contains the maximum negative mismatch, $-\Delta I_{max}$, (or vice versa), then the worst-case condition will occur at midscale with the actual output current being

$$I_{out} = \sum_{k=1}^{2^{N-1}} (I + \Delta I_k) = 2^{N-1} \cdot I + 2^{N-1} \cdot |\Delta I|_{max} = I_{out,ideal} + 2^{N-1} \cdot |\Delta I|_{max} \tag{29.19}$$

Since the INL is simply the actual output current minus the ideal, the worst-case INL is

$$|INL|_{max} = 2^{N-1} \cdot |\Delta I|_{max,INL} \tag{29.20}$$

The term, $|\Delta I|_{max,INL}$ represents the maximum current source mismatch error that will keep the INL less than ½ LSB. Each current source represents the value of 1 LSB; therefore, ½ LSB is equal to 0.5 I. Because the maximum INL should correspond to the ½ LSB, equating Eq. (29.20) to ½ I results in the value for $|\Delta I|_{max,INL}$,

$$|\Delta I|_{max,INL} = \frac{0.5I}{2^{N-1}} = \frac{I}{2^N} \tag{29.21}$$

Equation (29.21) illustrates the difficulty of using this architecture at high resolutions. If the value of I is set to be 5 μA, and the N is desired to be 12 bits, then

$$|\Delta I|_{max,INL} = \frac{5 \times 10^{-6}}{2^{12}} = 1.221 \text{ nA!} \tag{29.22}$$

which means that each of the 5 μA current sources must lie between the bounds of,

$$4.99878\ \mu A \le I_k \le 5.001221\ \mu A \tag{29.23}$$

to achieve a worst-case INL, which is within ½ LSB error.

The DNL is easily obtained since the step height in the transfer curve is equivalent to the value of the ideal current source, I. The maximum difference between any two adjacent values of output current will simply be the value of the single source, I_k, which contains the largest mismatch error for which the DNL will be less than ½ LSB, $|\Delta I|_{max,DNL}$:

$$I_{out(x)} - I_{out(x-1)} = I_k + |\Delta I|_{max,DNL} \tag{29.24}$$

Therefore, the DNL is simply

$$|DNL|_{max} = I_k + |\Delta I|_{max,DNL} - I_k = |\Delta I|_{max,DNL} \tag{29.25}$$

Equating the maximum DNL to the value of ½ LSB,

$$|\Delta I|_{max,DNL} = \tfrac{1}{2}\, LSB = \tfrac{1}{2}\, I \tag{29.26}$$

which is much easier to attain than the requirement for the INL.

For the binary-weighted current sources seen in Fig. 29.10, a slightly different analysis is needed to determine the requirements for INL and DNL. In this case, it will be assumed that the current source corresponding to the MSB (D_{N-1}) has a maximum positive mismatch error value and that the remainder of the bits (D_0 to D_{N-2}) contain a maximum negative mismatch error, so that the sum of all the errors equals zero. The INL is

$$|INL|_{max} = 2^{N-1}(I + |\Delta I|_{max,INL}) - 2^{N-1} \cdot I = 2^{N-1} \cdot |\Delta I|_{max,INL} \tag{29.27}$$

which is equivalent to the value of the current steering array in Fig. 29.9.

The DNL is slightly different because of the binary weighting of the current sources. One cannot add a single current source with each incremental increase in the digital input code. However, the worst-case condition for binary-weighted arrays tends to occur at midscale when the code transitions from 011111....111 to 100000....000. The worst-case DNL at this point is

$$DNL_{max} = \left[2^{N-1} \cdot (I + |\Delta I|_{max,DNL}) - \sum_{k=1}^{N-1} 2^{k-1} \cdot (I - |\Delta I|_{max,DNL}) \right] - I \tag{29.28}$$

which can be written as

$$DNL_{max} = 2^{N-1} \cdot (I + |\Delta I|_{max,DNL}) - (2^{N-1} - 1) \cdot (I - |\Delta I|_{max,DNL}) - I = (2^N - 1) \cdot |\Delta I|_{max,DNL} \tag{29.29}$$

and setting this value equal to ½ LSB and solving for ΔI_{max},

$$|\Delta I|_{max,DNL} = \frac{0.5I}{2^N - 1} = \frac{I}{2^{N+1} - 2} \tag{29.30}$$

Therefore, the DNL requirements for the binary-weighted current source array is more stringent than the INL requirements.

One interesting issue regarding the previous derivation is that the challenging accuracy requirements in Eq. (29.30) are placed only on the MSB current source. For each of the remaining binary-weighted sources, the DNL requirements become more relaxed. This is simply because the size of the MSB source is equivalent to all of the other sources combined, and so its value plays the most important role in the DAC's accuracy.

Example 29.5

Determine the tolerance of the MSB current source on a 10-bit binary-weighted current source array with a unit current source of 1 μA, which will result in a worst-case DNL that is less than ½ LSB.

Since Eq. (29.30) defines the maximum $|\Delta I|$ needed to keep the DNL less than ½ LSB, we must first use this equation,

$$|\Delta I|_{max,DNL} = \frac{1 \times 10^{-6}}{2^{11} - 2} = 0.4888 \text{ nA}$$

For a 10-bit DAC, the MSB current source will have a value that is 2^9 times larger than the unit current source, or 0.512 mA. Therefore, the range of values for which this array will have a DNL that is less than ½ LSB is

$$0.51199995 \text{ mA} \le I_{MSB} \le 0.5120004888 \text{ mA} \quad \blacksquare$$

29.1.5 Charge-Scaling DACs

A very popular DAC architecture used in CMOS technology is the charge-scaling DAC. Shown in Fig. 29.12a, a parallel array of binary-weighted capacitors, totaling $2^N C$, is connected to an op-amp. The value, C, is a unit capacitance of any value. After initially being discharged, the digital signal switches each capacitor to either V_{REF} or ground, causing the output voltage, v_{OUT}, to be a function of the voltage division between the capacitors.

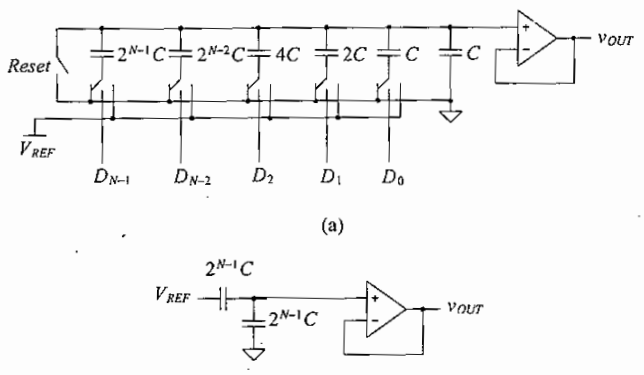


Figure 29.12 (a) A charge-scaling DAC, (b) the equivalent circuit with the MSB = 1, and all other bits set to zero.

The capacitor array totals $2^N C$. Therefore, if the MSB is high and the remaining bits are low, then a voltage divider occurs between the MSB capacitor and the rest of the array. The analog output voltage, v_{OUT}, becomes

$$v_{OUT} = V_{REF} \cdot \frac{2^{N-1}C}{(2^{N-1}+2^{N-2}+2^{N-3}+\ldots+4+2+1+1)C} = V_{REF} \cdot \frac{2^{N-1}C}{2^N C} = \frac{V_{REF}}{2} \tag{29.31}$$

which confirms the fact that the MSB changes the output of a DAC by ½ V_{REF}. Figure 29.12b shows the equivalent circuit under this condition. The ratio between v_{OUT} and V_{REF} due to each capacitor can be generalized to

$$v_{OUT} = \frac{2^k C}{2^N C} \cdot V_{REF} = 2^{k-N} \cdot V_{REF} \tag{29.32}$$

where it is assumed that the k-th bit, D_k, is one and all other bits are zero. Superposition can then be used to find the value of v_{OUT} for any digital input word by

$$v_{OUT} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{REF} \tag{29.33}$$

One limitation of this architecture as shown in Fig. 29.12a is the existence of a parasitic capacitance at the top plate of the capacitor array due to the op-amp. This will prohibit its use as a high-resolution data converter. A better implementation would include the use of a parasitic insensitive, switched-capacitor integrator (see Ch. 25) as the driving circuit. However, the capacitor array itself is the critical component of this data converter and is used in charge redistribution ADCs (Sec. 29.2.5).

The INL and DNL calculations for the binary-weighted capacitor array are identical to those for the binary-weighted current source array, except that the unit current source, I, and its corresponding error term, ΔI, are replaced by C and ΔC in Eqs. (29.27) – (29.30).

Example 29.6

Design a 3-bit charge-scaling DAC and find the value of the output voltage for $D_2D_1D_0 = 010$ and 101. Assume that $V_{REF} = 5$ V and $C = 0.5$ pF.

The 3-bit DAC can be seen in Fig. 29.13a. The equivalent circuits for the capacitor array can be seen in Fig. 29.13b and c. The value of the output voltage can be calculated by either using Eq. (29.32) or the equivalent circuits and performing the voltage division. For $D = 010$, the equivalent circuit in Fig. 29.13b yields

$$v_{OUT} = V_{REF} \cdot \left(\frac{1}{4}\right) = 1.25 \text{ V}$$

Using Eq. (29.33) to calculate v_{OUT} for $D = 101$ yields

$$v_{OUT} = \sum_{k=0}^{N-1} D_k 2^{k-N} \cdot V_{REF} = [1 \cdot (2^{-3}) + 0 \cdot (2^{-2}) + 1 \cdot (2^{-1})] \cdot 5 = \left(\frac{1}{8}+\frac{1}{2}\right) \cdot 5 = 3.125 \text{ V}$$

which is the result expected. ■

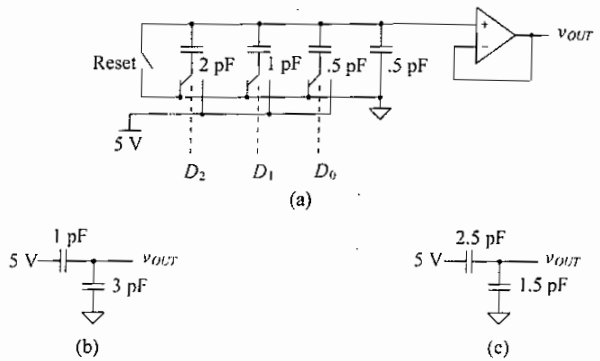


Figure 29.13 (a) A 3-bit charge-scaling DAC used in Ex. 29.6 and the equivalent circuits inputs equal to (b) 010 (c) 101.

Layout Considerations for a Binary-Weighted Capacitor Array

One problem with this converter is the need for precisely ratioed capacitors. As the number of bits increase, the ratio of the MSB capacitor to the LSB capacitor becomes more difficult to control. For example, Fig. 29.14a shows a 3-bit binary capacitor array using three capacitors. When the capacitor is fabricated, *undercutting* of the mask [2, 3] causes an error in the ratio of the capacitors, creating potentially large DNL and INL errors as N increases.

One solution to this problem is seen in Fig. 20.14b. Here, each capacitor in the array is constructed out of a unit capacitance. Undercutting then affects all of the capacitors in the same way, and the ratio between capacitors is maintained. Another problem that affects even this layout strategy is a nonuniform oxide growth. Gradients result in errors in the ratios of the capacitors. Figure 29.14c illustrates another layout strategy that overcomes this issue. The capacitors are laid out in a common-centroid scheme so that the first-order oxide errors average out to be the same for each capacitor.

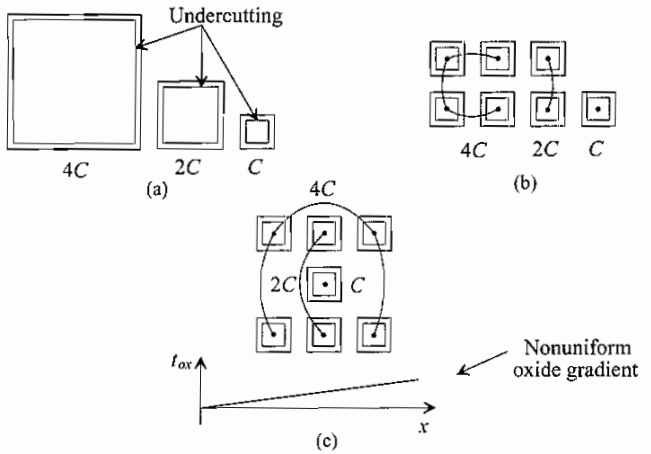


Figure 29.14 Layout of a binary-weighted capacitor array using (a) single capacitors (b) unit capacitors to minimize undercutting effect, and (c) common-centroid to minimize oxide gradients.

The Split Array

The charge-scaling architecture is very popular among CMOS designers because of its simplicity and relatively good accuracy. Although a linear capacitor is required using poly2, high resolutions in the 10- to 12-bit range can be achieved. Passive, double-poly capacitors have good matching accuracy as well. However, as the resolution increases, the size of the MSB capacitor becomes a major concern. For example if the unit capacitor, C,

were 0.5 pF, and a 16-bit DAC were to be designed, the MSB capacitor would need to be

$$C_{MSB} = 2^{N-1} \cdot 0.5 \text{ pF} = 16.384 \text{ nF} \tag{29.34}$$

If the capacitance between poly1 and poly2 is nominally 25 fF/μm², then the area required for this one capacitor is (roughly) 800 by 800 μm².

One method of reducing the size of the capacitors is to use a split array. A 6-bit example of the array is pictured in Fig. 29.15. This architecture [1] is slightly different from the charge-scaling DAC pictured in Fig. 29.13 in that the output is taken off a different node and an additional attenuation capacitor is used to separate the array into a LSB array and a MSB array. Note that the LSB, D_0, now corresponds to the leftmost switch and that the MSB, D_5, corresponds to the rightmost switch. The value of the attenuation capacitor can be found by

$$C_{atten} = \frac{sum\ of\ the\ LSB\ array\ capacitors}{sum\ of\ the\ MSB\ array\ capacitors} \cdot C \tag{29.35}$$

where the sum of the MSB array equals the sum of LSB capacitor array minus C. The value of the attenuation capacitor should be such that the series combination of the attenuation capacitor and the LSB array, assuming all bits are zero, equals C.

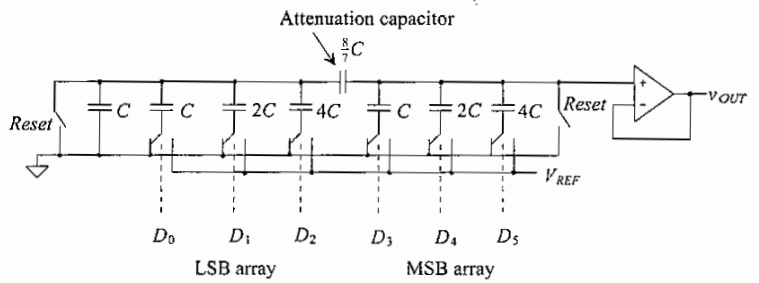


Figure 29.15 A charge-scaling DAC using a split array.

Example 29.7

Using the 6-bit charge-scaling DAC shown in Fig. 29.15, (a) show that the output voltage will be $\frac{1}{2} \cdot V_{REF}$ if (a) $D_5 D_4 D_3 D_2 D_1 D_0 = 100000$ and (b) the output will be $\frac{1}{64} \cdot V_{REF}$ if $D_5 D_4 D_3 D_2 D_1 D_0 = 000001$.

(a) If $D_5 = 1$ and the remaining bits are all zero, then the equivalent circuit for the DAC can be represented by Fig. 29.16a. The expression for the output voltage then becomes

$$v_{OUT} = \frac{4}{\left(\frac{8}{7} \text{ in series with } 8\right) + 3 + 4} \cdot V_{REF} = \frac{1}{2} \cdot V_{REF}$$

(b) For the second case, the equivalent circuit can be seen in Fig. 29.16b. The intermediate node voltage, V_A, is simply the voltage division between the C associated with D_0 and the remainder of the circuit, or

$$V_A = V_{REF} \cdot \frac{1}{\left(7 + \frac{\frac{8}{7}\cdot 7}{\frac{8}{7}+7}\right) + 1} = \frac{1}{8+\frac{56}{57}} \cdot V_{REF} \tag{29.36}$$

The output voltage can be written as

$$v_{OUT} = V_A \cdot \frac{\frac{8}{7}}{\frac{8}{7}+7} = \frac{8}{57} \cdot V_A \tag{29.37}$$

Plugging Eq. (29.36) into Eq. (29.37) yields

$$v_{OUT} = V_{REF} \cdot \frac{8}{(8\cdot 57)+56} = \frac{V_{REF}}{64} \tag{29.38}$$

which is the desired result. ■

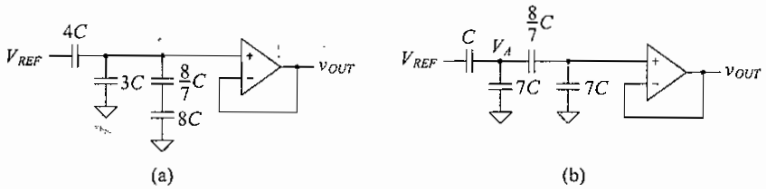


Figure 29.16 Equivalent circuits for Example 29.7.

29.1.6 Cyclic DAC

The cyclic DAC uses only a couple of simple components to perform the conversion. As seen in Fig. 29.17, a summer adds V_{REF} or ground to the feedback signal depending on the input bits. An amplifier with a gain of 0.5 feeds the output voltage back to the summer such that the output at the end of each cycle depends on the value of the output during the cycle before. Notice that the input bits must be read in a serial fashion. Therefore, the conversion is performed one bit at a time, resulting in N cycles required for each conversion. The voltage output at the end of the n-th cycle of the conversion can be written as

$$v_{OUT}(n) = \left(D_{n-1} \cdot V_{REF} + \frac{1}{2} \cdot v_A(n-1)\right) \cdot \frac{1}{2} \tag{29.39}$$

with a condition such that the output of the S/H is initially zero [$v_A(0) = 0$ V].

The accuracy of this converter is dependent on several factors. The gain of the 0.5 amplifier needs to be highly accurate (to within the accuracy of the DAC) and is usually generated with passive capacitors. Similarly, the summer and the sample-and-hold also need to be N-bit accurate. Limitations of the converter due to these fundamental building blocks will be discussed in more detail in Sec. 29.3. Since this converter uses a pseudo-"sampled-data" approach, implementing this architecture using switched capacitors is relatively easy.

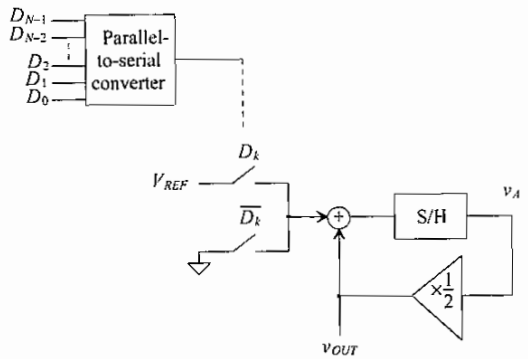


Figure 29.17 A cyclic digital-to-analog converter.

Example 29.8

Show the value of the output voltage at the end of each cycle for a 6-bit cyclic DAC with an input value of $D_5D_4D_3D_2D_1D_0 = 110101$. Assume that $V_{REF} = 5$ V.

We can predict the value of the output based on our previous experience with DACs. The digital input 110101 corresponds to 53_{10}. Therefore, the output voltage due to this input should be

$$v_{OUT} = \frac{53}{64} \cdot V_{REF} = 4.140625 \text{ V}$$

Now examine the cyclic converter in Fig. 29.17. By performing a 6-bit conversion and using Eq. (29.39), the outputs occurring at the end of each cycle can be seen in Fig. 29.18.

The output voltage at the end of the sixth cycle is precisely what was predicted. Note that had this been a 3-bit conversion, the output voltage at the end of cycle 3 would correspond to the value of the 3-bit DACs studied previously with an input of 101. ■

Cycle Number, n	D_{n-1}	$v_A(n-1)$	$v_{OUT}(n)$
1	1	0	½ (5 + 0) = 2.5 V
2	0	5	½ (0 + 2.5) = 1.25 V
3	1	2.5	½ (5 +1.25) = 3.125 V
4	0	6.25	½ (0 + 3.125) = 1.5625 V
5	1	3.125	½ (5 + 1.5625) = 3.28125 V
6	1	6.5625	½ (5 + 3.28125) = 4.140625 V

Figure 29.18 Output from the 6-bit cyclic DAC used in Ex. 29.8.

29.1.7 Pipeline DAC

The cyclic converter presented in the last section takes N clock cycles per N-bit conversion. Instead of recycling the output back to the input each time, we could extend the cyclic converter to N stages, where each stage performs one bit of the conversion. This extension of the cyclic converter is called a *pipeline* DAC and is seen in Fig. 29.19. Here, the signal is passed down the "pipeline," and as each stage works on one conversion, the previous stage can begin processing another. Therefore, an initial N clock cycle delay is experienced as the signal makes its way down the pipeline the very first time. After the N clock cycle delay, a conversion takes place at every clock cycle.

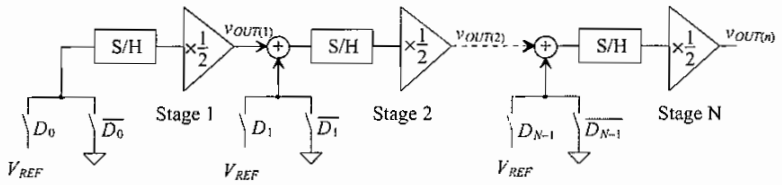


Figure 29.19 A pipeline digital-to-analog converter.

Besides the N clock cycle delay, this architecture can be very fast. However, the amplifier gains must be very accurate to produce high resolutions. Also, this architecture uses N times more circuitry than that of the cyclic, so there is a trade-off between speed and chip area. The output voltage of the n-th stage in the converter can be written as

$$v_{OUT(n)} = [D_{n-1} \cdot V_{REF} + v_{OUT(n-1)}] \cdot \frac{1}{2} \qquad (29.40)$$

The operation of each stage in the pipeline can be summarized as follows: if the input bit is a 1, add V_{REF} to the output of the previous stage, divide by two, and pass the value to the next stage. If the input bit is a 0, simply divide the output of the previous stage by two and pass along the resulting value.

Example 29.9

Find the output voltage for a 3-bit pipeline DAC for three cases: D_A = 001, D_B = 110, and D_C = 101. Show that the conversion time to perform all three conversions is five clock cycles using the pipeline approach. Assume that V_{REF} = 5 V.

The first stage operates on the LSBs of each word; the second stage operates on the middle bits; and the last stage, the MSBs. Based on the pipeline strategy, once the LSB of the first input word is performed and passed on, the LSB of the second word, D_B, can begin its conversion. Similarly, once the LSB of the second stage is completed and passed on, the LSB of the third word, D_C, can begin. The conversion cycle for all three input words produces the output shown in Fig. 29.20. The items that are in bold are associated with the first input word, D_A, whereas the italicized numbers represent the values associated with D_B and the underlined items, D_C.

Clock Cycle	$v_{OUT(1)}$	$v_{OUT(2)}$	$v_{OUT(3)}$	D_0	D_1	D_2
1	**2.5**	0	0	**1**	0	0
2	*0*	**1.25**	0	*0*	**0**	0
3	2.5	*2.5*	**0.625**	1	*1*	**0**
4		1.25	*3.75*		0	*1*
5			3.125			1

Figure 29.20 Output from the 3-bit pipeline DAC used in Example 29.9.

The first output of the DAC is not valid until the end of the third clock cycle and should look familiar as the 3-bit DAC output for an input word of $D_2 D_1 D_0$ = 001. The following two clock cycles that produce outputs for $D_2 D_1 D_0$ equal 110 and 101, respectively. ■

29.2 ADC Architectures

A survey of the field of current A/D converter research reveals that a majority of effort has been directed to four different types of architectures: pipeline, flash-type, successive approximation, and oversampled ADCs. Each has benefits that are unique to that architecture and span the spectrum of high speed and resolution.

Since the ADC has a continuous, infinite-valued signal as its input, the important analog points on the transfer curve x-axis for an ADC are the ones that correspond to changes in the digital output word. These input transitions determine the amount of INL and DNL associated with the converter.

29.2.1 Flash

Flash or parallel converters have the highest speed of any type of ADC. As seen in Fig. 29.21, they use one comparator per quantization level ($2^N - 1$) and 2^N resistors (a resistor-string DAC). The reference voltage is divided into 2^N values, each of which is fed into a comparator. The input voltage is compared with each reference value and results in a thermometer code at the output of the comparators. A thermometer code exhibits all zeros for each resistor level if the value of v_{IN} is less than the value on the resistor string, and ones if v_{IN} is greater than or equal to voltage on the resistor string. A simple $2^N - 1$:N digital thermometer decoder circuit converts the compared data into an N-bit digital word. The obvious advantage of this converter is the speed with which one conversion can take place. Each clock pulse generates an output digital word. The advantage of having high speed, however, is counterbalanced by the doubling of area with each bit of increased resolution. For example, an 8-bit converter requires 255 comparators, but a 9-bit ADC requires 511! Flash converters have traditionally been limited to 8-bit resolution with conversion speeds of 10–40 Ms/s using CMOS technology [4–6]. The disadvantages of the Flash ADC are the area and power requirements of the $2^N - 1$ comparators. The speed is limited by the switching of the comparators and the digital logic.

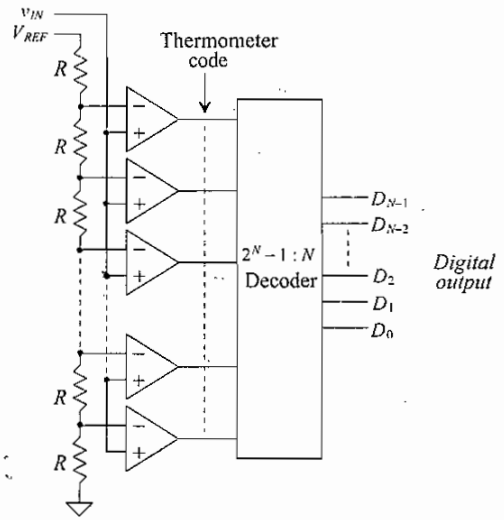


Figure 29.21 Block diagram of a Flash ADC.

Example 29.10

Design a 3-bit Flash converter, listing the values of the voltages at each resistor tap, and draw the transfer curve for $v_{IN} = 0$ to 5 V. Assume $V_{REF} = 5$ V. Construct a table listing the values of the thermometer code and the output of the decoder for $v_{IN} = 1.5$, 3.0, and 4.5 V.

The 3-bit converter can be seen in Fig. 29.22. As the values of all the resistors are equal, the voltage of each resistor tap, V_i, will be $V_i = V_{REF}\left(\frac{i}{8}\right)$ where i is the number of the resistor in the string for $i = 1$ to 7. Obviously, $V_1 = 0.625$ V, $V_2 = 1.25$ V, $V_3 = 1.875$ V, $V_4 = 2.5$ V, $V_5 = 3.125$, $V_6 = 3.75$ V, $V_7 = 4.375$ V. Therefore, when v_{IN} first becomes equal or greater than each of these values, a transition will occur in the transfer curve. The transfer curve can be seen in Fig. 29.23 and should look similar to those seen in Ch. 28. The quantization levels and their corresponding thermometer codes are summarized in Fig. 29.24.

The transfer curve of this ADC corresponds to the ADC with quantization error centered about +½ LSB, as discussed in Ch. 28 (Fig. 28.20). To shift the curve by ½ LSB so that the code transitions occur around the LSB values and the quantization error is centered around 0 LSB, the value of the last resistor in the string would have to be adjusted to $\frac{R}{2}$ and the value of the MSB resistor, closest to the reference voltage, would have to be made $1.5R$. Then the first code transition would occur at $v_{IN} = 0.3125$ V, and the last code transition would occur at $v_{IN} = 4.0625$, and so the transfer curve would exactly match that of Fig. 28.20.

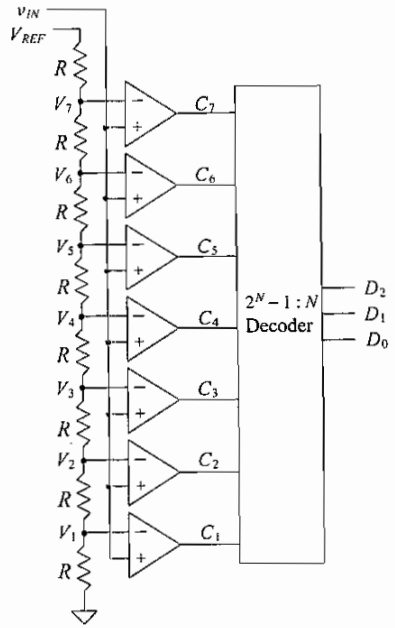


Figure 29.22 Three-bit Flash A/D converter to be used in Ex. 29.10.

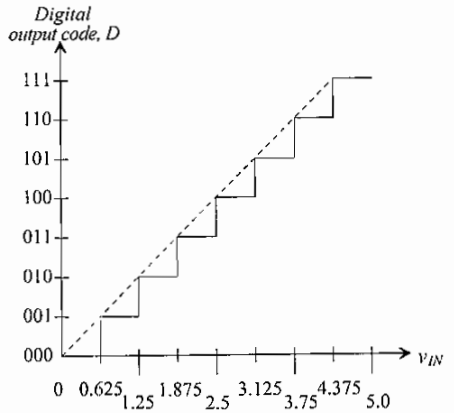


Figure 29.23 Transfer curve for the 3-bit Flash converter in Example 29.10.

v_{IN}	$C_7C_6C_5C_4C_3C_2C_1$	$D_2D_1D_0$
$0 \le v_{IN} < 0.625$ V	0000000	000
0.625 V $\le v_{IN} < 1.25$ V	0000001	001
1.25 V $\le v_{IN} < 1.875$ V	0000011	010
1.875 V $\le v_{IN} < 2.5$ V	0000111	011
2.5 V $\le v_{IN} < 3.125$ V	0001111	100
3.125 V $\le v_{IN} < 3.75$ V	0011111	101
3.75 V $\le v_{IN} < 4.375$ V	0111111	110
$4.375 \le v_{IN}$	1111111	111

Figure 29.24 Code transitions for the Flash ADC used in Ex. 29.10.

Based on Fig. 29.24, when $v_{IN} = 1.5$ V, only comparators C_1 and C_2 will have outputs of 1, since both V_1 and V_2 are less than 1.5 V. The remaining comparator outputs will be 0 since V_3 through V_8 will be greater than 1.5 V, thus generating the thermometer code, 0000011. The encoder must then convert this into a 3-bit digital word, resulting in 010. The same reasoning can be used to construct the data shown in Fig. 29.25. It should be obvious that if the polarity of the comparators were reversed, the thermometer code would be inverted. ■

v_{IN}	$C_7C_6C_5C_4C_3C_2C_1$	$D_2D_1D_0$
1.5	0000011	010
3.0	0001111	100
4.5	1111111	111

Figure 29.25 Output for the Flash ADC used in Ex. 29.10.

Accuracy Issues for the Flash ADC

Accuracy depends on the matching of the resistor string and the input offset voltage of the comparators. From our discussions earlier, we know that an ideal comparator should switch at the point at which the two inputs, v_+ and v_-, are the same potential. However, the offset voltage, V_{os}, prohibits this from occurring as the comparator output switches states as follows:

$$v_o = 1 \quad \text{when } v_+ \ge v_- + V_{os} \tag{29.41}$$

$$v_o = 0 \quad \text{when } v_+ < v_- + V_{os} \tag{29.42}$$

The resistor-string DAC was analyzed and presented in Sec. 29.1.2; the voltage on the i-th tap of the resistor string was found to be

$$V_i = V_{i,ideal} + \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} \tag{29.43}$$

where $V_{i,ideal}$ is the voltage at the i-th tap if all the resistors had an ideal value of R. The term, ΔR_k, is the value of the resistance error (difference from ideal) due to the mismatch. Note that for the resistor-string DAC, the sum of the mismatch terms plays an important factor in the overall voltage at each tap.

The switching point for the i-th comparator, $V_{sw,i}$, then becomes

$$V_{sw,i} = V_i + V_{os,i} \tag{29.44}$$

where $V_{os,i}$ is the input referred offset voltage of the i-th comparator. The INL for the converter can then be described as

$$INL = V_{sw,i} - V_{sw,ideal} = V_{sw,i} - V_{i,ideal} \tag{29.45}$$

which becomes

$$INL = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{i} \frac{\Delta R_k}{R} + V_{os,i} \tag{29.46}$$

The worst-case INL will occur at the middle of the string ($i = 2^{N-1}$), as described in Sec. 29.1.2 and Eq. (29.10). Including the offset voltage, the maximum INL will be

$$|INL|_{max} = \frac{V_{REF}}{2^N} \cdot \sum_{k=1}^{2^{N-1}} \frac{\Delta R_k}{R} + |V_{os,i}|_{max} = V_{REF} \cdot \frac{2^{N-1}}{2^N R} \cdot |\Delta R_k|_{max} + |V_{os,i}|_{max} \tag{29.47}$$

which can be rewritten as

$$|INL|_{max} = \frac{V_{REF}}{2} \cdot \left|\frac{\Delta R_k}{R}\right|_{max} + |V_{os,i}|_{max} \tag{29.48}$$

where it is assumed that the maximum positive mismatch occurs in all the resistors in the lower half of the string and the maximum negative mismatch occurs in the upper half (or vice versa) and that the comparator at the i-th tap contains the maximum offset voltage, $|V_{os,i}|_{max}$. Notice that the offset contributes directly to the maximum value for the INL. This explains another limitation to using Flash converters at high resolutions. The offset voltage alone can make the INL greater than ½ LSB.

Example 29.11

If a 10-bit Flash converter is designed, determine the maximum offset voltage of the comparators which will make the INL less than ½ LSB. Assume that the resistor string is perfectly matched and $V_{REF} = 5$ V.

Equation (29.48) requires that the offset voltage equal ½ LSB. Therefore,

$$|V_{os}|_{max} = \frac{5}{2^{11}} = 2.44 \text{ mV} \quad ■$$

The DNL calculation for the Flash converter is also attained using the analysis first presented in Sec. 29.1.2. Using the definition of DNL,

$$DNL = V_{sw,i} - V_{sw,i-1} - 1 \text{ LSB (in volts)} \tag{29.49}$$

Plugging in Eq. (29.44),

$$DNL = V_i + V_{os,i} - V_{i-1} - V_{os,i-1} - 1 \text{ LSB} \tag{29.50}$$

which can be written by using Eq. (29.6) as

$$DNL = V_{i,ideal} - V_{i-1,ideal} + \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} + V_{os,i} - V_{os,i-1} - 1\ LSB \qquad (29.51)$$

which becomes

$$DNL = \frac{V_{REF}}{2^N} \cdot \frac{\Delta R_i}{R} + V_{os,i} - V_{os,i-1} \qquad (29.52)$$

The maximum DNL will occur, assuming ΔR_i is at its maximum, $V_{os,i}$ is at its maximum positive value, and $V_{os,i-1}$ is at its maximum negative voltage. Thus,

$$|DNL|_{max} = \frac{V_{REF}}{2^N} \cdot \left|\frac{\Delta R_i}{R}\right|_{max} + 2|V_{os}|_{max} \qquad (29.53)$$

which assumes that the maximum offset voltage in the positive and negative directions are symmetrical. Therefore, both resistor-string matching and offset voltage affect the DNL of the converter.

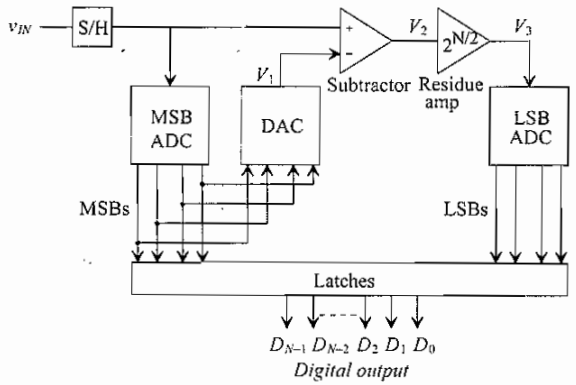


Figure 29.26 Block diagram of a two-step Flash ADC.

29.2.2 The Two-Step Flash ADC

Another type of Flash converter is called the two-step Flash converter or the parallel, feed-forward ADC [7–10]. The basic block diagram of a two-step converter is seen in Fig. 29.26. The converter is separated into two complete Flash ADCs with feed-forward circuitry. The first converter generates a rough estimate of the value of the input, and the second converter performs a fine conversion. The advantages of this architecture are that the number of comparators is greatly reduced from that of the Flash converter—from $2^N - 1$ comparators to $2(2^{N/2} - 1)$ comparators. For example, an 8-bit Flash converter requires 255 comparators, while the two-step Flash requires only 30. The trade-off is that the conversion process takes two steps instead of one, with the speed limited by the bandwidth and settling time required by the residue amplifier and the summer. The conversion process is as follows:

1. After the input is sampled, the most significant bits (MSBs) are converted by the first Flash ADC.
2. The result is then converted back to an analog voltage with the DAC and subtracted with the original input.
3. The result of the subtraction, known as the *residue*, is then multiplied by $2^{N/2}$ and input into the second ADC. The multiplication not only allows the two ADCs to be identical, but also increases the quantum level of the signal input into the second ADC.
4. The second ADC produces the least significant bits through a Flash conversion.

Some architectures use the same set of comparators in order to perform both steps. The multiplication mentioned in step 3 can be eliminated if the second converter is designed to handle very small input signals. The accuracy of the two-step ADC depends primarily on the linearity of the first ADC.

Figure 29.27 illustrates the two-step nature of the converter. A more intuitive approach can be explained with this picture. The first conversion identifies the segment in which the analog voltage resides. This is also known as a *coarse conversion* of the MSBs. The results of the coarse conversion are then multiplied by $2^{N/2}$ so that the segment within which V_{IN} resides will be scaled to the same reference as the first conversion. The second conversion is known as the *fine conversion* and will generate the final LSBs using the same Flash approach. One can see why the accuracy of the first converter is so important. If the input value is close to the boundary between two coarse segments and the first ADC is unable to choose the correct coarse segment, then the second conversion will be completely erroneous. The following example further illustrates the two-step algorithm.

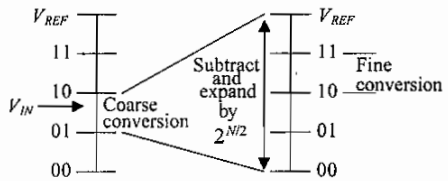


Figure 29.27 Coarse and fine conversions using a two-step ADC.

Example 29.12

Assume that the two-step ADC shown in Fig. 29.26 has four bits of resolution. Make a table listing the MSBs, V_1, V_2, V_3, and the LSBs for V_{IN} = 2, 4, 9, and 15 V assuming that V_{REF} = 16 V.

Since V_{REF} was conveniently made 16 V, each LSB will be 1 V. If V_{IN} = 2 V, the output of the first 2-bit Flash converter will be 00 since V_{REF} = 16 V and each resistor drops 4 V. The output of the 2-bit DAC, V_1, will therefore be 0, resulting in V_2 = 2 V. The multiplication of V_2 by the 4 results in V_3 = 8 V. Remember that

each 2-bit Flash converter resembles that of Fig. 29.21. The thermometer code from the second Flash converter will be 0011, which results in 10 as the LSBs. The other values can be calculated as seen in Fig. 29.28. ■

V_{IN}	D_3D_2 (MSBs)	V_1	V_2	V_3	D_1D_0 (LSBs)
2	00	0	2	8	10
4	01	4	0	0	00
9	10	8	1	4	01
15	11	12	3	12	11

Figure 29.28 Output for the Flash ADC used in Ex. 29.12.

Accuracy Issues Related to the Two-Step Flash Converters

As stated previously, the overall accuracy of the converter depends on the first ADC. The second Flash must have only the accuracy of a stand-alone Flash converter. This means that if an 8-bit, two-step Flash converter contains two 4-bit Flash converters, the second Flash needs only to have the resolution of a 4-bit Flash, which is not difficult to achieve. However, the first 4-bit Flash must have the accuracy of an 8-bit Flash, meaning that the worst-case INL and DNL for the first bit Flash must be less than ±½ LSB for an 8-bit ADC. Thus, the resistor matching and comparators contained in the first ADC must possess the accuracy of the overall converter. Refer to Sec. 29.2.1 for derivations on INL and DNL for a Flash. The DAC must also be accurate to within the resolution of the ADC.

Accuracy Issues Related to the Operational Amplifiers

With the addition of the summer and the amplifier, other sources of accuracy errors are present in this converter. The summer and the amplifier must add and amplify the signal to within ±½ LSB of the ideal value. It is difficult to implement standard operational amplifiers within high-resolution data converters because of these accuracy requirements. The nonideal characteristics of the op-amp are well known and in many cases alone limit the accuracy of the data converter. In this case, the amplifier is required to multiply the residue signal by some factor of two. Although this may not seem difficult at first glance, a closer examination will reveal a dependency on the open-loop gain.

Suppose that the amplifier were being used in a 12-bit, two-step data converter. Remember that in order for a data converter to be N-bit accurate, the INL and DNL need to be kept below ±½ LSB and one-half of an LSB can be defined as

$$0.5\ \text{LSB} = \frac{V_{REF}}{2^{N+1}} \tag{29.54}$$

Since the output of the amplifier gets quantized to 6 bits, the amplifier would need to be 6-bit accurate to within ±½ LSB, resulting in an accuracy of

$$\text{Accuracy} = \frac{0.5\ LSB}{\text{Full scale range } (V_{REF})} = \frac{1}{2^{6+1}} = \frac{1}{128} = 0.0078 = 0.78\% \tag{29.55}$$

And suppose that a feedback amplifier with a gain of 64, or $2^{N/2}$, is used as the residue amplifier. The gain would need to be within the following range:

$$63.5\ \text{V/V} < A_{CL} < 64.5\ \text{V/V} \tag{29.56}$$

where A_{CL} is the closed-loop gain of the amplifier. Already, one can see the limitations of using operational amplifiers with feedback in high-accuracy applications. Designing an op-amp based amplifier with a high degree of gain accuracy can be difficult.

Generalizing this concept for an N-bit application requires knowledge of feedback theory discussed in Ch. 24. The closed-loop gain of the amplifier is expressed as

$$A_{CL} = \frac{v_o}{v_i} = \frac{A_{OL}}{1 + A_{OL}\beta} \tag{29.57}$$

where A_{OL} is the open-loop gain of the amplifier and β is the feedback factor. Also, from Ch. 24, it is known that as A_{OL} increases in value, the closed-loop gain, A_{CL}, approaches the value of 1/β. Therefore, if it is assumed the closed-loop gain of the amplifier equals the ideal value of 1/β minus some maximum deviation from the ideal, ΔA, then,

$$A_{CL} = \frac{v_o}{v_i} = \frac{A_{OL}}{1 + A_{OL}\beta} = \frac{1}{\beta} - \Delta A \tag{29.58}$$

where 1/β is the desired value of the closed-loop gain (usually some factor of 2^N) and ΔA is the required accuracy (±½ LSB) of the gain (i.e., $(1/\beta) \cdot (1/2^{N+1})$). The right two terms of Eq. (29.58) can be solved for the open-loop gain of the amplifier,

$$|A_{OL}| = \frac{1}{\beta}(2^{N+1} - 1) \approx \frac{2^{N+1}}{\beta} \tag{29.59}$$

If the op-amp is used as a gain of 64 (1/β) and is required to amplify signals with 6-bit accuracy, then the open-loop gain of the amplifier must be at least $|A_{OL}| \geq 128 \cdot 64 = 8,192$ V/V. This is certainly an achievable specification. However, notice that for every bit increase in resolution, the open-loop gain requirement doubles. This is one reason two-step Flash converters are limited in resolution to approximately 12 bits [9–12].

The unity-gain frequency, f_{un}, required of an op-amp used in or with a data converter for a specific settling time t, (where $t < 1/f_{clk}$) can be estimated assuming linear settling, and requiring the output of the op-amp be ½ LSB accurate, by

$$v_{out} = V_{outfinal}\left(1 - \frac{1}{2^{N+1}}\right) = V_{outfinal}(1 - e^{-t/\tau}) \quad \text{or} \quad f_{un} \geq \frac{f_{clk} \cdot \ln 2^{N+1}}{2\pi \cdot \beta} \tag{29.60}$$

This equation can be used to determine the minimum op-amp gain-bandwidth product (= f_{un}) needed to achieve a specific settling time provided the op-amp slew-rate doesn't come into play and the op-amp can be modeled as a first-order system.

Linearity of the amplifier is another aspect of amplifier performance that must be considered when designing ADCs. The amplifier must be able to linearly amplify the input signal over an input voltage range to within ½ LSB of the number bits that its output is quantized. If the amplifier is not designed correctly, nonlinearity is introduced as devices in the amplifier go into nonsaturation. Harmonic distortion occurs, resulting in an error within the ADC. Linearity is typically measured in terms of total harmonic distortion, or THD, (refer to Sec. 21.3.3). However, the transfer curve illustrates the

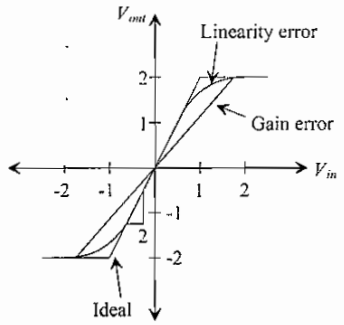


Figure 29.29 An op-amp transfer curve that distinguishes between gain error and linearity error.

limitation more effectively. Figure 29.29 shows a transfer curve of an op-amp with a gain of two. The ideal transfer curve is shown if the input range is known to be between –1 and 1 V. The actual transfer curve shows nonlinearity introduced at both ends of the input range. In order for the amplifier to be N-bit accurate, the slope of the actual transfer curve may not vary from the ideal by more than the accuracy required at the output of the amplifier. Note also in Fig. 29.29 the subtle difference between a gain error and nonlinearity. However, a gain error is much less harmful to an ADC's performance than harmonic distortion.

29.2.3 The Pipeline ADC

After examining the two-step ADC, one might wonder whether there is such a converter as a three-step or four-step ADC. In actuality, one could divide the number of conversions into many steps. The pipeline ADC is an N-step converter, with 1 bit being converted per stage. Able to achieve high resolution (10–13 bits) at relatively fast speeds [11–15], the pipeline ADC consists of N stages connected in series (Fig. 29.30). Each stage contains a 1-bit ADC (a comparator), a sample-and-hold, a summer, and a gain of two amplifier. Each stage of the converter performs the following operation:

1. After the input signal has been sampled, compare it to $\frac{V_{REF}}{2}$. The output of each comparator is the bit conversion for that stage.
2. If $v_{IN} > \frac{V_{REF}}{2}$ (comparator output is 1), $\frac{V_{REF}}{2}$ is subtracted from the held signal and pass the result to the amplifier. If $v_{IN} < \frac{V_{REF}}{2}$ (comparator output is 0), then pass the original input signal to the amplifier. The output of each stage in the converter is referred to as the *residue*.
3. Multiply the result of the summation by 2 and pass the result to the sample-and-hold of the next stage.

A main advantage of the pipeline converter is its high throughput. After an initial *latency* of N clock cycles, one conversion will be completed per clock cycle. While the residue of the first stage is being operated on by the second stage, the first stage is free to operate on the next samples. Each stage operates on the residue passed down from the

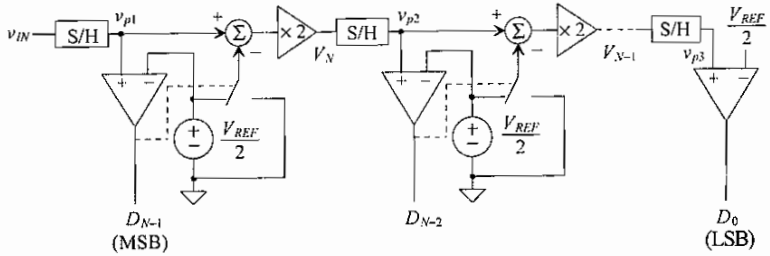


Figure 29.30 Block diagram of a pipeline ADC.

previous stage, thereby allowing for fast conversions. The disadvantage is having the initial N clock cycle delay before the first digital output appears. The severity of this disadvantage depends, of course, on the application.

One interesting aspect of this converter is its dependency on the most significant stages for accuracy. A slight error in the first stage propagates through the converter and results in a much larger error at the end of the conversion. Each succeeding stage requires less accuracy than the one before, so special care must be taken when considering the first several stages.

Example 29.13

Assume that the pipeline converter shown in Fig. 29.30 is a 3-bit converter. Analyze the conversion process by making a table of the following variables: D_2, D_1, D_0, V_2, V_1, for v_{IN} = 2, 3, and 4.5 V. Assume that V_{REF} = 5 V, V_3 is the residue voltage out of the first stage, and V_2 is the residue voltage out of the second stage.

The output of the first comparator, D_2 = 0, since $v_{IN} < 2.5$ V. Since D_2 = 0, V_3 = 2(2) = 4 V. Passing this voltage down the pipeline, since $V_3 > 2.5$ V, D_1 = 1 and V_2 becomes

$$V_1 = \left(V_2 - \frac{V_{REF}}{2}\right) \times 2 = 3 \text{ V}$$

The LSB, D_0 = 1, since $V_2 > 2.5$ V, and the digital output corresponding to V_{IN} = 2 V, is $D_2D_1D_0$ = 011. The actual digital outputs are simply the comparator outputs, and the data can be completed as seen in Fig. 29.31. ■

v_{IN}	V_3 (V)	V_2 (V)	Digital Out ($D_2D_1D_0$)
2.0	4.0	3.0	011
3.0	1.0	2.0	100
4.5	4.0	3.0	111

Figure 29.31 Output for the pipeline ADC used in Ex. 29.13.

Accuracy Issues Related to the Pipeline Converter

The 1-bit per stage ADC can be analyzed by examining the switching point of each comparator for the ideal and nonideal case. Using Fig. 29.30, and assuming that all of the components are ideal, let $v_{IN,1}$ represent the value of the input voltage when the first comparator switches. This occurs when

$$v_{IN,1} = \frac{1}{2}V_{REF} \tag{29.61}$$

The positive input voltage on the second comparator, v_{p2}, can be written in terms of the previous stage, or

$$v_{p2} = [v_{IN} - \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot 2 \tag{29.62}$$

where D_{N-1} is the MSB output from the first comparator and is either a 1 or a 0. The second comparator switches when $v_{p2} = \frac{1}{2}V_{REF}$. The value of v_{IN} at this point, denoted as $v_{IN,2}$, is

$$v_{IN,2} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{4}V_{REF} \tag{29.63}$$

Continuing on in a similar manner, we can write the value of the voltage on the positive input of the third comparator in terms of the previous two stages as

$$v_{p3} = \left[[v_{IN} - \frac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot 2 - [\frac{1}{2} \cdot D_{N-2} \cdot V_{REF}] \right] \cdot 2 \tag{29.64}$$

and the third comparator will switch when $v_{p3} = \frac{1}{2}V_{REF}$, which corresponds to the point at which v_{IN} becomes

$$v_{IN,3} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{4} \cdot D_{N-2} \cdot V_{REF} + \tfrac{1}{8}V_{REF} \tag{29.65}$$

By now, a general trend can be recognized and the value of v_{IN} can be derived for the point at which the comparator of the N-th stage switches. This expression can be written as

$$v_{IN,N} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{4} \cdot D_{N-2} \cdot V_{REF} + \tfrac{1}{8} \cdot D_{N-3} \cdot V_{REF} + \ldots\ldots + \tfrac{1}{2^{N-1}} \cdot D_1 \cdot V_{REF} + \tfrac{1}{2^N} \cdot V_{REF} \tag{29.66}$$

Notice that the preceding equation does not include D_0. This is because D_0 is the output of the N-th stage comparator.

Now that we have derived the switching points for the ideal case, the nonideal case can be considered. Only the major sources of error will be included in the analysis so as not to overwhelm the reader. These include the comparator offset voltage, $V_{COS,x}$, and the sample-and-hold offset voltage, $V_{SOS,x}$. The variable, x, represents the number of the stage for which each of the errors is associated, and the "prime" notation will be used to distinguish between the ideal and nonideal case. The reader should also be aware that the offset voltages can be of either polarity. It will be assumed that all of the residue amplifiers have the same gain, denoted as A.

The positive input to the first nonideal comparator, v'_{p1}, will include the offset from the first sample-and-hold, such that

$$v'_{p1} = v_{IN} + V_{SOS,1} \tag{29.67}$$

Now the first comparator will not switch until the voltage on the positive input overcomes the comparator offset as well. This occurs when

$$v'_{p1} = \tfrac{1}{2}V_{REF} + V_{COS,1} \tag{29.68}$$

Thus, equating Eqs. (29.67) and (29.68) and solving for the value of the input voltage when the switching occurs for the first comparator yields

$$v'_{IN,1} = \tfrac{1}{2}V_{REF} + V_{COS,1} - V_{SOS,1} \tag{29.69}$$

The input to the second comparator, v'_{p2}, can be written as

$$v'_{p2} = [v_{IN} + V_{SOS,1} - \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF}] \cdot A + V_{SOS,2} \tag{29.70}$$

and the value of input voltage at the point which the second comparator switches occurs when

$$v'_{IN,2} = \frac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \frac{1}{2}\frac{V_{REF}}{A} - V_{SOS,1} - \frac{1}{A}(V_{SOS,2} - V_{COS,2}) \tag{29.71}$$

Continuing in the same manner, we can write the value of the input voltage that causes the third comparator to switch as

$$v'_{IN,3} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{2} \cdot D_{N-2} \cdot \tfrac{V_{REF}}{A} - V_{SOS,1} - \tfrac{1}{A}V_{SOS,2} - \tfrac{1}{A^2}V_{SOS,3} - \tfrac{1}{A^2}\left[V_{COS,3} - \tfrac{1}{2}V_{REF}\right] \tag{29.72}$$

which can be generalized to the N-th switching point as

$$v'_{IN,N} = \tfrac{1}{2} \cdot D_{N-1} \cdot V_{REF} + \tfrac{1}{2} \cdot D_{N-2} \cdot \tfrac{V_{REF}}{A} + \ldots + \tfrac{1}{2} \cdot D_1 \cdot \tfrac{V_{REF}}{A^{N-2}} + \tfrac{1}{2} \cdot \tfrac{V_{REF}}{A^{N-1}} + \tfrac{V_{COS,N}}{A^{N-1}} - \sum_{k=1}^{N} \tfrac{V_{SOS,k}}{A^{k-1}} \tag{29.73}$$

The INL can be calculated by subtracting switching point between the nonideal and ideal case. Therefore, the INL of the first stage is found by subtracting Eqs. (29.69) and (29.61).

$$INL_1 = v'_{IN,1} - v_{IN,1} = V_{COS,1} - V_{SOS,1} \tag{29.74}$$

The second stage INL is

$$INL_2 = v'_{IN,2} - v_{IN,2} = \frac{V_{REF}}{2}\left(\frac{1}{A} - \frac{1}{2}\right) - V_{SOS,1} - \frac{V_{SOS,2}}{A} + \frac{V_{COS,2}}{A} \tag{29.75}$$

and the INL for the N-th stage is

$$INL_N = \frac{1}{2} \cdot D_{N-2} \cdot V_{REF} \cdot \left(\frac{1}{A} - \frac{1}{2}\right) + \frac{1}{2} \cdot D_{N-3} \cdot V_{REF} \cdot \left(\frac{1}{A^2} - \frac{1}{4}\right) + \ldots$$
$$+ \frac{1}{2} \cdot D_1 \cdot V_{REF} \cdot \left(\frac{1}{A^{N-2}} - \frac{1}{2^{N-2}}\right) + \frac{1}{2} \cdot V_{REF} \cdot \left(\frac{1}{A^{N-1}} - \frac{1}{2^{N-1}}\right) + \frac{V_{COS,N}}{A^{N-1}} - \sum_{k=1}^{N} \frac{V_{SOS,K}}{A^{k-1}} \tag{29.76}$$

Equations (29.74)–(29.76) are very important to understanding the limitations of the pipeline ADC. Notice the importance of the comparator and summer offsets in Eq. (29.74). The worst-case addition of the offsets must be less than ½ LSB to keep the ADC

N-bit accurate. The second stage is more dependent on the gain of the residue amplifier as seen in Eq. (29.75). The gain error discussed in the previous section plays an important role in determining the overall accuracy of the converter. Now examine the effects of the offsets on the INL of the N-th stage. In Eq. (29.76), both the comparator and summer offsets of the N-th stage (when $k = N$) are divided by a large gain. Therefore, the latter stages in a pipeline ADC are not as critical to the accuracy as the first stages, and die area and power can be reduced by using less accurate designs for the least significant stages. The summation term in Eq. (29.76) also reveals that the summer offset of the first stage ($k = 1$) has a large effect on the N-th stage. However, this point is inconsequential since $V_{SOS,1}$ must be minimized to achieve N-bit accuracy for the first stage anyway. Typically, if the INL and DNL specifications can be made N-bit accurate in the first few stages, the latter stages will not adversely affect overall accuracy.

The DNL can be found by calculating the difference between the worst-case switching points and subtracting the ideal value for an LSB. As defined earlier, the worst case will occur at midscale when the output switches from 0111...111 to 1000...000 as v_{IN} increases. Thus, the DNL is

$$DNL_{max} = v'_{IN,1} - v'_{IN,N} - \frac{V_{REF}}{2^N} \tag{29.77}$$

where $v'_{IN,N}$ is calculated using Eq. (29.73) and assuming that D_{N-1} is a zero and that all of the other bits are ones. Plugging in Eqs. (29.69) and (29.73) into Eq. (29.77) yields

$$DNL_{max} = \frac{1}{2}V_{REF}\left(1-\sum_{k=1}^{N-1}\frac{1}{A^k}\right) + V_{COS,1} - \frac{V_{COS,N}}{A^{N-1}} + \sum_{k=2}^{N}\frac{V_{SOS,K}}{A^{k-1}} - \frac{V_{REF}}{2^N} \tag{29.78}$$

Again, the term that dominates this expression is the comparator offset associated with the first stage and the summer offset of the second stage. The entire expression in Eq. (29.78) must be less than ½ LSB for the ADC to have N-bit resolution.

29.2.4 Integrating ADCs

Another type of ADC performs the conversion by integrating the input signal and correlating the integration time with a digital counter. Known as single- and dual-slope ADCs, these types of converters are used in high-resolution applications but have relatively slow conversions. However, they are very inexpensive to produce and are commonly found in slow-speed, cost-conscious applications.

Single-Slope Architecture

Figure 29.32 illustrates the single-slope converter in block level form. A counter determines the number of clock pulses that are required before the integrated value of a reference voltage is equal to the sampled input signal. The number of clock pulses is proportional to the actual value of the input, and the output of the counter is the actual digital representation of the analog voltage.

Since the reference is a DC voltage, the output of the integrator should start at zero and linearly increase with a slope that depends on the gain of the integrator. Notice that the reference voltage is defined as negative so that the output of the inverting integrator is positive. At the time when the output of the integrator surpasses the value of the S/H output, the comparator switches states, thus triggering the control logic to latch

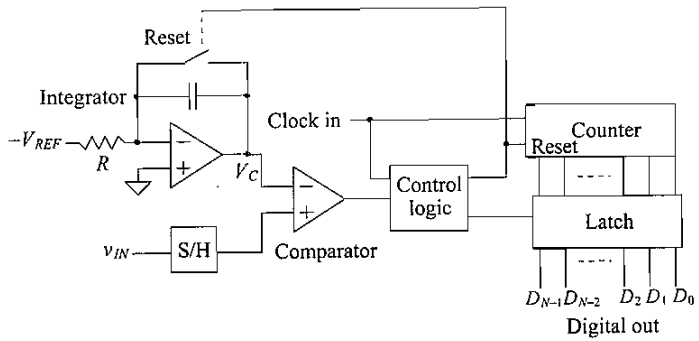


Figure 29.32 Block diagram of a single-slope ADC.

the value of the counter. The control logic also resets the system for the next sample. Figure 29.33 illustrates the behavior of the integrator output and the clock.

Note that if the input voltage is very small, the conversion time is very short, as the counter has to increment only a few times before the comparator latches the data. However, if the input voltage is at its full-scale value, the counter must increment to its maximum value of 2^N clock cycles. Thus, the clock frequency must be many times faster than the bandwidth of the input signal. The conversion time, t_c, depends on the value of the input signal and can be described as

$$t_c = \frac{v_{IN}}{V_{REF}} \cdot 2^N \cdot T_{CLK} \tag{29.79}$$

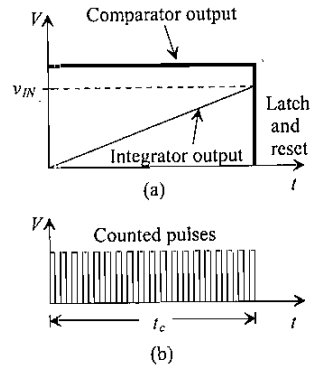


Figure 29.33 Single-slope ADC timing diagrams for (a) the comparator inputs and outputs and (b) the resulting counted pulses.

where T_{CLK} is the period of the clock. The sampling rate is inversely proportional to the conversion time and can be written as

$$f_{Sample} = \frac{V_{REF}}{V_{IN} \cdot 2^N} \cdot f_{CLK} \tag{29.80}$$

Example 29.14

Determine the clock frequency needed to form an 8-bit, single-slope converter, if the analog signal bandwidth is 20 kHz.

Since the sampling rate required is 40 kHz, then the worst-case situation would occur for a full-scale input, in which event the integrator output would have to climb to its maximum value and the counter would increment 2^N times during the corresponding 25 μs period between samples. Therefore, the clock frequency would need to be 2^N times faster than the sampling rate or 10.24 MHz. ■

Accuracy Issues Related to the Single-Slope ADC

Obviously, many potential error sources abound in this architecture. At the end of the conversion, the voltage across the integrating capacitor, V_C , assuming no initial condition, will be

$$V_C = \frac{1}{C}\int_0^{t_c} \frac{V_{REF}}{R} dt = \frac{V_{REF} \cdot t_c}{RC} \tag{29.81}$$

where t_c is the conversion time. Plugging Eq. (29.79) into Eq. (29.81) yields

$$V_C = \frac{2^N \cdot T_{CLK} \cdot v_{IN}}{RC} = \frac{2^N \cdot v_{IN}}{f_{CLK} \cdot RC} \tag{29.82}$$

Equation (29.82) is a revealing one in that the final voltage on the integrator output depends not only on the value of the input voltage, which is to be expected, but also on the value of R, C, and f_{CLK}. Therefore, any nonideal effects affecting these values will have an influence on the accuracy of the integrator output from sample to sample. For example, if an integrated diffused-resistor is used, then the voltage coefficient of the resistor could limit the accuracy, since the resistor will be effectively nonlinear. Similarly, the capacitor may have charge leakage or aging effects associated with it. Also, any jitter in the clock will affect the overall accuracy. The integrator must have a linear slope to within the accuracy of the converter, which depends on the specifications of the op-amp (open- loop gain, settling time, offset, etc.) and must be considered accordingly.

Offset voltages on the comparator, the S/H, or the integrator result in additional or fewer clock pulses, depending on the polarity of the offset. A delay also exists from the time that the inputs to the comparator are equal and the time that the output of the counter is actually latched. The reference voltage must also stay constant to within the accuracy of the converter.

Dual-Slope Architecture

A slightly more sophisticated design known as the dual-slope, integrating ADC (Fig. 29.34) eliminates most of the problems encountered when using the single-slope converter. Here, two integrations are performed, one on the input signal and one on V_{REF}. The input voltage in this case is assumed to be negative, so that the output of the

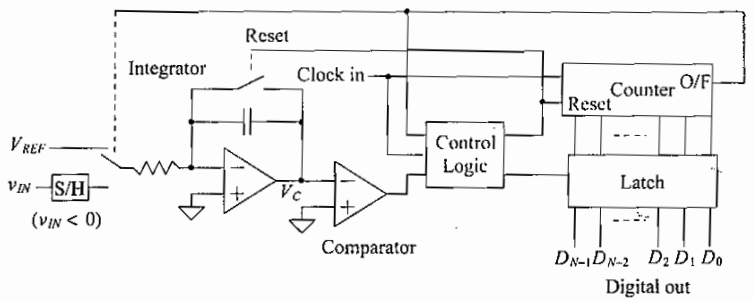


Figure 29.34 Block diagram of a dual-slope ADC.

inverting integrator results in a positive slope during the first integration. Figure 29.35 illustrates the behavior for two separate samples. The first integration is of fixed length, dictated by the counter, in which the sample-and-held signal is integrated, resulting in the first slope. After the counter overflows and is reset, the reference voltage is connected to the input of the integrator. Since v_{IN} was negative and the reference voltage is positive, the inverting integrator output begins discharging back down to zero at a constant slope. A counter again measures the amount of time for the integrator to discharge, thus generating the digital output.

For Fig. 29.35, a 3-bit ADC is being used. Thus, the first integration period continues until the beginning of the eighth (2^3) clock pulse, which corresponds to the counter's overflow bit. Note that the integrator's output corresponding to V_B is twice the value of the output corresponding to V_A. Thus, it requires twice as many clock pulses for

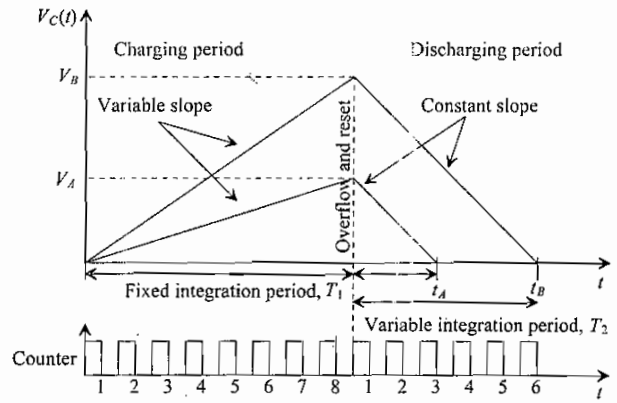


Figure 29.35 Integration periods and counter output for two separate samples of a 3-bit dual-slope ADC.

the integrator to discharge back to zero from V_B than from V_A. The output of the counter at t_A is three or 011, while the counter output at t_B is twice that value or six (110) and the quantization is complete.

Notice that the first slope varies according to the value of the input signal, while the second slope, dependent only on V_{REF}, is constant. Similarly, the time required to generate the first slope is constant, since it is limited by the size of the counter. However, the discharging period is variable and results in the digital representation of the input voltage.

Accuracy Issues Related to the Dual-Slope ADC

One may wonder how the dual-slope converter is an improvement over the single-slope architecture, since a significantly longer conversion time is required. The first integration period requires a full 2^N clock cycle and cannot be decreased, because the second integration might require the full 2^N clock cycles to discharge if the maximum value of v_{IN} is being converted. However, the dual slope is the preferred architecture because the same integrator and clock are used to produce both slopes. Therefore, any nonidealities will essentially be canceled. For example, assuming that the S/H is ideal, the gain of the integrator at the end of the first integration period, T_1, becomes

$$V_C = -\frac{1}{C}\int_0^{T_1}\frac{v_{IN}}{R}dt = \frac{|v_{IN}|\cdot T_1}{RC} \tag{29.83}$$

The output at the end of T_1 is positive since the input voltage is considered to be negative and the integrator is inverting. After the clock has been reset, the discharging commences, with the initial condition defined by the value of the integrator output at the end of the charging period, or

$$V_C = \frac{|v_{IN}|\cdot T_1}{RC} - \frac{1}{C}\int_0^{T_2}\frac{V_{REF}}{R}dt \tag{29.84}$$

Once the value of the integrator output, V_C, reaches zero volts, Eq. (29.84) becomes

$$V_C = \frac{|v_{IN}|\cdot T_1}{RC} - \frac{V_{REF}\cdot T_2}{RC} = 0 \tag{29.85}$$

or,

$$|v_{IN}|\cdot T_1 = V_{REF}\cdot T_2 \tag{29.86}$$

At the end of the conversion, the dependencies on R and C have canceled out. Since we also know that the counter increments 2^N times at time, T_1, and the counter increments D times at time, T_2, Eq. (29.86) can be rewritten as

$$\frac{D}{2^N} = \frac{|v_{IN}|}{V_{REF}} \tag{29.87}$$

where D is the counter output that is actually the digital representation of the input voltage. Thus, it can be written that the ratio of the input voltage and the reference voltage is proportional to the ratio of the binary value of the digital word, D, and 2^N. Therefore, since the same clock pulse is responsible for the charging and discharging times, any irregularities will also cancel out.

29.2.5 The Successive Approximation ADC

The successive approximation converter performs basically a binary search through all possible quantization levels before converging on the final digital answer. The block diagram is seen in Fig. 29.36. An N-bit register controls the timing of the conversion where N is the resolution of the ADC. V_{IN} is sampled and compared to the output of the DAC. The comparator output controls the direction of the binary search, and the output of the successive approximation register (SAR) is the actual digital conversion. The successive approximation algorithm is as follows.

1. A 1 is applied to the input to the shift register. For each bit converted, the 1 is shifted to the right 1-bit position. $B_{N-1} = 1$ and B_{N-2} through $B_0 = 0$.
2. The MSB of the SAR, D_{N-1}, is initially set to 1, while the remaining bits, D_{N-2} through D_0, are set to 0.
3. Since the SAR output controls the DAC and the SAR output is 100...0, the DAC output will be set to $\frac{V_{REF}}{2}$.
4. Next, v_{IN} is compared to $\frac{V_{REF}}{2}$. If $\frac{V_{REF}}{2}$ is greater than v_{IN}, then the comparator output is a 1 and the comparator resets D_{N-1} to 0. If $\frac{V_{REF}}{2}$ is less than v_{IN}, then the comparator output is a 0 and the D_{N-1} remains a 1. D_{N-1} is the actual MSB of the final digital output code.
5. The 1 applied to the shift register is then shifted by one position so that $B_{N-2} = 1$, while the remaining bits are all 0.
6. D_{N-2} is set to a 1, D_{N-3} through D_0 remain 0, while D_{N-1} remains the value from the MSB conversion. The output of the DAC will now either equal $\frac{V_{REF}}{4}$ (if $D_{N-1} = 0$) or $\frac{3V_{REF}}{4}$ (if $D_{N-1} = 1$).

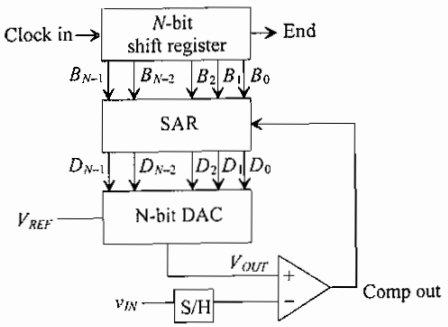


Figure 29.36 Block diagram of the successive approximation ADC.

7. Next, v_{IN} is compared to the output of the DAC. If the DAC output is greater than v_{IN}, the comparator the D_{N-2} is reset to 0. If v_{IN} is less than the DAC output, D_{N-2} remains a 1.

8. The process repeats until the output of the DAC converges to the value of v_{IN} within the resolution of the converter.

Figure 29.37 shows an example of the binary search nature of the converter. The bolded line shows the path of the conversion for 101, corresponding to $\frac{5}{8}V_{REF}$. All possible quantization levels are represented in the binary tree. With each bit decided, the search space decreases by one-half until the correct answer is converged upon.

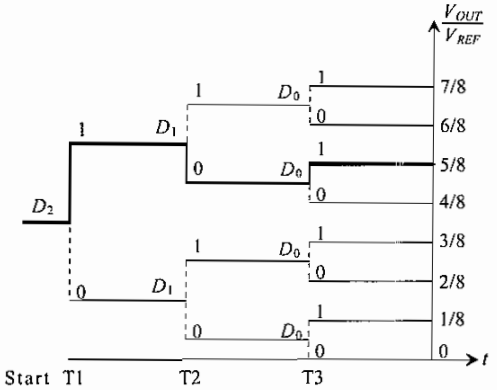


Figure 29.37 Binary search performed by a 3-bit successive approximation ADC for D=101.

Example 29.15

Perform the operation of a 3-bit successive approximation ADC similar to Fig. 29.36 with V_{REF} = 8. Make a table that consists of $D_2D_1D_0$, $B_2B_1B_0$, V_{OUT} (the output from the DAC) and the comparator output, which shows the binary search algorithm of the converter for v_{IN} = 5.5 V and 2.5 V.

We will designate $D_2.D_1.D_0$. as the initial output of the SAR before the comparator makes its decision. The final value is designated as $D_2D_1D_0$. Notice that if the comparator is a 1, $D_2.D_1.D_0$. differs from $D_2D_1D_0$, but if the comparator outputs a 0, then $D_2.D_1.D_0$. = $D_2D_1D_0$. The output of the shift register is designated as $B_2B_1B_0$.

Following the algorithm discussed previously, initially v_{IN} = 5.5 V and is compared with 4 V. Since the comparator output is 0, the MSB remains a 1. The next bit is examined, and the output of the DAC is now 6 V. Since $V_{OUT} > v_{IN}$, the comparator output is 1, which resets the current SAR bit, D_1, to a 0 at the end of period T2. Lastly, the LSB is examined, and v_{IN} is compared with 5 V. Since $v_{IN} > V_{OUT}$, the comparator output is a 0, and the current SAR bit, D_0, remains a 1. The

results can be examined in Fig. 29.38a. The final value for $D_2D_1D_0$ is 101, which is what is expected considering that 101 in binary is equivalent to 5_{10}. Figure 29.38b shows the data for the ADC using v_{IN} = 2.5 V. The final value for v_{IN} = 2.5 is 010, which again is what is expected for 3-bit resolution. ■

Step	v_{IN}	$B_2B_1B_0$	$D_2.D_1.D_0.$	V_{OUT}	Comp Out	$D_2D_1D_0$
T1	5.5	100	100	1/2 V_{REF} = 4 V	0	100
T2	5.5	010	110	(1/2+1/4)V_{REF} = 6 V	1	100
T3	5.5	001	101	(1/2+1/8)V_{REF} = 5 V	0	101

(a)

Step	v_{IN}	$B_2B_1B_0$	$D'_2D'_1D'_0$	V_{OUT}	Comp Out	$D_2D_1D_0$
T1	2.5	100	100	1/2 V_{REF} = 4 V	1	000
T2	2.5	010	010	1/4 V_{REF} = 2 V	0	010
T3	2.5	001	011	(1/4+1/8)V_{REF} = 3 V	1	010

(b)

Figure 29.38 Results from the 3-bit successive approximation ADC using (a) v_{IN} = 5.5 and (b) 2.5 V.

The successive approximation ADC is one of the most popular architectures used today. The simplicity of the design allows for both high speed and high resolution while maintaining relatively small area. The limit to the ADC's accuracy depends mainly on the accuracy of the DAC. If the DAC does not produce the correct analog voltage with which to compare the input voltage, the entire converter output will contain an error. Referring again to Fig. 29.37, we can see that if a wrong decision is made early, a massive error will result as the converter attempts to search for the correct quantization level in the wrong half of the binary tree.

The Charge-Redistribution Successive Approximation ADC

One of the most popular types of successive approximation architectures uses the binary-weighted capacitor array (analyzed in Sec. 29.1.5) as its DAC. Called a charge-redistribution, successive-approximation ADC [2,16,17], this converter samples the input signal and then performs the binary search based on the amount of charge on each of the DAC capacitors. Figure 29.39 shows an N-bit architecture. A comparator has replaced the unity gain buffer used in the DAC architecture. The binary-weighted capacitor array also samples the input voltage, so no external sample-and-hold is needed.

The conversion process begins by discharging the capacitor array, via the reset switch. Although this may appear to be an insignificant action, the converter is also performing automatic offset cancellation. Once the reset switch is closed, the comparator acts as a unity gain buffer. Thus, the capacitor array charges to the offset voltage of the comparator. This requires that the comparator is designed to be unity-gain stable, which means that internal compensation may have to be switched in during the reset period. Next, the input voltage, v_{IN}, is sampled onto the capacitor array. The reset switch is still closed, for the top plate of the capacitor array needs to be connected to virtual ground of the unity gain buffer. The equivalent circuit is seen in Fig. 29.40a. The reset switch is

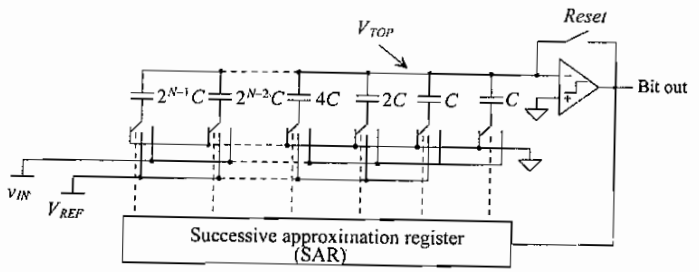


Figure 29.39 A charge redistribution ADC using a binary-weighted capacitor array DAC.

then opened, and the bottom plates of each capacitor in the array are switched to ground, so that the voltage appearing at the top plate of the array is now $V_{OS} - v_{IN}$ (Fig. 29.40b). The conversion process begins by switching the bottom plate of the MSB capacitor to V_{REF} (Fig. 29.40c). If the output of the comparator is high, the bottom plate of the MSB capacitor remains connected to V_{REF}. If the comparator output is low, the bottom plate of the MSB is connected back to ground. The output of the comparator is D_{N-1}. The voltage at the top of the capacitor array, V_{TOP}, is now

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} \tag{29.88}$$

The next largest capacitor is tested in the same manner as seen in Fig. 29.40d. The voltage at the top plate of the capacitor after the second capacitor is tested becomes

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} + D_{N-2} \cdot \frac{V_{REF}}{4} \tag{29.89}$$

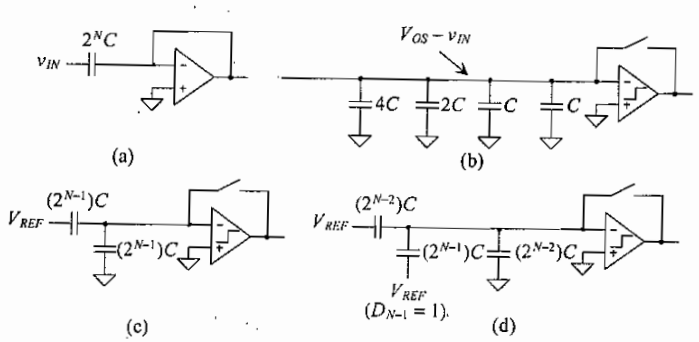


Figure 29.40 The charge redistribution process: (a) Sampling the input while autozeroing the offset, (b) the voltage at the top plate after sampling, (c) the equivalent circuit while converting the MSB, and (d) the equivalent circuit while converting the next largest capacitor with the MSB result equal to one.

The conversion process continues on with the remaining capacitors so that the voltage on the top plate of the array, V_{TOP}, converges to the value of the offset voltage, V_{OS} (within the resolution of the converter), or

$$V_{TOP} = -v_{IN} + V_{OS} + D_{N-1} \cdot \frac{V_{REF}}{2} + D_{N-2} \cdot \frac{V_{REF}}{4} + ... + D_1 \cdot \frac{V_{REF}}{2^{N-2}} + D_0 \cdot \frac{V_{REF}}{2^{N-1}} \approx V_{OS} \tag{29.90}$$

Note that the initial charge stored on the capacitor array is now redistributed onto only those capacitors that have their bottom plates connected to V_{REF}.

Accuracy Issues Related to the Charge-Redistribution, Successive-Approximation ADC

Obviously, the limitation of this architecture is the capacitor matching. The mismatch is analyzed in the same manner as the binary-weighted current source array of Sec. 29.1.4. Thus, substituting the value of the unit capacitance, C, for the value of the unit current source, I, and using Eqs. (29.27)–(29.30),

$$|INL|_{max} = 2^{N-1}(C + |\Delta C|_{max,INL}) - 2^{N-1} \cdot C = 2^{N-1} \cdot |\Delta C|_{max,INL} \tag{29.91}$$

where the maximum ΔC that will result in an INL that is less than ½ LSB is

$$|\Delta C|_{max,INL} = \frac{0.5C}{2^{N-1}} = \frac{C}{2^N} \tag{29.92}$$

and DNL is defined by

$$DNL_{max} = (2^N - 1) \cdot |\Delta C|_{max,DNL} \tag{29.93}$$

with the maximum ΔC, which results in a DNL less than ½ LSB:

$$|\Delta C|_{max,DNL} = \frac{0.5C}{2^N - 1} = \frac{C}{2^{N+1} - 2} \tag{29.94}$$

29.2.6 The Oversampling ADC

ADCs can be separated into two categories depending on the rate of sampling. The first category samples the input at the Nyquist rate, or $f_N = 2F$ where F is the bandwidth of the signal and f_N is the sampling rate. The second type samples the signal at a rate much higher than the signal bandwidth. This type of converter is called an oversampling converter. Traditionally, successive approximation or dual-slope converters are used when high resolution is desired. However, trimming is required when attempting to achieve higher accuracy. Dual-slope converters require high-speed, high-accuracy integrators that are only available using a high f_T bipolar process. Having to design a high-precision sample-and-hold is another factor that limits the realization of a high-resolution ADC using these architectures.

The oversampling ADC [18–20] is able to achieve much higher resolution than the Nyquist rate converters. This is because digital signal processing techniques are used in place of complex and precise analog components. The accuracy of the converter does not depend on the component matching, precise sample-and-hold circuitry, or trimming, and only a small amount of analog circuitry is required. Switched-capacitor implementations are easily achieved, and, as a result of the high sampling rate, only simplistic anti-aliasing circuitry needs to be used. However, because of the amount of time required to sample the input signal, the throughput is considerably less than the Nyquist rate ADCs.

Differences in Nyquist Rate and Oversampling ADCs

The typical process used in analog-to-digital conversion is seen in Fig. 29.41a, while the block diagram for the oversampling ADC is seen in Fig. 29.41b. After filtering the signal to help minimize aliasing effects, the signal is sampled, quantized, and encoded or decoded using simple digital logic to provide the digital data in the proper format. When using oversampling ADCs, little if any, anti-alias filtering is needed, no dedicated S/H is required, the quantization is performed with a modulator, and the encoding usually takes the form of a digital filter.

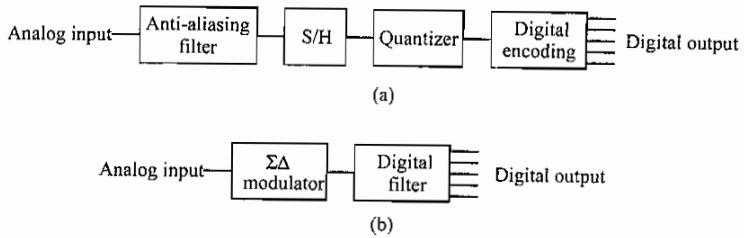


Figure 29.41 Typical block diagram for (a) Nyquist rate converters and (b) oversampling ADCs.

Since the oversampling converter samples the signal bandwidth at many times, aliasing is not a serious problem. A discussion of the frequency characteristics of aliasing was presented in Ch. 28. Figure 29.42a shows that when using Nyquist rate converters, a sampled signal in the frequency domain appears as a series of band-limited signals at multiples of the sampling frequency (see Fig. 28.26 for more details). As the sampling frequency decreases, the frequency spectra begin to overlap, and aliasing (Fig. 29.42b) occurs. Complex, "brickwall" filters are needed to correct the problem.

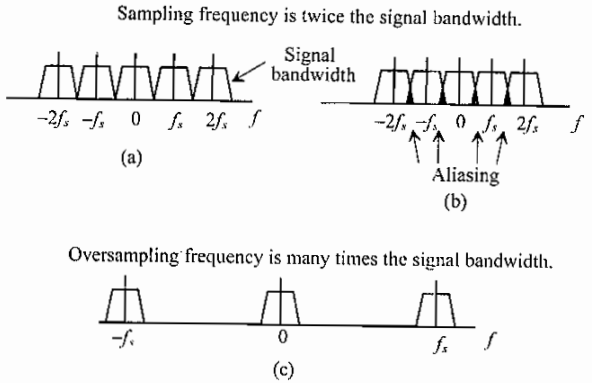


Figure 29.42 Frequency domain for (a) Nyquist rate converters, (b) the aliasing that occurs, and (c) an oversampling converter.

For oversampled ADCs, aliasing becomes much less of a factor. Since the sampling rate is much greater than the bandwidth of the signal, the frequency domain representation shows that the spectra are widely spaced, as seen in Fig. 29.42c. Therefore, overlapping of the spectra, and thus aliasing, will not occur, and only simple, first-order filters are required.

Oversampling converters typically employ switched-capacitor circuits and therefore do not need sample-and-hold circuits. The output of the modulator is a pulse-density modulated signal that represents the average of the input signal. The modulator constructs these pulses in real time, and so it is not necessary to hold the input value and perform the conversion.

As stated previously, the modulator actually provides the quantization in the form of a pulse-density modulated signal. Referred to as sigma-delta (ΣΔ) or delta-sigma (ΔΣ) modulation, the density of the pulses represents the average value of the signal over a specific period. Figure 29.43 illustrates the output of the modulator for the positive half of a sine wave input. Note that for the peak of the sine wave, most of the pulses are high. As the sine wave decreases in value, the pulses become distributed between high and low according to the sine wave value.

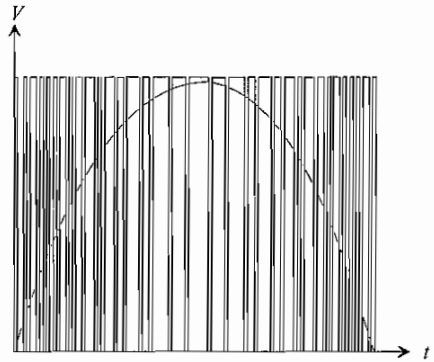


Figure 29.43 Pulse-density output from a sigma-delta modulator for a sine wave input.

If the frequency of the sine wave represented the highest frequency component of the input signal, a Nyquist rate converter would take only two samples. The oversampling converter, however, may take hundreds of samples over the same period to produce this pulse-density signal.

Digital signal processing is then used, which has two purposes: to filter any out-of-band quantization noise and to attenuate any spurious out-of-band signals. The output of the filter is then downsampled to the Nyquist rate so that the resulting output of the ADC is the digital data. This data represents the average value of the analog voltage over the oversampling period. The effective resolution of oversampling converters is determined by the values of signal-to-noise ratio and dynamic range obtained.

The First-Order-ΣΔ Modulator

Now that the basic function of the ΣΔ modulator has been described, it would be useful to examine its inner workings and determine why ΣΔ modulation is so beneficial for generating high-resolution data. A basic first-order ΣΔ modulator can be seen in Fig. 29.44. Here, an integrator and a 1-bit ADC are in the forward path, and a 1-bit DAC is in the feedback path of a single-feedback loop system. The variables labeled are in terms of time, T, which is the inverse of the sampling frequency and k, which is an integer. The 1-bit ADC is simply a comparator that converts an analog signal into either a high or a low. The 1-bit DAC uses the comparator output to determine if $+V_{REF}$ or $-V_{REF}$ is summed with the input.

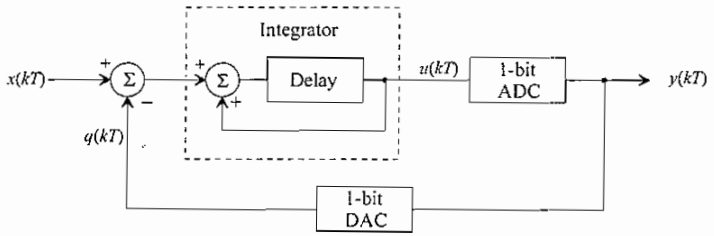


Figure 29.44 A first-order sigma-delta modulator.

While the benefits of ΣΔ modulation are not obvious, a simple derivation of the output, $y(kT)$, illuminates its distinct advantages. The output of the integrator, $u(kT)$, can be described as

$$u(kT) = x(kT - T) - q(kT - T) + u(kT - T) \tag{29.95}$$

where, $x(kT - T) - q(kT - T)$ is equal to the integrator's previous input, and $u(kT - T)$ is its previous output. The quantization error for the 1-bit ADC, as discussed in Ch. 28, is again defined as the difference between its output and input such that

$$Q_e(kT) = y(kT) - u(kT) \tag{29.96}$$

Plugging Eq. (29.95) into Eq. (29.96), the output response, $y(kT)$ is

$$y(kT) = Q_e(kT) + x(kT - T) - q(kT - T) + u(kT - T) \tag{29.97}$$

An ideal 1-bit DAC has the following characteristic: if the input, $y(kT) = 0$, the output, $q(kT) = -V_{REF}$, and if $y(kT) = 1$, then $q(kT) = V_{REF}$. In reality, a 1-bit DAC consists of a couple of switches connecting V_{REF} or $-V_{REF}$ to a common node, so it is not difficult to assume that the DAC is ideal. Therefore,

$$y(kT) = q(kT) \tag{29.98}$$

Utilizing Eq. (29.96) and Eq. (29.97), we find that Eq. (29.98) becomes

$$y(kT) = x(kT - T) + Q_e(kT) - Q_e(kT - T) \tag{29.99}$$

Therefore, the output of the modulator consists of a quantized value of the input signal delayed by one sample period, plus a differencing of the quantization error between the present and previous values. Thus, the real power of ΣΔ modulation is that the quantization noise, Q_e, cancels itself out to the first order.

A frequency domain example further illuminates this important fact. Suppose that the first-order modulator can be modeled in the s domain, as seen in Fig. 29.45, with an ideal integrator represented with transfer function of $\frac{1}{s}$, the 1-bit ADC modeled as a simple error source, $Q_e(s)$, and again the DAC considered to be ideal, such that $y(s)$ is equal to $q(s)$. It is also assumed that the bandwidth of the input signal is much less than the bandwidth of the modulator. Therefore, using simple feedback theory, $v_{OUT}(s)$ becomes

$$v_{OUT}(s) = Q_e(s) + \frac{1}{s} \cdot [v_{IN}(s) - v_{OUT}(s)] \tag{29.100}$$

and solving for v_{OUT} yields,

$$v_{OUT}(s) = Q_e(s) \cdot \frac{s}{s+1} + v_{IN}(s)\frac{1}{s+1} \tag{29.101}$$

Note that the transfer function from v_{IN} to v_{OUT} follows that of a low-pass filter and that the transfer function of the quantization noise follows that of a high-pass filter. Plotted together in Fig. 29.46, it is seen that in the region where the signal is of interest, the noise has a small value while the signal has a high gain, and that at higher frequencies, beyond the bandwidth of the signal, the noise increases. The modulator has essentially pushed the power of the noise out of the bandwidth of the signal. This high-pass characteristic is known as *noise shaping* and is a powerful concept used within oversampling ADCs. Low-pass filtering is then performed by the digital filter in order to remove all of the out-of-band quantization noise, which then permits the signal to be downsampled to yield the final high-resolution output.

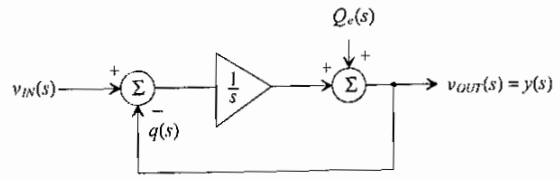


Figure 29.45 A frequency domain model for the first-order sigma-delta modulator.

As the ΣΔ modulator is generating the pulse-density modulated output, it is interesting to examine the mechanics occurring in the loop, which result in an average of the input. An actual ΣΔ modulator might resemble Fig. 29.47. A switched-capacitor integrator provides the summing as well as the delay needed. The 1-bit ADC is a simple comparator, and 1-bit DAC is simply two voltage-controlled switches that select either V_{REF} or $-V_{REF}$ to be summed with the input. A latched comparator provides the necessary loop delay. Notice that the variables are voltage representations of the variables used in

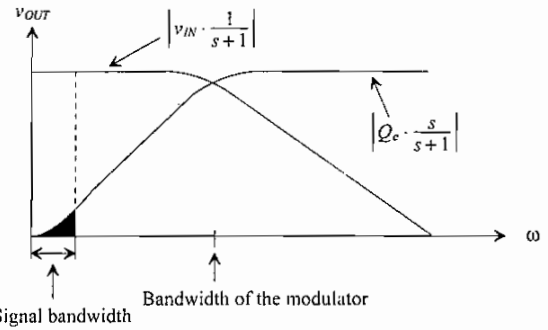


Figure 29.46 Frequency response of the first-order sigma-delta modulator.

Fig. 29.44. Remember that the function of the integrator is to accumulate differences between the input signal and the output of the DAC. If it is assumed that the input, $v_x(kT)$, is a positive DC voltage, then the output of the integrator should increase. However, the feedback mechanism is such that the 1-bit ADC (the comparator) has a low output if the integrator output, $v_u(kT)$, is positive. Thus, V_{REF} appears at the output of the DAC and is subtracted from the input, and the integrator output is driven back toward zero. The opposite occurs when $v_u(kT)$ is negative such that the integrator output is always driven toward zero by the feedback mechanism. An example will illustrate the operation of the modulator in more detail.

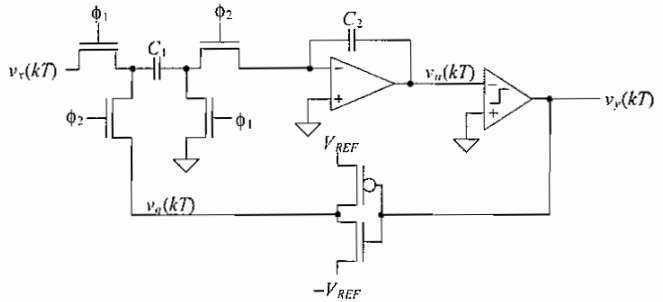


Figure 29.47 Implementation of a first-order sigma-delta modulator using a switched capacitor integrator.

Example 29.16

Using a general first-order ΣΔ modulator, assume that the input to the modulator, $v_x(kT)$ is a positive DC voltage of 0.4 V. Show the values of each variable around the ΣΔ modulator loop and prove that the overall average output of the DAC

approaches 0.4 V after 10 cycles. Assume that the DAC output is ±1 V, and that the integrator output has a unity gain with an initial output voltage of 0.1 V, and that the comparator output is either ±1 V.

The present integrator output will be equal to the sum of the previous integrator output and the previous integrator input. Therefore, Eq. (29.95) becomes

$$v_u(kT) = v_u(kT - T) + v_a(kT - T) \tag{29.102}$$

where

$$v_a(kT) = v_x(kT) - v_q(kT) \tag{29.103}$$

and the quantizing error, $Q_e(kT)$, is defined by Eqs. (29.96) and (29.98) as

$$Q_e(kT) = v_q(kT) - v_u(kT) \tag{29.104}$$

The initial conditions define the values of the variable for $k = 0$. The output of the integrator is given to be 0.1 V. Thus, the ADC output is low, the DAC output is V_{REF}, and the output of the summer, $v_a(0)$, is $0.4 - V_{REF} = -0.6$ V.

The output for $k = 1$ begins again with the integrator output. Using Eq. (29.102), $v_u(kT)$ becomes

$$v_u(T) = 0.1 + (-0.6) = -0.5 \text{ V}$$

Since the output of the integrator is negative, the output of the comparator is positive and $-V_{REF}$ is subtracted from 0.4 to arrive at the value for $v_a(T)$.

Continuing in the same manner and using the previous equations, we note the voltages for each cycle in Fig. 29.48. After 10 cycles through the modulator, the average value of $v_q(kT)$ becomes,

$$\overline{v_q(kT)} = \frac{7-3}{10} = 0.4 \text{ V}$$

k	$v_a(kT)$	$v_u(kT)$	$v_q(kT) = v_y(kT)$	$Q_e(kT)$	$\overline{v_q(kT)}$
0	−0.6	0.1	1.0	0.9	1.0
1	1.4	−0.5	−1.0	−0.5	0
2	−0.6	0.9	1.0	0.1	0.333
3	−0.6	0.3	1.0	0.7	0.50
4	1.4	−0.3	−1.0	−0.7	0.20
5	−0.6	1.1	1.0	−0.1	0.333
6	−0.6	0.5	1.0	0.5	0.429
7	1.4	−0.1	−1.0	−0.9	0.25
8	−0.6	1.3	1.0	−0.3	0.333
9	−0.6	0.7	1.0	0.3	0.40

Figure 29.48 Data from the first-order ΣΔ modulator.

Notice that the behavior of $\overline{v_q(kT)}$ swings around the desired value 0.4 V. If we were to continue computing values, as k increases, the amount that $\overline{v_q(kT)}$ differs from 0.4 V would decrease. Ideally, we could make the deviation of $\overline{v_q(kT)}$ as small as desired by allowing the modulator to take as many samples as necessary to meet that accuracy. ■

It is interesting to examine the effects of using a nonideal comparator. Suppose the integrator's output was smaller than the offset voltage of the comparator. A wrong decision would be made, causing $v_y(kT)$ to be the opposite of the desired value. However, as k increases, this error is averaged out, and the modulator still converges on the correct answer. Therefore, the comparator does not have to be very accurate in its ability to distinguish between two voltages, in contrast to Nyquist rate comparators.

The Higher Order ΣΔ Modulators

Higher order ΣΔ modulators exist which provide a greater amount of noise shaping. A second-order ΣΔ modulator can be seen in Fig. 29.49. A derivation of the second-order transfer function would reveal that the output contained a delayed version of the input plus a second-order differencing of the quantization noise, Q_e (see Problem 29.38). A third-order modulator would contain third-order differencing of the quantization and can be constructed by adding another integrator similar to integrator A into the system.

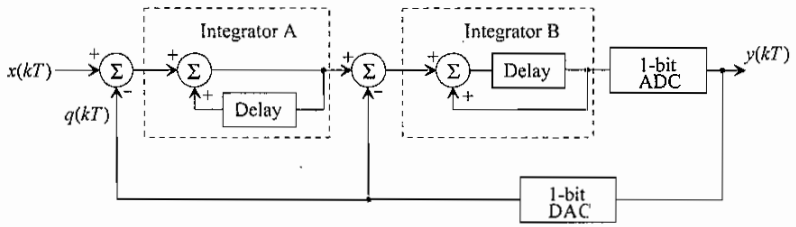


Figure 29.49 A second-order, sigma-delta modulator.

Figure 29.50 shows the noise-shaping functions of a first-, second-, and third-order modulator. The cross-hatched area under each of the curves represents the noise that remains in the signal bandwidth and is a magnified version of the blackened area of Fig. 29.46. As the order increases, notice that more of the noise is pushed out into the higher frequencies, thus decreasing the noise in the signal bandwidth. It should be reiterated that ΣΔ modulators do not attenuate noise at all. In fact, they add quantization noise that is very large at high frequencies. But because almost all of the noise is out of the signal bandwidth, it can easily be filtered, leaving only a small portion within the signal bandwidth. This point is important because the ΣΔ modulator should not be construed as a filtering circuit.

The resolution also increases as the order of the ΣΔ modulator and the oversampling ratio increases, as seen in Fig. 29.51 [20]. Using a first-order modulator, one can expect an increase in dynamic range of 9 dB with every doubling of the

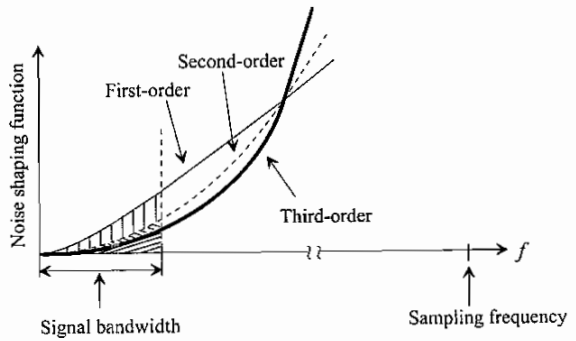


Figure 29.50 Noise shaping comparison of a first-, second- and third-order modulator.

oversampling ratio. This correlates to an approximate increase in resolution of 1.5 bits according to Eq. (28.28). The higher-order modulators have even greater gains in resolution as a 2.5-bit increase is attained with each doubling of the oversampling ratio using a second-order modulator, while the third-order modulator increases 3.5 bits.

One could essentially construct a high-order ΣΔ modulator with many integrators. However, as with any system employing feedback, stability becomes a critical issue. The same holds true for the high-order ΣΔ modulators. Several other topologies have been developed which can implement modulators in a cascaded fashion and are guaranteed to be stable [21, 22]. However, considerable matching requirements need to be overcome.

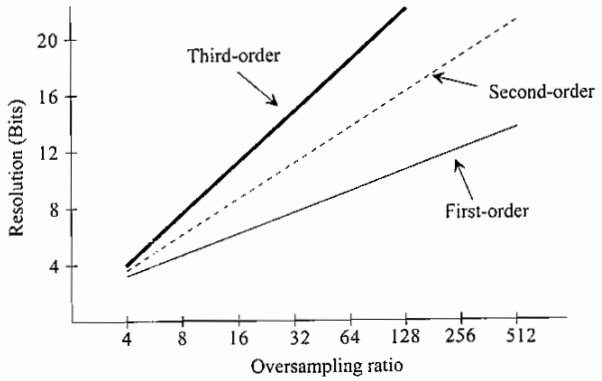


Figure 29.51 Comparison of first-, second-, and third-order modulators versus oversampling ratio and resolution.

REFERENCES

[1] R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI - Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Co., 1990.

[2] R. E. Suarez, P. R. Gray, and D. A. Hodges, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part II," *IEEE Journal of Solid State Circuits*, vol. 10, no. 6, pp. 379–385, December 1975.

[3] J. Shyu, G. C. Temes, and F. Krummenacher, "Random Errors in MOS Capacitors and Current Sources," *IEEE Journal of Solid State Circuits*, vol. 16, no. 6, pp. 948–955, December 1984.

[4] M. J. M. Pelgrom, et. al, "25-Ms/s 8-bit CMOS A/D Converter for Embedded Application," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 8, pp. 879–886, August 1994.

[5] D. Choi, et. al, "Analog Front-End Signal Processor for a 64 Mbits/s PRML Hard-Disk Drive Channel," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1596–1605, December 1994.

[6] N. Shiwaku, "A Rail-to-Rail Video-band Full Nyquist 8-bit A/D Converter," *Proceedings of the 1991 Custom Integrated Circuits Conference.*

[7] B. Razavi and B. A. Wooley, "A 12-b, 5-MSample/s Two-Step CMOS A/D Converter," *IEEE Journal of Solid State Circuits*, vol. 27, no. 12, pp. 1667–1678, December 1992.

[8] J. Dornberg, P. R. Gray, and D. A. Hodges, "A 10-bit, 5-Msample/s CMOS Two-Step Flash ADC," *IEEE Journal of Solid State Circuits*, vol. 24, no. 2, pp. 241–249, April 1989.

[9] T. Shimizu, et al., "A 10-bit, 20 MHz Two-Step Parallel A/D Converter with Internal S/H," *IEEE Journal of Solid State Circuits*, vol. 24, no. 1, pp. 13–20, February 1989.

[10] B. S. Song, S. H. Lee, and M. F. Tompsett, "A 10-bit 15 MHz CMOS Recycling Two-Step A/D Converter," *IEEE Journal of Solid State Circuits*, vol. 25, no. 12, pp. 1328–1338, December 1990.

[11] B. S. Song, M. F. Tompsett, and K. R. Lakshmikumar, "A 12-bit, 1-MSample/s Capacitor Error-Averaging Pipelined A/D Converter," *IEEE Journal of Solid State Circuits*, vol. 23, no. 6, pp. 1324–1333, December 1988.

[12] S. H. Lewis and P. R. Gray, "A Pipelined 5-Msample/s 9 bit Analog-to-Digital Converter," *IEEE Journal of Solid State Circuits*, vol. 22, no. 6, pp. 954–961, December 1987.

[13] S. Sutarja and P. R. Gray, "A Pipelined 13-bit, 250-ks/s, 5-V Analog-to-Digital Converter," *IEEE Journal of Solid State Circuits*, vol. 23, no. 6, pp. 1316–1323, December 1988.

[14] P. Vorenkamp and J. P. M. Verdaasdonk, "A 10 b 50 Ms/s Pipelined ADC," *IEEE ISSCC Digest of Technical Papers*, pp. 34–35, February 1992.

[15] M. Yotsuyanagi, T. Etoh, and K. Hirata, "A 10 Bit 50 MHz Pipelined CMOS A/D Converter with S/H," *IEEE Journal of Solid State Circuits*, vol. 28, no. 3, pp. 292–300, March 1993.

[16] J. L. McCreary and P. R. Gray, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part I," *IEEE Journal of Solid State Circuits*, vol. 10, no. 6, pp. 371–379, December 1975.

[17] K. Bacrania, "A 12 Bit Successive-Approximation ADC with Digital Error Correction," *IEEE Journal of Solid State Circuits*, vol. 21, no. 6, pp. 1016–1025, December 1986.

[18] M. Ismail and T. Fiez, *Analog VLSI Signal and Information Processing*, McGraw-Hill, 1994.

[19] B. E. Boser, "Design and Implementation of Oversampled Analog-to-Digital Converters," Ph.D. Dissertation, Stanford University, 1988.

[20] B. P. Brandt, *Oversampled Analog-to-Digital Conversion*, Integrated Circuits Laboratory, Technical Report No. ICL91-009, Stanford University, 1991.

[21] Y. Matsuya, K. Uchimura, et al, "A 16-bit Oversampling A/D Conversion Technology Using Triple Integration Noise Shaping," *IEEE Journal of Solid State Circuits*, vol. 22, no. 6, pp. 921–929, December 1987.

[22] K. Uchimura et al, "Oversampling A-to-D and D-to-A Converters with Multistage Noise Shaping Modulators," *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 1899–1905, December 1988.

[23] J. W. Bruce, "Meeting the analog world challenge: Nyquist rate analog to digital converter architectures," *IEEE Potentials*, vol. 17, no. 5, pp. 36–39, January 1999. Very good introduction to ADCs.

[24] J. W. Bruce, "Nyquist-rate digital to analog converter architectures," *IEEE Potentials*, vol. 20, no. 3, pp. 24–28, August 2001. Good overview of digital to analog converters.

[25] R. J. Baker, *CMOS Mixed-Signal Circuit Design*, John Wiley and Sons, 2002. ISBN 0-471-22754-4. Volume II of this book. Provides much more detail concerning the implementation of data converters discussed in this chapter and the characterizations presented in Ch. 28.

Special Thanks and Credit

The material presented in this chapter and Ch. 28 was written by Dr. Harry Li, a former Associate Professor of Electrical Engineering at the University of Idaho. In 2002, he left academics to become a pastor of the Mosaic Church of Central Arkansas in Little Rock, a church that specializes in cross-cultural ministry. Special thanks and credit goes to Harry for authoring this material. For more information consult www.mosaicchurch.net.

The author would also like to extend special thanks to Dr. Terry Sculley. Dr. Sculley derived the INL and DNL calculations used in Ch. 29 (not a trivial task) while he was at Washington State University. Thanks and credit go to him for allowing us (again) to include this material in the second edition of the book.

PROBLEMS

29.1 A 3-bit, resistor-string DAC similar to the one shown in Fig. 29.2a was designed with a desired resistor of 500 Ω. After fabrication, mismatch caused the actual value of the resistors to be

$R_1 = 500, R_2 = 480, R_3 = 470, R_4 = 520, R_5 = 510, R_6 = 490, R_7 = 530, R_8 = 500$

Determine the maximum INL and DNL for the DAC assuming $V_{REF} = 5$ V.

29.2 An 8-bit resistor string DAC similar to the one shown in Fig. 29.2b was fabricated with a nominal resistor value of 1 kΩ. If the process was able to provide matching of resistors to within 1%, find the effective resolution of the converter. What is the maximum INL and DNL of the converter? Assume that $V_{REF} = 5$ V.

29.3 Compare the digital input codes necessary to generate all eight output values for a 3-bit resistor string DAC similar to those shown in Fig. 29.2a and b. Design a digital circuit that will allow a 3-bit binary digital input code to be used for the DAC in Fig. 29.2a. Discuss the advantages and disadvantages of both architectures.

29.4 Plot the transfer curve of a 3-bit R-$2R$ DAC if all Rs = 1.1 kΩ and $2R$s = 2 kΩ. What is the maximum INL and DNL for the converter? Assume all of the switches to be ideal and $V_{REF} = 5$ V.

29.5 Suppose that a 3-bit R-$2R$ DAC contained resistors that were perfectly matched and that $R = 1$ kΩ and $V_{REF} = 5$ V. Determine the maximum switch resistance that can be tolerated for which the converter will still have 3-bit resolution. What are the values of INL and DNL?

29.6 The circuit illustrated in Fig. 29.5 is known as a current-mode R-$2R$ DAC, since the output voltage is defined by the current through R_F. Shown in Fig. P29.6 is an N-bit voltage-mode R-$2R$ DAC. Design a 3-bit voltage mode DAC and determine the output voltage for each of the eight input codes. Label each node voltage for each input. Assume that $R = 1$ kΩ and that $R_2 = R_1 = 10$ kΩ and $V_{REF} = 5$ V.

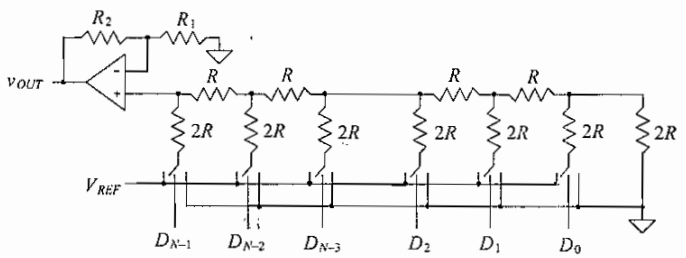


Figure 29.52 DAC used in Problem 29.6.

29.7 Design a 3-bit, current-steering DAC using the generic current-steering DAC shown in Fig. 29.9. Assume that each current source, I, is 5 mA, and find the total output current for each input code.

29.8 A certain process is able to fabricate matched current sources to within 0.05%. Determine the maximum resolution that a current-steering (nonbinary-weighted) DAC can attain using this process.

29.9 Design an 8-bit current-steering DAC using binary-weighted current sources. Assume that the smallest current source will have a value of 1 μA. What is the range of values that the current source corresponding to the MSB can have while maintaining an INL of ½ LSB? Repeat for a DNL less than or equal to ½ LSB.

29.10 Prove that the 3-bit charge-scaling DAC used in Ex. 29.6 has the same output voltage increments as the R-$2R$ DAC in Ex. 29.3 for $V_{REF} = 5$ V and $C = 0.5$ pF.

29.11 Determine the output of the 6-bit, charge-scaling DAC used in Ex. 29.7 for each of the following inputs: $D = 000010, 000100, 001000$, and 010000.

29.12 Design a 4-bit, charge-scaling DAC using a split array. Assume that $V_{REF} = 5$ V and that $C = 0.5$ pF. Draw the equivalent circuit for each of the following input words and determine the value of the output voltage: $D = 0001, 0010, 0100, 1000$. Assuming the capacitor associated with the MSB had a mismatch of 4 percent, calculate the INL and DNL.

29.13 For the cyclic converter shown in Fig. 29.17, determine the gain error for a 3-bit conversion if the feedback amplifier had a gain of 0.45 V/V. Assume that $V_{REF} = 5$ V.

29.14 Repeat Problem 29.13 assuming that the output of the summer was always 0.2 V greater than the ideal and that the amplifier in the feedback path had a perfect gain of 0.5 V/V.

29.15 Repeat Problem 29.13 assuming that the output of the summer was always 0.2 V greater than ideal and that the amplifier in the feedback path had a gain of 0.45 V/V.

29.16 Design a 3-bit pipeline DAC using $V_{REF} = 5$ V. (a) Determine the maximum and minimum gain values for the first-stage amplifier for the DAC to have less than ±½ LSBs of DNL assuming that the rest of the circuit is ideal. (b) Repeat for the second-stage amplifier. (c) Repeat for the last-stage amplifier.

29.17 Using the same DAC designed in Problem 29.16, (a) determine the overall error (offset, DNL, and INL) for the DAC if the S/H amplifier in the first stage produces an offset at its output of 0.25 V. Assume that all of the remaining components are ideal. (b) Repeat for the second-stage S/H. (c) Repeat for the last-stage S/H.

29.18 Design a 3-bit Flash ADC with its quantization error centered about zero LSBs. Determine the worst-case DNL and INL if resistor matching is known to be 5%. Assume that $V_{REF} = 5$ V.

29.19 Using the ADC designed in Problem 29.18, determine the maximum offset that can be tolerated if all of the comparators have the same magnitude of offset, but with different polarities, to attain a DNL of less than or equal to ±½ LSB.

29.20 A 4-bit Flash ADC converter has a resistor string with mismatch as shown in Table 29.1. Determine the DNL and INL of the converter. How many bits of resolution does this converter possess? $V_{REF} = 5$ V.

Resistor	Mismatch (%)
1	2
2	1.5
3	0
4	−1
5	−0.5
6	1
7	1.5
8	2
9	2.5
10	1
11	−0.5
12	−1.5
13	−2
14	0
15	1
16	1

Table 29.1 Mismatch in resistors used in Problem 29.20

29.21 Determine the open-loop gain required for the residue amplifier of a two-step ADC necessary to keep the converter to within ½ LSB of accuracy with resolutions of (a) 4 bits, (b) 8 bits, and (c) 10 bits.

29.22 Assume that a 4-bit, two-step Flash ADC uses two separate Flash converters for the MSB and LSB ADCs. Assuming that all other components are ideal, show that the first Flash converter needs to be more accurate than the second converter. Assume that $V_{REF} = 5$ V.

29.23 Repeat Ex. 29.12 for $V_{IN} = 3, 5, 7.5, 14.75$ V.

29.24 Repeat Ex. 29.13 for $V_{IN} = 1, 4, 6, 7$ V and $V_{REF} = 8$ V.

29.25 Assume that an 8-bit pipeline ADC was fabricated and that all the amplifiers had a gain of 2.1 V/V instead of 2 V/V. If $V_{IN} = 3$ V and $V_{REF} = 5$ V, what would be the resulting digital output if the remaining components were considered to be ideal? What are the DNL and INL for this converter?

29.26 Show that the first-stage accuracy is the most critical for a 3-bit, 1-bit per stage pipeline ADC by generating a transfer curve and determining DNL and INL for the ADC for three cases: (1) The gain of the first-stage residue amplifier set equal to 2.2 V/V, (2) the second-stage residue amplifier set equal to 2.2 V/V, and (3) the third-stage residue amplifier set equal to 2.2 V/V. For each case, assume that the remaining components are ideal. Assume that $V_{REF} = 5$ V.

29.27 An 8-bit single-slope ADC with a 5 V reference is used to convert a slow-moving analog signal. What is the maximum conversion time assuming that the clock frequency is 1 MHz? What is the maximum frequency of the analog signal? What is the maximum value of the analog signal which can be converted?

29.28 An 8-bit single slope ADC with a 5 V reference uses a clock frequency of 1 MHz. Assuming that all of the other components are ideal, what is the limitation on the value of RC? What is the tolerance of the clock frequency which will ensure less than 0.5 LSB of INL?

29.29 An 8-bit dual slope ADC with a 5 V reference is used to convert the same analog signal in Problem 29.27. What is the maximum conversion time assuming that the clock frequency is 1 MHz? What is the minimum conversion time that can be attained? If the analog signal is 2.5 V, what will be the total conversion time?

29.30 Discuss the advantages and disadvantages of using a dual-slope versus a single slope ADC architecture.

29.31 Repeat Ex. 29.15 for a 4-bit successive approximation ADC using $V_{REF} = 5$ V for $v_{IN} = 1, 3$, and full-scale.

29.32 Assume that $v_{IN} = 2.49$ V for the ADC used in Problem 29.31 and that the comparator, because of its offset, makes the wrong decision for the MSB conversion. What will be the final digital output? Repeat for $v_{IN} = 0.3025$, assuming that the comparator makes the wrong decision on the LSB.

29.33 Design a 3-bit, charge-redistribution ADC similar to that shown in Fig. 29.39 and determine the voltage on the top plate of the capacitor array throughout the conversion process for $v_{IN} = 2, 3$, and 4 V, assuming that $V_{REF} = 5$ V. Assume that all components are ideal. Draw the equivalent circuit for each bit decision.

29.34 Determine the maximum INL and maximum DNL of the ADC designed in Problem 29.33 assuming that the capacitor array matching is 1%. Assume that the remaining components are ideal and that the unit capacitance, C, is 1 pF.

29.35 Show that the charge redistribution ADC used in Problems 29.32 and 29.33 is immune to comparator offset by assuming an initial offset voltage of 0.3 and determining the conversion for $v_{IN} = 2$ V.

29.36 Discuss the differences between Nyquist rate ADCs and oversampling ADCs.

29.37 Write a simple computer program or use a math program to perform the analysis shown in Ex. 29.16. Run the program for $k = 200$ clock cycles and show that the average value of $v_q(kT)$ converges to the correct answer. How many clock cycles will it take to obtain an average value if $v_q(kT)$ stays within 8-bit accuracy of the ideal value of 0.4 V? 12-bit accuracy? 16-bit accuracy?

29.38 Prove that the output of the second-order ΣΔ modulator shown in Fig. 29.49 is,

$$y(kT) = x(kT - T) + Q_e(kT) - 2Q_e(kT - T) + Q_e(kT - 2T)$$

29.39 Assume that a first order ΣΔ ADC used on a satellite in a low earth orbit experiences radiation in which an energetic particle causes a noise spike resulting in the comparator making the wrong decision on the 10th clock period. Using the program written in Problem 29.37, determine the number of clock cycles required before the average value of $v_q(kT)$ is within 12-bit accuracy of the ideal value of 0.4 V. How many extra clock cycles were required for this case versus the ideal conversion used in Prob. 36?

Index

A

Abut, 98, 427, 429-430
AC small-signal analysis, 280
Accumulation, 132-134, 136, 159, 514
Active area, 83-99
  design rules, 99
Active-PI loop filter, 573-574
Active load, 657, 872
Active pixel sensor, 504-505
Adaptive biasing, 920-923
ADC, see Analog-to-digital converter
Adder, 367-369, 373, 395, 407-410, 412, 419-421
Aliasing, see Analog-to-digital converter
Amplifier,
  adaptive biasing, 920-923
  biasing, 291-292, 297-302, 613, 621, 624-625, 635-652, 699, 863-866
  body effect 691-692, 709
  cascode, 686-689, 696-697
  class A, 672, 688, 691, 697, 701, 727
  class AB, 672, 697-701, 709-710, 727-728, 800, 802-805, 810, 814, 817, 819-820, 865, 878
  common gate, 671, 689-690, 708-709, 732, 783
  common source, 657-658, 666-667, 671-673, 698-699, 717, 751, 756, 773-774, 793, 800, 814, 820-823
  current source load, 657, 665, 671-674, 686, 689-691, 698-699, 704, 719, 731-733, 774, 876
  distortion, 701, 704-706, 710, 737-740, 790, 807, 831
  efficiency, 699-702
  frequency response, 661-666, 674-685, 687, 696-697, 708, 721
  gain, 279, 291, 657-673, 686-692, 719-720
  input capacitance, 693
  noise, 219-262, 669-670, 686, 688, 694
  noise figure, 233-240
  overshoot, 811, 895, 936
  phase response, 662-665, 675-688
  pole splitting, 676-685
  push-pull, 698-703
  slew rate, 672-673
  source follower, 670, 690-698
  transimpedance, 666-667
Analog-to-digital converter, 931, 947-957, 985-1015
  0.5 LSB, 943
  aliasing, 953-955, 1008
  aperture error, 956-957
  architectures, 985-1016
  binary search, 1003-1005
  charge redistribution, 1005-1007
  coarse conversion, 991
  DFT, see discrete Fourier transform
  differences of oversampling and Nyquist rate converters, 1008-1009
  differential nonlinearity (DNL), 950-951, 989-990, 998, 1007
  discrete Fourier transform (DFT), 218, 956
  fine conversion, 991
  flash, 985-990
  flash accuracy issues, 988-990
  folding frequency, 955
  gain error, 953, 992-994
  higher-order sigma-delta modulation, 1014
  integral nonlinearity (INL), 951-952, 989, 997-998, 1007
  least significant bit (LSB), 940
  missing codes, 951
  noise shaping, 1011-1012
  Nyquist rate, 1007
  offset, 953, 989-990, 996-998, 1007
  operational amplifier issues, 992-994
  oversampling, 1007-1015
  parallel, feed-forward, 990
  pipeline, 994-998
  pipeline accuracy issues, 996-998
  pulse-density modulation, 1009
  quantization error, 948-949
  quantization levels, 947-949
  quantization noise, 956
  residue, 994
  resolution versus oversampling, 1014-1015
  sampling function, 954-955
  SAR, 1003-1005
  scale factor error, 953
  sigma-delta modulation, 1009
  signal-to-noise ratio (SNR), 956
  specifications, 947
  successive approximation, 1003-1007
  successive approximation accuracy issues, 1007
  successive approximation register, see SAR, 1003-1005
  transfer curve, 948-953, 987, 994

two-step, 990-992
two-step accuracy issues, 992
Analog models, 269-310
body effect transconductance, 287-288
transconductances, 279-280, 288
Analog signal characteristics of, 932
Anisotropic etch, 171
Anode, 43, 757, 760
AOI logic, 364-369
Aperture error, see sample-and- hold
Arbiter, 383
Avalanche breakdown, 153, 253-254
Averaging noise,
flicker, 246
thermal (white), 240-241
Averaging circuit, 539-540, 873-874

B

β (transconductance parameter), 142
Back end of the line (BEOL), 90, 177, 199-208
Backend processes, 208-211
Bandgap energy of silicon, 759
Bandgap voltage references (BGR), 745, 761-770
Battery, 547, 695, 730
BCD, see binary coded decimal
BEOL, see back end of the line
Beta-multiplier reference, 624-630, 647, 650, 750-754
compensation, 630
Biasing,
general long channel analog design, 291-293
general short channel design, 297-300
high-speed design, 299-302, 863-866
long-channel circuit, 647
push-pull amplifier, 699
short-channel circuit, 650
Binary coded decimal, 965
Bi-phase, 584-585
Bipolar junction transistor (BJT), 47, 150
lateral, 758
parasitic, 2, 37, 56, 341-342, 757-758
resistor in SRAM, 464
temperature behavior, 758-759
vertical, 757
Bird's beak, 181
BJT, see bipolar junction transistor
Bloats, 35
BMR, see beta-multiplier reference
Body effect, 139, 148-149, 276-277, 691-692, 709
gamma, γ, 139, 276
Body transconductance, 287-288
Bonding pad, 4, 59
layout, 59-62, 100-102
Bottom plate capacitance, 115, 839-840
Bottom plate sampling, 839-842
Breakdown
oxide, 154
voltage, 58, 143
BSIM4, 154-157
Buffers,
class AB output, 672, 697-701, 709-710, 727-728, 800, 802-805, 810, 814, 817, 819-820, 865, 878-891, 900-901
digital, 344-348
fully differential input, 920
input, 534-542
NMOS only, 350
op-amp, 793-810, 865, 878-891, 900-901
source follower, 696-697
Bulk CMOS process, 31, 180
Bus, 68, 100-103, 321, 351, 412-419, 427, 527

C

C'_{ox}, 114-115, 123-124, 132-135
Calma stream format, see GDS, 21, 36
Caltech intermediate format, see CIF, 21
Capacitance, 44, 135
bottom plate capacitance, 115, 839-840
bottom and sidewall, 117
depletion, 43-45, 47-49, 58, 94, 100, 117, 122, 133, 147, 244, 315, 342-343, 398, 478, 546, 618, 844
diffusion, 45-47, 117
junction, see depletion
metal layer parasitics, 61
MOSFETs 116-118, 132-135
sidewall, 44-45, 117
Capacitive feedthrough, 449, 831-832
Capacitor
common-centroid layout, 980
depletion, 43
poly-poly, 113-115
layout, 113-115, 851-852
parasitics, 61, 115
temp co, 116
voltage coefficient, 116
Carrier lifetime, τ_r, see diode, minority carrier lifetime
Carrier transit time, τ_T, see diode, minority carrier lifetime
Cascode,
amplifier, 686-689, 696-697
current mirror, 636-652
differential amplifier, 733-735, 793-794, 877-891
origin of the name, 636
Cathode, 43, 757, 760
CGBO, 135
CGDO, 123-124, 135
CGSO, 123-124, 135
Channel hot electron (CHE), 468-469
Channel-length modulation (CLM), 143-147, 273, 277, 280-281, 289, 613-617, 639, 856
Charge injection, 830-840, 856, 863, 937

Charge pumps, 329, 461, 542-546, 558-561, 578-580, 591-593, 600, 603
CHE, see channel hot electron
Chemical/mechanical polishing (CMP), 163, 170, 173, 182, 185, 187, 201-207, 211
Chemical vapor deposition (CVD), 176-177, 179, 182, 184-185, 187, 190, 193, 196, 199, 201-207
Chip, 3, 32, 36
cross-sectional view, 4, 32
design, 6
IC design flowchart, 2
packaging, 4, 211
layout view, 4, 32
organization, 447
CIF, 21, 36, 53, 78
Class AB, 672, 697-701, 709-710, 727-728, 800, 802-805, 810, 814, 817, 819-820, 865, 878
CLM, see Channel length modulation
Clock feedthrough, 449, 456, 830, 832-834, 838-840, 860, 863, 937
Clock generation, 577
nonoverlapping, 401-402, 843, 850, 853-856, 874, 896
Clock recovery circuit, 551-552, 584, 588-589, 591, 602-607, 608, 611
self-correcting, 588-594
Clock synchronization, 383, 551, 591, 596
Clocked logic, 397, 403-408
CMOS, 1
active pixel sensor, 504-505
IC design process, 1-2
passive elements, 105
process flow, 91
process integration, 161
scaling, 54, 152
trends, 152
twin-tub, 31
unit processes, 161
CMOS fabrication, 161-212
APCVD, 176
back end of the line (BEOL), 90, 177-178, 199-208
backend processes, 208-211
BEOL, see back end of the line
boule, 163
burn-in, 208, 211
chemical/mechanical polishing (CMP), 163, 170, 173, 182, 185, 187, 201-207, 211
chemical vapor deposition (CVD), 176-177, 179, 182, 184-185, 187, 190, 193, 196, 199, 201-207
critical dimension (CD), 179
contact module, 202-203
contaminates, 171, 208
Czochralski (CZ) growth, 162-163
degree of anisotropy, 170-171
deposition, 90, 161, 173-177, 179, 191-192, 196, 200-208, 210-211
depth of focus (DOF), 168-169, 201, 205
die seperation, 3, 211
diffraction effects, 168-169
diffusion, 117, 163, 165-167, 211
dry etching, 170-171, 208
doping processes, 52-53, 161, 165-167
electrical coupling, 202, 205
electronic grade silicon (EGS), 162
electromigration problems, 203
etching, 33, 90-91, 93, 170-172, 175-176, 183, 190, 192, 200, 202, 208, 211, 444
etch rate, 170-171
etch profile, 171
EGS, see electronic grade silicon
epitaxial wafers, 31-32, 180
FEOL, see front end of the line
field oxide (FOX), 38, 83-84, 86-87, 90-91, 116-117, 181
front end of the line (FEOL), 90, 177-178, 180-199
final test, 208, 211
gate oxides, 163, 191-192
IMD, see intra-metal dielectric
intra-metal dielectric (IMD), 69, 178, 205-208
ion implantation, 140, 165-167, 170, 184, 211
ion implanter, 165
implant profile, 165-166
isolation, 90-91, 173, 177, 181-183, 205
lattice damage, 165, 184
LOCOS, 181
mask alignment, 84, 86, 94, 170
masks, 34-35, 53, 69, 84, 86, 163, 167-168, 170, 173, 179
metallization, 178, 203-205
metallurgical grade silicon (MGS), 162
MGS, see metallurgical grade silicon
n-wall, 184-186
n-well, 179, 181, 184, 187-190
numerical aperture, 168
p-wall, 184-186
p-well, 179, 181, 184, 187-191
packaging, 3-5, 59, 208, 211
passivation, 62, 173, 178-179, 208-210
pattern transfer, 168
PECVD, 176-177
PECVD reactor, 177
photolithography, 161, 167-171, 173, 182-193, 196, 202-203, 206, 208-211
physical vapor deposition (PVD), 175-176, 200, 204, 208
plasma etching, 173-177
polysilicon, 166, 178-179, 191-194
pre-metal dielectric (PMD), 178, 200-204
process description, 89-91, 178-179
process integration, 177-208
rapid thermal annealing, 199-201
rapid thermal processing (RTP), 164
reactive ion etching (RIE), 171-173

registration errors, 170
resolution, 168-169, 171
reticle, see mask
salicide module, 199
scribe lines, 75, 209, 211
selectivity, 170
shallow trench isolation (STI), 90-91, 177-178, 180-188
silica, 161-162
silicide, 83, 88-92, 94, 130, 199-201
SOI wafers, 180
source drain module, 193-199
starting material, 180-181
step coverage, 174
stepper, 34, 167-168
straggle, 165
streets, see scribe, 209, 211
technology computer assisted design (TCAD), 179
thermal oxidation, 34, 161, 163-164, 182, 184, 190, 192, 211
thin film deposition, 173-177
thin film removal, 161, 170-171
trench liner, 184
twin tub (well) module, 187-190
unit processes, 161-177
via 1 module, 205-207
wafer manufacture, 161-163
wafer probe, 208-211
wet etching, 170-171
yield, 152, 209-211
CMOSEDU design rules, 53-55, 69, 79
CMFB, see common-mode feedback
CMRR, see common-mode rejection ratio
Common-centroid, 111-113, 618, 724, 851, 916, 980
Common drain amplifier, see source follower
Common gate amplifier, 671, 689-690, 708, 732, 783
Common-mode feedback (CMFB), 836-837, 863, 869-875, 881-885, 888-895, 896-904, 906-908
compensating, 871-873
settling time, 895
switched-capacitor, 874-875, 896-904, 907-908
Common-mode rejection ratio (CMRR), 721-724, 742, 773, 789-790, 823, 825, 833
Common-mode voltage (V_{CM}), 837
Common source amplifier, 657-658, 666-667, 671-673, 698-699, 717, 751, 756, 773-774, 793, 800, 814, 820-823
Communication system, 582-583
Comparators, 909-930
block diagram, 910
characterizing, 915-918
clamped input stage, 917
clocked, 918-918
dc response, 915-916
decision circuit, 910-914
dynamic, 854-856
gain and offset, 916
hysterisis, 912-914
input slew rate, 918
inverter based, 854-856
kickback noise, 448-449, 456, 910, 918, 930
operation, 909
output buffer, 913-915
pre-amp, 910-913
propagation delay, 918
self-biased, 920
sensitivity, 909-910, 930
transient response, 916
Compensation, 773-792, 794, 814-815, 819-820, 892, 895
beta-multiplier reference, 630
common mode feedback (CMFB), 871-873
constant-g_m, 738-740
high-speed, 783-787
indirect, 783-787, 791-792, 825
gain enhancement, 809-812
lead, 782
Miller, 774, 788
nested Miller, 820
pole splitting, 674-679, 685, 780, 783, 814, 821
slew-rate limitations, 787-789
Conductivity, 39-41
silicon-dioxide, 468
Constant-g_m,
bias circuit, 625
diff-amp, 738-740
Contact resistance, 47, 64, 70-71, 80, 147, 209, 962
Contact potentials, 42-43, 124, 137-138
Convergence, 27-28, 401
Complementary to absolute temperature (CTAT), 745, 759, 761-762, 765, 768
Counter, 329, 485-486, 506, 998-1002
as a digital filter, 506-507
Counter-doped, 52-53
Coupled noise rejection, 838
Critical dimension (CD), 179
Cross-sectional view, 3-5
Crosstalk, 59, 71-72, 427
CTAT, see complementary to absolute temperature
Current differential amplifier, 737, 921-922
Current mirrors, 273-274, 510, 512, 563, 613-656
biasing, 621-626, 647, 650, 863-866
cascode, 273-274, 636-646
design, 615
dynamic, 856-858
floating, 647, 651-652, 698-699, 703-704, 710, 804-805, 808, 820, 865
layout, 616-620, 642
low-voltage, wide-swing, 639-651
matching, 616-620
output resistance, 288, 298, 613, 622, 636-644
regulated drain, 645-646

SPICE statement, 620
subthreshold operation, 635
supply independant biasing, 624
temperature coefficient, 632
V_{THN} mismatch, 616
KP mismatch, 616-617
wide swing, 639-651
Current mirror load, 715-717, 731
Current starved VCO, 329, 561-565, 571, 574
CVSL logic, 369-370
Czochralski (CZ) growth, 162-163

D

DAC, see Digital-to-analog converter
Damascene process, 90, 207
Data conversion
signal characteristics, 931-935
DC sweep convergence, 27
Decoder, 329, 43-34, 447, 449, 457-462, 480, 966, 985-987
Decoupling capacitor, 73-74, 81, 417, 419
Delay time, 49-52, 65, 89, 317-318, 348, 358-359, 362, 526-527, 531-533
NAND gate, 392
phase shift, 665
TG, 376
word line, 443
Delay elements, 408, 595-596
Delay locked loop (DLL), 592-602
Delay & transition times 317-320
Delta-sigma modulation, see Sensing using delta-sigma modulation (DSM), see also Sigma-delta modulation
Depletion capacitance, 43-45, 47-49, 58, 94, 100, 117, 122, 133, 147, 244, 315, 342-343, 398, 478, 546, 618, 844
Depletion device, 140
Depth of focus (DOF), 168-169, 201, 205
Design rules, 53-54, 69, 75, 98-99
active and poly, 98-99
CMOSEDU, 53-55, 69, 79
n-well, 36-37, 53-54
metal layers, 69
DFT, see discrete Fourier transform
DIBL, see drain induced barrier lowering
Dickson's charge pump, 544
Dielectric constants, 114
Die, 3-4, 32, 59-60, 62, 68, 75, 110, 116, 209, 211, 411-412, 427, 429, 443, 746, 958, 960, 998
Die seperation, 211
Differential amplifier, 711-744
ac operation, 718-721
adaptive biasing, 920-922
body effect, 730, 742
cascode, 733-735
class AB, 727-733
common-centroid layout, 724
common mode range (CMR), 713-716, 732-733, 735
common mode rejection ratio (CMRR), 721-723
constant transconductance, 738-740
current, 737, 921-922
current mirror load, 715-717, 731-733
dc operation, 711
gain, 719, 722, 733, 736
low voltage, 735
matching, 724-726
minimum power supply voltage, 717
noise, 726-727
output resistance, 719, 733-734
self-biased, 534, 536, 537, 538, 920
slew rate, 727
source cross coupled pair, 727-733
wide swing, 736-740, 918-919
Differential output op-amp, 836-842, 863-908
benefits, 838
simulating, 841, 878-904, 927
Diffusion,
capacitance, 45-47, 117
process, 117, 163, 165-167, 211
Digital-to-analog converter, 931, 938-947, 965-985
0.5 LSB, 943
accuracy, 941
binary switch array, 966-967
binary weighted, 971, 974-975, 977-980
charge scaling, 978-982, 1005-1007
current source mismatch, 976-977
current steering, 973-975
cyclic, 982-983, 1019
decoder, 966
differential nonlinearity (DNL), 941-943, 963, 967-971, 976-980
dynamic range, 947, 963
full scale voltage, 939, 941, 963
gain error, 945-946, 953, 1019
input combinations, 938
integral nonlinearity (INL), 943-945, 967-971, 976-977, 979, 980
ladder, 967, 971
latency, 945
least significant bit (LSB), 940-941
LSB, see least significant bit
monotonicity, 943
most significant bit (MSB), 940
MSB, see most significant bit
nonmonotonic, 943
offset, 943-946
op-amp issues, 992-994
pipeline, 984-985
R-2R ladder, 971-973
reference voltage, 938-941, 956
resistor string, 966-971
resistor string mismatch, 967-970
resolution, 933-934, 937-938, 940-943, 946-947
settling time, 945

signal-to-noise ratio (SNR), 945-946, 956
specifications, 938-947
split-array, 980-981, 1019
transfer curve, 940
Digital models, 311-319
Digital phase-locked loop (DPLL), 551-612
active-PI loop filter, 573-574, 581, 588, 591
block diagram, 552
charge pump, 558-561, 578-580, 591-593, 600, 603
dead zone, 577, 600, 609
delay elements, 595-596
examples, 596-607
jitter, 557, 564, 571, 579, 582, 591-592, 594, 597, 600
lock range, 567, 569-570, 573-574, 576-577, 579-580, 592
loop filter, 552, 554-557, 559-561, 567-582
PFD, 552-553, 557-561, 575-580, 591, 600-603, 608
pull-in range, 567-569, 571, 574, 576-577, 579, 592
static phase error, 572-573, 577, 579, 600, 604, 610
system concerns, 582
tristate output, 559-560, 575-579, 581
XOR DPLL, 568-574
XOR Phase Detector, 552-557, 559, 561
Diode,
anode, 43, 757, 760
breakdown voltage, 58, 253
carrier transit time, τ_T, see minority carrier lifetime
cathode, 43, 757, 760
depletion capacitance, 43-45, 117
diffusion capacitance, 45-46, 117
drain or source to substrate, 47, 117, 143, 398, 403, 543, 618, 937
electrostatic discharge, 100-102
ESD, see diode, electrostatic discharge
long base, 45
minority carrier lifetime, τ_T, 45-48, 242-243
n-well/p-substrate, 31-32, 39-45, 53, 57-58, 94, 96, 543
noise, 242-244, 249, 252-253
parasitic BJT, 757-759
photodiode, 249, 504
reverse recovery time, 46-49
saturation (scale) current, I_S, 39, 47, 181, 398
Schottky, 92, 760
short base, 45
SPICE modeling, 47-49
storage capacitance, see diffusion capacitance
storage time, 46-49
temperature behavior, 759
t_{rr}, see reverse recovery time
t_s, see storage time
τ_T, see minority carrier lifetime
Zener, 253-254
Discrete Fourier transform (DFT), 218, 956
Distortion,
amplifier, 701, 704-706, 710, 737-740, 807, 993-994
common-mode level variation, 740, 790, 807
multiplier, 923
pulse width, 541, 582-583
signal-to-noise plus distortion ratio (SNDR), 956
switches, 831
THD, see total harmonic distortion
total harmonic distortion, 704-706, 923
Distortionless transmission, 582
DLL, see Delay-locked loop
Dominant pole, 680
Domino logic, 405-407, 409
Doping processes, 52-53, 161, 165-167
Doublet, 810
DPLL, see Digital phase-locked loop
Drain induced barrier lowering (DIBL), 154
DRAM, see memory, DRAM
Drawn length, 87, 116-117, 132, 144
Drawn width, 68, 87, 116-117, 132, 315
DRC (design rule check), 36, 52
Drive current, see on current
Drivers, 344, 449, 459-463, 481, 546
distributed, 347-348
Dry etching, 170-171, 208
DSL logic, 370-372
DSM, see Sensing using delta-sigma modulation (DSM)
Dummy elements, 113, 618, 851
Dynamic circuits, 829-862
amplifiers, 858
capacitive feedthrough, 67, 830-832
charge injection, 830, 856, 830-833
clock feedthrough, see capacitive feedthrough
comparator, 854-856
current mirrors, 856-858
dummy switch, 832
IOS, 856-857
kT/C noise, 833-834, 856
logic, 397-410
OOS, 856-857
op-amp settling time, 852-853
reducing charge injection & clock feed-through, 832
reducing op-amp offset-voltage, 853-854
sample-and-hold, 838-840
simulating, 401
storage capacitors, 852-853, 856
switch, 829-834
switched capacitor integrator, 845-850
switched capacitor resistor, 843-845
Dynamic logic, 397-410
charge leakage, 399-400
clocked CMOS, 403-408

domino logic, 405-406
PE logic, 405
NP logic, 407
shift register, 402
simulating, 401
zipper logic, 407
Dynamic power dissipation, 339-340
Dynamic range, 490, 506, 833, 839, 947, 963, 1009, 1014

E

Edge detector, 374, 587
Effective channel length, 132
Effective switching resistance, 312-315, 317, 320, 335, 344, 390, 392
Electrode layer, 115
Electronic grade silicon (EGS), 162
Electrical length, 144
Electromigration, 59, 68, 70
Electrostatic discharge (ESD),
diode, 59, 100-103, 530-531, 549
Energy,
bands in silicon, 39-43
electrical, 213-215, 217
implant, 166
thermal, 225-226, 236
Energy spectral density (ESD), 217
Enhancement device, 140
Epi layer, 31
Epitaxial wafers, 31-32, 180
EGS, see electrical grade silicon
ESD, see electrostatic discharge, see also energy spectral density
Etching, 33, 90-91, 93, 170-172, 175-176, 183, 190, 192, 200, 202, 208, 211, 444
Etch profile, 171
Equalizer, 539, 582-583, 610
Excess gate voltage, see overdrive voltage, 271
Excess phase shift, 665

F

Fall time, 317-318
Fermi energy level, 42-43
FEOL (front end of the line), 90, 177-178, 180-199
Field device, 38, 181, 184-185
Field implant, 38-39
Field oxide, see FOX (field oxide)
Flash memory, 113, 463, 469-476, 481-482, 487-489
Flatband voltage, 139
Flicker noise, 213, 244-253, 261-262, 267
averaging, 246
MOSFET, 302-303, 833
Flip-flops, 380-389
data (D), 386-388
hold time, 388-389
metastability, 383-385, 396
setup time, 388-389
Floating current source, 647, 651-652, 698-699, 703-704, 710, 804-805, 808, 820, 865
Floating gate, 113, 466
Floor plan, 412, 958
FNT, see Fowler-Nordheim tunneling
Folded-cascode, 641, 648-649, 714-715, 802-810, 827-828, 872
Fowler-Nordheim tunneling (FNT), 469-474, 487
FOX (field oxide), 38-39, 83-87, 181
FPGA (field programmable gate arrays), 1, 412
Frequency response,
amplifier, 661-666, 674-688, 696-697
doublet, 810
noise, 220-223
op-amp, 680, 792-793
pole splitting, 674-679, 685, 780, 783, 814, 821
transmission channel, 538-541
Frequency synthesis, 577, 591
f_T, unity current gain frequency, 290-292, 297-302, 309, 316, 863, 903, 906
Fully-differential circuits, 833, 836-842, 863-930
benefits, 833, 838
bottom plate sampling, 839-842
common-mode feedback, 836-837, 863, 869-875, 881-885, 888-895, 896-904, 906-908
delay elements, 595-602
gain, 836
noise rejection, 838
sample and hold, 838-842, 880-881, 901-904
SPICE model, 841-842
f_{un}, op-amp unity-gain frequency, 260-261, 680-685, 740, 779-781, 785-786, 792-793, 797-799, 852-853, 993
gain-enhancement, 808-812
plot (general), 680

G

Gain-enhancement (GE), 808-812, 889-890
Gain-f_T product (GFT), see MOSFET, GFT (gain-f_T product)
Gain margin, 781
Gamma, γ, 139, 276
Gaussian probability distribution function, 248-249
GDS, 21-23, 53, 78
GE, see gain-enhancement
Generation, 40
GFT, see MOSFET, GFT (gain-f_T product)
g_m, 279-280
long-channel equation, 280
short-channel, 293, 297-299
subthreshold, 292-293
GM, see Operational amplifier, gain margin
g_{mb}, 289-290
GMIN, 28, 401
GOX (gate oxide), 86-87, 114
Gradual channel approximation, 140, 151

Graphical design format, see GDS, 21, 36
Gray code, 965-966
Ground bounce, 71-74, 81
Guard rings, 110, 343, 757, 851, 957, 960

H

Harmonic distortion, see Distortion, 704-706, 831, 923, 993-994
Hi-Z, 321-322, 350-351, 370, see also Tri-state outputs
Hierarchy, 11, 14, 19
High-speed op-amp, 783-787, 891
High impedance node, 397, 629-630, 673, 696, 721, 796, 819
Hold time, 388-389
Hot-carrier effect, 151, 153
Hysteresis, 523-529, 537-538, 547-548, 910-916, 918

I

I_{drive}, see on current
$I_{D,sat}$, 144-145, 158, 271-272
Imaging,
  CMOS active pixel sensor, 504-505
  noise, 246
  sensing, 504-519
Inductance, 72
Incomplete settling, 496, 502-503, 520
Input buffers, 531-542, 920
Input offset storage (IOS), 856-857
Input-referred offset, see Offset voltage
Instanced (or placed), 429
Integrator,
  ADC, 998-1002
  delta-sigma, 492, 1010-1014
  noise, 245, 267
  offset, 846
  PLL, 573, 575
  switched capacitor, 845-850, 861, 901, 907-908
Inter layer dielectric (ILD), 69, 81
Interconnect burden, 429
Interdigitated layout, 110-113, 960
Intrinsic,
  carriers, 40-41
  propagation delay, 65-66, 81
  silicon, 40-42, 52-53
Inverter, 331-352
  buffer design, 344-347
  dynamic power dissipation, 339-340
  layout, 341-343, 413, 422-425
  noise margins, 333
  other configurations, 349-351
  power delay product, 340
  switching characteristics, 337-338
  sizing, 344
  transfer characteristics, 332-336
  tri-state output, 351
  VTC, see transfer characteristics
I_{off}, see off current
I_{on}, see on current
Ion implantation, 165
I_S, see Diode, saturation current
Isotropic etch, 171

J

Jitter, 557, 564, 571, 579, 582, 591-592, 594, 597, 600, 1000
  peaking, 582
J_s, scale current density, 57
Junction capacitance, see depletion capacitance, 43

K

Keeper MOSFET, 406-407, 461
Kickback noise, 448-449, 456, 910, 918, 930
KP, 142
kT/C noise, 229, 240, 244, 250-251, 517, 833-834, 846, 856, 896

L

Lambda, λ,
  channel-length modulation (CLM), 144, 147, 289, 298, 617, 620
  layout (in the MOSIS rules), 31, 54
  wavelength, 168
LASI, 1, 6-23, 52-55, 75-79, 124-129
  backups, 19-20
  basics, 6-23
  box, 8-9
  cells, 7, 11-14
  CIF, 21, 36, 53, 78
  common problems, 20-21
  copying, 18-19
  cursor, 21
  design examples, 124-129
  drawing grid, 10
  edit-in-place, 19-20
  frustration, 14
  GDS, 21-23, 53, 78
  grid (dot), 8, 10
  help, 6-7, 28
  hierarchy, 11, 14, 19
  invisible cells, 14, 21
  key assignments, 11
  measurements, 10
  navigating, 8
  path, 15-16
  polygon, 15-16
  problems, 20-21
  ranking cells, 11
  reference marker, 7
  saving your work, 20
  scaling, 21-23
  schematics, 124-127

  speed-up, 17-18
  text, 16-17, 78-79
  TLC/TLD, 19
  unit grid, 10
  vertices, 15-17
  working grid, 10
LASICKT, 78-79, 124-127
LASIDRC, 52-55
Latch, 380-386, 400, 403-404
Latch-up, 2, 46, 180, 341-343, 349-350
Lateral diffusion,
  LOCOS, 181
  MOSFET source/drain, 116-117, 124, 144, 315, 616, 620
  n-well fabrication, 35-36, 56
  SPICE, 147
Layout,
  active, 83-85
  adder, 420-421
  binary-weighted capacitor array, 980
  bipolar junction transistor, 757-758
  bonding pad, 59-60, 62, 75-77
  capacitor, 113-116, 419, 851,
  capacitor bottom plate, 115, 840
  circle, 29
  common-centroid, 111-113, 724
  corners, 38
  design rules, 37, 68, 99
  diff-pair, 724
  diode, 757-760
  DRAM cell, 443-446
  drawing order, 4
  dummy elements, 113, 618, 851
  electrode, 115
  flash memory cell, 466-471
  guard rings, 110, 343, 757, 851, 957, 960
  hierarchy, 11, 14, 19
  inverter, 341-343, 413, 422-425
  large-width MOSFETs, 121
  lateral bipolar, 757-758
  long-length MOSFETs, 120
  matching (to improve), 617-620, 724
  MOSFET, 86-87, 95 (NMOS), 96 (PMOS) 116-124, 127, 617
  MUX/DEMUX, 422
  n-well, 32, 36-37, 94
  NAND and NOR gates, 358, 415
  pad, 60, 75, 102
  parasitic pnp, 757
  power and ground, 417
  resistor, 32, 94, 109-113
  resistor divider, 125
  Schottky diode, 760
  serpentine pattern, 57
  SR latch, 416-417
  SRAM, 463-464
  standard-cell, 97-98, 343, 413, 427-431
  standard cell frame, 97, 343
  stick diagram, 422
  TG (transmission gate), 415
  unit cell, 109, 851-852
  view, 3-5
  VLSI, 1, 411
Layout view, 3-5
LDD, see lightly-doped drain
L_{lfff}, see lateral diffusion
L_{drawn}, see drawn length
L_{elec}, see electrical length
L_{eff}, see effective channel length
Leakage current, 93, 181, 184, 200, 244, 249, 397-403, 546
Level shifting, 547-548, 692
  pumped output voltage driver, 546
Li, Harry, 1017
Lifetime of a minority carrier, τ_T, see diode, minority carrier lifetime
Lightly-doped drain (LDD), 90-91, 116, 151-152, 179, 193-196
Linear region, see Triode region
LOCOS, 181
Long-channel,
  analog models, 269-296
  digital models, 312-316
  MOSFETs, xxxi, 132-150
  tables, 292, 317, 320
Low impedance node, 673
Low power, 149, 299, 451, 476, 563, 635, 727, 729, 737
Low voltage, 476
  current mirror, 637-639
  differential amplifiers, 727, 729, 736-740
  reference, 760, 768-770

M

Manchester NRZ, 584
Masks, 21, 34-35, 53, 69, 84, 86, 163, 167-170, 170, 173, 179, 851
  aligning, 84, 86, 94, 170
  CMOS fabrication, 182-209
Matching,
  capacitor, 116
  common-centroid, 111-113, 618, 724, 851, 916, 980
  current mirror, 615-618, 636, 642
  diff-pair, 724-726
  layout, 617-620
  NMOS vs PMOS, 440-441
  resistors, 110-113
  threshold voltage, 616, 726, 856, 867
  transconductance, 616
Measurements, 246, 267
  LASI, 10
  noise, 213, 219-224
  probe, 326-327

Memory circuits, 433-482
array architectures, 434-447
buried capacitor cell, 446
chip organization, 447
decoders, 457-461
DRAM, 348, 438-446
EEPROM, 469
efficiency, 447
EPROM, 468
erased, 466, 471
F, feature size, 443-445
flash, 113, 463, 469-476, 481-482, 487-489
folded bit line, 441-446
floating gate, 466-474
global decoders, 458
local decoders, 458
memory cells, 463
NAND flash, 471
NSA, 435-439
open array, 436
organization, 457
peripheral circuits, 448
PE decoder, 461
PROM, 464-465
programmed, 466, 471
PSA, 440-441
RAM block diagram, 433, 457
refresh operation, 441
ROM, 464
row drivers, 461
sense-amp design, 448-456
sensing basics, 435-441
SRAM, 463-464
trench capacitor cell, 444-445
Metal layers, 59-74
bond pad, 4, 59
capacitive feedthrough, 66
cross-sectional view, 64
delays, 65
design rules, 69
electromigration, 59, 68, 70
inter layer dielectric (ILD), 69
intrinsic propagation delay, 65
parasitics, 61, 64-65
sizing, 68-74
Metallurgical grade silicon (MGS), 162
Metastability, 383-385, 396, 450, 452, 478, 918
Miller,
capacitance, 311-312, 661-662, 664, 674, 684
compensation, 774, 788
effect, 311, 687, 693
eliminating, 686
neglected zero, 662-665
nested, 820
theorem, 660-661, 676
Minority carrier lifetime, τ_T, see diode, minority carrier lifetime
Mixed signal, 706, 842, 957, 960
Mobility, 40, 106, 144, 153
Moore's law, xxxi
MOSFET, 131
accumulation, 132-134, 136, 159, 514
adjusting threshold voltage, 140
capacitances, 117,123, 135
capacitors, 544
C_{gs} calculation, 145
channel-length modulation, 144
depletion, 133
depletion device, 140
diode-connected, 272, 534, 542, 545, 579, 624, 627, 629, 639, 648, 673, 708, 719, 737, 819-820, 872
dummy poly strips, 618
effective width, 116
enhancement device, 140
excess gate voltage, see overdrive voltage
field device, 38, 181, 184-185
flatband voltage, 139
flicker noise, 302-303, 833
f_T, unity current gain frequency, 290-292, 297-302, 309, 316, 863, 903, 906
gain-f_T product (GFT), 300-302
gate tunnel current, 154, 242, 403, 474-475
$I_{D,sat}$, 144-145, 271
I_{off}, 150, 294, 398, 408
I_{on}, 150, 152-153, 157-158, 297, 314, 316
IV characteristics, 140, 157
lateral diffusion, 116-117, 124, 144, 315, 616, 620
layout and cross-sectional views, 86-87, 95-96, 118-122
linear region, see MOSFET, triode region
long channel, xxxi, 132-150, 269-296, 312-317, 320
NMOS layout, 95
ohmic region, see MOSFET, triode region
output resistance, 288
overdrive voltage, 271, 297-302, 616, 800, 802, 804, 863-864, 884, 892, 896, 903, 906
overlap capacitances, 132
oxide encroachment, 116-117, 132, 181-182
pad oxide, 182
parallel connection, 358
parasitic resistances, 118
pinch-off, 143
PMOS layout, 96
punch-through, 143
saturation region, 143-145, 271
scaling, 152
series connection, 359
short channel, xxxi, 151-158, 297-302, 314-317, 320
SPICE statement, 119-120
source/drain depletion capacitances, 117
substrate current-induce body effect, 154
subthreshold slope, 150

symbols, 96-97, 131
temperature effects, 293-295
threshold voltage, 135-140, 293
triode region, 141, 271, 278
$V_{DS,sat}$, 144-145, 271
MOSIS, 3, 6, 53-55
multiproject wafer, 3
web address, 3
Multipliers, 923-929
multiplying quad, 924
simulating, 926-928
squaring circuits, 928-929
Multivibrator circuits, 529-531
source coupled, 565-567
MUX/DEMUX, 378-381, 422

N

N-well, 31-58
cross-section, 32, 36, 39
design rules, 36, 54
patterning, 35
resistor, 32, 38, 94
n-well substrate diode, 39
N-select, 84-85, 179
NAND gate, 353-356
layout, 358, 414-415
switching characteristics, 358-363
switching point voltage, 354-355
Negative frequency, 218
n_i, see Intrinsic, carrier
NMOS, 95-97, see also MOSFET
NMOS inverter, 349
Noise, 213-268
1/f, see flicker
amplifier, 220-223
analysis, 235
avalanche, 252-254, 268
averaging thermal (white) noise, 240-241
averaging flicker noise, 246
burst (popcorn), 252-253
correlation, 256-258
differential amplifier, 726-727
diode, 242-244, 249, 252-254
equivalent bandwidth, 220-223
estimating RMS value from the time-domain amplitude, 249
excess, 253
feedback, 259-261
figure, 233-240
flicker, 213, 244-253, 261-262, 267, 302-303, 833
gate tunnel current, 242
Gaussian probability distribution function, 248-249
generator, 253-254
imaging chip, 245-246
input-referred, 220-224, 226-240, 247, 249, 251, 256-261
integrator, 245, 267
kT/C, 229, 240, 244, 250-251, 517, 833-834, 846, 856, 896
margins, 333
mean squared, 214-215, 262
MOSFET, 302
modeling, 219
noiseless resistor model in SPICE, 236-238
op-amp, 247-252, 259-261
phase noise, 592
photodiode, 249-250
pink, 244
power spectral density, 215-218
red, 245
resistors, 225-229
RMS, 215, 217, 219-224, 248-249
root mean squared, see RMS
shot, 242-244, 268
signal-to-noise ratio, 230-234, 246, 266-267, 945-946, 956, 964
spectrum analyzer, 216-218
standard deviation, 248-249
substrate, 851
temperature, 239-240
thermal, 225
transimpedance amplifier, 249-251
variance, see Noise, mean squared
Zener, 253-254
Nonoverlapping clocks, 401-402, 843, 850, 853-856, 874, 896
NOR gate, 353-357
layout, 358, 414-415
switching characteristics, 358-363
switching point voltage, 356-357
NRD and NRS, 118-120, 813
Nyquist criterion, 934

O

Off currrent (I_{off}), 150, 294, 398, 408
Offset binary, 965
Offset voltage,
ADC, 953, 1000, 1005-1007, 1014
adding to a sense amp, 498
charge injection, 831, 836, 841
CMRR, 723, 789-790
common-mode voltage, 792
constant-g_m, 740, 792
DAC, 945-946
delta-sigma ADC, 1014
diff-amp, 725-726, 867-868, 875
flash ADC, 988-990
gain error, 936
integrator, 846
modeling in an op-amp, 776, 827, 867-888
modeling with SPICE, 654, 867-868, 870-871

multiplier, 923
op-amp, 775-776, 804, 813, 818, 825, 853-857, 863, 867-868, 870-871, 892-895, 902-903
pipeline, 996-998
random, 775
removal, 518-519, 853-857
sensing, 497-500, 502
systematic, 775, 818, 916, 930
Ohmic contact, 83
Ohmic region, see Triode region
On current (I_{on}), 150, 152-153, 157-158, 297, 314, 316
One shot, 529
Open circuit gain, 298, 300-302, 668, 673, 678, 876
Operational Amplifiers, 773-828, 863-908
biasing, 865-866, 876
CMFB, see common mode feedback
CMRR, see common mode rejection ratio
common-mode output voltage, 838
common mode feedback (CMFB), 836-837, 863, 869-875, 881-885, 888-895, 896-904, 906-908
common mode rejection ratio (CMRR), 721-724, 742, 773, 789-790, 823, 825, 833
folded-cascode, 641, 648-649, 714-715, 802-810, 827-828, 872
fully-differential, 836-842, 863-904
f_{un}, op-amp unity-gain frequency, 260-261, 680-685, 740, 779-781, 785-786, 792-793, 797-799, 808-812, 852-853, 993
gain-enhancement, 808-812, 889-890
gain margin (GM), 781
high-speed design, 783-785, 806-812, 863
input common-mode range (CMR), 476, 730, 732, 741, 743, 774, 787, 701-702, 802, 807, 870, 872, 889
layout of differential op-amps, 865
offset-voltage, 775, 853, 867, 893
open-loop response, 780, 781, 795, 799, 801,804, 805, 806, 811, 812, 822, 823
OTA with buffer, 800-808
output voltage swing, 775, 880, 884, 886, 891
phase margin (PM), 781
power amplifier, 807-808
power dissipation, 775
PSRR, 790-791, 838
offsets, 775, 892
RHP zero, 662, 782
settling time, 852, 895, see also Settling time
simulation results, 886, 902
single-ended to differential, 894
slew-rate limitations, 787, 800
spice model, 841
start-up problems, 887
switched capacitor CMFB, 874-875, 896-897
voltage regulator, 812-817
Operational transconductance amplifier (OTA), 796-808
Oscillator,
astable, 530-531
ring, 339-341, 547
Schmitt trigger, 527-529
Voltage controlled, 561-567
OTA, see operation transconductance amplifier
Output buffer, see Buffers
Output Offset Storage (OOS), 856-857
Overdrive voltage, 271, 297-302, 616, 800, 802, 804, 863-864, 884, 892, 896, 903, 906
Overglass layer, 62-63, 75-77, 80, 83
Overlap capacitance, 123-124, 132
Oxide,
breakdown, 154
capacitance, 114-115, 132
encroachment, 116-117, 132, 181-182
growth, 34, 163-164

P

Packaging, 3-5, 211
P-select, 84-85, 179
P-well, 31, 52-54, 179, 181, 184, 187-191, 469-474
Pad layer, 62
Parasitics, 1-2
Pass gate (PG), 321-326, 357
Pass transistor, see Pass gate (PG)
Passivation, 62, 178-179, 208-210
Path selector, 378
Patterning, 32-35
PCE, see power conversion efficiency
PE logic, 404-405
Peak detector, 539-540
PG, see Pass gate (PG)
Phase-locked loops, see Digital phase-locked loop (DPLL)
Phase margin, 781
Phase shift, 665
Photolithography, 161, 167-171, 173, 182-193, 196, 202-203, 206, 208-211
Physical vapor deposition (PVD), 175-176, 200, 204, 208
Pie, 5
Pinch-off, 143
Pipelining, 407
Pitch, 443
Plasma etching, 173
PLL, see Digital phase-locked loops (DPLLP
PM, see Operational amplifier, phase margin
PMOS, 96-97, see also MOSFET
PN junction physics, 39-43
PNP, lateral, 757
model, 758
Pole splitting, 674-679, 685, 780, 783, 814, 821
Poly (polysilicon), 83, 179

design rules, 99
poly2, 113-115
Polycide, 88
Power and ground, 72-74, 417
Power conversion efficiency (PCE), 699-702
Power delay product (PDP), 340
Power dissipation,
inverter, 339-340
Probe, 326-327
Process characteristic time constant, 316
Process flow, 89-91, 161
Process integration, 177-208
Proportional to absolute temperature (PTAT), 745, 762
PSRR, 790-791, 838
PTAT, see proportional to absolute temperature
Pulse density modulation, 1008
Pulse statement, 26
Pumps, 542-550
driver, 546
Punch-through, 143
Push-pull amplifier, 698-704

R

Rail, 698
Random offsets, 775
Ranking, 11
Rapid thermal processing (RTP), 164
RC circuit, 50
distributed, 49-52, 65, 89, 348
Reactive ion etching (RIE), 171
Recombination, 40, 252
References, 745-772
bandgap, 757-772
CTAT, 745, 761-762
diode based, 757-772
low voltage, 760, 770
PTAT, 745, 762-765
Registration errors, 170
Regularity, 412
Regulated drain CM, 645
Resistance, 37
calculation, 37
looking into the drain, 668
looking into the source, 669
resistivity, 37
sheet, 37
Resistor,
common-centroid layout, 111-113
delay through, 49-52
dummy elements, 113
guard rings, 110
interdigitated layout, 110
layout, 38, 56-57, 109-113
layout of corners, 38
n-well, 32, 38-39, 94
properties, 88
sheet resistance, 37
switched-capacitor, 843
temp co, 106, 116, 631,
unit elements, 109
Reticle, see mask
Reverse recovery time, 46-49
RHP zero, 662, 685
Ring oscillator, 339, 547
Ringing, 695
Rise time, 50, 317-320
Routing, 430
ROX (recessed oxide), see FOX (field oxide)
RSHUNT, 28

S

Saturation current, see Diode, saturation (scale) current, I_s
Salicide, 199-200
Saturation region, 143-145, 271
S/H, see Sample-and-hold
Sample-and-hold, 834-836, 838-842, 880-881, 901-904, 935-938
acquisition time, 936
aperture error, 937
aperture jitter, 937
aperture uncertainty, 937
droop, 937
gain error, 936
hold mode, 937
linearity, 936
offset, 936
overshoot, 936
sample mode, 936
sampling error, 937
settling time, 936
Scale current, see Diode, saturation (scale) current, I_s
Scaling, 152
Schmitt Trigger, 523-529
applications, 527-529
Schottky diode, 92, 760, 772
Scope probe, 326-327
Select, 84-85, 179
Self-aligned gate, 86-88
Self-biased diff-amp, 534-538, 914, 920
Self-biasing, 624-626, 750-770
start-up circuit, 625, 629, 752-754
Self-correcting phase detector, 588-590
Sense amplifier design, 448-456
clock feedthrough, 449
contention current, 450
creating an imbalance, 451-455
kickback noise, 449-451, 456
increasing input range, 454, 918-919
removing sense-amp memory, 451
reducing power, 453
simulation examples, 454

Sensing using delta-sigma modulation (DSM), 483-519
  dynamic range, 490
  examples, 484-486
  feedback signal, 492-496
  Flash memory, 487-496
  incomplete settling, 496
  imagers, 504-519
  low pass filter, 507
  mismatches, 517-519
  noise issues, 506-507, 517
  offset, 497, 517-519
  parasitic capacitance, 492
  pixel, 504
  precision of sense, 485
  programmed state, 487
  resistive memory, 497-504
  sampling, 505
  sensing circuit, 488, 499, 511
  sigma-delta modulation, 484
  voltage to current conversion, 508-510
Sensitivity analysis, 623
Series connection of MOSFETs, 322
  delay, 325-326
Settling time, 663, 782, 788-789, 800, 802, 804, 810, 819, 840, 852-853, 880, 892, 895, 901-904, 936, 945, 990, 993, 1000
  incomplete, 496, 502-503, 520
Setup time, 388-389
Shallow trench isolation (STI), 90-91, 177-178, 180-188
Sheet resistance, 37
Shielding, 961
Shift register, 402
Shotgun shell, 242
Short-channel,
  analog models, 297-302
  digital models, 314-316
  effects. 153-154
  MOSFETs, xxxi, 151-158
  tables, 300, 317, 320
Sigma-delta modulation, 1007-1015, see also Sensing using delta-sigma modulation
  first-order, 1010-1014
  higher order, 1014
  noise shaping, 10-141015
  resolution verses oversampling, 1015
Signal to noise ratio (SNR), 230-234, 246, 266-267, 945-946, 956, 964
Signal to noise and distortion ratio (SNDR), 956
Silicide, 83, 88-92, 94, 130, 199-201
Silicide block, 89
Silcion bandgap energy, 759
Single-ended to differential conversion, 894
Single-ended to differential converter,
Skew, 533
Slew rate, 727, 787, 800, 853, 918, 922,
Small-signal,
  AC analysis, 280
  models, 279-288
  signal parameters, 280, 287-289
SNDR, see signal to noise and distortion ratio
SNR, see signal to noise ratio
Source coupled multivibrators, 565
Source coupled VCO, 565
Source follower, 670, 690-698
  body-effect, 691
  input capacitance, 693
Spectrum Analyzer, 216
SPICE, 1, 23, 67, 78, 145
  ABSTOL, 27
  AC analysis, 26, 27, 285
  breakdown, 58
  common mistakes, 28
  convergence, 27-28, 401
  diode breakdown voltage, 58
  DC analysis, 27
  diode model, 47-49, 758
  distortion, 705-706
  generating netlist, 23
  GMIN, 28, 401
  initial conditions, see UIC
  jagged plots, 25
  LASICKT, 127
  level, 145
  measuring currents, 274-275
  mistakes, 28
  models, parameters for short channel CMOS process, 154, 158
  netlist, see cmosedu.com, 24-26
  noise analysis, 243
  noiseless resistor model, 236-238
  opening a netlist, 24
  operating point analysis, 282-283
  problems, 27-28
  pulse statement, 26
  RELTOL, 27
  RSHUNT, 28, 401
  scale factors, 129
  Schottky diode, 772
  simulating dynamic circuits, 401
  SIN source, 25, 282-287,
  .tran statement, 25, 286, 665-666, 705-706
  transient analysis, 24-28, 286
  UIC, 28
  VNTOL, 27
Squaring circuits, 928
Square-law equations, 142-144, 271-272
Stability, 776-777
Standard-cell, 97
Standard deviation, 248-249
Start-up circuit, 625, 629, 752-754
Static logic, 353
Stepper, 34, 167
STI, see shallow trench isolation
Stick diagram, 422

Storage node, 397
Storage capacitance, 45-46
Storage time, 46-49
Strong inversion, 133
Student projects, 329
Substrate, 31
Substrate pump, 547
Subthreshold,
  current, 149, 278
  current source/sink, 635
  g_m, 292-293
  output resistance, 278
  slope, 150
  transconductance, 292
Switch logic,
  CVSL, 369
  DVSL, 369
  examples, 370
Switched capacitor circuits, 492-494, 843-853
  capacitor layout, 851-852
  filter, 848-849
  integrator, 845-848
  resistor, 488-489, 843-848
  slew-rate requirements, 862
Synchronization, 383, 551, 591, 596
Systematic offsets, 775

T

Technology computer assisted design (TCAD), 179
Telescopic diff-amp, see Cascode diff-amp
Temperature coefficient,
  bandgap references, 761-770
  diode, 759
  diode self-biasing, 761-762
  MOSFET-only, 633-635
  MOSFET voltage divider, 749-750
  positive, 106-107
  resistors, 105-107
  resistor-MOSFET divider, 631-632, 746-749
  thermal voltage self-biasing, 762-765
  threshold voltage, 293-296, 750
  transconductance multiplier, 634-635, 754-755
Temp co, see Temperature coefficient
Temperature effects,
  bandgap of silicon, 293
  capacitor, 116
  KP, 295, 633
  MOSFETs, 293-296
  thermal voltage, 294
  threshold voltage, 293-296
TG (transmission gate), 324, 375
THD, see Total harmonic distortion
Thermal oxidation, 163
Thermal voltage, V_T, 39. 758
  change with temperature. 294
Thin film depostion. 173
Thin film removal, 170
Thermometer code, 965-966, 974-975, 985-986, 988, 992
Threshold voltage, 135-140, 276, 467
  adjusting, 140, 190-192
  determining, 293
  matching, 616
  NMOS, 292, 300
  PMOS, 292, 300
  temperature behavior. 293-296
Timing errors, 583
TLC files, 21
Total harmonic distortion, 704-706
t_{ox}, 86-87, 114, 292, 300
TPSC, see true-single phase clocked logic
Transconductance, 280, 297
  parameter, β, 142
  variation with frequency, 695
Transient analysis, 24-27
Transimpedance amplifier, 689
Transmission channel, 538-541
Transmission gate, 324, 375, 402
  applications, 378
  delay through, 376
  layout, 415
  series connection, 377
  static gates, 379
Tri-state outputs, 351, 370
  buffer, 372-373
  phase detector output, 559
Triode region, 141, 271, 278
  channel resistance, 278
t_{rr}, see diode, reverse recovery time
True-single phase clocked logic, 409-410
t_s, see Diode, storage time
τ_T, see Diode, minority carrier lifetime
Tub, 52
Tunneling gate current, 154, 468-476
Two's complement, 965-966
Twin tub, 31, 187

U

ULSI, 1

V

V_{CM}, see common-mode voltage
V_{DSsat}, 144-145, 271
Velocity saturation, 151
VLSI, 1, 411
Voltage coefficient, 107, 116
Voltage controlled oscillator (VCO), 552, 561, 595
  characteristics, 561
  current starved, 561
  for use with the PFD, 561
  logic based, 567
  source coupled, 565

use with XOR phase detector, 557
Voltage dividers, 746-756
Voltage generators (charge pumps), 542-548
example, 546-548
higher voltages (Dickson), 544-545
Voltage references, 745-772
Voltage regulator, 812-819
V_{ov}, see Overdrive voltage
V_T, see Thermal voltage
VTC, see Inverter, transfer characteristics
V_{THN}, see Threshold voltage, NMOS
V_{THP}, see Threshold voltage, PMOS

W

Wafer, 3, 32, 161, 180
manufacture, 161
probe, 208
production, 162
Waterfowl, 242
W_{drawn}, see drawn width
Weak inversion, 150
Wet etching, 171
Well, 31-58, 53
triple, 53, 187, 278
Wide swing diff-amp, 736
WinSpice, (see SPICE)

X

X_d, 136-137
X_{dl}, 144
XOR gate, 366, 380
XOR phase detector, 553
characteristics, 555
gain, 556

Y

Yield, 209, 412

Z

Zener diode, 253-254, 268
Zero-nulling resistor, 685
Zipper Logic, 407

About the Author

Russel Jacob (Jake) Baker was born in Ogden, Utah, on October 5, 1964. He received the B.S. and M.S. degrees in electrical engineering from the University of Nevada, Las Vegas, and the Ph.D. degree in electrical engineering from the University of Nevada, Reno.

From 1981 to 1987, he was in the United States Marine Corps Reserves. From 1985 to 1993, he worked for E. G. & G. Energy Measurements and the Lawrence Livermore National Laboratory designing nuclear diagnostic instrumentation for underground nuclear weapons tests at the Nevada test site. During this time, he designed over 30 electronic and electro-optic instruments including high-speed (750 Mb/s) fiber-optic receiver/transmitters, PLLs, frame- and bit-syncs, data converters, streak-camera sweep circuits, micro-channel plate gating circuits, and analog oscilloscope electronics. From 1993 to 2000, he was a faculty member in the department of electrical engineering at the University of Idaho. In 2000, he joined a new electrical and computer engineering program at the Boise State University where he is currently a professor and chair of the department. Also, since 1993, he has consulted for various companies and laboratories including Micron Technology, Amkor Wafer Fabrication Services, Tower Semiconductor, Rendition, Lawrence Berkeley Laboratory, and the Tower ASIC Design Center.

Professor Baker holds over 75 granted or pending patents in integrated circuit design. He is a member of the electrical engineering honor society Eta Kappa Nu and an author/coauthor of the books *CMOS: Circuit Design, Layout, and Simulation*, *DRAM Circuit Design: A Tutorial*, and *CMOS: Mixed-Signal Circuit Design*. His research interests are in the areas of CMOS mixed-signal integrated circuit design and the design of memory in new and emerging fabrication technologies. Professor Baker was a co-recipient of the 2000 Prize Paper Award of the IEEE Power Electronics Society.

This document was created with Win2PDF available at http://www.win2pdf.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.