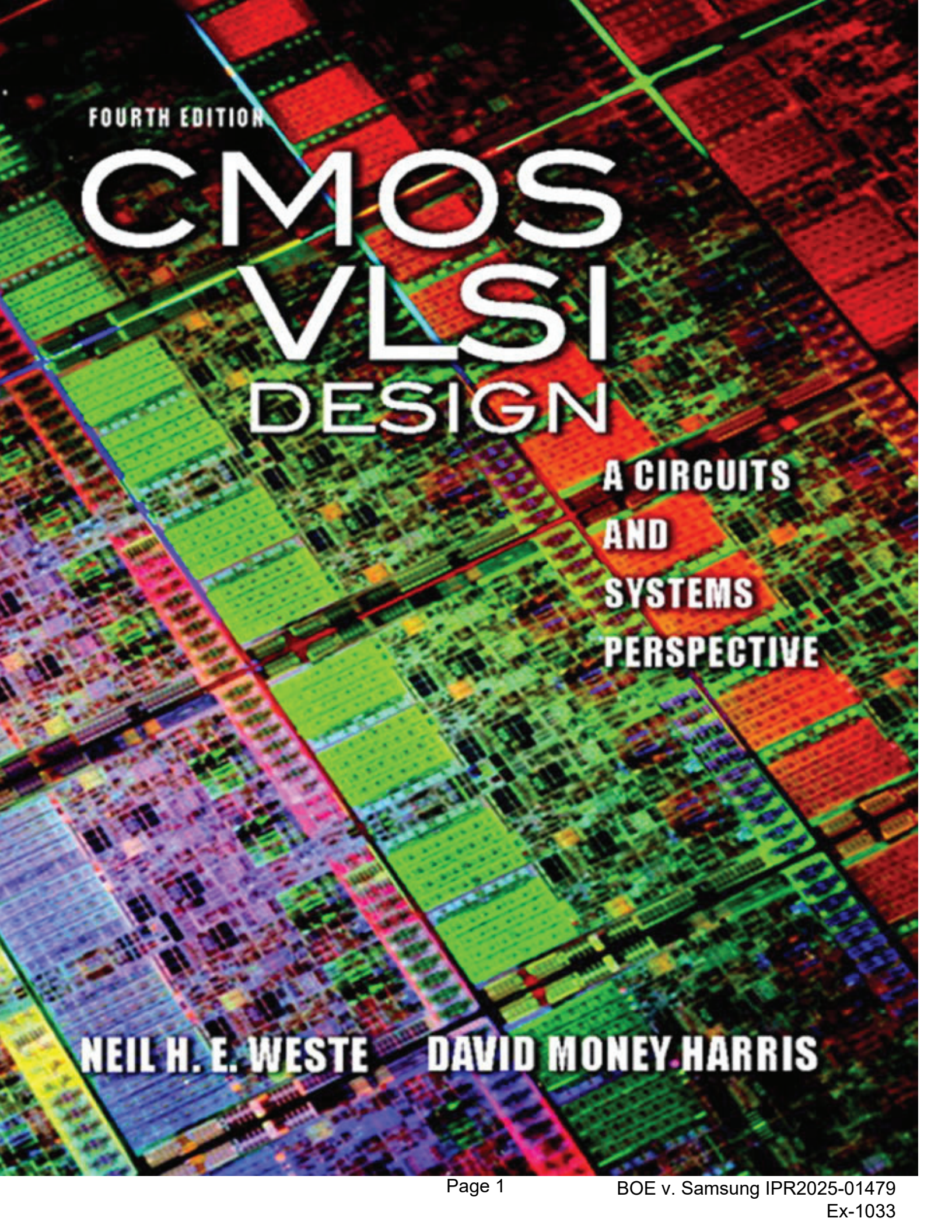




FOURTH EDITION

CMOS VLSI DESIGN

A CIRCUITS
AND
SYSTEMS
PERSPECTIVE

NEIL H. E. WESTE

DAVID MONEY HARRIS

CMOS VLSI Design

A Circuits and Systems Perspective

Fourth Edition

This page intentionally left blank

CMOS VLSI Design

A Circuits and Systems Perspective

Fourth Edition

Neil H. E. Weste

*Macquarie University and
The University of Adelaide*

David Money Harris

Harvey Mudd College

Addison-Wesley

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editor in Chief: Michael Hirsch
Acquisitions Editor: Matt Goldstein
Editorial Assistant: Chelsea Bell
Managing Editor: Jeffrey Holcomb
Senior Production Project Manager: Marilyn Lloyd
Media Producer: Katelyn Boller
Director of Marketing: Margaret Waples
Marketing Coordinator: Kathryn Ferranti
Senior Manufacturing Buyer: Carol Melville
Senior Media Buyer: Ginny Michaud
Text Designer: Susan Raymond
Art Director, Cover: Linda Knowles
Cover Designer: Joyce Cosentino Wells/J Wells Design
Cover Image: Cover photograph courtesy of Nick Knupffer—Intel Corporation.
Copyright © 2009 Intel Corporation. All rights reserved.
Full Service Vendor: Gillian Hall/The Aardvark Group Publishing Service
Copyeditor: Kathleen Cantwell, C4 Technologies
Proofreader: Holly McLean-Aldis
Indexer: Jack Lewis
Printer/Binder: Edwards Brothers
Cover Printer: Lehigh-Phoenix Color/Hagerstown

Credits and acknowledgments borrowed from other sources and reproduced with permission in this textbook appear on appropriate page within text or on page 838.

The interior of this book was set in Adobe Caslon and Trade Gothic.

Copyright © 2011, 2005, 1993, 1985 Pearson Education, Inc., publishing as Addison-Wesley. All rights reserved. Manufactured in the United States of America. This publication is protected by Copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, 501 Boylston Street, Suite 900, Boston, Massachusetts 02116.

Many of the designations by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Cataloging-in-Publication Data is
on file with the Library of Congress.

Addison-Wesley
is an imprint of



10 9 8 7 6 5 4 3 2 1—EB—14 13 12 11 10
ISBN 10: 0-321-54774-8
ISBN 13: 978-0-321-54774-3

*To Avril, Melissa, Tamara, Nicky, Jocelyn,
Makayla, Emily, Danika, Dan and Simon*
N. W.

To Jennifer, Samuel, and Abraham
D. M. H.

This page intentionally left blank

Contents

Preface xxv

Chapter 1 Introduction

1.1	A Brief History	1
1.2	Preview	6
1.3	MOS Transistors	6
1.4	CMOS Logic	9
1.4.1	The Inverter	9
1.4.2	The NAND Gate	9
1.4.3	CMOS Logic Gates	9
1.4.4	The NOR Gate	11
1.4.5	Compound Gates	11
1.4.6	Pass Transistors and Transmission Gates	12
1.4.7	Tristates	14
1.4.8	Multiplexers	15
1.4.9	Sequential Circuits	16
1.5	CMOS Fabrication and Layout	19
1.5.1	Inverter Cross-Section	19
1.5.2	Fabrication Process	20
1.5.3	Layout Design Rules	24
1.5.4	Gate Layouts	27
1.5.5	Stick Diagrams	28
1.6	Design Partitioning	29
1.6.1	Design Abstractions	30
1.6.2	Structured Design	31
1.6.3	Behavioral, Structural, and Physical Domains	31
1.7	Example: A Simple MIPS Microprocessor	33
1.7.1	MIPS Architecture	33
1.7.2	Multicycle MIPS Microarchitecture	34
1.8	Logic Design	38
1.8.1	Top-Level Interfaces	38
1.8.2	Block Diagrams	38
1.8.3	Hierarchy	40
1.8.4	Hardware Description Languages	40
1.9	Circuit Design	42

1.10 Physical Design	45
1.10.1 Floorplanning	45
1.10.2 Standard Cells	48
1.10.3 Pitch Matching	50
1.10.4 Slice Plans	50
1.10.5 Arrays	51
1.10.6 Area Estimation	51
1.11 Design Verification	53
1.12 Fabrication, Packaging, and Testing	54
Summary and a Look Ahead	55
Exercises	57

Chapter 2 MOS Transistor Theory

2.1 Introduction	61
2.2 Long-Channel I-V Characteristics	64
2.3 C-V Characteristics	68
2.3.1 Simple MOS Capacitance Models	68
2.3.2 Detailed MOS Gate Capacitance Model	70
2.3.3 Detailed MOS Diffusion Capacitance Model	72
2.4 Nonideal I-V Effects	74
2.4.1 Mobility Degradation and Velocity Saturation	75
2.4.2 Channel Length Modulation	78
2.4.3 Threshold Voltage Effects	79
2.4.4 Leakage	80
2.4.5 Temperature Dependence	85
2.4.6 Geometry Dependence	86
2.4.7 Summary	86
2.5 DC Transfer Characteristics	87
2.5.1 Static CMOS Inverter DC Characteristics	88
2.5.2 Beta Ratio Effects	90
2.5.3 Noise Margin	91
2.5.4 Pass Transistor DC Characteristics	92
2.6 Pitfalls and Fallacies	93
Summary	94
Exercises	95

Chapter 3 CMOS Processing Technology

3.1 Introduction	99
3.2 CMOS Technologies	100
3.2.1 Wafer Formation	100
3.2.2 Photolithography	101

3.2.3	Well and Channel Formation	103
3.2.4	Silicon Dioxide (SiO ₂)	105
3.2.5	Isolation	106
3.2.6	Gate Oxide	107
3.2.7	Gate and Source/Drain Formations	108
3.2.8	Contacts and Metallization	110
3.2.9	Passivation	112
3.2.10	Metrology	112
3.3	Layout Design Rules	113
3.3.1	Design Rule Background	113
3.3.2	Scribe Line and Other Structures	116
3.3.3	MOSIS Scalable CMOS Design Rules	117
3.3.4	Micron Design Rules	118
3.4	CMOS Process Enhancements	119
3.4.1	Transistors	119
3.4.2	Interconnect	122
3.4.3	Circuit Elements	124
3.4.4	Beyond Conventional CMOS	129
3.5	Technology-Related CAD Issues	130
3.5.1	Design Rule Checking (DRC)	131
3.5.2	Circuit Extraction	132
3.6	Manufacturing Issues	133
3.6.1	Antenna Rules	133
3.6.2	Layer Density Rules	134
3.6.3	Resolution Enhancement Rules	134
3.6.4	Metal Slotting Rules	135
3.6.5	Yield Enhancement Guidelines	135
3.7	Pitfalls and Fallacies	136
3.8	Historical Perspective	137
	Summary	139
	Exercises	139

Chapter 4 Delay

4.1	Introduction	141
4.1.1	Definitions	141
4.1.2	Timing Optimization	142
4.2	Transient Response	143
4.3	RC Delay Model	146
4.3.1	Effective Resistance	146
4.3.2	Gate and Diffusion Capacitance	147
4.3.3	Equivalent RC Circuits	147
4.3.4	Transient Response	148
4.3.5	Elmore Delay	150

4.3.6	Layout Dependence of Capacitance	153
4.3.7	Determining Effective Resistance	154
4.4	Linear Delay Model	155
4.4.1	Logical Effort	156
4.4.2	Parasitic Delay	156
4.4.3	Delay in a Logic Gate	158
4.4.4	Drive	159
4.4.5	Extracting Logical Effort from Datasheets	159
4.4.6	Limitations to the Linear Delay Model	160
4.5	Logical Effort of Paths	163
4.5.1	Delay in Multistage Logic Networks	163
4.5.2	Choosing the Best Number of Stages	166
4.5.3	Example	168
4.5.4	Summary and Observations	169
4.5.5	Limitations of Logical Effort	171
4.5.6	Iterative Solutions for Sizing	171
4.6	Timing Analysis Delay Models	173
4.6.1	Slope-Based Linear Model	173
4.6.2	Nonlinear Delay Model	174
4.6.3	Current Source Model	174
4.7	Pitfalls and Fallacies	174
4.8	Historical Perspective	175
	Summary	176
	Exercises	176

Chapter 5 Power

5.1	Introduction	181
5.1.1	Definitions	182
5.1.2	Examples	182
5.1.3	Sources of Power Dissipation	184
5.2	Dynamic Power	185
5.2.1	Activity Factor	186
5.2.2	Capacitance	188
5.2.3	Voltage	190
5.2.4	Frequency	192
5.2.5	Short-Circuit Current	193
5.2.6	Resonant Circuits	193
5.3	Static Power	194
5.3.1	Static Power Sources	194
5.3.2	Power Gating	197
5.3.3	Multiple Threshold Voltages and Oxide Thicknesses	199

5.3.4	Variable Threshold Voltages	199
5.3.5	Input Vector Control	200
5.4	Energy-Delay Optimization	200
5.4.1	Minimum Energy	200
5.4.2	Minimum Energy-Delay Product	203
5.4.3	Minimum Energy Under a Delay Constraint	203
5.5	Low Power Architectures	204
5.5.1	Microarchitecture	204
5.5.2	Parallelism and Pipelining	204
5.5.3	Power Management Modes	205
5.6	Pitfalls and Fallacies	206
5.7	Historical Perspective	207
	Summary	209
	Exercises	209

Chapter 6 Interconnect

6.1	Introduction	211
6.1.1	Wire Geometry	211
6.1.2	Example: Intel Metal Stacks	212
6.2	Interconnect Modeling	213
6.2.1	Resistance	214
6.2.2	Capacitance	215
6.2.3	Inductance	218
6.2.4	Skin Effect	219
6.2.5	Temperature Dependence	220
6.3	Interconnect Impact	220
6.3.1	Delay	220
6.3.2	Energy	222
6.3.3	Crosstalk	222
6.3.4	Inductive Effects	224
6.3.5	An Aside on Effective Resistance and Elmore Delay	227
6.4	Interconnect Engineering	229
6.4.1	Width, Spacing, and Layer	229
6.4.2	Repeaters	230
6.4.3	Crosstalk Control	232
6.4.4	Low-Swing Signaling	234
6.4.5	Regenerators	236
6.5	Logical Effort with Wires	236
6.6	Pitfalls and Fallacies	237
	Summary	238
	Exercises	238

Chapter 7 Robustness

7.1	Introduction	241
7.2	Variability	241
7.2.1	Supply Voltage	242
7.2.2	Temperature	242
7.2.3	Process Variation	243
7.2.4	Design Corners	244
7.3	Reliability	246
7.3.1	Reliability Terminology	246
7.3.2	Oxide Wearout	247
7.3.3	Interconnect Wearout	249
7.3.4	Soft Errors	251
7.3.5	Overvoltage Failure	252
7.3.6	Latchup	253
7.4	Scaling	254
7.4.1	Transistor Scaling	255
7.4.2	Interconnect Scaling	257
7.4.3	International Technology Roadmap for Semiconductors	258
7.4.4	Impacts on Design	259
7.5	Statistical Analysis of Variability	263
7.5.1	Properties of Random Variables	263
7.5.2	Variation Sources	266
7.5.3	Variation Impacts	269
7.6	Variation-Tolerant Design	274
7.6.1	Adaptive Control	275
7.6.2	Fault Tolerance	275
7.7	Pitfalls and Fallacies	277
7.8	Historical Perspective	278
	Summary	284
	Exercises	284

Chapter 8 Circuit Simulation

8.1	Introduction	287
8.2	A SPICE Tutorial	288
8.2.1	Sources and Passive Components	288
8.2.2	Transistor DC Analysis	292
8.2.3	Inverter Transient Analysis	292
8.2.4	Subcircuits and Measurement	294
8.2.5	Optimization	296
8.2.6	Other HSPICE Commands	298

8.3	Device Models	298
8.3.1	Level 1 Models	299
8.3.2	Level 2 and 3 Models	300
8.3.3	BSIM Models	300
8.3.4	Diffusion Capacitance Models	300
8.3.5	Design Corners	302
8.4	Device Characterization	303
8.4.1	I-V Characteristics	303
8.4.2	Threshold Voltage	306
8.4.3	Gate Capacitance	308
8.4.4	Parasitic Capacitance	308
8.4.5	Effective Resistance	310
8.4.6	Comparison of Processes	311
8.4.7	Process and Environmental Sensitivity	313
8.5	Circuit Characterization	313
8.5.1	Path Simulations	313
8.5.2	DC Transfer Characteristics	315
8.5.3	Logical Effort	315
8.5.4	Power and Energy	318
8.5.5	Simulating Mismatches	319
8.5.6	Monte Carlo Simulation	319
8.6	Interconnect Simulation	319
8.7	Pitfalls and Fallacies	322
	Summary	324
	Exercises	324

Chapter 9 Combinational Circuit Design

9.1	Introduction	327
9.2	Circuit Families	328
9.2.1	Static CMOS	329
9.2.2	Ratioed Circuits	334
9.2.3	Cascode Voltage Switch Logic	339
9.2.4	Dynamic Circuits	339
9.2.5	Pass-Transistor Circuits	349
9.3	Circuit Pitfalls	354
9.3.1	Threshold Drops	355
9.3.2	Ratio Failures	355
9.3.3	Leakage	356
9.3.4	Charge Sharing	356
9.3.5	Power Supply Noise	356
9.3.6	Hot Spots	357

9.3.7	Minority Carrier Injection	357
9.3.8	Back-Gate Coupling	358
9.3.9	Diffusion Input Noise Sensitivity	358
9.3.10	Process Sensitivity	358
9.3.11	Example: Domino Noise Budgets	359
 9.4	More Circuit Families	360
9.5	Silicon-On-Insulator Circuit Design	360
9.5.1	Floating Body Voltage	361
9.5.2	SOI Advantages	362
9.5.3	SOI Disadvantages	362
9.5.4	Implications for Circuit Styles	363
9.5.5	Summary	364
9.6	Subthreshold Circuit Design	364
9.6.1	Sizing	365
9.6.2	Gate Selection	365
9.7	Pitfalls and Fallacies	366
9.8	Historical Perspective	367
	Summary	369
	Exercises	370

Chapter 10 Sequential Circuit Design

10.1	Introduction	375
10.2	Sequencing Static Circuits	376
10.2.1	Sequencing Methods	376
10.2.2	Max-Delay Constraints	379
10.2.3	Min-Delay Constraints	383
10.2.4	Time Borrowing	386
10.2.5	Clock Skew	389
10.3	Circuit Design of Latches and Flip-Flops	391
10.3.1	Conventional CMOS Latches	392
10.3.2	Conventional CMOS Flip-Flops	393
10.3.3	Pulsed Latches	395
10.3.4	Resettable Latches and Flip-Flops	396
10.3.5	Enabled Latches and Flip-Flops	397
10.3.6	Incorporating Logic into Latches	398
10.3.7	Klass Semidynamic Flip-Flop (SDFF)	399
10.3.8	Differential Flip-Flops	399
10.3.9	Dual Edge-Triggered Flip-Flops	400
10.3.10	Radiation-Hardened Flip-Flops	401
 10.3.11	True Single-Phase-Clock (TSPC) Latches and Flip-Flops	402
10.4	Static Sequencing Element Methodology	402
10.4.1	Choice of Elements	403
10.4.2	Characterizing Sequencing Element Delays	405

- 10.4.3 State Retention Registers 408
- 10.4.4 Level-Converter Flip-Flops 408
- 10.4.5 Design Margin and Adaptive Sequential Elements 409
- 10.4.6 Two-Phase Timing Types 411

	10.5 Sequencing Dynamic Circuits	411
	10.6 Synchronizers	411
	10.6.1 Metastability 412	
	10.6.2 A Simple Synchronizer 415	
	10.6.3 Communicating Between Asynchronous Clock Domains 416	
	10.6.4 Common Synchronizer Mistakes 417	
	10.6.5 Arbiters 419	
	10.6.6 Degrees of Synchrony 419	
	10.7 Wave Pipelining	420
	10.8 Pitfalls and Fallacies	422
	10.9 Case Study: Pentium 4 and Itanium 2 Sequencing Methodologies	423
	Summary 423	
	Exercises 425	

Chapter 11 Datapath Subsystems

11.1 Introduction	429
11.2 Addition/Subtraction	429
11.2.1 Single-Bit Addition 430	
11.2.2 Carry-Propagate Addition 434	
11.2.3 Subtraction 458	
11.2.4 Multiple-Input Addition 458	
11.2.5 Flagged Prefix Adders 459	
11.3 One/Zero Detectors	461
11.4 Comparators	462
11.4.1 Magnitude Comparator 462	
11.4.2 Equality Comparator 462	
11.4.3 $K = A + B$ Comparator 463	
11.5 Counters	463
11.5.1 Binary Counters 464	
11.5.2 Fast Binary Counters 465	
11.5.3 Ring and Johnson Counters 466	
11.5.4 Linear-Feedback Shift Registers 466	
11.6 Boolean Logical Operations	468
11.7 Coding	468
11.7.1 Parity 468	
11.7.2 Error-Correcting Codes 468	
11.7.3 Gray Codes 470	
11.7.4 XOR/XNOR Circuit Forms 471	

11.8 Shifters	472
11.8.1 Funnel Shifter	473
11.8.2 Barrel Shifter	475
11.8.3 Alternative Shift Functions	476
11.9 Multiplication	476
11.9.1 Unsigned Array Multiplication	478
11.9.2 Two's Complement Array Multiplication	479
11.9.3 Booth Encoding	480
11.9.4 Column Addition	485
11.9.5 Final Addition	489
11.9.6 Fused Multiply-Add	490
11.9.7 Serial Multiplication	490
11.9.8 Summary	490
11.10 Parallel-Prefix Computations	491
11.11 Pitfalls and Fallacies	493
Summary	494
Exercises	494

Chapter 12 Array Subsystems

12.1 Introduction	497
12.2 SRAM	498
12.2.1 SRAM Cells	499
12.2.2 Row Circuitry	506
12.2.3 Column Circuitry	510
12.2.4 Multi-Ported SRAM and Register Files	514
12.2.5 Large SRAMs	515
12.2.6 Low-Power SRAMs	517
12.2.7 Area, Delay, and Power of RAMs and Register Files	520
12.3 DRAM	522
12.3.1 Subarray Architectures	523
12.3.2 Column Circuitry	525
12.3.3 Embedded DRAM	526
12.4 Read-Only Memory	527
12.4.1 Programmable ROMs	529
12.4.2 NAND ROMs	530
12.4.3 Flash	531
12.5 Serial Access Memories	533
12.5.1 Shift Registers	533
12.5.2 Queues (FIFO, LIFO)	533
12.6 Content-Addressable Memory	535
12.7 Programmable Logic Arrays	537

12.8 Robust Memory Design	541
12.8.1 Redundancy	541
12.8.2 Error Correcting Codes (ECC)	543
12.8.3 Radiation Hardening	543
12.9 Historical Perspective	544
Summary	545
Exercises	546

Chapter 13 Special-Purpose Subsystems

13.1 Introduction	549
13.2 Packaging and Cooling	549
13.2.1 Package Options	549
13.2.2 Chip-to-Package Connections	551
13.2.3 Package Parasitics	552
13.2.4 Heat Dissipation	552
13.2.5 Temperature Sensors	553
13.3 Power Distribution	555
13.3.1 On-Chip Power Distribution Network	556
13.3.2 IR Drops	557
13.3.3 $L di/dt$ Noise	558
13.3.4 On-Chip Bypass Capacitance	559
13.3.5 Power Network Modeling	560
13.3.6 Power Supply Filtering	564
13.3.7 Charge Pumps	564
13.3.8 Substrate Noise	565
13.3.9 Energy Scavenging	565
13.4 Clocks	566
13.4.1 Definitions	566
13.4.2 Clock System Architecture	568
13.4.3 Global Clock Generation	569
13.4.4 Global Clock Distribution	571
13.4.5 Local Clock Gaters	575
13.4.6 Clock Skew Budgets	577
13.4.7 Adaptive Deskewing	579
13.5 PLLs and DLLs	580
13.5.1 PLLs	580
13.5.2 DLLs	587
13.5.3 Pitfalls	589
13.6 I/O	590
13.6.1 Basic I/O Pad Circuits	591
13.6.2 Electrostatic Discharge Protection	593
13.6.3 Example: MOSIS I/O Pads	594
13.6.4 Mixed-Voltage I/O	596

13.7 High-Speed Links	597
13.7.1 High-Speed I/O Channels	597
13.7.2 Channel Noise and Interference	600
13.7.3 High-Speed Transmitters and Receivers	601
13.7.4 Synchronous Data Transmission	606
13.7.5 Clock Recovery in Source-Synchronous Systems	606
13.7.6 Clock Recovery in Mesochronous Systems	608
13.7.7 Clock Recovery in Pleisochronous Systems	610
13.8 Random Circuits	610
13.8.1 True Random Number Generators	610
13.8.2 Chip Identification	611
13.9 Pitfalls and Fallacies	612
Summary	613
Exercises	614

Chapter 14 Design Methodology and Tools

14.1 Introduction	615
14.2 Structured Design Strategies	617
14.2.1 A Software Radio—A System Example	618
14.2.2 Hierarchy	620
14.2.3 Regularity	623
14.2.4 Modularity	625
14.2.5 Locality	626
14.2.6 Summary	627
14.3 Design Methods	627
14.3.1 Microprocessor/DSP	627
14.3.2 Programmable Logic	628
14.3.3 Gate Array and Sea of Gates Design	631
14.3.4 Cell-Based Design	632
14.3.5 Full Custom Design	634
14.3.6 Platform-Based Design—System on a Chip	635
14.3.7 Summary	636
14.4 Design Flows	636
14.4.1 Behavioral Synthesis Design Flow (ASIC Design Flow)	637
14.4.2 Automated Layout Generation	641
14.4.3 Mixed-Signal or Custom-Design Flow	645
14.5 Design Economics	646
14.5.1 Non-Recurring Engineering Costs (NREs)	647
14.5.2 Recurring Costs	649
14.5.3 Fixed Costs	650
14.5.4 Schedule	651
14.5.5 Personpower	653
14.5.6 Project Management	653
14.5.7 Design Reuse	654

14.6	Data Sheets and Documentation	655
14.6.1	The Summary	655
14.6.2	Pinout	655
14.6.3	Description of Operation	655
14.6.4	DC Specifications	655
14.6.5	AC Specifications	656
14.6.6	Package Diagram	656
14.6.7	Principles of Operation Manual	656
14.6.8	User Manual	656
14.7	CMOS Physical Design Styles	656
14.8	Pitfalls and Fallacies	657
	Exercises	657

Chapter 15 Testing, Debugging, and Verification

15.1	Introduction	659
15.1.1	Logic Verification	660
15.1.2	Debugging	662
15.1.3	Manufacturing Tests	664
15.2	Testers, Test Fixtures, and Test Programs	666
15.2.1	Testers and Test Fixtures	666
15.2.2	Test Programs	668
15.2.3	Handlers	669
15.3	Logic Verification Principles	670
15.3.1	Test Vectors	670
15.3.2	Testbenches and Harnesses	671
15.3.3	Regression Testing	671
15.3.4	Version Control	672
15.3.5	Bug Tracking	673
15.4	Silicon Debug Principles	673
15.5	Manufacturing Test Principles	676
15.5.1	Fault Models	677
15.5.2	Observability	679
15.5.3	Controllability	679
15.5.4	Repeatability	679
15.5.5	Survivability	679
15.5.6	Fault Coverage	680
15.5.7	Automatic Test Pattern Generation (ATPG)	680
15.5.8	Delay Fault Testing	680
15.6	Design for Testability	681
15.6.1	<i>Ad Hoc</i> Testing	681
15.6.2	Scan Design	682
15.6.3	Built-In Self-Test (BIST)	684
15.6.4	IDDQ Testing	687
15.6.5	Design for Manufacturability	687

	15.7 Boundary Scan	688
	15.8 Testing in a University Environment	689
	15.9 Pitfalls and Fallacies	690
	Summary	697
	Exercises	697

Appendix A Hardware Description Languages

A.1 Introduction	699
A.1.1 Modules	700
A.1.2 Simulation and Synthesis	701
A.2 Combinational Logic	702
A.2.1 Bitwise Operators	702
A.2.2 Comments and White Space	703
A.2.3 Reduction Operators	703
A.2.4 Conditional Assignment	704
A.2.5 Internal Variables	706
A.2.6 Precedence and Other Operators	708
A.2.7 Numbers	708
A.2.8 Zs and Xs	709
A.2.9 Bit Swizzling	711
A.2.10 Delays	712
A.3 Structural Modeling	713
A.4 Sequential Logic	717
A.4.1 Registers	717
A.4.2 Resettable Registers	718
A.4.3 Enabled Registers	719
A.4.4 Multiple Registers	720
A.4.5 Latches	721
A.4.6 Counters	722
A.4.7 Shift Registers	724
A.5 Combinational Logic with Always / Process Statements	724
A.5.1 Case Statements	726
A.5.2 If Statements	729
A.5.3 SystemVerilog Casez	731
A.5.4 Blocking and Nonblocking Assignments	731
A.6 Finite State Machines	735
A.6.1 FSM Example	735
A.6.2 State Enumeration	736
A.6.3 FSM with Inputs	738
A.7 Type Idiosyncracies	740

A.8	Parameterized Modules	742
A.9	Memory	745
A.9.1	RAM	745
A.9.2	Multiported Register Files	747
A.9.3	ROM	748
A.10	Testbenches	749
A.11	SystemVerilog Netlists	754
A.12	Example: MIPS Processor	755
A.12.1	Testbench	756
A.12.2	SystemVerilog	757
A.12.3	VHDL	766
	Exercises	776

References 785

Index 817

Credits 838

This page intentionally left blank

Preface

In the two-and-a-half decades since the first edition of this book was published, CMOS technology has claimed the preeminent position in modern electrical system design. It has enabled the widespread use of wireless communication, the Internet, and personal computers. No other human invention has seen such rapid growth for such a sustained period. The transistor counts and clock frequencies of state-of-the-art chips have grown by orders of magnitude.

	1st Edition	2nd Edition	3rd Edition	4th Edition
Year	1985	1993	2004	2010
Transistor Counts	10^5 – 10^6	10^6 – 10^7	10^8 – 10^9	10^9 – 10^{10}
Clock Frequencies	10^7	10^8	10^9	10^9
Worldwide Market	\$25B	\$60B	\$170B	\$250B

This edition has been heavily revised to reflect the rapid changes in integrated circuit design over the past six years. While the basic principles are largely the same, power consumption and variability have become primary factors for chip design. The book has been reorganized to emphasize the key factors: delay, power, interconnect, and robustness. Other chapters have been reordered to reflect the order in which we teach the material.

How to Use This Book

This book intentionally covers more breadth and depth than any course would cover in a semester. It is accessible for a first undergraduate course in VLSI, yet detailed enough for advanced graduate courses and is useful as a reference to the practicing engineer. You are encouraged to pick and choose topics according to your interest. Chapter 1 previews the entire field, while subsequent chapters elaborate on specific topics. Sections are marked with the “Optional” icon (shown here in the margin) if they are not needed to understand subsequent sections. You may skip them on a first reading and return when they are relevant to you.

We have endeavored to include figures whenever possible (“a picture is worth a thousand words”) to trigger your thinking. As you encounter examples throughout the text, we urge you to think about them before reading the solutions. We have also provided extensive references for those who need to delve deeper into topics introduced in this text. We



have emphasized the best practices that are used in industry and warned of pitfalls and fallacies. Our judgments about the merits of circuits may become incorrect as technology and applications change, but we believe it is the responsibility of a writer to attempt to call out the most relevant information.

Supplements

Numerous supplements are available on the Companion Web site for the book, www.cmosvlsi.com. Supplements to help students with the course include:

- A lab manual with laboratory exercises involving the design of an 8-bit microprocessor covered in Chapter 1.
- A collection of links to VLSI resources including open-source CAD tools and process parameters.
- A student solutions manual that includes answers to odd-numbered problems.
- Certain sections of the book moved online to shorten the page count. These sections are indicated by the “Web Enhanced” icon (shown here in the margin).



Supplements to help instructors with the course include:

- A sample syllabus.
- Lecture slides for an introductory VLSI course.
- An instructor’s manual with solutions.

These materials have been prepared exclusively for professors using the book in a course. Please send email to computing@aw.com for information on how to access them.

Acknowledgments

We are indebted to many people for their reviews, suggestions, and technical discussions. These people include: Bharadwaj “Birdy” Amrutur, Mark Anders, Adnan Aziz, Jacob Baker, Kaustav Banerjee, Steve Bibyk, David Blaauw, Erik Brunvand, Neil Burgess, Wayne Burleson, Robert Drost, Jo Ebergen, Sarah Harris, Jacob Herbold, Ron Ho, David Hopkins, Mark Horowitz, Steven Hsu, Tanay Karnik, Omid Kaveh, Matthew Keeter, Ben Keller, Ali Keshavarzi, Bruce Khailany, Jaeha Kim, Volkan Kursun, Simon Knowles, Ram Krishnamurthy, Austin Lee, Ana Sonia Leon, Shih-Lien Lu, Sanu Mathew, Aleksandar Milenkovic, Sam Naffziger, Braden Phillips, Stefan Rusu, Justin Schauer, James Stine, Jason Stinson, Aaron Stratton, Ivan Sutherland, Jim Tschanz, Alice Wang, Gu-Yeon Wei, and Peiyi Zhao. We apologize in advance to anyone we overlooked.

MOSIS and IBM kindly provided permission to use nanometer SPICE models for many examples. Nathaniel Pinckney spent a summer revising the laboratory exercises and updating simulations. Jaeha Kim contributed new sections on phase-locked loops and high-speed I/O for Chapter 13. David would like to thank Bharadwaj Amrutur of the Indian Institute of Science and Braden Phillips of the University of Adelaide for hosting him during two productive summers of writing.

Addison-Wesley has done an admirable job with the grueling editorial and production process. We would particularly like to thank our editor, Matt Goldstein, and our compositor, Gillian Hall.

Sally Harris has been editing family books since David was an infant on her lap. She read the page proofs with amazing attention to detail and unearthed hundreds of errors.

This book would not have existed without the support of our families. David would particularly like to thank his wife Jennifer and sons Abraham and Samuel for enduring two summers of absence while writing, and to our extended family for their tremendous assistance.

We have become painfully aware of the ease with which mistakes creep into a book. Scores of 3rd edition readers have reported bugs that are now corrected. Despite our best efforts at validation, we are confident that we have introduced a similar number of new errors. Please check the errata sheet at www_cmosvlsi.com/errata.pdf to see if the bug has already been reported. Send your reports to bugs@cmosvlsi.com.

N. W.
D. M. H.
January 2010

This page intentionally left blank

Introduction

1

1.1 A Brief History

In 1958, Jack Kilby built the first integrated circuit flip-flop with two transistors at Texas Instruments. In 2008, Intel's Itanium microprocessor contained more than 2 billion transistors and a 16 Gb Flash memory contained more than 4 billion transistors. This corresponds to a compound annual growth rate of 53% over 50 years. No other technology in history has sustained such a high growth rate lasting for so long.

This incredible growth has come from steady miniaturization of transistors and improvements in manufacturing processes. Most other fields of engineering involve trade-offs between performance, power, and price. However, as transistors become smaller, they also become faster, dissipate less power, and are cheaper to manufacture. This synergy has not only revolutionized electronics, but also society at large.

The processing performance once dedicated to secret government supercomputers is now available in disposable cellular telephones. The memory once needed for an entire company's accounting system is now carried by a teenager in her iPod. Improvements in integrated circuits have enabled space exploration, made automobiles safer and more fuel-efficient, revolutionized the nature of warfare, brought much of mankind's knowledge to our Web browsers, and made the world a flatter place.

Figure 1.1 shows annual sales in the worldwide semiconductor market. Integrated circuits became a \$100 billion/year business in 1994. In 2007, the industry manufactured approximately 6 quintillion (6×10^{18}) transistors, or nearly a billion for every human being on the planet. Thousands of engineers have made their fortunes in the field. New fortunes lie ahead for those with innovative ideas and the talent to bring those ideas to reality.

During the first half of the twentieth century, electronic circuits used large, expensive, power-hungry, and unreliable vacuum tubes. In 1947, John Bardeen and Walter Brattain built the first functioning point contact transistor at Bell Laboratories, shown in Figure 1.2(a) [Riordan97]. It was nearly classified as a military secret, but Bell Labs publicly introduced the device the following year.

We have called it the Transistor, T-R-A-N-S-I-S-T-O-R, because it is a resistor or semiconductor device which can amplify electrical signals as they are transferred through it from input to output terminals. It is, if you will, the electrical equivalent of a vacuum tube amplifier. But there the similarity ceases. It has no vacuum, no filament, no glass tube. It is composed entirely of cold, solid substances.

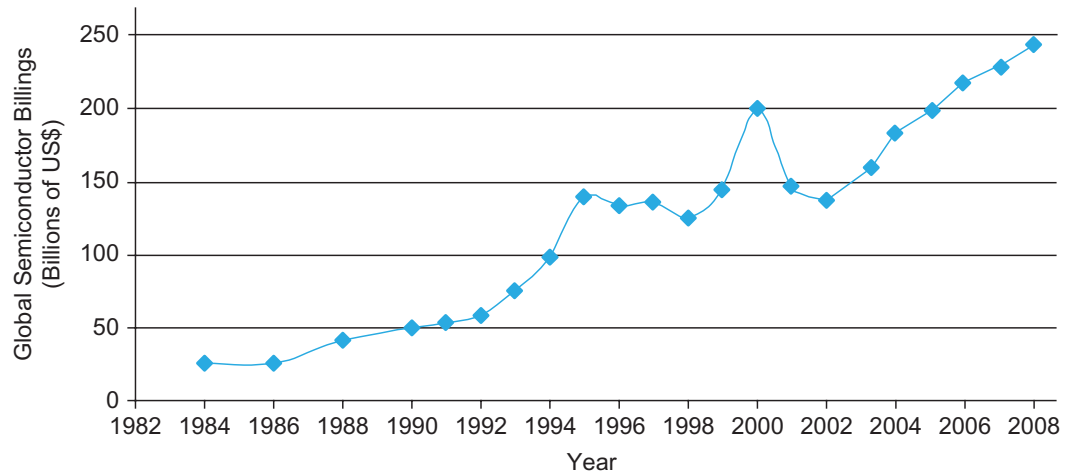
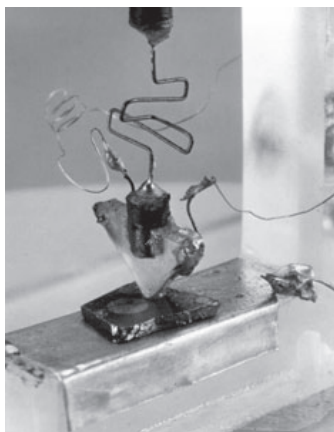


FIGURE 1.1 Size of worldwide semiconductor market (Courtesy of Semiconductor Industry Association.)

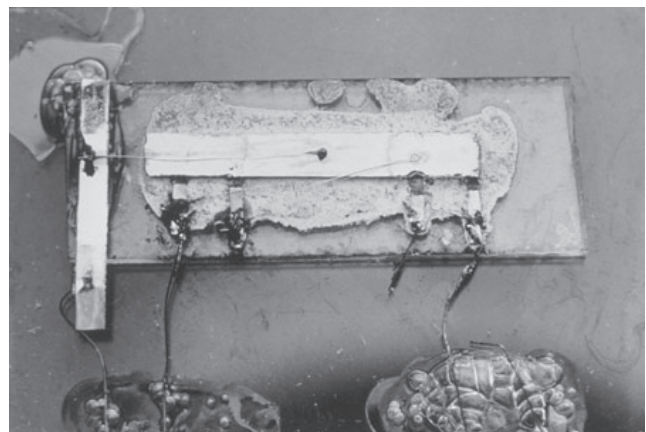
Ten years later, Jack Kilby at Texas Instruments realized the potential for miniaturization if multiple transistors could be built on one piece of silicon. Figure 1.2(b) shows his first prototype of an integrated circuit, constructed from a germanium slice and gold wires.

The invention of the transistor earned the Nobel Prize in Physics in 1956 for Bardeen, Brattain, and their supervisor William Shockley. Kilby received the Nobel Prize in Physics in 2000 for the invention of the integrated circuit.

Transistors can be viewed as electrically controlled switches with a control terminal and two other terminals that are connected or disconnected depending on the voltage or current applied to the control. Soon after inventing the point contact transistor, Bell Labs developed the bipolar junction transistor. Bipolar transistors were more reliable, less noisy, and more power-efficient. Early integrated circuits primarily used bipolar transistors. Bipolar transistors require a small current into the control (base) terminal to switch much larger currents between the other two (emitter and collector) terminals. The quiescent power dissipated by these base currents, drawn even when the circuit is not switching,



(a)



(b)

FIGURE 1.2 (a) First transistor (Property of AT&T Archives. Reprinted with permission of AT&T.) and (b) first integrated circuit (Courtesy of Texas Instruments.)

limits the maximum number of transistors that can be integrated onto a single die. By the 1960s, Metal Oxide Semiconductor Field Effect Transistors (MOSFETs) began to enter production. MOSFETs offer the compelling advantage that they draw almost zero control current while idle. They come in two flavors: nMOS and pMOS, using n-type and p-type silicon, respectively. The original idea of field effect transistors dated back to the German scientist Julius Lilienfeld in 1925 [US patent 1,745,175] and a structure closely resembling the MOSFET was proposed in 1935 by Oskar Heil [British patent 439,457], but materials problems foiled early attempts to make functioning devices.

In 1963, Frank Wanlass at Fairchild described the first logic gates using MOSFETs [Wanlass63]. Fairchild's gates used both nMOS and pMOS transistors, earning the name Complementary Metal Oxide Semiconductor, or CMOS. The circuits used discrete transistors but consumed only nanowatts of power, six orders of magnitude less than their bipolar counterparts. With the development of the silicon planar process, MOS integrated circuits became attractive for their low cost because each transistor occupied less area and the fabrication process was simpler [Vadasz69]. Early commercial processes used only pMOS transistors and suffered from poor performance, yield, and reliability. Processes using nMOS transistors became common in the 1970s [Mead80]. Intel pioneered nMOS technology with its 1101 256-bit static random access memory and 4004 4-bit microprocessor, as shown in Figure 1.3. While the nMOS process was less expensive than CMOS, nMOS logic gates still consumed power while idle. Power consumption became a major issue in the 1980s as hundreds of thousands of transistors were integrated onto a single die. CMOS processes were widely adopted and have essentially replaced nMOS and bipolar processes for nearly all digital logic applications.

In 1965, Gordon Moore observed that plotting the number of transistors that can be most economically manufactured on a chip gives a straight line on a semilogarithmic scale [Moore65]. At the time, he found transistor count doubling every 18 months. This observation has been called *Moore's Law* and has become a self-fulfilling prophecy. Figure 1.4 shows that the number of transistors in Intel microprocessors has doubled every 26 months since the invention of the 4004. Moore's Law is driven primarily by *scaling* down the size of transistors and, to a minor extent, by building larger chips. The level of integration of chips has been classified as small-scale, medium-scale, large-scale, and very large-scale. *Small-scale integration* (SSI) circuits, such as the 7404 inverter, have fewer than 10

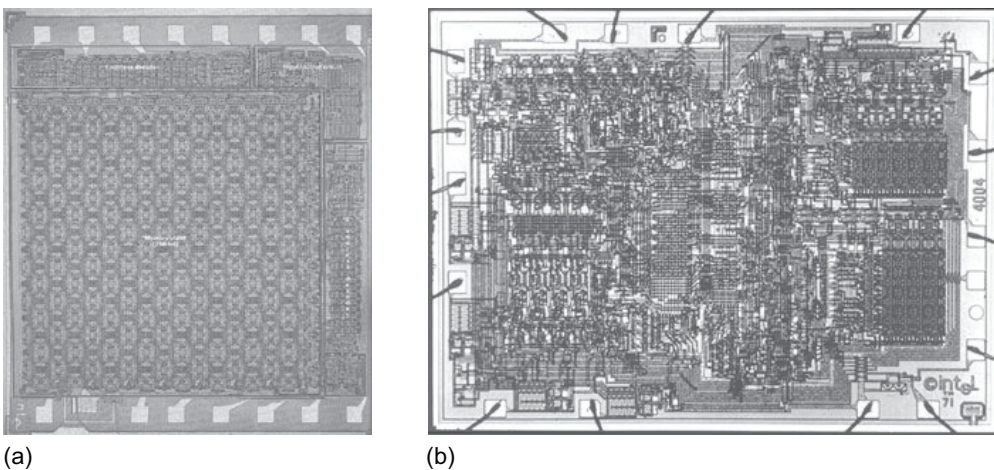


FIGURE 1.3 (a) Intel 1101 SRAM (© IEEE 1969 [Vadasz69]) and (b) 4004 microprocessor (Reprinted with permission of Intel Corporation.)

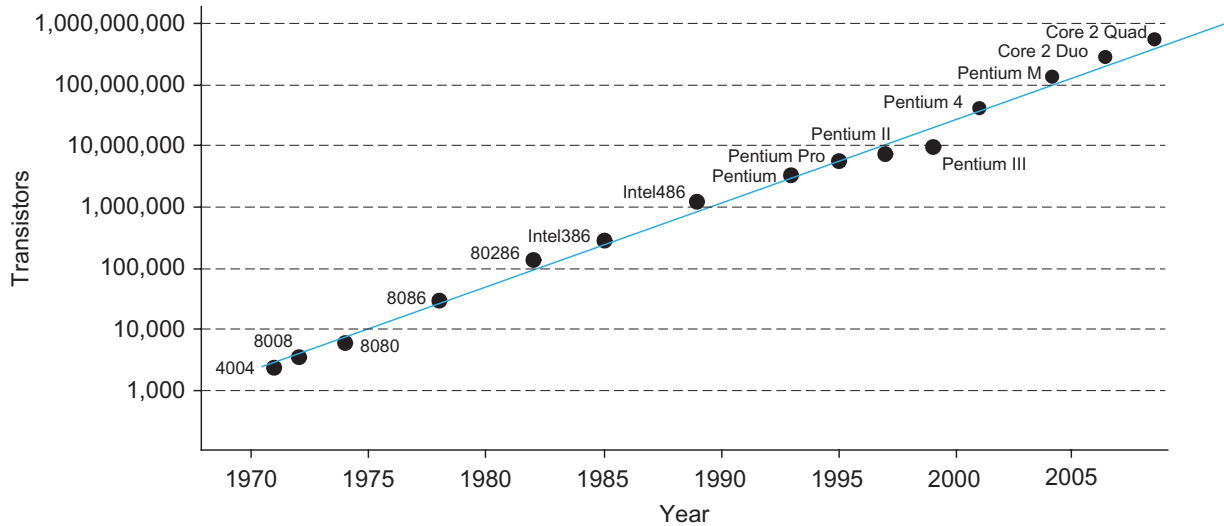


FIGURE 1.4 Transistors in Intel microprocessors [Intel10]

gates, with roughly half a dozen transistors per gate. *Medium-scale integration* (MSI) circuits, such as the 74161 counter, have up to 1000 gates. *Large-scale integration* (LSI) circuits, such as simple 8-bit microprocessors, have up to 10,000 gates. It soon became apparent that new names would have to be created every five years if this naming trend continued and thus the term *very large-scale integration* (VLSI) is used to describe most integrated circuits from the 1980s onward. A corollary of Moore's law is *Dennard's Scaling Law* [Dennard74]: as transistors shrink, they become faster, consume less power, and are cheaper to manufacture. Figure 1.5 shows that Intel microprocessor clock frequencies have doubled roughly every 34 months. This frequency scaling hit the power wall around 2004, and clock frequencies have leveled off around 3 GHz. Computer performance, measured in time to run an application, has advanced even more than raw clock speed. Presently, the performance is driven by the number of cores on a chip rather than by the clock. Even though an individual CMOS transistor uses very little energy each time it switches, the enormous number of transistors switching at very high rates of speed have made power consumption a major design consideration again. Moreover, as transistors have become so small, they cease to turn completely OFF. Small amounts of current leaking through each transistor now lead to significant power consumption when multiplied by millions or billions of transistors on a chip.

The feature size of a CMOS manufacturing process refers to the minimum dimension of a transistor that can be reliably built. The 4004 had a feature size of 10 μm in 1971. The Core 2 Duo had a feature size of 45 nm in 2008. Manufacturers introduce a new process generation (also called a technology node) every 2–3 years with a 30% smaller feature size to pack twice as many transistors in the same area. Figure 1.6 shows the progression of process generations. Feature sizes down to 0.25 μm are generally specified in microns (10^{-6} m), while smaller feature sizes are expressed in nanometers (10^{-9} m). Effects that were relatively minor in micron processes, such as transistor leakage, variations in characteristics of adjacent transistors, and wire resistance, are of great significance in nanometer processes.

Moore's Law has become a self-fulfilling prophecy because each company must keep up with its competitors. Obviously, this scaling cannot go on forever because transistors cannot be smaller than atoms. Dennard scaling has already begun to slow. By the 45 nm

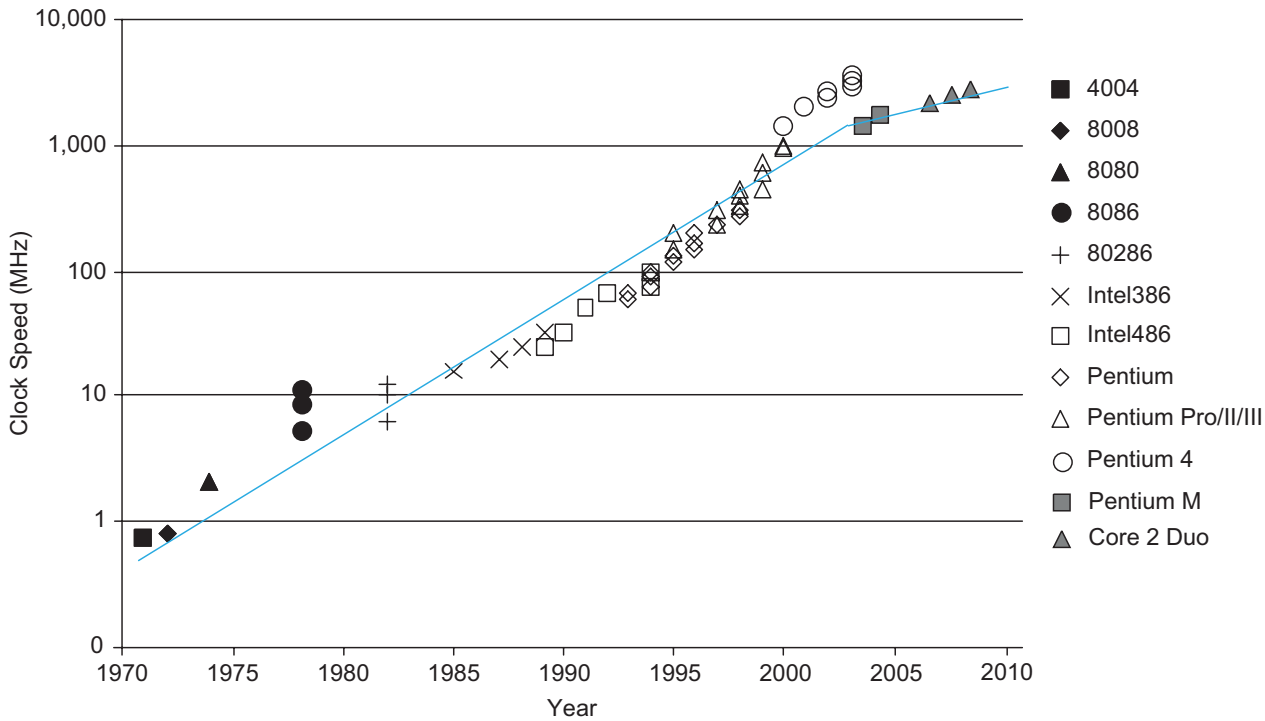


FIGURE 1.5 Clock frequencies of Intel microprocessors

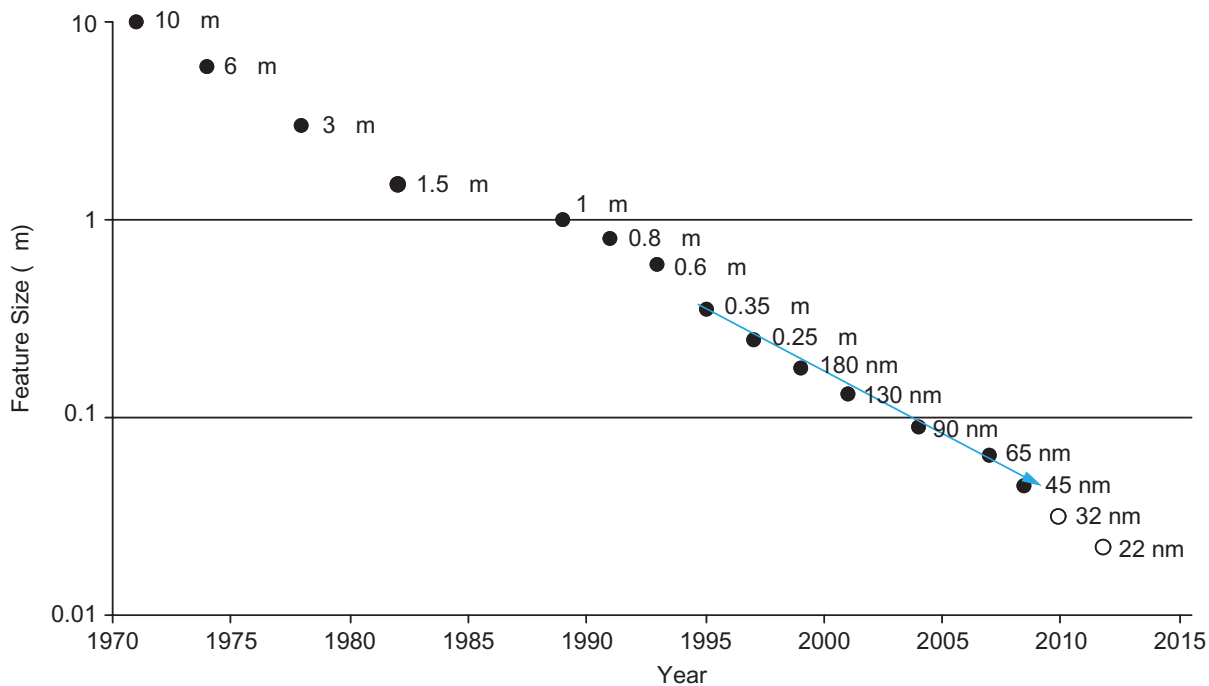


FIGURE 1.6 Process generations. Future predictions from [SIA2007].

generation, designers are having to make trade-offs between improving power and improving delay. Although the cost of printing each transistor goes down, the one-time design costs are increasing exponentially, relegating state-of-the-art processes to chips that will sell in huge quantities or that have cutting-edge performance requirements. However, many predictions of fundamental limits to scaling have already proven wrong. Creative engineers and material scientists have billions of dollars to gain by getting ahead of their competitors. In the early 1990s, experts agreed that scaling would continue for at least a decade but that beyond that point the future was murky. In 2009, we still believe that Moore's Law will continue for at least another decade. The future is yours to invent.

1.2 Preview

As the number of transistors on a chip has grown exponentially, designers have come to rely on increasing levels of automation to seek corresponding productivity gains. Many designers spend much of their effort specifying functions with hardware description languages and seldom look at actual transistors. Nevertheless, chip design is not software engineering. Addressing the harder problems requires a fundamental understanding of circuit and physical design. Therefore, this book focuses on building an understanding of integrated circuits from the bottom up.

In this chapter, we will take a simplified view of CMOS transistors as switches. With this model we will develop CMOS logic gates and latches. CMOS transistors are mass-produced on silicon wafers using lithographic steps much like a printing press process. We will explore how to lay out transistors by specifying rectangles indicating where dopants should be diffused, polysilicon should be grown, metal wires should be deposited, and contacts should be etched to connect all the layers. By the middle of this chapter, you will understand all the principles required to design and lay out your own simple CMOS chip. The chapter concludes with an extended example demonstrating the design of a simple 8-bit MIPS microprocessor chip. The processor raises many of the design issues that will be developed in more depth throughout the book. The best way to learn VLSI design is by doing it. A set of laboratory exercises are available at www.cmosvlsi.com to guide you through the design of your own microprocessor chip.

1.3 MOS Transistors

Silicon (Si), a semiconductor, forms the basic starting material for most integrated circuits [Tsividis99]. Pure silicon consists of a three-dimensional *lattice* of atoms. Silicon is a Group IV element, so it forms covalent bonds with four adjacent atoms, as shown in Figure 1.7(a). The lattice is shown in the plane for ease of drawing, but it actually forms a cubic crystal. As all of its valence electrons are involved in chemical bonds, pure silicon is a poor conductor. The conductivity can be raised by introducing small amounts of impurities, called *dopants*, into the silicon lattice. A dopant from Group V of the periodic table, such as arsenic, has five valence electrons. It replaces a silicon atom in the lattice and still bonds to four neighbors, so the fifth valence electron is loosely bound to the arsenic atom, as shown in Figure 1.7(b). Thermal vibration of the lattice at room temperature is enough to set the electron free to move, leaving a positively charged As^+ ion and a free electron. The free electron can carry current so the conductivity is higher. We call this an *n*-type

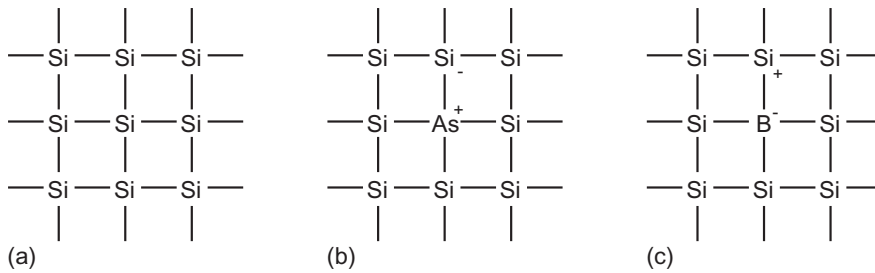


FIGURE 1.7 Silicon lattice and dopant atoms

semiconductor because the free carriers are negatively charged electrons. Similarly, a Group III dopant, such as boron, has three valence electrons, as shown in Figure 1.7(c). The dopant atom can borrow an electron from a neighboring silicon atom, which in turn becomes short by one electron. That atom in turn can borrow an electron, and so forth, so the missing electron, or *hole*, can propagate about the lattice. The hole acts as a positive carrier so we call this a *p*-type semiconductor.

A junction between *p*-type and *n*-type silicon is called a *diode*, as shown in Figure 1.8. When the voltage on the *p*-type semiconductor, called the *anode*, is raised above the *n*-type *cathode*, the diode is *forward biased* and current flows. When the anode voltage is less than or equal to the cathode voltage, the diode is *reverse biased* and very little current flows.

A Metal-Oxide-Semiconductor (*MOS*) structure is created by superimposing several layers of conducting and insulating materials to form a sandwich-like structure. These structures are manufactured using a series of chemical processing steps involving oxidation of the silicon, selective introduction of dopants, and deposition and etching of metal wires and contacts. Transistors are built on nearly flawless single crystals of silicon, which are available as thin flat circular wafers of 15–30 cm in diameter. CMOS technology provides two types of transistors (also called *devices*): an *n*-type transistor (*nMOS*) and a *p*-type transistor (*pMOS*). Transistor operation is controlled by electric fields so the devices are also called Metal Oxide Semiconductor Field Effect Transistors (*MOSFETs*) or simply *FETs*. Cross-sections and symbols of these transistors are shown in Figure 1.9. The *n*+ and *p*+ regions indicate heavily doped *n*- or *p*-type silicon.

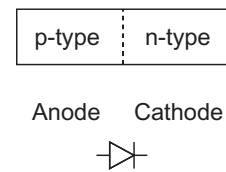


FIGURE 1.8
p-n junction diode
structure and symbol

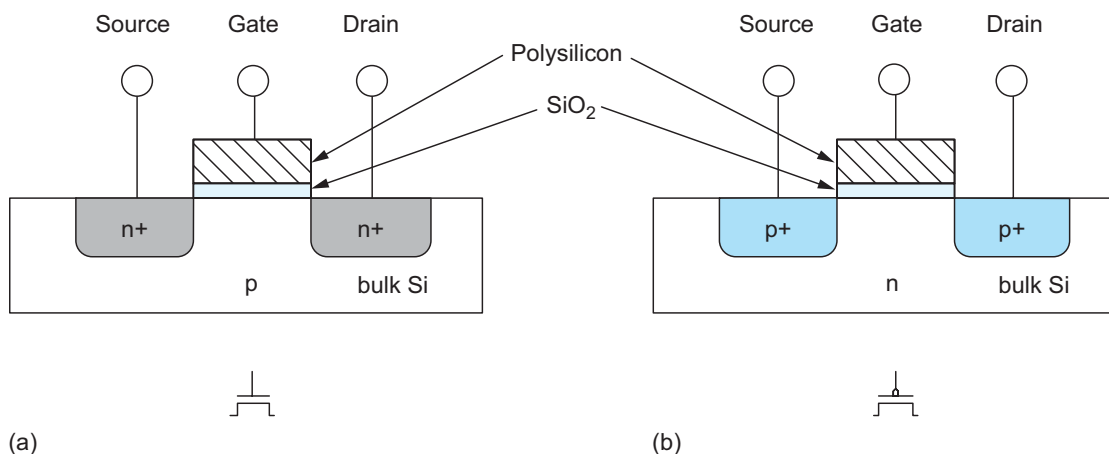


FIGURE 1.9 nMOS transistor (a) and pMOS transistor (b)

Each transistor consists of a stack of the conducting *gate*, an insulating layer of silicon dioxide (SiO_2 , better known as glass), and the silicon wafer, also called the *substrate*, *body*, or *bulk*. Gates of early transistors were built from metal, so the stack was called metal-oxide-semiconductor, or MOS. Since the 1970s, the gate has been formed from polycrystalline silicon (*polysilicon*), but the name stuck. (Interestingly, metal gates reemerged in 2007 to solve materials problems in advanced manufacturing processes.) An nMOS transistor is built with a p-type body and has regions of n-type semiconductor adjacent to the gate called the *source* and *drain*. They are physically equivalent and for now we will regard them as interchangeable. The body is typically grounded. A pMOS transistor is just the opposite, consisting of p-type source and drain regions with an n-type body. In a CMOS technology with both flavors of transistors, the substrate is either n-type or p-type. The other flavor of transistor must be built in a special *well* in which dopant atoms have been added to form the body of the opposite type.

The gate is a control input: It affects the flow of electrical current between the source and drain. Consider an nMOS transistor. The body is generally grounded so the p–n junctions of the source and drain to body are reverse-biased. If the gate is also grounded, no current flows through the reverse-biased junctions. Hence, we say the transistor is OFF. If the gate voltage is raised, it creates an electric field that starts to attract free electrons to the underside of the Si– SiO_2 interface. If the voltage is raised enough, the electrons outnumber the holes and a thin region under the gate called the *channel* is inverted to act as an n-type semiconductor. Hence, a conducting path of electron carriers is formed from source to drain and current can flow. We say the transistor is ON.

For a pMOS transistor, the situation is again reversed. The body is held at a positive voltage. When the gate is also at a positive voltage, the source and drain junctions are reverse-biased and no current flows, so the transistor is OFF. When the gate voltage is lowered, positive charges are attracted to the underside of the Si– SiO_2 interface. A sufficiently low gate voltage inverts the channel and a conducting path of positive carriers is formed from source to drain, so the transistor is ON. Notice that the symbol for the pMOS transistor has a bubble on the gate, indicating that the transistor behavior is the opposite of the nMOS.

The positive voltage is usually called V_{DD} or POWER and represents a logic 1 value in digital circuits. In popular logic families of the 1970s and 1980s, V_{DD} was set to 5 volts. Smaller, more recent transistors are unable to withstand such high voltages and have used supplies of 3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V, 1.0 V, and so forth. The low voltage is called GROUND (GND) or V_{SS} and represents a logic 0. It is normally 0 volts.

In summary, the gate of an MOS transistor controls the flow of current between the source and drain. Simplifying this to the extreme allows the MOS transistors to be viewed as

simple ON/OFF switches. When the gate of an nMOS transistor is 1, the transistor is ON and there is a conducting path from source to drain. When the gate is low, the nMOS transistor is OFF and almost zero current flows from source to drain. A pMOS transistor is just the opposite, being ON when the gate is low and OFF when the gate is high. This switch model is illustrated in Figure 1.10, where g , s , and d indicate gate, source, and drain. This model will be our most common one when discussing circuit behavior.

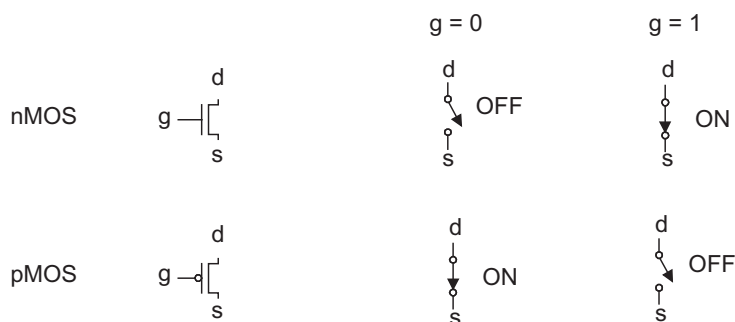


FIGURE 1.10 Transistor symbols and switch-level models

1.4 CMOS Logic

1.4.1 The Inverter

Figure 1.11 shows the schematic and symbol for a CMOS inverter or NOT gate using one nMOS transistor and one pMOS transistor. The bar at the top indicates V_{DD} and the triangle at the bottom indicates GND. When the input A is 0, the nMOS transistor is OFF and the pMOS transistor is ON. Thus, the output Y is pulled up to 1 because it is connected to V_{DD} but not to GND. Conversely, when A is 1, the nMOS is ON, the pMOS is OFF, and Y is pulled down to '0.' This is summarized in Table 1.1.

TABLE 1.1 Inverter truth table

A	Y
0	1
1	0

1.4.2 The NAND Gate

Figure 1.12(a) shows a 2-input CMOS NAND gate. It consists of two series nMOS transistors between Y and GND and two parallel pMOS transistors between Y and V_{DD} . If either input A or B is 0, at least one of the nMOS transistors will be OFF, breaking the path from Y to GND. But at least one of the pMOS transistors will be ON, creating a path from Y to V_{DD} . Hence, the output Y will be 1. If both inputs are 1, both of the nMOS transistors will be ON and both of the pMOS transistors will be OFF. Hence, the output will be 0. The truth table is given in Table 1.2 and the symbol is shown in Figure 1.12(b). Note that by DeMorgan's Law, the inversion bubble may be placed on either side of the gate. In the figures in this book, two lines intersecting at a T-junction are connected. Two lines crossing are connected if and only if a dot is shown.

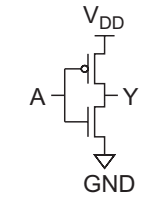
TABLE 1.2 NAND gate truth table

A	B	Pull-Down Network	Pull-Up Network	Y
0	0	OFF	ON	1
0	1	OFF	ON	1
1	0	OFF	ON	1
1	1	ON	OFF	0

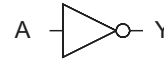
k -input NAND gates are constructed using k series nMOS transistors and k parallel pMOS transistors. For example, a 3-input NAND gate is shown in Figure 1.13. When any of the inputs are 0, the output is pulled high through the parallel pMOS transistors. When all of the inputs are 1, the output is pulled low through the series nMOS transistors.

1.4.3 CMOS Logic Gates

The inverter and NAND gates are examples of *static CMOS logic gates*, also called *complementary CMOS gates*. In general, a static CMOS gate has an nMOS *pull-down network* to connect the output to 0 (GND) and pMOS *pull-up network* to connect the output to 1 (V_{DD}), as shown in Figure 1.14. The networks are arranged such that one is ON and the other OFF for any input pattern.



(a)



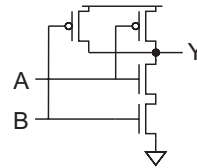
(b)

FIGURE 1.11

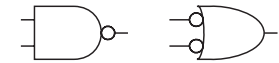
Inverter schematic

(a) and symbol

(b) $Y = \bar{A}$



(a)



(b)

FIGURE 1.12 2-input NAND

gate schematic (a) and symbol

(b) $Y = \overline{A \cdot B}$

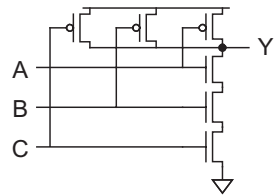


FIGURE 1.13 3-input NAND

gate schematic $Y = \overline{A \cdot B \cdot C}$

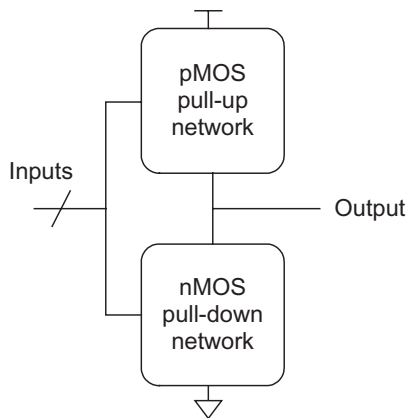


FIGURE 1.14 General logic gate using pull-up and pull-down networks

The pull-up and pull-down networks in the inverter each consist of a single transistor. The NAND gate uses a series pull-down network and a parallel pull-up network. More elaborate networks are used for more complex gates. Two or more transistors in series are ON only if all of the series transistors are ON. Two or more transistors in parallel are ON if any of the parallel transistors are ON. This is illustrated in Figure 1.15 for nMOS and pMOS transistor pairs. By using combinations of these constructions, CMOS combinational gates can be constructed. Although such static CMOS gates are most widely used, Chapter 9 explores alternate ways of building gates with transistors.

In general, when we join a pull-up network to a pull-down network to form a logic gate as shown in Figure 1.14, they both will attempt to exert a logic level at the output. The possible levels at the output are shown in Table 1.3. From this table it can be seen that the output of a CMOS logic gate can be in four states. The 1 and 0 levels have been encountered with the inverter and NAND gates, where either the pull-up or pull-down is OFF and the other structure is ON. When both pull-up and pull-down are OFF, the *high-impedance* or *floating Z* output state results. This is of importance in multiplexers, memory elements, and tristate bus drivers. The *crowbarred* (or *contention*) X level exists when both pull-up and pull-down are simultaneously turned ON. Contention between the two networks results in an indeterminate output level and dissipates static power. It is usually an unwanted condition.

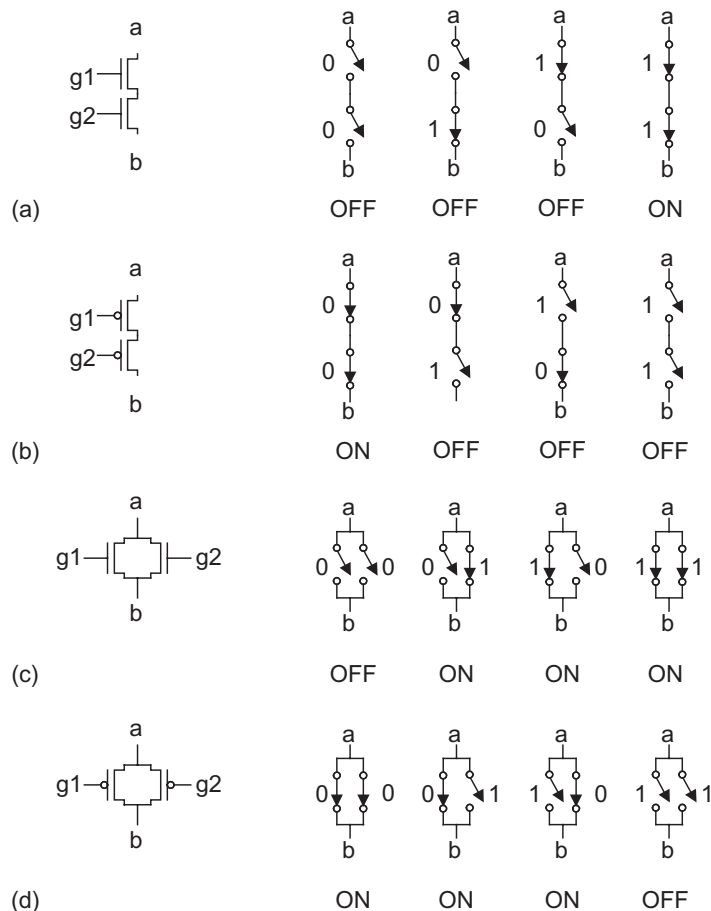


FIGURE 1.15 Connection and behavior of series and parallel transistors

TABLE 1.3 Output states of CMOS logic gates

	pull-up OFF	pull-up ON
pull-down OFF	Z	1
pull-down ON	0	crowbarred (X)

1.4.4 The NOR Gate

A 2-input NOR gate is shown in Figure 1.16. The nMOS transistors are in parallel to pull the output low when either input is high. The pMOS transistors are in series to pull the output high when both inputs are low, as indicated in Table 1.4. The output is never crowbarred or left floating.

TABLE 1.4 NOR gate truth table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

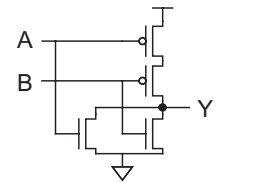
Example 1.1

Sketch a 3-input CMOS NOR gate.

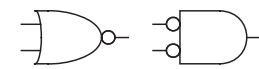
SOLUTION: Figure 1.17 shows such a gate. If any input is high, the output is pulled low through the parallel nMOS transistors. If all inputs are low, the output is pulled high through the series pMOS transistors.

1.4.5 Compound Gates

A *compound gate* performing a more complex logic function in a single stage of logic is formed by using a combination of series and parallel switch structures. For example, the derivation of the circuit for the function $Y = (A \cdot B) + (C \cdot D)$ is shown in Figure 1.18. This function is sometimes called AND-OR-INVERT-22, or AOI22 because it performs the NOR of a pair of 2-input ANDs. For the nMOS pull-down network, take the uninverted expression $((A \cdot B) + (C \cdot D))$ indicating when the output should be pulled to '0.' The AND expressions $(A \cdot B)$ and $(C \cdot D)$ may be implemented by series connections of switches, as shown in Figure 1.18(a). Now ORing the result requires the parallel connection of these two structures, which is shown in Figure 1.18(b). For the pMOS pull-up network, we must compute the complementary expression using switches that turn on with inverted polarity. By DeMorgan's Law, this is equivalent to interchanging AND and OR operations. Hence, transistors that appear in series in the pull-down network must appear in parallel in the pull-up network. Transistors that appear in parallel in the pull-down network must appear in series in the pull-up network. This principle is called *conduction complements* and has already been used in the design of the NAND and NOR gates. In the pull-up network, the parallel combination of A and B is placed in series with the parallel combination of C and D . This progression is evident in Figure 1.18(c) and Figure 1.18(d). Putting the networks together yields the full schematic (Figure 1.18(e)). The symbol is shown in Figure 1.18(f).



(a)



(b)

FIGURE 1.16 2-input NOR gate schematic (a) and symbol (b) $Y = \overline{A + B}$

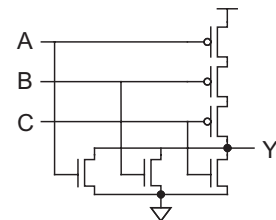


FIGURE 1.17 3-input NOR gate schematic $Y = \overline{A + B + C}$

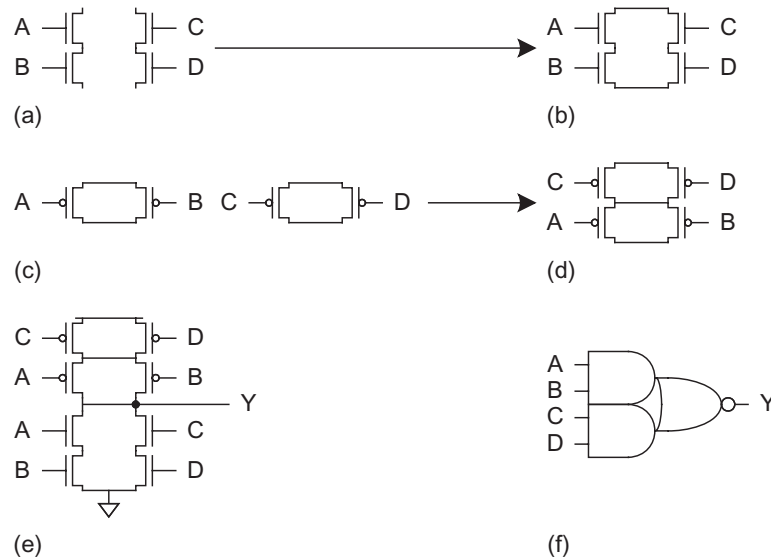


FIGURE 1.18 CMOS compound gate for function $Y = \overline{(A \cdot B) + (C \cdot D)}$

This AOI22 gate can be used as a 2-input inverting multiplexer by connecting $C = \bar{A}$ as a select signal. Then, $Y = \bar{B}$ if C is 0, while $Y = \bar{D}$ if C is 1. Section 1.4.8 shows a way to improve this multiplexer design.

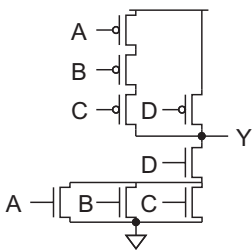


FIGURE 1.19
CMOS compound gate
for function
 $Y = (A + B + C) \cdot D$

Example 1.2

Sketch a static CMOS gate computing $Y = \overline{(A + B + C) \cdot D}$.

SOLUTION: Figure 1.19 shows such an OR-AND-INVERT-3-1 (OAI31) gate. The nMOS pull-down network pulls the output low if D is 1 and either A , B , and C are 1, so D is in series with the parallel combination of A , B , and C . The pMOS pull-up network is the conduction complement, so D must be in parallel with the series combination of A , B , and C .

1.4.6 Pass Transistors and Transmission Gates

The *strength* of a signal is measured by how closely it approximates an ideal voltage source. In general, the stronger a signal, the more current it can source or sink. The power supplies, or *rails*, (V_{DD} and GND) are the source of the strongest 1s and 0s.

An nMOS transistor is an almost perfect switch when passing a 0 and thus we say it passes a *strong* 0. However, the nMOS transistor is imperfect at passing a 1. The high voltage level is somewhat less than V_{DD} , as will be explained in Section 2.5.4. We say it passes a *degraded* or *weak* 1. A pMOS transistor again has the opposite behavior, passing strong 1s but degraded 0s. The transistor symbols and behaviors are summarized in Figure 1.20 with g , s , and d indicating gate, source, and drain.

When an nMOS or pMOS is used alone as an imperfect switch, we sometimes call it a *pass transistor*. By combining an nMOS and a pMOS transistor in parallel (Figure 1.21(a)), we obtain a switch that turns on when a 1 is applied to g (Figure 1.21(b)) in which 0s and 1s are both passed in an acceptable fashion (Figure 1.21(c)). We term this a *transmission gate* or *pass gate*. In a circuit where only a 0 or a 1 has to be passed, the appropriate transistor (n or p) can be deleted, reverting to a single nMOS or pMOS device.

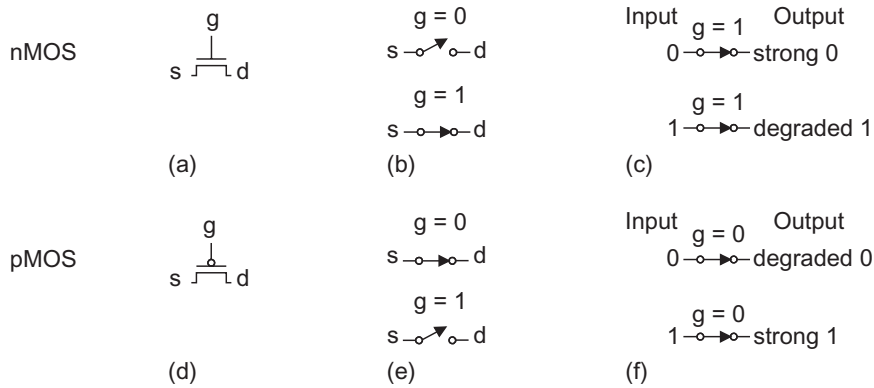


FIGURE 1.20 Pass transistor strong and degraded outputs

Note that both the control input and its complement are required by the transmission gate. This is called *double rail* logic. Some circuit symbols for the transmission gate are shown in Figure 1.21(d).¹ None are easier to draw than the simple schematic, so we will use the schematic version to represent a transmission gate in this book.

In all of our examples so far, the inputs drive the gate terminals of nMOS transistors in the pull-down network and pMOS transistors in the complementary pull-up network, as was shown in Figure 1.14. Thus, the nMOS transistors only need to pass 0s and the pMOS only pass 1s, so the output is always strongly driven and the levels are never degraded. This is called a *fully restored* logic gate and simplifies circuit design considerably. In contrast to other forms of logic, where the pull-up and pull-down switch networks have to be ratioed in some manner, static CMOS gates operate correctly independently of the physical sizes of the transistors. Moreover, there is never a path through ‘ON’ transistors from the 1 to the 0 supplies for any combination of inputs (in contrast to single-channel MOS, GaAs technologies, or bipolar). As we will find in subsequent chapters, this is the basis for the low static power dissipation in CMOS.

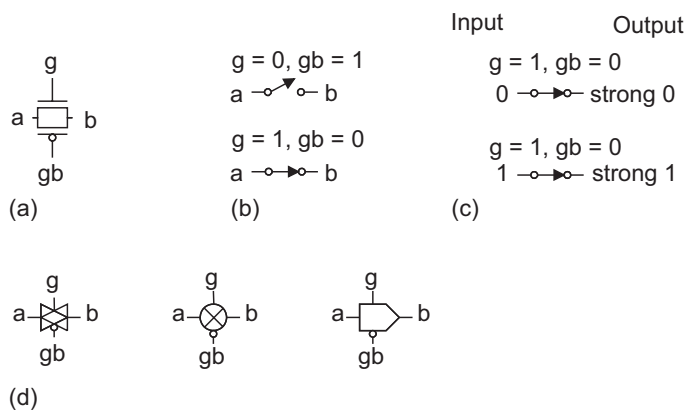


FIGURE 1.21 Transmission gate

¹We call the left and right terminals a and b because each is technically the source of one of the transistors and the drain of the other.

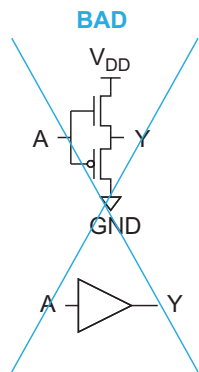


FIGURE 1.22
Bad noninverting buffer

A consequence of the design of static CMOS gates is that they must be inverting. The nMOS pull-down network turns ON when inputs are 1, leading to 0 at the output. We might be tempted to turn the transistors upside down to build a noninverting gate. For example, Figure 1.22 shows a noninverting buffer. Unfortunately, now both the nMOS and pMOS transistors produce degraded outputs, so the technique should be avoided. Instead, we can build noninverting functions from multiple stages of inverting gates. Figure 1.23 shows several ways to build a 4-input AND gate from two levels of inverting static CMOS gates. Each design has different speed, size, and power trade-offs.

Similarly, the compound gate of Figure 1.18 could be built with two AND gates, an OR gate, and an inverter. The AND and OR gates in turn could be constructed from NAND/NOR gates and inverters, as shown in Figure 1.24, using a total of 20 transistors, as compared to eight in Figure 1.18. Good CMOS logic designers exploit the efficiencies of compound gates rather than using large numbers of AND/OR gates.

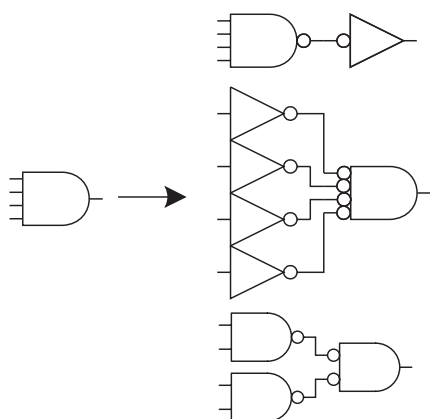


FIGURE 1.23 Various implementations
of a CMOS 4-input AND gate

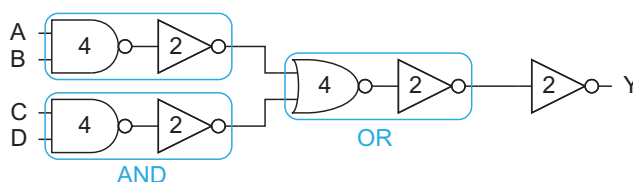


FIGURE 1.24 Inefficient discrete gate implementation of AOI22
with transistor counts indicated

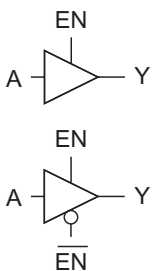


FIGURE 1.25
Tristate buffer
symbol

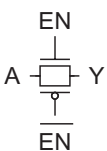


FIGURE 1.26
Transmission gate

1.4.7 Tristates

Figure 1.25 shows symbols for a *tristate buffer*. When the enable input EN is 1, the output Y equals the input A, just as in an ordinary buffer. When the enable is 0, Y is left floating (a 'Z' value). This is summarized in Table 1.5. Sometimes both true and complementary enable signals EN and $\overline{\text{EN}}$ are drawn explicitly, while sometimes only EN is shown.

TABLE 1.5 Truth table for tristate

EN / $\overline{\text{EN}}$	A	Y
0 / 1	0	Z
0 / 1	1	Z
1 / 0	0	0
1 / 0	1	1

The transmission gate in Figure 1.26 has the same truth table as a tristate buffer. It only requires two transistors but it is a *nonrestoring* circuit. If the input is noisy or otherwise degraded, the output will receive the same noise. We will see in Section 4.4.2 that the delay of a series of nonrestoring gates increases rapidly with the number of gates.

Figure 1.27(a) shows a *tristate inverter*. The output is actively driven from V_{DD} or GND, so it is a restoring logic gate. Unlike any of the gates considered so far, the tristate inverter does not obey the conduction complements rule because it allows the output to float under certain input combinations. When EN is 0 (Figure 1.27(b)), both enable transistors are OFF, leaving the output floating. When EN is 1 (Figure 1.27(c)), both enable transistors are ON. They are conceptually removed from the circuit, leaving a simple inverter. Figure 1.27(d) shows symbols for the tristate inverter. The complementary enable signal can be generated internally or can be routed to the cell explicitly. A tristate buffer can be built as an ordinary inverter followed by a tristate inverter.

Tristates were once commonly used to allow multiple units to drive a common bus, as long as exactly one unit is enabled at a time. If multiple units drive the bus, contention occurs and power is wasted. If no units drive the bus, it can float to an invalid logic level that causes the receivers to waste power. Moreover, it can be difficult to switch enable signals at exactly the same time when they are distributed across a large chip. Delay between different enables switching can cause contention. Given these problems, multiplexers are now preferred over tristate busses.

1.4.8 Multiplexers

Multiplexers are key components in CMOS memory elements and data manipulation structures. A *multiplexer* chooses the output from among several inputs based on a select signal. A 2-input, or 2:1 multiplexer, chooses input $D0$ when the select is 0 and input $D1$ when the select is 1. The truth table is given in Table 1.6; the logic function is $Y = \bar{S} \cdot D0 + S \cdot D1$.

TABLE 1.6 Multiplexer truth table

$S / \bar{S}$	$D1$	$D0$	Y
0 / 1	X	0	0
0 / 1	X	1	1
1 / 0	0	X	0
1 / 0	1	X	1

Two transmission gates can be tied together to form a compact 2-input multiplexer, as shown in Figure 1.28(a). The select and its complement enable exactly one of the two transmission gates at any given time. The complementary select $\bar{S}$ is often not drawn in the symbol, as shown in Figure 1.28(b).

Again, the transmission gates produce a nonrestoring multiplexer. We could build a restoring, inverting multiplexer out of gates in several ways. One is the compound gate of Figure 1.18(e), connected as shown in Figure 1.29(a). Another is to gang together two tristate inverters, as shown in Figure 1.29(b). Notice that the schematics of these two approaches are nearly identical, save that the pull-up network has been slightly simplified and permuted in Figure 1.29(b). This is possible because the select and its complement are mutually exclusive. The tristate approach is slightly more compact and faster because it

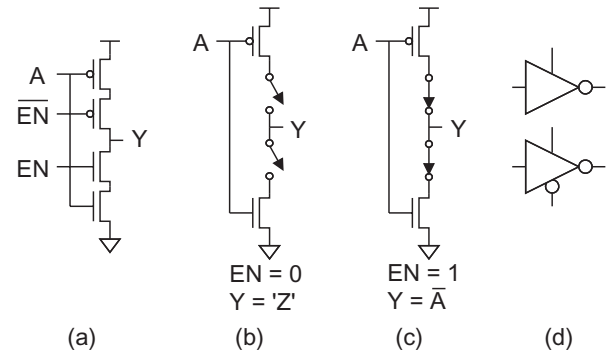
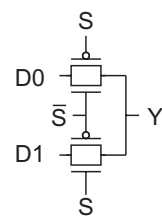
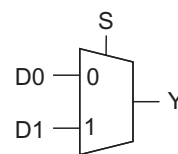


FIGURE 1.27 Tristate Inverter



(a)



(b)

FIGURE 1.28 Transmission gate multiplexer

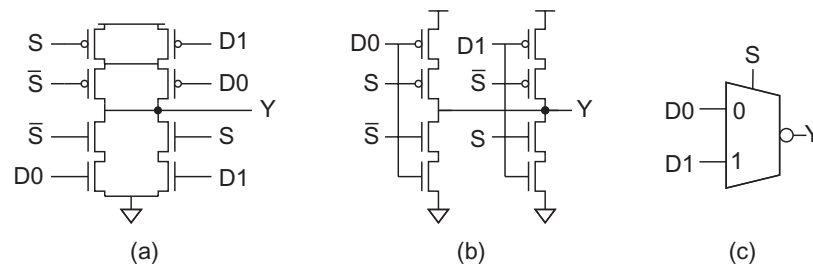


FIGURE 1.29 Inverting multiplexer

requires less internal wire. Again, if the complementary select is generated within the cell, it is omitted from the symbol (Figure 1.29(c)).

Larger multiplexers can be built from multiple 2-input multiplexers or by directly ganging together several tristates. The latter approach requires decoded enable signals for each tristate; the enables should switch simultaneously to prevent contention. 4-input (4:1) multiplexers using each of these approaches are shown in Figure 1.30. In practice, both inverting and noninverting multiplexers are simply called multiplexers or muxes.

1.4.9 Sequential Circuits

So far, we have considered *combinational circuits*, whose outputs depend only on the current inputs. *Sequential circuits* have memory: their outputs depend on both current and previous inputs. Using the combinational circuits developed so far, we can now build sequential circuits such as latches and flip-flops. These elements receive a clock, CLK , and a data input, D , and produce an output, Q . A D latch is *transparent* when $CLK = 1$, meaning that Q follows D . It becomes *opaque* when $CLK = 0$, meaning Q retains its previous value and ignores changes in D . An *edge-triggered flip-flop* copies D to Q on the rising edge of CLK and remembers its old value at other times.

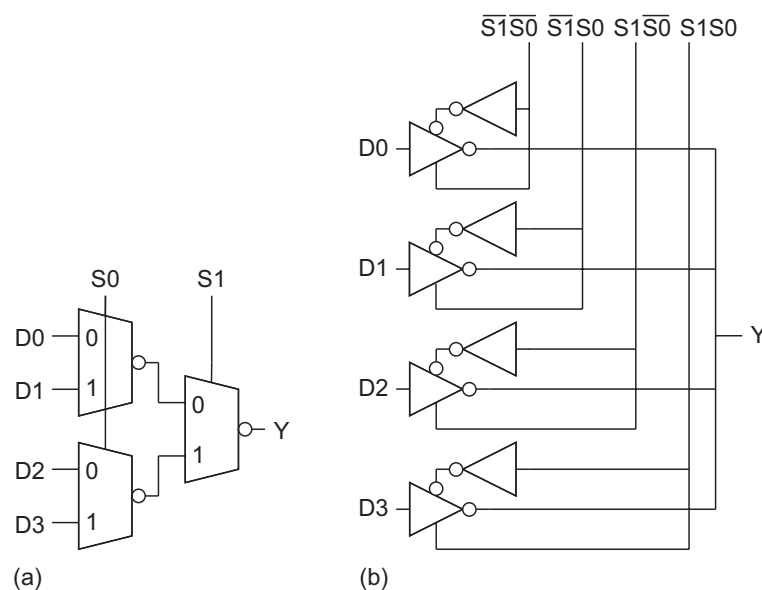


FIGURE 1.30 4:1 multiplexer

1.4.9.1 Latches A D latch built from a 2-input multiplexer and two inverters is shown in Figure 1.31(a). The multiplexer can be built from a pair of transmission gates, shown in Figure 1.31(b), because the inverters are restoring. This latch also produces a complementary output, $\bar{Q}$. When $CLK = 1$, the latch is transparent and D flows through to Q (Figure 1.31(c)). When CLK falls to 0, the latch becomes opaque. A feedback path around the inverter pair is established (Figure 1.31(d)) to hold the current state of Q indefinitely.

The D latch is also known as a *level-sensitive latch* because the state of the output is dependent on the level of the clock signal, as shown in Figure 1.31(e). The latch shown is a positive-level-sensitive latch, represented by the symbol in Figure 1.31(f). By inverting the control connections to the multiplexer, the latch becomes negative-level-sensitive.

1.4.9.2 Flip-Flops By combining two level-sensitive latches, one negative-sensitive and one positive-sensitive, we construct the edge-triggered flip-flop shown in Figure 1.32(a–b). The first latch stage is called the *master* and the second is called the *slave*.

While CLK is low, the master negative-level-sensitive latch output ($\bar{Q}$) follows the D input while the slave positive-level-sensitive latch holds the previous value (Figure 1.32(c)). When the clock transitions from 0 to 1, the master latch becomes opaque and holds the D value at the time of the clock transition. The slave latch becomes transparent, passing the stored master value ($\bar{Q}$) to the output of the slave latch (Q). The D input is blocked from affecting the output because the master is disconnected from the D input (Figure 1.32(d)). When the clock transitions from 1 to 0, the slave latch holds its value and the master starts sampling the input again.

While we have shown a transmission gate multiplexer as the input stage, good design practice would buffer the input and output with inverters, as shown in Figure 1.32(e), to

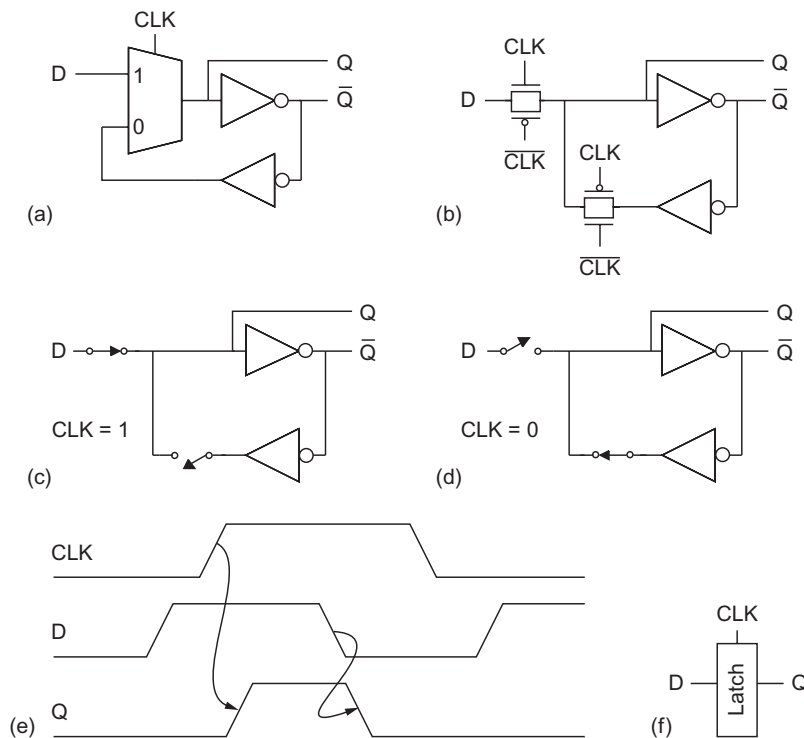


FIGURE 1.31 CMOS positive-level-sensitive D latch

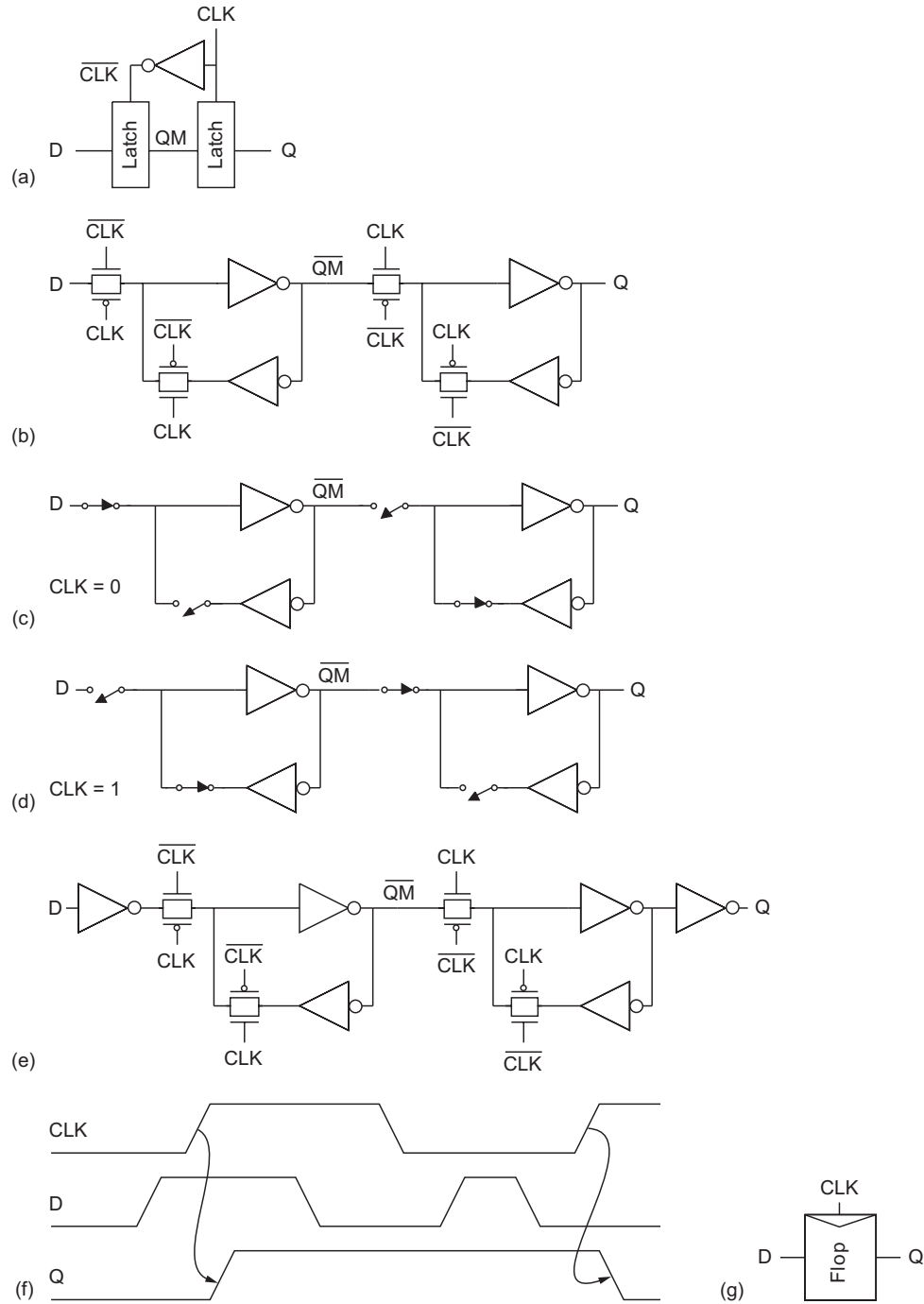


FIGURE 1.32 CMOS positive-edge-triggered *D* flip-flop

preserve what we call “modularity.” Modularity is explained further in Section 1.6.2 and robust latches and registers are discussed further in Section 10.3.

In summary, this flip-flop copies *D* to *Q* on the rising edge of the clock, as shown in Figure 1.32(f). Thus, this device is called a positive-edge triggered flip-flop (also called a *D flip-flop*, *D register*, or *master-slave flip-flop*). Figure 1.32(g) shows the circuit symbol for the flip-flop. By reversing the latch polarities, a negative-edge triggered flip-flop may be

constructed. A collection of D flip-flops sharing a common clock input is called a *register*. A register is often drawn as a flip-flop with multi-bit D and Q busses.

In Section 10.2.5 we will see that flip-flops may experience hold-time failures if the system has too much *clock skew*, i.e., if one flip-flop triggers early and another triggers late because of variations in clock arrival times. In industrial designs, a great deal of effort is devoted to timing simulations to catch hold-time problems. When design time is more important (e.g., in class projects), hold-time problems can be avoided altogether by distributing a two-phase nonoverlapping clock. Figure 1.33 shows the flip-flop clocked with two nonoverlapping phases. As long as the phases never overlap, at least one latch will be opaque at any given time and hold-time problems cannot occur.

1.5 CMOS Fabrication and Layout

Now that we can design logic gates and registers from transistors, let us consider how the transistors are built. Designers need to understand the physical implementation of circuits because it has a major impact on performance, power, and cost.

Transistors are fabricated on thin silicon wafers that serve as both a mechanical support and an electrical common point called the *substrate*. We can examine the physical layout of transistors from two perspectives. One is the top view, obtained by looking down on a wafer. The other is the cross-section, obtained by slicing the wafer through the middle of a transistor and looking at it edgewise. We begin by looking at the cross-section of a complete CMOS inverter. We then look at the top view of the same inverter and define a set of masks used to manufacture the different parts of the inverter. The size of the transistors and wires is set by the mask dimensions and is limited by the resolution of the manufacturing process. Continual advancements in this resolution have fueled the exponential growth of the semiconductor industry.

1.5.1 Inverter Cross-Section

Figure 1.34 shows a cross-section and corresponding schematic of an inverter. (See the inside front cover for a color cross-section.) In this diagram, the inverter is built on a p-type substrate. The pMOS transistor requires an n-type body region, so an n-well is diffused into the substrate in its vicinity. As described in Section 1.3, the nMOS transistor

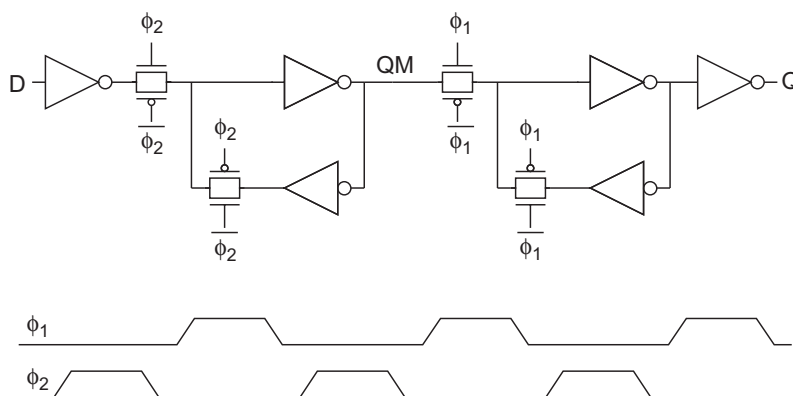


FIGURE 1.33 CMOS flip-flop with two-phase nonoverlapping clocks

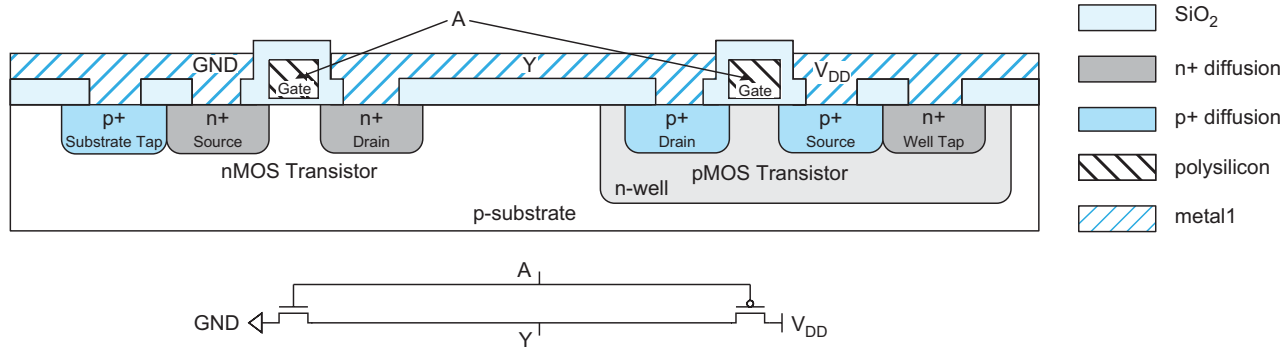


FIGURE 1.34 Inverter cross-section with well and substrate contacts. Color version on inside front cover.

has heavily doped n-type source and drain regions and a polysilicon gate over a thin layer of silicon dioxide (SiO_2 , also called *gate oxide*). n+ and p+ diffusion regions indicate heavily doped n-type and p-type silicon. The pMOS transistor is a similar structure with p-type source and drain regions. The polysilicon gates of the two transistors are tied together somewhere off the page and form the input A . The source of the nMOS transistor is connected to a metal ground line and the source of the pMOS transistor is connected to a metal V_{DD} line. The drains of the two transistors are connected with metal to form the output Y . A thick layer of SiO_2 called *field oxide* prevents metal from shorting to other layers except where contacts are explicitly etched.

A junction between metal and a lightly doped semiconductor forms a *Schottky diode* that only carries current in one direction. When the semiconductor is doped more heavily, it forms a good ohmic contact with metal that provides low resistance for bidirectional current flow. The substrate must be tied to a low potential to avoid forward-biasing the p-n junction between the p-type substrate and the n+ nMOS source or drain. Likewise, the n-well must be tied to a high potential. This is done by adding heavily doped substrate and well contacts, or *taps*, to connect GND and V_{DD} to the substrate and n-well, respectively.

1.5.2 Fabrication Process

For all their complexity, chips are amazingly inexpensive because all the transistors and wires can be printed in much the same way as books. The fabrication sequence consists of a series of steps in which layers of the chip are defined through a process called *photolithography*. Because a whole wafer full of chips is processed in each step, the cost of the chip is proportional to the chip area, rather than the number of transistors. As manufacturing advances allow engineers to build smaller transistors and thus fit more in the same area, each transistor gets cheaper. Smaller transistors are also faster because electrons don't have to travel as far to get from the source to the drain, and they consume less energy because fewer electrons are needed to charge up the gates! This explains the remarkable trend for computers and electronics to become cheaper and more capable with each generation.

The inverter could be defined by a hypothetical set of six masks: n-well, polysilicon, n+ diffusion, p+ diffusion, contacts, and metal (for fabrication reasons discussed in Chapter 3, the actual mask set tends to be more elaborate). Masks specify where the components will be manufactured on the chip. Figure 1.35(a) shows a top view of the six masks. (See also the inside front cover for a color picture.) The cross-section of the inverter from Figure 1.34 was taken along the dashed line. Take some time to convince yourself how the top view and cross-section relate; this is critical to understanding chip layout.

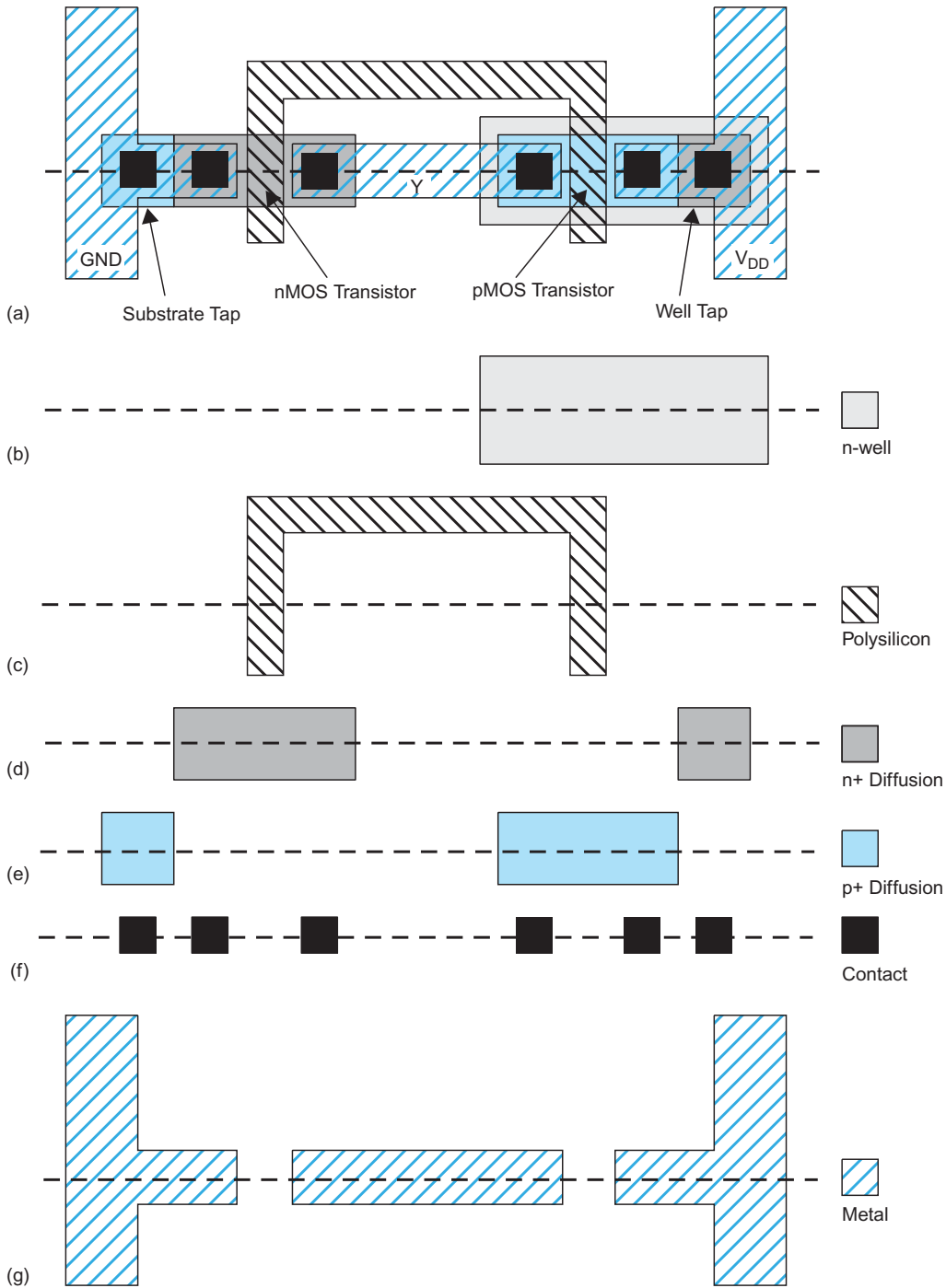


FIGURE 1.35 Inverter mask set. Color version on inside front cover.

Consider a simple fabrication process to illustrate the concept. The process begins with the creation of an n-well on a bare p-type silicon wafer. Figure 1.36 shows cross-sections of the wafer after each processing step involved in forming the n-well; Figure 1.36(a) illustrates the bare substrate before processing. Forming the n-well requires adding enough Group V dopants into the silicon substrate to change the substrate from p-type to n-type in the region of the well. To define what regions receive n-wells, we grow a protective layer of

oxide over the entire wafer, then remove it where we want the wells. We then add the n-type dopants; the dopants are blocked by the oxide, but enter the substrate and form the wells where there is no oxide. The next paragraph describes these steps in detail.

The wafer is first *oxidized* in a high-temperature (typically 900–1200 °C) furnace that causes Si and O₂ to react and become SiO₂ on the wafer surface (Figure 1.36(b)). The oxide must be *patterned* to define the n-well. An organic photoresist² that softens where

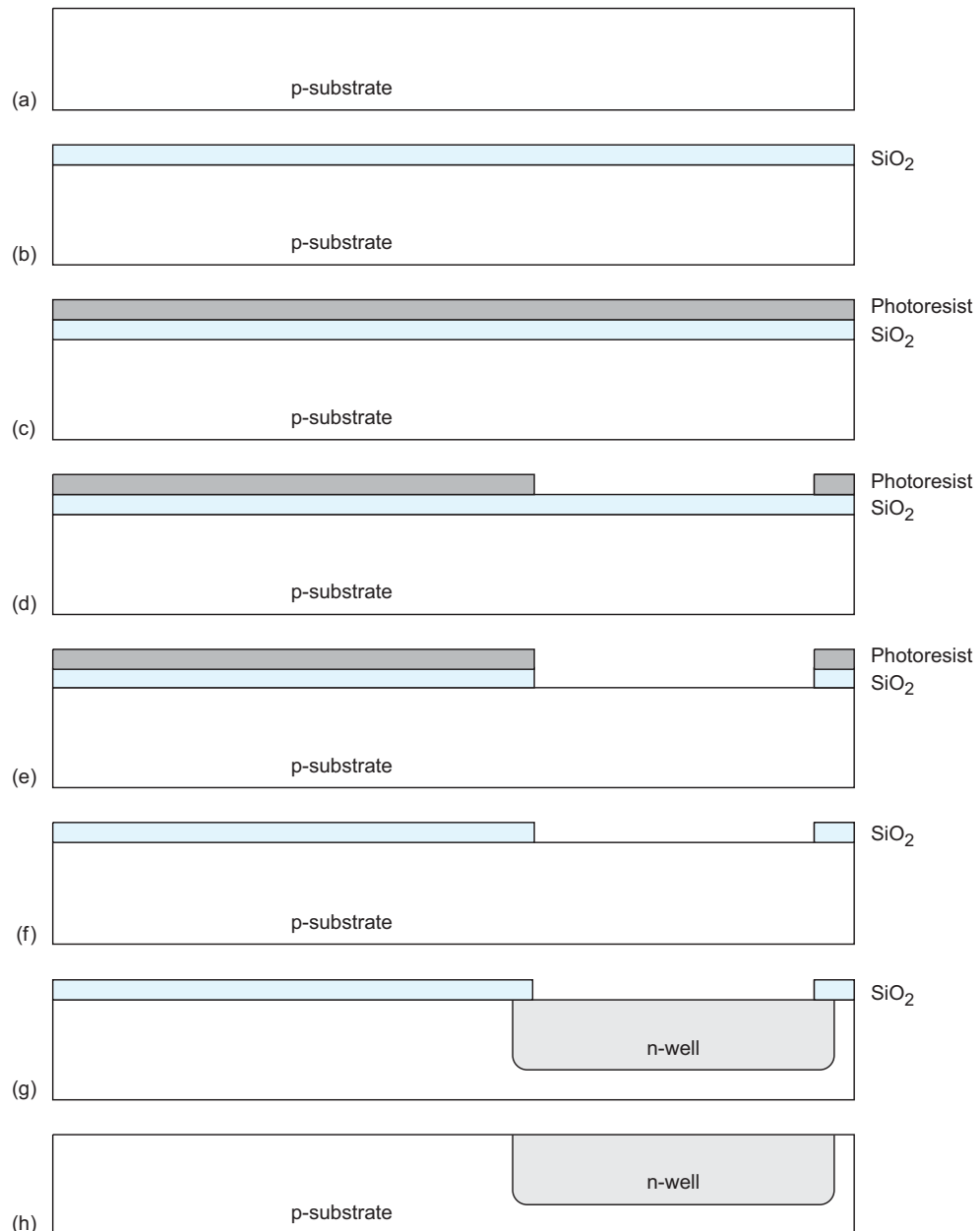


FIGURE 1.36 Cross-sections while manufacturing the n-well

²Engineers have experimented with many organic polymers for photoresists. In 1958, Brumford and Walker reported that Jello™ could be used for masking. They did extensive testing, observing that “various Jellos™ were evaluated with lemon giving the best result.”

exposed to light is spun onto the wafer (Figure 1.36(c)). The photoresist is exposed through the n-well mask (Figure 1.35(b)) that allows light to pass through only where the well should be. The softened photoresist is removed to expose the oxide (Figure 1.36(d)). The oxide is etched with hydrofluoric acid (HF) where it is not protected by the photoresist (Figure 1.36(e)), then the remaining photoresist is stripped away using a mixture of acids called *piranha etch* (Figure 1.36(f)). The well is formed where the substrate is not covered with oxide. Two ways to add dopants are diffusion and ion implantation. In the *diffusion* process, the wafer is placed in a furnace with a gas containing the dopants. When heated, dopant atoms diffuse into the substrate. Notice how the well is wider than the hole in the oxide on account of *lateral diffusion* (Figure 1.36(g)). With *ion implantation*, dopant ions are accelerated through an electric field and blasted into the substrate. In either method, the oxide layer prevents dopant atoms from entering the substrate where no well is intended. Finally, the remaining oxide is stripped with HF to leave the bare wafer with wells in the appropriate places.

The transistor gates are formed next. These consist of polycrystalline silicon, generally called *polysilicon*, over a thin layer of oxide. The thin oxide is grown in a furnace. Then the wafer is placed in a reactor with silane gas (SiH_4) and heated again to grow the polysilicon layer through a process called *chemical vapor deposition*. The polysilicon is heavily doped to form a reasonably good conductor. The resulting cross-section is shown in Figure 1.37(a). As before, the wafer is patterned with photoresist and the polysilicon mask (Figure 1.35(c)), leaving the polysilicon gates atop the thin gate oxide (Figure 1.37(b)).

The n+ regions are introduced for the transistor active area and the well contact. As with the well, a protective layer of oxide is formed (Figure 1.37(c)) and patterned with the n-diffusion mask (Figure 1.35(d)) to expose the areas where the dopants are needed (Figure 1.37(d)). Although the n+ regions in Figure 1.37(e) are typically formed with ion implantation, they were historically diffused and thus still are often called *n-diffusion*. Notice that the polysilicon gate over the nMOS transistor blocks the diffusion so the source and drain are separated by a channel under the gate. This is called a *self-aligned* process because the source and drain of the transistor are automatically formed adjacent to the gate without the need to precisely align the masks. Finally, the protective oxide is stripped (Figure 1.37(f)).

The process is repeated for the p-diffusion mask (Figure 1.35(e)) to give the structure of Figure 1.38(a). Oxide is used for masking in the same way, and thus is not shown. The field oxide is grown to insulate the wafer from metal and patterned with the contact mask (Figure 1.35(f)) to leave contact cuts where metal should attach to diffusion or polysilicon (Figure 1.38(b)). Finally, aluminum is sputtered over the entire wafer, filling the contact cuts as well. Sputtering involves blasting aluminum into a vapor that evenly coats the wafer. The metal is patterned with the metal mask (Figure 1.35(g)) and plasma etched to remove metal everywhere except where wires should remain (Figure 1.38(c)). This completes the simple fabrication process.

Modern fabrication sequences are more elaborate because they must create complex doping profiles around the channel of the transistor and print features that are smaller than the wavelength of the light being used in lithography. However, masks for these elaborations can be automatically generated from the simple set of masks we have just examined. Modern processes also have 5–10+ layers of metal, so the metal and contact steps must be repeated for each layer. Chip manufacturing has become a commodity, and many different foundries will build designs from a basic set of masks.

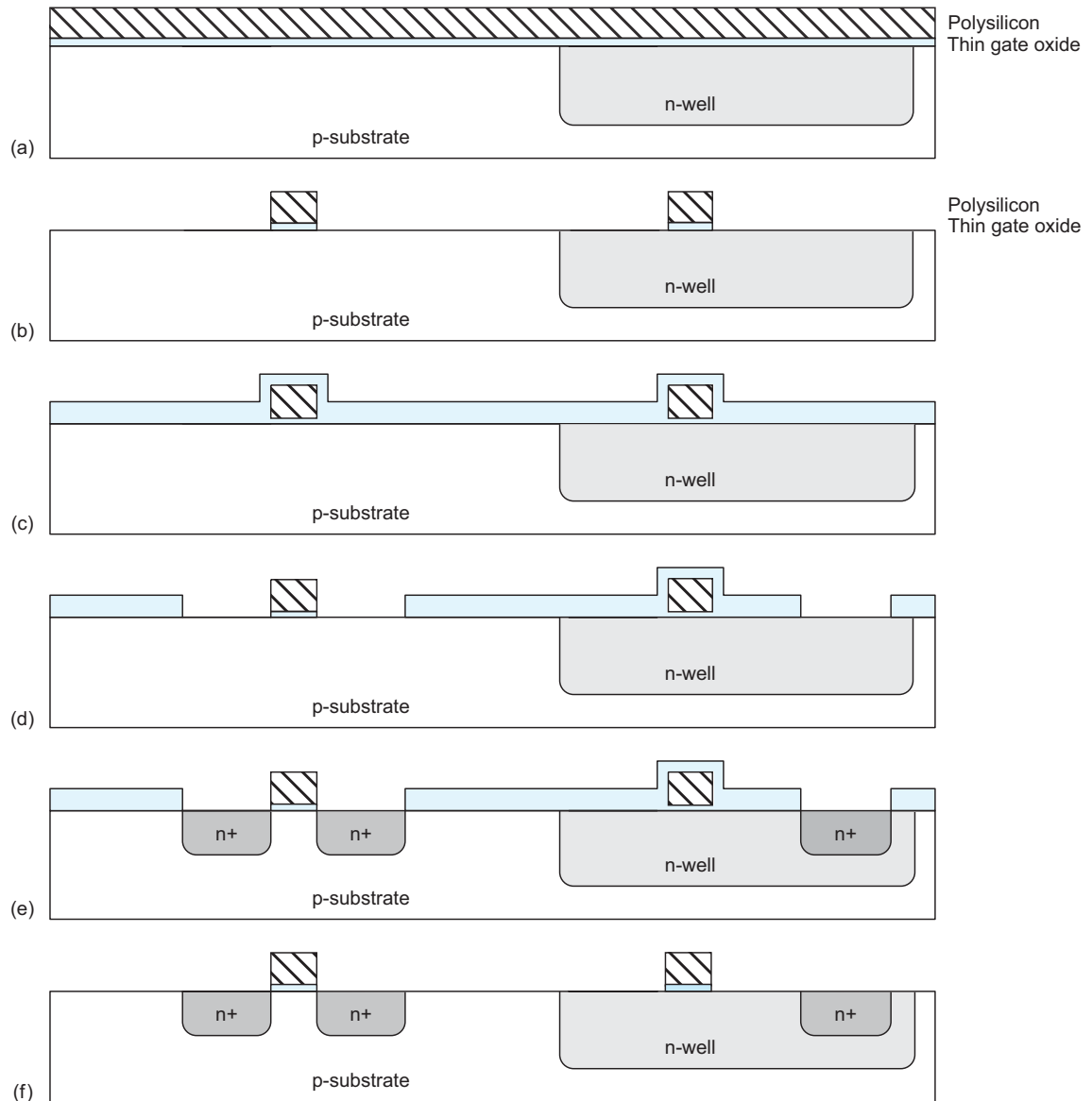


FIGURE 1.37 Cross-sections while manufacturing polysilicon and n-diffusion

1.5.3 Layout Design Rules

Layout design rules describe how small features can be and how closely they can be reliably packed in a particular manufacturing process. Industrial design rules are usually specified in microns. This makes migrating from one process to a more advanced process or a different foundry's process difficult because not all rules scale in the same way.

Universities sometimes simplify design by using scalable design rules that are conservative enough to apply to many manufacturing processes. Mead and Conway [Mead80] popularized scalable design rules based on a single parameter, λ , that characterizes the resolution of the process. λ is generally half of the minimum drawn transistor channel length. This length is the distance between the source and drain of a transistor and is set by the minimum width of a polysilicon wire. For example, a 180 nm process has a minimum polysilicon width (and hence transistor length) of 0.18 μm and uses design rules with

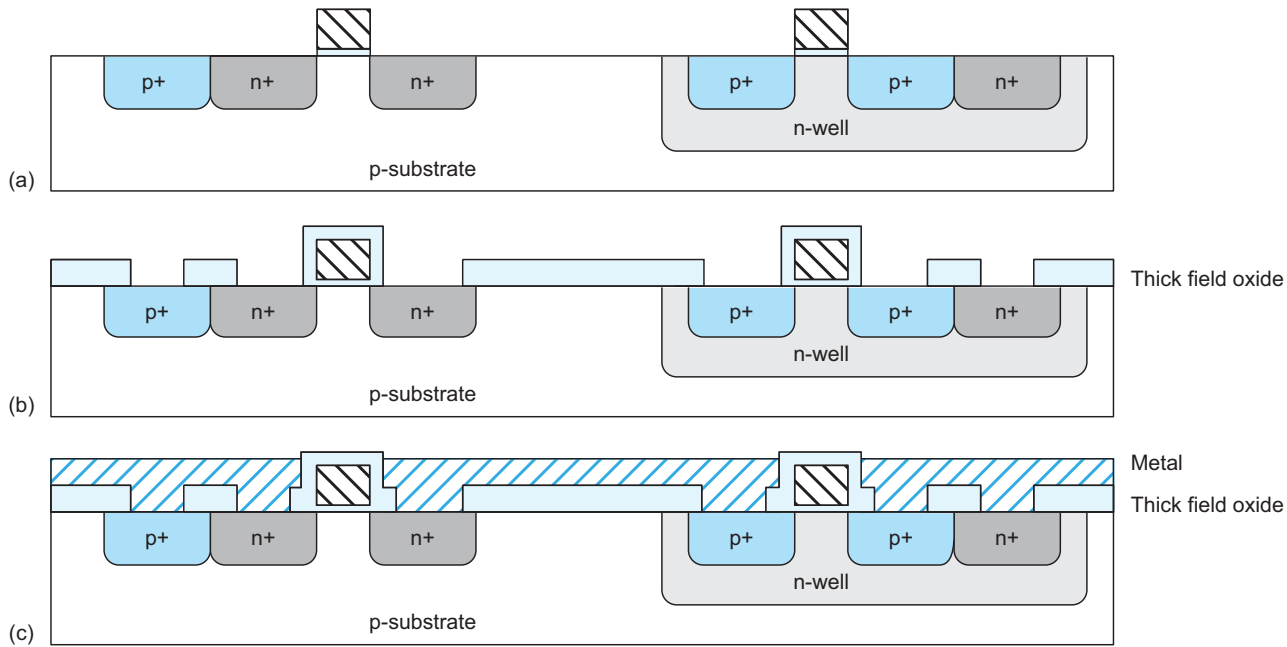


FIGURE 1.38 Cross-sections while manufacturing p-diffusion, contacts, and metal

$\lambda = 0.09 \mu\text{m}$.³ Lambda-based rules are necessarily conservative because they round up dimensions to an integer multiple of λ . However, they make scaling layout trivial; the same layout can be moved to a new process simply by specifying a new value of λ . This chapter will present design rules in terms of λ . The potential density advantage of micron rules is sacrificed for simplicity and easy scalability of lambda rules. Designers often describe a process by its *feature size*. Feature size refers to minimum transistor length, so λ is half the feature size.

Unfortunately, below 180 nm, design rules have become so complex and process-specific that scalable design rules are difficult to apply. However, the intuition gained from a simple set of scalable rules is still a valuable foundation for understanding the more complex rules. Chapter 3 will examine some of these process-specific rules in more detail.

The MOSIS service [Piña02] is a low-cost prototyping service that collects designs from academic, commercial, and government customers and aggregates them onto one mask set to share overhead costs and generate production volumes sufficient to interest fabrication companies. MOSIS has developed a set of scalable lambda-based design rules that covers a wide range of manufacturing processes. The rules describe the minimum width to avoid breaks in a line, minimum spacing to avoid shorts between lines, and minimum overlap to ensure that two layers completely overlap.

A conservative but easy-to-use set of design rules for layouts with two metal layers in an n-well process is as follows:

- Metal and diffusion have minimum width and spacing of 4λ .
- Contacts are $2\lambda \times 2\lambda$ and must be surrounded by 1λ on the layers above and below.
- Polysilicon uses a width of 2λ .

³Some 180 nm lambda-based rules actually set $\lambda = 0.10 \mu\text{m}$, then shrink the gate by 20 nm while generating masks. This keeps 180 nm gate lengths but makes all other features slightly larger.

- Polysilicon overlaps diffusion by 2λ where a transistor is desired and has a spacing of 1λ away where no transistor is desired.
- Polysilicon and contacts have a spacing of 3λ from other polysilicon or contacts.
- N-well surrounds pMOS transistors by 6λ and avoids nMOS transistors by 6λ .

Figure 1.39 shows the basic MOSIS design rules for a process with two metal layers. Section 3.3 elaborates on these rules and compares them with industrial design rules.

In a three-level metal process, the width of the third layer is typically 6λ and the spacing 4λ . In general, processes with more layers often provide thicker and wider top-level metal that has a lower resistance.

Transistor dimensions are often specified by their Width/Length (W/L) ratio. For example, the nMOS transistor in Figure 1.39 formed where polysilicon crosses n-diffusion has a W/L of $4/2$. In a $0.6\mu\text{m}$ process, this corresponds to an actual width of $1.2\mu\text{m}$ and a length of $0.6\mu\text{m}$. Such a minimum-width contacted transistor is often called a unit transistor.⁴ pMOS transistors are often wider than nMOS transistors because holes move more slowly than electrons so the transistor has to be wider to deliver the same current. Figure 1.40(a) shows a unit inverter layout with a unit nMOS transistor and a double-sized pMOS transistor. Figure 1.40(b) shows a schematic for the inverter annotated with Width/Length for each transistor. In digital systems, transistors are typically chosen to have the minimum possible length because short-channel transistors are faster, smaller, and consume less power. Figure 1.40(c) shows a shorthand we will often use, specifying multiples of unit width and assuming minimum length.

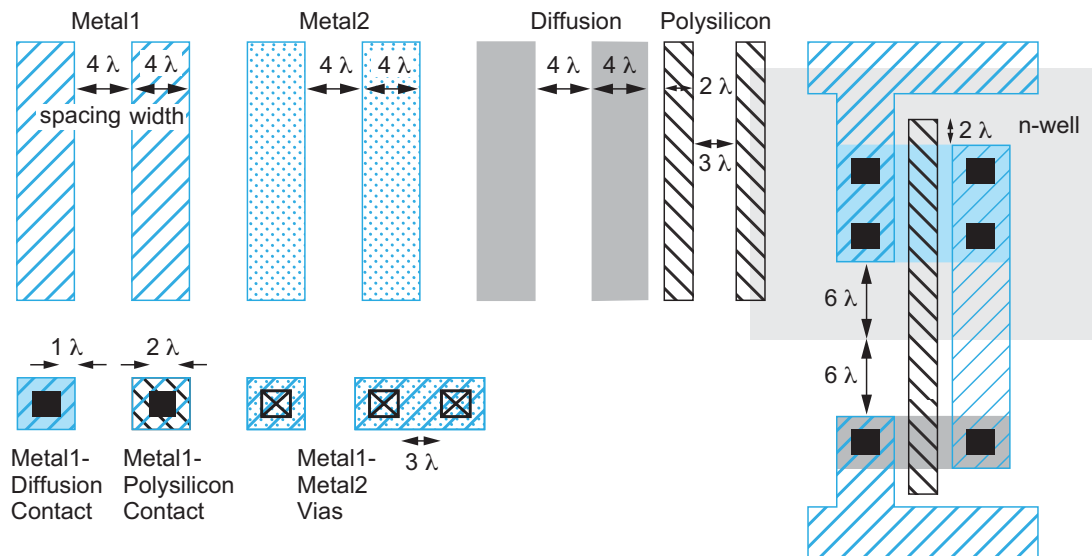


FIGURE 1.39 Simplified λ -based design rules

⁴Such small transistors in modern processes often behave slightly differently than their wider counterparts. Moreover, the transistor will not operate if either contact is damaged. Industrial designers often use a transistor wide enough for two contacts (9λ) as the unit transistor to avoid these problems.

1.5.4 Gate Layouts

A good deal of ingenuity can be exercised and a vast amount of time wasted exploring layout topologies to minimize the size of a gate or other *cell* such as an adder or memory element. For many applications, a straightforward layout is good enough and can be automatically generated or rapidly built by hand. This section presents a simple layout style based on a “line of diffusion” rule that is commonly used for standard cells in automated layout systems. This style consists of four horizontal strips: metal ground at the bottom of the cell, n-diffusion, p-diffusion, and metal power at the top. The power and ground lines are often called *supply rails*. Polysilicon lines run vertically to form transistor gates. Metal wires within the cell connect the transistors appropriately.

Figure 1.41(a) shows such a layout for an inverter. The input A can be connected from the top, bottom, or left in polysilicon. The output Y is available at the right side of the cell in metal. Recall that the p-substrate and n-well must be tied to ground and power, respectively. Figure 1.41(b) shows the same inverter with well and substrate taps placed under the power and ground rails, respectively. Figure 1.42 shows a 3-input NAND gate. Notice how the nMOS transistors are connected in series while the pMOS transistors are connected in parallel. Power and ground extend 2λ on each side so if two gates were abutted the contents would be separated by 4λ , satisfying design rules. The height of the cell is 36λ , or 40λ if the 4λ space between the cell and another wire above it is counted. All these examples use transistors of width 4λ . Choice of transistor width is addressed further in Chapters 4–5 and cell layout styles are discussed in Section 14.7.

These cells were designed such that the gate connections are made from the top or bottom in polysilicon. In contemporary standard cells, polysilicon is generally not used as a routing layer so the cell must allow metal2 to metal1 and metal1 to polysilicon contacts

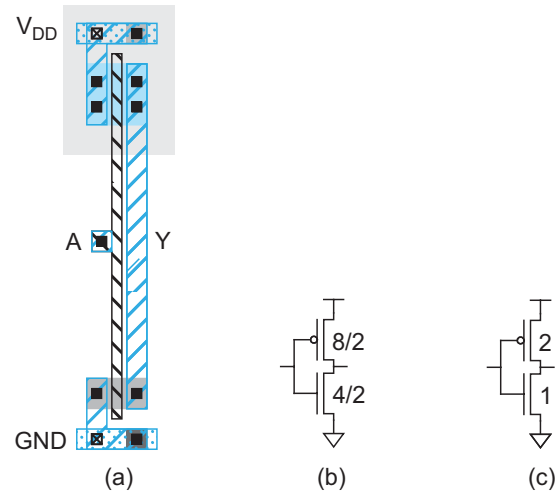


FIGURE 1.40 Inverter with dimensions labeled

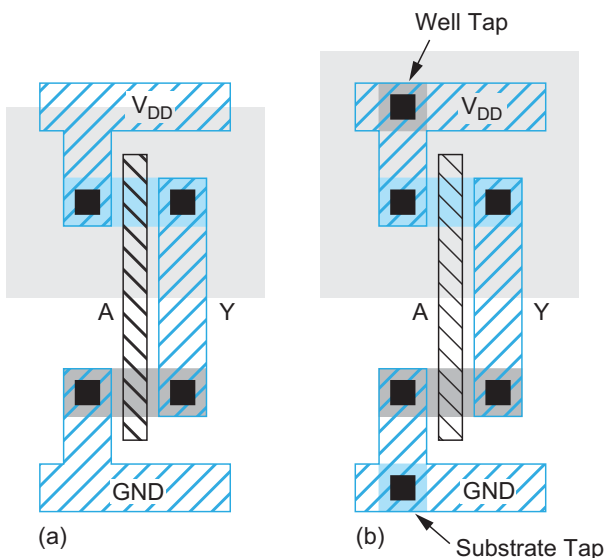


FIGURE 1.41 Inverter cell layout

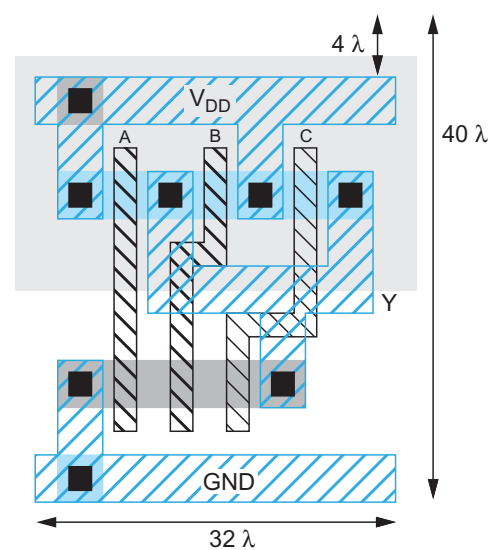


FIGURE 1.42 3-input NAND standard cell gate layouts

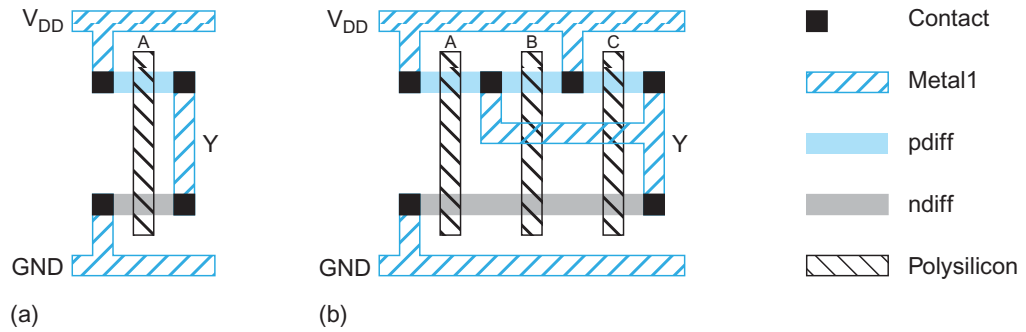


FIGURE 1.43 Stick diagrams of inverter and 3-input NAND gate. Color version on inside front cover.

to each gate. While this increases the size of the cell, it allows free access to all terminals on metal routing layers.

1.5.5 Stick Diagrams

Because layout is time-consuming, designers need fast ways to plan cells and estimate area before committing to a full layout. *Stick diagrams* are easy to draw because they do not need to be drawn to scale. Figure 1.43 and the inside front cover show stick diagrams for an inverter and a 3-input NAND gate. While this book uses stipple patterns, layout designers use dry-erase markers or colored pencils.

With practice, it is easy to estimate the area of a layout from the corresponding stick diagram even though the diagram is not to scale. Although schematics focus on transistors, layout area is usually determined by the metal wires. Transistors are merely widgets that fit under the wires. We define a *routing track* as enough space to place a wire and the required spacing to the next wire. If our wires have a width of 4λ and a spacing of 4λ to the next wire, the track *pitch* is 8λ , as shown in Figure 1.44(a). This pitch also leaves room for a transistor to be placed between the wires (Figure 1.44(b)). Therefore, it is reasonable to estimate the height and width of a cell by counting the number of metal tracks and multiplying by 8λ . A slight complication is the required spacing of 12λ between nMOS and pMOS transistors set by the well, as shown in Figure 1.45(a). This space can be occupied by an additional track of wire, shown in Figure 1.45(b). Therefore, an extra track must be allocated between nMOS and pMOS transistors regardless of whether wire is actually used in that track. Figure 1.46 shows how to count tracks to estimate the size of a 3-input NAND. There are four vertical wire tracks, multiplied by 8λ per track to give a cell width of 32λ . There are five horizontal tracks, giving a cell height of 40λ . Even though the horizontal tracks are not drawn to scale, they are still easy to count. Figure 1.42

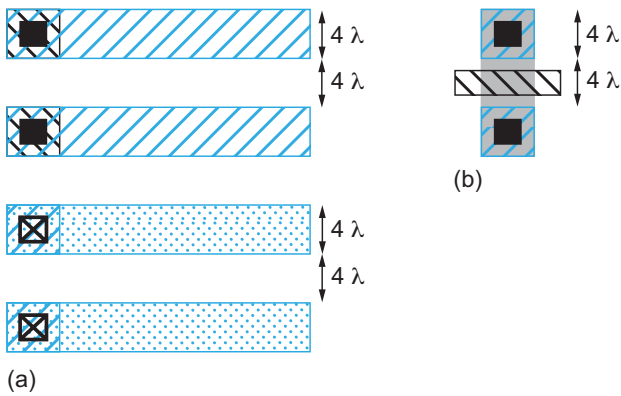


FIGURE 1.44 Pitch of routing tracks

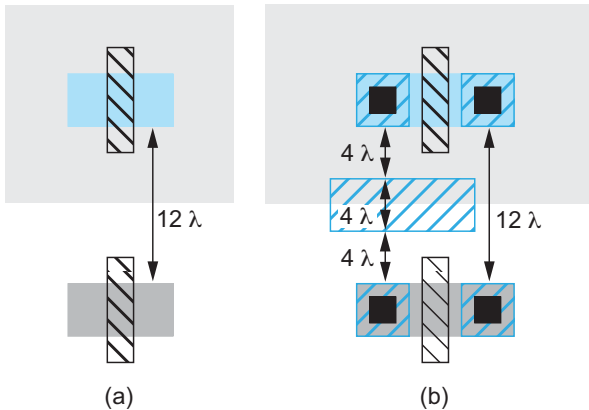


FIGURE 1.45 Spacing between nMOS and pMOS transistors

shows that the actual NAND gate layout matches the dimensions predicted by the stick diagram. If transistors are wider than 4λ , the extra width must be factored into the area estimate. Of course, these estimates are oversimplifications of the complete design rules and a trial layout should be performed for truly critical cells.

Example 1.3

Sketch a stick diagram for a CMOS gate computing $Y = \overline{(A + B + C)} \cdot \overline{D}$ (see Figure 1.18) and estimate the cell width and height.

SOLUTION: Figure 1.47 shows a stick diagram. Counting horizontal and vertical pitches gives an estimated cell size of 40 by 48 λ .

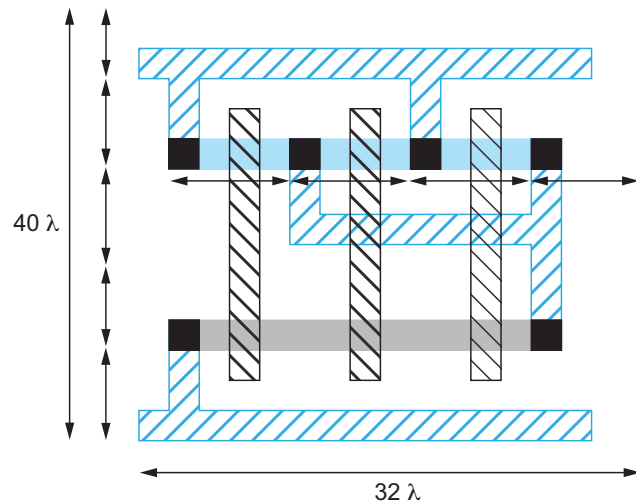


FIGURE 1.46 3-input NAND gate area estimation

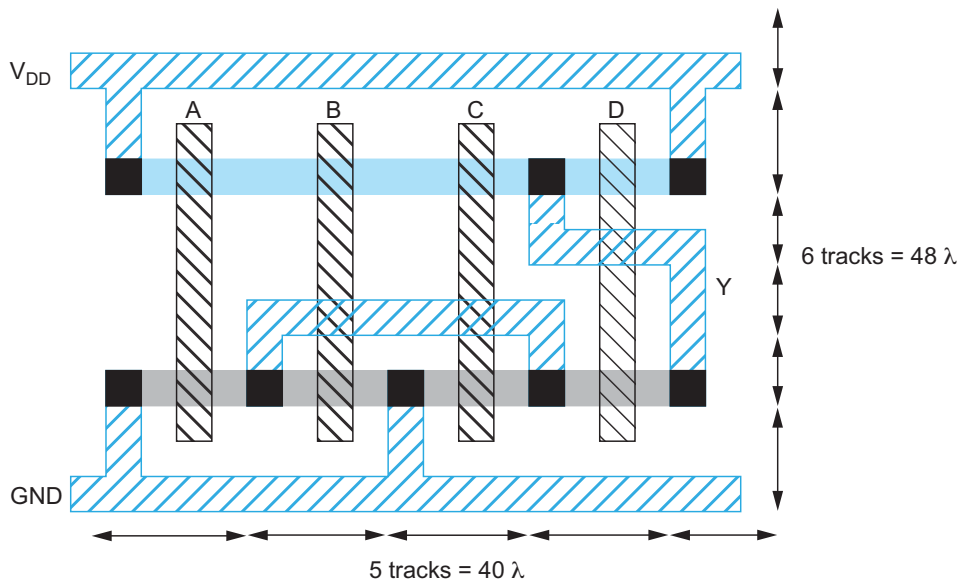


FIGURE 1.47 CMOS compound gate for function $Y = \overline{(A + B + C)} \cdot \overline{D}$

1.6 Design Partitioning

By this point, you know that MOS transistors behave as voltage-controlled switches. You know how to build logic gates out of transistors. And you know how transistors are fabricated and how to draw a layout that specifies how transistors should be placed and connected together. You know enough to start building your own simple chips.

The greatest challenge in modern VLSI design is not in designing the individual transistors but rather in managing system complexity. Modern *System-On-Chip* (SOC) designs combine memories, processors, high-speed I/O interfaces, and dedicated application-specific logic on a single chip. They use hundreds of millions or billions of transistors and cost tens of millions of dollars (or more) to design. The implementation

must be divided among large teams of engineers and each engineer must be highly productive. If the implementation is too rigidly partitioned, each block can be optimized without regard to its neighbors, leading to poor system results. Conversely, if every task is interdependent with every other task, design will progress too slowly. Design managers face the challenge of choosing a suitable trade-off between these extremes. There is no substitute for practical experience in making these choices, and talented engineers who have experience with multiple designs are very important to the success of a large project. Design proceeds through multiple levels of abstraction, hiding details until they become necessary. The practice of *structured design*, which is also used in large software projects, uses the principles of hierarchy, regularity, modularity, and locality to manage the complexity.

1.6.1 Design Abstractions

Digital VLSI design is often partitioned into five levels of abstractions: *architecture* design, *microarchitecture* design, *logic* design, *circuit* design, and *physical* design. Architecture describes the functions of the system. For example, the x86 microprocessor architecture specifies the instruction set, register set, and memory model. Microarchitecture describes how the architecture is partitioned into registers and functional units. The 80386, 80486, Pentium, Pentium II, Pentium III, Pentium 4, Core, Core 2, Atom, Cyrix MII, AMD Athlon, and Phenom are all microarchitectures offering different performance / transistor count / power trade-offs for the x86 architecture. Logic describes how functional units are constructed. For example, various logic designs for a 32-bit adder in the x86 integer unit include ripple carry, carry lookahead, and carry select. Circuit design describes how transistors are used to implement the logic. For example, a carry lookahead adder can use static CMOS circuits, domino circuits, or pass transistors. The circuits can be tailored to emphasize high performance or low power. Physical design describes the layout of the chip. Analog and RF VLSI design involves the same steps but with different layers of abstraction.

These elements are inherently interdependent and all influence each of the design objectives. For example, choices of microarchitecture and logic are strongly dependent on the number of transistors that can be placed on the chip, which depends on the physical design and process technology. Similarly, innovative circuit design that reduces a cache access from two cycles to one can influence which microarchitecture is most desirable. The choice of clock frequency depends on a complex interplay of microarchitecture and logic, circuit design, and physical design. Deeper pipelines allow higher frequencies but consume more power and lead to greater performance penalties when operations early in the pipeline are dependent on those late in the pipeline. Many functions have various logic and circuit designs trading speed for area, power, and design effort. Custom physical design allows more compact, faster circuits and lower manufacturing costs, but involves an enormous labor cost. Automatic layout with CAD systems reduces the labor and achieves faster times to market.

To deal with these interdependencies, microarchitecture, logic, circuit, and physical design must occur, at least in part, in parallel. Microarchitects depend on circuit and physical design studies to understand the cost of proposed microarchitectural features. Engineers are sometimes categorized as “short and fat” or “tall and skinny” (nothing personal, we assure you!). Tall, skinny engineers understand something about a broad range of topics. Short, fat engineers understand a large amount about a narrow field. Digital VLSI design favors the tall, skinny engineer who can evaluate how choices in one part of the system impact other parts of the system.

1.6.2 Structured Design

Hierarchy is a critical tool for managing complex designs. A large system can be partitioned hierarchically into multiple *cores*. Each core is built from various *units*. Each unit in turn is composed of multiple *functional blocks*.⁵ These blocks in turn are built from *cells*, which ultimately are constructed from transistors. The system can be more easily understood at the top level by viewing components as black boxes with well-defined interfaces and functions rather than looking at each individual transistor. Logic, circuit, and physical views of the design should share the same hierarchy for ease of verification. A design hierarchy can be viewed as a tree structure with the overall chip as the *root* and the primitive cells as *leafs*.

Regularity aids the management of design complexity by designing the minimum number of different blocks. Once a block is designed and verified, it can be reused in many places. *Modularity* requires that the blocks have well-defined interfaces to avoid unanticipated interactions. *Locality* involves keeping information where it is used, physically and temporally. Structured design is discussed further in Section 14.2.

1.6.3 Behavioral, Structural, and Physical Domains

An alternative way of viewing design partitioning is shown with the Y-chart shown in Figure 1.48 [Gajski83, Kang03]. The radial lines on the Y-chart represent three distinct design domains: behavioral, structural, and physical. These domains can be used to describe the design of almost any artifact and thus form a general taxonomy for describing

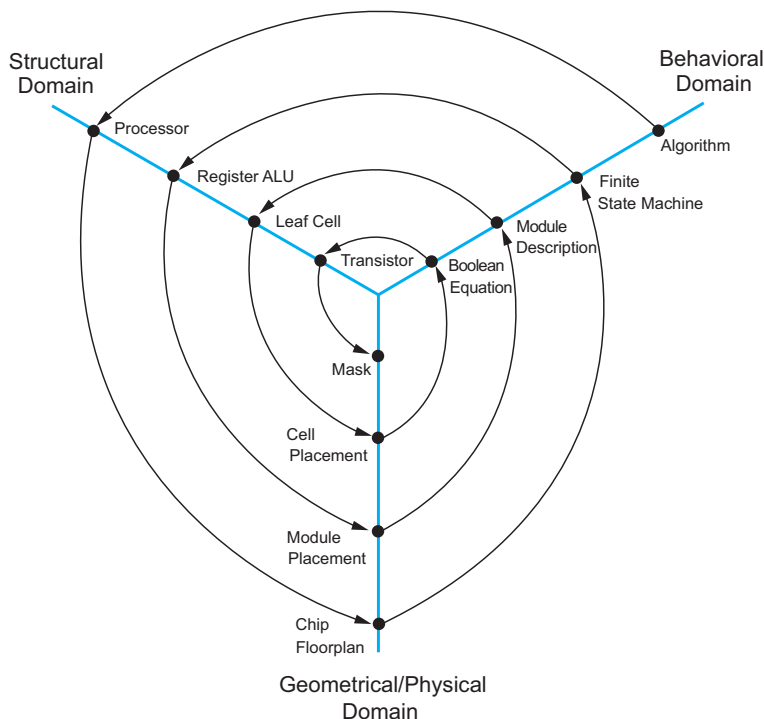


FIGURE 1.48 Y Diagram (Reproduced from [Kang03] with permission of The McGraw-Hill Companies.)

⁵Some designers refer to both units and functional blocks as *modules*.

the design process. Within each domain there are a number of levels of design abstraction that start at a very high level and descend eventually to the individual elements that need to be aggregated to yield the top level function (i.e., transistors in the case of chip design).

The behavioral domain describes what a particular system does. For instance, at the highest level we might specify a telephone touch-tone generator. This behavior can be successively refined to more precisely describe what needs to be done in order to build the tone generator (i.e., the frequencies desired, output levels, distortion allowed, etc.).

At each abstraction level, a corresponding structural description can be developed. The structural domain describes the interconnection of modules necessary to achieve a particular behavior. For instance, at the highest level, the touch-tone generator might consist of a keypad, a tone generator chip, an audio amplifier, a battery, and a speaker. Eventually at lower levels of abstraction, the individual gate and then transistor connections required to build the tone generator are described.

For each level of abstraction, the physical domain description explains how to physically construct that level of abstraction. At high levels, this might consist of an engineering drawing showing how to put together the keypad, tone generator chip, battery, and speaker in the associated housing. At the top chip level, this might consist of a floorplan, and at lower levels, the actual geometry of individual transistors.

The design process can be viewed as making transformations from one domain to another while maintaining the equivalency of the domains. Behavioral descriptions are transformed to structural descriptions, which in turn are transformed to physical descriptions. These transformations can be manual or automatic. In either case, it is normal design practice to verify the transformation of one domain to the other. This ensures that the design intent is carried across the domain boundaries. Hierarchically specifying each domain at successively detailed levels of abstraction allows us to design very large systems.

The reason for strictly describing the domains and levels of abstraction is to define a precise design process in which the final function of the system can be traced all the way back to the initial behavioral description. In an ideal flow, there should be no opportunity to produce an incorrect design. If anomalies arise, the design process is corrected so that those anomalies will not reoccur in the future. A designer should acquire a rigid discipline with respect to the design process, and be aware of each transformation and how and why it is failproof. Normally, these steps are fully automated in a modern design process, but it is important to be aware of the basis for these steps in order to debug them if they go astray.

The Y diagram can be used to illustrate each domain and the transformations between domains at varying levels of design abstraction. As the design process winds its way from the outer to inner rings, it proceeds from higher to lower levels of abstraction and hierarchy.

Most of the remainder of this chapter is a case study in the design of a simple microprocessor to illustrate the various aspects of VLSI design applied to a nontrivial system. We begin by describing the architecture and microarchitecture of the processor. We then consider logic design and discuss hardware description languages. The processor is built with static CMOS circuits, which we examined in Section 1.4; transistor-level design and netlist formats are discussed. We continue exploring the physical design of the processor including floorplanning and area estimation. Design verification is critically important and happens at each level of the hierarchy for each element of the design. Finally, the layout is converted into masks so the chip can be manufactured, packaged, and tested.

1.7 Example: A Simple MIPS Microprocessor

We consider an 8-bit subset of the MIPS microprocessor architecture [Patterson04, Harris07] because it is widely studied and is relatively simple, yet still large enough to illustrate hierarchical design. This section describes the architecture and the multicycle microarchitecture we will be implementing. If you are not familiar with computer architecture, you can regard the MIPS processor as a black box and skip to Section 1.8.

A set of laboratory exercises is available at www.cmosvlsi.com in which you can learn VLSI design by building the microprocessor yourself using a free open-source CAD tool called *Electric* or with commercial design tools from Cadence and Synopsys.

1.7.1 MIPS Architecture

The MIPS32 architecture is a simple 32-bit RISC architecture with relatively few idiosyncrasies. Our subset of the architecture uses 32-bit instruction encodings but only eight 8-bit general-purpose registers named \$0–\$7. We also use an 8-bit program counter (PC). Register \$0 is hardwired to contain the number 0. The instructions are ADD, SUB, AND, OR, SLT, ADDI, BEQ, J, LB, and SB.

The function and encoding of each instruction is given in Table 1.7. Each instruction is encoded using one of three templates: R, I, and J. R-type instructions (*register*-based) are used for arithmetic and specify two source registers and a destination register. I-type instructions are used when a 16-bit constant (also known as an *immediate*) and two registers must be specified. J-type instructions (*jumps*) dedicate most of the instruction word to a 26-bit jump destination. The format of each encoding is defined in Figure 1.49. The six most significant bits of all formats are the operation code (op). R-type instructions all share op = 000000 and use six more funct bits to differentiate the functions.

TABLE 1.7 MIPS instruction set (subset supported)

Instruction	Function	Encoding	op	funct
add \$1, \$2, \$3	addition: $\$1 \leftarrow \$2 + \$3$	R	000000	100000
sub \$1, \$2, \$3	subtraction: $\$1 \leftarrow \$2 - \$3$	R	000000	100010
and \$1, \$2, \$3	bitwise and: $\$1 \leftarrow \$2 \text{ and } \$3$	R	000000	100100
or \$1, \$2, \$3	bitwise or: $\$1 \leftarrow \$2 \text{ or } \$3$	R	000000	100101
slt \$1, \$2, \$3	set less than: $\$1 \leftarrow 1 \text{ if } \$2 < \$3$ $\$1 \leftarrow 0 \text{ otherwise}$	R	000000	101010
addi \$1, \$2, imm	add immediate: $\$1 \leftarrow \$2 + \text{imm}$	I	001000	n/a
beq \$1, \$2, imm	branch if equal: $\text{PC} \leftarrow \text{PC} + \text{imm} \times 4^a$	I	000100	n/a
j destination	jump: $\text{PC} \leftarrow \text{destination}^a$	J	000010	n/a
lb \$1, imm(\$2)	load byte: $\$1 \leftarrow \text{mem}[\$2 + \text{imm}]$	I	100000	n/a
sb \$1, imm(\$2)	store byte: $\text{mem}[\$2 + \text{imm}] \leftarrow \1	I	101000	n/a

a. Technically, MIPS addresses specify bytes. Instructions require a 4-byte word and must begin at addresses that are a multiple of four. To most effectively use instruction bits in the full 32-bit MIPS architecture, branch and jump constants are specified in words and must be multiplied by four (shifted left 2 bits) to be converted to byte addresses.

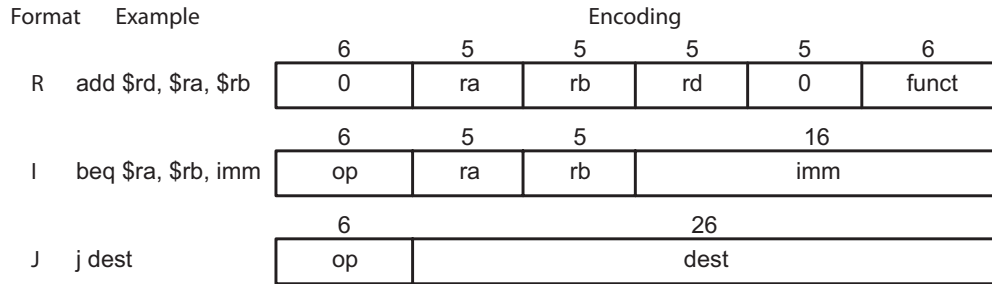


FIGURE 1.49 Instruction encoding formats

We can write programs for the MIPS processor in *assembly language*, where each line of the program contains one instruction such as ADD or BEQ. However, the MIPS hardware ultimately must read the program as a series of 32-bit numbers called *machine language*. An *assembler* automates the tedious process of translating from assembly language to machine language using the encodings defined in Table 1.7 and Figure 1.49. Writing nontrivial programs in assembly language is also tedious, so programmers usually work in a *high-level language* such as C or Java. A *compiler* translates a program from high-level language *source code* into the appropriate machine language *object code*.

Example 1.4

Figure 1.50 shows a simple C program that computes the n th Fibonacci number f_n defined recursively for $n > 0$ as $f_n = f_{n-1} + f_{n-2}$, $f_{-1} = -1$, $f_0 = 1$. Translate the program into MIPS assembly language and machine language.

SOLUTION: Figure 1.51 gives a commented assembly language program. Figure 1.52 translates the assembly language to machine language.

```
int fib(void)
{
    int n = 8;           /* compute nth Fibonacci number */
    int f1 = 1, f2 = -1; /* last two Fibonacci numbers */

    while (n != 0) {     /* count down to n = 0 */
        f1 = f1 + f2;
        f2 = f1 - f2;
        n = n - 1;
    }
    return f1;
}
```

FIGURE 1.50 C Code for Fibonacci program

1.7.2 Multicycle MIPS Microarchitecture

We will implement the multicycle MIPS microarchitecture given in Chapter 5 of [Patterson04] and Chapter 7 of [Harris07] modified to process 8-bit data. The microarchitecture is illustrated in Figure 1.53. Light lines indicate individual signals while heavy

```

# fib.asm
# Register usage: $3: n $4: f1 $5: f2
# return value written to address 255
fib:  addi $3, $0, 8      # initialize n=8
      addi $4, $0, 1      # initialize f1 = 1
      addi $5, $0, -1     # initialize f2 = -1
loop: beq $3, $0, end     # Done with loop if n = 0
      add $4, $4, $5      # f1 = f1 + f2
      sub $5, $4, $5      # f2 = f1 - f2
      addi $3, $3, -1     # n = n - 1
      j loop              # repeat until done
end:  sb $4, 255($0)      # store result in address 255

```

FIGURE 1.51 Assembly language code for Fibonacci program

Instruction	Binary Encoding	Hexadecimal Encoding
addi \$3, \$0, 8	001000 00000 00011 0000000000001000	20030008
addi \$4, \$0, 1	001000 00000 00100 0000000000000001	20040001
addi \$5, \$0, -1	001000 00000 00101 1111111111111111	2005ffff
beq \$3, \$0, end	000100 00011 00000 00000000000000100	10600004
add \$4, \$4, \$5	000000 00100 00101 00100 00000 100000	00852020
sub \$5, \$4, \$5	000000 00100 00101 00101 00000 100010	00852822
addi \$3, \$3, -1	001000 00011 00011 1111111111111111	2063ffff
j loop	000010 00000000000000000000000000011	08000003
sb \$4, 255(\$0)	101000 00000 00100 0000000011111111	a00400ff

FIGURE 1.52 Machine language code for Fibonacci program

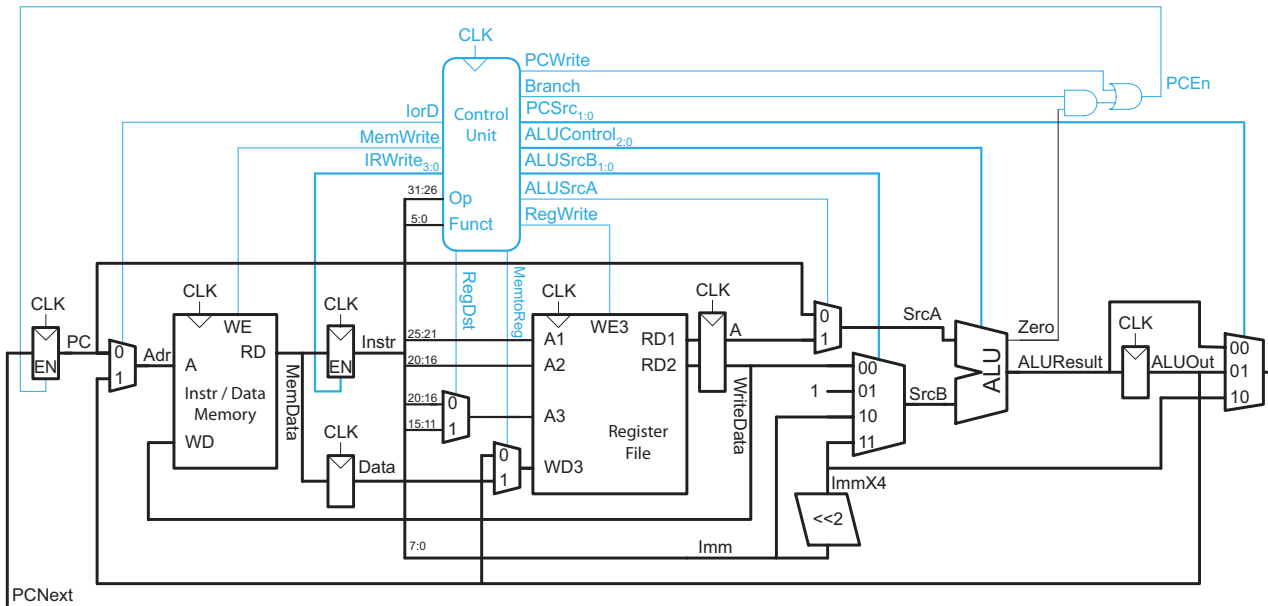


FIGURE 1.53 Multicycle MIPS microarchitecture. Adapted from [Patterson04] and [Harris07] with permission from Elsevier.

lines indicate busses. The control logic and signals are highlighted in blue while the datapath is shown in black. Control signals generally drive multiplexer select signals and register enables to tell the datapath how to execute an instruction.

Instruction execution generally flows from left to right. The program counter (PC) specifies the address of the instruction. The instruction is loaded 1 byte at a time over four cycles from an off-chip memory into the 32-bit instruction register (IR). The Op field (bits 31:26 of the instruction) is sent to the controller, which sequences the datapath through the correct operations to execute the instruction. For example, in an ADD instruction, the two source registers are read from the register file into temporary registers A and B. On the next cycle, the aludec unit commands the Arithmetic/Logic Unit (ALU) to add the inputs. The result is captured in the ALUOut register. On the third cycle, the result is written back to the appropriate destination register in the register file.

The controller contains a finite state machine (FSM) that generates multiplexer select signals and register enables to sequence the datapath. A state transition diagram for the FSM is shown in Figure 1.54. As discussed, the first four states fetch the instruction from

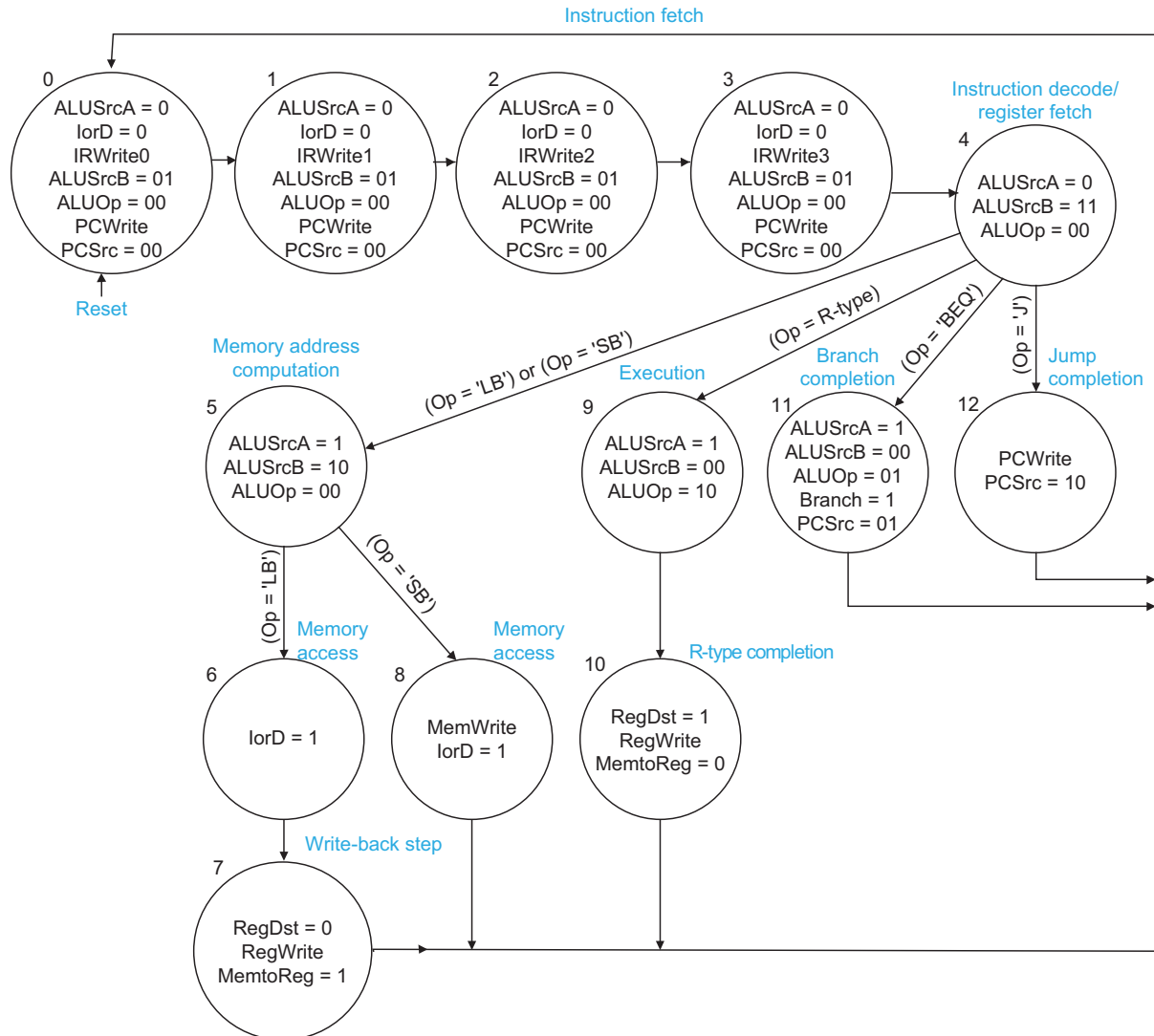


FIGURE 1.54 Multicycle MIPS control FSM (Adapted from [Patterson04] and [Harris07] with permission from Elsevier.)

memory. The FSM then is dispatched based on `Op` to execute the particular instruction. The FSM states for `ADDI` are missing and left as an exercise for the reader.

Observe that the FSM produces a 2-bit `ALUOp` output. The ALU decoder unit in the controller uses combinational logic to compute a 3-bit `ALUControl` signal from the `ALUOp` and `Funct` fields, as specified in Table 1.8. `ALUControl` drives multiplexers in the ALU to select the appropriate computation.

TABLE 1.8 ALUControl determination

ALUOp	Funct	ALUControl	Meaning
00	x	010	ADD
01	x	110	SUB
10	100000	010	ADD
10	100010	110	SUB
10	100100	000	AND
10	100101	001	OR
10	101010	111	SLT
11	x	x	undefined

Example 1.5

Referring to Figures 1.53 and 1.54, explain how the MIPS processor fetches and executes the `SUB` instruction.

SOLUTION: The first step is to fetch the 32-bit instruction. This takes four cycles because the instruction must come over an 8-bit memory interface. On each cycle, we want to fetch a byte from the address in memory specified by the program counter, then increment the program counter by one to point to the next byte.

The fetch is performed by states 0–3 of the FSM in Figure 1.54. Let us start with state 0. The program counter (PC) contains the address of the first byte of the instruction. The controller must select `IoRD` = 0 so that the multiplexer sends this address to the memory. `MemRead` must also be asserted so the memory reads the byte onto the `MemData` bus. Finally, `IRWrite0` should be asserted to enable writing `memdata` into the least significant byte of the instruction register (IR).

Meanwhile, we need to increment the program counter. We can do this with the ALU by specifying PC as one input, 1 as the other input, and `ADD` as the operation. To select PC as the first input, `ALUSrcA` = 0. To select 1 as the other input, `ALUSrcB` = 01. To perform an addition, `ALUOp` = 00, according to Table 1.8. To write this result back into the program counter at the end of the cycle, `PCSrc` = 00 and `PCEn` = 1 (done by setting `PCWrite` = 1).

All of these control signals are indicated in state 0 of Figure 1.54. The other register enables are assumed to be 0 if not explicitly asserted and the other multiplexer selects are don't cares. The next three states are identical except that they write bytes 1, 2, and 3 of the IR, respectively.

The next step is to read the source registers, done in state 4. The two source registers are specified in bits 25:21 and 20:16 of the IR. The register file reads these registers and puts the values into the A and B registers. No control signals are necessary for `SUB` (although state 4 performs a branch address computation in case the instruction is `BEQ`).

The next step is to perform the subtraction. Based on the `Op` field (IR bits 31:26), the FSM jumps to state 9 because `SUB` is an R-type instruction. The two source registers are selected as input to the ALU by setting `ALUSrcA = 1` and `ALUSrcB = 00`. Choosing `ALUOp = 10` directs the ALU Control decoder to select the `ALUControl` signal as 110, subtraction. Other R-type instructions are executed identically except that the decoder receives a different `Func` code (IR bits 5:0) and thus generates a different `ALUControl` signal. The result is placed in the `ALUOut` register.

Finally, the result must be written back to the register file in state 10. The data comes from the `ALUOut` register so `MemtoReg = 0`. The destination register is specified in bits 15:11 of the instruction so `RegDst = 1`. `RegWrite` must be asserted to perform the write. Then, the control FSM returns to state 0 to fetch the next instruction.

1.8 Logic Design

We begin the logic design by defining the top-level chip interface and block diagram. We then hierarchically decompose the units until we reach leaf cells. We specify the logic with a Hardware Description Language (HDL), which provides a higher level of abstraction than schematics or layout. This code is often called the Register Transfer Level (RTL) description.

1.8.1 Top-Level Interfaces

The top-level inputs and outputs are listed in Table 1.9. This example uses a two-phase clocking system to avoid hold-time problems. `Reset` initializes the PC to 0 and the control FSM to the start state.

TABLE 1.9 Top-level inputs and outputs

Inputs	Outputs
<code>ph1</code>	<code>MemWrite</code>
<code>ph2</code>	<code>Adr[7:0]</code>
<code>reset</code>	<code>WriteData[7:0]</code>
<code>MemData[7:0]</code>	

The remainder of the signals are used for an 8-bit memory interface (assuming the memory is located off chip). The processor sends an 8-bit address `Adr` and optionally asserts `MemWrite`. On a read cycle, the memory returns a value on the `MemData` lines while on a write cycle, the memory accepts input from `WriteData`. In many systems, `MemData` and `WriteData` can be combined onto a single bidirectional bus, but for this example we preserve the interface of Figure 1.53. Figure 1.55 shows a simple computer system built from the MIPS processor, external memory, reset switch, and clock generator.

1.8.2 Block Diagrams

The chip is partitioned into two top-level units: the controller and datapath, as shown in the block diagram in Figure 1.56. The controller comprises the control FSM, the ALU decoder, and the two gates used to compute `PCEn`. The ALU decoder consists of combina-

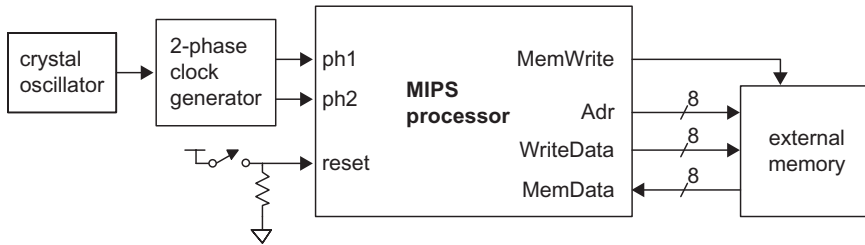


FIGURE 1.55 MIPS computer system

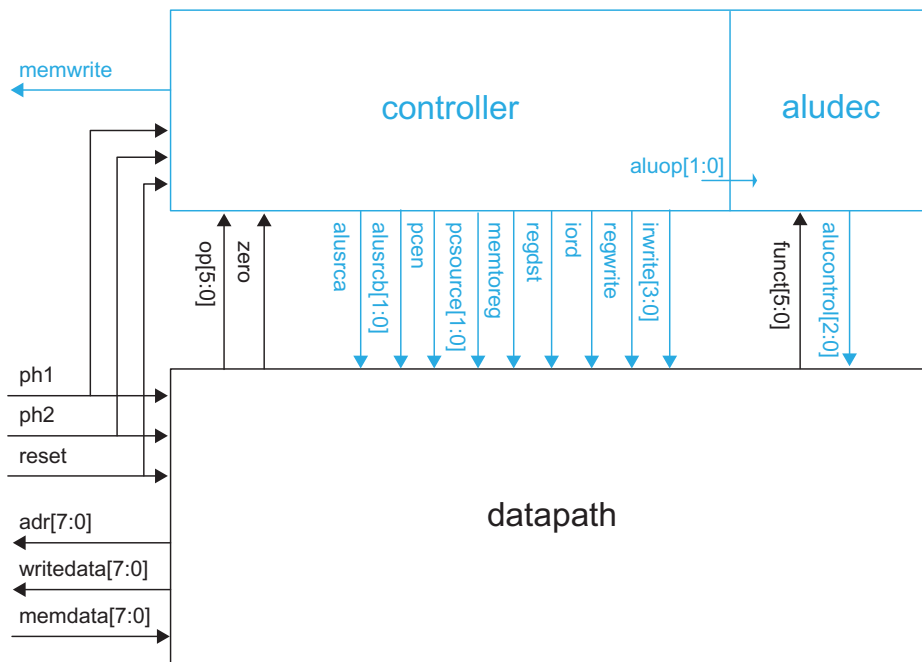


FIGURE 1.56 Top-level MIPS block diagram

tional logic to determine `ALUControl`. The 8-bit datapath contains the remainder of the chip. It can be viewed as a collection of wordslices or bitslices. A *wordslice* is a column containing an 8-bit flip-flop, adder, multiplexer, or other element. For example, Figure 1.57 shows a wordslice for an 8-bit 2:1 multiplexer. It contains eight individual 2:1 multiplexers, along with a *zipper* containing a buffer and inverter to drive the true and complementary select signals to all eight multiplexers.⁶ Factoring these drivers out into the zipper saves space as compared to putting inverters in each multiplexer. Alternatively, the datapath can be viewed as eight rows of *bitslices*. Each bitslice has one bit of each component, along with the horizontal wires connecting the bits together.

The chip partitioning is influenced by the intended physical design. The datapath contains most of the transistors and is very regular in structure. We can achieve high density with moderate design effort by handcrafting each wordslice or bitslice and tiling the

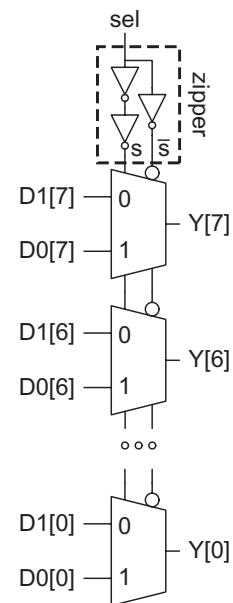


FIGURE 1.57 8-bit 2:1 multiplexer wordslice

⁶In this example, the zipper is shown at the top of the wordslice. In wider datapaths, the zipper is sometimes placed in the middle of the wordslice so that it drives shorter wires. The name comes from the way the layout resembles a plaid sweatshirt with a zipper down the middle.

circuits together. Building datapaths using wordslices is usually easier because certain structures, such as the zero detection circuit in the ALU, are not identical in each bitslice. However, thinking about bitslices is a valuable way to plan the wiring across the datapath. The controller has much less structure. It is tedious to translate an FSM into gates by hand, and in a new design, the controller is the most likely portion to have bugs and last-minute changes. Therefore, we will specify the controller more abstractly with a hardware description language and automatically generate it using synthesis and place & route tools or a programmable logic array (PLA).

1.8.3 Hierarchy

The best way to design complex systems is to decompose them into simpler pieces. Figure 1.58 shows part of the design hierarchy for the MIPS processor. The controller contains the controller_pla and aludec, which in turn is built from a library of standard cells such as NANDs, NORs, and inverters. The datapath is composed of 8-bit wordslices, each of which also is typically built from standard cells such as adders, register file bits, multiplexers, and flip-flops. Some of these cells are reused in multiple places.

The design hierarchy does not necessarily have to be identical in the logic, circuit, and physical designs. For example, in the logic view, a memory may be best treated as a black box, while in the circuit implementation, it may have a decoder, cell array, column multiplexers, and so forth. Different hierarchies complicate verification, however, because they must be *flattened* until the point that they agree. As a matter of practice, it is best to make logic, circuit, and physical design hierarchies agree as far as possible.

1.8.4 Hardware Description Languages

Designers need rapid feedback on whether a logic design is reasonable. Translating block diagrams and FSM state transition diagrams into circuit schematics is time-consuming and prone to error; before going through this entire process it is wise to know if the top-level design has major bugs that will require complete redesign. HDLs provide a way to specify the design at a higher level of abstraction to raise designer productivity. They were originally intended for documentation and simulation, but are now used to synthesize gates directly from the HDL.

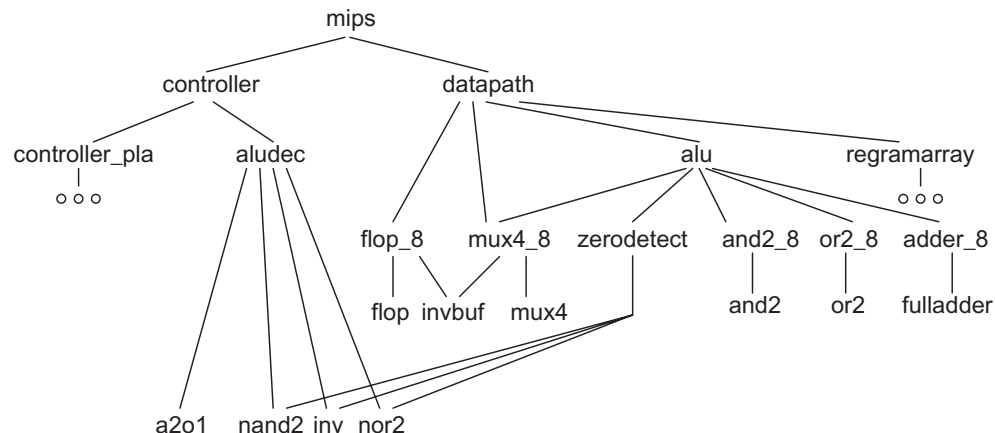


FIGURE 1.58 MIPS design hierarchy

The two most popular HDLs are *Verilog* and *VHDL*. Verilog was developed by Advanced Integrated Design Systems (later renamed Gateway Design Automation) in 1984 and became a *de facto* industry open standard by 1991. In 2005, the SystemVerilog extensions were standardized, and some of these features are used in this book. VHDL, which stands for VHSIC Hardware Description Language, where VHSIC in turn was a Department of Defense project on Very High Speed Integrated Circuits, was developed by committee under government sponsorship. As one might expect from their pedigrees, Verilog is less verbose and closer in syntax to C, while VHDL supports some abstractions useful for large team projects. Many Silicon Valley companies use Verilog while defense and telecommunications companies often use VHDL. Neither language offers a decisive advantage over the other so the industry is saddled with supporting both. Appendix A offers side-by-side tutorials on Verilog and VHDL. Examples in this book are given in Verilog for the sake of brevity.

When coding in an HDL, it is important to remember that you are specifying hardware that operates in parallel rather than software that executes in sequence. There are two general coding styles. *Structural* HDL specifies how a cell is composed of other cells or primitive gates and transistors. *Behavioral* HDL specifies what a cell does.

A *logic simulator* simulates HDL code; it can report whether results match expectations, and can display waveforms to help debug discrepancies. A *logic synthesis* tool is similar to a compiler for hardware: it maps HDL code onto a *library* of gates called *standard cells* to minimize area while meeting some timing constraints. Only a subset of HDL constructs are synthesizable; this subset is emphasized in the appendix. For example, file I/O commands used in testbenches are obviously not synthesizable. Logic synthesis generally produces circuits that are neither as dense nor as fast as those handcrafted by a skilled designer. Nevertheless, integrated circuit processes are now so advanced that synthesized circuits are good enough for the great majority of application-specific integrated circuits (ASICs) built today. Layout may be automatically generated using place & route tools.

Verilog and VHDL models for the MIPS processor are listed in Appendix A.12. In Verilog, each cell is called a *module*. The inputs and outputs are declared much as in a C program and bit widths are given for busses. Internal signals must also be declared in a way analogous to local variables. The processor is described hierarchically using structural Verilog at the upper levels and behavioral Verilog for the leaf cells. For example, the controller module shows how a finite state machine is specified in behavioral Verilog and the *aludec* module shows how complex combinational logic is specified. The datapath is specified structurally in terms of wordslices, which are in turn described behaviorally.

For the sake of illustration, the 8-bit adder wordslice could be described structurally as a ripple carry adder composed of eight cascaded full adders. The full adder could be expressed structurally as a sum and a carry subcircuit. In turn, the sum and carry subcircuits could be expressed behaviorally. The full adder block is shown in Figure 1.59 while the carry subcircuit is explored further in Section 1.9.

```

module adder(input  logic [7:0] a, b,
             input  logic    c,
             output logic [7:0] s,
             output logic    cout);

    wire [6:0] carry;

```

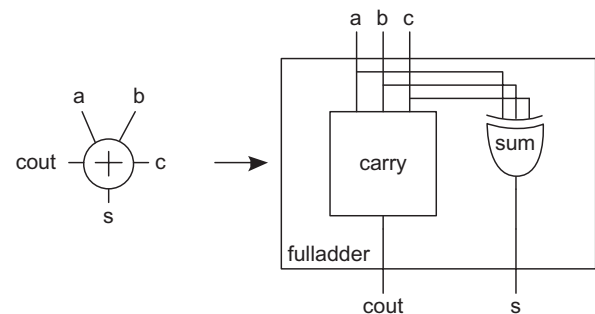


FIGURE 1.59 Full adder

```

fulladder fa0(a[0], b[0], c,          s[0], carry[0]);
fulladder fa1(a[1], b[1], carry[0], s[1], carry[1]);
fulladder fa2(a[2], b[2], carry[1], s[2], carry[2]);
...
fulladder fa7(a[7], b[7], carry[6], s[7], cout);
endmodule

module fulladder(input  logic a, b, c,
                  output logic s, cout);

    sum s1(a, b, c, s);
    carry c1(a, b, c, cout);
endmodule

module carry(input  logic a, b, c,
             output logic cout);

    assign cout = (a&b) | (a&c) | (b&c);
endmodule

```

1.9 Circuit Design

Circuit design is concerned with arranging transistors to perform a particular logic function. Given a circuit design, we can estimate the delay and power. The circuit can be represented as a schematic, or in textual form as a netlist. Common transistor level netlist formats include Verilog and SPICE. Verilog netlists are used for functional verification, while SPICE netlists have more detail necessary for delay and power simulations.

Because a transistor gate is a good insulator, it can be modeled as a capacitor, C . When the transistor is ON, some current I flows between source and drain. Both the current and capacitance are proportional to the transistor width.

The delay of a logic gate is determined by the current that it can deliver and the capacitance that it is driving, as shown in Figure 1.60 for one inverter driving another inverter. The capacitance is charged or discharged according to the constitutive equation

$$I = C \frac{dV}{dt}$$

If an average current I is applied, the time t to switch between 0 and V_{DD} is

$$t = \frac{C}{I} V_{DD}$$

Hence, the delay increases with the load capacitance and decreases with the drive current. To make these calculations, we will have to delve below the switch-level model of a transistor. Chapter 2 develops more detailed models of transistors accounting for the current and capacitance. One of the goals of circuit design is to choose transistor widths to meet delay requirements. Methods for doing so are discussed in Chapter 4.

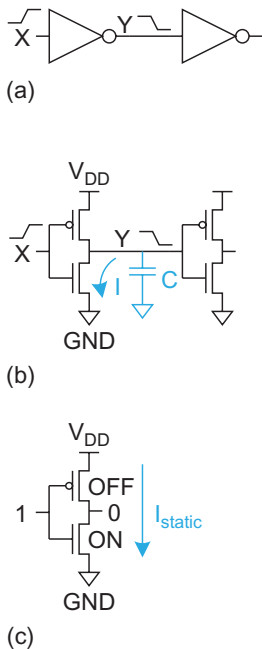


FIGURE 1.60 Circuit delay and power: (a) inverter pair, (b) transistor-level model showing capacitance and current during switching, (c) static leakage current during quiescent operation

Energy is required to charge and discharge the load capacitance. This is called dynamic power because it is consumed when the circuit is actively switching. The dynamic power consumed when a capacitor is charged and discharged at a frequency f is

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

Even when the gate is not switching, it draws some static power. Because an OFF transistor is leaky, a small amount of current I_{static} flows between power and ground, resulting in a static power dissipation of

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

Chapter 5 examines power in more detail.

A particular logic function can be implemented in many ways. Should the function be built with ANDs, ORs, NANDs, or NORs? What should be the fan-in and fan-out of each gate? How wide should the transistors be on each gate? Each of these choices influences the capacitance and current and hence the speed and power of the circuit, as well as the area and cost.

As mentioned earlier, in many design methodologies, logic synthesis tools automatically make these choices, searching through the standard cells for the best implementation. For many applications, synthesis is good enough. When a system has critical requirements of high speed or low power or will be manufactured in large enough volume to justify the extra engineering, custom circuit design becomes important for critical portions of the chip.

Circuit designers often draw schematics at the transistor and/or gate level. For example, Figure 1.61 shows two alternative circuit designs for the carry circuit in a full adder. The gate-level design in Figure 1.61(a) requires 26 transistors and four stages of gate delays (recall that ANDs and ORs are built from NANDs and NORs followed by inverters). The transistor-level design in Figure 1.61(b) requires only 12 transistors and two stages of gate delays, illustrating the benefits of optimizing circuit designs to take advantage of CMOS technology.

These schematics are then *netlisted* for simulation and verification. One common netlist format is structural Verilog HDL. The gate-level design can be netlisted as follows:

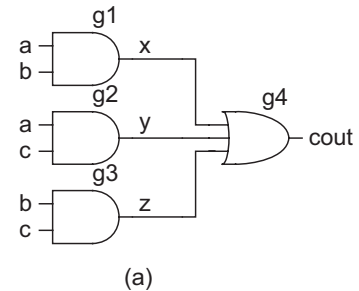
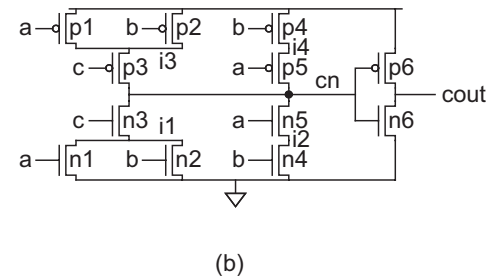


FIGURE 1.61 Carry subcircuit

```
module carry(input logic a, b, c,
            output logic cout);
```

```
    logic x, y, z;
```

```
    and g1(x, a, b);
```

```
    and g2(y, a, c);
```

```
    and g3(z, b, c);
```

```
    or g4(cout, x, y, z);
```

```
endmodule
```

This is a technology-independent structural description, because generic gates have been used and the actual gate implementations have not been specified. The transistor-level netlist follows:

```

module carry(input  logic a, b, c,
              output tri  cout);

    tri      i1, i2, i3, i4, cn;
    supply0 gnd;
    supply1 vdd;

    tranif1 n1(i1, gnd, a);
    tranif1 n2(i1, gnd, b);
    tranif1 n3(cn, i1, c);
    tranif1 n4(i2, gnd, b);
    tranif1 n5(cn, i2, a);
    tranif0 p1(i3, vdd, a);
    tranif0 p2(i3, vdd, b);
    tranif0 p3(cn, i3, c);
    tranif0 p4(i4, vdd, b);
    tranif0 p5(cn, i4, a);
    tranif1 n6(cout, gnd, cn);
    tranif0 p6(cout, vdd, cn);
endmodule

```

Transistors are expressed as

```

Transistor-type name(drain, source, gate);

```

tranif1 corresponds to nMOS transistors that turn ON when the gate is 1 while **tranif0** corresponds to pMOS transistors that turn ON when the gate is 0. Appendix A.11 covers Verilog netlists in more detail.

With the description generated so far, we still do not have the information required to determine the speed or power consumption of the gate. We need to specify the size of the transistors and the stray capacitance. Because Verilog was designed as a switch-level and gate-level language, it is poorly suited to structural descriptions at this level of detail. Hence, we turn to another common structural language used by the circuit simulator SPICE. The specification of the transistor-level carry subcircuit at the circuit level might be represented as follows:

```

.SUBCKT CARRY A B C COUT VDD GND
MN1 I1 A GND GND NMOS W=2U L=0.6U AD=1.8P AS=3P
MN2 I1 B GND GND NMOS W=2U L=0.6U AD=1.8P AS=3P
MN3 CN C I1 GND NMOS W=2U L=0.6U AD=3P AS=3P
MN4 I2 B GND GND NMOS W=2U L=0.6U AD=0.9P AS=3P
MN5 CN A I2 GND NMOS W=2U L=0.6U AD=3P AS=0.9P
MP1 I3 A VDD VDD PMOS W=4U L=0.6U AD=3.6P AS=6P
MP2 I3 B VDD VDD PMOS W=4U L=0.6U AD=3.6P AS=6P
MP3 CN C I3 VDD PMOS W=4U L=0.6U AD=6P AS=6P

```

```

MP4 I4 B VDD VDD PMOS W=4U L=0.6U AD=1.8P AS=6P
MP5 CN A I4 VDD PMOS W=4U L=0.6U AD=6P AS=1.8P
MN6 COUT CN GND GND NMOS W=4U L=0.6U AD=6P AS=6P
MP6 COUT CN VDD VDD PMOS W=8U L=0.6U AD=12P AS=12P
CI1 I1 GND 6FF
CI3 I3 GND 9FF
CA A GND 12FF
CB B GND 12FF
CC C GND 6FF
CCN CN GND 12FF
CCOUT COUT GND 6FF
.ENDS

```

Transistors are specified by lines beginning with an M as follows:

```

Mname  drain  gate  source  body  type  W=width  L=length
      AD=drain area  AS=source area

```

Although MOS switches have been masquerading as three terminal devices (gate, source, and drain) until this point, they are in fact four terminal devices with the substrate or well forming the *body* terminal. The body connection was not listed in Verilog but is required for SPICE. The type specifies whether the transistor is a p-device or n-device. The width, length, and area parameters specify physical dimensions of the actual transistors. Units include U (micro, 10^{-6}), P (pico, 10^{-12}), and F (femto, 10^{-15}). Capacitors are specified by lines beginning with C as follows:

```

Cname  node1  node2  value

```

In this description, the MOS model in SPICE calculates the parasitic capacitances inherent in the MOS transistor using the device dimensions specified. The extra capacitance statements in the above description designate additional routing capacitance not inherent to the device structure. This depends on the physical design of the gate. Long wires also contribute resistance, which increases delay. At the circuit level of structural specification, all connections are given that are necessary to fully characterize the carry gate in terms of speed, power, and connectivity. Chapter 8 describes SPICE models in more detail.

1.10 Physical Design

1.10.1 Floorplanning

Physical design begins with a floorplan. The floorplan estimates the area of major units in the chip and defines their relative placements. The floorplan is essential to determine whether a proposed design will fit in the chip area budgeted and to estimate wiring lengths and wiring congestion. An initial floorplan should be prepared as soon as the logic is loosely defined. As usual, this process involves feedback. The floorplan will often suggest changes to the logic (and microarchitecture), which in turn changes the floorplan. For example, suppose microarchitects assume that a cache requires a 2-cycle access latency. If the floorplan shows that the data cache can be placed adjacent to the execution units in the

datapath, the cache access time might reduce to a single cycle. This could allow the microarchitects to reduce the cache capacity while providing the same performance. Once the cache shrinks, the floorplan must be reconsidered to take advantage of the newly available space near the datapath. As a complex design begins to stabilize, the floorplan is often hierarchically subdivided to describe the functional blocks within the units.

The challenge of floorplanning is estimating the size of each unit without proceeding through a detailed design of the chip (which would depend on the floorplan and wire lengths). This section assumes that good estimates have been made and describes what a floorplan looks like. The next sections describe each of the types of components that might be in a floorplan and suggests ways to estimate the component sizes.

Figure 1.62 shows the chip floorplan for the MIPS processor including the pad frame. The top-level blocks are the controller and datapath. A wiring channel is located between the two blocks to provide room to route control signals to the datapath. The datapath is further partitioned into wordslices. The *pad frame* includes 40 I/O pads, which are wired to the pins on the chip package. There are 29 pads used for signals; the remainder are V_{DD} and GND.

The floorplan is drawn to scale and annotated with dimensions. The chip is designed in a $0.6\text{ }\mu\text{m}$ process on a $1.5 \times 1.5\text{ mm}$ die so the die is 5000λ on a side. Each pad is $750\lambda \times$

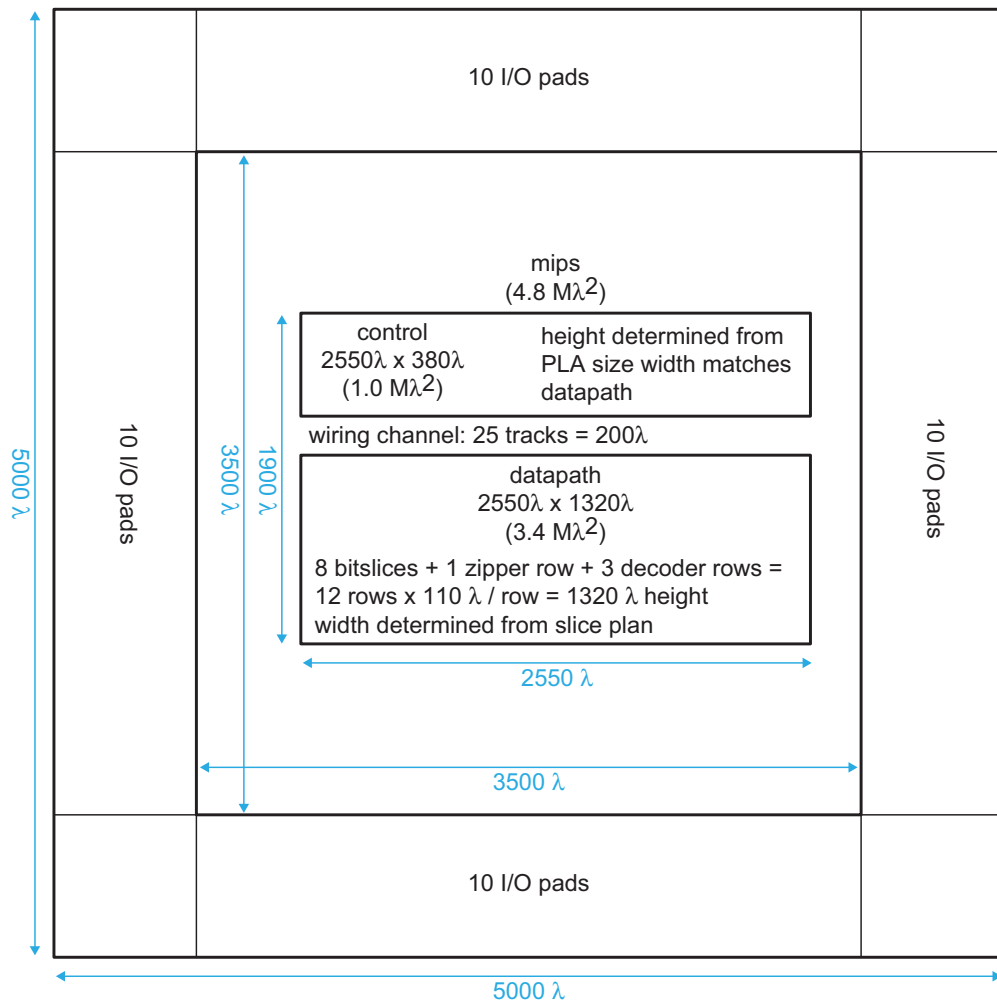


FIGURE 1.62 MIPS floorplan

350λ , so the maximum possible core area inside the pad frame is $3500\lambda \times 3500\lambda = 12.25\text{ M}\lambda^2$. Due to the wiring channel, the actual core area of $4.8\text{ M}\lambda^2$ is larger than the sum of the block areas. This design is said to be *pad-limited* because the I/O pads set the chip area. Most commercial chips are *core-limited* because the chip area is set by the logic excluding the pads. In general, blocks in a floorplan should be rectangular because it is difficult for a designer to stuff logic into an odd-shaped region (although some CAD tools do so just fine).

Figure 1.63 shows the actual chip layout. Notice the 40 I/O pads around the periphery. Just inside the pad frame are metal2 V_{DD} and GND rings, marked with + and -.

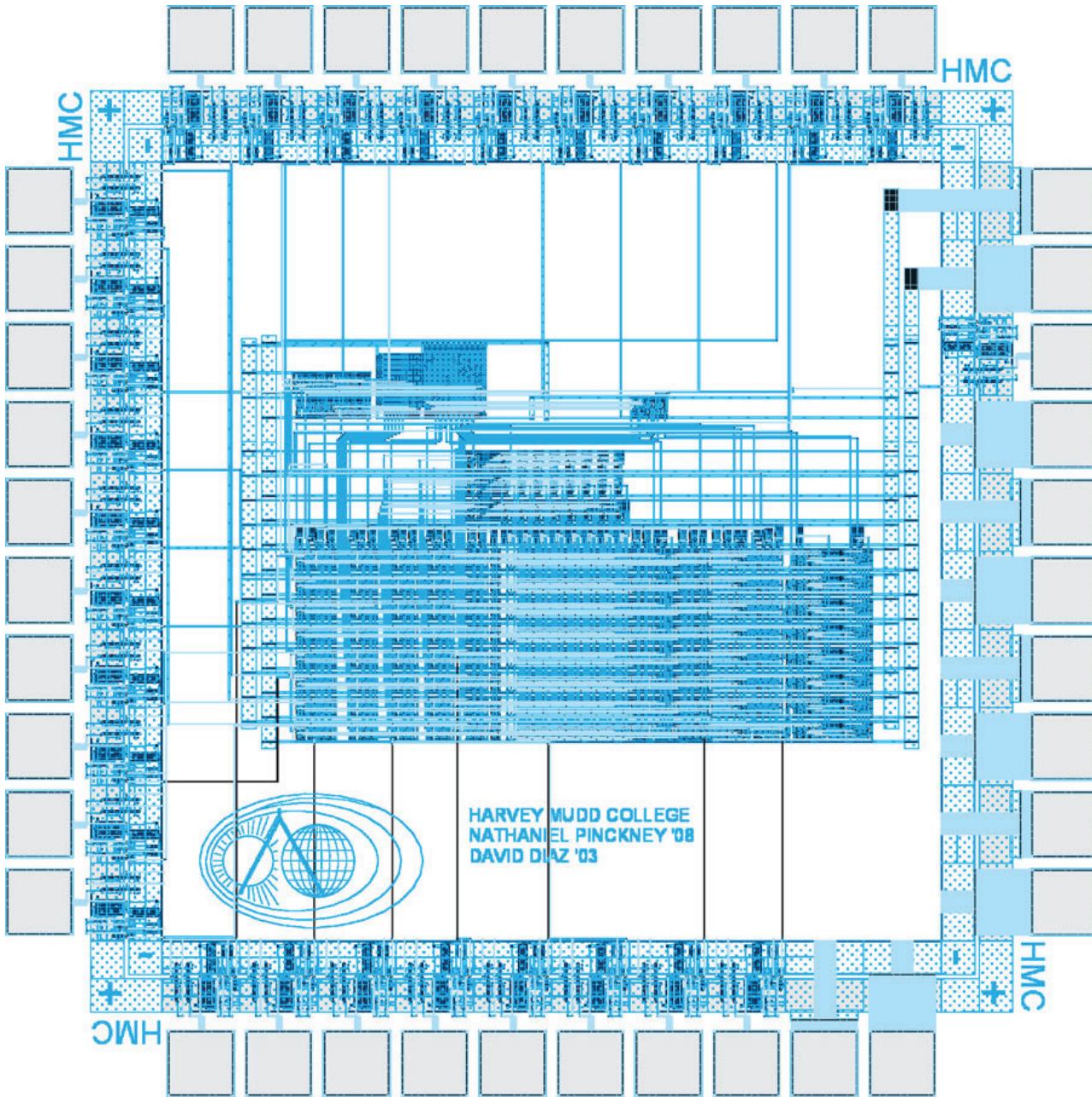


FIGURE 1.63 MIPS layout

On-chip structures can be categorized as *random logic*, *datapaths*, *arrays*, *analog*, and *input/output* (I/O). Random logic, like the aludecoder, has little structure. Datapaths operate on multi-bit data words and perform roughly the same function on each bit so they consist of multiple N -bit wordslices. Arrays, like RAMs, ROMs, and PLAs, consist of identical cells repeated in two dimensions. Productivity is highest if layout can be reused or automatically generated. Datapaths and arrays are good VLSI building blocks because a single carefully crafted cell is reused in one or two dimensions. Automatic layout generators exist for memory arrays and random logic but are not as mature for datapaths. Therefore, many design methodologies ignore the potential structure of datapaths and instead lay them out with random logic tools except when performance or area are vital. Analog circuits still require careful design and simulation but tend to involve only small amounts of layout because they have relatively few transistors. I/O cells are also highly tuned to each fabrication process and are often supplied by the process vendor.

Random logic and datapaths are typically built from *standard cells* such as inverters, NAND gates, and flip-flops. Standard cells increase productivity because each cell only needs to be drawn and verified once. Often, a standard cell library is purchased from a third party vendor.

Another important decision during floorplanning is to choose the metal orientation. The MIPS floorplan uses horizontal metal1 wires, vertical metal2 wires, and horizontal metal3 wires. Alternating directions between each layer makes it easy to cross wires on different layers.

1.10.2 Standard Cells

A simple standard cell library is shown on the inside front cover. Power and ground run horizontally in metal1. These supply rails are 8λ wide (to carry more current) and are separated by 90λ center-to-center. The nMOS transistors are placed in the bottom 40λ of the cell and the pMOS transistors are placed in the top 50λ . Thus, cells can be connected by abutment with the supply rails and n-well matching up. Substrate and well contacts are placed under the supply rails. Inputs and outputs are provided in metal2, which runs vertically. Each cell is a multiple of 8λ in width so that it offers an integer number of metal2 tracks. Within the cell, poly is run vertically to form gates and diffusion and metal1 are run horizontally, though metal1 can also be run vertically to save space when it does not interfere with other connections.

Cells are tiled in rows. Each row is separated vertically by at least 110λ from the base of the previous row. In a 2-level metal process, horizontal metal1 wires are placed in *routing channels* between the rows. The number of wires that must be routed sets the height of the routing channels. Layout is often generated with automatic place & route tools. Figure 1.64 shows the controller layout generated by such a tool. Note that in this and subsequent layouts, the n-well around the pMOS transistors will usually not be shown.

When more layers of metal are available, routing takes place over the cells and routing channels may become unnecessary. For example, in a 3-level metal process, metal3 is run horizontally on a 10λ pitch. Thus, 11 horizontal tracks can run over each cell. If this is sufficient to accommodate all of the horizontal wires, the routing channels can be eliminated.

Automatic synthesis and place & route tools have become good enough to map entire designs onto standard cells. Figure 1.65 shows the entire 8-bit MIPS processor synthesized from the VHDL model given in Appendix A.12 onto a cell library in a 130 nm process with

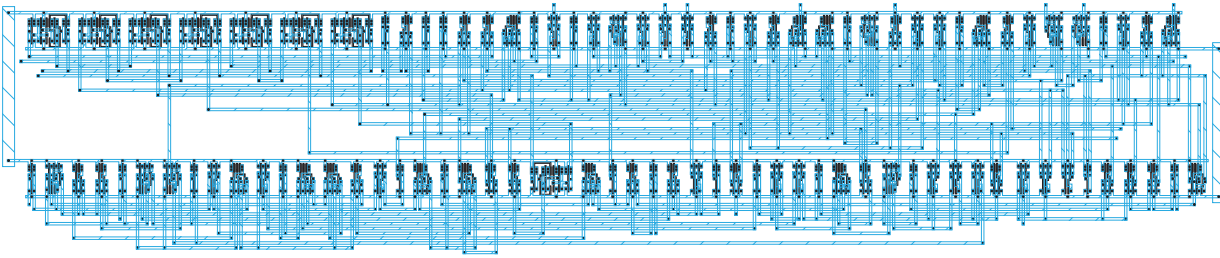


FIGURE 1.64 MIPS controller layout (synthesized)

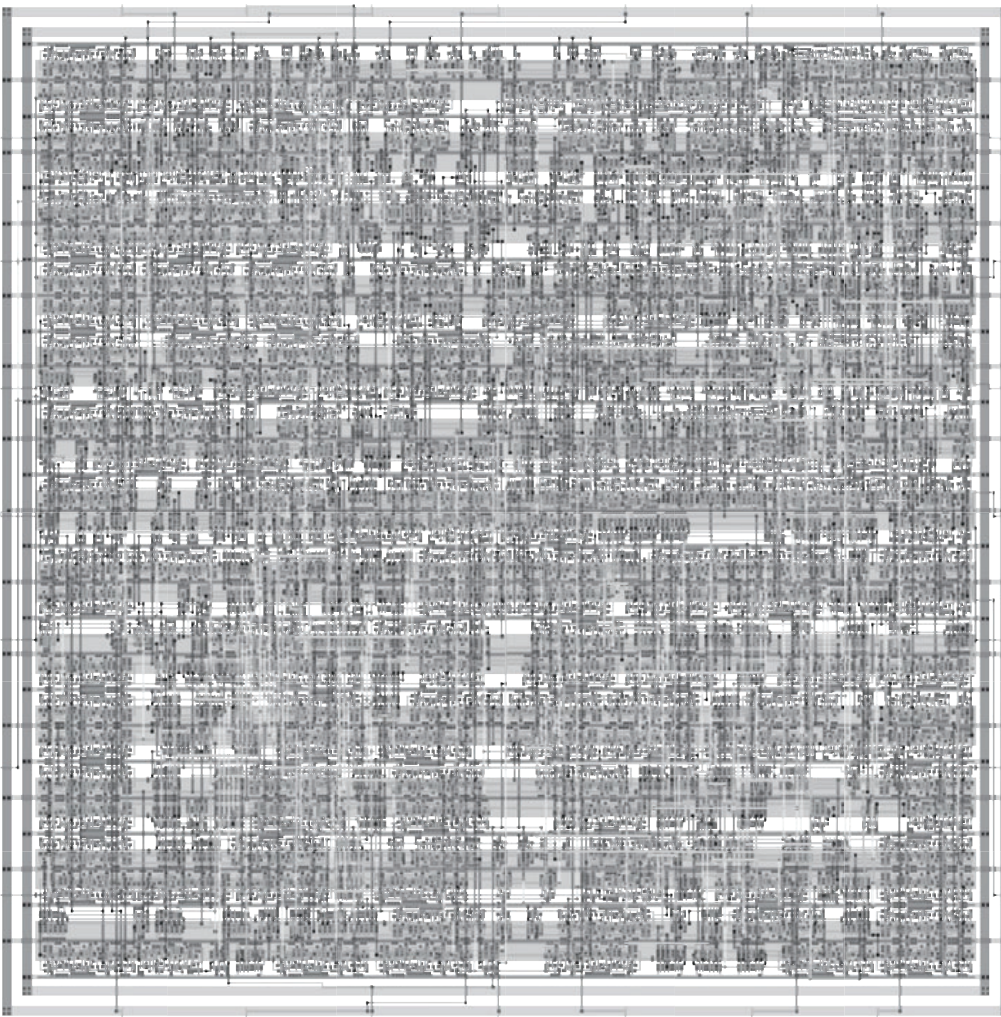


FIGURE 1.65 Synthesized MIPS processor

seven metal layers. Compared to Figure 1.63, the synthesized design shows little discernible structure except that 26 rows of standard cells can be identified beneath the wires. The area is approximately $4 \text{ M}\lambda^2$. Synthesized designs tend to be somewhat slower than a good custom design, but they also take an order of magnitude less design effort.

A	A	A	A	B
A	A	A	A	B
A	A	A	A	B
A	A	A	A	B
C	C	C	C	D

FIGURE 1.66 Pitch-matching of snap-together cells

1.10.3 Pitch Matching

The area of the controller in Figure 1.64 is dominated by the routing channels. When the logic is more regular, layout density can be improved by including the wires in cells that “snap together.” Snap-together cells require more design and layout effort but lead to smaller area and shorter (i.e., faster) wires. The key issue in designing snap-together cells is *pitch-matching*. Cells that connect must have the same size along the connecting edge. Figure 1.66 shows several pitch-matched cells. Reducing the size of cell *D* does not help the layout area. On the other hand, increasing the size of cell *D* also affects the area of *B* and/or *C*.

Figure 1.67 shows the MIPS datapath in more detail. The eight horizontal bitslices are clearly visible. The zipper at the top of the layout includes three rows for the decoder that is pitch-matched to the register file in the datapath. Vertical metal2 wires are used for control, including clocks, multiplexer selects, and register enables. Horizontal metal3 wires run over the tops of cells to carry data along a bitslice.

The width of the transistors in the cells and the number of wires that must run over the datapath determines the minimum height of the datapath cells. 60–100 λ are typical heights for relatively simple datapaths. The width of the cell depends on the cell contents.

1.10.4 Slice Plans

Figure 1.68 shows a *slice plan* of the datapath. The diagram illustrates the ordering of wordslices and the allocation of wiring tracks within each bitslice. Dots indicate that a bus passes over a cell and is also used in that cell. Each cell is annotated with its type and width (in number of tracks). For example, the program counter (pc) is an output of the PC flop and is also used as an input to the srcA and address multiplexers. The slice plan

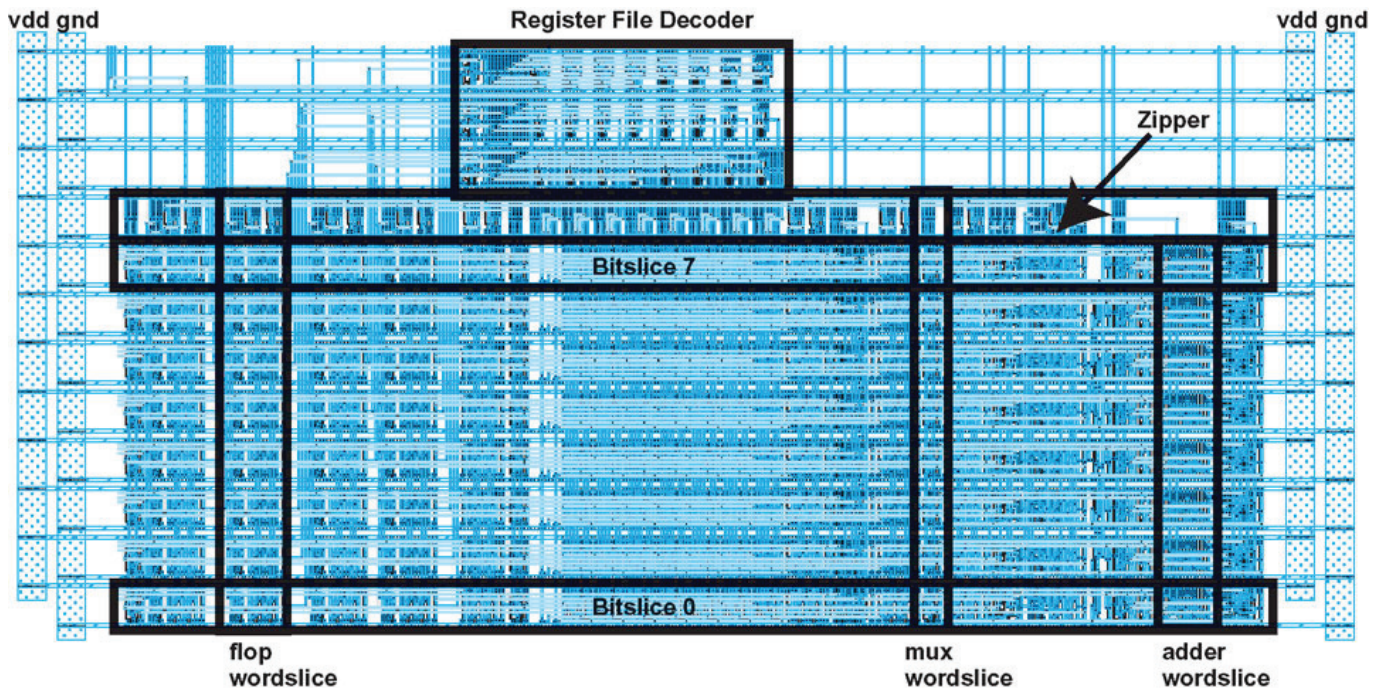


FIGURE 1.67 MIPS datapath layout

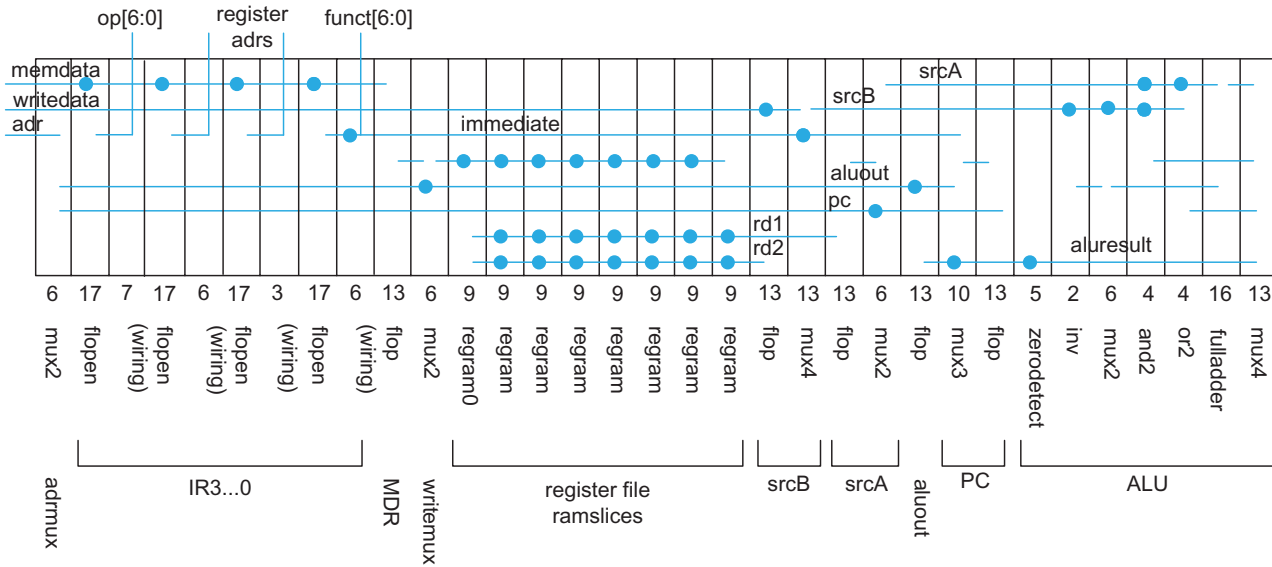


FIGURE 1.68 Datapath slice plan

makes it easy to calculate wire lengths and evaluate wiring congestion before laying out the datapath. In this case, it is evident that the greatest congestion takes place over the register file, where seven wiring tracks are required.

The slice plan is also critical for estimating area of datapaths. Each wordslice is annotated with its width, measured in tracks. This information can be obtained by looking at the cell library layouts. By adding up the widths of each element in the slice plan, we see that the datapath is 319 tracks wide, or 2552 λ wide. There are eight bitslices in the 8-bit datapath. In addition, there is one more row for the zipper and three more for the three register file address decoders, giving a total of 12 rows. At a pitch of 110 λ / row, the datapath is 1320 λ tall. The address decoders only occupy a small fraction of their rows, leaving wasted empty space. In a denser design, the controller could share some of the unused area.

1.10.5 Arrays

Figure 1.69 shows a programmable logic array (PLA) used for the control FSM next state and output logic. A PLA can compute any function expressed in sum of products form. The structure on the left is called the AND plane and the structure on the right is the OR plane. PLAs are discussed further in Section 12.7.

This PLA layout uses 2 vertical tracks for each input and 3 for each output plus about 6 for overhead. It uses 1.5 horizontal tracks for each product or minterm, plus about 14 for overhead. Hence, the size of a PLA is easy to calculate. The total PLA area is 500 $\lambda \times 350 \lambda$, plus another 336 $\lambda \times 220 \lambda$ for the four external flip-flops needed in the control FSM. The height of the controller is dictated by the height of the PLA plus a few wiring tracks to route inputs and outputs. In comparison, the synthesized controller from Figure 1.64 has a size of 1500 $\lambda \times 400 \lambda$ because the wiring tracks waste so much space.

1.10.6 Area Estimation

A good floorplan depends on reasonable area estimates, which may be difficult to make before logic is finalized. An experienced designer may be able to estimate block area by

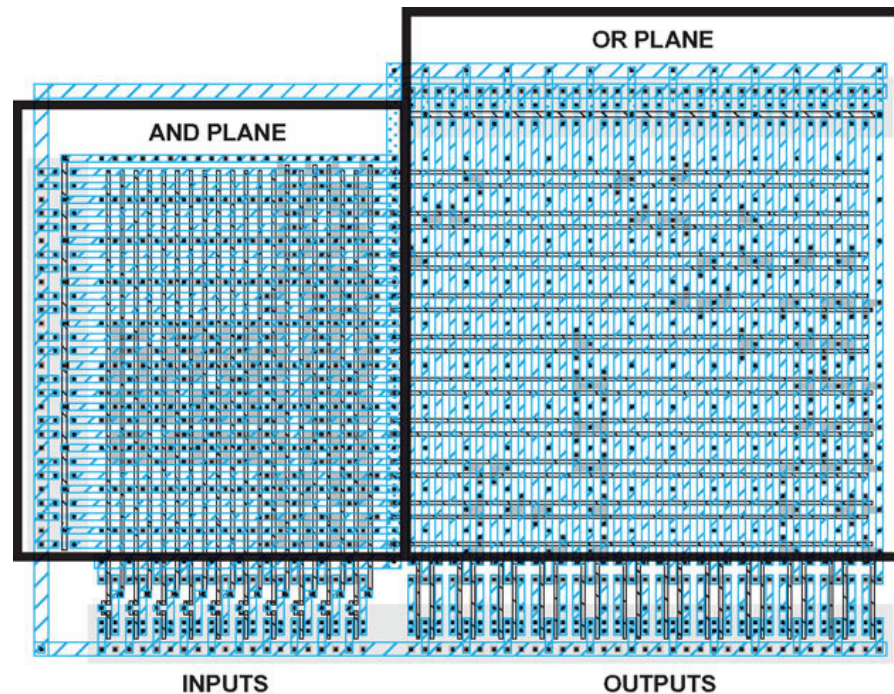


FIGURE 1.69 PLA for control FSM

comparison to the area of a comparable block drawn in the past. In the absence of data for such comparison, Table 1.10 lists some typical numbers. Be certain to account for large wiring channels at a pitch of 8λ / track. Larger transistors clearly occupy a greater area, so this may be factored into the area estimates as a function of W and L (width and length). For memories, don't forget about the decoders and other periphery circuits, which often take as much area as the memory bits themselves. Your mileage may vary, but datapaths and arrays typically achieve higher densities than standard cells.

TABLE 1.10 Typical layout densities

Element	Area
random logic (2-level metal process)	$1000 - 1500 \lambda^2$ / transistor
datapath	$250 - 750 \lambda^2$ / transistor or $6 WL + 360 \lambda^2$ / transistor
SRAM	$1000 \lambda^2$ / bit
DRAM (in a DRAM process)	$100 \lambda^2$ / bit
ROM	$100 \lambda^2$ / bit

Given enough time, it is nearly always possible to shave a few lambda here or there from a design. However, such efforts are seldom a good investment unless an element is repeated so often that it accounts for a major fraction of the chip area or if floorplan errors have led to too little space for a block and the block must be shrunk before the chip can be completed. It is wise to make conservative area estimates in floorplans, especially if there is risk that more functionality may be added to a block.

Some cell library vendors specify typical routed standard cell layout densities in kgates / mm².⁷ Commonly, a gate is defined as a 3-input static CMOS NAND or NOR with six transistors. A 65 nm process ($\lambda \approx 0.03 \mu\text{m}$) with eight metal layers may achieve a density of 160–500 kgates / mm² for random logic. This corresponds to about 370–1160 λ^2 / transistor. Processes with many metal layers obtain high density because routing channels are not needed.

1.11 Design Verification

Integrated circuits are complicated enough that if anything can go wrong, it probably will. Design verification is essential to catching the errors before manufacturing and commonly accounts for half or more of the effort devoted to a chip.

As design representations become more detailed, verification time increases. It is not practical to simulate an entire chip in a circuit-level simulator such as SPICE for a large number of cycles to prove that the layout is correct. Instead, the design is usually tested for functionality at the architectural level with a model in a language such as C and at the logic level by simulating the HDL description. Then, the circuits are checked to ensure that they are a faithful representation of the logic and the layout is checked to ensure it is a faithful representation of the circuits, as shown in Figure 1.70. Circuits and layout must meet timing and power specifications as well.

A *testbench* is used to verify that the logic is correct. The testbench instantiates the logic under test. It reads a file of inputs and expected outputs called *test vectors*, applies them to the module under test, and logs mismatches. Appendix A.12 provides an example of a testbench for verifying the MIPS processor logic.

A number of techniques are available for circuit verification. If the logic is synthesized onto a cell library, the postsynthesis gate-level netlist can be expressed in an HDL again and simulated using the same test vectors. Alternatively, a transistor-level netlist can be simulated against the test vector, although this can result in tricky race conditions for sequential circuits. Powerful *formal verification* tools are also available to check that a circuit performs the same Boolean function as the associated logic. Exotic circuits should be simulated thoroughly to ensure that they perform the intended logic function and have adequate noise margins; circuit pitfalls are discussed throughout this book.

Layout vs. Schematic tools (LVS) check that transistors in a layout are connected in the same way as in the circuit schematic. *Design rule checkers* (DRC) verify that the layout satisfies design rules. *Electrical rule checkers* (ERC) scan for other potential problems such as noise or premature wearout; such problems will also be discussed later in the book.

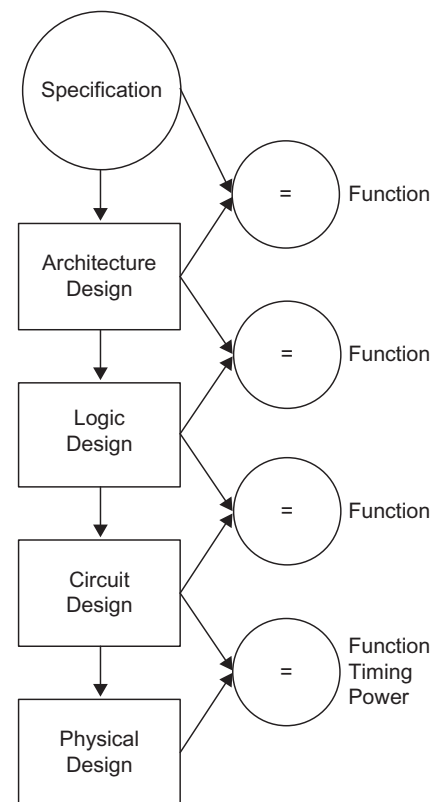


FIGURE 1.70 Design and verification sequence

⁷1 kgate = 1000 gates.

1.12 Fabrication, Packaging, and Testing

Once a chip design is complete, it is taped out for manufacturing. *Tapeout* gets its name from the old practice of writing a specification of masks to magnetic tape; today, the mask descriptions are usually sent to the manufacturer electronically. Two common formats for mask descriptions are the Caltech Interchange Format (CIF) [Mead80] (mainly used in academia) and the Calma GDS II Stream Format (GDS) [Calma84] (used in industry).

Masks are made by etching a pattern of chrome on glass with an electron beam. A set of masks for a nanometer process can be very expensive. For example, masks for a large chip in a 180 nm process may cost on the order of a quarter of a million dollars. In a 65 nm process, the mask set costs about \$3 million. The MOSIS service in the United States and its EUROPRACTICE and VDEC counterparts in Europe and Japan make a single set of masks covering multiple small designs from academia and industry to amortize the cost across many customers. With a university discount, the cost for a run of 40 small chips on a multi-project wafer can run about \$10,000 in a 130 nm process down to \$2000 in a 0.6 μm process. MOSIS offers certain grants to cover fabrication of class project chips.

Integrated circuit fabrication plants (fabs) now cost billions of dollars and become obsolete in a few years. Some large companies still own their own fabs, but an increasing number of fabless semiconductor companies contract out manufacturing to foundries such as TSMC, UMC, and IBM.

Multiple chips are manufactured simultaneously on a single silicon wafer, typically 150–300 mm (6"–12") in diameter. Fabrication requires many deposition, masking, etching, and implant steps. Most fabrication plants are optimized for wafer throughput rather than latency, leading to turnaround times of up to 10 weeks. Figure 1.71 shows an engineer in a *clean room* holding a completed 300 mm wafer. Clean rooms are filtered to eliminate most dust and other particles that could damage a partially processed wafer. The engineer is wearing a “bunny suit” to avoid contaminating the clean room. Figure 1.72 is a



FIGURE 1.71 Engineer holding processed 12-inch wafer (Photograph courtesy of the Intel Corporation.)

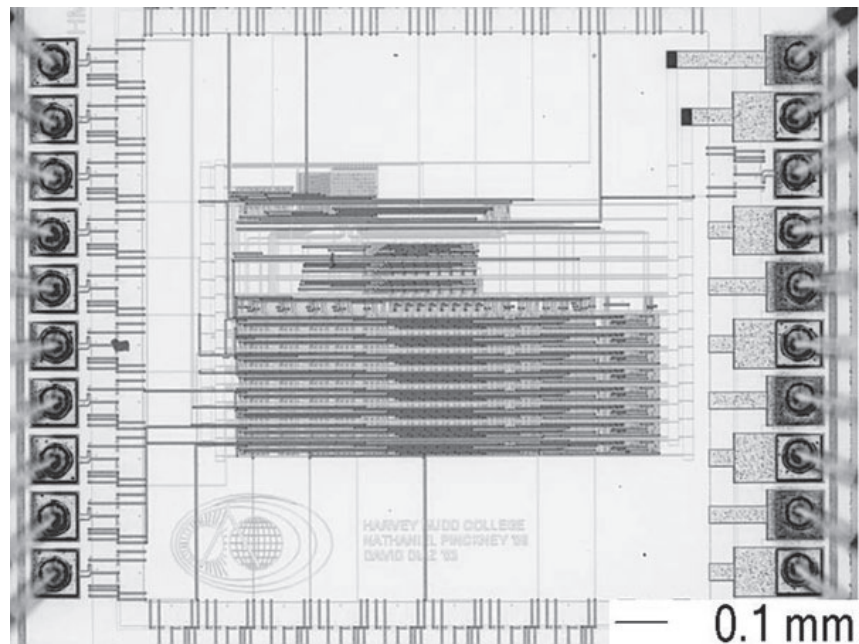


FIGURE 1.72 MIPS processor photomicrograph (only part of pad frame shown)

photomicrograph (a photograph taken under a microscope) of the 8-bit MIPS processor.

Processed wafers are sliced into dice (chips) and packaged. Figure 1.73 shows the 1.5×1.5 mm chip in a 40-pin *dual-inline package* (DIP). This *wire-bonded* package uses thin gold wires to connect the pads on the die to the lead frame in the center cavity of the package. These wires are visible on the pads in Figure 1.72. More advanced packages offer different trade-offs between cost, pin count, pin bandwidth, power handling, and reliability, as will be discussed in Section 13.2. Flip-chip technology places small solder balls directly onto the die, eliminating the bond wire inductance and allowing contacts over the entire chip area rather than just at the periphery.

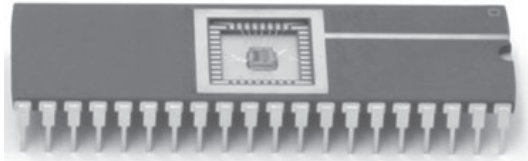


FIGURE 1.73 Chip in a 40-pin dual-inline package

Even tiny defects in a wafer or dust particles can cause a chip to fail. Chips are tested before being sold. Testers capable of handling high-speed chips cost millions of dollars, so many chips use built-in self-test features to reduce the tester time required. Chapter 15 is devoted to design verification and testing.

Summary and a Look Ahead

“If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost \$100, get one million miles to the gallon, and explode once a year . . .”

—Robert X. Cringely

CMOS technology, driven by Moore’s Law, has come to dominate the semiconductor industry. This chapter examined the principles of designing a simple CMOS integrated circuit. MOS transistors can be viewed as electrically controlled switches. Static CMOS gates are built from pull-down networks of nMOS transistors and pull-up networks of pMOS transistors. Transistors and wires are fabricated on silicon wafers using a series of deposition, lithography, and etch steps. These steps are defined by a set of masks drawn as a chip layout. Design rules specify minimum width and spacing between elements in the layout. The chip design process can be divided into architecture, logic, circuit, and physical design. The performance, area, and power of the chip are influenced by interrelated decisions made at each level. Design verification plays an important role in constructing such complex systems; the reliability requirements for hardware are much greater than those typically imposed on software.

Primary design objectives include reliability, performance, power, and cost. Any chip should, with high probability, operate reliably for its intended lifetime. For example, the chip must be designed so that it does not overheat or break down from excessive voltage. Performance is influenced by many factors including clock speed and parallelism. CMOS transistors dissipate power every time they switch, so the dynamic power consumption is related to the number and size of transistors and the rate at which they switch. At feature

sizes below 180 nm, transistors also leak a significant amount of current even when they should be OFF. Thus, chips now draw static power even when they are idle. One of the central challenges of VLSI design is making good trade-offs between performance and power for a particular application. The cost of a chip includes nonrecurring engineering (NRE) expenses for the design and masks, along with per-chip manufacturing costs related to the size of the chip. In processes with smaller feature sizes, the per-unit cost goes down because more transistors can be packed into a given area, but the NRE increases. The latest manufacturing processes are only cost-effective for chips that will sell in huge volumes. Nevertheless, plenty of interesting markets exist for chips in mature, inexpensive manufacturing processes.

To quantify how a chip meets these objectives, we must develop and analyze more complete models. The remainder of this book will expand on the material introduced in this chapter. Of course, transistors are not simply switches. Chapter 2 examines the current and capacitance of transistors, which are essential for estimating delay and power. A more detailed description of CMOS processing technology and layout rules is presented in Chapter 3. The next four chapters address the fundamental concerns of circuit designers. The models from Chapter 2 are too detailed to apply by hand to large systems, yet not detailed enough to fully capture the complexity of modern transistors. Chapter 4 develops simplified models to estimate the delay of circuits. If modern chips were designed to squeeze out the ultimate possible performance without regard to power, they would burn up. Thus, it is essential to estimate and trade off the power consumption against performance. Moreover, low power consumption is crucial to mobile battery-operated systems. Power is considered in Chapter 5. Wires are as important as transistors in their contribution to overall performance and power, and are discussed in Chapter 6. Chapter 7 addresses design of robust circuits with a high yield and low failure rate.

Simulation is discussed in Chapter 8 and is used to obtain more accurate performance and power predictions as well as to verify the correctness of circuits and logic. Chapter 9 considers combinational circuit design. A whole kit of circuit families are available with different trade-offs in speed, power, complexity, and robustness. Chapter 10 continues with sequential circuit design, including clocking and latching techniques.

The next three chapters delve into CMOS subsystems. Chapter 11 catalogs designs for a host of datapath subsystems including adders, shifters, multipliers, and counters. Chapter 12 similarly describes memory subsystems including SRAMs, DRAMs, CAMs, ROMs, and PLAs. Chapter 13 addresses special-purpose subsystems including power distribution, clocking, and I/O.

The final chapters address practicalities of CMOS system design. Chapter 14 focuses on a range of current design methods, identifying the issues peculiar to CMOS. Testing, design-for-test, and debugging techniques are discussed in Chapter 15. Hardware description languages (HDLs) are used in the design of nearly all digital integrated circuits today. Appendix A provides side-by-side tutorials for Verilog and VHDL, the two dominant HDLs.

A number of sections are marked with an “optional” icon. These sections describe particular subjects in greater detail. You may skip over these sections on a first reading and return to them when they are of practical relevance. To keep the length of this book under control, some optional topics have been published on the Internet rather than in print. These sections can be found at www.cmosvlsi.com and are labeled with a “Web Enhanced” icon. A companion text, *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools* [Brunvand09], covers practical details of using the leading industrial CAD tools to build chips.

Exercises

- 1.1 Extrapolating the data from Figure 1.4, predict the transistor count of a microprocessor in 2016.
- 1.2 Search the Web for transistor counts of Intel's more recent microprocessors. Make a graph of transistor count vs. year of introduction from the Pentium Processor in 1993 to the present on a semilogarithmic scale. How many months pass between doubling of transistor counts?
- 1.3 As the cost of a transistor drops from a microbuck ($\$10^{-6}$) toward a nanobuck, what opportunities can you imagine to change the world with integrated circuits?
- 1.4 Read a biography or history about a major event in the development of integrated circuits. For example, see *Crystal Fire* by Lillian Hoddeson, *Microchip* by Jeffrey Zygmunt, or *The Pentium Chronicles* by Robert Colwell. Pick a team or individual that made a major contribution to the field. In your opinion, what were the characteristics that led to success? What traits of the team management would you seek to emulate or avoid in your own professional life?
- 1.5 Sketch a transistor-level schematic for a CMOS 4-input NOR gate.
- 1.6 Sketch a transistor-level schematic for a compound CMOS logic gate for each of the following functions:
 - a) $Y = \overline{ABC + D}$
 - b) $Y = (\overline{AB + C}) \cdot \overline{D}$
 - c) $Y = \overline{AB + C \cdot (A + B)}$
- 1.7 Use a combination of CMOS gates (represented by their symbols) to generate the following functions from A , B , and C .
 - a) $Y = A$ (buffer)
 - b) $Y = A\overline{B} + \overline{A}B$ (XOR)
 - c) $Y = \overline{A}\overline{B} + AB$ (XNOR)
 - d) $Y = AB + BC + AC$ (majority)
- 1.8 Sketch a transistor-level schematic of a CMOS 3-input XOR gate. You may assume you have both true and complementary versions of the inputs available.
- 1.9 Sketch transistor-level schematics for the following logic functions. You may assume you have both true and complementary versions of the inputs available.
 - a) A 2:4 decoder defined by

$$Y_0 = \overline{A_0} \cdot \overline{A_1}$$

$$Y_1 = A_0 \cdot \overline{A_1}$$

$$Y_2 = \overline{A_0} \cdot A_1$$

$$Y_3 = A_0 \cdot A_1$$
 - b) A 3:2 priority encoder defined by

$$Y_0 = \overline{A_0} \cdot (A_1 + \overline{A_2})$$

$$Y_1 = \overline{A_0} \cdot \overline{A_1}$$

- 1.10 Sketch a stick diagram for a CMOS 4-input NOR gate from Exercise 1.5.
- 1.11 Estimate the area of your 4-input NOR gate from Exercise 1.10.
- 1.12 Using a CAD tool of your choice, layout a 4-input NOR gate. How does its size compare to the prediction from Exercise 1.11?
- 1.13 Figure 1.74 shows a stick diagram of a 2-input NAND gate. Sketch a side view (cross-section) of the gate from X to X'.
- 1.14 Figure 1.75 gives a stick diagram for a level-sensitive latch. Estimate the area of the latch.
- 1.15 Draw a transistor-level schematic for the latch of Figure 1.75. How does the schematic differ from Figure 1.31(b)?
- 1.16 Consider the design of a CMOS compound OR-AND-INVERT (OAI21) gate computing $F = (\overline{A + B}) \cdot \overline{C}$.
 - a) sketch a transistor-level schematic
 - b) sketch a stick diagram
 - c) estimate the area from the stick diagram
 - d) layout your gate with a CAD tool using unit-sized transistors
 - e) compare the layout size to the estimated area

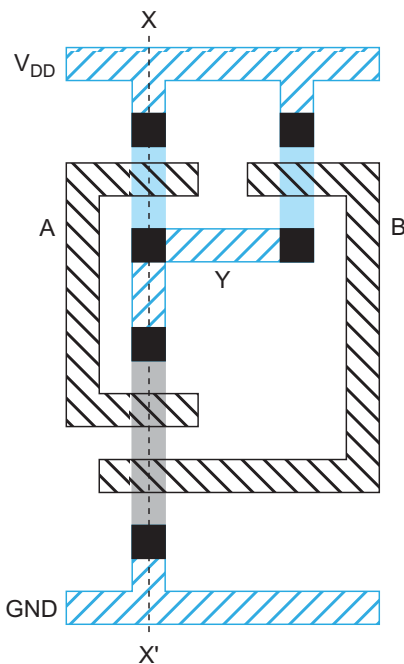


FIGURE 1.74 2-input NAND gate stick diagram

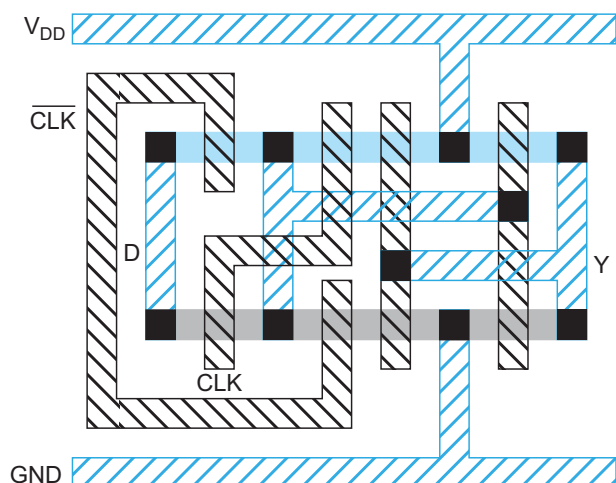


FIGURE 1.75 Level-sensitive latch stick diagram

- 1.17 Consider the design of a CMOS compound OR-OR-AND-INVERT (OAI22) gate computing $F = \overline{(A + B) \cdot (C + D)}$.
- a) sketch a transistor-level schematic
 - b) sketch a stick diagram
 - c) estimate the area from the stick diagram
 - d) layout your gate with a CAD tool using unit-sized transistors
 - e) compare the layout size to the estimated area
- 1.18 A 3-input majority gate returns a true output if at least two of the inputs are true. A minority gate is its complement. Design a 3-input CMOS minority gate using a single stage of logic.
- a) sketch a transistor-level schematic
 - b) sketch a stick diagram
 - c) estimate the area from the stick diagram
- 1.19 Design a 3-input minority gate using CMOS NANDs, NORs, and inverters. How many transistors are required? How does this compare to a design from Exercise 1.18(a)?
- 1.20 A carry lookahead adder computes $G = G_3 + P_3(G_2 + P_2(G_1 + P_1 G_0))$. Consider designing a compound gate to compute $\bar{G}$.
- a) sketch a transistor-level schematic
 - b) sketch a stick diagram
 - c) estimate the area from the stick diagram
- 1.21 www.cmosvlsi.com has a series of four labs in which you can learn VLSI design by completing the multicycle MIPS processor described in this chapter. The labs use the open-source Electric CAD tool or commercial tools from Cadence and Synopsys. They cover the following:
- a) leaf cells: schematic entry, layout, icons, simulation, DRC, ERC, LVS; hierarchical design
 - b) datapath design: wordslices, ALU assembly, datapath routing
 - c) control design: random logic or PLAs
 - d) chip assembly, pad frame, global routing, full-chip verification, tapeout

CMOS Processing Technology

3

3.1 Introduction

Chapter 1 summarized the steps in a basic CMOS process. These steps are expanded upon in this chapter. Where possible, the processing details are related to the way CMOS circuits and systems are designed. Modern CMOS processing is complex, and while coverage of every nuance is not within the scope of this book, we focus on the fundamental concepts that impact design.

A fair question from a designer would be “Why do I care how transistors are made?” In many cases, if designers understand the physical process, they will comprehend the reason for the underlying design rules and in turn use this knowledge to create a better design. Understanding the manufacturing steps is also important when debugging some difficult chip failures and improving yield.

Fabrication plants, or *fabs*, are enormously expensive to develop and operate. In the early days of the semiconductor industry, a few bright physicists and engineers could bring up a fabrication facility in an industrial building at a modest cost and most companies did their own manufacturing. Now, a fab processing 300 mm wafers in a 45 nm process costs about \$3 billion. The research and development underlying the technology costs another \$2.4 billion. Only a handful of companies in the world have the sales volumes to justify such a large investment. Even these companies are forming consortia to share the costs of technology development with their market rivals. Some companies, such as TSMC, UMC, Chartered, and IBM operate on a *foundry* model, selling space on their fab line to fabless semiconductor firms. Figure 3.1 shows workers and machinery in the cavernous clean room at IBM’s East Fishkill 300 mm fab.

Recall that silicon in its pure or *intrinsic* state is a semiconductor, having bulk electrical resistance somewhere between that of a conductor and an insulator. The conductivity of silicon can be raised by several orders of magnitude by introducing *impurity* atoms into the silicon crystal lattice. These dopants can supply either free electrons or holes. Group III impurity elements such as boron that use up electrons are referred to as *acceptors* because they accept some of the electrons already in the silicon, leaving holes. Similarly, Group V *donor* elements such as arsenic and phosphorous provide electrons. Silicon that contains a majority of donors is known as *n-type*, while silicon that contains a majority of acceptors is known as *p-type*. When n-type and p-type materials are brought together, the region where the silicon changes from n-type to p-type is called a *junction*. By arranging junctions in certain physical structures and combining them with wires and insulators, various semiconductor devices can be constructed. Over the years, silicon semiconductor processing has evolved sophisticated techniques for building these junctions and other insulating and conducting structures.



FIGURE 3.1 IBM, East Fishkill, NY fab (Courtesy of International Business Machines Corporation. Unauthorized use not permitted.)

The chapter begins with the steps of a generic process characteristic of commercial 65 nm manufacturing. It also surveys a variety of process enhancements that benefit certain applications. The chapter examines layout design rules in more detail and discusses layout CAD issues such as design rule checking.

3.2 CMOS Technologies

CMOS processing steps can be broadly divided into two parts. Transistors are formed in the *Front-End-of-Line* (FEOL) phase, while wires are built in the *Back-End-of-Line* (BEOL) phase. This section examines the steps used through both phases of the manufacturing process.

3.2.1 Wafer Formation

The basic raw material used in CMOS fabs is a *wafer* or disk of silicon, roughly 75 mm to 300 mm (12"—a dinner plate!) in diameter and less than 1 mm thick. Wafers are cut from *boules*, cylindrical ingots of single-crystal silicon, that have been pulled from a crucible of pure molten silicon. This is known as the *Czochralski* method and is currently the most common method for producing single-crystal material. Controlled amounts of impurities are added to the melt to provide the crystal with the required electrical properties. A seed crystal is dipped into the melt to initiate crystal growth. The silicon ingot takes on the same crystal orientation as the seed. A graphite radiator heated by radio-frequency induction surrounds the quartz crucible and maintains the temperature a few degrees above the melting point of silicon (1425 °C). The atmosphere is typically helium or argon to prevent the silicon from oxidizing.

The seed is gradually withdrawn vertically from the melt while simultaneously being rotated, as shown in Figure 3.2. The molten silicon attaches itself to the seed and recrystallizes as it is withdrawn. The seed withdrawal and rotation rates determine the diameter of the ingot. Growth rates vary from 30 to 180 mm/hour.

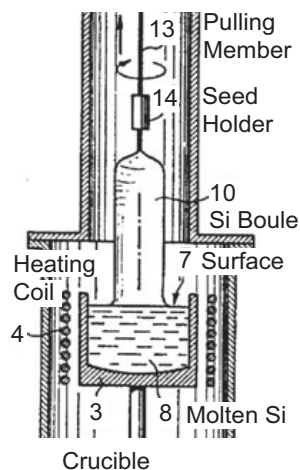


FIGURE 3.2 Czochralski system for growing Si boules (Adapted from [Schulmann98].)

3.2.2 Photolithography

Recall that regions of dopants, polysilicon, metal, and contacts are defined using masks. For instance, in places covered by the mask, ion implantation might not occur or the dielectric or metal layer might be left intact. In areas where the mask is absent, the implantation can occur, or dielectric or metal could be etched away. The patterning is achieved by a process called *photolithography*, from the Greek *photo* (light), *lithos* (stone), and *graphe* (picture), which literally means “carving pictures in stone using light.” The primary method for defining areas of interest (i.e., where we want material to be present or absent) on a wafer is by the use of *photoresists*. The wafer is coated with the photoresist and subjected to selective illumination through the *photomask*. After the initial patterning of photoresist, other barrier layers such as polycrystalline silicon, silicon dioxide, or silicon nitride can be used as physical masks on the chip. This distinction will become more apparent as this chapter progresses.

A photomask is constructed with chromium (chrome) covered quartz glass. A UV light source is used to expose the photoresist. Figure 3.3 illustrates the lithography process. The photomask has chrome where light should be blocked. The UV light floods the mask from the backside and passes through the clear sections of the mask to expose the organic photoresist (PR) that has been coated on the wafer. A developer solvent is then used to dissolve the soluble unexposed photoresist, leaving islands of insoluble exposed photoresist. This is termed a *negative* photoresist. A *positive* resist is initially insoluble, and when exposed to UV becomes soluble. Positive resists provide for higher resolution than negative resists, but are less sensitive to light. As feature sizes become smaller, the photoresist layers have to be made thinner. In turn, this makes them less robust and more subject to failure which can impact the overall yield of a process and the cost to produce the chip.

The photomask is commonly called a *reticle* and is usually smaller than the wafer, e.g., 2 cm on a side. A *stepper* moves the reticle to successive locations to completely expose the wafer. *Projection printing* is normally used, in which lenses between the reticle and wafer focus the pattern on the wafer surface. Older techniques include *contact printing*, where the mask and wafer are in contact, and *proximity printing*, where the mask and wafer are close but not touching. The reticle can be the same size as the area to be patterned (1×) or larger. For instance, 2.5× and 5× steppers with optical reduction have been used in the industry.

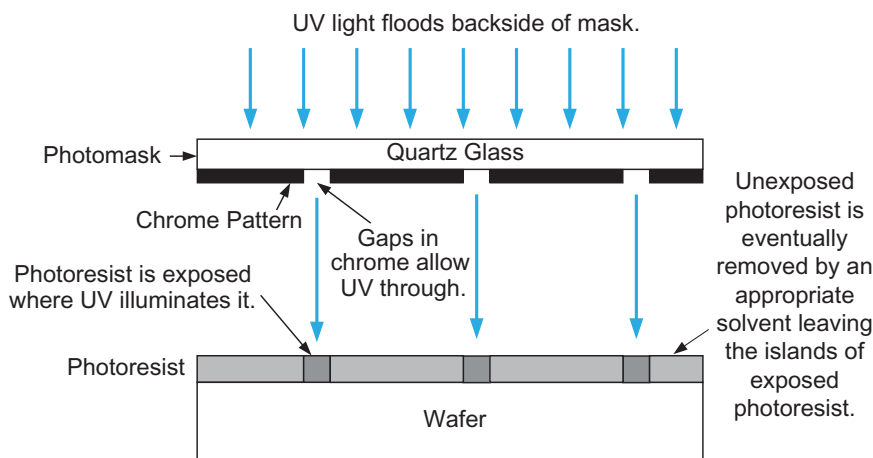


FIGURE 3.3 Photomasking with a negative resist (lens system between mask and wafer omitted to improve clarity and avoid diffracting the reader ☺)

The wavelength of the light source influences the minimum feature size that can be printed. Define the minimum *pitch* (width + spacing) of a process to be $2b$. The resolution of a lens depends on the wavelength λ of the light and the *numerical aperture* NA of the lens:

$$2b = k_1 \frac{\lambda}{\text{NA}} \quad (3.1)$$

The numerical aperture is

$$\text{NA} = n \sin \alpha \quad (3.2)$$

where n is the refractive index of the medium (1 for air, 1.33 for water, and up to 1.5 for oil), and α is the angle of acceptance of the lens. Increasing α requires larger optics. Lenses used in the 1970s had a numerical aperture of 0.2. Intel uses a numerical aperture of 0.92 for their 45 nm process [Mistry07]. Nikon and ASML broke the 1.0 barrier by introducing *immersion lithography* that takes advantage of water's higher refractive index [Geppert04], and in 2008, $\text{NA} = 1.35$ had been reached. All of these advances have come at the expense of multimillion dollar optics systems. k_1 depends on the coherence of the light, antireflective coatings, photoresist parameters, and resolution enhancement techniques. Presently, 0.8 is considered easy, while 0.5 is very hard.

The *depth of focus* is

$$\text{DOF} = \frac{k_2 \lambda}{\text{NA}^2} \quad (3.3)$$

where k_2 ranges from 0.5 to 1. Advanced lithography systems with short wavelengths and large numerical apertures have a very shallow depth of focus, requiring that the surface of the wafer be maintained extremely flat.

In the 1980s, mercury lamps with 436 nm or 365 nm wavelengths were used. At the 0.25 μm process generation, excimer lasers with 248 nm (deep ultraviolet) were adopted and have been used down to the 180 nm node. Currently, 193 nm argon-fluoride lasers are used for the *critical layers* down to the 45 nm node and beyond. The critical layers are those that define the device behavior. An example would be the gate (polysilicon), source/drain (diffusion), first metal, and contact masks. With such a laser, a numerical aperture of 1.35, and $k_1 = 0.5$, the best achievable pitch is $2b = 72$ nm, corresponding to a polysilicon half-pitch of 36 nm. It is amazing that we can print features so much smaller than the wavelength of the light, but even so, lithography is becoming a serious problem at the 45 nm node and below.

Efforts to develop 157 nm deep UV lithography systems were unsuccessful and have been abandoned by the industry. In the future, 13.5 nm extreme ultraviolet (EUV) light sources may be used, but presently, these sources require prohibitively expensive reflective optics and vacuum processing and are not strong enough for production purposes. Some predict that EUV will be ready by 2011 or 2012, while others are skeptical [Mack08].

Wavelengths comparable to or greater than the feature size cause distortion in the patterns exposed on the photoresist. *Resolution enhancement techniques* (RETs) precompensate for this distortion so the desired patterns are obtained [Schellenberg03]. These techniques involve modifying the amplitude, phase, or direction of the incoming light. The

ends of a line in a layout receive less light than the center, causing nonuniform exposure. *Optical proximity correction* (OPC) makes small changes to the patterns on the masks to compensate for these local distortions. Figure 3.4 shows an example of printing with and without optical proximity correction. OPC predistorts the corners to reduce undesired rounding. *Phase shift masks* (PSM) takes advantage of the diffraction grating effect of parallel lines on a mask, varying the thickness of the mask to change the phase such that light from adjacent lines are out of phase and cancel where no light is desired. *Off-axis illumination* can also improve contrast for certain types of dense, repetitive patterns. *Double-patterning* is a sequence of two precisely aligned exposure steps with different masks for the same photoresist layer [Mack08]. OPC became necessary at the 180 nm node and all of these techniques are in heavy use by the 45 nm node.

Each successive UV stepper is more expensive and the throughput of the stepper may decrease. This is just another contributory issue to the spiraling cost of chip manufacturing. The cost of masks is also skyrocketing, forcing chip designers to amortize design and mask expenses across the largest volume possible. This theme will be reinforced in Section 14.3.

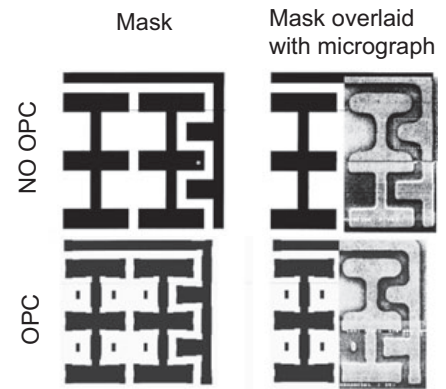


FIGURE 3.4 Subwavelength features printed with and without OPC. Predistortion of corners in OPC reduces undesired rounding. (Adapted from [Schellenberg98] with permission of SPIE.)

3.2.3 Well and Channel Formation

The following are main CMOS technologies:

- n-well process
- p-well process
- twin-well process
- triple-well process

Silicon-on-insulator processes are also available through some manufacturers (see Section 3.4.1.2).

Chapter 1 outlined an n-well process. Historically, p-well processes preceded n-well processes. In a p-well process, the nMOS transistors are built in a p-well and the pMOS transistor is placed in the n-type substrate. p-well processes were used to optimize the pMOS transistor performance. Improved techniques allowed good pMOS transistors to be fabricated in an n-well and excellent nMOS transistors to be fabricated in the p-type substrate of an n-well process. In the n-well process, each group of pMOS transistors in an n-well shares the same body node but is isolated from the bodies of pMOS transistors in different wells. However, all the nMOS transistors on the chip share the same body, which is the substrate. Noise injected into the substrate by digital circuits can disturb sensitive analog or memory circuits. Twin-well processes accompanied the emergence of n-well processes. A twin-well process allows the optimization of each transistor type. A third well can be added to create a triple-well process. The triple-well process has emerged to provide good isolation between analog and digital blocks in mixed-signal chips; it is also used to isolate high-density dynamic memory from logic. Most fabrication lines provide a baseline twin-well process that can be upgraded to a triple-well process with the addition of a single mask level.

Wells and other features require regions of doped silicon. Varying proportions of donor and acceptor dopants can be achieved using *epitaxy*, *deposition*, or *implantation*.

Epitaxy involves growing a single-crystal film on the silicon surface (which is already a single crystal) by subjecting the silicon wafer surface to an elevated temperature and a source of dopant material.

Epitaxy can be used to produce a layer of silicon with fewer defects than the native wafer surface and also can help prevent latchup (see Section 7.3.6). Foundries may provide a choice of *epi* (with epitaxial layer) or non-*epi* wafers. Microprocessor designers usually prefer to use *epi* wafers for uniformity of device performance.

Deposition involves placing dopant material onto the silicon surface and then driving it into the bulk using a thermal diffusion step. This can be used to build deep junctions. A step called *chemical vapor deposition* (CVD) can be used for the deposition. As its name suggests, CVD occurs when heated gases react in the vicinity of the wafer and produce a product that is deposited on the silicon surface. CVD is also used to lay down thin films of material later in the CMOS process.

Ion implantation involves bombarding the silicon substrate with highly energized donor or acceptor atoms. When these atoms impinge on the silicon surface, they travel below the surface of the silicon, forming regions with varying doping concentrations. At elevated temperature ($>800^\circ\text{C}$) diffusion occurs between silicon regions having different densities of impurities, with impurities tending to diffuse from areas of high concentration to areas of low concentration. Therefore, it is important to keep the remaining process steps at as low a temperature as possible once the doped areas have been put into place. However, a high-temperature annealing step is often performed after ion implantation to redistribute dopants more uniformly. Ion implantation is the standard well and source/drain implant method used today. The placement of ions is a random process, so doping levels cannot be perfectly controlled, especially in tiny structures with relatively small numbers of dopant atoms. Statistical dopant fluctuations lead to variations in the threshold voltage that will be discussed in Section 7.5.2.2.

The first step in most CMOS processes is to define the well regions. In a triple-well process, a deep n-well is first driven into the p-type substrate, usually using high-energy Mega electron volt levels (MeV) ion implantation as opposed to a thermally diffused operation. This avoids the thermal cycling (i.e., the wafers do not have to be raised significantly in temperature), which improves throughput and reliability. A 2–3 MeV implantation can yield a 2.5–3.5 μm deep n-well. Such a well has a peak dopant concentration just under the surface and for this reason is called a *retrograde* well. This can enhance device performance by providing improved latchup characteristics and reduced susceptibility to vertical punch-through (see Section 7.3.5). A thick (3.5–5.5 μm) resist has to be used to block the high energy implantation where no well should be formed. Thick resists and deep implants necessarily lead to fairly coarse feature dimensions for wells, compared to the minimum feature size. Shallower n-well and p-well regions are then implanted. After the wells have been formed, the doping levels can be adjusted (using a *threshold implant*) to set the desired threshold voltages for both nMOS and pMOS transistors. With multiple threshold implant masks, multiple V_t options can be provided on the same chip. For a given gate and substrate material, the threshold voltage depends on the doping level in the substrate (N_A), the oxide thickness (t_{ox}), and the surface state charge (Q_{fc}). The implant can affect both N_A and Q_{fc} and hence V_t . Figure 3.5 shows a typical triple-well structure. As discussed, the nMOS transistor is situated in the p-well located in the deep n-well. Other nMOS transistors could be built in different p-wells so that they do not share the same body node. Transistors in a p-well in a triple-well process will have different characteristics than transistors in the substrate because of the different doping levels. The pMOS

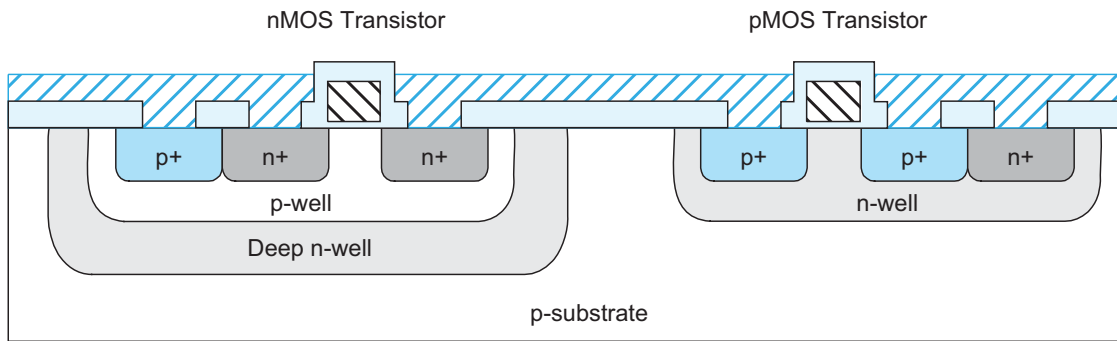


FIGURE 3.5 Well structure in triple-well process

transistors are located in the shallow (normal) n-well. The figure shows the cross-section of an inverter.

Wells are defined by separate masks. In the case of a twin-well process, only one mask need be defined because the other well by definition is its complement. Triple-well processes have to define at least two masks, one for the deep well and the other for either n-well or p-well.

Transistors near the edge of a retrograde well (e.g., within about $1\ \mu\text{m}$) may have different threshold voltages than those far from the edge because ions scatter off the photoresist mask into the edge of the well, as shown in Figure 3.6 [Hook03]. This is called the *well-edge proximity effect*.

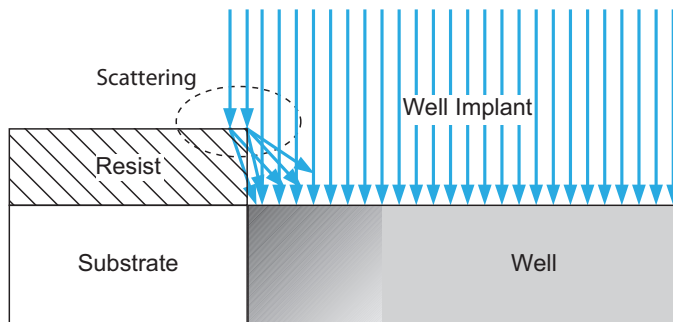


FIGURE 3.6 Well-edge proximity effect, in which dopants scattering off photoresist increase the doping level near the edge of a well
(© IEEE 2003.)

3.2.4 Silicon Dioxide (SiO_2)

Many of the structures and manufacturing techniques used to make silicon integrated circuits rely on the properties of SiO_2 . Therefore, reliable manufacture of SiO_2 is extremely important. In fact, unlike competing materials, silicon has dominated the industry because it has an easily processable oxide (i.e., it can be grown and etched). Various thicknesses of SiO_2 may be required, depending on the particular process. Thin oxides are required for transistor gates; thicker oxides might be required for higher voltage devices, while even thicker oxide layers might be required to ensure that transistors are not formed unintentionally in the silicon beneath polysilicon wires (see the next section).

Oxidation of silicon is achieved by heating silicon wafers in an oxidizing atmosphere. The following are some common approaches:

- *Wet oxidation*—when the oxidizing atmosphere contains water vapor. The temperature is usually between 900 °C and 1000 °C. This is also called *pyrogenic* oxidation when a 2:1 mixture of hydrogen and oxygen is used. Wet oxidation is a rapid process.
- *Dry oxidation*—when the oxidizing atmosphere is pure oxygen. Temperatures are in the region of 1200 °C to achieve an acceptable growth rate. Dry oxidation forms a better quality oxide than wet oxidation. It is used to form thin, highly controlled gate oxides, while wet oxidation may be used to form thick field oxides.
- *Atomic layer deposition (ALD)*—when a thin chemical layer (material *A*) is attached to a surface and then a chemical (material *B*) is introduced to produce a thin layer of the required layer (i.e., SiO₂—this can also be used for other various dielectrics and metals). The process is then repeated and the required layer is built up layer by layer. [George96, Klaus98].

The oxidation process normally consumes part of the silicon wafer (deposition and ALD do not). Since SiO₂ has approximately twice the volume of silicon, the SiO₂ layer grows almost equally in both vertical directions. Thus, after processing, the SiO₂ projects above and below the original unoxidized silicon surface.

3.2.5 Isolation

Individual devices in a CMOS process need to be isolated from one another so that they do not have unexpected interactions. In particular, channels should only be inverted beneath transistor gates over the active area; wires running elsewhere shouldn't create parasitic MOS channels. Moreover, the source/drain diffusions of unrelated transistors should not interfere with each other.

The process flow in Section 1.5 was historically used to provide this isolation. The transistor gate consists of a thin *gate oxide* layer. Elsewhere, a thicker layer of field oxide separates polysilicon and metal wires from the substrate. The MOS sandwich formed by the wire, thick oxide, and substrate behaves as an unwanted parasitic transistor. However, the thick oxide effectively sets a threshold voltage greater than V_{DD} that prevents the transistor from turning ON during normal operation. Actually, these *field devices* can be used for I/O protection and are discussed in Section 13.6.2. The source and drain of the transistors form reverse-biased p-n junctions with the substrate or well, isolating them from their neighbors.

The thick oxide used to be formed by a process called *Local Oxidation of Silicon* (LOCOS). A problem with LOCOS-based processes is the transition between thick and thin oxide, which extended some distance laterally to form a so-called *bird's beak*. The lateral distance is proportional to the oxide thickness, which limits the packing density of transistors.

Starting around the 0.35 μm node, *shallow trench isolation* (STI) was introduced to avoid the problems with LOCOS. STI forms insulating trenches of SiO₂ surrounding the transistors (everywhere except the active area). The trench width is independent of its depth, so transistors can be packed as closely as the lithography permits. The trenches isolate the wires from the substrate, preventing unwanted channel formation. They also reduce the sidewall capacitance and junction leakage current of the source and drain.

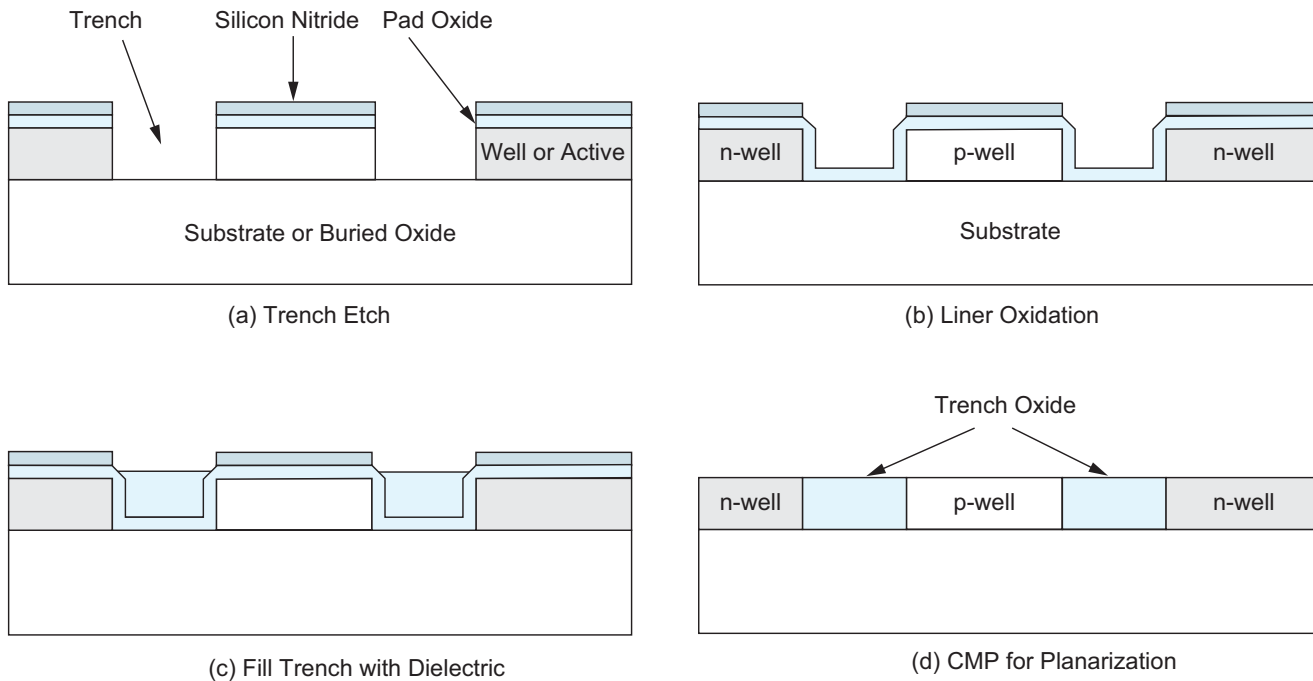


FIGURE 3.7 Shallow trench isolation

STI starts with a *pad oxide* and a silicon nitride layer, which act as the masking layers, as shown in Figure 3.7. Openings in the pad oxide are then used to etch into the well or substrate region (this process can also be used for source/drain diffusion). A liner oxide is then grown to cover the exposed silicon (Figure 3.7(b)). The trenches are filled with SiO₂ or other fillers using CVD that does not consume the underlying silicon (Figure 3.7(c)). The pad oxide and nitride are removed and a *Chemical Mechanical Polishing* (CMP) step is used to planarize the structure (Figure 3.7(d)). CMP, as its name suggests, combines a mechanical grinding action in which the rotating wafer is contacted by a stationary polishing head while an abrasive mixture is applied. The mixture also reacts chemically with the surface to aid in the polishing action. CMP is used to achieve flat surfaces, which are of central importance in modern processes with many layers.

From the designer's perspective, the presence of a deep n-well and/or trench isolation makes it easier to isolate noise-sensitive (analog or memory) portions of a chip from digital sections. Trench isolation also permits nMOS and pMOS transistors to be placed closer together because the isolation provides a higher source/drain *breakdown voltage*—the voltage at which a source or drain diode starts to conduct in the reverse-biased condition. The breakdown voltage must exceed the supply voltage (so junctions do not break down during normal operation) and is determined by the junction dimensions and doping levels of the junction formed. Deeper trenches increase the breakdown voltage.

3.2.6 Gate Oxide

The next step in the process is to form the gate oxide for the transistors. As mentioned, this is most commonly in the form of *silicon dioxide* (SiO₂).

In the case of STI-defined source/drain regions, the gate oxide is grown on top of the planarized structure that occurs at the stage shown in Figure 3.7(d). This is shown in

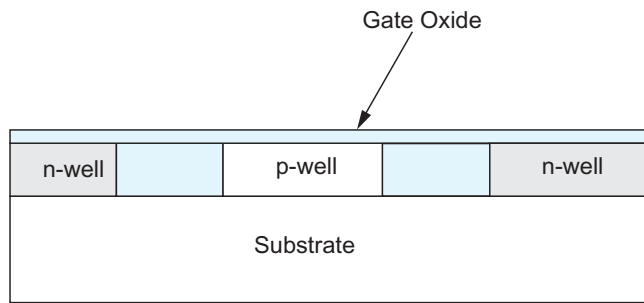


FIGURE 3.8 Gate oxide formation

Figure 3.8. The oxide structure is called the *gate stack*. This term arises because current processes seldom use a pure SiO_2 gate oxide, but prefer to produce a stack that consists of a few atomic layers, each 3–4 Å thick, of SiO_2 for reliability, overlaid with a few layers of an *oxynitrided* oxide (one with nitrogen added). The presence of the nitrogen increases the dielectric constant, which decreases the *effective oxide thickness* (EOT); this means that for a given oxide thickness, it performs like a thinner oxide. Being able to use a thicker oxide improves the robustness of the process. This concept is revisited in Section 3.4.1.3.

Many processes in the 180 nm generation and beyond provide at least two oxide thicknesses, as will be discussed in Section 3.4.1.1 (thin for logic transistors and thick for I/O transistors that must withstand higher voltages). At the 65 nm node, the effective thickness of the thin gate oxide is only 10.5–15 Å.

3.2.7 Gate and Source/Drain Formations

When silicon is deposited on SiO_2 or other surfaces without crystal orientation, it forms polycrystalline silicon, commonly called *polysilicon* or simply *poly*. An annealing process is used to control the size of the single crystal domains and to improve the quality of the polysilicon. Undoped polysilicon has high resistivity. The resistance can be reduced by implanting it with dopants and/or combining it with a refractory metal. The polysilicon gate serves as a mask to allow precise alignment of the source and drain on either side of the gate. This process is called a *self-aligned polysilicon gate* process. Aluminum could not be used because it would melt during formation of the source and drain.

As a historical note, early metal-gate processes first diffused source and drain regions, and then formed a metal gate. If the gate was misaligned, it could fail to cover the entire channel and lead to a transistor that never turned ON. To prevent this, the metal gate had to overhang the source and drain by more than the alignment tolerance of the process. This created large parasitic gate-to-source and gate-to-drain overlap capacitances that degraded switching speeds.

The steps to define the gate, source, and drain in a self-aligned polysilicon gate are as follows:

- Grow gate oxide wherever transistors are required (area = source + drain + gate)—elsewhere there will be thick oxide or trench isolation (Figure 3.9(a))
- Deposit polysilicon on chip (Figure 3.9(b))
- Pattern polysilicon (both gates and interconnect) (Figure 3.9(c))
- Etch exposed gate oxide—i.e., the area of gate oxide where transistors are required that was not covered by polysilicon; at this stage, the chip has windows down to the well or substrate wherever a source/drain diffusion is required (Figure 3.9(d))
- Implant pMOS and nMOS source/drain regions (Figure 3.9(e))

The source/drain implant density is relatively low, typically in the range 10^{18} – 10^{20} cm^{-3} of impurity atoms. Such a *lightly doped drain* (LDD) structure reduces the electric field at the drain junction (the junction with the highest voltage), which improves the immunity of the device to hot electron damage (see Section 7.3.6) and suppresses short-channel effects. The LDD implants are shallow and lightly doped, so they exhibit low

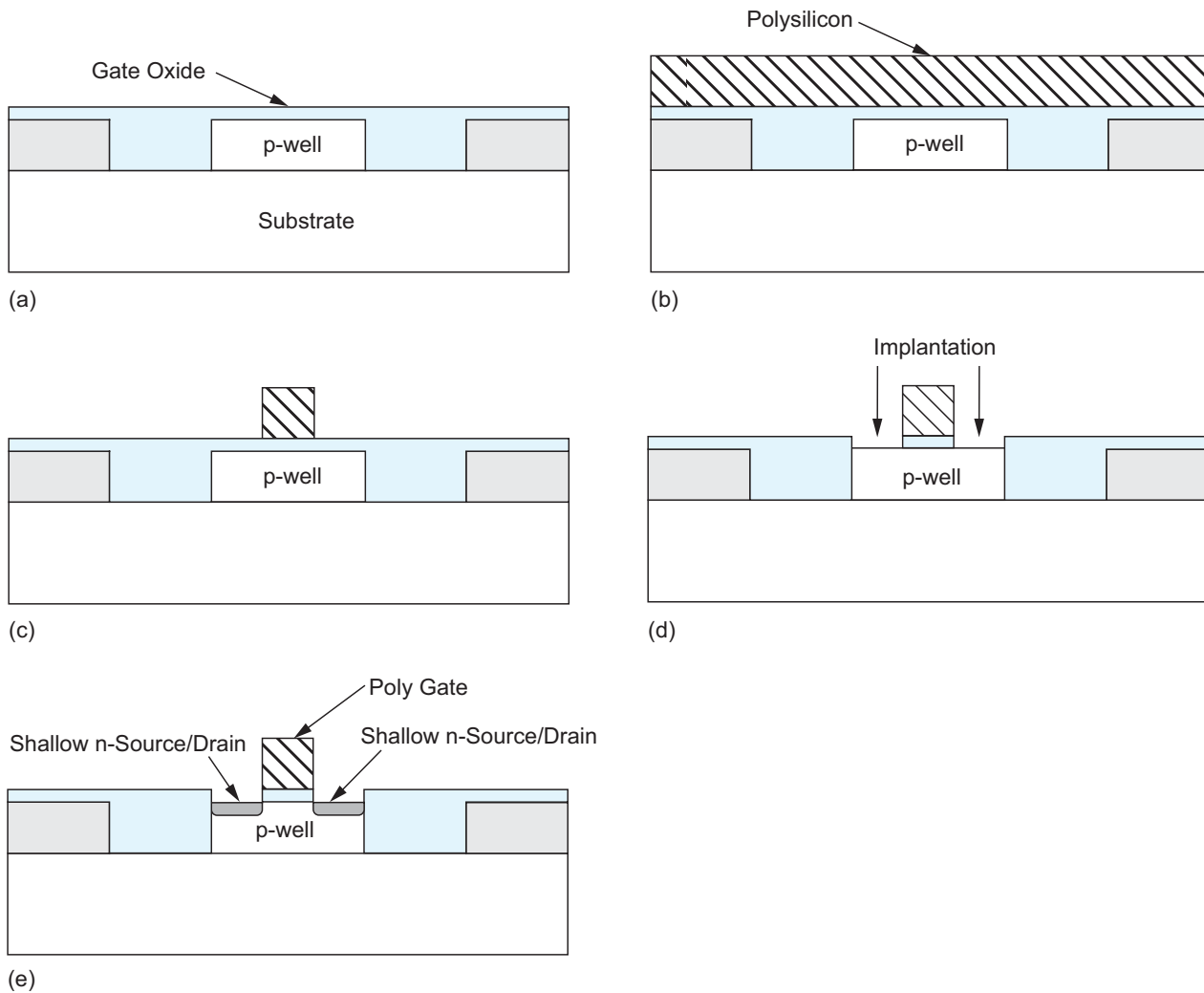


FIGURE 3.9 Gate and shallow source/drain definition

capacitance but high resistance. This reduces device performance somewhat because of the resistance in series with the transistor. Consequently, deeper, more heavily doped source/drain implants are needed in conjunction with the LDD implants to provide devices that combine hot electron suppression with low source/drain resistance. A silicon nitride (Si_3N_4) *spacer* along the edge of the gate serves as a mask to define the deeper diffusion regions, as shown in Figure 3.10(a). For in-depth coverage of various LDD structures, see [Ziegler02].

As mentioned, the polysilicon gate and source/drain diffusion have high resistance due to the resistivity of silicon and their extremely small dimensions. Modern processes form a surface layer of a refractory metal on the silicon to reduce the resistance. A *refractory metal* is one with a high melting point that will not be damaged during subsequent processing. Tantalum, nickel, molybdenum, titanium, or cobalt are commonly used. The metal is deposited on the silicon (specifically on the gate polysilicon and/or source/drain regions). A layer of *silicide* is formed when the two substances react at elevated temperatures. In a *polycide* process, only the gate polysilicon is silicided. In a silicide process (usually implemented as a self-aligned silicidization—from whence comes the synonymous

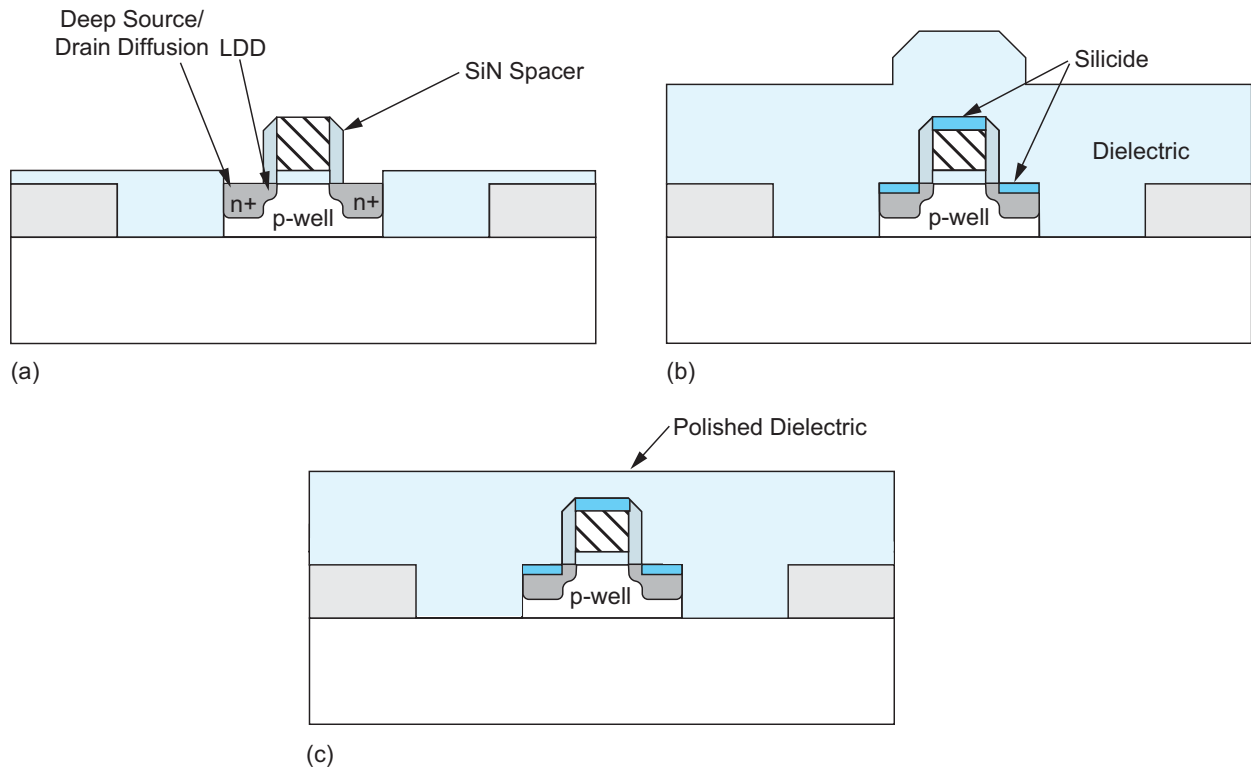


FIGURE 3.10 Transistor with LDD and deep diffusion, salicide, and planarized dielectric

term *salicide*) both gate polysilicon and source/drain regions are silicided. This process lowers the resistance of the polysilicon interconnect and the source and drain diffusion.

Figure 3.10(b) shows the resultant structure with gate and source/drain regions silicided. In addition, SiO_2 or an alternative dielectric has been used to cover all areas prior to the next processing steps. The figure shows a resulting structure with some vertical topography typical of older processes. The rapid transitions in surface height can lead to breaks in subsequent layers that fail to conform, or can entail a plethora of design rules that relate to metal edges. To avoid these problems, a CMP step is used to planarize the dielectric, leaving a flat surface for metallization as shown in Figure 3.10(c).

Nanometer processes involve another implantation step called *halo doping* that increases the doping of the substrate or well near the ends of the channels. The halo doping alleviates DIBL, short channel effects, and punchthrough but increases GIDL and BTBT leakage at the junction between the diffusion and channel [Roy03].

3.2.8 Contacts and Metallization

Contact cuts are made to source, drain, and gate according to the contact mask. These are holes etched in the dielectric after the source/drain step discussed in the previous section. Older processes commonly use aluminum (Al) for wires, although newer ones offer copper (Cu) for lower resistance. Tungsten (W) can be used as a *plug* to fill the contact holes (to alleviate problems of aluminum not conforming to small contacts). In some processes, the tungsten can also be used as a local interconnect layer.

Metallization is the process of building wires to connect the devices. As mentioned previously, conventional metallization uses aluminum. Aluminum can be deposited either

by *evaporation* or *sputtering*. Evaporation is performed by passing a high electrical current through a thick aluminum wire in a vacuum chamber. Some of the aluminum atoms are vaporized and deposited on the wafer. An improved form of evaporation that suffers less from contamination focuses an electron beam at a container of aluminum to evaporate the metal. Sputtering is achieved by generating a gas plasma by ionizing an inert gas using an RF or DC electric field. The ions are focused on an aluminum target and the plasma dislodges metal atoms, which are then deposited on the wafer.

Wet or *dry etching* can be used to remove unwanted metal. *Piranha solution* is a 3:1 to 5:1 mix of sulfuric acid and hydrogen peroxide that is used to clean wafers of organic and metal contaminants or photoresist after metal patterning. Plasma etching is a dry etch process with fluorine or chlorine gas used for metallization steps. The plasma charges the etch gas ions, which are attracted to the appropriately charged silicon surface. Very sharp etch profiles can be achieved using plasma etching. The result of the contact and metallization patterning steps is shown in Figure 3.11.

Subsequent intermetal vias and metallization are then applied. Some processes offer uniform metal dimensions for levels 2 to $n-1$, where n is the top level of metal. The top level is normally a thicker layer for use in power distribution and as such has relaxed width and spacing constraints. Other processes use successively thicker and wider metal for the upper layers, as will be explored in Section 6.1.2.

Polysilicon over diffusion normally forms a transistor gate, so a short metal1 wire is necessary to connect a diffusion output node to a polysilicon input. Some processes add tungsten (W) layer above polysilicon and below metal1; this layer is called *local interconnect* and can be drawn on a finer pitch than metal1. Local interconnect offers denser cell layouts, especially in static RAMs. Figure 3.12 shows a scanning electron micrograph of a partially completed SRAM array. The oxide has been removed to show the diffusion, polysilicon, local interconnect, and metal1. Local interconnect is used to connect the nMOS and pMOS transistors without rising up to metal1. SRAM cells are discussed further in Section 12.2.

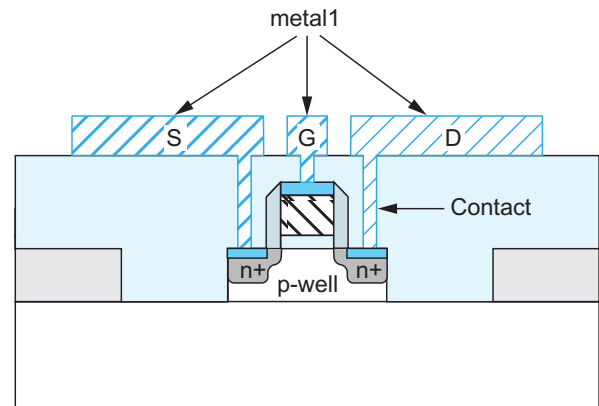


FIGURE 3.11 Aluminum metallization

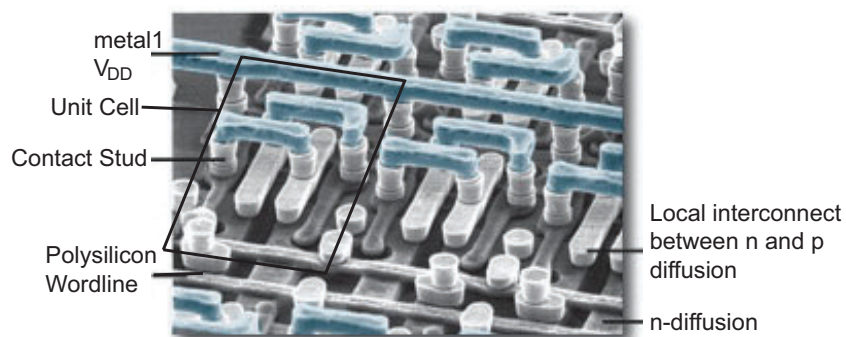


FIGURE 3.12 Partially completed 6-transistor SRAM array using local interconnect (Courtesy of International Business Machines Corporation. Unauthorized use not permitted.)

Contemporary logic processes use copper interconnects and low-k dielectrics to reduce wire resistance and capacitance. These enhancements are discussed in Section 3.4.2.

Figure 3.13 shows a cross-section of an IBM microprocessor showing the 11 layers of metal in a 90 nm process. The bottom level is tungsten local interconnect. The next five layers are on a 1× width and thickness (0.12 μm width and spacing). Metal 6–8 are on a 2× width, spacing, and thickness and metal 9–10 are 4×. These ten layers use copper wires with low-k dielectrics. The top level is aluminum and is primarily used for I/O pads. The local interconnect and metal1 are used in both directions, while the upper layers are used in alternating preferred directions. A pair of vias between metal 9 and 10 are visible. The interfaces between dielectric levels after each step of CMP are also visible.

Figure 3.14 shows a micrograph in which the oxide between metal layers has been stripped away to reveal the complex three-dimensional structure of chip wiring.

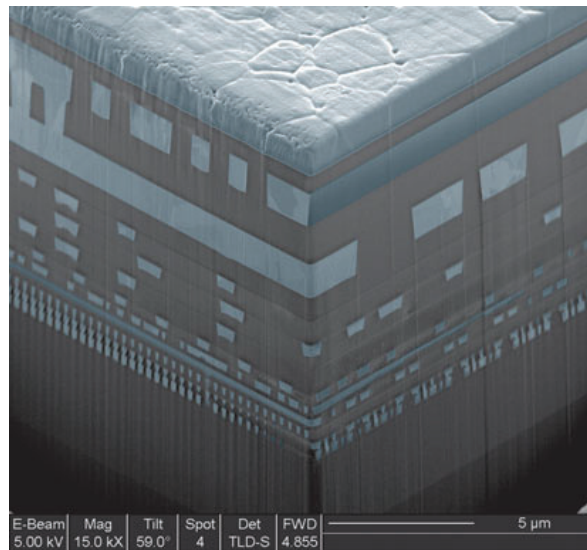


FIGURE 3.13 Cross-section showing 11 levels of metallization (Courtesy of International Business Machines Corporation. Unauthorized use not permitted.)

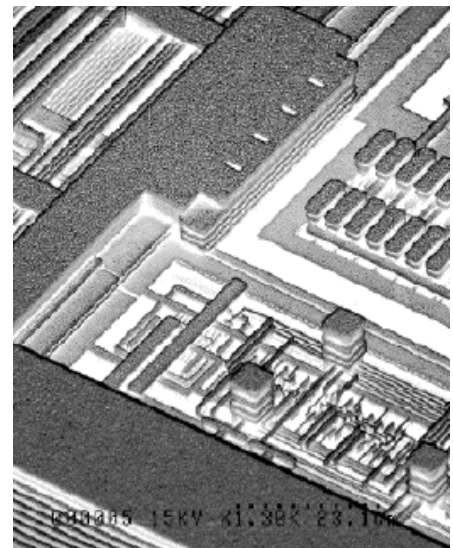


FIGURE 3.14 Micrograph of metallization in six-layer copper process (Courtesy of International Business Machines Corporation. Unauthorized use not permitted.)

3.2.9 Passivation

The final processing step is to add a protective glass layer called *passivation* or *overglass* that prevents the ingress of contaminants. Openings in the passivation layer, called *overglass cuts*, allow connection to I/O pads and test probe points if needed. After passivation, further steps can be performed such as *bumping*, which allows the chip to be directly connected to a circuit board using plated solder bumps in the pad openings.

3.2.10 Metrology

Metrology is the science of measuring. Everything that is built in a semiconductor process has to be measured to give feedback to the manufacturing process. This ranges from simple optical measurements of line widths to advanced techniques to measure thin films and defects such as voids in copper interconnect. A natural requirement exists for *in situ*

real-time measurements so that the manufacturing process can be controlled in a direct feedback manner.

Optical microscopes are used to observe large structures and defects, but are no longer adequate for structures smaller than the wavelength of visible light ($\sim 0.5 \mu\text{m}$). Scanning electron microscopy (SEM) is used to observe very small features. An SEM raster scans a structure under observation and observes secondary electron emission to produce an image of the surface of the structure. Energy Dispersive Spectroscopy (EDX) bombards a circuit with electrons causing x-ray emission. This can be used for imaging as well. A Transmission Electron Microscope (TEM), which observes the results of passing electrons through a sample (rather than bouncing them off the sample), is sometimes also used to measure structures.

3.3 Layout Design Rules

Layout rules, also referred to as *design rules*, were introduced in Chapter 1 and can be considered a prescription for preparing the photomasks that are used in the fabrication of integrated circuits. The rules are defined in terms of *feature sizes* (widths), *separations*, and *overlaps*. The main objective of the layout rules is to build reliably functional circuits in as small an area as possible. In general, design rules represent a compromise between performance and yield. The more conservative the rules are, the more likely it is that the circuit will function. However, the more aggressive the rules are, the greater the opportunity for improvements in circuit performance and size.

Design rules specify to the designer certain geometric constraints on the layout artwork so that the patterns on the processed wafer will preserve the topology and geometry of the designs. It is important to note that design rules do not represent some hard boundary between correct and incorrect fabrication. Rather, they represent a tolerance that ensures high probability of correct fabrication and subsequent operation. For example, you may find that a layout that violates design rules can still function correctly and vice versa. Nevertheless, any significant or frequent departure (*design rule waiver*) from design rules will seriously prejudice the success of a design.

Chapter 1 described a version of design rules based on the MOSIS CMOS scalable rules. The MOSIS rules are expressed in terms of λ . These rules allow some degree of scaling between processes, as in principle, you only need to reduce the value of λ and the designs will be valid in the next process down in size. Unfortunately, history has shown that processes rarely shrink uniformly. Thus, industry usually uses the actual micron design rules for layouts. At this time, custom layout is usually constrained to a number of often-used standard cells or memories, where the effort expended is amortized over many instances. Only for extremely high-volume chips is the cost savings of a smaller full-custom layout worth the labor cost of that layout.

3.3.1 Design Rule Background

We begin by examining the reasons for the most important design rules.

3.3.1.1 Well Rules The n-well is usually a deeper implant (especially a deep n-well) than the transistor source/drain implants, and therefore, it is necessary to provide sufficient clearance between the n-well edges and the adjacent n^+ diffusions. The clearance between the well edge and an enclosed diffusion is determined by the transition of the field oxide

across the well boundary. Processes that use STI may permit zero inside clearance. In older LOCOS processes, problems such as the bird's beak effect usually force substantial clearances. Being able to place nMOS and pMOS transistors closer together can significantly reduce the size of SRAM cells.

Because the n-well sheet resistance can be several $k\Omega$ per square, it is necessary to ground the well thoroughly by providing a sufficient number of well taps. This will prevent excessive voltage drops due to well currents. Guidelines on well and substrate taps are given in Section 7.3.6. Where wells are connected to different potentials (say in analog circuits), the spacing rules may differ from *equipotential* wells (all wells at the same voltage—the normal case in digital logic).

Mask Summary: The masks encountered for well specification may include n-well, p-well, and deep n-well. These are used to specify where the various wells are to be placed. Often only one well is specified in a twin-well process (i.e., n-well) and by default the p-well is in areas where the n-well isn't (i.e., p-well equals the logical NOT of the n-well).

3.3.1.2 Transistor Rules CMOS transistors are generally defined by at least four physical masks. These are active (also called diffusion, diff, thinox, OD, or RX), n-select (also called n-implant, nimp, or nplus), p-select (also called p-implant, pimp, or pplus) and polysilicon (also called poly, polyg, PO, or PC). The active mask defines all areas where either n- or p-type diffusion is to be placed *or* where the gates of transistors are to be placed. The gates of transistors are defined by the logical AND of the polysilicon mask and the active mask, i.e., where polysilicon crosses diffusion. The select layers define what type of diffusion is required. n-select surrounds active regions where n-type diffusion is required. p-select surrounds areas where p-type diffusion is required. n-diffusion areas inside p-well regions define nMOS transistors (or n-diffusion wires). n-diffusion areas inside n-well regions define n-well contacts. Likewise, p-diffusion areas inside n-wells define pMOS transistors (or p-diffusion wires). p-diffusion areas inside p-wells define substrate contacts (or p-well contacts). Frequently, design systems will define only n-diffusion (ndiff) and p-diffusion (pdiff) to reduce the complexity of the process. The appropriate selects are generated automatically. That is, ndiff will be converted automatically into active with an overlapping rectangle or polygon of n-select.

It is essential for the poly to cross active completely; otherwise the transistor that has been created will be shorted by a diffusion path between source and drain. Hence, poly is required to extend beyond the edges of the active area. This is often termed the *gate extension*. Active must extend beyond the poly gate so that diffused source and drain regions exist to carry charge into and out of the channel. Poly and active regions that should not form a transistor must be kept separated; this results in a spacing rule from active to polysilicon.

Figure 3.15(a) shows the mask construction for the final structures that appear in Figure 3.15(b).

Mask Summary: The basic masks (in addition to well masks) used to define transistors, diffusion interconnect (possibly resistors), and gate interconnect are active, n-select, p-select, and polysilicon. These may be called different names in some processes. Sometimes n-diffusion (ndiff) and p-diffusion (pdiff) masks are used in place of active to alleviate designer confusion.

3.3.1.3 Contact Rules There are several generally available contacts:

- Metal to p-active (p-diffusion)
- Metal to n-active (n-diffusion)

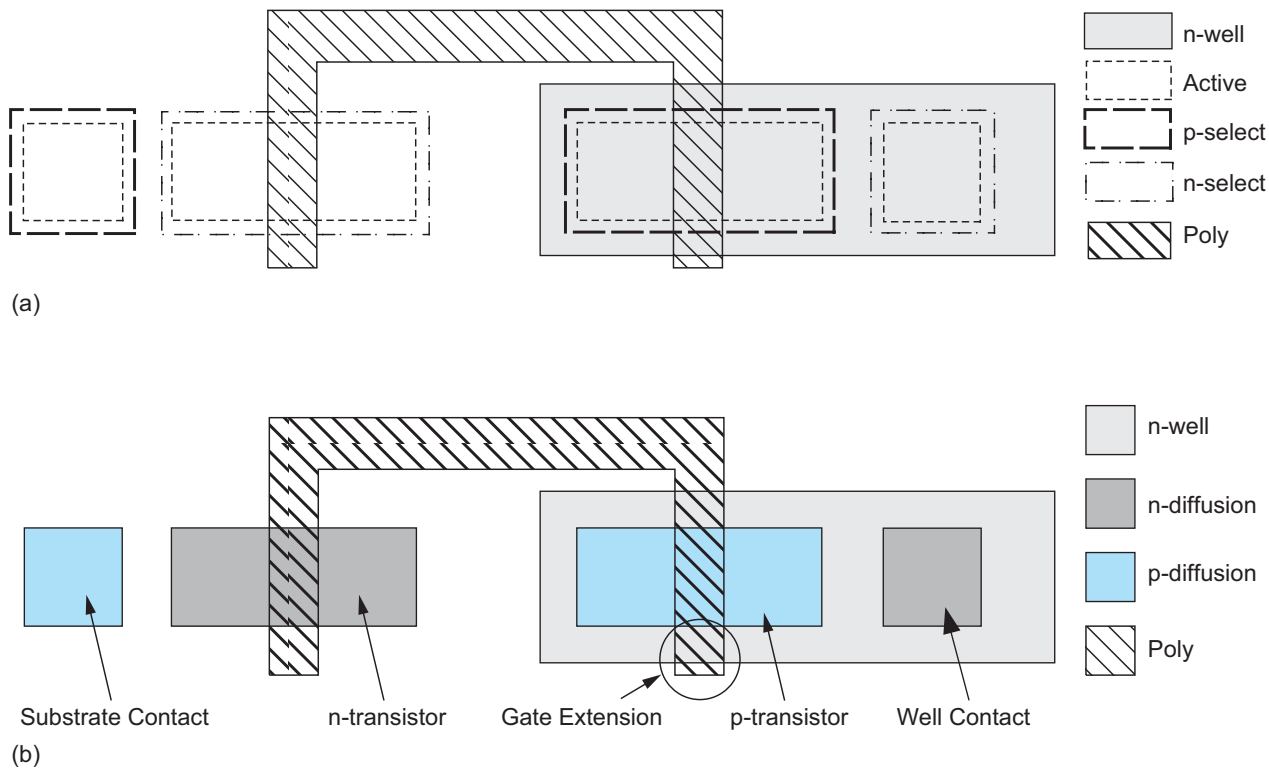


FIGURE 3.15 CMOS n-well process transistor and well/substrate contact construction

- Metal to polysilicon
- Metal to well or substrate

Depending on the process, other contacts such as *buried* polysilicon-active contacts may be allowed for local interconnect.

Because the substrate is divided into well regions, each isolated well must be tied to the appropriate supply voltage; i.e., the n-well must be tied to V_{DD} and the substrate or p-well must be tied to GND with well or substrate contacts. As mentioned in Section 1.5.1, metal makes a poor connection to the lightly doped substrate or well. Hence, a heavily doped active region is placed beneath the contact, as shown at the source of the nMOS transistor in Figure 3.16.

Whenever possible, use more than one contact at each connection. This significantly improves yield in many processes because the connection is still made even if one of the contacts is malformed.

Mask Summary: The only mask involved with contacts to active or poly is the contact mask, commonly called CONT or CA. Contacts are normally of uniform size to allow for consistent etching of very small features.

3.3.1.4 Metal Rules Metal spacing may vary with the width of the metal line (so called fat-metal rules). That is, above some metal wire width, the minimum spacing may be increased. This is due to etch characteristics of small versus large metal wires. There may also be maximum metal width rules. That is, single metal wires cannot be greater than a certain width. If wider wires are desired, they are constructed by paralleling a number of smaller wires and

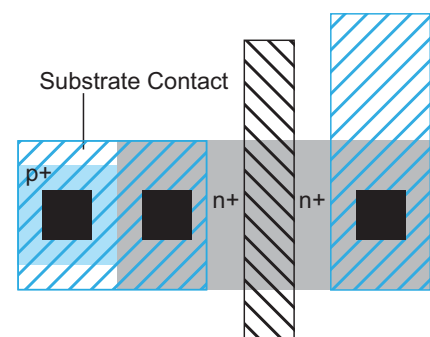


FIGURE 3.16 Substrate contact

adding checkerboard links to tie the wires together. Additionally, there may be spacing rules that are applied to long, closely spaced parallel metal lines.

Older nonplanarized processes required greater width and spacing on upper-level metal wires (e.g., metal3) to prevent breaks or shorts between adjoining wires caused by the vertical topology of the underlying layers. This is no longer a consideration for modern planarized processes. Nevertheless, width and spacing are still greater for thicker metal layers.

Mask Summary: Metal rules may be complicated by varying spacing dependent on width: As the width increases, the spacing increases. Metal overlap over contact might be zero or nonzero. Guidelines will also exist for electromigration, as discussed in Section 7.3.3.1.

3.3.1.5 Via Rules Processes may vary in whether they allow *stacked* vias to be placed over polysilicon and diffusion regions. Some processes allow vias to be placed within these areas, but do not allow the vias to straddle the boundary of polysilicon or diffusion. This results from the sudden vertical topology variations that occur at sublayer boundaries. Modern planarized processes permit stacked vias, which reduces the area required to pass from a lower-level metal to a high-level metal.

Mask Summary: Vias are normally of uniform size within a layer. They may increase in size toward the top of a metal stack. For instance, large vias required on power busses are constructed from an array of uniformly sized vias.

3.3.1.6 Other Rules The passivation or overglass layer is a protective layer of SiO₂ (glass) that covers the final chip. Appropriately sized openings are required at pads and any internal test points.

Some additional rules that might be present in some processes are as follows:

- Extension of polysilicon or metal beyond a contact or via
- Differing gate poly extensions depending on the device length
- Maximum width of a feature
- Minimum area of a feature (small pieces of photoresist can peel off and float away)
- Minimum notch sizes (small notches are rarely beneficial and can interfere with resolution enhancement techniques)

3.3.1.7 Summary Whereas earlier processes tended to be process driven and frequently had long and involved design rules, processes have become increasingly “designer friendly” or, more specifically, computer friendly (most of the mask geometries for designs are algorithmically produced). Companies sometimes create “generic” rules that span a number of different CMOS foundries that they might use. Some processes have design guidelines that feature structures to be avoided to ensure good yields. Traditionally, engineers followed yield-improvement cycles to determine the causes of defective chips and modify the layout to avoid the most common systematic failures. Time to market and product life cycles are now so short that yield improvement is only done for the highest volume parts. It is often better to reimplement a successful product in a new, smaller technology rather than to worry about improving the yield on the older, larger process.

3.3.2 Scribe Line and Other Structures

The *scribe line* surrounds the completed chip where it is cut with a diamond saw. The construction of the scribe line varies from manufacturer to manufacturer. It is designed to

prevent the ingress of contaminants from the side of the chip (as opposed to the top of the chip, which is protected by the overglass).

Several other structures are included on a mask including the *alignment mark*, *critical dimension* structures, *vernier* structures, and *process check* structures [Hess94]. The mask alignment mark is usually placed by the foundry to align one mask to the next. Critical dimension test structures can be measured after processing to check proper etching of narrow polysilicon or metal lines. Vernier structures are used to judge the alignment between layers. A vernier is a set of closely spaced parallel lines on two layers. Misalignment between the two layers can be judged by the alignment of the two verniers. Test structures such as chains of contacts and vias, test transistors, and ring oscillators are used to evaluate contact resistance and transistor parameters. Often these structures can be placed along the scribe line so they do not consume useful wafer area.

3.3.3 MOSIS Scalable CMOS Design Rules

Class project designs often use the λ -based scalable CMOS design rules from MOSIS because they are simple and freely available. MOSIS once offered a wide variety of processes, from 2 μm to 180 nm, compatible with the scalable CMOS rules. Indeed, MOSIS also supports three variants of these rules: SCMOS, SUBM, and DEEP, which are progressively more conservative to support feature sizes down to 180 nm. Chips designed in the conservative DEEP rules could be fabricated on any of the MOSIS processes.

As time has passed, the older processes became obsolete and the newer processes have too many nuances to be compatible with scalable design rules. The MOSIS processes most commonly used today are the ON Semiconductor (formerly AMI) 0.5 μm process and the IBM 130, 90, 65, and 45 nm processes.

The 0.5 μm process is popular for university class projects because MOSIS Educational Program offers generous grants to cover fabrication costs for 1.5 mm $\times$ 1.5 mm “TinyChips.” The best design rules for this process are the scalable SUBM rules¹ using $\lambda = 0.3 \mu\text{m}$. Thus, a TinyChip is 5000 $\lambda \times$ 5000 λ . Polysilicon is drawn at 2 $\lambda = 0.6 \mu\text{m}$, then biased by MOSIS by $-0.1 \mu\text{m}$ prior to mask generation to give a true 0.5 μm gate length. When simulating circuits, be sure to use the biased channel lengths to model the transistor behavior accurately. In SPICE, the XL parameter is added to the specified transistor length to find the actual length. Thus, a SPICE deck could specify a drawn channel length of $L = 0.6 \mu\text{m}$ for each transistor and include $XL = -0.1 \mu\text{m}$ in the model file to indicate a biased length of 0.5 μm . There is a tutorial at www.cmosvlsi.com on designing in this process with the Electric CAD tool suite. [Brunvand09] explains how to design in this process with the Cadence and Synopsys tool suites; this flow has a steeper learning curve but better mirrors industry practices.

Credible research chips need more advanced processes to reflect contemporary design challenges. The IBM processes are presently discounted for universities, and MOSIS offers certain research grants as well. The best way to design in these processes is with the Cadence and Synopsys tools using IBM’s proprietary micron-based design rules. The design flow is presently poorly documented by MOSIS and ranges from difficult at the 130 nm node to worse at deeper nodes. Unfortunately, this presently limits access to these processes to highly sophisticated research groups.

¹Technically, MOSIS has two sets of contact rules [MOSIS09]. The standard rules require polysilicon and active to overlap contacts by 1.5 λ . Half-lambda rules reduce productivity because they force the designer off a λ grid. The “alternate contact rules” are preferable because they require overlap by 1 λ , at the expense of more conservative spacing rules; these alternate rules are used in the examples in this text.

Section 1.5.3 introduced the SCMOS design rules. More extensive rules are illustrated and summarized on the inside back cover. Layouts consist of a set of rectangles on various layers such as polysilicon or metal. *Width* is the minimum width of a rectangle on a particular layer. *Spacing* is the minimum spacing between two rectangles on the same or different layers. *Overlap* specifies how much a rectangle must surround another on another layer. Dimensions are all specified in λ except for overglass cuts that do not scale well because they must contact large bond wires or probe tips. Select layers are often generated automatically and thus are not shown in the layout. If the active layer satisfies design rules, the select will too.

Contacts and vias must be exactly $2 \times 2 \lambda$. Larger connections are made from arrays of small vias to prevent current crowding at the periphery. The spacing rules of polysilicon or diffusion to arrays of multiple contacts is slightly larger than that to a single contact.

Section 1.5.5 estimated the pitch of lower-level metal to be 8λ : 4λ for the width and 4λ for spacing. Technically, the minimum width and spacing are 3λ , but the minimum metal contact size is $2 \times 2 \lambda$ plus 1λ surround on each side, for a width of 4λ . Thus, the pitch for contacted metal lines can be reduced to 7λ . Moreover, if the lines are drawn at 3λ and the contacts are staggered so two adjacent lines never have adjacent contacts, the pitch reduces to 6.5λ . Nevertheless, using a pitch of 8λ for planning purposes is good practice and leaves a bit of “wiggle room” to solve difficult layout problems.

3.3.4 Micron Design Rules

Table 3.1 lists a set of micron design rules for a hypothetical 65 nm process representing an amalgamation of several real processes. Rule numbers reference the diagram on the inside back cover. Observe that the rules differ slightly but not immensely from lambda-based rules with $\lambda = 0.035 \mu\text{m}$. A complete set of micron design rules in this generation fills hundreds of pages. Note that upper level metal rules are highly variable depending on the metal thickness; thicker wires require greater widths and spacings and bigger vias.

TABLE 3.1 Micron design rules for 65 nm process

Layer	Rule	Description	65 nm Rule (μm)
Well	1.1	Width	0.5
	1.2	Spacing to well at different potential	0.7
	1.3	Spacing to well at same potential	0.7
Active (diffusion)	2.1	Width	0.10
	2.2	Spacing to active	0.12
	2.3	Source/drain surround by well	0.15
	2.4	Substrate/well contact surround by well	0.15
	2.5	Spacing to active of opposite type	0.25
Poly	3.1	Width	0.065
	3.2	Spacing to poly over field oxide	0.10
	3.2a	Spacing to poly over active	0.10
	3.3	Gate extension beyond active	0.10
	3.4	Active extension beyond poly	0.10
	3.5	Spacing of poly to active	0.07

TABLE 3.1 Micron design rules for 65 nm process (continued)

Layer	Rule	Description	65 nm Rule (μm)
Select	4.1	Spacing from substrate/well contact to gate	0.15
	4.2	Overlap of active	0.12
	4.3	Overlap of substrate/well contact	0.12
	4.4	Spacing to select	0.20
Contact (to poly or active)	5.1, 6.1	Width (exact)	0.08
	5.2b, 6.2b	Overlap by poly or active	0.01
	5.3, 6.3	Spacing to contact	0.10
	5.4	Spacing to gate	0.07
Metal1	7.1	Width	0.09
	7.2	Spacing to well metal1	0.09
	7.3, 8.3	Overlap of contact or via	0.01
	7.4	Spacing to metal for lines wider than $0.5 \mu\text{m}$	0.30
Via1–Via6	8.1, 14.1, ...	Width (exact)	0.10
	8.2, 14.2, ...	Spacing to via on same layer	0.10
Metal2–Metal7	9.1, ...	Width	0.10
	9.2, ...	Spacing to same layer metal	0.10
	9.3, ...	Overlap of via	0.01
	9.4, ...	Spacing to metal for lines wider than $0.5 \mu\text{m}$	0.30
Via7–8		Width	0.20
		Spacing	0.20
Metal8–9		Width	0.40
		Spacing to same layer metal	0.40
		Overlap of via	0.10
		Spacing to metal for lines wider than $1.0 \mu\text{m}$	0.50

3.4 CMOS Process Enhancements

3.4.1 Transistors

3.4.1.1 Multiple Threshold Voltages and Oxide Thicknesses Some processes offer multiple threshold voltages and/or oxide thicknesses. Low-threshold transistors deliver more ON current, but also have greater subthreshold leakage. Providing two or more thresholds permits the designer to use low- V_t devices on critical paths and higher- V_t devices elsewhere to limit leakage power. Multiple masks and implantation steps are used to set the various thresholds. Alternatively, transistors with slightly longer channels can be used; these transistors naturally have higher thresholds because of the short channel effect (see Section 2.4.3.3) [Rohrer05].

Thin gate oxides also permit more ON current. However, they break down when exposed to the high voltages needed in I/O circuits. Oxides thinner than about $15 \text{ \AA}$ also

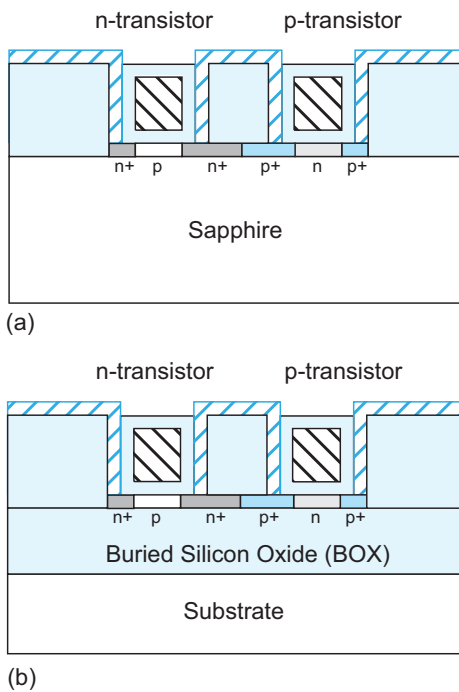


FIGURE 3.17 SOI types

contribute to large gate leakage currents. Many processes offer a second, thicker oxide for the I/O transistors (see Section 13.6). For example, 3.3 V I/O circuits commonly use $0.35\ \mu\text{m}$ channel lengths and 7 nm gate oxides. When gate leakage is a problem and high- k dielectrics are unavailable, an intermediate oxide thickness may also be provided to reduce leakage. Again, multiple masks are used to define the different oxides.

3.4.1.2 Silicon on Insulator A variant of CMOS that has been available for many years is Silicon on Insulator (SOI). As the name suggests, this is a process where the transistors are fabricated on an insulator. SOI stands in contrast to conventional bulk processes in which the transistors are fabricated on a conductive substrate. Two main insulators are used: SiO_2 and sapphire. One major advantage of an insulating substrate is the elimination of the capacitance between the source/drain regions and body, leading to higher-speed devices. Another major advantage is lower subthreshold leakage due to steeper subthreshold slope resulting from a smaller n in EQ (2.44). The drawbacks are time-dependent threshold variations caused by the floating body.

Figure 3.17 shows two common types of SOI. Figure 3.17(a) illustrates a sapphire substrate. In this technology (for example, Peregrine Semiconductor's UltraCMOS), a thin layer of silicon is formed on the sapphire surface. The thin layer of silicon is selectively doped to define different threshold transistors. Gate oxide is grown on top of this and then polysilicon gates are defined. Following this, the nMOS and pMOS transistors are formed by implantation. Figure 3.17(b) shows a silicon-based SOI process. Here, a silicon substrate is used and a *buried oxide* (BOX) is grown on top of the silicon substrate. A thin silicon layer is then grown on top of the buried oxide and this is selectively implanted to form nMOS and pMOS transistor regions. Gate, source, and drain regions are then defined in a similar fashion to a bulk process. Sapphire is optically and RF transparent. As such, it can be of use in optoelectronic areas when merged with III-V based light emitters. SOI devices and circuits are discussed further in Section 9.5.

3.4.1.3 High- k Gate Dielectrics MOS transistors need high gate capacitance to attract charge to the channel. This leads to very thin SiO_2 gate dielectrics (e.g., 10.5–12 Å, merely four atomic layers, in a 65 nm process). Gate leakage increases unacceptably below these thicknesses, which brings an end to classical scaling [Bai04]. Simple SiO_2 has a dielectric constant of $k = 3.9$. As shown in EQ (2.2), gates could use thicker dielectrics and hence leak less if a material with a higher dielectric constant were available.

A first step in this direction was the introduction of nitrogen to form *oxynitride* gate dielectrics, called SiON, around the 130 nm generation, providing k of about 4.1–4.2. High- k dielectrics entered commercial manufacturing in 2007, first with a hafnium-based material in Intel's 45 nm process [Auth08]. Hafnium oxide (HfO_2) has $k = 20$.

A depletion region forms at the interface of polysilicon and the gate dielectric. This effectively increases t_{ox} , which is undesirable for performance. Moreover, polysilicon gates can be incompatible with high- k dielectrics because of effects such as *threshold voltage pinning* and *phonon scattering*, which make it difficult to obtain low thresholds and reduce the mobility. The Intel 45 nm process returned to metal gates to solve these problems and also to reduce gate resistance, as shown in Figure 3.18 [Mistry07]. Thus, the term MOS is

technically accurate again! nMOS and pMOS transistors use different types of metal with different work functions (energy required to free an electron from a solid) to set the threshold voltages. A second lower-resistance metal layer plays a role similar to a silicide.

One of the challenges with metal gates is that they melt if exposed to the high temperature source/drain formation steps. But if the gate were formed after the source and drain, the self-alignment advantage would be lost. Intel sidesteps this conundrum by first building the transistor with a high-k dielectric and a standard polysilicon gate. After the transistor is complete and the interlayer dielectric is formed, the wafer is polished to expose the polysilicon gates and etched to remove the undesired poly. A thin metal gate is deposited in the trench. Different metals with different workfunctions are required for the nMOS and pMOS transistors. Finally, the trench is filled with a thicker layer of Al for low gate resistance, and the wafer is planarized again.

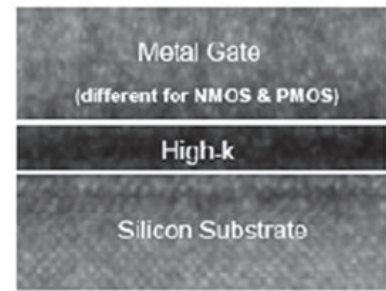


FIGURE 3.18 High-k gate stack TEM (© IEEE 2007.)

3.4.1.4 Higher Mobility Increasing the mobility (μ) of the semiconductor improves drive current and transistor speed. One way to improve the mobility is to introduce mechanical strain in the channel. This is called *strained silicon*.

Figure 3.19 shows strained nMOS and pMOS transistors in the Intel 65 nm process that achieve 40% and 100% higher mobility than unstrained transistors, respectively [Tyagi05, Thompson02, Thompson04]. The nMOS channel is under tensile stress created by an insulating film of silicon nitride (SiN) capping the gate. The pMOS channel is under compressive stress produced by etching a recess into the source and drain, then filling the slot with an epitaxial layer of silicon germanium (SiGe). Germanium is another group IV semiconductor with a larger atomic radius than silicon. When a small fraction of the silicon atoms are replaced by germanium, the lattice retains its shape but undergoes mechanical strain because of the larger atoms. Using separate strain mechanisms for the nMOS and pMOS transistors improves mobility of both electrons and holes. An alternative approach is to implant germanium atoms in the channel, introducing tensile stress that only improves electron mobility. STI also introduces stress that affects mobility, so the diffusion layout can impact performance [Topaloglu07].

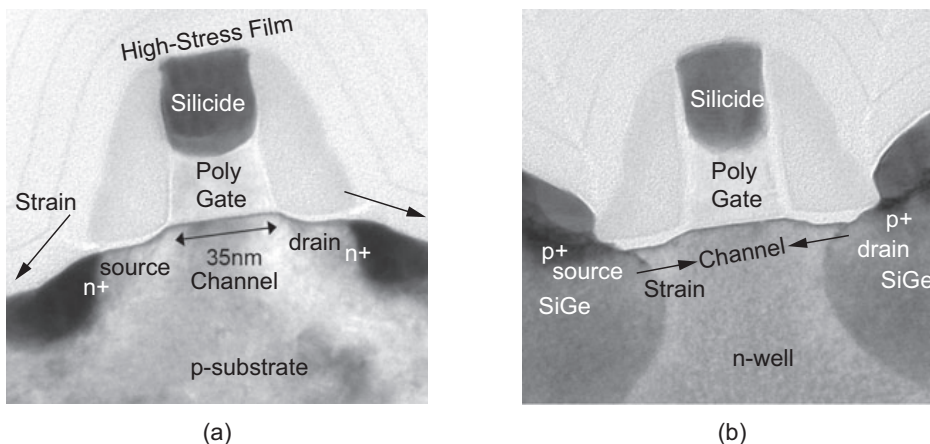


FIGURE 3.19 Strained silicon transistor micrographs: (a) nMOS, (b) pMOS (© IEEE 2005.)

SiGe is also used in high-performance bipolar transistors, especially for radio-frequency (RF) applications. SiGe bipolar devices can be combined with conventional CMOS on the same substrate, which is valuable for low-cost system-on-chip applications that require both digital and RF circuits [Hashimoto02, Harame01a, Harame01b].

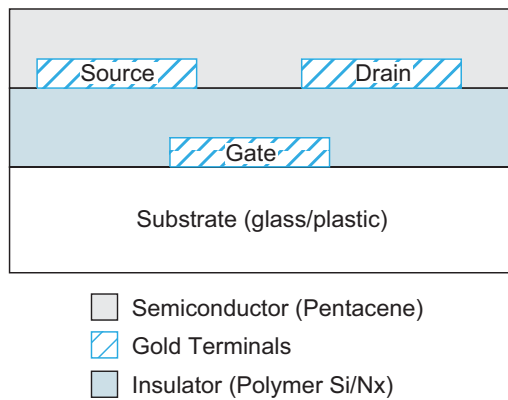


FIGURE 3.20 Plastic transistors

3.4.1.5 Plastic Transistors MOS transistors can be fabricated with organic chemicals. These transistors show promise in active matrix displays, flexible electronic paper, and radio-frequency ID tags because the devices can be manufactured from an inexpensive chemical solution [Huitema03, Myny09]. Figure 3.20 shows the structure of a plastic pMOS transistor. The transistor is built “upside down” with the gold gates and interconnect patterned first on the substrate. Then an organic insulator or silicon nitride is laid down, followed by the gold source and drain connections. Finally, the organic semiconductor (pentacene) is laid down. The mobility of the carriers in the plastic pMOS transistor is about $0.15 \text{ cm}^2/\text{V} \cdot \text{s}$. This is three orders of magnitude lower than that of a comparable silicon device, but is good enough for special applications. Typical lengths and widths are $5 \mu\text{m}$ and $400 \mu\text{m}$, respectively.

3.4.1.6 High-Voltage Transistors High-voltage MOSFETs can also be integrated onto conventional CMOS processes for switching and high-power applications. Gate oxide thickness and channel length have to be larger than usual to prevent breakdown. Specialized process steps are necessary to achieve very high breakdown voltages.

3.4.2 Interconnect

Interconnect has advanced rapidly. While two or three metal layers were once the norm, CMP has enabled inexpensive processes to include seven or more layers. Copper metal and low-k dielectrics are almost universal to reduce the resistance and capacitance of these wires.

3.4.2.1 Copper Damascene Process While aluminum was long the interconnect metal of choice, copper has largely superseded it in nanometer processes. This is primarily due to the higher conductivity of copper compared to aluminum. Some challenges of adopting copper include the following [Merchant01]:

- Copper atoms diffuse into the silicon and dielectrics, destroying transistors.
- The processing required to etch copper wires is tricky.
- Copper oxide forms readily and interferes with good contacts.
- Care has to be taken not to introduce copper into the environment as a pollutant.

Barrier layers have to be used to prevent the copper from entering the silicon surface. A new metallization procedure called the *damascene process* was invented to form this barrier. The process gets its name from the medieval metallurgists of Damascus who crafted fine inlaid swords. In a conventional subtractive aluminum-based metallization step, as we have seen, aluminum is layered on the silicon surface (where vias also have been etched) and then a mask and resist are used to define which areas of metal are to be retained. The unneeded metal is etched away. A dielectric (SiO_2 or other) is then placed over the aluminum conductors and the process can be repeated.

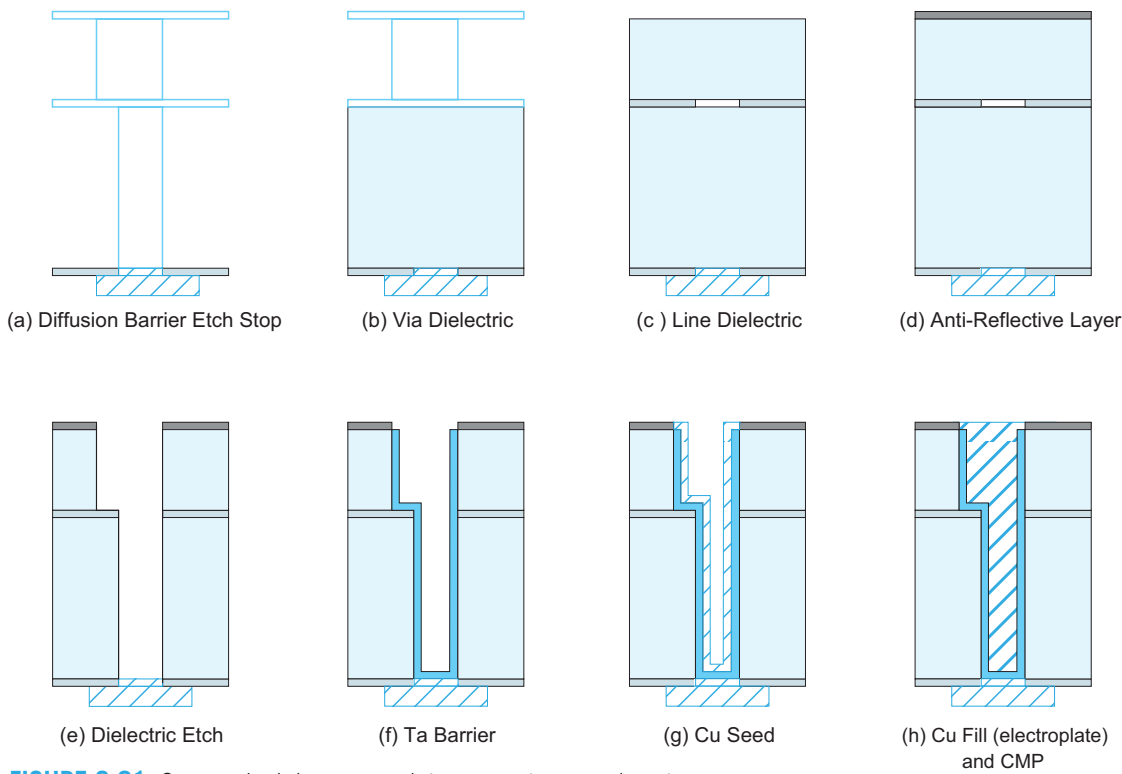


FIGURE 3.21 Copper dual damascene interconnect processing steps

A typical copper damascene process is shown in Figure 3.21, which is an adaptation of a dual damascene process flow from Novellus. Figure 3.21(a) shows a barrier layer over the prior metallization layer. This stops the copper from diffusing into the dielectric and silicon. The via dielectric is then laid down (Figure 3.21(b)). A further barrier layer can then be patterned, and the line dielectric is layered on top of the structure, as shown in Figure 3.21(c). An anti-reflective layer (which helps in the photolithographic process) is added to the top of the sandwich. The two dielectrics are then etched away where the lines and vias are required. A barrier layer such as 10 nm thick Ta or TaN film is then deposited to prevent the copper from diffusing into the dielectrics [Peng02]. As can be seen, a thin layer of the barrier remains at the bottom of the via so the barrier must be conductive. A copper seed layer is then coated over the barrier layer (Figure 3.21(g)). The resulting structure is electroplated full of copper, and finally the structure is ground flat with CMP, as shown in Figure 3.21(h).

3.4.2.2 Low-k Dielectrics SiO_2 has a dielectric constant of $k = 3.9\text{--}4.2$. Low-k dielectrics between wires are attractive because they decrease the wire capacitance [Brown03]. This reduces wire delay, noise, and power consumption. Adding fluorine to the silicon dioxide creates *fluorosilicate glass* (FSG or SiOF) with a dielectric constant of 3.6, widely used in 130 nm processes. Adding carbon to the oxide can reduce the dielectric constant to about 2.8–3; such SiCOH (also called *carbon-doped oxide*, CDO) is commonly used at the 90 and 65 nm generation. Alternatively, porous polymer-based dielectrics can deliver even lower dielectric constants. For example, SiLK, from Dow Chemical, has $k = 2.6$ and may scale to $k = 1.6\text{--}2.2$ by increasing the porosity. IBM has demonstrated air (or vacuum)

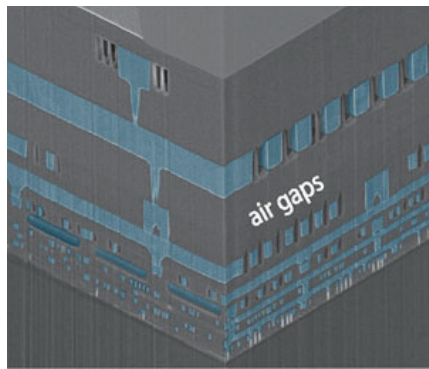


FIGURE 3.22 Micrograph showing air gap insulation between copper wires (Courtesy of International Business Machines Corporation. Unauthorized use not permitted.)

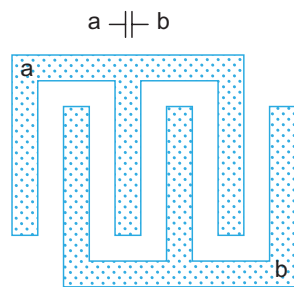


FIGURE 3.23
Fringe capacitor

gaps, which have $k = 1$ where the dielectric has been eliminated entirely, as shown in Figure 3.22. Developing low- k dielectrics that can withstand the high temperatures during processing and the forces applied during CMP is a major challenge.

3.4.3 Circuit Elements

While CMOS transistors provide for almost complete digital functionality, the use of CMOS technology as the mixed signal and RF process of choice has driven the addition of special process options to enhance the performance of circuit elements required for these purposes.

3.4.3.1 Capacitors In a conventional CMOS process, a capacitor can be constructed using the gate and source/drain of an MOS transistor, a diffusion area (to ground or V_{DD}), or a parallel metal plate capacitor (using stacked metal layers). The MOS capacitor has good capacitance per area but is relatively nonlinear if operated over large voltage ranges. The diffusion capacitor cannot be used for a floating capacitor (where neither terminal is connected to ground). The metal parallel plate capacitor has low capacitance per area. Normally, the aim in using a floating capacitor is to have the highest ratio of desired capacitance value to stray capacitance (to ground normally). The bottom metal plate contributes stray capacitance to ground.

Analog circuits frequently require capacitors in the range of 1 to 10 pF. The first method for doing this was to add a second polysilicon layer so that a *poly-insulator-poly* (PIP) capacitor could be constructed. A thin oxide was placed between the two polysilicon layers to achieve capacitance of approximately $1 \text{ fF}/\mu\text{m}^2$.

The most common capacitor used in CMOS processes today is a fringe capacitor, which consists of interdigitated fingers of metal, as shown in Figure 3.23. Multiple layers can be stacked to increase the capacitance per area.

3.4.3.2 Resistors In unaugmented processes, resistors can be built from any layer, with the final resistance depending on the resistivity of the layer. Building large resistances in a small area requires layers with high resistivity, particularly polysilicon, diffusion, and n-wells. Diffusion has a large parasitic capacitance to ground, making it unsuitable for high-frequency applications. Polysilicon gates are usually silicided to have low resistivity. The fix for this is to allow for *undoped* high-resistivity polysilicon. This is specified with a *silicide block* mask where high-value poly resistors are required. The resistivity can be tuned to around $300\text{--}1000 \text{ } \Omega/\text{square}$, depending on doping levels. Another material used for precision resistors is nichrome, although this requires a special processing step.

A typical resistor layout is shown in Figure 3.24. This geometry is sometimes called a *meander* structure. A number of *unit* resistors have been used so that a variety of matched resistor values can be constructed. For instance, if $20 \text{ k}\Omega$, and $15 \text{ k}\Omega$ resistors were required, a unit value of $5 \text{ k}\Omega$ could be used. Then three resistors (as shown) would construct a $15 \text{ k}\Omega$ resistor. The two resistors at the ends are called *dummy resistors* or *fingers*. They perform no circuit function, but replicate the proximity effects (such as etch and implant) that the interior resistors see during processing. This helps ensure that all resistors are matched.

The various resistor options have temperature and voltage coefficients. Foundry design manuals normally include these values.

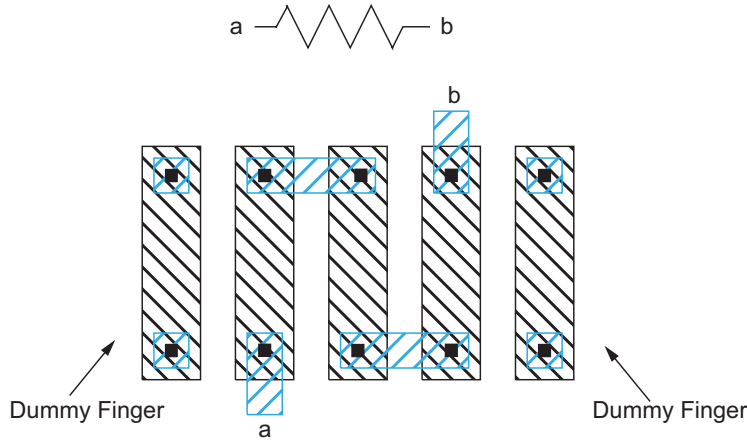


FIGURE 3.24 Resistor layout

3.4.3.3 Inductors The desire to integrate inductors on chips has increased radically with the upsurge in interest in RF circuits. The most common monolithic inductor is the *spiral* inductor, which is a spiral of upper-level metal. A typical inductor is shown in Figure 3.25(a). As the process is planar, an underpass connection has to be made to complete the inductor. A typical equivalent model is shown in Figure 3.25(b). In addition to the required inductance L , there are several parasitic components. R_s is the series resistance of the metal (and contacts) used to form the inductor. C_p is the parallel capacitance to ground due to the area of the metal wires forming the inductor. C_s is the shunt capacitance of the underpass. Finally, R_p is an element that models the loss incurred in the resistive substrate.

Usually, when considering an inductor, the parameters of interest to a designer are its inductance, the Q of the inductor, and the self-resonant frequency. High Q s are sought to create low phase-noise oscillators, narrow filters, and low-loss circuits in general. Q values for typical planar inductors on a bulk process range from 5 to 10.

The number of turns n required to achieve some inductance L if the wire pitch, in turns per meter, is $P = 1/2(W + S)$, is [Lee98]

$$n \approx \sqrt[3]{\frac{LP}{\mu_0}} \quad (3.4)$$

where $\mu_0 = 1.2 \times 10^{-6}$ H/m is the permeability of free space. Figure 3.25 has $n = 1.75$ turns. Higher-quality inductors can also be manufactured using bond wires between I/O pads. The inductance of a wire of length l and radius r is approximately

$$L \approx \frac{\mu_0 l}{2\pi} \left[\ln \frac{2l}{r} - 0.75 \right] \quad (3.5)$$

or about 1 nH/mm for standard 1 mil (25 μ m) bond wires.

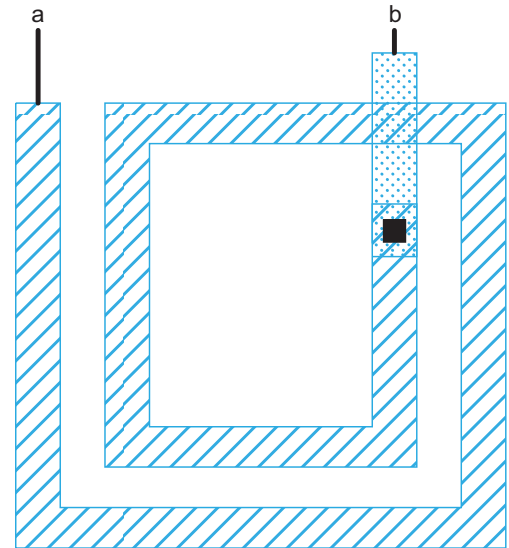
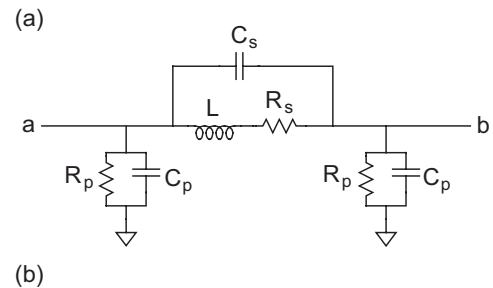


FIGURE 3.25 Typical spiral inductor and equivalent circuit [Rotella02]

Reduction in Q occurs because of the resistive loss in the conductors used to build the inductor (R_s), and the eddy current loss in the resistive silicon substrate (R_p). In an effort to increase Q , designers have resorted to removing the substrate below the inductor using MEMS techniques [Yoon02]. The easiest way to improve the Q of monolithic inductors is to increase the thickness of the top-level metal. The Q can also be improved by using a patterned ground shield in polysilicon under the inductor to decrease substrate losses.

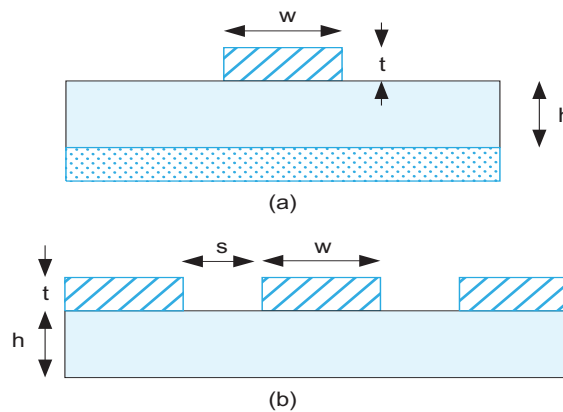


FIGURE 3.26 Microstrip and coplanar waveguide

3.4.3.4 Transmission Lines A *transmission line* can be used on a chip to provide a known impedance wire. Two basic kinds of transmission lines are commonly used: *microstrips* and *coplanar waveguides*.

A microstrip transmission line, as shown in Figure 3.26(a), is composed of a wire of width w and thickness t placed over a ground plane and separated by a dielectric of height h and dielectric constant k . In the chip case, the wire might be the top level of metallization and the ground plane the next metal layer down.

A coplanar waveguide does not require a sublayer ground plane and is shown in Figure 3.26(b). It consists of a wire of width w spaced s on each side from *coplanar* ground wires. The reader is referred to [Wadell91] for detailed design equations.

3.4.3.5 Bipolar Transistors Bipolar transistors were mentioned previously in our discussion of SiGe process options. Both *nnp* and *pnp* bipolar transistors can be added to a CMOS process, which is then called a BiCMOS process. These processes tend to be used for specialized analog or high-voltage circuits. In a regular n-well process, a parasitic vertical pnp transistor is present that can be used for circuits such as bandgap voltage references. This transistor is shown in Figure 3.27 with the p-substrate collector, the n-well base, and the p-diffusion emitter. Both process cross-section and layout are shown. This transistor, in conjunction with a parasitic npn, is the cause of latchup (see Section 7.3.6).

3.4.3.6 Embedded DRAM Dynamic RAM (DRAM) uses a single transistor and a capacitor to store a bit of information. It is about five times denser than static RAM (SRAM) conventionally used on CMOS logic chips, so it can reduce the size of a chip containing large amounts of memory. DRAM was conventionally manufactured on specialized processes that produced low-performance logic transistors. DRAM requires specialized structures to build capacitors in a small area. One common structure is a *trench*, which is etched

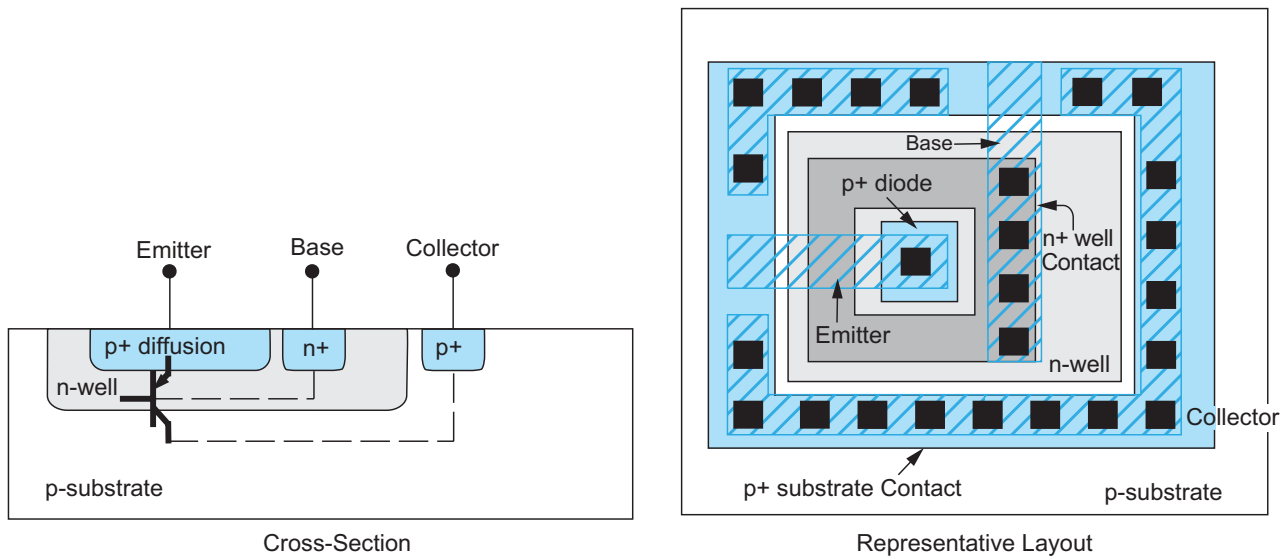


FIGURE 3.27 Vertical pnp bipolar transistor

down into the substrate. Some recent processes have introduced compact capacitor structures for building *embedded DRAM* alongside high-performance logic. Section 12.3 discusses DRAM in more depth.

3.4.3.7 Non-Volatile Memory *Non-volatile* memory (NVM) retains its state when the power is removed from the circuit. The simplest NVM is a *mask-programmed* ROM cell (see Section 12.4). This type of NVM is not reprogrammable or programmable after the device is manufactured. A *one-time programmable* (OTP) memory can be implemented using a fuse constructed of a thin piece of metal through which is passed a current that vaporizes the metal by exceeding the current density in the wire. The first reprogrammable memories used a stacked polysilicon gate structure and were programmed by applying a high voltage to the device in a manner that caused Fowler-Nordheim tunneling to store a charge on a floating gate. The whole memory could be erased by exposing it to UV light that knocked the charge off the gate. These memories evolved to *electrically erasable* memories, which are today represented by *Flash* memory.

A typical Flash memory transistor is shown in Figure 3.28 [She02]. The source and drain structures can vary considerably to allow for high-voltage operation, but the dual-gate structure is fairly common. The gate structure is a stacked configuration commencing with a thin tunnel oxide or nitride. A floating polysilicon gate sits on top of this oxide and a conventional gate oxide is placed on top of the floating gate. Finally, a polysilicon control gate is placed on top of the gate oxide. The operation of the cell is also shown in Figure

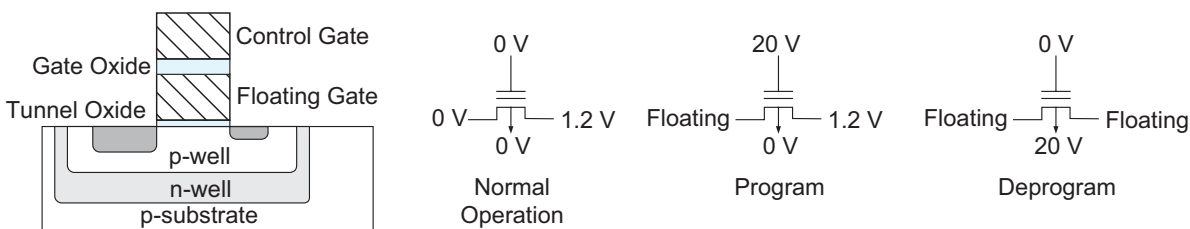
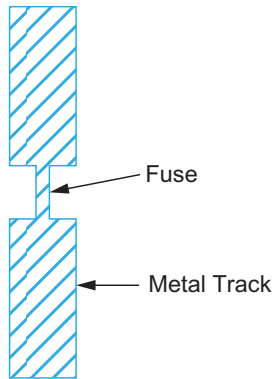


FIGURE 3.28 Flash memory construction and operation

**FIGURE 3.29**

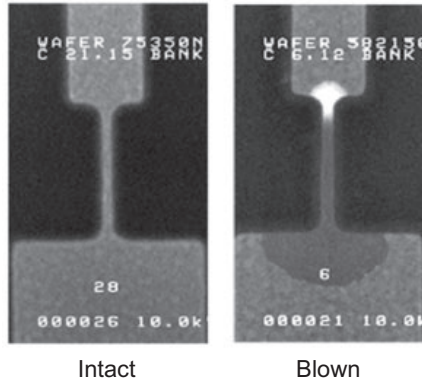
A typical metal fuse

3.28. In normal operation, the floating gate determines whether or not the transistor is conducting. To program the cell, the source is left floating and the control gate is raised to approximately 20 V (using an on-chip voltage multiplier). This causes electrons to tunnel into the floating gate, and thus program it. To deprogram a cell, the drain and source are left floating and the substrate (or well) is connected to 20 V. The electrons stored on the floating gate tunnel away, leaving the gate in an unprogrammed state.

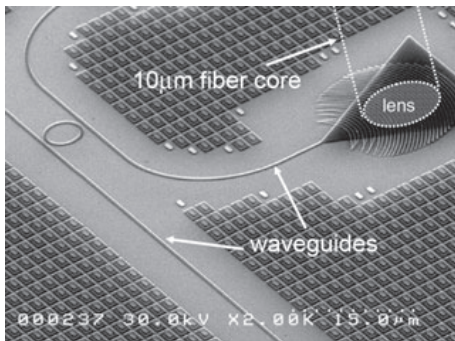
3.4.3.8 Fuses and Antifuses During manufacturing, fuses can be blown with a high current or zapped by a laser. In the latter case, an area is normally left in the passivation oxide to allow the laser direct access to the metal link that is to be cut. Figure 3.29 shows the layout of a metal fuse.

Laser-blown fuses are large and the blow process can damage adjacent devices. *Electronic fuses* are structures whose characteristics can be nondestructively altered by applying a high current. For example, IBM eFUSEs are narrow polysilicon wires silicided with cobalt. The resistance is initially about 200 Ω . If a programming current of 10–15 mA is applied for 200 μ s, the cobalt will migrate to the anode, as shown in Figure 3.30. This raises the resistance by an order of magnitude. Simple sense circuits are used to detect the state of the eFUSE. IBM uses fuses for chip serial numbers, thermal sensor calibration, and to reconfigure defective components [Rohrer05, Rizzolo07].

An *antifuse* is a similar device that initially has a high resistivity but can become low resistance when a programming voltage is applied. This device requires special processing and is used in programmable logic devices (see Section 12.7).

**FIGURE 3.30** eFUSE (© IEEE 2005.)

3.4.3.9 Microelectromechanical Systems (MEMS) Semiconductor processes and especially CMOS processes have been used to construct tiny mechanical systems monolithically. A typical device is the well-known air-bag sensor, which is a small accelerometer consisting of an air bridge capacitor that can detect sudden changes in acceleration when co-integrated with some conditioning electronics. MEMS micromirrors on torsional hinges are used in inexpensive, high-resolution digital light projectors. Structures such as cantilevers, mechanical resonators, and even micromotors have been built. A full discussion of MEMS is beyond the scope of this book, but further material can be found in texts such as [Maluf04].

**FIGURE 3.31** Optical waveguide and holographic lens integrated with a 130 nm CMOS process (© IEEE 2006.)

3.4.3.10 Integrated Photonics Although silicon is opaque at visible wavelengths, it is transparent in the infrared range used in optical fibers. Semiconductor photonic components are rapidly evolving. Components compatible with a conventional CMOS process include waveguides, modulators, and photodetectors [Salib04, Young10]. A key missing component is an optical source, such as a laser. However, just as V_{DD} is generated off-chip from a DC power supply, light can be generated off-chip and brought onto the chip through an optical fiber. Figure 3.31 shows a holographic lens used to couple an optical fiber to an on-chip waveguide [Huang06]. Integrated photonics shows particular promise for optical transceivers to replace copper wires in high-speed networks.

3.4.3.11 Three-Dimensional Integrated Circuits 3D ICs contain multiple layers of devices. Stacking transistors in layers can reduce wire lengths, improving speed and power. It also can permit heterogeneous technologies to be combined in one package; for example, logic, memory, and analog/RF chips can be stacked into one package.

IBM has described a process in which 200 mm wafers are ground down to a remarkable $20\text{ }\mu\text{m}$ thickness after fabrication [Topol06]. They are aligned to $1\text{ }\mu\text{m}$ tolerance, and then one is bonded on top of another using oxide-fusion or copper bonding. Tall skinny through-silicon vias (TSVs) between the wafers are etched and metallized; the aspect ratio of the vias and the thickness of the wafers sets the density of contacts between wafers. Densities of 10^4 TSVs/ mm^2 or more can presently be achieved. Some of the challenges in 3D integration include wafer bowing, testing layers before they are bonded, and managing cooling and power delivery. Figure 3.32 shows two wafers bonded together. The bottom wafer has four levels of metal and the top wafer has two levels. The $8\text{-}\mu\text{m}$ wide landing pad on the top metal layer of the bottom wafer provides tolerances for misalignment with the 3D vias protruding from beneath the top wafer.

3D ICs are starting to move from research into production [Emma08]. An initial application is to stack multiple memory chips to provide a higher capacity in a standard form factor.

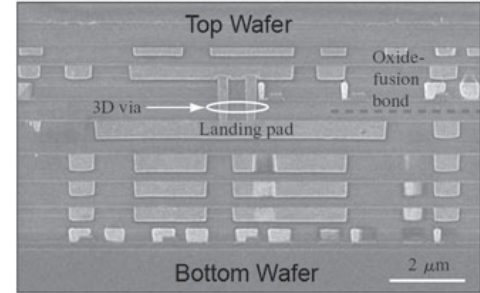


FIGURE 3.32 Scanning electron micrograph of 3-dimensional integration of two wafers (Reprinted from [Koester08]. Courtesy of International Business Machines Corporation. Unauthorized use not permitted.)

3.4.4 Beyond Conventional CMOS

A major problem with scaling bulk transistors is the subthreshold leakage from drain to source caused by the inability of the gate to turn off the channel completely. This can be improved by a gate structure where the gate is placed on two, three, or four sides of the channel to gain better control over the charge in the channel. A promising structure solves the problem by forming a vertical channel and constructing the gate in a pincer-like arrangement around three sides. These devices have been given the generic name “*finfets*” because the source/drain regions form fins on the silicon surface [Hisamoto98]. Figure 3.33(a) shows a 3D view of a finfet, while Figure 3.33(b) shows the cross-section and Figure 3.33(c) shows the top view. The gate wraps around three sides of the vertical

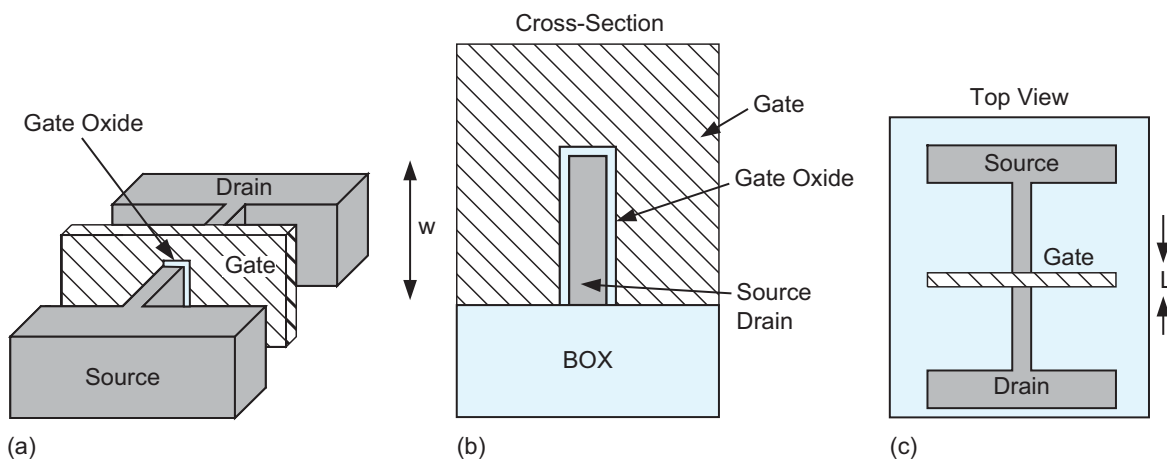


FIGURE 3.33 Finfet structure

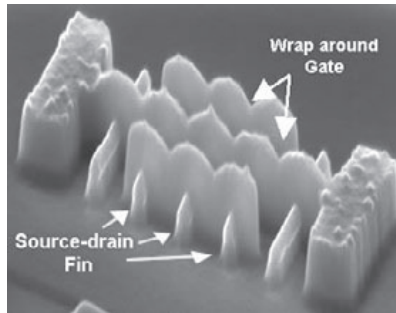


FIGURE 3.34 Trigate transistor (Reprinted with permission of Intel Corporation.)

source/drain fins. The width of the device is defined by the height of the fin, so wide devices are constructed by paralleling fins. Figure 3.34 shows a micrograph of a prototype finfet that Intel calls a *trigate transistor* [Kavalleros06].

Compounds from groups III and V of the periodic table, such as GaAs, offer electron mobilities up to 30 times higher than silicon. Such III-V materials have been research topics for decades. GaAs was once used for very high frequency applications, but has largely been replaced by advanced CMOS processes. However, III-V materials might be integrated into CMOS some day in the future.

Nanotechnology is presently a hot research area seeking alternative structures to replace CMOS when conventional scaling finally runs out of steam. Little obvious progress in radical new device structures has been made since the previous edition of the book, but conventional sub-100 nm CMOS transistors are now being called nanotechnology! Alternative technologies have a large hurdle to overcome competing with the hundreds of billions of dollars that have been invested in advancing CMOS over four decades.

Carbon nanotubes are one nanotechnology that have been used to demonstrate transistor behavior and build inverters [Liu01]. Nanotubes are cylinders with a diameter of a few nanometers. They are of interest because the nanotube is smaller than the predicted endpoint for CMOS gate lengths, and because the nanotubes offer high mobility. A theoretical nanotube transistor is shown in Figure 3.35 [Wong03]. Presently, the speeds are quite slow and the manufacturing techniques are limited, but they may be of interest in the future [Raychowdhury07, Patil09].

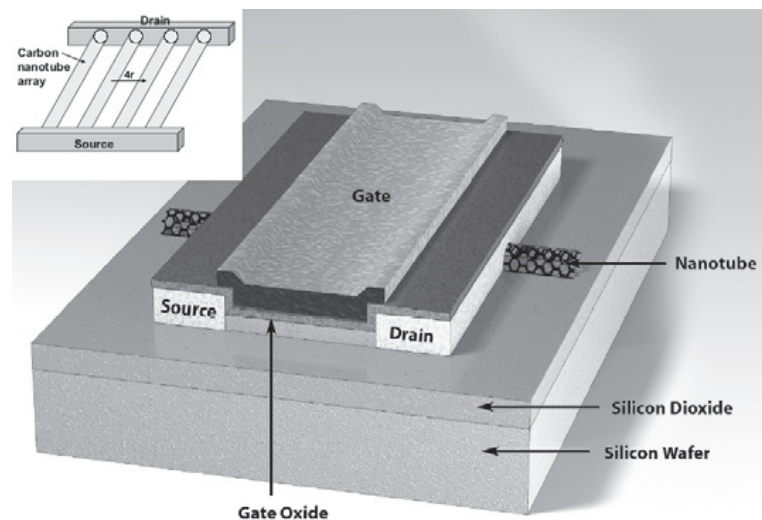


FIGURE 3.35 Carbon nanotube transistor (© IEEE 2003.)

3.5 Technology-Related CAD Issues

The mask database is the interface between the semiconductor manufacturer and the chip designer. Two basic checks have to be completed to ensure that this description can be turned into a working chip. First, the specified geometric design rules must be obeyed. Second, the interrelationship of the masks must, upon passing through the manufacturing

process, produce the correct interconnected set of circuit elements. To check these two requirements, two basic CAD tools are required: a *Design Rule Check* (DRC) program and a mask circuit *extraction* program. The most common approach to implementing these tools is a set of subprograms that perform general geometry operations. A particular set of DRC rules or extraction rules for a given CMOS process (or any semiconductor process) defines the operations that must be performed on each mask and the inter-mask checks that must be completed. Accompanied by a written description, these *run sets* are usually the defining specification for a process.

In this section, we will examine a hypothetical DRC and extraction system to illustrate the nature of these run sets.

3.5.1 Design Rule Checking (DRC)

Although we can design the physical layout in a certain set of mask layers, the actual masks used in fabrication can be derived from the original specification. Similarly, when we want a program to determine what we have designed by examining the interrelationship of the various mask layers, it may be necessary to determine various logical combinations between masks.

To examine these concepts, let us posit the existence of the following functions (loosely based on the Cadence DRACULA DRC program), which we will apply to a geometric database (i.e., rectangles, polygons, and paths):

AND layer1 layer2 -> layer3

ANDs layer1 and layer2 together to produce layer3
(i.e., the intersection of the two input mask descriptions)

OR layer1 layer2 -> layer3

ORs layer1 and layer2 together to produce layer3
(i.e., the union of the two input mask descriptions)

NOT layer1 layer2 -> layer3

Subtracts layer2 from layer1 to produce layer3
(i.e., the difference of the two input mask descriptions)

WIDTH layer > dimension -> layer3

Checks that all geometry on layer is larger than dimension
Any geometry that is not is placed in layer3

SPACE layer > dimension -> layer3

Checks that all geometry on layer is spaced further than dimension
Any geometry that is not is placed in layer3

The following layers will be assumed as input:

```
nwell
active
p-select
n-select
poly
poly-contact
active-contact
metal
```


Typically, useful sublayers are generated initially. First, the four kinds of active area are isolated. The rule set to accomplish this is as follows:

```
NOT   all nwell -> substrate
AND   nwell active -> nwell-active
NOT   active nwell -> pwell-active
AND   nwell-active p-select -> pdiff
AND   nwell-active n-select -> vddn
AND   pwell-active n-select -> ndiff
AND   pwell-active p-select -> gndp
```

In the above specification, a number of new layers have been designated. For instance, the first rule states that wherever nwell is absent, a layer called substrate exists. The second rule states that all active areas within the nwell are nwell-active. A combination of nwell-active and p-select or n-select yields pdiff (p diffusion) or vddn (well tap).

To find the transistors, the following rule set is used:

```
AND   poly ndiff -> ngates
AND   poly pdiff -> pgates
```

The first rule states that the combination of poly and ndiff yields the ngates region—all of the n-transistor gates.

Typical design rule checks (DRC) might include the following:

```
WIDTH metal < 0.13 -> metal-width-error
SPACE metal < 0.13 -> metal-space-error
```

For instance, the first rule determines if any metal is narrower than 0.13 μm and places the errors in the metal-width-error layer. This layer might be interactively displayed to highlight the errors.

3.5.2 Circuit Extraction

Now imagine that we want to determine the electrical connectivity of a mask database. The following commands are required:

CONNECT layer1 layer2

Electrically connect layer1 and layer2.

MOS name drain-layer gate-layer source-layer substrate-layer

Define an MOS transistor in terms of the component terminal layers. (This is, admittedly, a little bit of magic.)

The connections between layers can be specified as follows:

```
CONNECT active-contact pdiff
CONNECT active-contact ndiff
CONNECT active-contact vddn
CONNECT active-contact gndp
CONNECT active-contact metal
CONNECT gndp substrate
CONNECT vddn nwell
CONNECT poly-contact poly
CONNECT poly-contact metal
```

The connections between the diffusions and metal are specified by the first seven statements. The last two statements specify how metal is connected to poly.

Finally, the active devices are specified in terms of the layers that we have derived:

```
MOS nmos ndiff ngates ndiff substrate
MOS pmos pdiff pgates pdiff nwell
```

An output statement might then be used to output the extracted transistors in some netlist format (i.e., SPICE format). The extracted netlist is often used to compare the layout against the intended schematic.

It is important to realize that the above run set is manually generated. The data you extract from such a program is only as good as the input. For instance, if parasitic routing capacitances are required, then each layer interaction must be coded. If parasitic resistance is important in determining circuit performance, it also must be specifically included in the extraction run set.

3.6 Manufacturing Issues

As processes have evolved, various rules and guidelines have emerged that reflect the complexity of the processing. These rules are often called *Design for Manufacturability* (DFM).

3.6.1 Antenna Rules

When a metal wire contacted to a transistor gate is plasma-etched, it can charge up to a voltage sufficient to zap the thin gate oxides. This is called *plasma-induced gate-oxide damage*, or simply the *antenna effect*. It can increase the gate leakage, change the threshold voltage, and reduce the life expectancy of a transistor. Longer wires accumulate more charge and are more likely to damage the gates.

During the high-temperature plasma etch process, the diodes formed by source and drain diffusions can conduct significant amounts of current. These diodes bleed off charge from wires before gate oxide is damaged.

Antenna rules specify the maximum area of metal that can be connected to a gate without a source or drain to act as a discharge element. Larger gates can withstand more charge buildup. The design rules normally define the maximum ratio of metal area to gate area such that charge on the metal will not damage the gate. The ratios can vary from 100:1 to 5000:1 depending on the thickness of the gate oxide (and hence breakdown voltage) of the transistor in question. Higher ratios apply to thicker gate oxide transistors (i.e., 3.3 V I/O transistors).

Figure 3.36 shows an antenna rule violation and two ways to fix it. In Figure 3.36(a), a long metal1 line is connected to a transistor gate. It has no connection to diffusion until metal2 is formed, so the gate may be damaged during the metal1 plasma etch. In Figure 3.36(b), the metal1 line is interrupted with a jumper to metal2. This reduces the amount of charge that could zap the gate during the metal1 etch and solves the problem. In Figure 3.36(c), an antenna diode is added, providing a discharge path during the etch. The diode is reverse-biased during normal operation and thus does not disturb circuit function (except for the area and capacitance that it contributes). Note that the problem could also have been solved by making the gate wider.

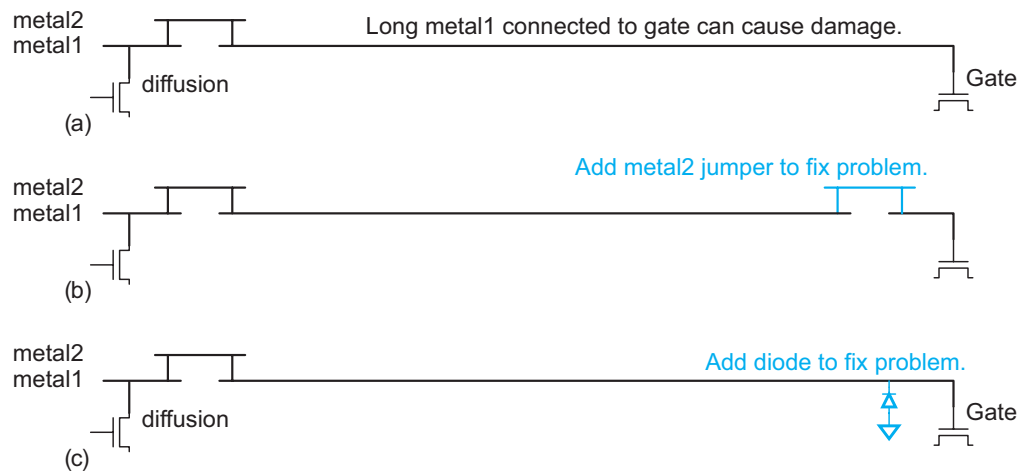


FIGURE 3.36 Antenna rule violation and fixes

For circuits requiring good matching, such as analog and memory cells, transistor gates should connect directly to diffusion with a short segment of metal1 to avoid gate damage that could introduce mismatches.

3.6.2 Layer Density Rules

Another set of rules that pertain to advanced processes are layer density rules, which specify a minimum and maximum density of a particular layer within a specified area. Etch rates have some sensitivity to the amount of material that must be removed. For example, if polysilicon density were too high or too low, transistor gates might end up over- or under-etched, resulting in channel-length variations. Similarly, the CMP process may cause dishing (excessive removal) of copper when the density is not uniform.

To prevent these issues, a metal layer might be required to have 30% minimum and 70% maximum density within a $100\text{ }\mu\text{m}$ by $100\text{ }\mu\text{m}$ area. For digital circuits, these density levels are normally reached with routine routing unless empty spaces exist. Analog and RF circuits, on the other hand, are almost by definition sparse. Thus, diffusion, polysilicon, and metal layers may have to be added manually or by a fill program after design has been completed. The fill can be grounded or left floating. Floating fill contributes lower total capacitance but more coupling capacitance to nearby wires. Grounded fill requires routing the ground net to the fill structures. Clever fill patterns such as staggered rectangles, plus-sign patterns, or diamonds result in lower and more predictable capacitance than do simple geometrical grids [Kahng08]. Designers must be aware of the fill so that it does not introduce unexpected parasitic capacitance to nearby wires.

3.6.3 Resolution Enhancement Rules

Some resolution enhancement techniques impose further design rules. For example, polysilicon typically uses the narrowest lines and thus needs the most enhancement. This can be simplest if polysilicon gates are only drawn in a single orientation (horizontal or vertical). Using a single orientation also reduces systematic process variability. Avoid small jogs and notches (those less than the minimum layer width), because such notches can interfere with proper OPC analysis.

The design community is presently debating a move toward *restrictive design rules* to facilitate RET and reduce manufacturing variability by limiting designers to a smaller set of uniform layout features. These rules might come at the expense of greater area. For example, Intel introduced restrictive design rules for polysilicon in the 45 nm process to control variation and facilitate 193 nm double-patterning lithography [Webb08]. Under these rules, polysilicon is limited to one pitch and direction in layout. This also simplified contact and metal1 rules: the contact pitch is the same as the gate pitch, and metal1 parallel to the gates also has the same pitch. Wide poly pads for contacts and orthogonal polysilicon routing were eliminated by introducing a *trench contact* suitable for local interconnect. Intel found that the restrictive rules did not impact standard cell density and that excellent yield is achieved.

3.6.4 Metal Slotting Rules

Some processes have special rules requiring that wide (e.g. $> 10\text{--}40\text{ }\mu\text{m}$) metal wires have slots. Slots are long slits, on the order of $3\text{ }\mu\text{m}$ wide, in the wire running parallel to the direction of current flow, as shown in Figure 3.37. They provide stress relief, help keep the wire in place, and reduce the risk of electromigration failure (see Section 7.3.3.1). Design rules vary widely between manufacturers.

3.6.5 Yield Enhancement Guidelines

To improve yield, some processes recommend increasing certain widths and spacings where they do not impact area or performance. For example, increasing the polysilicon gate extension slightly reduces the risk of transistor failures from poly/diffusion mask misalignment. Increasing space between metal lines where possible reduces the risk of shorts and also reduces wire capacitance. Other good practices to improve yield include the following:

- Space out wires to reduce risk of short circuits and reduce capacitance.
- Use non-minimum-width wires to reduce risk of open circuits and to reduce resistance.
- Use at least two vias for every connection to avoid open circuits if one via is malformed, and to reduce electromigration wearout.
- Surround contacts and vias by landing pads with more than the minimum overlap to reduce resistance variation and open circuits caused by misaligned contacts.
- Use wider-than-minimum transistors; minimum-width transistors are subject to greater variability and tend not to perform as well.
- Avoid non-rectangular shapes such as 45-degree angles and circles. For specialized circuits such as RAMs that strongly benefit from 45-degree angles, verify masks after optical proximity correction analysis.
- Place dummy transistors or cells at the edge of arrays and sensitive circuits to improve uniformity and matching.
- If it looks nice, it will work better.

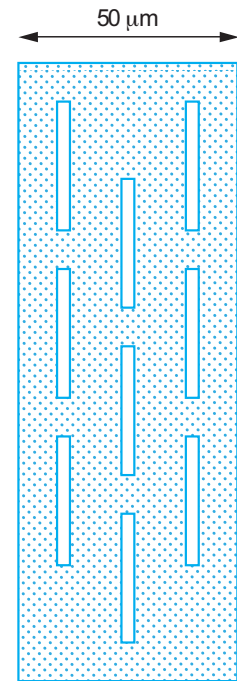


FIGURE 3.37 Slots in wide metal power bus

3.7 Pitfalls and Fallacies

Targeting a bleeding-edge process

There is a fine balance when you are deciding whether or not to move to a new process for a new design. On the one hand, you are tempted by increased density and speed. On the other hand, support for the new process can initially be expensive (becoming familiar with process rules, CAD tool scripts, porting analog and RF designs, locating logic libraries, etc.). In addition, CMOS foundries frequently tune their processes in the first few months of production, and often yield improvement steps can reflect back to design rule changes that impact designs late in their tapeout schedule. For this reason, it is frequently prudent not to jump immediately into a new process when it becomes available. On the other hand, if you are limited in speed or some other attribute that is solved by the new process, then you don't have much choice but to bite the bullet.

Using lambda design rules on commercial designs

Lambda rules have been used in this text for ease of explanation and consistency. They are usable for class designs. However, they are not very useful for production designs for deep sub-micron processes. Of particular concern are the metal width and spacing rules, which are too conservative for most production processes.

Failing to account for the parasitic effects of metal fill

With area density rules, particularly in metal, most design flows include an automatic fill step to achieve the correct metal density. Particularly in analog and RF circuits, it is important to either exclude the automatic fill operation from that area or check circuit performance after the fill by completing a full parasitic extract and rerunning the verification simulation scripts.

Failing to include process calibration test structures

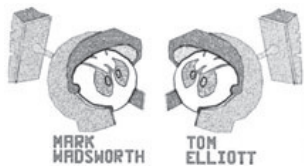
In the discussion on scribe line structures, it was mentioned that test structures are frequently inserted here by the silicon manufacturer. Documentation is often unavailable, so it is prudent for designers (particularly in academic designs, which receive less support from a foundry) to include their own test structures such as transistors or ring oscillators. This allows designers to calibrate the silicon against simulation models.

Waiving design rules

Sometimes it is tempting to ignore a design rule when you are certain it does not apply. For example, consider two wires separated by only 2λ . This violates a design rule because the wires might short together during manufacturing. If the wires are actually connected elsewhere, one might ignore the rule because further shorting is harmless. However, it is possible that the "antifeature" between the wires would produce a narrow strip of photoresist that could break off and float around during manufacturing, damaging some other structure. Moreover, even if the rule violation is safe, keeping track of all the legitimate exceptions is too much work, especially on a large design. It is better to simply fix the design rule error.

Placing cute logos on a chip

Designers have a tradition of hiding their initials on the chip or embedding cute logos in an unused corner of the die. Some automatic wafer inspection tools find that the logos look more like a spec of dust than a legitimate chip structure and mark all of the chips as defective! Some companies now ban the inclusion of layout that is not essential to the operation of the device. Others require placing the logo in the corner of the chip and covering it with a special pseudolayer called LOGO to tell RET and wafer inspection tools to ignore the logo.



Logos on the image sensor for the Spirit Mars rover (Reprinted from Molecular Expressions Silicon Zoo, micro.magnet.fsu.edu/creatures, with permission of Michael Davidson.)

Process development has become so expensive that IBM has formed the Common Platform alliance with partners including Chartered Semiconductor, Samsung, Infineon, and STMicro, to share the R&D costs. IBM offers both SOI and bulk processes; Table 3.2 focuses on their SOI devices that have better $I_{\text{dsat}} / I_{\text{off}}$ ratios. All of the processes in this table are considered *high-performance processes* that focus on a high I_{dsat} . Many manufacturers also offer *low-power processes* with higher threshold voltages and thicker oxides to reduce leakage, especially in battery-powered communications devices.

TABLE 3.2 CMOS process characteristics

Manufacturer		Intel							IBM			
Feature Size f	nm	250	180	130	90	65	45	32	130	90	65	45
Reference		[Bohr96]	[Yang98]	[Tyagi00]	[Thompson02]	[Bai04]	[Mistry07]	[Natarajan08]	[Sleight01]	[Khare02]	[Lee05]	[Narasimha06]
V_{DD}	V	1.8	1.5	1.3	1.2	1.2	1	1	1.2	1	1	1
L_{gate}	nm	180	140	70	50	35	35	30	60	45	40	35
t_{ox}	nm	4.1	3	1.5	1.2	1.2	1	0.9	2.3	1.85	1.05	1.15
$I_{\text{dsat}-n}$	$\mu\text{A}/\mu\text{m}$	700	940	1170	1449	1460	1360	1550	915	1000	1137	1140
$I_{\text{dsat}-p}$	$\mu\text{A}/\mu\text{m}$	320	420	600	725	880	1070	1210	520	480	700	800
I_{off}	$\text{nA}/\mu\text{m}$	1	3	100	400	100	100	100	100	200	200	200
Strain		no	no	no	yes	yes	yes	yes	no	no	yes	yes
High-k Gates		no	no	no	no	no	yes	yes	no	no	no	no
Gate Pitch	nm	640	480	336	260	220	160	112.5	325	245		190
Metal1 Pitch	nm	640	500	350	220	210	150	112.5	350	245		140
Metal Layers		5	6	6	7	8	9	9	8	10	10	10
Material		Al	Al	Cu	Cu	Cu	Cu	Cu	Cu	Cu	Cu	Cu
Low-k Dielectric		none	FSG	FSG	CDO	CDO	CDO	CDO	SiLK	SiLK	CDO	Porous
k		3.9	3.55	3.6	2.9	2.9					2.75	2.4
SRAM Cell Size	μm^2	10.26	5.59	2.09	1	0.57	0.346	0.171	1.8	0.99	0.65	0.37

The transistor characteristics are listed for low- V_t transistors. Since the 130 nm generation, nearly all processes have offered a regular- V_t transistor offering an order of magnitude lower I_{off} at the expense of a 15% reduction in I_{dsat} . Some low-power processes provide a high- V_t transistor to reduce leakage by another order of magnitude. Most manufacturers use a separate implant mask to specify the threshold voltage, but Intel reduces manufacturing cost by using a slightly (~10%) longer channel length instead, which increases V_t on account of the short-channel effect [Rusu07].

Reported subthreshold slopes range from 85–100 mV/decade. DIBL coefficients range from 100–130 mV/V and tend to get larger with technology scaling.

Summary

CMOS process options and directions can greatly influence design decisions. Frequently, the combination of performance and cost possibilities in a new process can provide new product opportunities that were not available previously. Similarly, venerable processes can offer good opportunities with the right product.

One issue that has to be kept in mind is the ever-increasing cost of having a CMOS design fabricated in a leading-edge process. Mask cost for critical layers is in the vicinity of \$100K per mask. A full mask set for a 65 nm process can exceed \$1M in cost, and the price has been roughly doubling at each technology node. This in turn is reflected in the types of design and approaches to design that are employed for CMOS chips of the future. For instance, making a design programmable so that it can have a longer product life is a good first start. Chapter 14 covers these approaches in depth.

For more advanced reading on silicon processing, consult textbooks such as [Wolf00].

Exercises

- 3.1 A 248 nm UV step and scan machine costs \$10M and can produce 80 300 mm diameter, 90 nm node wafers per hour. A 193 nm UV step and scan machine costs \$40M and can process 20 300 mm diameter, 50 nm node wafers per hour. If the machines have a depreciation period of four years, what is the difference in the cost per chip for a chip that occupies 50 square mm at 90 nm resolution if the stepper is used 10 times per process run for the critical layers?
- 3.2 If the gate oxide thickness in a SiO_2 -based structure is 2 nm, what would be the thickness of an HfO_2 -based dielectric providing the same capacitance?
- 3.3 Explain the difference between a polycide and a salicide CMOS process. Which would be likely to have higher performance and why?
- 3.4 Draw the layout for a pMOS transistor in an n-well process that has active, p-select, n-select, polysilicon, contact, and metal1 masks. Include the well contact to V_{DD} .
- 3.5 What is the lowest resistance metal for interconnect? Why isn't it used?
- 3.6 Calculate the minimum contacted pitch as shown in Figure 3.39 for metal1 in terms of λ using the SUBM rules. Is there a wiring strategy that can reduce this pitch?

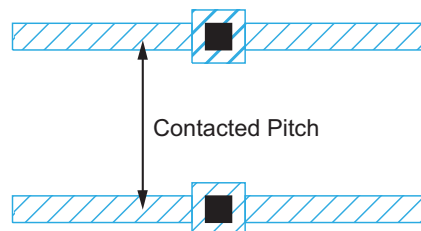


FIGURE 3.39 Contacted metal pitch

- 3.7 Using the SUBM rules, calculate the minimum uncontacted and contacted transistor pitch, as shown in Figure 3.40.

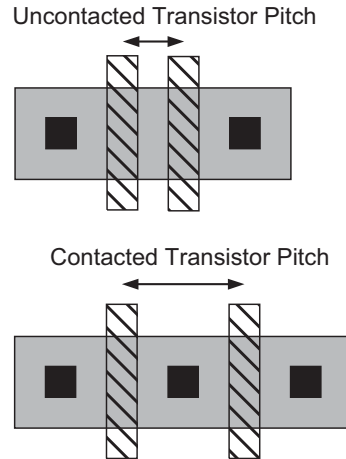


FIGURE 3.40 Uncontacted and contacted transistor pitch

- 3.8 Using Figure 3.41 and the SUBM design rules, calculate the minimum n to p pitch and the minimum inverter height with and without the poly contact to the gate (in). If an SOI process has 2λ spacing between n and p diffusion, to what are the two pitches reduced?

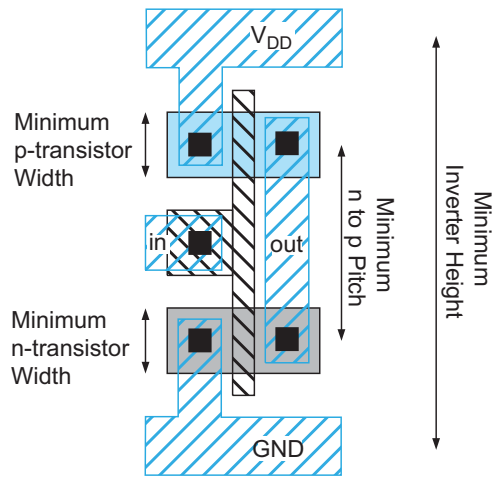


FIGURE 3.41 Minimum inverter height

- 3.9 Design a metal6 fuse ROM cell in a process where the minimum metal width is $0.5\mu\text{m}$ and the maximum current density is $2\text{ mA}/\mu\text{m}$. A fuse current of less than 10 mA is desired.

References

A majority of references are from IEEE publications and can be obtained from ieeexplore.ieee.org. An electronic version of this bibliography with hyperlinks is available at www.cmosvlsi.com. The *IEEE Journal of Solid-State Circuits* is cited heavily and is abbreviated as *JSSC*.

- [Abdollahi04] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage current reduction in CMOS VLSI circuits by input vector control," *IEEE Trans. VLSI*, vol. 12, no. 2, Feb. 2004, pp. 140–154.
- [Acken83] J. Acken, "Testing for bridging faults (shorts) in CMOS circuits," *Proc. Design Automation Conf.*, 1983, pp. 717–718.
- [Afghahi90] M. Afghahi and C. Svensson, "A unified single-phase clocking scheme for VLSI systems," *JSSC*, vol. 25, no. 1, Feb. 1990, pp. 225–233.
- [Agans06] D. Agans, *Debugging*, New York: Amacon, 2006, www.debuggingrules.com.
- [Agarwal01] V. Agarwal, S. Keckler, and D. Burger, "The effect of technology scaling on microarchitectural structures," *Computer Architecture and Technology Laboratory Technical Report TR2000-02*, University of Texas at Austin, 2001.
- [Agarwal04] A. Agarwal, V. Zolotov, and D. Blaauw, "Statistical clock skew analysis considering intra-die process variations," *IEEE Trans. CAD*, vol. 23, no. 8, Aug. 2004, pp. 1231–1242.
- [Agarwal07] A. Agarwal, K. Kang, S. Bhunia, J. Gallagher, and K. Roy, "Device-aware yield-centric dual- V_t design under parameter variations in nanoscale technologies," *IEEE Trans. VLSI*, vol. 15, no. 6, Jun. 2007, pp. 660–671.
- [Agarwal07b] K. Agarwal, R. Rao, D. Sylvester, and R. Brown, "Parametric yield analysis and optimization in leakage dominated technologies," *IEEE Trans. VLSI*, vol. 15, no. 6, Jun. 2007, pp. 613–623.
- [Agrawal08] B. Agrawal and T. Sherwood, "Ternary CAM power and delay model: extensions and uses," *IEEE Trans. VLSI*, vol. 16, no. 5, May 2008, pp. 554–564.
- [Aisaka02] K. Aisaka et al., "Design rule for frequency-voltage cooperative power control and its application to an MPEG-4 decoder," *Proc. VLSI Circuits Symp.*, 2002, pp. 216–217.
- [Alexander75] J. Alexander, "Clock recovery from random binary signals," *Electronics Letters*, vol. 11, no. 22, Oct. 30, 1975, pp. 541–542.
- [Allam00] M. Allam, M. Anis, and M. Elmasry, "High-speed dynamic logic styles for scaled-down CMOS and MTCMOS technologies," *Proc. Intl. Symp. Low Power Electronics and Design*, 2000, pp. 155–160.
- [Alon05] E. Alon, V. Stojanovic, and M. Horowitz, "Circuits and techniques for high-resolution measurement of on-chip power supply noise," *JSSC*, vol. 40, no. 4, Apr. 2005, pp. 820–828.
- [Alvandpour02] A. Alvandpour, R. Krishnamurthy, K. Soumyanath, and S. Borkar, "A sub-130-nm conditional keeper technique," *JSSC*, vol. 37, no. 5, May 2002, pp. 633–638.
- [Amrutur98] B. Amrutur and M. Horowitz, "A replica technique for wordline and sense control in low-power SRAM's," *JSSC*, vol. 33, no. 8, Aug. 1998, pp. 1208–1219.

- [Amrutur00] B. Amrutur and M. Horowitz, "Speed and power scaling of SRAM's," *JSSC*, vol. 35, no. 2, Feb. 2000, pp. 175–185.
- [Amrutur01] B. Amrutur and M. Horowitz, "Fast low-power decoders for RAMs," *JSSC*, vol. 36, no. 10, Oct. 2001, pp. 1506–1515.
- [Anastasakis02] D. Anastasakis, R. Damiano, H. Ma, and T. Stanion, "A practical and efficient method for compare-point matching," *Proc. Design Automation Conf.*, Jun. 2002, pp. 305–310.
- [Anderson02] F. Anderson, J. Wells, and E. Berta, "The core clock system on the next generation Itanium microprocessor," *Proc. IEEE Intl. Solid-State Circuits Conf.*, Feb. 2002, pp. 146–147, 453.
- [Ando80] H. Ando, "Testing VLSI with random access scan," *Digest of Papers COMPCON 80*, Feb. 1980, pp. 50–52.
- [Anis03] M. Anis and M. Elmasry, *Multi-Threshold CMOS Digital Circuits*, Norwell, MA: Kluwer, 2003.
- [Arnaud08] F. Arnaud et al., "32 nm general purpose bulk CMOS technology for high performance applications at low voltage," *Proc. Intl. Electron Devices Meeting*, Dec. 2008, pp. 1–4.
- [Artisan02] Artisan Components, *TSMC 0.18 μm Process 1.8-Volt SAGE-X Standard Cell Library Databook*, Release 4.0, Feb. 2002.
- [Asenov07] A. Asenov, "Simulation of statistical variability in nano MOSFETs," *Proc. VLSI Technology Symp.*, Jun. 2007, pp. 86–87.
- [Auth08] C. Auth et al., "45 nm high-k+ metal gate strain-enhanced transistors," *Intel Technology Journal*, vol. 12, no. 2, Jun. 2008, pp. 77–85.
- [Baghini02] M. Baghini and M. Desai, "Impact of technology scaling on metastability performance of CMOS synchronizing latches," *Proc. Intl. Conf. VLSI Design*, 2002, pp. 317–322.
- [Bai04] P. Bai et al., "A 65 nm logic technology featuring 35 nm gate lengths, enhanced channel strain, 8 Cu interconnect layers, low-k ILD and 0.57 μm^2 SRAM cell," *Proc. Intl. Electron Devices Meeting*, Dec. 2004, pp. 657–660.
- [Bailey98] D. Bailey and B. Benschneider, "Clocking design and analysis for a 600-MHz Alpha microprocessor," *JSSC*, vol. 33, no. 11, Nov. 1998, pp. 1627–1633.
- [Baker97] K. Baker and J. van Beers, "Shmoo plotting: the black art of IC testing," *IEEE Design and Test of Computers*, vol. 14, no. 3, Jul.–Sep. 1997, pp. 90–97.
- [Bakoglu90] H. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Reading, MA: Addison-Wesley, 1990.
- [Barke88] E. Barke, "Line-to-ground capacitance calculation for VLSI: a comparison," *IEEE Trans. Computer-Aided Design*, vol. 7, no. 2, Feb. 1988, pp. 295–298.
- [Barry03] J. Barry, E. Lee, and D. Messerschmitt, *Digital Communication*, 3rd ed., New York: Springer, 2003.
- [Barth08] J. Barth et al., "A 500 MHz random cycle, 1.5 ns latency, SOI embedded DRAM macro featuring a three-transistor micro sense amplifier," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 86–95.
- [Baugh73] C. Baugh and B. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Trans. Computers*, vol. C-22, no. 12, Dec. 1973, pp. 1045–1047.
- [Baumann01] R. Baumann, "Soft errors in advanced semiconductor devices—part I: the three radiation sources," *IEEE Trans. Device & Materials Reliability*, vol. 1, no. 1, Mar. 2001, pp. 17–22.
- [Baumann05] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design & Test of Computers*, vol. 22, no. 3, May–Jun. 2005, pp. 258–266.
- [Beaumont-Smith99] A. Beaumont-Smith, N. Burgess, S. Lefrere, and C. Lim, "Reduced latency IEEE floating-point adder architectures," *Proc. IEEE Symp. Computer Arithmetic*, Apr. 1999, pp. 35–42.
- [Beaumont-Smith01] A. Beaumont-Smith and C. Lim, "Parallel prefix adder design," *Proc. IEEE Symp. Computer Arithmetic*, 2001, pp. 218–225.
- [Bedrij62] O. Bedrij, "Carry-select adder," *IRE Trans. Electronic Computers*, vol. 11, Jun. 1962, pp. 340–346.
- [Belluomini05] W. Belluomini et al., "An 8GHz floating-point multiply," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2005, pp. 374–375, 604.
- [Bergeron05] J. Bergeron, E. Cerny, A. Hunter, and A. Nightingale, *Verification Methodology Manual for SystemVerilog*, New York: Springer, 2005.

- [Bernstein99] K. Bernstein, K. Carrig, C. Durham, P. Hansen, D. Hogenmiller, E. Nowak, and N. Roher, *High Speed CMOS Design Styles*, Boston: Kluwer Academic Publishers, 1999.
- [Bernstein00] K. Bernstein and N. Rohrer, *SOI Circuit Design Concepts*, Boston: Kluwer Academic Publishers, 2000.
- [Bernstein06] K. Bernstein et al., "High-performance CMOS variability in the 65-nm regime and beyond," *IBM J. Research and Dev.*, vol. 50, no. 4/5, Jul./Sep. 2006, pp. 433–449.
- [Bewick94] G. Bewick, "Fast Multiplication: Algorithms and Implementation," Ph.D. thesis, Stanford University, CSL-TR-94-617, 1994.
- [Bhavnagarwala01] A. Bhavnagarwala, Xinghai Tang, and J. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *JSSC*, vol. 36, no. 4, Apr. 2001, pp. 658–665.
- [Bhavnagarwala04] A. Bhavnagarwala et al., "A transregional CMOS SRAM with single, logic V_{DD} and dynamic power rails," *Proc. VLSI Circuits Symp.*, Jun. 2004, pp. 292–293.
- [Bhavnagarwala05] A. Bhavnagarwala et al., "Fluctuation limits & scaling opportunities for CMOS SRAM cells," *Proc. Intl. Electron Devices Meeting*, Dec. 2005, pp. 659–662.
- [Black69] J. Black, "Electromigration—A brief survey and some recent results," *IEEE Trans. Electron Devices*, vol. ED-16, no. 4, Apr. 1969, pp. 338–347.
- [Blackburn96] J. Blackburn, L. Arndt, and E. Swartzlander, "Optimization of spanning tree carry lookahead adders," *Proc. 30th Asilomar Conf. Signals, Systems, and Computers*, vol. 1, 1996, pp. 177–181.
- [Bohr95] M. Bohr, "Interconnect scaling—the real limiter to high performance ULSI," *Proc. Intl. Electron Devices Meeting*, Dec. 1995, pp. 241–244.
- [Bohr96] M. Bohr et al., "A high performance 0.25 μm logic technology optimized for 1.8 V operation," *Proc. Intl. Electron Devices Meeting*, Dec. 1996, pp. 847–850.
- [Booth51] A. Booth, "A signed binary multiplication technique," *Quarterly J. Mechanics and Applied Mathematics*, vol. IV, pt. 2, Jun. 1951, pp. 236–240.
- [Bowhill95] W. Bowhill et al., "Circuit implementation of a 300-MHz 64-bit second-generation CMOS Alpha CPU," *Digital Technical Journal*, vol. 7, no. 1, 1995, pp. 100–115.
- [Bowman02] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *JSSC*, vol. 37, no. 2, Feb. 2002, pp. 183–190.
- [Bowman09] K. Bowman et al., "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 49–63.
- [Brederlow06] R. Brederlow, R. Prakash, C. Paulus, and R. Thewes, "A low-power true random number generator using random telegraph noise of single oxide-traps," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2006, pp. 1666–1675.
- [Brent82] R. Brent and H. Kung, "A regular layout for parallel adders," *IEEE Trans. Computers*, vol. C-31, no. 3, Mar. 1982, pp. 260–264.
- [Brewer08] J. Brewer and M. Gill, eds., *Nonvolatile Memory Technology with Emphasis on Flash*, Piscataway, NJ: IEEE Press, 2008.
- [Brooks95] F. Brooks, *The Mythical Man-Month*, Boston: Addison-Wesley, 1995.
- [Brown03] A. Brown, "Fast films," *IEEE Spectrum*, vol. 40, no. 2, Feb. 2003, pp. 36–40.
- [Brunvand09] E. Brunvand, *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*, Boston: Addison Wesley, 2009.
- [Brusamarello08] L. Brusamarello, R. da Silva, G. Wirth, and R. Reis, "Probabilistic approach for yield analysis of dynamic logic circuits," *IEEE Trans. Circuits & Systems*, vol. 55, no. 8, Sep. 2008, pp. 2238–2248.
- [Budnik06] M. Budnik and K. Roy, "A power delivery and decoupling network minimizing ohmic loss and supply voltage variation in silicon nanoscale technologies," *IEEE Trans. VLSI*, vol. 14, no. 12, Dec. 2006, pp. 1336–1346.
- [Burd00] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," *JSSC*, vol. 35, no. 11, Nov. 2000, pp. 1571–1580.

- [Burgess02] N. Burgess, "The flagged prefix adder and its applications in integer arithmetic," *J. VLSI Signal Processing*, 2002, pp. 263–271.
- [Burgess09] N. Burgess, "Implementation of recursive Ling adders in CMOS VLSI," *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2009.
- [Burks46] A. Burks, H. Goldstine, and J. von Neumann, *Preliminary discussion of the logical design of an electronic computing instrument, part 1*, vol. 1, Inst. Advanced Study, Princeton, NJ, 1946.
- [Burleson98] W. Burleson, M. Ciesielski, F. Klass, and W. Liu, "Wave-pipelining: a tutorial and research survey," *IEEE Trans. VLSI*, vol. 6, no. 3, Sep. 1998, pp. 464–474.
- [Calhoun04] B. Calhoun, F. Honore, and A. Chandrakasan, "A leakage reduction methodology for distributed MTCMOS," *JSSC*, vol. 39, no. 5, May 2004, pp. 818–826.
- [Calhoun05] B. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *JSSC*, vol. 40, no. 9, Sep. 2005, pp. 1778–1786.
- [Calhoun06a] B. Calhoun and A. Chandrakasan, "Ultra-dynamic voltage scaling (UDVS) using sub-threshold operation and local voltage dithering," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 238–245.
- [Calhoun06b] B. Calhoun and A. Chandrakasan, "Static noise margin variation for sub-threshold SRAM in 65-nm CMOS," *JSSC*, vol. 41, no. 7, Jul. 2006, pp. 1673–1679.
- [Calhoun07] B. Calhoun and A. Chandrakasan, "A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation," *JSSC*, vol. 42, no. 3, Mar. 2007, pp. 680–688.
- [Calin96] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Trans. Nuclear Science*, vol. 43, no. 6, Dec. 1996, pp. 2874–2878.
- [Calma84] Calma Corporation, GDS II Stream Format, Jul. 1984.
- [Carr72] W. Carr and J. Mize, *MOS/LSI Design and Application*, New York: McGraw-Hill, 1972.
- [Celik02] M. Celik, L. Pileggi, and A. Odabasioglu, *IC Interconnect Analysis*, Boston: Kluwer Academic Publishers, 2002.
- [Cenker79] R. Cenker, D. Clemons, W. Huber, J. Petrizzi, F. Procyk, and G. Trout, "A fault-tolerant 64K dynamic random-access memory," *IEEE Trans. Electron Devices*, vol. 26, no. 6, Jun. 1979, pp. 853–860.
- [Chan90] P. Chan and M. Schlag, "Analysis and design of CMOS Manchester adders with variable carry-skip," *IEEE Trans. Computers*, vol. 39, no. 8, Aug. 1990, pp. 983–992.
- [Chan05] S. Chan, K. Shepard, and P. Restle, "Uniform-phase uniform-amplitude resonant-load global clock distributions," *JSSC*, vol. 40, no. 1, Jan. 2005, pp. 102–109.
- [Chan09] S. Chan et al., "A resonant global clock distribution for the cell broadband engine processor," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 64–72.
- [Chandrakasan92] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *JSSC*, vol. 27, no. 4, Apr. 1992, pp. 473–484.
- [Chandrakasan01] A. Chandrakasan, W. Bowhill, and F. Fox, ed., *Design of High-Performance Microprocessor Circuits*, Piscataway, NJ: IEEE Press, 2001.
- [Chaney73] T. Chaney and C. Molnar, "Anomalous behavior of synchronizer and arbiter circuits," *IEEE Trans. Computers*, vol. C-22, Apr. 1973, pp. 421–422.
- [Chaney83] T. Chaney, "Measured flip-flop responses to marginal triggering," *IEEE Trans. Computers*, vol. C-32, no. 12, Dec. 1983, pp. 1207–1209.
- [Chang05] J. Chang et al., "A 130-nm triple-V_t 9-MB third-level on-die cache for the 1.7-GHz Itanium 2 processor," *JSSC*, vol. 40, no. 1, Jan. 2005, pp. 195–203.
- [Chang07] J. Chang et al., "The 65-nm 16-MB shared on-die L3 cache for the dual-core Intel Xeon processor 7100 Series," *JSSC*, vol. 42, no. 4, Apr. 2007, pp. 846–852.
- [Chang08] L. Chang et al., "An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches," *JSSC*, vol. 43, no. 4, Apr. 2008, pp. 956–963.
- [Chao89] H. Chao and C. Johnston, "Behavior analysis of CMOS D flip-flops," *JSSC*, vol. 24, no. 5, Oct. 1989, pp. 1454–1458.

- [Chappell91] T. Chappell, B. Chappell, S. Schuster, J. Allan, S. Klepner, R. Joshi, and R. Franch, "A 2-ns cycle, 3.8-ns access 512-kb CMOS ECL SRAM with a fully pipelined architecture," *JSSC*, vol. 26, no. 11, Nov. 1991, pp. 1577–1585.
- [Chen96] K. Chen, H. Wann, J. Dunster, P. Ko, C. Hu, and M. Yoshida, "MOSFET carrier mobility model based on gate oxide thickness, threshold and gate voltages," *Solid-State Electronics*, vol. 39, no. 10, 1996, pp. 1515–1518.
- [Chen97] K. Chen, C. Hu, P. Fang, M. Lin, and D. Wollesen, "Predicting CMOS speed with gate oxide and voltage scaling and interconnect loading effects," *IEEE Trans. Electron Devices*, vol. 44, no. 11, Nov. 1997, pp. 1951–1957.
- [Chen03] T. Chen and S. Naffziger, "Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation," *IEEE Trans. VLSI*, vol. 11, no. 5, Oct. 2003, pp. 888–899.
- [Chen06] J. Chen, L. Clark, and T. Chen, "An ultra-low-power memory with a subthreshold power supply voltage," *JSSC*, vol. 41, no. 10, Oct. 2006, pp. 2344–2353.
- [Chen07] G. Chen, D. Blaauw, T. Mudge, D. Sylvester, and Nam Sung Kim, "Yield-driven near-threshold SRAM design," *Proc. Intl. Conf. Computer-Aided Design*, Nov. 2007, pp. 660–666.
- [Cheng99] Y. Cheng and C. Hu, *MOSFET Modeling & BSIM3 User's Guide*, Boston: Kluwer Academic Publishers, 1999.
- [Cheng00] Y. Cheng, C. Tsai, C. Teng, and S. Kang, *Electrothermal Analysis of VLSI Systems*, Boston: Kluwer Academic Publishers, 2000.
- [Chern92] J. Chern, J. Huang, L. Arledge, P. Li, and P. Yang, "Multilevel metal capacitance models for CAD design synthesis systems," *IEEE Electron Device Letters*, vol. 13, no. 1, Jan. 1992, pp. 32–34.
- [Childs84] R. Childs, J. Crawford, D. House, and R. Noyce, "A processor family for personal computers," *Proc. IEEE*, vol. 72, no. 3, Mar. 1984, pp. 363–376.
- [Chinnery02] D. Chinnery and K. Keutzer, *Closing the Gap Between ASIC and Custom: Tools and techniques for high-performance ASIC design*, Boston: Kluwer Academic Publishers, 2002.
- [Chinnery07] D. Chinnery & K. Keutzer, *Closing the Power Gap Between ASIC and Custom*, New York: Springer, 2007.
- [Choi97] J. Choi, L. Jang, S. Jung, and J. Choi, "Structured design of a 288-tap FIR filter by optimized partial product tree compression," *JSSC*, vol. 32, no. 3, Mar. 1997, pp. 468–476.
- [Chong06] K. Chong, L. McMurchie, and C. Sechen, "A 64b adder using self-calibrating differential output prediction logic," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2006, pp. 1745–1754.
- [Chopra06] K. Chopra, C. Kashyap, H. Su, and D. Blaauw, "Current source driver model synthesis and worst-case alignment for accurate timing and noise analysis," *Intl. Workshop on Timing in Synthesis and Specification (TAU)*, Feb. 2006.
- [Choudhury97] M. Choudhury and J. Miller, "A 300 MHz CMOS microprocessor with multi-media technology," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1997, pp. 170–171.
- [Christiansen06] C. Christiansen, J. Gambino, J. Therrien, D. Hunt, and J. Gill, "Effect of wire thickness on electromigration and stress migration lifetime of Cu," *Proc. Failure Analysis of Integrated Circuits*, Jul. 2006, pp. 349–354.
- [Chu86] K. Chu and D. Pulfrey, "Design procedures for differential cascode voltage switch circuits," *JSSC*, vol. SC-21, no. 6, Dec. 1986, pp. 1082–1087.
- [Chu87] K. Chu and D. Pulfrey, "A comparison of CMOS circuit techniques: differential cascode voltage switch logic versus conventional logic," *JSSC*, vol. SC-22, no. 4, Aug. 1987, pp. 528–532.
- [Clark61] C. Clark, "The greatest of a finite set of random variables," *Operations Research*, vol. 9, no. 2, Mar.–Apr. 1961, pp. 145–162.
- [Clark01] L. Clark et al., "An embedded 32-b microprocessor core for low-power and high-performance applications," *JSSC*, vol. 36, no. 11, Nov. 2001, pp. 1599–1608.

- [Clark02] L. Clark, S. Demmons, N. Deutscher, and F. Ricci, "Standby power management for a 0.18 μm microprocessor," *Proc. Intl. Symp. Low Power Electronics and Design*, Aug. 2002, pp. 7–12.
- [Cobbold66] R. Cobbold, "Temperature effects on M.O.S. transistors," *Electronics Letters*, vol. 2, no. 6, Jun. 1966, pp. 190–192.
- [Cobbold70] R. Cobbold, *Theory and Application of Field Transistors*, New York: Wiley Interscience, 1970.
- [Colwell95] R. Colwell and R. Steck, "A 0.6 μm BiCMOS processor with dynamic execution," *Proc. IEEE Solid-State Circuits Conf.*, 1995, pp. 176–177.
- [Colwell06] R. Colwell, *The Pentium Chronicles: The People, Passion, and Politics Behind Intel's Landmark Chips*, New York: Wiley, 2006.
- [Cooley07] J. Cooley, "Verilog vs. VHDL," www.deepchip.com/items/dvcon07-02.html, Apr. 2007.
- [Cortadella92] J. Cortadella and J. Llabería, "Evaluation of $A + B = K$ conditions without carry propagation," *IEEE Trans. Computers*, vol. 41, no. 11, Nov. 1992, pp. 1484–1487.
- [Crews03] M. Crews and Y. Yuenyongsgool, "Practical design for transferring signals between clock domains," *EDN Magazine*, Feb. 20, 2003, pp. 65–71.
- [Curran02] B. Curran et al., "IBM eServer z900 high-frequency microprocessor technology, circuits, and design methodology," *IBM J. Research and Development*, vol. 46, no. 4/5, Jul./Sep. 2002, pp. 631–644.
- [Dabral98] S. Dabral and T. Maloney, *Basic ESD and I/O Design*, New York: John Wiley & Sons, 1998.
- [Dadda65] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*, vol. 34, no. 5, May 1965, pp. 349–356.
- [Dally98] W. Dally and J. Poulton, *Digital Systems Engineering*, Cambridge, UK: Cambridge University Press, 1998.
- [Das06] S. Das et al., "A self-tuning DVS processor using delay-error detection and correction," *JSSC*, vol. 41, no. 4, Apr. 2006, pp. 792–804.
- [Das09] S. Das et al., "RazorII: in situ error detection and correction for PVT and SER tolerance," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 32–48.
- [Davari99] B. Davari, "CMOS technology: present and future," *Symp. VLSI Circuits Digest Tech. Papers*, 1999, pp. 5–10.
- [Davis69] R. Davis, "The ILLIAC IV processing element," *IEEE Trans. Computers*, vol. C-18, no. 9, Sep. 1969, pp. 800–816.
- [Degalahal05] V. Degalahal, L. Li, V. Narayanan, M. Kandemir, and M. Irwin, "Soft errors issues in low-power caches," *IEEE Trans. VLSI*, vol. 13, no. 10, Oct. 2005, pp. 1157–1166.
- [DeHon93] A. DeHon, T. Knight Jr., and T. Simon, "Automatic impedance control," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 1993, pp. 164–165, 283.
- [Deleganes02] D. Deleganes, J. Douglas, B. Kommandur, and M. Patyra, "Designing a 3GHz, 130nm, Intel Pentium 4 processor," *Symp. VLSI Circuits Digest Tech. Papers*, 2002, pp. 130–133.
- [Deleganes04] D. Deleganes et al., "LVS technology for the Intel Pentium 4 processor on 90nm technology," *Intel Technology Journal*, vol. 8, no. 1, Feb. 2004, pp. 43–53.
- [Deleganes05] D. Deleganes et al., "Low-voltage swing logic circuits for a Pentium 4 processor integer core," *JSSC*, vol. 40, no. 1, Jan. 2005, pp. 36–43.
- [Delgado-Frias00] J. Delgado-Frias and J. Nyathi, "A high-performance encoder with priority lookahead," *IEEE Trans. Circuits and Systems I*, vol. 47, no. 9, Sep. 2000, pp. 1390–1393.
- [Dennard68] R. Dennard, "Field-effect transistor memory," US Patent 3,387,286, 1968.
- [Dennard74] R. Dennard et al., "Design of ion-implanted MOSFET's with very small physical dimensions," *JSSC*, vol. SC-9, no. 5, Oct. 1974, pp. 256–268.
- [Dhanesha95] H. Dhanesha, K. Falakshahi, and M. Horowitz, "Array-of-arrays architecture for parallel floating point multiplication," *Proc. Conf. Advanced Research in VLSI*, 1995, pp. 150–157.
- [Dickson76] J. Dickson, "On-chip high-voltage generation in MNOS integrated circuits using an improved voltage multiplier technique," *JSSC*, vol. 11, no. 3, Jun. 1976, pp. 374–378.

- [Dike99] C. Dike and E. Burton, "Miller and noise effects in a synchronizing flip-flop," *JSSC*, vol. 34, no. 6, Jun. 1999, pp. 849–855.
- [Dobbalaere95] I. Dobbalaere, M. Horowitz, and A. El Gamal, "Regenerative feedback repeaters for programmable interconnect," *JSSC*, vol. 30, no. 11, Nov. 1995, pp. 1246–1253.
- [Dobberpuhl92] D. Dobberpuhl et al., "A 200-MHz 64-b dual-issue CMOS microprocessor," *JSSC*, vol. 27, no. 11, Nov. 1992, pp. 1555–1567.
- [Dobson95] J. Dobson and G. Blair, "Fast two's complement VLSI adder design," *Electronics Letters*, vol. 31, no. 20, Sep. 1995, pp. 1721–1722.
- [Donnay03] S. Donnay and G. Gielen, eds., *Substrate Noise Coupling in Mixed-Signal ASICs*, Boston: Kluwer Academic Publishers, 2003.
- [Donovan02] C. Donovan and M. Flynn, "A 'digital' 6-bit ADC in 0.25 μm CMOS," *JSSC*, vol. 37, no. 3, Mar. 2002, pp. 432–437.
- [Doran88] R. Doran, "Variants of an improved carry look-ahead adder," *IEEE Trans. Computers*, vol. 37, no. 9, Sep. 1988, pp. 1110–1113.
- [Doyle91] B. Doyle, B. Fishbein, and K. Mistry, "NBTI-enhanced hot carrier damage in p-channel MOSFETs," *Proc. Intl. Electron Devices Meeting*, 1991, pp. 529–532A.
- [Drake07] A. Drake et al., "A distributed critical-path timing monitor for a 65nm high-performance microprocessor," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2007, pp. 398–399.
- [Draper97] D. Draper et al., "Circuit techniques in a 266-MHz MMX-enabled processor," *JSSC*, vol. 32, no. 11, Nov. 1997, pp. 1650–1664.
- [Dunga07] M. Dunga et al., *BSIM 4.6.1 MOSFET Model User's Manual*, Department of Electrical Engineering and Computer Sciences, UC Berkeley, 2007.
- [Earle65] J. Earle, "Latched carry-save adder," *IBM Tech. Disclosure Bulletin*, vol. 7, no. 10, Mar. 1965, pp. 909–910.
- [Edwards93] B. Edwards, A. Corry, N. Weste, and C. Greenberg, "A single-chip video ghost canceller," *JSSC*, vol. 28, no. 3, Mar. 1993, pp. 379–383.
- [Eichelberger78] E. Eichelberger and T. Williams, "A logic design structure for LSI testability," *J. Design Automation and Fault Tolerant Computing*, vol. 2, no. 2, May 1978, pp. 165–178.
- [Elmore48] W. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *J. Applied Physics*, vol. 19, no. 1, Jan. 1948, pp. 55–63.
- [Emma08] P. Emma and E. Kursan, "Is 3D chip technology the next growth engine for performance improvement?" *IBM J. Research & Dev.*, vol. 52, no. 6, Nov. 2008, pp. 541–552.
- [EPA07] US Environmental Protection Agency, *Report to Congress on Server and Data Center Energy Efficiency*, Public Law 109-431, Aug. 2, 2007.
- [Ercegovac89] M. Ercegovac and T. Lang, "Binary counter with counting period of one half adder independent of counter size," *IEEE Trans. Circuits & Systems*, vol. 36, no. 6, Jun. 1989, pp. 924–926.
- [Ernst03] D. Ernst et al., "Razor: a low-power pipeline based on circuit-level timing speculation," *Proc. Intl. Symp. Microarchitecture*, Dec. 2003, pp. 7–18.
- [Estreich82] D. Estreich and R. Dutton, "Modeling latch-up in CMOS integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. CAD-1, no. 4, Oct. 1982, pp. 157–162.
- [Evans95] R. Evans and P. Franzon, "Energy consumption modeling and optimization for SRAM's," *JSSC*, vol. 30, no. 5, May 1995, pp. 571–579.
- [Faggin96] F. Faggin, M. Hoff, S. Mazor, and M. Shima, "The history of the 4004," *IEEE Micro*, vol. 16, no. 6, Dec. 1996, pp. 10–20.
- [Fetzer02] E. Fetzer, M. Gibson, A. Klein, N. Calick, C. Zhu, E. Busta, and B. Mohammad, "A fully bypassed six-issue integer datapath and register file on the Itanium-2 microprocessor," *JSSC*, vol. 37, no. 11, Nov. 2002, pp. 1433–1440.
- [Fetzer06] E. Fetzer, D. Dahle, C. Little, and K. Safford, "The parity protected, multithreaded register files on the 90-nm Itanium microprocessor," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 246–255.

- [Fischer06] T. Fischer, J. Desai, B. Doyle, S. Naffziger, and B. Patella, "A 90-nm variable frequency clock system for a power-managed Itanium architecture processor," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 218–228.
- [Fishburn85] J. Fishburn and A. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," *Proc. Intl. Conf. Computer-Aided Design*, Nov. 1985, pp. 326–328.
- [Flannagan85] S. Flannagan, "Synchronization reliability in CMOS technology," *JSSC*, vol. SC-20, no. 4, Aug. 1985, pp. 880–882.
- [Foty96] D. Foty, *MOSFET Modeling with SPICE: Principles and Practices*, Upper Saddle River, NJ: Prentice Hall, 1996.
- [Friedman84] V. Friedman and S. Liu, "Dynamic logic CMOS circuits," *JSSC*, vol. SC-19, no. 2, Apr. 1984, pp. 263–266.
- [Frohman69] D. Frohman-Bentchkowsky and A. Grove, "Conductance of MOS transistors in saturation," *IEEE Trans. Electron Devices*, vol. ED-16, no. 1, Jan. 1969, pp. 108–113.
- [Frowerk77] R. Frowerk, "Signature Analysis: A New Digital Field Service Method," *Hewlett Packard Journal*, May 1977, pp. 2–8.
- [Fujiwara06] E. Fujiwara, *Code Design for Dependable Systems: Theory and Practical Applications*, New York: Wiley, 2006.
- [Gabara92] T. Gabara and S. Knauer, "Digitally adjustable resistors in CMOS for high-performance applications," *JSSC*, vol. 27, no. 8, Aug. 1992, pp. 1176–1185.
- [Gago93] A. Gago, R. Escano, and J. Hidalgo, "Reduced implementation of D-type DET flip-flops," *JSSC*, vol. 28, no. 3, Mar. 1993, pp. 400–402.
- [Gajski83] D. Gajski and R. Kuhn, "New VLSI tools," *Computer*, vol. 16, no. 12, Dec. 1983, pp. 11–14.
- [Galiay80] J. Galiay, Y. Crouzet, and M. Verginiault, "Physical versus logical fault models MOS LSI circuits: impact on their testability," *IEEE Trans. Computers*, vol. C-29, no. 6, Jun. 1980, pp. 527–531.
- [Gauthier02] C. Gauthier and B. Amick, "Inductance: Implications and solutions for high-speed digital circuits: the chip electrical interface," *Proc. IEEE Intl. Solid-State Circuits Conf.*, vol. 2, 2002, pp. 563–565.
- [Geannopoulos98] G. Geannopoulos and X. Dai, "An adaptive digital deskewing circuit for clock distribution networks," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1998, pp. 400–401.
- [Gelsinger01] P. Gelsinger, "Microprocessors for the new millennium: challenges, opportunities, and new frontiers," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 2001, pp. 22–25.
- [George96] S. George, A. Ott, and J. Klaus, "Surface chemistry for atomic layer growth," *J. Phys. Chem.*, vol. 100, 1996, pp. 13121–13131.
- [George07] V. George et al., "Penryn: 45-nm next generation Intel Core 2 processor," *Proc. Intl. Solid-State Circuits Conf.*, Nov. 2007, pp. 14–17.
- [Geppert04] L. Geppert, "Chip making's wet new world," *IEEE Spectrum*, vol. 41, no. 5, May 2004, pp. 29–33.
- [Gerosa94] G. Gerosa et al., "A 2.2 W, 80 MHz superscalar RISC microprocessor," *JSSC*, vol. 29, no. 12, Dec. 1994, pp. 1440–1452.
- [Gerosa09] G. Gerosa et al., "A sub-2 W low power IA processor for mobile internet devices in 45 nm high-k metal gate CMOS," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 73–82.
- [Ginosar03] R. Ginosar, "Fourteen ways to fool your synchronizer," *Proc. Intl. Symp. Asynchronous Circuits and Systems*, May 2003, pp. 89–96.
- [Glasser85] L. Glasser and D. Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Reading, MA: Addison Wesley, 1985.
- [Gochman03] S. Gochman et al., "The Intel Pentium M processor: microarchitecture and performance," *Intel Technology Journal*, vol. 7, no. 2, May 2003, pp. 21–36.
- [Golden99] M. Golden et al., "A seventh-generation x86 microprocessor," *JSSC*, vol. 34, no. 11, Nov. 1999, pp. 1466–1477.
- [Golomb81] S. Golomb, *Shift Register Sequences*, Revised Edition, Laguna Hills, CA: Aegean Park Press, 1981.
- [Gonclaves83] N. Gonclaves and H. DeMan, "NORA: a racefree dynamic CMOS technique for pipelined logic structures," *JSSC*, vol. SC-18, no. 3, Jun. 1983, pp. 261–266.

- [Gonzalez96] R. Gonzalez and M. Horowitz, "Energy dissipation in general purpose microprocessors," *JSSC*, vol. 31, no. 9, Sep. 1996, pp. 1277–1284.
- [Gonzalez97] R. Gonzalez, B. Gordon, and M. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *JSSC*, vol. 32, no. 8, Aug. 1997, pp. 1210–1216.
- [Goto92] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54×54-b regularly structured tree multiplier," *JSSC*, vol. 27, no. 9, Sep. 1992, pp. 1229–1236.
- [Goto97] G. Goto et al., "A 4.1-ns compact 54×54-b multiplier utilizing sign-select Booth encoders," *JSSC*, vol. 32, no. 11, Nov. 1997, pp. 1676–1682.
- [Grad04] J. Grad and J. Stine, "A hybrid Ling carry-select adder," *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2004, pp. 1363–1367.
- [Gray53] F. Gray, "Pulse code communications," US Patent 2,632,058, 1953.
- [Gray01] P. Gray, P. Hurst, S. Lewis, and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th ed., New York: John Wiley & Sons, 2001.
- [Greenhill97] D. Greenhill et al., "A 330 MHz 4-way superscalar microprocessor," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 1997, pp. 166–167, 449.
- [Gregg07] J. Gregg and T. Chen, "Post silicon power/performance optimization in the presence of process variations using individual well-adaptive body biasing," *IEEE Trans. VLSI*, vol. 15, no. 3, Mar. 2007, pp. 366–376.
- [Griffin83] W. Griffin and J. Hildebeitel, "CMOS 4-way XOR circuit," *IBM Technical Disclosure Bulletin*, vol. 25, no. 11B, Apr. 1983, pp. 6066–6067.
- [Gronowski96] P. Gronowski et al., "A 433-MHz 64-b quad-issue RISC microprocessor," *JSSC*, vol. 31, no. 11, Nov. 1996, pp. 1687–1696.
- [Gronowski98] P. Gronowski, W. Bowhill, R. Preston, M. Gowan, and R. Allmon, "High-performance microprocessor design," *JSSC*, vol. 33, no. 5, May 1998, pp. 676–686.
- [Grotjohn86] T. Grotjohn and B. Hoefflinger, "Sample-set differential logic (SSDL) for complex high-speed VLSI," *JSSC*, vol. SC-21, no. 2, Apr. 1986, pp. 367–369.
- [Guilar09] N. Guilar, T. Kleeburg, A. Chen, D. Yankelevich, and R. Amirtharajah, "Integrated solar energy harvesting and storage," *IEEE Trans. VLSI*, vol. 17, no. 5, May 2009, pp. 627–637.
- [Gunning92] B. Gunning, L. Yuan, T. Nguyen, and T. Wong, "A CMOS low-voltage-swing transmission-line transceiver," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 1992, pp. 58–59.
- [Guo05] X. Guo and C. Sechen, "High speed redundant adder and divider in output prediction logic," *Proc. Computer Society Symp. VLSI*, May 2005, pp. 34–41.
- [Gutierrez01] E. Gutierrez, J. Deen, and C. Claeys (eds.), *Low Temperature Electronics: Physics, Devices, Circuits, and Applications*, New York: Academic Press, 2001.
- [Gutnik97] V. Gutnik and A. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Trans. VLSI*, vol. 5, no. 4, Dec. 1997, pp. 425–435.
- [Guyot87] A. Guyot, B. Hochet, and J. Muller, "A way to build efficient carry-skip adders," *IEEE Trans. Computers*, vol. 36, no. 10, Oct. 1987, pp. 1144–1152.
- [Guyot97] A. Guyot and S. Abou-Samra, "Modeling power consumption in arithmetic operators," *Microelectronic Engineering*, vol. 39, 1997, pp. 245–253.
- [Hall00] S. Hall, G. Hall, and J. McCall, *High-Speed Digital System Design*, New York: Wiley, 2000.
- [Hamada98] M. Hamada et al., "A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme," *Proceedings of the IEEE*, May 11–14, 1998, pp. 495–498.
- [Hamming50] R. Hamming, "Error Detecting and Error Correcting Codes," *Bell Systems Technical Journal*, vol. 29, pp. 147–160.
- [Hamzaoglu02] F. Hamzaoglu and M. Stan, "Circuit-level techniques to control gate leakage for sub-100 nm CMOS," *Proc. Intl. Symp. Low Power Electronics and Design*, 2002, pp. 60–63.

- [Hamzaoglu09] F. Hamzaoglu et al., "A 3.8 GHz 153 Mb SRAM design with dynamic stability enhancement and leakage reduction in 45 nm high-k metal gate CMOS technology," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 148–154.
- [Han87] T. Han and D. Carlson, "Fast area-efficient VLSI adders," *Proc. IEEE Symp. Computer Arithmetic*, 1987, pp. 49–56.
- [Hanson06] S. Hanson et al., "Ultralow-voltage minimum-energy CMOS," *IBM J. Research & Dev.*, vol. 50, no. 4/5, Jul./Sep. 2006, pp. 469–490.
- [Hanson09] S. Hanson et al., "A low-voltage processor for sensing applications with picowatt standby mode," *JSSC*, vol. 44, no. 4, Apr. 2009, pp. 1145–1155.
- [Harame01a] D. Harame and B. Meyerson, "The early history of IBM's SiGe mixed signal technology," *IEEE Transactions on Electron Devices*, vol. 48, no. 11, Nov. 2001, pp. 2555–2567.
- [Harame01b] D. Harame et al., "Current status and future trends of SiGe BiCMOS technology," *IEEE Transactions on Electron Devices*, vol. 48, no. 11, Nov. 2001, pp. 2575–2594.
- [Haring96] R. Haring et al., "Self-resetting logic register and incrementer," *Symp. VLSI Circuits Digest Tech. Papers*, 1996, pp. 18–19.
- [Harris97] D. Harris and M. Horowitz, "Skew-tolerant domino circuits," *JSSC*, vol. 32, no. 11, Nov. 1997, pp. 1702–1711.
- [Harris99] D. Harris, M. Horowitz, and D. Liu, "Timing analysis including clock skew," *IEEE Trans. Computer-Aided Design*, vol. 18, no. 11, Nov. 1999, pp. 1608–1618.
- [Harris01a] D. Harris, *Skew-Tolerant Circuit Design*, San Francisco, CA: Morgan Kaufmann, 2001.
- [Harris01b] D. Harris and S. Naffziger, "Statistical clock skew modeling with data delay variations," *IEEE Trans. VLSI*, vol. 9, no. 6, Dec. 2001, pp. 888–898.
- [Harris03] D. Harris, "A taxonomy of prefix networks," *Proc. 37th Asilomar Conf. Signals, Systems, and Computers*, 2003, pp. 2213–2217.
- [Harris04] D. Harris, "Logical effort of higher valency adders," *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2004, pp. 1358–1362.
- [Harris07] D. Harris and S. Harris, *Digital Design and Computer Architecture*, San Francisco: Morgan Kaufmann Publishers, 2007.
- [Hart06] J. Hart et al., "Implementation of a fourth-generation 1.8-GHz dual-core SPARC V9 microprocessor," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 210–217.
- [Hashemian92] R. Hashemian and C. Chen, "A new parallel technique for design of decrement/increment and two's complement circuits," *Proc. IEEE Midwest Symp. Circuits and Systems*, vol. 2, 1992, pp. 887–890.
- [Hashimoto02] T. Hashimoto et al., "Integration of a 0.13- μ m CMOS and a high performance self-aligned SiGe HBT featuring low base resistance," *Proc. Intl. Electron Devices Meeting*, Dec. 2002, pp. 779–782.
- [Hatamian86] M. Hatamian and G. Cash, "A 70-MHz 8-bit $\times$ 8-bit parallel pipelined multiplier in 2.5- μ m CMOS," *JSSC*, vol. 21, no. 4, Aug. 1986, pp. 505–513.
- [Haykin00] S. Haykin, *Digital Communications*, New York: John Wiley & Sons, 2000.
- [Hazucha00] P. Hazucha, C. Svensson, and S. Wender, "Cosmic-ray soft error rate characterization of a standard 0.6- μ m CMOS process," *JSSC*, vol. 35, no. 10, Oct. 2000, pp. 1422–1429.
- [Hazucha04] P. Hazucha et al., "Measurements and analysis of SER-tolerant latch in a 90-nm dual- V_t CMOS process," *JSSC*, vol. 39, no. 9, Sep. 2004, pp. 1536–1543.
- [Heald93] R. Heald and J. Holst, "A 6-ns cycle 256 kb cache memory and memory management unit," *JSSC*, vol. 28, no. 11, Nov. 1993, pp. 1078–1083.
- [Heald98] R. Heald et al., "64-Kbyte sum-addressed-memory cache with 1.6-ns cycle and 2.6-ns latency," *JSSC*, vol. 33, no. 11, Nov. 1998, pp. 1682–1689.
- [Heald00] R. Heald et al., "A third-generation SPARC v9 64-b microprocessor," *JSSC*, vol. 35, no. 11, Nov. 2000, pp. 1526–1538.
- [Hedenstierna87] N. Hedenstierna and K. Jeppson, "CMOS circuit speed and buffer optimization," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 2, Mar. 1987, pp. 270–281.

- [Heikes94] C. Heikes, "A 4.5mm² multiplier array for a 200MFLOP pipelined coprocessor," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1994, pp. 290–291.
- [Heller84] L. Heller, W. Griffin, J. Davis, and N. Thoma, "Cascode voltage switch logic: a differential CMOS logic family," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1984, pp. 16–17.
- [Hennessy90] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1990.
- [Heo02] S. Heo, K. Barr, M. Hampton, and K. Asanovic, "Dynamic fine-grain leakage reduction using leakage-biased bitlines," *Proc. Intl. Symp. Computer Architecture*, 2002, pp. 137–147.
- [Hess94] C. Hess and L. Weiland, "Drop-in process control checkerboard test structure for efficient online process characterization and defect problem debugging," *Proc. IEEE Int. Conf. Microelectronic Test Structures*, vol. 7, Mar., 1994, pp. 152–159.
- [Hicks08] J. Hicks, "45nm transistor reliability," *Intel Technology Journal*, vol. 12, no. 2, Jun. 2008, pp. 131–144.
- [Hidaka89] H. Hidaka, K. Fujishima, Y. Matsuda, M. Asakura, and T. Yoshihara, "Twisted bit-line architectures for multi-megabit DRAM's," *JSSC*, vol. 24, no. 1, Feb. 1989, pp. 21–27.
- [Hilewitz04] Y. Hilewitz, Z. Shi, and R. Lee, "Comparing fast implementations of bit permutation instructions," *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2004, pp. 1856–1863.
- [Hilewitz07] Y. Hilewitz and R. Lee, "Performing advanced bit manipulations efficiently in general-purpose processors," *Proc. Computer Arithmetic Symp.*, June 2007, pp. 251–260.
- [Hill68] C. Hill, "Noise margin and noise immunity in logic circuits," *Microelectronics*, vol. 1, Apr. 1968, pp. 16–21.
- [Hinton01] G. Hinton et al., "A 0.18- μ m CMOS IA-32 processor with a 4-GHz integer execution unit," *JSSC*, vol. 36, no. 11, Nov. 2001, pp. 1617–1627.
- [Hisamoto98] D. Hisamoto et al., "A folded-channel MOSFET for deep-sub-tenth micron era," *Tech. Digest Intl. Electron Devices Meeting*, San Francisco, Dec. 1998, pp. 1032–1034.
- [Ho01] R. Ho, K. Mai, and M. Horowitz, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, Apr. 2001, pp. 490–504.
- [Ho03a] R. Ho, K. Mai, and M. Horowitz, "Efficient on-chip global interconnects," *Symp. VLSI Circuits Digest Tech. Papers*, 2003, pp. 271–274.
- [Ho03b] R. Ho, K. Mai, and M. Horowitz, "Managing wire scaling: a circuit perspective," *Proc. IEEE Interconnect Technology Conf.*, 2003, pp. 177–179.
- [Ho07] R. Ho, "Dealing with issues in VLSI interconnect scaling," *Intl. Solid-State Circuits Conf. Tutorial*, Feb. 2007.
- [Hoeneisen72] B. Hoeneisen and C. Mead, "Fundamental limitations in Microelectronics-I. MOS technology," *Solid-State Electronics*, vol. 15, 1972, pp. 819–829.
- [Hogge85] C. Hogge Jr., "A self correcting clock recovery circuit," *IEEE Trans. Electron Devices*, vol. 32, no. 12, Dec. 1985, pp. 2704–2706.
- [Hook03] T. Hook et al., "Lateral ion implant straggle and mask proximity effect," *IEEE Trans. Electron Devices*, vol. 50, no. 9, Sep. 2003, pp. 1946–1951.
- [Horowitz83] M. Horowitz and R. Dutton, "Resistance extraction from mask layout data," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, no. 3, Jul. 1983, pp. 145–150.
- [Horowitz87] M. Horowitz et al., "MIPS-X: a 20-MIPS peak, 32-bit microprocessor with on-chip cache," *JSSC*, vol. SC-22, no. 5, Oct. 1987, pp. 790–799.
- [Horowitz04] M. Horowitz and W. Dally, "How scaling will change processor architecture," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2004, pp. 132–133.
- [Horstmann89] J. Horstmann, H. Eichel, and R. Coates, "Metastability behavior of CMOS ASIC flip-flops in theory and test," *JSSC*, vol. 24, no. 1, Feb. 1989, pp. 146–157.
- [Hrishikesh02] M. Hrishikesh et al., "The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays," *Proc. Intl. Symp. Computer Architecture*, 2002, pp. 14–24.

- [Hsiao70] M. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM J. Research & Dev.*, vol. 14, no. 4, Jul. 1970, pp. 395–401.
- [Hsu91] W. Hsu, B. Sheu, and S. Gowda, "Design of reliable VLSI circuits using simulation techniques," *JSSC*, vol. 26, no. 3, Mar. 1991, pp. 452–457.
- [Hsu92] W. Hsu, B. Sheu, S. Gowda, and C. Hwang, "Advanced integrated-circuit reliability simulation including dynamic stress effects," *JSSC*, vol. 27, no. 3, Mar. 1992, pp. 247–257.
- [Hsu06a] S. Hsu et al., "A 110 GOPS/W 16-bit multiplier and reconfigurable PLA loop in 90-nm CMOS," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 256–264.
- [Hsu06b] S. Hsu, A. Agarwal, M. Anders, S. Mathew, R. Krishnamurthy, and S. Borkar, "An 8.8GHz 198mW 16 × 64b 1R/1W variation tolerant register file in 65nm CMOS," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2006, pp. 1785–1797.
- [Hu90] Y. Hu and S. Chen, "GM_Plan: A Gate Matrix Layout Algorithm Based on Artificial Intelligence Planning Techniques," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 8, Aug. 1990, pp. 836–845.
- [Hu92] C. Hu, "IC Reliability Simulation," *JSSC*, vol. 27, no. 3, Mar. 1992, pp. 241–246.
- [Hu95] C. Hu, K. Rodbell, T. Sullivan, K. Lee, and D. Bouldin, "Electromigration and stress-induced voiding in fine Al and Al-alloy thin-film lines," *IBM J. Research and Development*, vol. 39, no. 4, Jul. 1995, pp. 465–497.
- [Huang00] Z. Huang and M. Ercegovac, "Effect of wire delay on the design of prefix adders in deep-submicron technology," *Proc. 34th Asilomar Conf. Signals, Systems, and Computers*, vol. 2, 2000, pp. 1713–1717.
- [Huang02] C. Huang, J. Wang, and Y. Huang, "Design of high-performance CMOS priority encoders and incrementers/decrementers using multilevel lookahead and multilevel folding techniques," *JSSC*, vol. 37, no. 1, Jan. 2002, pp. 63–76.
- [Huang03] X. Huang et al., "Loop-based interconnect modeling and optimization approach for multigigahertz clock network design," *JSSC*, vol. 38, no. 3, Mar. 2003, pp. 457–463.
- [Huang05] Z. Huang and M. Ercegovac, "High-performance low-power left-to-right array multiplier design," *IEEE Trans. Computers*, vol. 54, no. 3, Mar. 2005, pp. 272–283.
- [Huang06] A. Huang et al., "A 10Gb/s photonic modulator and WDM MUX/DEMUX integrated with electronics in 0.13/spl mu/m SOI CMOS," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2006, pp. 922–929.
- [Huh98] Y. Huh, Y. Sung, and S. Kang, "A study of hot-carrier-induced mismatch drift: a reliability issue for VLSI circuits," *JSSC*, vol. 33, no. 6, Jun. 1998, pp. 921–927.
- [Huitema03] E. Huitema et al., "Plastic transistors in active-matrix displays," *Proc. IEEE Intl. Solid-State Circuits Conf.*, Feb. 2003, pp. 380–381.
- [Huntzicker08] S. Huntzicker, M. Dayringer, J. Soprano, A. Weerasinghe, D. Harris, and D. Patil, "Energy-delay tradeoffs in 32-bit static shifter designs," *Proc. Intl. Conf. Computer Design*, Oct. 2008, pp. 626–632.
- [Hwang89] I. Hwang and A. Fisher, "Ultrafast compact 32-bit CMOS adders in multiple-output domino logic," *JSSC*, vol. 24, no. 2, Apr. 1989, pp. 358–369.
- [Hwang99a] W. Hwang, R. Joshi, and W. Henkels, "A 500-MHz, 32-Word × 64-bit, eight-port self-resetting CMOS register file," *JSSC*, vol. 34, no. 1, Jan. 1999, pp. 56–67.
- [Hwang99b] W. Hwang, G. Gristede, P. Sanda, S. Wang, and D. Heidel, "Implementation of a self-resetting CMOS 64-bit parallel adder with enhanced testability," *JSSC*, vol. 34, no. 8, Aug. 1999, pp. 1108–1117.
- [Hwang02] D. Hwang, F. Dengwei, and A. Willson Jr, "A 400-MHz processor for the efficient conversion of rectangular to polar coordinates for digital communications applications," *Symp. VLSI Circuits Digest Tech. Papers*, Jun. 2002, pp. 248–251.
- [ICKnowledge02] IC Knowledge, "Defect density trends," 2002, www.icknowledge.com/trends/defects.pdf.
- [IEEE1076-08] IEEE Standard 1076-2008 (Revision of IEEE Standard 1076-2002), *VHDL Language Reference Manual*, 2009.
- [IEEE1149.1-01] IEEE Standard 1149.1-2001, *Test Access Port and Boundary-Scan Architecture*, 2001.

- [IEEE1364-01] IEEE Standard 1364-2001, *Verilog Hardware Description Language*, 2001.
- [IEEE 1800-2009] IEEE Standard 1800-2009, *System Verilog-Unified Hardware Design, Specification, and Verification Language*, 2009.
- [Intel10] Intel Corporation, *Microprocessor Quick Reference Guide*, www.intel.com/pressroom/kits/quickreffam.htm, 2010.
- [Isaac08] R. Isaac, "The remarkable story of the DRAM industry," *IEEE SSCS News*, Winter 2008, pp. 45–49.
- [Ishihara04] F. Ishihara, F. Sheikh, and B. Nikolic, "Level conversion for dual-supply systems," *IEEE Trans. VLSI*, vol. 12, no. 2, Feb. 2004, pp. 185–195.
- [Ismail99] Y. Ismail, E. Friedman, and J. Neves, "Figures of merit to characterize the importance of on-chip interconnect," *IEEE Trans. VLSI*, vol. 7, no. 4, Dec. 1999, pp. 442–449.
- [Itoh96] K. Itoh, A. Fridi, A. Bellaouar, and M. Elmasry, "A deep sub-V_t single power-supply SRAM cell with multi-V_t boosted storage node and dynamic load," *Proc. VLSI Circuits Symp.*, Jun. 1996, pp. 132–133.
- [Itoh97] K. Itoh, Y. Nakagome, S. Kimura, and T. Watanabe, "Limitations and challenges of multigigabit DRAM chip design," *JSSC*, vol. 32, no. 5, May 1997, pp. 624–634.
- [Itoh01] N. Itoh, Y. Naemura, H. Makino, Y. Nakase, T. Yoshihara, and Y. Horiba, "A 600-MHz 54×54-bit multiplier with rectangular-styled Wallace tree," *JSSC*, vol. 36, no. 2, Feb. 2001, pp. 249–257.
- [Itoh01k] K. Itoh, *VLSI Memory Chip Design*, Berlin: Springer-Verlag, 2001.
- [Itoh01n] N. Itoh et al., "A 600-MHz 54 × 54-bit multiplier with rectangular-styled Wallace tree," *JSSC*, vol. 36, no. 2, Feb. 2001, pp. 249–257.
- [Itoh09] K. Itoh, "Adaptive circuits for the 0.5-V nanoscale CMOS era," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2009, pp. 14–20.
- [Jackson04] R. Jackson and S. Talwar, "High speed binary addition," *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2004, pp. 1350–1353.
- [Jacoboni77] C. Jacoboni, C. Canali, G. Ottaviani, and A. Alberigi Quaranta, "A review of some charge transport properties of silicon," *Solid-State Electronics*, vol. 20, 1977, pp. 77–89.
- [Jayasumana91] A. Jayasumana, Y. Malaiya, and R. Rajsuman, "Design of CMOS circuits for stuck-open fault testability," *JSSC*, vol. 26, no. 1, Jan. 1991, pp. 58–61.
- [Ji-ren87] Y. Ji-ren, I. Karlsson, and C. Svensson, "A true single-phase-clock dynamic CMOS circuit technique," *JSSC*, vol. SC-22, no. 5, Oct. 1987, pp. 899–901.
- [Johnson88] M. Johnson, "A symmetric CMOS NOR gate for high-speed applications," *JSSC*, vol. SC-23, no. 5, Oct. 1988, pp. 1233–1236.
- [Johnson91] B. Johnson, T. Quarles, A. Newton, D. Pederson, and A. Sangiovanni-Vincentelli, *SPICE3 Version 3e User's Manual*, UC Berkeley, Apr. 1991.
- [Johnston96] A. Johnston, "The influence of VLSI technology evolution on radiation-induced latchup in space systems," *IEEE Trans. Nuclear Science*, vol. 43, no. 2, Apr. 1996, pp. 505–521.
- [Josephson02] D. Josephson, "The manic depression of microprocessor debug," *Proc. Intl. Test Conf.*, 2002, pp. 657–663.
- [Jung01] S. Jung, S. Yoo, K. Kim, and S. Kang, "Skew-tolerant high-speed (STHS) domino logic," *Proc. IEEE Intl. Symp. Circuits and Systems*, 2001, pp. 154–157.
- [Kahng08] A. Kahng and K. Samadi, "CMP fill synthesis: A survey of recent studies," *IEEE Trans. CAD*, vol. 27, no. 1, Jan. 2008, pp. 3–19.
- [Kamon94] M. Kamon, J. Tsuk, and J. White, "FASTHENRY: a multipole-accelerated 3-D inductance extraction program," *IEEE Trans. Microwave Theory and Techniques*, vol. 42, no. 9, Sep. 1994, pp. 1750–1758.
- [Kanamoto07] T. Kanamoto et al., "Impact of well edge proximity effect on timing," *Proc. European Solid State Device Research Conf.*, Sep. 2007, pp. 115–118.

- [Kanda02] K. Kanda, T. Miyazaki, M. Sik, H. Kawaguchi, and T. Sakurai, "Two orders of magnitude leakage power reduction of low voltage SRAMs by row-by-row dynamic V_{DD} control (RRDV) scheme," *Proc. Intl. ASIC/SOC Conf.*, Sep. 2002, pp. 381–385.
- [Kang03] S. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits*, 3rd ed., Boston: McGraw Hill, 2003.
- [Kanj06] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," *Proc. Design Automation Conf.*, 2006, pp. 69–72.
- [Kanno07] Y. Kanno et al., "Hierarchical power distribution with power tree in dozens of power domains for 90-nm low-power multi-CPU SoCs," *JSSC*, vol. 42, no. 1, Jan. 2007, pp. 74–83.
- [Kantabutra91] V. Kantabutra, "Designing optimum carry-skip adders," *Proc. IEEE Symp. Computer Arithmetic*, 1991, pp. 146–153.
- [Kantabutra93] V. Kantabutra, "A recursive carry-lookahead/carry-select hybrid adder," *IEEE Trans. Computers*, vol. 42, no. 12, Dec. 1993, pp. 1495–1499.
- [Kapur02] P. Kapur, J. McVittie, and K. Saraswat, "Technology and reliability constrained future copper interconnects. I. Resistance modeling," *IEEE Trans. Electron Devices*, vol. 49, no. 4, Apr. 2002, pp. 590–597.
- [Karnik01] T. Karnik, B. Bloechel, K. Soumyanath, V. De, and S. Borkar, "Scaling trends of cosmic rays induced soft errors in static latches beyond 0.18 μm ," *Symp. VLSI Circuits Digest Tech. Papers*, 2001, pp. 61–62.
- [Kavaleros06] J. Kavaleros et al., "Tri-gate transistor architecture with high-k gate dielectrics, metal gates and strain engineering," *Proc. VLSI Technology Symp.*, 2006, pp. 50–51.
- [Keating07] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual*, New York: Springer, 2007.
- [Keeth07] B. Keeth, J. Baker, B. Johnson, F. Lin, *DRAM Circuit Design: Fundamental and High-Speed Topics*, IEEE Press, 2007.
- [Keshavarzi01] A. Keshavarzi et al., "Effectiveness of reverse body bias for leakage control in scaled dual V_t CMOS ICs," *Proc. Intl. Symp. Low Power Electronics and Design*, 2001, pp. 207–212.
- [Keyes70] R. Keyes, E. Harris, and K. Konnerth, "The role of low temperatures in the operation of logic circuitry," *Proc. IEEE*, vol. 58, no. 12, Dec. 1970, pp. 1914–1932.
- [Keys75] R. Keyes, "The effect of randomness in the distribution of impurity atoms on FET thresholds," *Applied Physics*, 8, 1975, pp. 251–259.
- [Khalil08] D. Khalil, M. Khellah, Nam-Sung Kim, Y. Ismail, T. Karnik, and V. De, "Accurate estimation of SRAM dynamic stability," *IEEE Trans. VLSI*, vol. 16, no. 12, Dec. 2008, pp. 1639–1647.
- [Khare02] M. Khare et al., "A high performance 90nm SOI technology with 0.992 μm^2 6T-SRAM cell," *Proc. Intl. Electron Devices Meeting*, 2002, pp. 407–410.
- [Khellah06] M. Khellah et al., "Wordline & bitline pulsing schemes for improving SRAM cell stability in low- V_{cc} 65nm CMOS designs," *Proc. VLSI Circuits Symp.*, 2006, pp. 9–10.
- [Khellah07] M. Khellah et al., "A 256-Kb dual-VCC SRAM building block in 65-nm CMOS process with actively clamped sleep transistor," *JSSC*, vol. 42, no. 1, Jan. 2007, pp. 233–242.
- [Khellah09] M. Khellah et al., "Process, temperature, and supply-noise tolerant 45 nm dense cache arrays with diffusion-notch-free (DNF) 6T SRAM cells and dynamic multi- V_{cc} circuits," *JSSC*, vol. 44, no. 4, Apr. 2009, pp. 1199–1208.
- [Kielkowski95] R. Kielkowski, *SPICE: Practical Device Modeling*, Boston: McGraw-Hill, 1995.
- [Kilburn59] T. Kilburn, D. Edwards, and D. Aspinall, "Parallel addition in digital computers—a new fast 'carry' circuit," *Proc. IEE*, vol. 106B, 1959, pp. 460–464.
- [Kim00] J. Kim, Y. Jang, and H. Park, "CMOS sense amplifier-based flip-flop with two N-C2MOS output latches," *Electronics Letters*, vol. 36, no. 6, Mar. 16, 2000, pp. 498–500.
- [Kim03] C. Kim, K. Roy, S. Hsu, A. Alvandpour, R. Krishnamurthy, and S. Borkar, "A process variation compensating technique for sub-90 nm dynamic circuits," *Proc. VLSI Circuits Symp.*, Jun. 2003, pp. 205–206.

- [Kim03b] J. Kim, M. Horowitz, and Gu-Yeon Wei, "Design of CMOS adaptive-bandwidth PLL/DLLs: a general approach," *IEEE Trans. Circuits & Systems*, vol. 50, no. 11, Nov. 2003, pp. 860–869.
- [Kim05] C. Kim, J. Kim, S. Mukhopadhyay, and K. Roy, "A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations," *IEEE Trans. VLSI*, vol. 13, no. 3, Mar. 2005, pp. 349–357.
- [Kim07] J. Kim, K. Jones, and M. Horowitz, "Variable domain transformation for linear PAC analysis of mixed-signal systems," *Proc. Intl. Conf. Computer-Aided Design*, Nov. 2007, pp. 887–894.
- [Kim09] T. Kim, J. Liu, and C. Kim, "A voltage scalable 0.26 V, 64 kb 8T SRAM with $V_{\min}$ lowering techniques and deep sleep mode," *JSSC*, vol. 44, no. 6, Jun. 2009, pp. 1785–1795.
- [Kinniment02] D. Kinniment and E. Chester, "Design of an on-chip random number generator using metastability," *Proc. Intl. Solid-State Circuits Conf.*, Sep. 2002, pp. 595–598.
- [Kio01] S. Kio, L. McMurchie, and C. Sechen, "Application of output prediction logic to differential CMOS," *Proc. IEEE Computer Society Workshop on VLSI*, 2001, pp. 57–65.
- [Kitsukawa93] G. Kitsukawa et al., "256-Mb DRAM circuit technologies for file applications," *JSSC*, vol. 28, no. 11, Nov. 1993, pp. 1105–1113.
- [Klass99] F. Klass et al., "A new family of semidynamic and dynamic flip-flops with embedded logic for high-performance processors," *JSSC*, vol. 34, no. 5, May 1999, pp. 712–716.
- [Klaus98] J. Klaus, A. Ott, A. Dillon, and S. George, "Atomic layer controlled growth of Si_3N_4 films using sequential surface reactions," *Surf. Sci.*, vol. 418, 1998, pp. L14–L19.
- [Knebel98] D. Knebel et al., "Diagnosis and characterization of timing-related defects by time-dependent light emission," *IEEE Intl. Test Conf.*, 1998, pp. 733–739.
- [Knowles01] S. Knowles, "A family of adders," *Proc. IEEE Symp. Computer Arithmetic*, 2001, pp. 277–284.
- [Koenemann79] B. Koenemann, J. Mucha, and G. Zwiehoff, "Built-in logic block observation techniques," *Proc. Intl. Test Conf.*, Oct. 1979, pp. 37–41.
- [Koester08] S. Koester et al., "Wafer-level 3D integration technology," *IBM J. Research and Dev.*, vol. 52, no. 6, Nov. 2008, pp. 583–597.
- [Kogge73] P. Kogge and H. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Computers*, vol. C-22, no. 8, Aug. 1973, pp. 786–793.
- [Koh01] M. Koh et al., "Limit of gate oxide thickness scaling in MOSFETs due to apparent threshold voltage fluctuation induced by tunnel leakage current," *IEEE Trans. Electron Devices*, vol. 48, no. 2, Feb. 2001, pp. 259–264.
- [Konstadinidis09] G. Konstadinidis et al., "Architecture and physical implementation of a third generation 65 nm, 16 core, 32 thread chip-multithreading SPARC processor," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 7–17.
- [Kozu96] S. Kozu et al., "A 100 MHz 0.4W RISC processor with 200 MHz multiply-adder, using pulse-register technique," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1996, pp. 140–141.
- [Krambeck82] R. Krambeck, C. Lee, and H. Law, "High speed compact circuits with CMOS," *JSSC*, vol. SC-17, no. 3, Jun. 1982, pp. 614–619.
- [Kuang05] J. Kuang et al., "A double-precision multiplier with fine-grained clock-gating support for a first-generation CELL processor," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2005, pp. 378–605.
- [Kulkarni04] S. Kulkarni and D. Sylvester, "High performance level conversion for dual V_{DD} design," *IEEE Trans. VLSI*, vol. 12, no. 9, Sep. 2004, pp. 926–936.
- [Kumar94] R. Kumar, "ACMOS: an adaptive CMOS high performance logic," *Electronics Letters*, vol. 30, no. 6, Mar. 1994, pp. 483–484.
- [Kumar01] R. Kumar, "Interconnect and noise immunity design for the Pentium 4 processor," *Intel Technology Journal*, vol. 5, no. 1, Q1 2001, pp. 1–12.
- [Kumar06] R. Kumar and V. Kursun, "Reversed temperature-dependent propagation delay characteristics in nanometer CMOS circuits," *IEEE Trans. Circuits & Systems*, vol. 53, no. 10, Oct. 2006, pp. 1078–1082.

- [Kumar09] R. Kumar and G. Hinton, "A family of 45nm IA processors," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2009, pp. 58–59.
- [Kuo01] J. Kuo and S. Lin, *Low-Voltage SOI CMOS VLSI Devices and Circuits*, New York: Wiley Interscience, 2001.
- [Kurd01] N. Kurd, J. Barkarullah, R. Dizon, T. Fletcher, and P. Madland, "A multigigahertz clocking scheme for the Pentium 4 microprocessor," *JSSC*, vol. 36, no. 11, Nov. 2001, pp. 1647–1653.
- [Kuroda96] T. Kuroda et al., "A 0.9-V, 150-MHz, 10-mW, 4 mm², 2-D discrete cosine transform core processor with variable threshold-voltage (VT) scheme," *JSSC*, vol. 31, no. 11, Nov. 1996, pp. 1770–1779.
- [Kwong06] J. Kwong and A. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," *Proc. Intl. Symp. Low Power Electronics & Design*, Oct. 2006, pp. 8–13.
- [Kwong09] J. Kwong, Y. Ramadass, N. Verma, and A. Chandrakasan, "A 65 nm sub-V_t microcontroller with integrated SRAM and switched capacitor DC-DC converter," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 115–126.
- [Kynett88] V. Kynett et al., "An in-system reprogrammable 256k CMOS flash memory," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 1988, pp. 132–133, 330.
- [Ladner80] R. Ladner and M. Fischer, "Parallel prefix computation," *J. ACM*, vol. 27, no. 4, Oct. 1980, pp. 831–838.
- [Lai97] F. Lai and W. Hwang, "Design and implementation of differential cascode voltage switch with pass-gate (DCVSPG) logic for high-performance digital systems," *JSSC*, vol. 32, no. 4, Apr. 1997, pp. 563–573.
- [LaPedus07] M. LaPedus, "Costs cast ICs into Darwinian struggle," *EE Times*, Mar. 30, 2007.
- [Larsson94] P. Larsson and C. Svensson, "Impact of clock slope on true single phase clocked (TSPC) CMOS circuits," *JSSC*, vol. 29, no. 6, Jun. 1994, pp. 723–726.
- [Lasserre99] F. Lasserre et al., "Laser beam backside probing of CMOS integrated circuits," *Microelectronics and Reliability*, Jun. 1999, vol. 39, no. 6, pp. 957–961.
- [Le06] T. Le, J. Han, A. von Jouanne, K. Mayaram, and T. Fiez, "Piezoelectric micro-power generation interface circuits," *JSSC*, vol. 41, no. 6, Jun. 2006, pp. 1411–1420.
- [Leblebici96] Y. Leblebici, "Design considerations for CMOS digital circuits with improved hot-carrier reliability," *JSSC*, vol. 31, no. 7, Jul. 1996, pp. 1014–1024.
- [Lee86] C. Lee and E. Szeto, "Zipper CMOS," *IEEE Circuits and Systems Magazine*, May 1986, pp. 10–16.
- [Lee92] K. Lee and M. Breuer, "Design and test rules for CMOS circuits to facilitate IDDQ testing of bridging faults," *IEEE Trans. On CAD of Integrated circuits*, vol. 11, no. 5, May 1992, pp. 659–670.
- [Lee98] M. Lee, "A multilevel parasitic interconnect capacitance modeling and extraction for reliable VLSI on-chip clock delay evaluation," *JSSC*, vol. 33, no. 4, Apr. 1998, pp. 657–661.
- [Lee03] D. Lee, W. Kwong, D. Blaauw, and D. Sylvester, "Analysis and minimization techniques for total leakage considering gate oxide leakage," *Proc. Design Automation Conf.*, Jun. 2003, pp. 175–180.
- [Lee05] W. Lee et al., "High performance 65 nm SOI technology with enhanced transistor strain and advanced-low-K BEOL," *Proc. Intl. Electron Devices Meeting*, Dec. 2005.
- [Lee06] L. Lee, D. Weinlader, and C.K. Yang, "A sub-10-ps multiphase sampling system using redundancy," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 265–273.
- [Lehman61] M. Lehman and N. Burla, "Skip technique for high-speed carry-propagation in binary arithmetic units," *IRE Trans. Electronic Computers*, vol. 10, Dec. 1961, pp. 691–698.
- [Leighton92] F. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays; Trees; Hypercubes*, San Francisco: Morgan Kaufmann, 1992.
- [Leon07] A. Leon, K. Tam, J. Shin, D. Weisner, and F. Schumacher, "A power-efficient high-throughput 32-thread SPARC processor," *JSSC*, vol. 42, no. 1, Jan. 2007, pp. 7–16.
- [Lhermet08] H. Lhermet, C. Condemine, M. Plissonnier, R. Salot, P. Audebert, and M. Rosset, "Efficient power management circuit: from thermal energy harvesting to above-IC microbattery energy storage," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 246–255.

- [Liew90] B. Liew, N. Cheung, and C. Hu, "Projecting interconnect electromigration lifetime for arbitrary current waveforms," *IEEE Trans. Electron Devices*, vol. 37, no. 5, May 1990, pp. 1343–1351.
- [Lih07] Y. Lih, N. Tzartzanis, and W. Walker, "A leakage current replica keeper for dynamic circuits," *JSSC*, vol. 42, no. 1, Jan. 2007, pp. 48–55.
- [Lim72] R. Lim, "A barrel switch design," *Computer Design*, Aug. 1972, pp. 76–78.
- [Lim05] D. Lim, J. Lee, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. VLSI*, vol. 13, no. 10, Oct. 2005, pp. 1200–1205.
- [Lin83] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*, Upper Saddle River, NJ: Prentice Hall, 1983.
- [Linderman04] M. Linderman, D. Harris, and D. Diaz, "Bounding bus delay and noise effects of on-chip inductance," *Proc. Workshop on Signal Propagation on Interconnects*, May 2004, pp. 167–170.
- [Ling81] H. Ling, "High-speed binary adder," *IBM J. Research and Development*, vol. 25, no. 3, May 1981, pp. 156–166.
- [Liu01] X. Liu, C. Lee, C. Zhou, and J. Han, "Carbon nanotube field-effect inverters," *Appl. Phys. Letters*, vol. 79, no. 20, Nov. 2001, pp. 3329–3331.
- [Lofstrom00] K. Lofstrom, W. Daasch, and D. Taylor, "IC identification circuit using device mismatch," *Proc. Intl. Solid-State Circuits Conf.*, 2000, pp. 372–373.
- [Lohstroh79] J. Lohstroh, "Static and dynamic noise margins of logic circuits," *JSSC*, vol. SC-14, no. 3, Jun. 1979, pp. 591–598.
- [Lohstroh83] J. Lohstroh, E. Seevinck, and J. de Groot, "Worst-case static noise margin criteria for logic circuits and their mathematical equivalence," *JSSC*, vol. SC-18, no. 6, Dec. 1983, pp. 803–807.
- [Lu88] S. Lu, "Implementation of iterative networks with CMOS differential logic," *JSSC*, vol. 23, no. 4, Aug. 1988, pp. 1013–1017.
- [Lu91] S. Lu and M. Ercegovic, "Evaluation of two-summand adders implemented in ECDL CMOS differential logic," *JSSC*, vol. 26, no. 8, Aug. 1991, pp. 1152–1160.
- [Lu93] F. Lu, H. Samueli, J. Yuan, and C. Svensson, "A 700 MHz 24-b pipelined accumulator in 1.2- μ m CMOS for application as a numerically controlled oscillator," *JSSC*, vol. 28, no. 8, Aug. 1993, pp. 878–886.
- [Lu93b] F. Lu and H. Samueli, "A 200-MHz CMOS pipelined multiplier-accumulator using a quasi-domino dynamic full-adder cell design," *JSSC*, vol. 28, no. 2, Feb. 1993, pp. 123–132.
- [Lu08] S. Lu, S. Hsu, and D. Somasekhar, "Memory arrays circuits for computer architects," *IEEE Micro Tutorial*, 2008.
- [Lynch92] T. Lynch and E. Swartzlander, "A spanning tree carry lookahead adder," *IEEE Trans. Computers*, vol. 41, no. 8, Aug. 1992, pp. 931–939.
- [Lyon87] R. Lyon and R. Schediwy, "CMOS static memory with a new four-transistor memory cell," *Proc. Advanced Research in VLSI*, Mar. 1987, pp. 111–132.
- [Lyons62] R. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM Journal*, Apr. 1962, pp. 200–209.
- [Ma94] S. Ma and P. Franzon, "Energy control and accurate delay estimation in the design of CMOS buffers," *JSSC*, vol. 29, no. 9, Sep. 1994, pp. 1150–1153.
- [Mack08] C. Mack, "Seeing double," *IEEE Spectrum*, vol. 45, no. 11, Nov. 2008, pp. 46–51.
- [MacSorley61] O. MacSorley, "High-Speed arithmetic in binary computers," *Proc. IRE*, vol. 49, pt. 1, Jan. 1961, pp. 67–91.
- [Mahalingam85] M. Mahalingam, "Thermal management in semiconductor device packages," *Proc. IEEE Custom Integrated Circuits Conf.*, 1985, pp. 46–49.
- [Mai05] K. Mai et al., "Architecture and circuit techniques for a 1.1-GHz 16-kb reconfigurable memory in 0.18- μ m CMOS," *JSSC*, vol. 40, no. 1, Jan. 2005, pp. 261–275.
- [Maier97] C. Maier et al., "A 533-MHz BiCMOS superscalar RISC microprocessor," *JSSC*, vol. 32, no. 11, Nov. 1997, pp. 1625–1634.

- [Majerski67] S. Majerski, "On determination of optimal distributions of carry skips in adders," *IEEE Trans. Electronic Computers*, vol. EC-16, no. 1, 1967, pp. 45–58.
- [Maksimović00] D. Maksimović, V. Oklobdzija, B. Nikolic, and K. Current, "Clocked CMOS adiabatic logic with integrated single-phase power-clock supply," *IEEE Trans. VLSI*, vol. 8, no. 4, Aug. 2000, pp. 460–463.
- [Maluf04] N. Maluf and K. Williams, *An Introduction to Microelectromechanical Systems*, 2nd ed., Norwood, MA: Artech House, 2004.
- [Maneatis03] J. Maneatis, I. McClatchie, J. Maxey, and M. Shankaradas, "Self-biased high-bandwidth low-jitter 1-to-4096 multiplier clock generator PLL," *JSSC*, vol. 38, no. 11, Nov. 2003, pp. 1795–1803.
- [Markovic04] D. Markovic, V. Stojanovic, B. Nikolic, M. Horowitz, and R. Brodersen, "Methods for true energy-performance optimization," *JSSC*, vol. 39, no. 8, Aug. 2004, pp. 1282–1293.
- [Masuoka84] F. Masuoka, M. Asano, H. Iwahashi, T. Komuro, and S. Tanaka, "A new flash E2PROM cell using triple polysilicon technology," *Proc. Intl. Electron Devices Meeting*, 1984, pp. 464–467.
- [Masuoka87] F. Masuoka, M. Momodomi, Y. Iwata, and R. Shirota, "New ultra high density EPROM and flash EEPROM with NAND structure cell," *Proc. Intl. Electron Devices Meeting*, 1987, pp. 552–555.
- [Mathew03] S. Mathew, M. Anders, R. Krishnamurthy, and S. Borkar, "A 4-GHz 130-nm address generation unit with 32-bit sparse-tree adder core," *JSSC*, vol. 38, no. 5, May 2003, pp. 689–695.
- [Mathew05] S. Mathew, M. Anders, B. Bloechel, Trang Nguyen, R. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," *JSSC*, vol. 40, no. 1, Jan. 2005, pp. 44–51.
- [Matsui94] M. Matsui et al., "A 200 MHz 13 mm² 2-D DCT macrocell using sense-amplifier pipeline flip-flop scheme," *JSSC*, vol. 29, no. 12, Dec. 1994, pp. 1482–1490.
- [May79] T. May and M. Woods, "Alpha-particle-induced soft errors in dynamic memories," *IEEE Trans. Electron Devices*, vol. ED-26, no. 1, Jan. 1979, pp. 2–9.
- [McGowen06] R. McGowen et al., "Power and temperature control on a 90-nm Itanium family processor," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 229–237.
- [McMurchie00] L. McMurchie, S. Kio, G. Yee, T. Thorp, and C. Sechen, "Output prediction logic: a high-performance CMOS design technique," *Proc. Intl. Conf. Computer Design*, 2000, pp. 247–254.
- [Mead80] C. Mead and L. Conway, *Introduction to VLSI Systems*, Reading, MA: Addison-Wesley, 1980.
- [Mears96] J. Mears, "Transmission line RAPIDDESIGNER operation and applications guide," National Semiconductor Application Note 905, May 1996, www.national.com/an/AN/AN-905.pdf.
- [Mehta99] G. Mehta, D. Harris, and D. Singh, "Pulsed Domino Latches," US Patent 5,880,608, 1999.
- [Meier99] N. Meier, T. Marieb, P. Flinn, R. Gleixner, and J. Bravman, "In-situ studies of electromigration voiding in passivated copper interconnects," *AIP Conf. Proc. 491*, Fifth Intl. Workshop on Stress-Induced Phenomena in Metallization, Jun. 1999, p. 180.
- [Meijs84] N. van der Meijs, and J. Fokkema, "VLSI circuit reconstruction from mask topology," *Integration, The VLSI Journal*, vol. 2, no. 2, Jun. 1984, pp. 85–119.
- [Meindl00] J. Meindl and J. Davis, "The fundamental limit on binary switching energy for terascale integration (TSI)," *JSSC*, vol. 35, no. 10, Oct. 2000, pp. 1515–1516.
- [Meng08] X. Meng, R. Saleh, and K. Arabi, "Layout of decoupling capacitors in IP blocks for 90-nm CMOS," *IEEE Trans. VLSI*, vol. 16, no. 11, Nov. 2008, pp. 1581–1588.
- [Merchant01] S. Merchant, S. Kang, M. Sangneria, B. van Schravendijk, and T. Mountsier, "Copper interconnects for semiconductor devices," *JOM: Journal of the Minerals, Metals, and Materials Society*, vol. 53, no. 6, Jun. 2001, pp. 43–48.
- [Messerschmitt90] D. Messerschmitt, "Synchronization in digital system design," *IEEE J. Selected Areas Communications*, vol. 8, no. 8, Oct. 1990, pp. 1404–1419.
- [Min06] K. Min, H. Choi, H. Choi, H. Kawaguchi, and T. Sakurai, "Leakage-suppressed clock-gating circuit with Zigzag Super Cut-off CMOS (ZSCCMOS) for leakage-dominant sub-70-nm and sub-1-V_{DD} LSIs," *IEEE Trans. VLSI*, vol. 14, no. 4, Apr. 2006, pp. 430–435.

- [Misaka96] A. Misaka, A. Goda, K. Matsuoka, H. Umimoto, and S. Odanaka, "A statistical critical dimension control at CMOS cell level," *Proc. Intl. Electron Devices Meeting*, Dec. 1996, pp. 631–634.
- [Mistry07] K. Mistry et al., "A 45nm logic technology with high-k+ metal gate transistors, strained silicon, 9 Cu interconnect layers, 193nm dry patterning, and 100% Pb-free packaging," *Proc. Intl. Electron Devices Meeting*, Dec. 2007, pp. 247–250.
- [Mitra05] S. Mitra, T. Karnik, N. Seifert, and Ming Zhang, "Logic soft errors in sub-65nm technologies design and CAD challenges," *Proc. Design Automation Conf.*, Jun. 2005, pp. 2–4.
- [Mizuno94] T. Mizuno, J. Okumtura, and A. Toriumi, "Experimental study of threshold voltage fluctuation due to statistical variation of channel dopant number in MOSFET's," *IEEE Trans. Electron Devices*, vol. 41, no. 11, Nov. 1994, pp. 2216–2221.
- [Moazzami90] R. Moazzami and C. Hu, "Projecting gate oxide reliability and optimizing reliability screens," *IEEE Trans. Electron Devices*, vol. 37, no. 7, Jul. 1990, pp. 1643–1650.
- [Monsieur01] F. Monsieur, E. Vincent, D. Roy, S. Bruyre, G. Pananakakis, and G. Ghibaudo, "Time to breakdown and voltage to breakdown modeling for ultra-thin oxides ($T_{ox} < 32\text{\AA}$)," *Proc. Intl. Integrated Reliability Workshop*, 2001, pp. 20–25.
- [Montanaro96] J. Montanaro et al., "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *JSSC*, vol. 31, no. 11, Nov. 1996, pp. 1703–1714.
- [Montoye90] R. Montoye, P. Cook, E. Hokenek, and R. Havreluk, "An 18 ns 56-bit multiply-adder circuit," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 1990, pp. 46–47.
- [Moon08] P. Moon et al., "Process and electrical results for the on-die interconnect stack for Intel's 45nm process generation," *Intel Technology Journal*, vol. 12, no. 2, Jun. 2008, pp. 87–92.
- [Moore65] G. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, Apr. 1965.
- [Moore03] G. Moore, "No exponential is forever: but 'forever' can be delayed!" *Proc. IEEE Intl. Solid-State Circuits Conf.*, 2003, pp. 1–19.
- [Morgan59] C. Morgan and D. Jarvis, "Transistor logic using current switching routing techniques and its application to a fast carry-propagation adder," *Proc. IEE*, vol. 106B, 1959, pp. 467–468.
- [Morgenshtein09] A. Morgenshtein, E. Friedman, R. Ginosar, and A. Kolodny, "Unified logical effort—a method for delay evaluation and minimization in logic paths with RC interconnect," to appear in *IEEE Trans. VLSI*, 2010.
- [Mori91] J. Mori et al., "A 10 ns 54×54 b parallel structured full array multiplier with 0.5 μm CMOS technology," *JSSC*, vol. 26, no. 4, Apr. 1991, pp. 600–606.
- [Morita06] Y. Morita et al., "A V_{th} -variation-tolerant SRAM with 0.3-V minimum operation voltage for memory-rich SoC under DVS environment," *Proc. VLSI Circuits Symp.*, 2006, pp. 13–14.
- [Morrison61] P. Morrison and E. Morrison, eds., *Charles Babbage: On the Principles and Development of the Calculator*, New York: Dover, 1961.
- [Morton99] S. Morton, "On-chip inductance issues in multiconductor systems," *Proc. Design Automation Conf.*, 1999, pp. 921–926.
- [Mou90] Z. Mou and F. Jutand, "A class of close-to-optimum adder trees allowing regular and compact layout," *Proc. IEEE Intl. Conf. on Computer Design*, 1990, pp. 251–254.
- [Mukhopadhyay03] S. Mukhopadhyay, C. Neau, R. Cakici, A. Agarwal, C. Kim, and K. Roy, "Gate leakage reduction for scaled devices using transistor stacking," *IEEE Trans. VLSI*, vol. 11, no. 4, Aug. 2003, pp. 716–730.
- [Mukhopadhyay05] S. Mukhopadhyay, A. Raychowdhury, and K. Roy, "Accurate estimation of total leakage in nanometer-scale bulk CMOS circuits based on device geometry and doping profile," *IEEE Trans. CAD*, vol. 24, no. 3, Mar. 2005, pp. 363–381.
- [Mule02] A. Mule, E. Glytsis, T. Gaylord, and J. Meindl, "Electrical and optical clock distribution networks for gigascale microprocessors," *IEEE Trans. VLSI*, vol. 10, no. 5, Oct. 2002, pp. 582–594.
- [Muller03] R. Muller, T. Kamins, and M. Chan, *Device Electronics for Integrated Circuits*, 3rd ed., New York: John Wiley & Sons, 2003.

- [Muller08] M. Muller, "Embedded processing at the heart of life and style," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2008, pp. 32–37.
- [Murabayashi96] F. Murabayashi et al., "2.5 V CMOS circuit techniques for a 200 MHz superscalar RISC processor," *JSSC*, vol. 31, no. 7, Jul. 1996, pp. 972–980.
- [Mutoh95] S. Mutoh et al., "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *JSSC*, vol. 30, no. 8, Aug. 1995, pp. 847–854.
- [Mutoh99] S. Mutoh, S. Shigematsu, Y. Gotoh, and S. Konaka, "Design method of MTCMOS power switch for low-voltage high-speed LSIs," *Proc. Design Automation Conf.*, Jan. 1999, pp. 113–116.
- [Myny09] K. Myny et al., "A 128b organic RFID transponder chip, including Manchester encoding and ALOHA anti-collision protocol, operating with a data rate of 1529b/s," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2009, pp. 206–207.
- [Na02] M. Na, E. Nowak, W. Haensch, and J. Cai, "The effective drive current in CMOS inverters," *Proc. Intl. Electron Devices Meeting*, 2002, pp. 121–124.
- [Nabors92] K. Nabors, S. Kim, and J. White, "Fast capacitance extraction of general three-dimensional structures," *IEEE Trans. Microwave Theory and Techniques*, vol. 40, no. 7, Jul. 1992, pp. 1496–1506.
- [Nadig77] H. Nadig, "Signature analysis—concepts, examples and guidelines," *Hewlett Packard Journal*, vol. 28, no. 9, May 1977, pp. 15–21.
- [Naffziger96] S. Naffziger, "A subnanosecond 0.5 μ m 64b adder design," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1996, pp. 362–363.
- [Naffziger98] S. Naffziger, "High speed addition using Ling's equations and dynamic CMOS logic," US Patent 5,719,803, 1998.
- [Naffziger02] S. Naffziger, G. Colon-Bonet, T. Fischer, R. Riedlinger, T. Sullivan, and T. Grutkowski, "The implementation of the Itanium 2 microprocessor," *JSSC*, vol. 37, no. 11, Nov. 2002, pp. 1448–1460.
- [Naffziger06] S. Naffziger et al., "The implementation of a 2-core, multi-threaded Itanium family processor," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 197–209.
- [Naffziger06b] S. Naffziger, "High-performance processors in a power-limited world," *Proc. VLSI Circuits Symp.*, 2006, pp. 93–97.
- [Nagel75] L. Nagel, *SPICE2: a computer program to simulate semiconductor circuits*, Memo ERL-M520, Dept. of Electrical Engineering and Computer Science, University of California at Berkeley, May 9, 1975.
- [Najm07] F. Najm, N. Menezes, and I. Ferzli, "A yield model for integrated circuits and its application to statistical timing analysis," *IEEE Trans. CAD*, vol. 26, no. 3, Mar. 2007, pp. 574–591.
- [Nakagome03] Y. Nakagome, M. Horiguchi, T. Kawahara, and K. Itoh, "Review and future prospects of low-voltage RAM circuits," *IBM J. Research and Dev.*, vol. 47, no. 5/6, Sep./Nov. 2003, pp. 525–552.
- [Nalamalpu02] A. Nalamalpu, S. Srinivasan, and W. Burleson, "Boosters for driving long on-chip interconnects—design issues, interconnect synthesis, and comparison with repeaters," *IEEE Trans. Computer-Aided Design*, vol. 21, no. 1, Jan. 2002, pp. 50–62.
- [Nambu98] H. Nambu et al., "A 1.8-ns access, 550-MHz, 4.5-Mb CMOS SRAM," *JSSC*, vol. 33, no. 11, Nov. 1998, pp. 1650–1658.
- [Narasimha06] S. Narasimha et al., "High performance 45-nm SOI technology with enhanced strain, porous low-k BEOL, and immersion lithography," *Proc. Intl. Electron Devices Meeting*, Dec. 2006, pp. 1–4.
- [Narayanan96] V. Narayanan, B. Chappell, and B. Fleischer, "Static timing analysis for self-resetting circuits," *Proc. Intl. Conf. Computer-Aided Design*, 1996, pp. 119–126.
- [Narendra99] S. Narendra, D. Antoniadis, and V. De, "Impact of using adaptive body bias to compensate die-to-die V_t variation on within-die V_t variation," *Proc. Intl. Symp. Low Power Electronics and Design*, 1999, pp. 229–232.
- [Narendra01] S. Narendra, S. Borkar, V. De, D. Antoniadis, and A. Chandrakasan, "Scaling of stack effect and its application for leakage reduction," *Proc. Intl. Symp. Low Power Electronics and Design*, 2001, pp. 195–200.

- [Narendra03] S. Narendra, A. Keshavarzi, B. Bloechel, S. Borkar, and V. De, "Forward body bias for microprocessors in 130-nm technology generation and beyond," *JSSC*, vol. 38, no. 5, May 2003, pp. 696–701.
- [Narendra06] S. Narendra and A. Chandrakasan, *Leakage in Nanometer CMOS Technologies*, New York: Springer, 2006.
- [Natarajan08] S. Natarajan et al., "A 32 nm logic technology featuring 2nd-generation high-k + metal-gate transistors, enhanced channel strain and 0.171 μm^2 SRAM cell size in a 291 Mb array," *Proc. Intl. Electron Devices Meeting*, Dec. 2008, pp. 1–3.
- [National08] National Semiconductor, *LVDS Owner's Manual*, 4th ed., 2008, www.national.com/LVDS.
- [Nawathe08] U. Nawathe, M. Hassan, K. Yen, A. Kumar, A. Ramachandran, and D. Greenhill, "Implementation of an 8-core, 64-thread, power-efficient SPARC server on a chip," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 6–20.
- [Needham98] W. Needham, C. Prunty, and E. Yeoh, "High volume microprocessor test escapes, an analysis of defects our tests are missing," *Proc. Intl. Test Conf.*, 1998, pp. 25–34.
- [Ng96] P. Ng, P. Balsara, and D. Steiss, "Performance of CMOS differential circuits," *JSSC*, vol. 31, no. 6, Jun. 1996, pp. 841–846.
- [Nii04] K. Nii et al., "A 90-nm low-power 32-kB embedded SRAM with gate leakage suppression circuit for mobile applications," *JSSC*, vol. 39, no. 4, Apr. 2004, pp. 684–693.
- [Nikolić00] B. Nikolić, V. Oklobdzija, V. Stojanović, W. Jia, J. Chiu, and M. Leung, "Improved sense-amplifier-based flip-flop: design and measurements," *JSSC*, vol. 35, no. 6, Jun. 2000, pp. 876–884.
- [NIST02] National Institute of Standards and Technology, "Security requirements for cryptographic modules," FIPS 140-2, 2001.
- [Noguchi07] K. Noguchi and M. Nagata, "An on-chip multichannel waveform monitor for diagnosis of systems-on-a-chip integration," *IEEE Trans. VLSI*, vol. 15, no. 10, Oct. 2007, pp. 1101–1110.
- [Noice83] D. Noice, *A clocking discipline for two-phase digital integrated circuits*, Stanford University Technical Report, Jan. 1983.
- [Northrop99] G. Northrop et al., "609 MHz G5 S/399 microprocessor," *Proc. Intl. Solid-State Circuits Conf.*, 1999, pp. 88–89.
- [Nose00a] K. Nose and T. Sakurai, "Analysis and future trend of short-circuit power," *IEEE Trans. CAD*, vol. 19, no. 9, Sep. 2000, pp. 1023–1030.
- [Nose00b] K. Nose, Soo-Ik Chae, and T. Sakurai, "Voltage dependent gate capacitance and its impact in estimating power and delay of CMOS digital circuits with low supply voltage," *Proc. Intl. Symp. Low Power Electronics & Design*, 2000, pp. 228–230.
- [Nose00c] K. Nose and T. Sakurai, "Optimization of V_{DD} and V_{TH} for low-power and high-speed applications," *Proc. Design Automation Conf.*, 2000, pp. 469–474.
- [Nose06] K. Nose, M. Kajita, and M. Mizuno, "A 1ps-resolution jitter-measurement macro using interpolated jitter oversampling," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2006, pp. 2112–2121.
- [Nowka98] K. Nowka and T. Galambos, "Circuit design techniques for a gigahertz integer microprocessor," *Proc. Intl. Conf. Computer Design*, 1998, pp. 11–16.
- [Oh06] H. Oh et al., "A fully pipelined single-precision floating-point unit in the synergistic processor element of a CELL processor," *JSSC*, vol. 41, no. 4, Apr. 2006, pp. 759–771.
- [Ohbayashi07] S. Ohbayashi et al., "A 65-nm SoC embedded 6T-SRAM designed for manufacturability with read and write operation stabilizing circuits," *JSSC*, vol. 42, no. 4, Apr. 2007, pp. 820–829.
- [Ohkubo95] N. Ohkubo et al., "A 4.4 ns CMOS 54×54 -b multiplier using pass-transistor multiplexer," *JSSC*, vol. 30, no. 3, Mar. 1995, pp. 251–257.
- [Oklobdzija96] V. Oklobdzija, D. Villeger, and S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Trans. Computers*, vol. 45, no. 3, Mar. 1996, pp. 294–306.

- [Oklobdzija85] V. Oklobdzija and E. Barnes, "Some optimal schemes for ALU implementation in VLSI technology," *Proc. Computer Arithmetic Symp.*, 1985, pp. 137–143.
- [Oklobdzija86] V. Oklobdzija and R. Montoye, "Design-performance trade-offs in CMOS-domino logic," *JSSC*, vol. SC-21, no. 2, April 1986, pp. 304–309.
- [Oklobdzija05] V. Oklobdzija, B. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," *IEEE Trans. VLSI*, vol. 13, no. 6, Jun. 2005, pp. 754–758.
- [Ortiz-Conde02] A. Ortiz-Conde, F. Sánchez, J. Liou, A. Cerdeira, M. Estrada, and Y. Yue, "A review of recent MOSFET threshold voltage extraction methods," *Microelectronics Reliability*, vol. 42, 2002, pp. 583–596.
- [Osada01] K. Osada et al., "Universal- V_{DD} 0.65-2.0-V 32-kB cache using a voltage-adapted timing-generation scheme and a lithographically symmetrical cell," *JSSC*, vol. 36, no. 11, Nov. 2001, pp. 1738–1744.
- [Osada04] K. Osada, K. Yamaguchi, Y. Saitoh, and T. Kawahara, "SRAM immunity to cosmic-ray-induced multierrors based on analysis of an induced parasitic bipolar effect," *JSSC*, vol. 39, no. 5, May 2004, pp. 827–833.
- [Osada06] K. Osada, "Reviews and prospects of nanoscale SRAMs," *Proc. Intl Conf. Integrated Circuit Design & Tech.*, 2006, pp. 1–8.
- [Pagiamtzis06] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: a tutorial and survey," *JSSC*, vol. 41, no. 3, Mar. 2006, pp. 712–727.
- [Paik96] W. Paik, H. Ki, and S. Kim, "Push-pull pass-transistor logic family for low voltage and low power," *Proc. 22nd European Solid-State Circuits Conf.*, 1996, pp. 116–119.
- [Paik08] P. Paik, V. Pamula, and K. Chakrabarty, "Adaptive cooling of integrated circuits using digital microfluidics," *IEEE Trans. VLSI*, vol. 16, no. 4, Apr. 2008, pp. 432–443.
- [Parameswar96] A. Parameswar, H. Hara, and T. Sakurai, "A swing restored pass-transistor logic-based multiply and accumulate circuit for multimedia applications," *JSSC*, vol. 31, no. 6, Jun. 1996, pp. 804–809.
- [Paraskevopoulos87] D. Paraskevopoulos and C. Fey, "Studies in LSI technology economics III: design schedules for application-specific integrated circuits," *JSSC*, vol. SC-22, no. 2, Apr. 1987, pp. 223–229.
- [Park00] J. Park, H. Ngo, J. Silberman, and S. Dhong, "470 ps 64-bit parallel binary adder," *Proc. VLSI Circuits Symp.*, 2000, pp. 192–193.
- [Parker03] K. Parker, *The Boundary-Scan Handbook*, Boston: Kluwer Academic Publishers, 2003.
- [Partovi96] H. Partovi et al., "Flow-through latch and edge-triggered flip-flop hybrid elements," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1996, pp. 138–139.
- [Pasternak87] J. Pasternak, A. Shubat, and C. Salama, "CMOS differential pass-transistor logic design," *JSSC*, vol. SC-22, no. 2, Apr. 1987, pp. 216–222.
- [Pasternak91] J. Pasternak and C. Salama, "Design of submicrometer CMOS differential pass-transistor logic circuits," *JSSC*, vol. 26, no. 9, Sep. 1991, pp. 1249–1258.
- [Patil07] D. Patil, O. Azizi, M. Horowitz, R. Ho, and R. Ananthraman, "Robust energy-efficient adder topologies," *Proc. Computer Arithmetic Symp.*, Jun. 2007, pp. 16–28.
- [Patil09] N. Patil, Jie Deng, S. Mitra, and H. Wong, "Circuit-level performance benchmarking and scalability analysis of carbon nanotube transistor circuits," *IEEE Trans. Nanotechnology*, vol. 8, no. 1, Jan. 2009, pp. 37–45.
- [Patterson04] D. Patterson and J. Hennessy, *Computer Organization and Design*, 3rd ed., San Francisco, CA: Morgan Kaufmann, 2004.
- [Paul02] B. Paul and K. Roy. Testing cross-talk induced delay faults in static CMOS circuit through dynamic timing analysis. *Proc. Intl. Test Conf.*, Oct. 2002, pp. 384–390.
- [Paul07] B. Paul, Kunhyuk Kang, H. Kufluoglu, M. Alam, and K. Roy, "Negative bias temperature instability: estimation and design for improved reliability of nanoscale circuits," *IEEE Trans. CAD*, vol. 26, no. 4, Apr. 2007, pp. 743–751.
- [Pedroni10] V. Pedroni, *Circuit Design and Simulation with VHDL*, 2nd ed., Cambridge, MA: MIT Press, 2010.

- [Pelgrom89] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *JSSC*, vol. 24, no. 5, Oct. 1989, pp. 1433–1440.
- [Peng02] C. Peng et al., "A 90 nm generation copper dual damascene technology with ALD TaN barrier," *Tech. Digest Intl. Electron Devices Meeting*, Dec. 2002, pp. 603–606.
- [Penney72] W. Penney and L. Lau, *MOS Integrated Circuits*, New York: Van Nostrand Reinhold, 1972.
- [Perry05] D. Perry and H. Foster, *Applied Formal Verification*, New York: McGraw-Hill, 2005.
- [Pertijs06] M. Pertijs and J. Huijsing, *Precision Temperature Sensors in CMOS Technology*, New York: Springer, 2006.
- [Petegem94] W. van Petegem, B. Geeraerts, W. Sansen, and B. Graindourze, "Electrothermal simulation and design of integrated circuits," *JSSC*, vol. 29, no. 2, Feb. 1994, pp. 143–146.
- [Pfenning85] L. Pfenning, W. Mol, J. Bastiens, and J. van Dijk, "Differential split-level CMOS logic for subnanosecond speeds," *JSSC*, vol. SC-20, no. 5, Oct. 1985, pp. 1050–1055.
- [Pham06] D. Pham et al., "Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 179–196.
- [Pihl98] J. Pihl, "Single-ended swing restoring pass transistor cells for logic synthesis and optimization," *Proc. IEEE Intl. Symp. Circuits and Systems*, vol. 2, 1998, pp. 41–44.
- [Piña02] C. Piña, "Evolution of the MOSIS VLSI educational program," *Proc. Electronic Design, Test, and Applications Workshop*, 2002, pp. 187–191.
- [Plass07] D. Plass and Y. Chan, "IBM POWER6 SRAM arrays," *IBM J. Research and Dev.*, vol. 51, no. 6, Nov. 2007, pp. 747–756.
- [Pollack99] F. Pollack, "New microarchitectural challenges in the coming generations of CMOS process technologies," *Intl. Symp. Microarchitecture*, Keynote address, 1999.
- [Pretorius86] J. Pretorius, A. Shubat, and A. Salama, "Latched domino CMOS logic," *JSSC*, vol. SC-21, no. 4, Aug. 1986, pp. 514–522.
- [Price95] D. Price, "Pentium FDIV flaw—lessons learned," *IEEE Micro*, vol. 15, no. 2, Apr. 1995, pp. 86–88.
- [Proakis08] J. Proakis and M. Salehi, *Digital Communications*, New York: McGraw Hill, 2008.
- [Proebsting91] R. Proebsting, "Speed enhancement technique for CMOS circuits," US Patent 4,985,643, 1991.
- [Quach92] N. Quach and M. Flynn, "High-speed addition in CMOS," *IEEE Trans. Computers*, vol. 41, no. 12, Dec. 1992, pp. 1612–1615.
- [Quader94] K. Quader, E. Minami, W. Huang, P. Ko, and C. Hu, "Hot-carrier-reliability design guidelines for CMOS logic circuits," *JSSC*, vol. 29, no. 3, Mar. 1994, pp. 253–262.
- [Ramadass10] Y. Ramadass and A. Chandrakasan, "An efficient piezoelectric energy harvesting interface circuit using a bias-flip rectifier and shared inductor," *JSSC*, vol. 45, no. 1, Jan. 2010, pp. 189–204.
- [Rao03] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical estimation of leakage current considering inter- and intra-die process variation," *Proc. Intl. Symp. Low Power Electronics & Design*, Aug. 2003, pp. 84–89.
- [Rao07] R. R. Rao, K. Chopra, D. Blaauw, and D. Sylvester, "Computing the soft error rate of a combinational logic circuit using parameterized descriptors," *IEEE Trans. CAD*, vol. 26, no. 3, Mar. 2007, pp. 468–479.
- [Raychowdhury07] A. Raychowdhury and K. Roy, "Carbon nanotube electronics: design of high-performance and low-power digital circuits," *IEEE Trans. Circuits & Systems*, vol. 54, no. 11, Nov. 2007, pp. 2391–2401.
- [Razavi03] B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw Hill, 2003.
- [Reddy02] V. Reddy et al., "Impact of negative bias temperature instability on digital circuit reliability," *Proc. 40th IEEE Intl. Reliability Physics Symp.*, 2002, pp. 248–254.
- [Restle01] P. Restle et al., "A clock distribution network for microprocessors," *JSSC*, vol. 36, no. 5, May 2001, pp. 792–799.
- [Restle98] P. Restle and A. Deutsch, "Designing the best clock distribution network," *Symp. VLSI Circuits Digest Tech. Papers*, 1998, pp. 2–5.

- [Riordan97] M. Riordan and L. Hoddeson, *Crystal Fire: The Invention of the Transistor and the Birth of the Information Age*, New York: W. W. Norton & Co, 1998.
- [Rizzolo07] R. Rizzolo, "IBM System z9 eFUSE applications and methodology," *IBM J. Research and Dev.*, vol. 51, no. 1/2, Jan./Mar. 2007, pp. 65–75.
- [Rohrer05] N. Rohrer et al., "A 64-bit microprocessor in 130-nm and 90-nm technologies with power management features," *JSSC*, vol. 40, no. 1, Jan. 2005, pp. 19–27.
- [Rotella02] F. Rotella, V. Blaschke, and D. Howard, "A broad-band scalable lumped-element inductor model using analytic expressions to incorporate skin effect, substrate loss, and proximity effect," *Tech. Digest Intl. Electron Devices Meeting*, Dec. 2002, pp. 471–474.
- [Roy03] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, Feb. 2003, pp. 305–327.
- [Ruehli73] A. Ruehli and P. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-21, no. 2, Feb. 1973, pp. 76–82.
- [Rusu00] S. Rusu and G. Singer, "The first IA-64 microprocessor," *JSSC*, vol. 35, no. 11, Nov. 2000, pp. 1539–1544.
- [Rusu03] S. Rusu, J. Stinson, S. Tam, J. Leung, H. Muljono, and B. Cherkauer, "A 1.5-GHz 130-nm Itanium 2 processor with 6-MB on-die L3 cache," *JSSC*, vol. 38, no. 11, Nov. 2003, pp. 1887–1895.
- [Rusu07] S. Rusu et al., "A 65-nm dual-core multithreaded Xeon processor with 16-MB L3 cache," *JSSC*, vol. 42, no. 1, Jan. 2007, pp. 17–25.
- [Rusu10] S. Rusu et al., "A 45 nm 8-core Enterprise Xeon Processor," *JSSC*, vol. 45, no. 1, Jan. 2010, pp. 7–14.
- [Rzepka98] S. Rzepka, K. Banerjee, E. Meusel, and C. Hu, "Characterization of self-heating in advanced VLSI interconnect lines based on thermal finite element simulation," *IEEE Trans. Components, Packaging, and Manufacturing Technology—Part A*, vol. 21, no. 3, Sep. 1998, pp. 406–411.
- [Sah64] C. Sah, "Characteristics of the Metal-Oxide-Semiconductor Transistors," *IEEE Trans. Electron Devices*, ED-11, Jul. 1964, pp. 324–345.
- [Saint02] C. Saint and J. Saint, *IC Mask Design: Essential Layout Techniques*, New York: McGraw-Hill, 2002.
- [Sakran07] N. Sakran, M. Yuffe, M. Mehalel, J. Doweck, E. Knoll, and A. Kovacs, "The implementation of the 65nm dual-core 64b Merom processor," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2007, pp. 106–107, 590.
- [Sakurai83] T. Sakurai, "Approximation of wiring delay in MOSFET LSI," *JSSC*, vol. SC-18, no. 4, Aug. 1983, pp. 418–426.
- [Sakurai86] T. Sakurai, K. Nogami, M. Kakumu, and T. Iizuka, "Hot-carrier generation in submicrometer VLSI environment," *JSSC*, vol. SC-21, no. 1, Feb. 1986, pp. 187–192.
- [Sakurai90] T. Sakurai and R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *JSSC*, vol. 25, no. 2, April 1990, pp. 584–594.
- [Sakurai91] T. Sakurai and A. Newton, "Delay analysis of series-connected MOSFET circuits," *JSSC*, vol. 26, no. 2, Feb. 1991, pp. 122–131.
- [Salib04] M. Salib et al., "Silicon photonics," *Intel Technology Journal*, vol. 08, no. 2, May 2004, pp. 143–160.
- [Samson08] G. Samson, N. Ananthapadmanabhan, S. Badrudduza, and L. Clark, "Low-power dynamic memory word line decoding for static random access memories," *JSSC*, vol. 43, no. 11, Nov. 2008, pp. 2524–2532.
- [Samson09] G. Samson and L. Clark, "Low-power race-free programmable logic arrays," *JSSC*, vol. 44, no. 3, Mar. 2009, pp. 935–946.
- [Santoro89] M. Santoro, "Design and Clocking of VLSI Multipliers," Ph.D. thesis, Stanford University, CSL-TR-89-397, 1989.
- [Sathe07] V. Sathe, J. Chueh, and M. Papaefthymiou, "Energy-efficient GHz-class charge-recovery logic," *JSSC*, vol. 42, no. 1, Jan. 2007, pp. 38–47.
- [Schellenberg98] F. Schellenberg, H. Zhang, and J. Morrow, "SEMATECH J111 Project: OPC validation," *Proc. Optical Microlithography XI*, SPIE vol. 3334, 1998, pp. 892–911.

- [Schellenberg03] F. Schellenberg, "A little light magic," *IEEE Spectrum*, vol. 40, no. 9, Sep. 2003, pp. 34–39.
- [Schmitt38] O. Schmitt, "A thermionic trigger," *J. Scientific Instruments*, vol. 15, Jan. 1938, pp. 24–26.
- [Schulmann98] W. Schulmann, F. Thimm, and H. Kaiser, "Rotating head for crystal pulling systems for carrying out the Czochralski process," US Patent 5766348, 1998.
- [Schultz90] K. Schultz, R. Francis, and K. Smith, "Ganged CMOS: trading standby power for speed," *JSSC*, vol. SC-25, no. 3, Jun. 1990, pp. 870–873.
- [Schutten03] R. Schutten, T. Fitzpatrick, "Design for verification—blueprint for productivity and product quality," Synopsys white paper, 2003.
- [Schutz98] J. Schutz and R. Wallace, "A 450 MHz IA32 P6 family microprocessor," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 1998, pp. 236–237.
- [Seeds67] R. Seeds, "Yield and cost analysis of bipolar LSI," *Intl. Electron Device Meeting*, Oct. 1967.
- [Seevinck87] E. Seevinck, F. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *JSSC*, vol. 22, no. 5, Oct. 1987, pp. 748–754.
- [Segura04] J. Segura and C. Hawkins, *CMOS Electronics: How it Works, How it Fails*, Hoboken, NJ: John Wiley & Sons (IEEE Press), 2004.
- [Shahidi02] G. Shahidi, "SOI technology for the GHz era," *IBM J. Research and Development*, vol. 46, no. 2/3, Mar./May 2002, pp. 121–131.
- [Sharifkhani09] M. Sharifkhani and M. Sachdev, "SRAM cell stability: a dynamic perspective," *JSSC*, vol. 44, no. 2, Feb. 2009, pp. 609–619.
- [She02] M. She et al., "JVD silicon nitride as tunnel dielectric in p-channel flash memory," *IEEE Electron Device Letters*, vol. 23, no. 2, Feb. 2002, pp. 91–93.
- [Shepard99] K. Shepard, V. Narayanan, and R. Rose, "Harmony: static noise analysis of deep submicron digital integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. 18, no. 8, Aug. 1999, pp. 1132–1150.
- [Sheu87] B. Sheu, D. Scharfetter, P. Ko, and M. Jeng, "BSIM: Berkeley short-channel IGFET model for MOS transistors," *JSSC*, vol. SC-22, no. 4, Aug. 1987, pp. 558–566.
- [Shichman68] H. Shichman and D. Hodges, "Modeling and simulation of insulated-gate field-effect transistor switching circuits," *JSSC*, vol. SC-3, no. 3, Sep. 1968, pp. 285–289.
- [Shigematsu97] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tanabe, and J. Yamada, "A 1-V high-speed MTCMOS circuit scheme for power-down application circuits," *JSSC*, vol. 32, no. 6, Jun. 1997, pp. 861–869.
- [Shin05] J. Shin, B. Petrick, M. Singh, and A. Leon, "Design and implementation of an embedded 512-KB level-2 cache subsystem," *JSSC*, vol. 40, no. 9, Sep. 2005, pp. 1815–1820.
- [Shockley52] W. Shockley, "A unipolar 'field-effect' transistor," *Proc. IRE*, vol. 40, 1952, pp. 1365–1376.
- [Shoji82] M. Shoji, "Electrical design of BELLMAC-32a microprocessor," *Proc. IEEE Intl. Conf. Circuits and Computers*, Sep. 1982, pp. 112–115.
- [Shoji86] M. Shoji, "Elimination of process-dependent clock skew in CMOS VLSI," *JSSC*, vol. SC-21, no. 5, Oct. 1986, pp. 875–880.
- [SIA97] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 1997.
- [SIA07] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 2007.
- [Silberman98] J. Silberman et al., "A 1.0-GHz single-issue 64-bit PowerPC integer microprocessor," *JSSC*, vol. 33, no. 11, Nov. 1998, pp. 1600–1608.
- [Singh08] P. Singh, J. Seo, D. Blaauw, and D. Sylvester, "Self-timed regenerators for high-speed and low-power on-chip global interconnect," *IEEE Trans. VLSI*, vol. 16, no. 6, Jun. 2008, pp. 673–677.
- [Sklansky60] J. Sklansky, "Conditional-sum addition logic," *IRE Trans. Electronic Computers*, vol. EC-9, Jun. 1960, pp. 226–231.
- [Sklar01] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed., Upper Saddle River, NJ: Prentice Hall, 2001.
- [Sleight01] J. Sleight et al., "A high performance 0.13 μm SOI CMOS technology with a 70 nm silicon film and with a second generation low-k Cu BEOL," *Proc. Intl. Electron Devices Meeting*, 2001, pp. 11.3.1–11.3.4.

- [Smith99] L. Smith, R. Anderson, D. Forehand, T. Pelc, and T. Roy, "Power distribution system design methodology and capacitor selection for modern CMOS technology," *IEEE Trans. Advanced Packaging*, vol. 22, no. 3, Aug. 1999, pp. 284–291.
- [Sodini84] C. Sodini, Ping-Keung Ko, and J. Moll, "The effect of high fields on MOS device and circuit performance," *IEEE Trans. Electron Devices*, vol. 31, no. 10, Oct. 1984, pp. 1386–1393.
- [Solomatnikov00] A. Solomatnikov, D. Somasekhar, K. Roy, and C. Koh, "Skewed CMOS: noise-immune high-performance low-power static circuit family," *Proc. IEEE Intl. Conf. Computer Design*, 2000, pp. 241–246.
- [Somasekhar96] D. Somasekhar and K. Roy, "Differential current switch logic: a low power DCVS logic family," *JSSC*, vol. 31, no. 7, Jul. 1996, pp. 981–991.
- [Somasekhar98] D. Somasekhar and K. Roy, "LVDCSL: a high fan-in, high-performance, low-voltage differential current switch logic family," *IEEE Trans. VLSI*, vol. 6, no. 4, Dec. 1998, pp. 573–577.
- [Somasekhar00] D. Somasekhar, S. Choi, K. Roy, Y. Ye, and V. De, "Dynamic noise analysis in precharge-evaluate circuits," *Proc. Design Automation Conf.*, 2000, pp. 243–246.
- [Song96] M. Song, G. Kang, S. Kim, and B. Kang, "Design methodology for high speed and low power digital circuits with energy economized pass-transistor logic (EEPL)," *Proc. 22nd European Solid-State Circuits Conf.*, 1996, pp. 120–123.
- [Song01] S. Song et al., "On the gate oxide scaling of high performance CMOS transistors," *Proc. Intl. Electron Devices Meeting*, 2001, pp. 3.2.1–3.2.4.
- [Sparsø01] J. Sparsø and S. Furber, eds., *Principles of Asynchronous Circuit Design: A Systems Perspective*, Boston: Kluwer Academic Publishers, 2001.
- [Srinivas92] H. Srinivas and K. Parhi, "A fast VLSI adder architecture," *JSSC*, vol. 27, no. 5, May 1992, pp. 761–767.
- [Srinivasan02] V. Srinivasan et al., "Optimizing pipelines for power and performance," *Proc. Intl. Symp. Microarchitecture*, 2002, pp. 333–344.
- [Stackhouse09] B. Stackhouse et al., "A 65 nm 2-billion transistor quad-core Itanium processor," *JSSC*, vol. 44, no. 1, Jan. 2009, pp. 18–31.
- [Stan98] M. Stan, A. Tenca, and M. Ercegovac, "Long and fast up/down counters," *IEEE Trans. Computers*, vol. 47, no. 7, Jul. 1998, pp. 722–735.
- [Stan99] M. Stan, "Optimal voltages and sizing for low power [CMOS VLSI]," *Proc. Intl. Conf. VLSI Design*, Jan. 1999, pp. 428–433.
- [Stelling98] P. Stelling, C. Martel, V. Oklobdzija, and R. Ravi, "Optimal circuits for parallel multipliers," *IEEE Trans. Computers*, vol. 47, no. 3, Mar. 1998, pp. 273–285.
- [Stinson03] J. Stinson and S. Rusu, "A 1.5 GHz third generation Itanium processor," *Proc. Design Automation Conf.*, 2003, pp. 706–709.
- [Stojanovic99] V. Stojanovic and V. Oklobdžija, "Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems," *JSSC*, vol. 34, no. 4, Apr. 1999, pp. 536–548.
- [Stolk98] P. Stolk, F. Widdershoven, and D. Klaassen, "Modeling statistical dopant fluctuations in MOS transistors," *IEEE Trans. Electron Devices*, vol. 45, no. 9, Sep. 1998, pp. 1960–1971.
- [Stolt08] B. Stolt et al., "Design and Implementation of the POWER6 microprocessor," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 21–28.
- [Strollo05] A. Strollo, D. De Caro, E. Napoli, and N. Petra, "A novel high-speed sense-amplifier-based flip-flop," *IEEE Trans. VLSI*, vol. 13, no. 11, Nov. 2005, pp. 1266–1274.
- [Stroud02] C. Stroud, *A Designer's Guide to Built-in Self-Test*, Boston: Kluwer Academic Publishers, 2002.
- [Su03] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full chip leakage-estimation considering power supply and temperature variations," *Proc. Intl. Symp. Low Power Electronics & Design*, Aug. 2003, pp. 78–83.
- [Su08] Y. Su, J. Holleman, and B. Otis, "A digital 1.6 pJ/bit chip identification circuit using process variations," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 69–77.

- [Sun87] J. Sun, Y. Taur, R. Dennard, and S. Klepner, "Submicrometer-channel CMOS for low-temperature operation," *IEEE Trans. Electron Devices*, vol. ED-34, no. 1, Jan. 1987, pp. 19–26.
- [Sutherland99] I. Sutherland, B. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*, San Francisco, CA: Morgan Kaufmann, 1999.
- [Suzuki73] Y. Suzuki, K. Odagawa and T. Abe, "Clocked CMOS calculator circuitry," *JSSC*, vol. SC-8, no. 6, Dec. 1973, pp. 462–469.
- [Suzuki93] M. Suzuki, N. Ohkubo, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, and Y. Nakagome, "A 1.5-ns 32-b CMOS ALU in double pass-transistor logic," *JSSC*, vol. 28, no. 11, Nov. 1993, pp. 1145–1151.
- [Svensson03] C. Svensson, "Forty years of feature-size predictions (1962–2002)," *Proc. Intl. Solid-State Circuits Conf.*, 2003, pp. 35–36.
- [Swanson72] R. Swanson and J. Meindl, "Ion-implanted complementary MOS transistors in low-voltage circuits," *JSSC*, vol. 7, no. 2, Apr. 1972, pp. 146–153.
- [Sweeney02] P. Sweeney, *Error Control Coding: From Theory to Practice*, New York: John Wiley & Sons, 2002.
- [Sylvester98] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron," *Proc. IEEE/ACM Intl. Conf. Computer-Aided Design*, 1998, pp. 203–211.
- [Takahashi98] M. Takahashi et al., "A 60-mW MPEG4 video codec using clustered voltage scaling with variable supply-voltage scheme," *JSSC*, vol. 33, no. 11, Nov. 1998, pp. 1772–1780.
- [Takeuchi94] K. Takeuchi and M. Fukuma, "Effects of the velocity saturated region on MOSFET characteristics," *IEEE Trans. Electron Devices*, vol. 41, no. 9, Sep. 1994, pp. 1623–1627.
- [Tam00] S. Tam, S. Rusu, U. Desai, R. Kim, J. Zhang, and I. Young, "Clock generation and distribution for the first IA-64 microprocessor," *JSSC*, vol. 35, no. 11, Nov. 2000, pp. 1545–1552.
- [Tam04] S. Tam, R. Limaye, and U. Desai, "Clock generation and distribution for the 130-nm Itanium 2 processor with 6-MB on-die L3 cache," *JSSC*, vol. 39, no. 4, Apr. 2004.
- [Tang97] X. Tang, V. De, and J. Meindl, "Intrinsic MOSFET parameter fluctuations due to random dopant placement," *IEEE Trans. VLSI*, vol. 5, no. 4, Dec 1997, pp. 369–376.
- [Tawfik09] S. Tawfik and V. Kursun, "Low power and high speed multi threshold voltage interface circuits," *IEEE Trans. VLSI*, vol. 17, no. 5, May 2009, pp. 638–645.
- [Tharakan92] G. Tharakan and S. Kang, "A new design of a fast barrel switch network," *JSSC*, vol. 27, no. 2, Feb. 1992, pp. 217–221.
- [Thompson02] S. Thompson et al., "A 90 nm logic technology featuring 50 nm strained silicon channel transistors, 7 layers of Cu interconnects, low k ILD, and 1 μm^2 SRAM cell," *Proc. Intl. Electron Devices Meeting*, 2002, pp. 61–64.
- [Thompson04] S. Thompson et al., "A logic nanotechnology featuring strained-silicon," *IEEE Electron Device Letters*, vol. 25, no. 4, Apr. 2004, pp. 191–193.
- [Thorp99] T. Thorp, G. Yee, and C. Sechen, "Design and synthesis of monotonic circuits," *Proc. IEEE Intl. Conf. Computer Design*, 1999, pp. 569–572.
- [Tierno08] J. Tierno, A. Rylyakov, and D. Friedman, "A wide power supply range, wide tuning range, all static CMOS all digital PLL in 65 nm SOI," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 42–51.
- [Tobias95] P. Tobias and D. Trindade, *Applied Reliability*, 2nd ed., New York: Van Nostrand Reinhold, 1995.
- [Toh88] K. Toh, P. Ko, and R. Meyer, "An engineering model for short-channel MOS devices," *JSSC*, vol. 23, no. 4, Aug. 1988, pp. 950–958.
- [Tokunaga08] C. Tokunaga, D. Blaauw, and T. Mudge, "True random number generator with a metastability-based quality control," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 78–85.
- [Topaloglu07] R. Topaloglu, "Standard cell and custom circuit optimization using dummy diffusions through STI width stress effect utilization," *Proc. Custom Integrated Circuits Conf.*, Sep. 2007, pp. 619–622.
- [Topol06] A. Topol et al., "Three-dimensional integrated circuits," *IBM J. Research and Dev.*, vol. 50, no. 4/5, Jul./Sep. 2006, pp. 491–506.

- [Trinh09] C. Trinh et al., "A 5.6MB/s 64Gb 4b/cell NAND flash memory in 43nm CMOS," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2009, pp. 246–247, 247a.
- [Troutman86] R. Troutman, *Latchup in CMOS Technology: The Problem and its Cure*, Boston: Kluwer Academic Publishers, 1986.
- [Tschanz01] J. Tschanz, S. Narendra, Zhanping Chen, S. Borkar, M. Sachdev, and Vivek De, "Comparative delay and energy of single edge-triggered and dual edge-triggered pulsed flip-flops for high-performance microprocessors," *Proc. Intl. Symp. Low Power Electronics & Design*, 2001, pp. 147–152.
- [Tschanz02] J. Tschanz et al., "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *JSSC*, vol. 37, no. 11, Nov. 2002, pp. 1396–1402.
- [Tschanz03] J. Tschanz, S. Narendra, Y. Ye, B. Bloechel, S. Borkar, and V. De, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *JSSC*, vol. 38, no. 11, Nov. 2003, pp. 1838–1845.
- [Tschanz03b] J. Tschanz, S. Narendra, R. Nair, and V. De, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *JSSC*, vol. 38, no. 5, May 2003, pp. 826–829.
- [Tschanz07] J. Tschanz et al., "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2007, pp. 292–604.
- [Tsividis99] Y. Tsividis, *Operation and Modeling of the MOS Transistor*, 2nd ed., Boston: McGraw-Hill, 1999.
- [Tyagi93] A. Tyagi, "A reduced-area scheme for carry-select adders," *JSSC*, vol. 42, no. 10, Oct. 1993, pp. 1163–1170.
- [Tyagi00] S. Tyagi et al., "A 130 nm generation logic technology featuring 70 nm transistors, dual V_t transistors and 6 layers of Cu interconnects," *Proc. Intl. Electron Devices Meeting*, 2000, pp. 567–570.
- [Tyagi05] S. Tyagi et al., "An advanced low power, high performance, strained channel 65nm technology," *Proc. Intl. Electron Devices Meeting*, Dec. 2005, pp. 245–247.
- [Uehara81] T. Uehara and W. van Cleemput, "Optimal layout of CMOS functional arrays," *IEEE Trans. Computers*, vol. C-30, no. 5, May 1981, pp. 305–312.
- [Unger86] S. Unger and C. Tan, "Clocking schemes for high-speed digital systems," *IEEE Trans. Computers*, vol. 35, no. 10, Oct. 1986, pp. 880–895.
- [Usami95] K. Usami and M. Horowitz, "Clustered voltage scaling for low-power design," *Proc. Intl. Symp. Low Power Electronics and Design*, 1995, pp. 3–8.
- [Vadasz66] L. Vadasz and A. Grove, "Temperature dependence of MOS transistor characteristics below saturation," *IEEE Trans. Electron Devices*, vol. ED-13, no. 13, 1966, pp. 863–866.
- [Vadasz69] L. Vadasz, A. Grove, T. Rowe, and G. Moore, "Silicon-gate technology," *IEEE Spectrum*, vol. 6, no. 10, Oct. 1969, pp. 28–35.
- [van Berkel99] C. van Berkel and C. Molnar, "Beware the three-way arbiter," *JSSC*, vol. 34, no. 6, Jun. 1999, pp. 840–848.
- [Vangal02] S. Vangal et al., "5-GHz 32-bit integer execution core in 130-nm dual- V_T CMOS," *JSSC*, vol. 37, no. 11, Nov. 2002, pp. 1421–1432.
- [Veendrick80] H. Veendrick, "The behavior of flip-flops used as synchronizers and prediction of their failure rate," *JSSC*, vol. SC-15, no. 2, Apr. 1980, pp. 169–176.
- [Veendrick84] H. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *JSSC*, vol. SC-19, no. 4, Aug. 1984, pp. 468–473.
- [Vittal99] A. Vittal et al., "Crosstalk in VLSI interconnections," *IEEE Trans. Computer-Aided Design*, vol. 18, no. 12, Dec. 1999, pp. 1817–1824.
- [Vittoz72] E. Vittoz, B. Gerber, and F. Leuenberger, "Silicon-gate CMOS frequency divider for electronic wrist watch," *JSSC*, vol. 7, no. 2, Apr. 1972, pp. 100–104.
- [Volk01] A. Volk, P. Stoll, and P. Metrovich, "Recollections of early chip development at Intel," *Intel Technology Journal*, Q1 2001, pp. 1–12.

- [Vollertsen99] R. Vollertsen, "Burn-in," *IEEE Integrated Reliability Workshop Final Report*, 1999, pp. 167–173.
- [von Arnim05] K. von Armin, et al., "Efficiency of body biasing in 90-nm CMOS for low-power digital circuits," *JSSC*, vol. 40, no. 7, July 2005, pp. 1549–1556.
- [von Neumann51] J. von Neumann, "Various techniques used in connection with random digits," notes by Forsythe, G., *National Bureau of Standards Applied Math Series*, 1951, vol. 12, pp. 36–38. Reprinted in *von Neumann's Collected Works*, vol. 5, Pergamon Press, 1963, pp. 768–770.
- [von Neumann66] J. von Neumann, *Theory of Self-Reproducing Automata*, Urbana, IL: Univ. Illinois Press, 1966.
- [Wadell01] B. Wadell, *Transmission Line Design Handbook*, Norwood, MA: Artech House, 1991.
- [Wakerly00] J. Wakerly, *Digital Design Principles and Practices*, 3rd ed., Upper Saddle River, NJ: Prentice Hall, 2000.
- [Wallace64] C. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electronic Computers*, Feb. 1964, pp. 14–17.
- [Wang86] L. Wang and E. McCluskey, "Complete feedback shift register design for built-in self test," *Proc. Design Automation Conf.*, Nov. 1986, pp. 56–59.
- [Wang89] J. Wang, C. Wu, and M. Tsai, "CMOS nonthreshold logic (NTL) and cascode nonthreshold logic (CNTL) for high-speed applications," *JSSC*, vol. 24, no. 3, Jun. 1989, pp. 779–786.
- [Wang93] Z. Wang, G. Jullien, W. Miller, and J. Wang, "New concepts for the design of carry-lookahead adders," *Proc. IEEE Intl. Symp. Circuits and Systems*, vol. 3, 1993, pp. 1837–1840.
- [Wang94] J. Wang, S. Fang, and W. Feng, "New efficient designs for XOR and XNOR functions on the transistor level," *JSSC*, vol. 29, no. 7, Jul. 1994, pp. 780–786.
- [Wang97] Z. Wang, G. Jullien, W. Miller, J. Wang, and S. Bizzan, "Fast adders using enhanced multiple-output domino logic," *JSSC*, vol. 32, no. 2, Feb. 1997, pp. 206–214.
- [Wang00] J. Wang and C. Huang, "High-speed and low-power CMOS priority encoders," *JSSC*, vol. 35, no. 10, Oct. 2000, pp. 1511–1514.
- [Wang01] J. Wang, C. Chang, and C. Yeh, "Analysis and design of high-speed and low-power CMOS PLAs," *JSSC*, vol. 36, no. 8, Aug. 2001, pp. 1250–1262.
- [Wang02] A. Wang, A. Chandrakasan, and S. Kosonocky, "Optimal supply and threshold scaling for subthreshold CMOS circuits," *Proc. Computer Society Symp. VLSI*, 2002, pp. 5–9.
- [Wang06] G. Wang et al., "A 0.127 μm^2 high performance 65 nm SOI based embedded DRAM for on-processor applications," *Proc. Intl. Electron Devices Meeting*, Dec. 2006, pp. 1–4.
- [Wang07] C. Wang, C. Lee, and W. Lin, "A 4-kb low-power SRAM design with negative word-line scheme," *IEEE Trans. Circuits & Systems*, vol. 54, no. 5, May 2007, pp. 1069–1076.
- [Wang08a] A. Wang and S. Naffziger, eds., *Adaptive Techniques for Dynamic Processor Optimization: Theory and Practice*, New York: Springer, 2008.
- [Wang08b] L. Wang, C. Stroud, and N. Touba, eds., *System-on-Chip Test Architectures: Nanometer Design for Testability*, Elsevier, 2008.
- [Wanlass63] F. Wanlass and C. Sah, "Nanowatt logic using field effect metal-oxide semiconductor triodes," *Proc. IEEE Intl. Solid-State Circuits Conf.*, 1963, pp. 32–33.
- [Warnock06] J. Warnock et al., "Circuit design techniques for a first-generation cell broadband engine processor," *JSSC*, vol. 41, no. 8, Aug. 2006, pp. 1692–1706.
- [Webb97] C. Webb et al., "A 400-MHz S/390 microprocessor," *JSSC*, vol. 32, no. 11, Nov. 1997, pp. 1665–1675.
- [Webb08] C. Webb, "45 nm design for manufacturing," *Intel Technology Journal*, vol. 12, no. 2, Jun. 2008, pp. 121–130.
- [Wei98] L. Wei, Z. Chen, M. Johnson, K. Roy, and V. De, "Design and optimization of low voltage high performance dual threshold CMOS circuits," *Proc. Design Automation Conf.*, 1998, pp. 489–494.
- [Wei00] G. Wei, J. Kim, D. Liu, S. Sidiropoulos, and M. Horowitz, "A variable-frequency parallel I/O interface with adaptive power-supply regulation," *JSSC*, vol. 35, no. 11, Nov. 2000, pp. 1600–1610.

- [Weinberger58] A. Weinberger and J. Smith, "A logic for high-speed addition," *System Design of Digital Computer at the National Bureau of Standards: Methods for High-Speed Addition and Multiplication*, National Bureau of Standards, circular 591, section 1, Feb. 1958, pp. 3–12.
- [Weinberger81] A. Weinberger, "4-2 carry-save adder module," *IBM Technical Disclosure Bulletin*, vol. 23, no. 8, Jan. 1981, pp. 3811–3814.
- [Weinlader00] D. Weinlader, Ron Ho, Chih-Kong Keng Yang, and M. Horowitz, "An eight channel 35 GSample/s CMOS timing analyzer," *Proc. Intl. Solid-State Circuits Conf.*, 2000, pp. 170–171.
- [Weiss02] D. Weiss, J. Wu, and V. Chin, "The on-chip 3-MB subarray-based third-level cache on an Itanium microprocessor," *JSSC*, vol. 37, no. 11, Nov. 2002, pp. 1523–1529.
- [Widmer83] A. Widmer and P. Franaszek, "A DC-balanced partitioned-block 8B/10B transmission code," *IBM J. Research and Dev.*, vol. 27, no. 5, Sep. 1983, pp. 440–451.
- [Wijeratne07] S. Wijeratne et al., "A 9-GHz 65-nm Intel Pentium 4 processor integer execution unit," *JSSC*, vol. 42, no. 1, Jan. 2007, pp. 26–37.
- [Williams83] T. Williams and K. Parker, "Design for Testability—A Survey," *Proc. IEEE*, vol. 71, no. 1, Jan. 1983, pp. 98–112.
- [Williams86] T. Williams, "Design for testability," *Proc. NATO Advanced Study Inst. Computer Design Aids for VLSI Circuits*, (P. Antognetti et al., eds.), NATO ASI Series, Martinus Nijhoff Publishers, 1986, pp. 359–416.
- [Williams91] T. Williams and M. Horowitz, "A zero-overhead self-timed 160-ns 54-b CMOS divider," *JSSC*, vol. 26, no. 11, Nov. 1991, pp. 1651–1661.
- [Wilton96] S. Wilton and N. Jouppi, "CACTI: an enhanced cache access and cycle time model," *JSSC*, vol. 31, no. 5, May 1996, pp. 677–688.
- [Wing82] O. Wing, "Automated gate matrix layout," *Proc. IEEE Intl. Symp. Circuits and Systems*, vol. 2, 1982, pp. 681–685.
- [Wolf00] S. Wolf and R. Tauber, *Silicon Processing for the VLSI Era*, 2nd ed., Sunset Beach, CA: Lattice Press, 2000.
- [Wong03] H. Wong, J. Appenzeller, V. Derycke, R. Martel, S. Wind, and P. Avouris, "Carbon nanotube field effect transistors—fabrication, device physics, and circuit implications," *Proc. Intl. Solid-State Circuits Conf.*, 2003, pp. 370–500.
- [Wong06] K. Wong, T. Rahal-Arabi, M. Ma, and G. Taylor, "Enhancing microprocessor immunity to power supply noise with clock-data compensation," *JSSC*, vol. 41, no. 4, Apr. 2006, pp. 749–758.
- [Wu91] C. Wu and K. Cheng, "Latched CMOS differential logic (LCDL) for complex high-speed VLSI," *JSSC*, vol. 26, no. 9, Sep. 1991, pp. 1324–1328.
- [Xu08] J. Xu et al., "A band-limited active damping circuit with 13 dB power supply resonance reduction," *JSSC*, vol. 43, no. 1, Jan. 2008, pp. 61–68.
- [Yabuuchi07] M. Yabuuchi et al., "A 45nm low-standby-power embedded SRAM with improved immunity against process and temperature variations," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2007, pp. 326–606.
- [Yamada95] H. Yamada, T. Hotta, T. Nishiyama, F. Murabayashi, T. Yamauchi, and H. Sawamoto, "A 13.3 ns double-precision floating-point ALU and multiplier," *Proc. Intl. Conf. Computer Design*, 1995, pp. 466–470.
- [Yamaoka04] M. Yamaoka, K. Osada, and K. Ishibashi, "0.4-V logic-library-friendly SRAM array using rectangular-diffusion cell and delta-boosted-array voltage scheme," *JSSC*, vol. 39, no. 6, Jun. 2004, pp. 934–940.
- [Yamaoka04b] M. Yamaoka et al., "A 300MHz 25 μ A/Mb leakage on-chip SRAM module featuring process-variation immunity and low-leakage-active mode for mobile-phone application processor," *Proc. Intl. Solid-State Circuits Conf.*, Feb. 2004, pp. 494–542.
- [Yamaoka06] M. Yamaoka et al., "90-nm process-variation adaptive embedded SRAM modules with power-line-floating write technique," *JSSC*, vol. 41, no. 3, Mar. 2006, pp. 705–711.

- [Yang96] C. K. Yang and M. Horowitz, "A 0.8- μ m CMOS 2.5 Gb/s oversampling receiver and transmitter for serial links," *JSSC*, vol. 31, no. 12, Dec. 1996, pp. 2015–2023.
- [Yang98] S. Yang et al., "A high performance 180 nm generation logic technology," *Tech. Digest Intl. Electron Device Meeting*, Dec. 1998, pp. 197–200.
- [Yano90] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu, "A 3.8-ns 16×16 -b multiplier using complementary pass-transistor logic," *JSSC*, vol. 25, no. 2, Apr. 1990, pp. 388–395.
- [Yano96] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, "Top-down pass-transistor logic design," *JSSC*, vol. 31, no. 6, Jun. 1996, pp. 792–803.
- [Ye98] Y. Ye, S. Borkar, and V. De, "A new technique for standby leakage reduction in high-performance circuits," *Symp. VLSI Circuits Digest Tech. Papers*, 1998, pp. 40–41.
- [Ye00] Y. Ye, J. Tschanz, S. Narendra, S. Borkar, M. Stan, and V. De, "Comparative delay, noise and energy of high-performance domino adders with stack node preconditioning (SNP)," *Symp. VLSI Circuits Digest Tech. Papers*, 2000, pp. 188–191.
- [Yee00] G. Yee and C. Sechen, "Clock-delayed domino for dynamic circuit design," *IEEE Trans. VLSI*, vol. 8, no. 4, Aug. 2000, pp. 425–430.
- [Yoon02] J. Yoon et al., "CMOS-compatible surface-micromachined suspended-spiral inductors for multi-GHz silicon RF ICs," *IEEE Electron Device Letters*, vol. 23, no. 10, Oct. 2002, pp. 591–593.
- [Yoshimoto83] M. Yoshimoto et al., "A divided word-line structure in the static RAM and its application to a 64K full CMOS RAM," *JSSC*, vol. SC-18, no. 5, Oct. 1983, pp. 479–485.
- [Young00] K. Young et al., "A 0.13 μ m CMOS technology with 193 nm lithography and Cu/Low-k for high performance applications," *Proc. Intl. Electron Devices Meeting*, 2000, pp. 563–566.
- [Young10] I. Young et al., "Optical I/O technology for tera-scale computing," *JSSC*, vol. 45, no. 1, Jan. 2010, pp. 235–248.
- [Yuan82] C. Yuan and T. Trick, "A simple formula for the estimation of the capacitance of two-dimensional interconnects in VLSI circuits," *IEEE Electron Device Letters*, vol. EDL-3, Dec. 1982, pp. 391–393.
- [Yuan89] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *JSSC*, vol. 24, no. 1, Feb. 1989, pp. 62–70.
- [Zerbe01] J. Zerbe et al., "A 2 Gb/s/pin 4-PAM parallel bus interface with transmit crosstalk cancellation, equalization, and integrating receivers," *Proc. Intl. Solid-State Circuits Conf.*, 2001, pp. 66–67, 432.
- [Zhai05a] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. VLSI*, vol. 13, no. 11, Nov. 2005, pp. 1239–1252.
- [Zhai05b] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Analysis and mitigation of variability in subthreshold design," *Proc. Intl. Symp. Low Power Electronics & Design*, Aug. 2005, pp. 20–25.
- [Zhai08] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "A variation-tolerant sub-200 mV 6-T subthreshold SRAM," *JSSC*, vol. 43, no. 10, Oct. 2008, pp. 2338–2348.
- [Zhang05] K. Zhang et al., "SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction," *JSSC*, vol. 40, no. 4, Apr. 2005, pp. 895–901.
- [Zhang06] K. Zhang et al., "A 3-GHz 70-mb SRAM in 65-nm CMOS technology with integrated column-based dynamic power supply," *JSSC*, vol. 41, no. 1, Jan. 2006, pp. 146–151.
- [Zhao07] P. Zhao, J. McNeely, P. Golconda, M. Bayoumi, R. Barcnas, and Weidong Kuang, "Low-power clock branch sharing double-edge triggered flip-flop," *IEEE Trans. VLSI*, vol. 15, no. 3, Mar. 2007, pp. 338–345.
- [Zhao09] P. Zhao et al., "Low-power clocked-pseudo-NMOS flip-flop for level conversion in dual supply systems," *IEEE Trans. VLSI*, vol. 17, no. 9, Sep. 2009, pp. 1196–1202.
- [Zhou99] X. Zhou, K. Lim, and D. Lim, "A simple and unambiguous definition of threshold voltage and its implications in deep-submicron MOS device modeling," *IEEE Trans. Electron Devices*, vol. 46, no. 4, Apr. 1999, pp. 807–809.
- [Ziegler96] J. Ziegler, "Terrestrial cosmic rays," *IBM J. Research and Development*, vol. 40, no. 1, Jan. 1996, pp. 19–39.

- [Ziegler02] J. Ziegler, *Ion-Implantation—Science and Technology*, 2002 ed., IIT Press, 2002.
- [Zimmermann96] R. Zimmermann, “Non-heuristic optimization and synthesis of parallel-prefix adders,” *Proc. Intl. Workshop on Logic and Architecture Synthesis*, Dec. 1996, pp. 123–132.
- [Zimmermann97] R. Zimmermann and W. Fichtner, “Low-power logic styles: CMOS versus pass-transistor logic,” *JSSC*, vol. 32, no. 7, Jul. 1997, pp. 1079–1090.
- [Zlatanovici09] R. Zlatanovici, S. Kao, and B. Nikolic, “Energy-delay optimization of 64-bit carry-lookahead adders with a 240 ps 90 nm CMOS design example,” *JSSC*, vol. 44, no. 2, Feb. 2009, pp. 569–583.
- [Zuras86] D. Zuras and W. McAllister, “Balanced delay trees and combinatorial division in VLSI,” *JSSC*, vol. SC-21, no. 5, Oct. 1986, pp. 814–819.

MOSIS SUBM design rules (3 metal, 1 poly with stacked vias & alternate contact rules)			
Layer	Rule	Description	Rule (λ)
N-well	1.1	Width	12
	1.2	Spacing to well at different potential	18
	1.3	Spacing to well at same potential	6
Active (diffusion)	2.1	Width	3
	2.2	Spacing to active	3
	2.3	Source/drain surround by well	6
	2.4	Substrate/well contact surround by well	3
	2.5	Spacing to active of opposite type	4
Poly	3.1	Width	2
	3.2	Spacing to poly over field oxide	3
	3.2a	Spacing to poly over active	3
	3.3	Gate extension beyond active	2
	3.4	Active extension beyond poly	3
	3.5	Spacing of poly to active	1
Select (n or p)	4.1	Spacing from substrate/well contact to gate	3
	4.2	Overlap of active	2
	4.3	Overlap of substrate/well contact	1
	4.4	Spacing to select	2
Contact (to poly or active)	5.1, 6.1	Width (exact)	2×2
	5.2b, 6.2b	Overlap by poly or active	1
	5.3, 6.3	Spacing to contact	3
	5.4, 6.4	Spacing to gate	2
	5.5b	Spacing of poly contact to other poly	5
	5.7b, 6.7b	Spacing to active/poly for multiple poly/active contacts	3
	6.8b	Spacing of active contact to poly contact	4
Metal1, Metal2	7.1, 9.1	Width	3
	7.2, 9.2	Spacing to same layer of metal	3
	7.3, 8.3, 9.3	Overlap of contact or via	1
	7.4, 9.4	Spacing to metal for lines wider than 10λ	6
Via1, Via2	8.1, 14.1	Width (exact)	2×2
	8.2, 14.2	Spacing to via on same layer	3
Metal3	15.1	Width	5
	15.2	Spacing to metal3	3
	15.3	Overlap of via2	2
	15.4	Spacing to metal for lines wider than 10λ	6
Overglass Cut	10.1	Width of bond pad opening	$60\mu\text{m}$
	10.2	Width of probe pad opening	$20\mu\text{m}$
	10.3	Metal3 overlap of overglass cut	$6\mu\text{m}$
	10.4	Spacing of pad metal to unrelated metal	$30\mu\text{m}$
	10.5	Spacing of pad metal to active or poly	$15\mu\text{m}$



Thank you for purchasing a new copy of *CMOS VLSI Design A Circuits and Systems Perspective, Fourth Edition* by Neil H.E. Weste, David Money Harris. The information below provides instruction on how to access the Companion site.

To access the Companion Website:

1. Go to <http://www.pearsonhighered.com/weste/>
2. From here you can register as a First-Time User or Returning User.
3. Your student access code will be sent to you by CourseSmart. On the registration page, enter your student access code. Do not type the dashes. You can use lower or uppercase letters.
4. Follow the on-screen instructions. If you need help during the online registration process, simply click on Need Help?
5. Once your personal Login Name and Password are confirmed, you can begin viewing the Companion Website.

To login to the website for the first time after you've registered:

Follow step 1 to return to the Companion Website. Then, follow the prompts for "Returning Users" to enter your Login Name and Password.

Note to Instructors: For access to the Instructor Resource Center, contact your Pearson Representative.

IMPORTANT: The access code on this page can only be used once to establish a subscription to the Companion Website for *CMOS VLSI Design A Circuits and Systems Perspective, Fourth Edition*. If this access code has already been redeemed, it will no longer be valid. If this is the case, you can purchase a subscription by going to the <http://www.pearsonhighered.com/weste/> website and selecting "Get Access."