

In Praise of Memory Systems: Cache, DRAM, Disk

Memory Systems: Cache, DRAM, Disk is the first book that takes on the whole hierarchy in a way that is consistent, covers the complete memory hierarchy, and treats each aspect in significant detail. This book will serve as a definitive reference manual for the expert designer, yet it is so complete that it can be read by a relative novice to the computer design space. While memory technologies improve in terms of density and performance, and new memory device technologies provide additional properties as design options, the principles and methodology presented in this amazingly complete treatise will remain useful for decades. I only wish that a book like this had been available when I started out more than three decades ago. It truly is a landmark publication. Kudos to the authors.

—Al Davis, University of Utah

Memory Systems: Cache, DRAM, Disk fills a huge void in the literature about modern computer architecture. The book starts by providing a high level overview and building a solid knowledge basis and then provides the details for a deep understanding of essentially all aspects of modern computer memory systems including architectural considerations that are put in perspective with cost, performance and power considerations. In addition, the historical background and politics leading to one or the other implementation are revealed. Overall, Jacob, Ng, and Wang have created one of the truly great technology books that turns reading about bits and bytes into an exciting journey towards understanding technology.

—Michael Schuette, Ph.D., VP of Technology Development at OCZ Technology

This book is a critical resource for anyone wanting to know how DRAM, cache, and hard drives really work. It describes the implementation issues, timing constraints, and trade-offs involved in past, present, and future designs. The text is exceedingly well-written, beginning with high-level analysis and proceeding to incredible detail only for those who need it. It includes many graphs that give the reader both explanation and intuition. This will be an invaluable resource for graduate students wanting to study these areas, implementers, designers, and professors.

—Diana Franklin, California Polytechnic University, San Luis Obispo

Memory Systems: Cache, DRAM, Disk fills an important gap in exploring modern disk technology with accuracy, lucidity, and authority. The details provided would only be known to a researcher who has also contributed in the development phase. I recommend this comprehensive book to engineers, graduate students, and researchers in the storage area, since details provided in computer architecture textbooks are woefully inadequate.

—Alexander Thomasian, IEEE Fellow, New Jersey Institute of Technology and Thomasian and Associates

Memory Systems: Cache, DRAM, Disk offers a valuable state of the art information in memory systems that can only be gained through years of working in advanced industry and research. It is about time that we have such a good reference in an important field for researchers, educators and engineers.

—Nagi Mekhiel, Department of Electrical and Computer Engineering, Ryerson University, Toronto

This is the only book covering the important DRAM and disk technologies in detail. Clear, comprehensive, and authoritative, I have been waiting for such a book for long time.

—Yiming Hu, University of Cincinnati

Memory is often perceived as the performance bottleneck in computing architectures. Memory Systems: Cache, DRAM, Disk, sheds light on the mystical area of memory system design with a no-nonsense approach to what matters and how it affects performance. From historical discussions to modern case study examples this book is certain to become as ubiquitous and used as the other Morgan Kaufmann classic textbooks in computer engineering including Hennessy and Patterson's Computer Architecture: A Quantitative Approach.

—R. Jacob Baker, Micron Technology, Inc. and Boise State University.

Memory Systems: Cache, DRAM, Disk is a remarkable book that fills a very large void. The book is remarkable in both its scope and depth. It ranges from high performance cache memories to disk systems. It spans circuit design to system architecture in a clear, cohesive manner. It is the memory architecture that defines modern computer systems, after all. Yet, memory systems are often considered as an appendage and are covered in a piecemeal fashion. This book recognizes that memory systems are the heart and soul of modern computer systems and takes a 'holistic' approach to describing and analyzing memory systems.

The classic book on memory systems was written by Dick Matick of IBM over thirty years ago. So not only does this book fill a void, it is a long-standing void. It carries on the tradition of Dick Matick's book extremely well, and it will doubtless be the definitive reference for students and designers of memory systems for many years to come. Furthermore, it would be easy to build a top-notch memory systems course around this book. The authors clearly and succinctly describe the important issues in an easy-to-read manner. And the figures and graphs are really great—one of the best parts of the book.

When I work at home, I make coffee in a little stove-top espresso maker I got in Spain. It makes good coffee very efficiently, but if you put it on the stove and forget it's there, bad things happen—smoke, melted gasket—'burned coffee meltdown.' This only happens when I'm totally engrossed in a paper or article. Today, for the first time, it happened twice in a row—while I was reading the final version of this book.

—Jim Smith, University of Wisconsin—Madison

Memory Systems

Cache, DRAM, Disk



Memory Systems

Cache, DRAM, Disk

Bruce Jacob

University of Maryland at College Park

Spencer W. Ng

Hitachi Global Storage Technologies

David T. Wang

MetaRAM

With Contributions By

Samuel Rodriguez

Advanced Micro Devices



ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier



MORGAN KAUFMANN PUBLISHERS

<i>Publisher</i>	Denise E.M. Penrose
<i>Acquisitions Editor</i>	Chuck Glaser
<i>Publishing Services Manager</i>	George Morrison
<i>Senior Production Editor</i>	Paul Gottehrer
<i>Developmental Editor</i>	Nate McFadden
<i>Assistant Editor</i>	Kimberlee Honjo
<i>Cover Design</i>	Joanne Blank
<i>Text Design</i>	Dennis Schaefer
<i>Composition</i>	diacriTech
<i>Interior printer</i>	Maple-Vail Book Manufacturing Group
<i>Cover printer</i>	Phoenix Color

Morgan Kaufmann Publishers is an imprint of Elsevier.
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA

This book is printed on acid-free paper.

© 2008 by Elsevier Inc. All rights reserved.

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, scanning, or otherwise—without prior written permission of the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, E-mail: permissions@elsevier.com. You may also complete your request online via the Elsevier homepage (<http://elsevier.com>), by selecting "Support & Contact" then "Copyright and Permission" and then "Obtaining Permissions."

Library of Congress Cataloging-in-Publication Data

Application submitted

ISBN: 978-0-12-379751-3

For information on all Morgan Kaufmann publications,
visit our Web site at www.mkp.com or www.books.elsevier.com

Printed in the United States of America

08 09 10 11 12 5 4 3 2 1

Working together to grow
libraries in developing countries

www.elsevier.com | www.bookaid.org | www.sabre.org

ELSEVIER

BOOK AID
International

Sabre Foundation

Dedication

*Jacob To my parents, Bruce and Ann Jacob, my wife,
Dorinda, and my children, Garrett, Carolyn,
and Nate*

*Ng Dedicated to the memory of my parents
Ching-Sum and Yuk-Ching Ng*

*Wang Dedicated to my parents Tu-Sheng Wang and
Hsin-Hsin Wang*

You can tell whether a person plays or not by the way he carries the instrument, whether it means something to him or not.

Then the way they talk and act. If they act too hip, you know they can't play [jack].

—Miles Davis

[...] in connection with musical continuity, Cowell remarked at the New School before a concert of works by Christian Wolff, Earle Brown, Morton Feldman, and myself, that here were four composers who were getting rid of glue. That is: Where people had felt the necessity to stick sounds together to make a continuity, we four felt the opposite necessity to get rid of the glue so that sounds would be themselves.

Christian Wolff was the first to do this. He wrote some pieces vertically on the page but recommended their being played horizontally left to right, as is conventional. Later he discovered other geometrical means for freeing his music of intentional continuity. Morton Feldman divided pitches into three areas, high, middle, and low, and established a time unit. Writing on graph paper, he simply inscribed numbers of tones to be played at any time within specified periods of time.

There are people who say, "If music's that easy to write, I could do it." Of course they could, but they don't. I find Feldman's own statement more affirmative. We were driving back from some place in New England where a concert had been given. He is a large man and falls asleep easily. Out of a sound sleep, he awoke to say, "Now that things are so simple, there's so much to do." And then he went back to sleep.

—John Cage, *Silence*

Contents

Preface	"It's the Memory, Stupid!"	xxxi
Overview	On Memory Systems and Their Design	1
	Ov.1 Memory Systems	2
	<i>Ov.1.1 Locality of Reference Breeds the Memory Hierarchy</i>	2
	<i>Ov.1.2 Important Figures of Merit</i>	7
	<i>Ov.1.3 The Goal of a Memory Hierarchy</i>	10
	Ov.2 Four Anecdotes on Modular Design	14
	<i>Ov.2.1 Anecdote I: Systemic Behaviors Exist</i>	15
	<i>Ov.2.2 Anecdote II: The DLL in DDR SDRAM</i>	17
	<i>Ov.2.3 Anecdote III: A Catch-22 in the Search for Bandwidth</i>	18
	<i>Ov.2.4 Anecdote IV: Proposals to Exploit Variability in Cell Leakage</i>	19
	<i>Ov.2.5 Perspective</i>	19
	Ov.3 Cross-Cutting Issues	20
	<i>Ov.3.1 Cost/Performance Analysis</i>	20
	<i>Ov.3.2 Power and Energy</i>	26
	<i>Ov.3.3 Reliability</i>	32
	<i>Ov.3.4 Virtual Memory</i>	34
	Ov.4 An Example Holistic Analysis	41
	<i>Ov.4.1 Fully-Buffered DIMM vs. the Disk Cache</i>	41
	<i>Ov.4.2 Fully Buffered DIMM: Basics</i>	43
	<i>Ov.4.3 Disk Caches: Basics</i>	46
	<i>Ov.4.4 Experimental Results</i>	47
	<i>Ov.4.5 Conclusions</i>	52
	Ov.5 What to Expect	54

Part I	Cache	55
Chapter 1	An Overview of Cache Principles	57
1.1	Caches, ‘Caches,’ and “Caches”	59
1.2	Locality Principles	62
1.2.1	<i>Temporal Locality</i>	<i>63</i>
1.2.2	<i>Spatial Locality</i>	<i>63</i>
1.2.3	<i>Algorithmic Locality</i>	<i>64</i>
1.2.4	<i>Geographical Locality? Demographical Locality?</i>	<i>65</i>
1.3	What to Cache, Where to Put It, and How to Maintain It	66
1.3.1	<i>Logical Organization Basics: Blocks, Tags, Sets</i>	<i>67</i>
1.3.2	<i>Content Management: To Cache or Not to Cache</i>	<i>68</i>
1.3.3	<i>Consistency Management: Its Responsibilities</i>	<i>69</i>
1.3.4	<i>Inclusion and Exclusion</i>	<i>70</i>
1.4	Insights and Optimizations	73
1.4.1	<i>Perspective</i>	<i>73</i>
1.4.2	<i>Important Issues, Future Directions</i>	<i>77</i>
Chapter 2	Logical Organization	79
2.1	Logical Organization: A Taxonomy	79
2.2	Transparently Addressed Caches	82
2.2.1	<i>Implicit Management: Transparent Caches</i>	<i>86</i>
2.2.2	<i>Explicit Management: Software-Managed Caches</i>	<i>86</i>
2.3	Non-Transparently Addressed Caches	90
2.3.1	<i>Explicit Management: Scratch-Pad Memories</i>	<i>91</i>
2.3.2	<i>Implicit Management: Self-Managed Scratch-Pads</i>	<i>92</i>
2.4	Virtual Addressing and Protection	92
2.4.1	<i>Virtual Caches</i>	<i>93</i>
2.4.2	<i>ASIDs and Protection Bits</i>	<i>96</i>
2.4.3	<i>Inherent Problems</i>	<i>96</i>
2.5	Distributed and Partitioned Caches	97
2.5.1	<i>UMA and NUMA</i>	<i>98</i>
2.5.2	<i>COMA</i>	<i>99</i>
2.5.3	<i>NUCA and NuRAPID</i>	<i>99</i>
2.5.4	<i>Web Caches</i>	<i>100</i>
2.5.5	<i>Buffer Caches</i>	<i>102</i>

2.6	Case Studies	103
2.6.1	<i>A Horizontal-Exclusive Organization: Victim Caches, Assist Caches</i>	103
2.6.2	<i>A Software Implementation: BSD's Buffer Cache</i>	104
2.6.3	<i>Another Dynamic Cache Block: Trace Caches</i>	106
Chapter 3	Management of Cache Contents	117
3.1	Case Studies: On-Line Heuristics	120
3.1.1	<i>On-Line Partitioning Heuristics</i>	120
3.1.2	<i>On-Line Prefetching Heuristics</i>	129
3.1.3	<i>On-Line Locality Optimizations</i>	141
3.2	Case Studies: Off-Line Heuristics	151
3.2.1	<i>Off-Line Partitioning Heuristics</i>	151
3.2.2	<i>Off-Line Prefetching Heuristics</i>	155
3.2.3	<i>Off-Line Locality Optimizations</i>	161
3.3	Case Studies: Combined Approaches	169
3.3.1	<i>Combined Approaches to Partitioning</i>	170
3.3.2	<i>Combined Approaches to Prefetching</i>	174
3.3.3	<i>Combined Approaches to Optimizing Locality</i>	180
3.4	Discussions	202
3.4.1	<i>Proposed Scheme vs. Baseline</i>	202
3.4.2	<i>Prefetching vs. Locality Optimizations</i>	203
3.4.3	<i>Application-Directed Management vs. Transparent Management</i>	203
3.4.4	<i>Real Time vs. Average Case</i>	204
3.4.5	<i>Naming vs. Cache Conflicts</i>	205
3.4.6	<i>Dynamic vs. Static Management</i>	208
3.5	Building a Content-Management Solution	212
3.5.1	<i>Degree of Dynamism</i>	212
3.5.2	<i>Degree of Prediction</i>	213
3.5.3	<i>Method of Classification</i>	213
3.5.4	<i>Method for Ensuring Availability</i>	214
Chapter 4	Management of Cache Consistency	217
4.1	Consistency with Backing Store	218
4.1.1	<i>Write-Through</i>	218
4.1.2	<i>Delayed Write, Driven By the Cache</i>	219
4.1.3	<i>Delayed Write, Driven by Backing Store</i>	220

4.2 Consistency with Self.....	220
4.2.1 <i>Virtual Cache Management</i>	220
4.2.2 <i>ASID Management.....</i>	225
4.3 Consistency with Other Clients.....	226
4.3.1 <i>Motivation, Explanation, Intuition.....</i>	226
4.3.2 <i>Coherence vs. Consistency</i>	231
4.3.3 <i>Memory-Consistency Models</i>	233
4.3.4 <i>Hardware Cache-Coherence Mechanisms.....</i>	240
4.3.5 <i>Software Cache-Coherence Mechanisms</i>	254
Chapter 5 Implementation Issues.....	257
5.1 Overview	257
5.2 SRAM Implementation.....	258
5.2.1 <i>Basic 1-Bit Memory Cell</i>	259
5.2.2 <i>Address Decoding.....</i>	262
5.2.3 <i>Physical Decoder Implementation.....</i>	268
5.2.4 <i>Peripheral Bitline Circuits.....</i>	278
5.2.5 <i>Sense Amplifiers.....</i>	281
5.2.6 <i>Write Amplifier.....</i>	283
5.2.7 <i>SRAM Partitioning.....</i>	285
5.2.8 <i>SRAM Control and Timing.....</i>	286
5.2.9 <i>SRAM Interface</i>	292
5.3 Advanced SRAM Topics.....	293
5.3.1 <i>Low-Leakage Operation</i>	293
5.4 Cache Implementation.....	297
5.4.1 <i>Simple Caches</i>	297
5.4.2 <i>Processor Interfacing.....</i>	298
5.4.3 <i>Multiporting.....</i>	298
Chapter 6 Cache Case Studies	301
6.1 Logical Organization	301
6.1.1 <i>Motorola MPC7450.....</i>	301
6.1.2 <i>AMD Opteron.....</i>	301
6.1.3 <i>Intel Itanium-2</i>	303

6.2 Pipeline Interface	304
6.2.1 <i>Motorola MPC7450.....</i>	304
6.2.2 <i>AMD Opteron.....</i>	304
6.2.3 <i>Intel Itanium-2</i>	304
6.3 Case Studies of Detailed Itanium-2 Circuits.....	305
6.3.1 <i>L1 Cache RAM Cell Array.....</i>	305
6.3.2 <i>L2 Array Bitline Structure.....</i>	305
6.3.3 <i>L3 Subarray Implementation.....</i>	308
6.3.4 <i>Itanium-2 TLB and CAM Implementation</i>	308

Part II DRAM.....313

Chapter 7 Overview of DRAMs.....	315
7.1 DRAM Basics: Internals, Operation	316
7.2 Evolution of the DRAM Architecture	322
7.2.1 <i>Structural Modifications Targeting Throughput</i>	322
7.2.2 <i>Interface Modifications Targeting Throughput.....</i>	328
7.2.3 <i>Structural Modifications Targeting Latency.....</i>	330
7.2.4 <i>Rough Comparison of Recent DRAMs</i>	331
7.3 Modern-Day DRAM Standards.....	332
7.3.1 <i>Salient Features of JEDEC's SDRAM Technology.....</i>	332
7.3.2 <i>Other Technologies, Rambus in Particular.....</i>	335
7.3.3 <i>Comparison of Technologies in Rambus and JEDEC DRAM.....</i>	341
7.3.4 <i>Alternative Technologies.....</i>	343
7.4 Fully Buffered DIMM: A Compromise of Sorts	348
7.5 Issues in DRAM Systems, Briefly	350
7.5.1 <i>Architecture and Scaling</i>	350
7.5.2 <i>Topology and Timing.....</i>	350
7.5.3 <i>Pin and Protocol Efficiency</i>	351
7.5.4 <i>Power and Heat Dissipation</i>	351
7.5.5 <i>Future Directions</i>	351

Chapter 8	DRAM Device Organization: Basic Circuits and Architecture	353
8.1	DRAM Device Organization	353
8.2	DRAM Storage Cells.....	355
8.2.1	<i>Cell Capacitance, Leakage, and Refresh</i>	356
8.2.2	<i>Conflicting Requirements Drive Cell Structure.....</i>	356
8.2.3	<i>Trench Capacitor Structure.....</i>	357
8.2.4	<i>Stacked Capacitor Structure.....</i>	357
8.3	RAM Array Structures.....	358
8.3.1	<i>Open Bitline Array Structure.....</i>	359
8.3.2	<i>Folded Bitline Array Structure.....</i>	360
8.4	Differential Sense Amplifier.....	360
8.4.1	<i>Functionality of Sense Amplifiers in DRAM Devices.....</i>	361
8.4.2	<i>Circuit Diagram of a Basic Sense Amplifier</i>	362
8.4.3	<i>Basic Sense Amplifier Operation.....</i>	362
8.4.4	<i>Voltage Waveform of Basic Sense Amplifier Operation.....</i>	363
8.4.5	<i>Writing into DRAM Array.....</i>	365
8.5	Decoders and Redundancy	366
8.5.1	<i>Row Decoder Replacement Example</i>	368
8.6	DRAM Device Control Logic.....	368
8.6.1	<i>Synchronous vs. Non-Synchronous</i>	369
8.6.2	<i>Mode Register-Based Programmability.....</i>	370
8.7	DRAM Device Configuration.....	370
8.7.1	<i>Device Configuration Trade-offs.....</i>	371
8.8	Data I/O.....	372
8.8.1	<i>Burst Lengths and Burst Ordering</i>	372
8.8.2	<i>N-Bit Prefetch.....</i>	372
8.9	DRAM Device Packaging	373
8.10	DRAM Process Technology and Process Scaling Considerations.....	374
8.10.1	<i>Cost Considerations.....</i>	375
8.10.2	<i>DRAM- vs. Logic-Optimized Process Technology.....</i>	375
Chapter 9	DRAM System Signaling and Timing	377
9.1	Signaling System.....	377

9.2	Transmission Lines on PCBs	379
9.2.1	<i>Brief Tutorial on the Telegrapher's Equations.....</i>	380
9.2.2	<i>RC and LC Transmission Line Models</i>	382
9.2.3	<i>LC Transmission Line Model for PCB Traces.....</i>	383
9.2.4	<i>Signal Velocity on the LC Transmission Line</i>	383
9.2.5	<i>Skin Effect of Conductors</i>	384
9.2.6	<i>Dielectric Loss</i>	384
9.2.7	<i>Electromagnetic Interference and Crosstalk</i>	386
9.2.8	<i>Near-End and Far-End Crosstalk</i>	387
9.2.9	<i>Transmission Line Discontinuities.....</i>	388
9.2.10	<i>Multi-Drop Bus</i>	390
9.2.11	<i>Socket Interfaces.....</i>	391
9.2.12	<i>Skew</i>	392
9.2.13	<i>Jitter.....</i>	392
9.2.14	<i>Inter-Symbol Interference (ISI)</i>	393
9.3	Termination	393
9.3.1	<i>Series Stub (Serial) Termination.....</i>	394
9.3.2	<i>On-Die (Parallel) Termination</i>	394
9.4	Signaling	395
9.4.1	<i>Eye Diagrams</i>	396
9.4.2	<i>Low-Voltage TTL (Transistor-Transistor Logic).....</i>	396
9.4.3	<i>Voltage References.....</i>	398
9.4.4	<i>Series Stub Termination Logic</i>	398
9.4.5	<i>RSL and DRSL</i>	399
9.5	Timing Synchronization.....	399
9.5.1	<i>Clock Forwarding.....</i>	400
9.5.2	<i>Phase-Locked Loop (PLL).....</i>	401
9.5.3	<i>Delay-Locked Loop (DLL)</i>	402
9.6	Selected DRAM Signaling and Timing Issues	402
9.6.1	<i>Data Read and Write Timing in DDRx SDRAM Memory Systems.....</i>	404
9.6.2	<i>The Use of DLL in DDRx SDRAM Devices</i>	406
9.6.3	<i>The Use of PLL in XDR DRAM Devices</i>	406
9.7	Summary.....	408

Chapter 10	DRAM Memory System Organization.....	409
10.1	Conventional Memory System	409
10.2	Basic Nomenclature	409
10.2.1	Channel.....	410
10.2.2	Rank.....	413
10.2.3	Bank.....	414
10.2.4	Row	415
10.2.5	Column.....	415
10.2.6	Memory System Organization: An Example.....	416
10.3	Memory Modules	417
10.3.1	Single In-line Memory Module (SIMM).....	418
10.3.2	Dual In-line Memory Module (DIMM).....	418
10.3.3	Registered Memory Module (RDIMM).....	418
10.3.4	Small Outline DIMM (SO-DIMM).....	419
10.3.5	Memory Module Organization	420
10.3.6	Serial Presence Detect (SPD).....	421
10.4	Memory System Topology.....	422
10.4.1	Direct RDRAM System Topology.....	422
10.5	Summary	423
Chapter 11	Basic DRAM Memory-Access Protocol	425
11.1	Basic DRAM Commands	425
11.1.1	Generic DRAM Command Format.....	427
11.1.2	Summary of Timing Parameters.....	427
11.1.3	Row Access Command	428
11.1.4	Column-Read Command.....	429
11.1.5	Column-Write Command	430
11.1.6	Precharge Command.....	431
11.1.7	Refresh Command	431
11.1.8	A Read Cycle	433
11.1.9	A Write Cycle.....	434
11.1.10	Compound Commands.....	434
11.2	DRAM Command Interactions.....	436
11.2.1	Consecutive Reads and Writes to Same Rank.....	437
11.2.2	Read to Precharge Timing	438
11.2.3	Consecutive Reads to Different Rows of Same Bank.....	438

11.2.4	<i>Consecutive Reads to Different Banks: Bank Conflict</i>	440
11.2.5	<i>Consecutive Read Requests to Different Banks</i>	441
11.2.6	<i>Consecutive Write Requests: Open Banks</i>	442
11.2.7	<i>Consecutive Write Requests: Bank Conflicts</i>	444
11.2.8	<i>Write Request Following Read Request: Open Banks</i>	444
11.2.9	<i>Write Request Following Read Request to Different Banks, Bank Conflict, Best Case, No Reordering</i>	445
11.2.10	<i>Read Following Write to Same Rank, Open Banks</i>	446
11.2.11	<i>Write to Precharge Timing</i>	447
11.2.12	<i>Read Following Write to Different Banks, Open Banks</i>	447
11.2.13	<i>Read Following Write to Same Bank, Bank Conflict</i>	448
11.2.14	<i>Read Following Write to Different Banks of Same Rank, Bank Conflict, Best Case, No Reordering</i>	449
11.2.15	<i>Column-Read-and-Precharge Command Timing</i>	449
11.2.16	<i>Column-Write-and-Precharge Timing</i>	450
11.3	Additional Constraints	450
11.3.1	<i>Device Power Limit</i>	450
11.3.2	t_{RRD} : <i>Row-to-Row (Activation) Delay</i>	452
11.3.3	t_{FAW} : <i>Four-Bank Activation Window</i>	453
11.3.4	<i>2T Command Timing in Unbuffered Memory Systems</i>	454
11.4	Command Timing Summary	454
11.5	Summary	454

Chapter 12	Evolutionary Developments of DRAM Device Architecture	457
12.1	DRAM Device Families	457
12.1.1	<i>Cost (Capacity), Latency, Bandwidth, and Power</i>	457
12.2	Historical-Commodity DRAM Devices	458
12.2.1	<i>The Intel 1103</i>	459
12.2.2	<i>Asynchronous DRAM Devices</i>	461
12.2.3	<i>Page Mode and Fast Page Mode DRAM (FPM DRAM)</i>	461
12.2.4	<i>Extended Data-Out (EDO) and Burst Extended Data-Out (BEDO) Devices</i>	463
12.3	Modern-Commodity DRAM Devices	464
12.3.1	<i>Synchronous DRAM (SDRAM)</i>	465

12.3.2	<i>Double Data Rate SDRAM (DDR SDRAM)</i>	471
12.3.3	<i>DDR2 SDRAM</i>	474
12.3.4	<i>Protocol and Architectural Differences</i>	475
12.3.5	<i>DDR3 SDRAM</i>	476
12.3.6	<i>Scaling Trends of Modern-Commodity DRAM Devices</i>	477
12.4	High Bandwidth Path	480
12.4.1	<i>Direct RDRAM</i>	480
12.4.2	<i>Technical and Pseudo-Technical Issues of Direct RDRAM</i>	487
12.4.3	<i>XDR Memory System</i>	491
12.5	Low Latency	494
12.5.1	<i>Reduced Latency DRAM (RLDRAM)</i>	494
12.5.2	<i>Fast Cycle DRAM (FCRAM)</i>	495
12.6	Interesting Alternatives	495
12.6.1	<i>Virtual Channel Memory (VCDRAM)</i>	495
12.6.2	<i>Enhanced SDRAM (ESDRAM)</i>	496
Chapter 13	DRAM Memory Controller	497
13.1	DRAM Controller Architecture	497
13.2	Row-Buffer-Management Policy	499
13.2.1	<i>Open-Page Row-Buffer-Management Policy</i>	499
13.2.2	<i>Close-Page Row-Buffer-Management Policy</i>	499
13.2.3	<i>Hybrid (Dynamic) Row-Buffer-Management Policies</i>	500
13.2.4	<i>Performance Impact of Row-Buffer-Management Policies</i>	500
13.2.5	<i>Power Impact of Row-Buffer-Management Policies</i>	501
13.3	Address Mapping (Translation)	502
13.3.1	<i>Available Parallelism in Memory System Organization</i>	503
13.3.2	<i>Parameter of Address Mapping Schemes</i>	504
13.3.3	<i>Baseline Address Mapping Schemes</i>	505
13.3.4	<i>Parallelism vs. Expansion Capability</i>	506
13.3.5	<i>Address Mapping in the Intel 82955X MCH</i>	506
13.3.6	<i>Bank Address Aliasing (Stride Collision)</i>	510
13.4	Performance Optimization	511
13.4.1	<i>Write Caching</i>	512
13.4.2	<i>Request Queue Organizations</i>	513
13.4.3	<i>Refresh Management</i>	514

	13.4.4	<i>Agent-Centric Request Queuing Organization</i>	516
	13.4.5	<i>Feedback-Directed Scheduling</i>	518
	13.5	Summary	518
Chapter 14		The Fully Buffered DIMM Memory System	519
	14.1	Introduction	519
	14.2	Architecture	521
	14.3	Signaling and Timing	524
	14.3.1	<i>Clock Data Recovery</i>	524
	14.3.2	<i>Unit Interval</i>	525
	14.3.3	<i>Resample and Resync</i>	525
	14.4	Access Protocol	526
	14.4.1	<i>Frame Definitions</i>	527
	14.4.2	<i>Command Definitions</i>	528
	14.4.3	<i>Frame and Command Scheduling</i>	528
	14.5	The Advanced Memory Buffer	530
	14.5.1	<i>SMBus Interface</i>	531
	14.5.2	<i>Built-In Self-Test (BIST)</i>	532
	14.5.3	<i>Thermal Sensor</i>	532
	14.6	Reliability, Availability, and Serviceability	532
	14.6.1	<i>Checksum Protection in the Transport Layer</i>	532
	14.6.2	<i>Bit Lane Steering</i>	533
	14.6.3	<i>Fail-over Modes</i>	534
	14.6.4	<i>Hot Add and Replace</i>	534
	14.7	FBDIMM Performance Characteristics	535
	14.7.1	<i>Fixed vs. Variable Latency Scheduling</i>	538
	14.8	Perspective	540
Chapter 15		Memory System Design Analysis	541
	15.1	Overview	541
	15.2	Workload Characteristics	543
	15.2.1	<i>164.zip: C Compression</i>	544
	15.2.2	<i>176.gcc: C Programming Language Compiler</i>	545
	15.2.3	<i>197.parser: C Word Processing</i>	545
	15.2.4	<i>255.vortex: C Object-Oriented Database</i>	546

15.2.5	<i>172.mgrid: Fortran 77 Multi-Grid Solver: 3D Potential Field</i>	547
15.2.6	<i>SETI@HOME</i>	547
15.2.7	<i>Quake 3</i>	548
15.2.8	<i>178.galgel, 179.art, 183.earthquake, 188.ammmp, JMark 2.0, and 3DWinbench</i>	548
15.2.9	<i>Summary of Workload Characteristics</i>	550
15.3	The RAD Analytical Framework	551
15.3.1	<i>DRAM-Access Protocol</i>	551
15.3.2	<i>Computing DRAM Protocol Overheads</i>	551
15.3.3	<i>Computing Row Cycle Time Constraints</i>	553
15.3.4	<i>Computing Row-to-Row Activation Constraints</i>	555
15.3.5	<i>Request Access Distance Efficiency Computation</i>	557
15.3.6	<i>An Applied Example for a Close-Page System</i>	558
15.3.7	<i>An Applied Example for an Open-Page System</i>	558
15.3.8	<i>System Configuration for RAD-Based Analysis</i>	559
15.3.9	<i>Open-Page Systems: 164.gzip</i>	561
15.3.10	<i>Open-Page Systems: 255.vortex</i>	562
15.3.11	<i>Open-Page Systems: Average of All Workloads</i>	562
15.3.12	<i>Close-Page Systems: 164.gzip</i>	565
15.3.13	<i>Close-Page Systems: SETI@HOME Processor Bus Trace</i>	566
15.3.14	<i>Close-Page Systems: Average of All Workloads</i>	567
15.3.15	<i>t_{FAW} Limitations in Open-Page System: All Workloads</i>	568
15.3.16	<i>Bandwidth Improvements: 8-Banks vs. 16-Banks</i>	568
15.4	Simulation-Based Analysis	570
15.4.1	<i>System Configurations</i>	570
15.4.2	<i>Memory Controller Structure</i>	571
15.4.3	<i>DRAM Command Scheduling Algorithms</i>	572
15.4.4	<i>Workload Characteristics</i>	575
15.4.5	<i>Timing Parameters</i>	575
15.4.6	<i>Protocol Table</i>	575
15.4.7	<i>Queue Depth, Scheduling Algorithms, and Burst Length</i>	577
15.4.8	<i>Effect of Burst Length on Sustainable Bandwidth</i>	578
15.4.9	<i>Burst Chop in DDR3 SDRAM Devices</i>	579
15.4.10	<i>Revisiting the 8-Bank and 16-Bank Issue with DRAMSim</i>	584
15.4.11	<i>8 Bank vs. 16 Banks — Relaxed t_{FAW} and t_{WTR}</i>	587
15.4.12	<i>Effect of Transaction Ordering on Latency Distribution</i>	587

15.5 A Latency-Oriented Study 590
 15.5.1 *Experimental Framework*..... 590
 15.5.2 *Simulation Input*..... 592
 15.5.3 *Limit Study: Latency Bandwidth Characteristics*..... 592
 15.5.4 *Latency*..... 593
15.6 Concluding Remarks 596

Part III Disk 599

Chapter 16 Overview of Disks 601
16.1 History of Disk Drives..... 601
 16.1.1 *Evolution of Drives*..... 603
 16.1.2 *Areal Density Growth Trend*..... 605
16.2 Principles of Hard Disk Drives 606
 16.2.1 *Principles of Rotating Storage Devices*..... 607
 16.2.2 *Magnetic Rotating Storage Device—Hard Disk Drive*..... 608
16.3 Classifications of Disk Drives..... 609
 16.3.1 *Form Factor* 609
 16.3.2 *Application*..... 609
 16.3.3 *Interface*..... 609
16.4 Disk Performance Overview 610
 16.4.1 *Disk Performance Metrics*..... 610
 16.4.2 *Workload Factors Affecting Performance*..... 612
 16.4.3 *Video Application Performance*..... 612
16.5 Future Directions in Disks 612

Chapter 17 The Physical Layer..... 615
17.1 Magnetic Recording..... 615
 17.1.1 *Ferromagnetism*..... 615
 17.1.2 *Magnetic Fields* 616
 17.1.3 *Hysteresis Loop* 618
 17.1.4 *Writing*..... 618
 17.1.5 *Reading*..... 620

17.2	Mechanical and Magnetic Components	620
17.2.1	<i>Disks</i>	621
17.2.2	<i>Spindle Motor</i>	623
17.2.3	<i>Heads.....</i>	625
17.2.4	<i>Slider and Head-Gimbal Assembly.....</i>	631
17.2.5	<i>Head-Stack Assembly and Actuator</i>	633
17.2.6	<i>Multiple Platters</i>	635
17.2.7	<i>Start/Stop.....</i>	636
17.2.8	<i>Magnetic Disk Recording Integration.....</i>	637
17.2.9	<i>Head-Disk Assembly.....</i>	639
17.3	Electronics	640
17.3.1	<i>Controller</i>	640
17.3.2	<i>Memory</i>	642
17.3.3	<i>Recording Channel.....</i>	642
17.3.4	<i>Motor Controls.....</i>	646
Chapter 18	The Data Layer	649
18.1	Disk Blocks and Sectors.....	649
18.1.1	<i>Fixed-Size Blocks</i>	649
18.1.2	<i>Variable Size Blocks</i>	650
18.1.3	<i>Sectors.....</i>	650
18.2	Tracks and Cylinders.....	652
18.3	Address Mapping	654
18.3.1	<i>Internal Addressing</i>	654
18.3.2	<i>External Addressing.....</i>	654
18.3.3	<i>Logical Address to Physical Location Mapping</i>	655
18.4	Zoned-Bit Recording.....	658
18.4.1	<i>Handling ZBR.....</i>	661
18.5	Servo.....	662
18.5.1	<i>Dedicated Servo</i>	662
18.5.2	<i>Embedded Servo</i>	663
18.5.3	<i>Servo ID and Seek</i>	666
18.5.4	<i>Servo Burst and Track Following.....</i>	667
18.5.5	<i>Anatomy of a Servo.....</i>	669
18.5.6	<i>ZBR and Embedded Servo.....</i>	669

18.6	Sector ID and No-ID Formatting	670
18.7	Capacity.....	672
18.8	Data Rate	673
18.9	Defect Management.....	673
	18.9.1 Relocation Schemes.....	674
	18.9.2 Types of Defects.....	675
	18.9.3 Error Recovery Procedure.....	676
Chapter 19	Performance Issues and Design Trade-Offs	677
19.1	Anatomy of an I/O	677
	19.1.1 Adding It All Up.....	679
19.2	Some Basic Principles.....	681
	19.2.1 Effect of User Track Capacity	681
	19.2.2 Effect of Cylinder Capacity.....	682
	19.2.3 Effect of Track Density.....	684
	19.2.4 Effect of Number of Heads.....	686
19.3	BPI vs. TPI.....	688
19.4	Effect of Drive Capacity.....	689
	19.4.1 Space Usage Efficiency	690
	19.4.2 Performance Implication.....	690
19.5	Concentric Tracks vs. Spiral Track.....	692
	19.5.1 Optical Disks.....	693
19.6	Average Seek.....	694
	19.6.1 Disks Without ZBR.....	694
	19.6.2 Disks With ZBR.....	695
Chapter 20	Drive Interface	699
20.1	Overview of Interfaces	699
	20.1.1 Components of an Interface.....	701
	20.1.2 Desirable Characteristics of Interface.....	701
20.2	ATA.....	702
20.3	Serial ATA.....	703
20.4	SCSI.....	705
20.5	Serial SCSI	706
20.6	Fibre Channel.....	707
20.7	Cost, Performance, and Reliability.....	709

Chapter 21	Operational Performance Improvement	711
21.1	Latency Reduction Techniques	711
21.1.1	<i>Dual Actuator</i>	711
21.1.2	<i>Multiple Copies.....</i>	712
21.1.3	<i>Zero Latency Access.....</i>	713
21.2	Command Queueing and Scheduling.....	715
21.2.1	<i>Seek Time-Based Scheduling</i>	716
21.2.2	<i>Total Access Time Based Scheduling.....</i>	717
21.2.3	<i>Sequential Access Scheduling.....</i>	723
21.3	Reorganizing Data on the Disk	723
21.3.1	<i>Defragmentation</i>	724
21.3.2	<i>Frequently Accessed Files.....</i>	724
21.3.3	<i>Co-Locating Access Clusters</i>	725
21.3.4	<i>ALIS.....</i>	726
21.4	Handling Writes	728
21.4.1	<i>Log-Structured Write</i>	728
21.4.2	<i>Disk Buffering of Writes.....</i>	729
21.5	Data Compression	729
Chapter 22	The Cache Layer	731
22.1	Disk Cache	731
22.1.1	<i>Why Disk Cache Works.....</i>	731
22.1.2	<i>Cache Automation</i>	732
22.1.3	<i>Read Cache, Write Cache</i>	732
22.2	Cache Organizations	735
22.2.1	<i>Desirable Features of Cache Organization</i>	735
22.2.2	<i>Fixed Segmentation</i>	736
22.2.3	<i>Circular Buffer</i>	737
22.2.4	<i>Virtual Memory Organization</i>	738
22.3	Caching Algorithms.....	741
22.3.1	<i>Perspective of Prefetch</i>	741
22.3.2	<i>Lookahead Prefetch</i>	742
22.3.3	<i>Look-behind Prefetch</i>	742
22.3.4	<i>Zero Latency Prefetch</i>	743

22.3.5 *ERP During Prefetch*..... 743
 22.3.6 *Handling of Sequential Access*..... 743
 22.3.7 *Replacement Policies*..... 745

Chapter 23 Performance Testing 747

23.1 Test and Measurement..... 747
 23.1.1 *Test Initiator* 747
 23.1.2 *Monitoring and Measuring* 749
 23.1.3 *The Test Drive* 750
23.2 Basic Tests 750
 23.2.1 *Media Data Rate*..... 751
 23.2.2 *Disk Buffer Data Rate*..... 751
 23.2.3 *Sequential Performance*..... 752
 23.2.4 *Random Performance* 752
 23.2.5 *Command Reordering Performance* 755
23.3 Benchmark Tests..... 755
 23.3.1 *Guidelines for Benchmarking*..... 756
23.4 Drive Parameters Tests 757
 23.4.1 *Geometry and More*..... 757
 23.4.2 *Seek Time* 759

Chapter 24 Storage Subsystems 763

24.1 Data Striping..... 763
24.2 Data Mirroring..... 765
 24.2.1 *Basic Mirroring*..... 766
 24.2.2 *Chained Decluster Mirroring*..... 766
 24.2.3 *Interleaved Decluster Mirroring* 767
 24.2.4 *Mirroring Performance Comparison* 767
 24.2.5 *Mirroring Reliability Comparison*..... 769
24.3 RAID 770
 24.3.1 *RAID Levels* 770
 24.3.2 *RAID Performance*..... 774
 24.3.3 *RAID Reliability*..... 775
 24.3.4 *Sparing*..... 776

24.3.5	RAID Controller.....	778
24.3.6	Advanced RAIDs.....	779
24.4	SAN	780
24.5	NAS	782
24.6	ISCSI	783
Chapter 25	Advanced Topics	785
25.1	Perpendicular Recording.....	785
25.1.1	Write Process, Write Head, and Media.....	786
25.1.2	Read Process and Read Head.....	788
25.2	Patterned Media.....	788
25.2.1	Fully Patterned Media.....	789
25.2.2	Discrete Track Media.....	790
25.3	Thermally Assisted Recording.....	790
25.4	Dual Stage Actuator	792
25.4.1	Microactuators	793
25.5	Adaptive Formatting.....	794
25.6	Hybrid Disk Drive.....	796
25.6.1	Benefits.....	796
25.6.2	Architecture.....	796
25.6.3	Proposed Interface.....	797
25.7	Object-Based Storage.....	798
25.7.1	Object Storage Main Concept	798
25.7.2	Object Storage Benefits.....	800
Chapter 26	Case Study	803
26.1	The Mechanical Components	803
26.1.1	Seek Profile.....	804
26.2	Electronics.....	806
26.3	Data Layout	806
26.3.1	Data Rate	808
26.4	Interface	809
26.5	Cache	810
26.6	Performance Testing.....	810

26.6.1	<i>Sequential Access</i>	810
26.6.2	<i>Random Access</i>	811

Part IV Cross-Cutting Issues 813

Chapter 27 The Case for Holistic Design 815

27.1	Anecdotes, Revisited	816
27.1.1	<i>Anecdote I: Systemic Behaviors Exist</i>	816
27.1.2	<i>Anecdote II: The DLL in DDR SDRAM</i>	818
27.1.3	<i>Anecdote III: A Catch-22 in the Search for Bandwidth</i>	822
27.1.4	<i>Anecdote IV: Proposals to Exploit Variability in Cell Leakage</i>	824
27.2	Perspective	827

Chapter 28 Analysis of Cost and Performance 829

28.1	Combining Cost and Performance	829
28.2	Pareto Optimality	830
28.2.1	<i>The Pareto-Optimal Set: An Equivalence Class</i>	830
28.2.2	<i>Stanley's Observation</i>	832
28.3	Taking Sampled Averages Correctly	833
28.3.1	<i>Sampling Over Time</i>	834
28.3.2	<i>Sampling Over Distance</i>	835
28.3.3	<i>Sampling Over Fuel Consumption</i>	836
28.3.4	<i>The Moral of the Story</i>	837
28.4	Metrics for Computer Performance	838
28.4.1	<i>Performance and the Use of Means</i>	838
28.4.2	<i>Problems with Normalization</i>	839
28.4.3	<i>The Meaning of Performance</i>	842
28.5	Analytical Modeling and the Miss-Rate Function	843
28.5.1	<i>Analytical Modeling</i>	843
28.5.2	<i>The Miss-Rate Function</i>	844

Chapter 29 Power and Leakage 847

29.1	Sources of Leakage in CMOS Devices	847
29.2	A Closer Look at Subthreshold Leakage	851

29.3	vCACTI and Energy/Power Breakdown of Pipelined Nanometer Caches	855
29.3.1	<i>Leakage in SRAM Cells</i>	855
29.3.2	<i>Pipelined Caches</i>	857
29.3.3	<i>Modeling</i>	858
29.3.4	<i>Dynamic and Static Power</i>	859
29.3.5	<i>Detailed Power Breakdown</i>	860
Chapter 30	Memory Errors and Error Correction	865
30.1	Types and Causes of Failures	865
30.1.1	<i>Alpha Particles</i>	866
30.1.2	<i>Primary Cosmic Rays and Terrestrial Neutrons</i>	866
30.1.3	<i>Soft Error Mechanism</i>	867
30.1.4	<i>Single-Bit and Multi-Bit Failures</i>	867
30.2	Soft Error Rates and Trends	868
30.3	Error Detection and Correction	869
30.3.1	<i>Parity</i>	869
30.3.2	<i>Single-Bit Error Correction (SEC ECC)</i>	870
30.3.3	<i>Single-Bit Error Correction, Double-Bit Error Detection (SEDED ECC)</i>	873
30.3.4	<i>Multi-Bit Error Detection and Correction: Bossen's b-Adjacent Algorithm</i>	874
30.3.5	<i>Bit Steering and Chipkill</i>	875
30.3.6	<i>Chipkill with $\times 8$ DRAM Devices</i>	877
30.3.7	<i>Memory Scrubbing</i>	879
30.3.8	<i>Bullet Proofing the Memory System</i>	880
30.4	Reliability of Non-DRAM Systems	880
30.4.1	<i>SRAM</i>	880
30.4.2	<i>Flash</i>	880
30.4.3	<i>MRAM</i>	880
30.5	Space Shuttle Memory System	881
Chapter 31	Virtual Memory	883
31.1	A Virtual Memory Primer	884
31.1.1	<i>Address Spaces and the Main Memory Cache</i>	885
31.1.2	<i>Address Mapping and the Page Table</i>	886
31.1.3	<i>Hierarchical Page Tables</i>	887

31.1.4	<i>Inverted Page Tables</i>	890
31.1.5	<i>Comparison: Inverted vs. Hierarchical</i>	892
31.1.6	<i>Translation Lookaside Buffers, Revisited</i>	893
31.1.7	<i>Perspective: Segmented Addressing Solves the Synonym Problem</i>	895
31.1.8	<i>Perspective: A Taxonomy of Address Space Organizations</i>	901
31.2	Implementing Virtual Memory	906
31.2.1	<i>The Basic In-Order Pipe</i>	908
31.2.2	<i>Precise Interrupts in Pipelined Computers</i>	910
References		921
Index		955



Preface

“It’s the Memory, Stupid!”

If you develop an ear for sounds that are musical it is like developing an ego. You begin to refuse sounds that are not musical and that way cut yourself off from a good deal of experience.

—John Cage

In 1996, Richard Sites, one of the fathers of computer architecture and lead designers of the DEC Alpha, had the following to say about the future of computer architecture research:

Across the industry, today’s chips are largely able to execute code faster than we can feed them with instructions and data. There are no longer performance bottlenecks in the floating-point multiplier or in having only a single integer unit. The real design action is in memory subsystems—caches, buses, bandwidth, and latency.

An anecdote: in a recent database benchmark study using TPC-C, both 200-MHz Pentium Pro and 400MHz 21164 Alpha systems were measured at 4.2–4.5 CPU cycles per instruction retired. In other words, three out of every four CPU cycles retired zero instructions: most were spent waiting for memory. Processor speed has seriously outstripped memory speed.

Increasing the width of instruction issue and increasing the number of simultaneous instruction streams only makes the memory bottleneck worse. If a CPU chip today needs to move 2 GBytes/s (say, 16 bytes every 8 ns) across the pins to keep itself busy, imagine a chip in the foreseeable future with twice the clock rate, twice the issue width, and two instruction

streams. All these factors multiply together to require about 16 GBytes/s of pin bandwidth to keep this chip busy. It is not clear whether pin bandwidth can keep up—32 bytes every 2ns?

*I expect that over the coming decade memory subsystems design will be the **only** important design issue for microprocessors. [Sites 1996, emphasis Sites’]*

The title of Sites’ article is “It’s the Memory, Stupid!” Sites realized in 1996 what we as a community are only now, more than a decade later, beginning to digest and internalize fully: *uh, guys, it really is the memory system ... little else matters right now, so stop wasting time and resources on other facets of the design.* Most of his colleagues designing next-generation Alpha architectures at Digital Equipment Corp. ignored his advice and instead remained focused on building ever faster microprocessors, rather than shifting their focus to the building of ever faster *systems*. It is perhaps worth noting that Digital Equipment Corp. no longer exists.

The increasing gap between processor and memory speeds has rendered the organization, architecture, and design of memory subsystems an increasingly important part of computer-systems design. Today, the divide is so severe we are now in one of those down-cycles where the processor is so good at

xxxi

number-crunching it has completely sidelined itself; it is too fast for its own good, in a sense. Sites' prediction came true: memory subsystems design is now and has been for several years the *only* important design issue for microprocessors and systems. Memory-hierarchy parameters affect system performance *significantly* more than processor parameters (e.g., they are responsible for 2–10× changes in execution time, as opposed to 2–10%), making it absolutely essential for any designer of computer systems to exhibit an in-depth knowledge of the memory system's organization, its operation, its not-so-obvious behavior, and its range of performance characteristics. This is true now, and it is likely to remain true in the near future.

Thus this book, which is intended to provide exactly that type of in-depth coverage over a wide range of topics.

Topics Covered

In the following chapters we address the logical design and operation, the physical design and operation, the performance characteristics (i.e., design trade-offs), and, to a limited extent, the energy consumption of modern memory hierarchies.

In the cache section, we present topics and perspectives that will be new (or at least interesting) to even veterans in the field. What this implies is that the cache section is *not* an overview of processor-cache organization and its effect on performance—instead, we build up the concept of cache from first principles and discuss topics that are incompletely covered in the computer-engineering literature. The section discusses a significant degree of historical development in cache-management techniques, the physical design of modern SRAM structures, the operating system's role in cache coherence, and the continuum of cache architectures from those that are fully transparent (to application software and/or the operating system) to those that are fully visible.

DRAM and disk are interesting technologies because, unlike caches, they are not typically integrated onto the microprocessor die. Thus any discussion of these topics necessarily deals with the

issue of communication: e.g., channels, signalling, protocols, and request scheduling.

DRAM involves one or more chip-to-chip crossings, and so signalling and signal integrity are as fundamental as circuit design to the technology. In the DRAM section, we present an intuitive understanding of exactly what happens inside the DRAM so that the ubiquitous parameters of the interface (e.g., t_{RC} , t_{RCD} , t_{CAS} , etc.) will make sense. We survey the various DRAM architectures that have appeared over the years and give an in-depth description of the technologies in the next generation memory-system architecture. We discuss memory-controller issues and investigate performance issues of modern systems.

The disk section builds from the bottom up, providing a view of the disk from physical recording principles to the configuration and operation of disks within system settings. We discuss the operation of the disk's read/write heads; the arrangement of recording media within the enclosure; and the organization-level view of blocks, sectors, tracks, and cylinders, as well as various protocols used to encode data. We discuss performance issues and techniques used to improve performance, including caching and buffering, prefetching, request scheduling, and data reorganization. We discuss the various disk interfaces available today (e.g., ATA, serial ATA, SCSI, fibre channel, etc.) as well as system configurations such as RAID, SAN, and NAS.

The last section of the book, *Cross-Cutting Issues*, covers topics that apply to all levels of the memory hierarchy, such as the tools of analysis and how to use them correctly, subthreshold leakage power in CMOS devices and circuits, a look at power breakdowns in future SRAMs, codes for error detection and error correction, the design and operation of virtual memory systems, and the hardware mechanisms that are required in microprocessors to support virtual memory.

Goals and Audience

The primary goal of this book is to bring the reader to a level of understanding at which the physical design and/or detailed software emulation of the entire hierarchy is possible, from cache to disk. As we argue in the initial chapter, this level of understanding

is important now and will become increasingly necessary over time. Another goal of the book is to discuss techniques of analysis, so that the next generation of design engineers is prepared to tackle the nontrivial multidimensional optimization problems that result from considering detailed side-effects that can manifest themselves at any point in the entire hierarchy.

Accordingly, our target audience are those planning to build and/or optimize memory systems: i.e., computer-engineering and computer-science faculty and graduate students (and perhaps advanced undergraduates) and developers in the computer design, peripheral design, and embedded systems industries.

As an educational textbook, this is targeted at graduate and undergraduate students with a solid background in computer organization and architecture. It could serve to support an advanced senior-level undergraduate course or a second-year graduate course specializing in computer-systems design. There is clearly far too much material here for any single course; the book provides depth on enough topics to support two to three separate courses. For example, at the University of Maryland we use the DRAM section to teach a graduate class called *High-Speed Memory Systems*, and we supplement both our general and advanced architecture classes with material from the sections on *Caches* and *Cross-Cutting Issues*. The *Disk* section could support a class focused solely on disks, and it is also possible to create for advanced students a survey class that lightly touches on all the topics in the book.

As a reference, this book is targeted toward both academics and professionals alike. It provides the breadth necessary to understand the wide scope of behaviors that appear in modern memory systems, and most of the topics are addressed in enough depth that a reader should be able to build (or at least model in significant detail) caches, DRAMs, disks, their controllers, their subsystems ... and understand their interactions.

What this means is that the book should not only be useful to developers, but it should also be useful to those responsible for long-range planning and forecasting for future product developments and their issues.

Acknowledgments and Thanks

Beyond the kind folks named in the book's Dedication, we have a lot of people to thank. The following people were enormously helpful in creating the contents of this book:

- Prof. Rajeev Barua, ECE Maryland, provided text to explain scratch-pad memories and their accompanying partitioning problem (see Chapter 3).
- Dr. Brinda Ganesh, a former graduate student in Bruce Jacob's research group, now at Intel DAP, wrote the latency-oriented section of the DRAM-performance chapter (see Section 15.5).
- Joseph Gross, a graduate student in Bruce Jacob's research group, updated the numbers in David Wang's Ph.D. proposal to produce the tables comparing the characteristics of modern DRAMs in the book's *Overview* chapter.
- Dr. Ed Growchowski, formerly of Hitachi Global Storage Technologies and currently a consultant for IDEMA, provided some of the disk recording density history charts.
- Dr. Martin Hassner, Dr. Bruce Wilson, Dr. Matt White, Chuck Cox (now with IBM), Frank Chu, and Tony Dunn of Hitachi Global Storage Technologies provided important consultation on various details of disk drive design.
- Dr. Windsor Hsu, formerly of IBM Almaden Research Center and now with Data Domain, Inc., provided important consultation on system issues related to disk drives.
- Dr. Aamer Jaleel, a former graduate student in Bruce Jacob's research group, now at Intel VSSAD, is responsible for Chapter 4's sections on cache coherence.
- Michael Martin, a graduate student in Bruce Jacob's research group, executed simulations for the final table (Ov. 5) and formatted the four large time-sequence graphs in the *Overview* chapter.
- Rami Nasr, a graduate student at Maryland who wrote his M.S. thesis on the Fully

Buffered DIMM architecture, provided much of the contents of Chapter 14.

- Prof. Marty Peckerar, ECE Maryland, provided brilliant text and illustrations to explain the subthreshold leakage problem (see the book's *Overview* and Section 29.2).
- Profs. Yan Solihin, ECE NCSU, and Donald Yeung, ECE Maryland, provided much of the material on hardware and software prefetching (Sections 3.1.2 and 3.2.2); this came from a book chapter on the topic written by the pair for Kaeli and Yew's *Speculative Execution in High Performance Computer Architectures* (CRC Press, 2005).
- Sadagopan Srinivasan, a graduate student in Bruce Jacob's research group, performed the study at the end of Chapter 1 and provided much insight on the memory behavior of both streaming applications and multi-core systems.
- Dr. Nuengwong Tuaycharoen, a former graduate student in Bruce Jacob's research group, now at Thailand's Dhurakijpundit University, performed the experiments, and wrote the example holistic analysis at the end of the book's *Overview* chapter, directly relating to the behavior of caches, DRAMs, and disks in a single study.
- Patricia Wadkins and others in the Rochester SITLab of Hitachi Global Storage Technologies provided test results and measurement data for the book's Section III on *Disks*.
- Mr. Yong Wang, a signal integrity expert, formerly of HP and Intel, now with MetaRAM, contributed extensively to the Chapter on *DRAM System Signaling and Timing* (Chapter 9).
- Michael Xu at Hitachi Global Storage Technologies drew the beautiful, complex illustrations, as well as provided some of the photographs, in the book's Section III on *Disks*.

In addition, several students involved in tool support over the years deserve special recognition:

- Brinda Ganesh took the reins from David Wang to maintain *DRAMsim*; among other

things, she is largely responsible for the FB-DIMM support in that tool.

- Joseph Gross also supports *DRAMsim* and is leading the development of the second generation version of the software, which is object-oriented and significantly streamlined.
- Nuengwong Tuaycharoen integrated the various disparate software modules to produce *SYSim*, a full-system simulator that gave us the wonderful time-sequence graphs at the end of the *Overview*.

Numerous reviewers spent considerable time reading early drafts of the manuscript and providing excellent critiques or our direction, approach, and raw content. The following reviewers directly contributed to the book in this way: Ashraf Abounnaga, *University of Waterloo*; Al Davis, *University of Utah*; Diana Franklin, *California Polytechnic University, San Luis Obispo*; Yiming Hu, *University of Cincinnati*; David Kaeli, *Northeastern University*; Nagi Mekhiel, *Ryerson University, Toronto*; Michael Schuette, Ph.D., VP of Technology Development at *OCZ Technology*; Jim Smith, *University of Wisconsin—Madison*; Yan Solin, *North Carolina State University*; and Several Anonymous reviewers (i.e., we the authors were told not to use their names).

The editorial and production staff at Morgan Kaufmann/Elsevier was amazing to work with: Denise Penrose pitched the idea to us in the first place (and enormous thanks go to Jim Smith, who pointed her in our direction) and Nate McFadden and Paul Gottehrer made the process of writing, editing, and proofing go incredibly smoothly.

Lastly, Dr. Richard Matick, the author of the original memory-systems book (*Computer Storage Systems and Technology*, John Wiley & Sons, 1976) and currently a leading researcher in embedded DRAM at IBM T. J. Watson, provided enormous help in direction, focus, and approach.

Dave, Spencer, Sam, and I are indebted to all of these writers, illustrators, coders, editors, and reviewers alike; they all helped to make this book what it is. To those contributors: thank you. You rock.

Bruce Jacob, Summer 2007
College Park, Maryland