

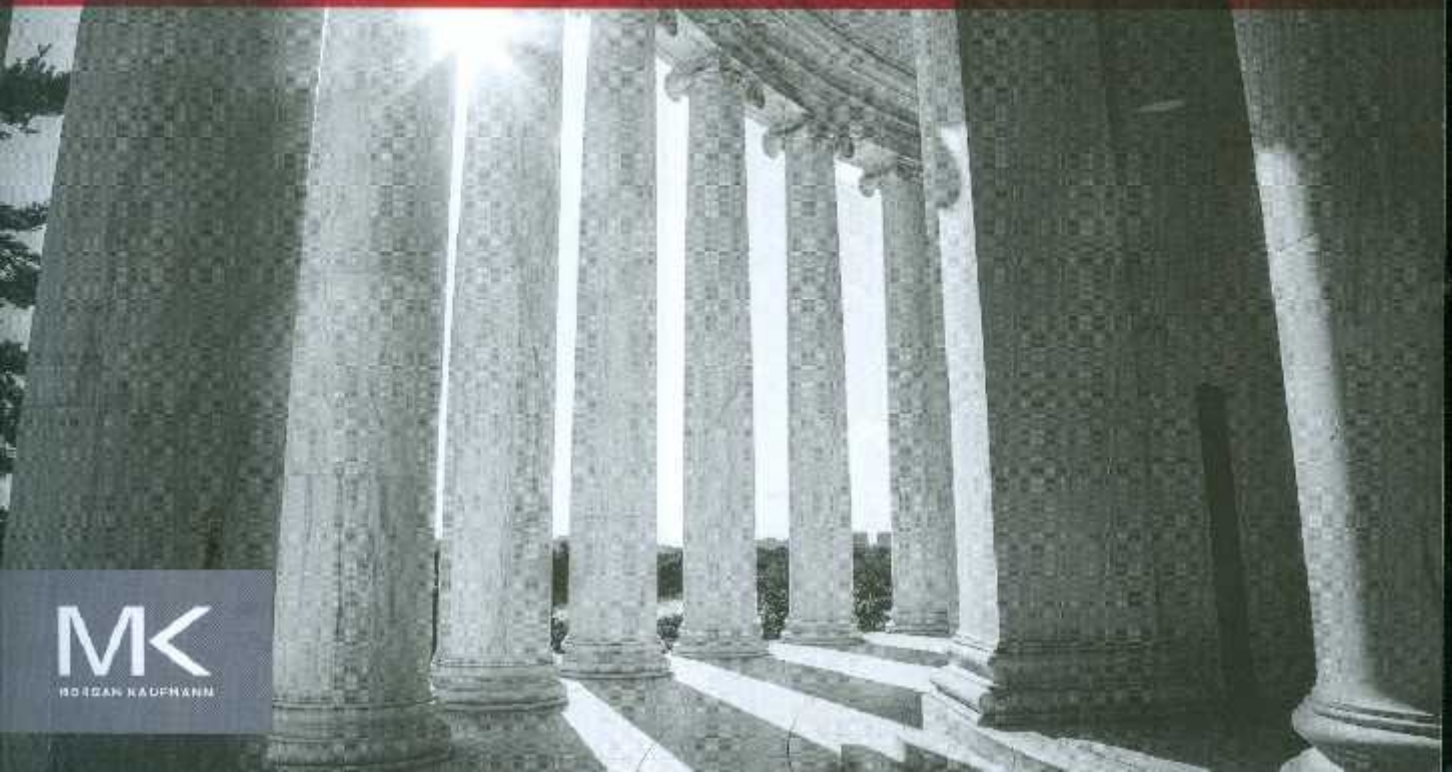
Sixth Edition

John L. Hennessy | David A. Patterson

# COMPUTER ARCHITECTURE

A Quantitative Approach

**MK**  
MORGAN KAUFMANN





# Contents

	<b>Foreword</b>	<b>ix</b>
	<b>Preface</b>	<b>xvii</b>
	<b>Acknowledgments</b>	<b>xxv</b>
<b>Chapter 1</b>	<b>Fundamentals of Quantitative Design and Analysis</b>	
	1.1 Introduction	2
	1.2 Classes of Computers	6
	1.3 Defining Computer Architecture	11
	1.4 Trends in Technology	18
	1.5 Trends in Power and Energy in Integrated Circuits	23
	1.6 Trends in Cost	29
	1.7 Dependability	36
	1.8 Measuring, Reporting, and Summarizing Performance	39
	1.9 Quantitative Principles of Computer Design	48
	1.10 Putting It All Together: Performance, Price, and Power	55
	1.11 Fallacies and Pitfalls	58
	1.12 Concluding Remarks	64
	1.13 Historical Perspectives and References	67
	Case Studies and Exercises by Diana Franklin	67
<b>Chapter 2</b>	<b>Memory Hierarchy Design</b>	
	2.1 Introduction	78
	2.2 Memory Technology and Optimizations	84
	2.3 Ten Advanced Optimizations of Cache Performance	94
	2.4 Virtual Memory and Virtual Machines	118
	2.5 Cross-Cutting Issues: The Design of Memory Hierarchies	126
	2.6 Putting It All Together: Memory Hierarchies in the ARM Cortex-A53 and Intel Core i7-6700	129
	2.7 Fallacies and Pitfalls	142
	2.8 Concluding Remarks: Looking Ahead	146
	2.9 Historical Perspectives and References	148

Case Studies and Exercises by Norman P. Jouppi, Rajeev Balasubramonian, Naveen Muralimanohar, and Sheng Li

148

### Chapter 3 **Instruction-Level Parallelism and Its Exploitation**

3.1	Instruction-Level Parallelism: Concepts and Challenges	168
3.2	Basic Compiler Techniques for Exposing ILP	176
3.3	Reducing Branch Costs With Advanced Branch Prediction	182
3.4	Overcoming Data Hazards With Dynamic Scheduling	191
3.5	Dynamic Scheduling: Examples and the Algorithm	201
3.6	Hardware-Based Speculation	208
3.7	Exploiting ILP Using Multiple Issue and Static Scheduling	218
3.8	Exploiting ILP Using Dynamic Scheduling, Multiple Issue, and Speculation	222
3.9	Advanced Techniques for Instruction Delivery and Speculation	228
3.10	Cross-Cutting Issues	240
3.11	Multithreading: Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput	242
3.12	Putting It All Together: The Intel Core i7 6700 and ARM Cortex-A53	247
3.13	Fallacies and Pitfalls	258
3.14	Concluding Remarks: What's Ahead?	264
3.15	Historical Perspective and References	266
	Case Studies and Exercises by Jason D. Bakos and Robert P. Colwell	266

### Chapter 4 **Data-Level Parallelism in Vector, SIMD, and GPU Architectures**

4.1	Introduction	282
4.2	Vector Architecture	283
4.3	SIMD Instruction Set Extensions for Multimedia	304
4.4	Graphics Processing Units	310
4.5	Detecting and Enhancing Loop-Level Parallelism	336
4.6	Cross-Cutting Issues	345
4.7	Putting It All Together: Embedded Versus Server GPUs and Tesla Versus Core i7	346
4.8	Fallacies and Pitfalls	353
4.9	Concluding Remarks	357
4.10	Historical Perspective and References	357
	Case Study and Exercises by Jason D. Bakos	357

### Chapter 5 **Thread-Level Parallelism**

5.1	Introduction	368
5.2	Centralized Shared-Memory Architectures	377
5.3	Performance of Symmetric Shared-Memory Multiprocessors	393

5.4	Distributed Shared-Memory and Directory-Based Coherence	404
5.5	Synchronization: The Basics	412
5.6	Models of Memory Consistency: An Introduction	417
5.7	Cross-Cutting Issues	422
5.8	Putting It All Together: Multicore Processors and Their Performance	426
5.9	Fallacies and Pitfalls	438
5.10	The Future of Multicore Scaling	442
5.11	Concluding Remarks	444
5.12	Historical Perspectives and References	445
	Case Studies and Exercises by Amir Zaky and David A. Wood	446
Chapter 6	<b>Warehouse-Scale Computers to Exploit Request-Level and Data-Level Parallelism</b>	
6.1	Introduction	466
6.2	Programming Models and Workloads for Warehouse-Scale Computers	471
6.3	Computer Architecture of Warehouse-Scale Computers	477
6.4	The Efficiency and Cost of Warehouse-Scale Computers	482
6.5	Cloud Computing: The Return of Utility Computing	490
6.6	Cross-Cutting Issues	501
6.7	Putting It All Together: A Google Warehouse-Scale Computer	503
6.8	Fallacies and Pitfalls	514
6.9	Concluding Remarks	518
6.10	Historical Perspectives and References	519
	Case Studies and Exercises by Parthasarathy Ranganathan	519
Chapter 7	<b>Domain-Specific Architectures</b>	
7.1	Introduction	540
7.2	Guidelines for DSAs	543
7.3	Example Domain: Deep Neural Networks	544
7.4	Google's Tensor Processing Unit, an Inference Data Center Accelerator	557
7.5	Microsoft Catapult, a Flexible Data Center Accelerator	567
7.6	Intel Crest, a Data Center Accelerator for Training	579
7.7	Pixel Visual Core, a Personal Mobile Device Image Processing Unit	579
7.8	Cross-Cutting Issues	592
7.9	Putting It All Together: CPUs Versus GPUs Versus DNN Accelerators	595
7.10	Fallacies and Pitfalls	602
7.11	Concluding Remarks	604
7.12	Historical Perspectives and References	606
	Case Studies and Exercises by Cliff Young	606