Contents

Foreword ix

Preface xv

Acknowledgments xxiii

Chapter 1 **Fundamentals of Quantitative Design and Analysis**

1.1 Introduction 2
1.2 Classes of Computers 5
1.3 Defining Computer Architecture 11
1.4 Trends in Technology 17
1.5 Trends in Power and Energy in Integrated Circuits 21
1.6 Trends in Cost 27
1.7 Dependability 33
1.8 Measuring, Reporting, and Summarizing Performance 36
1.9 Quantitative Principles of Computer Design 44
1.10 Putting It All Together: Performance, Price, and Power 52
1.11 Fallacies and Pitfalls 55
1.12 Concluding Remarks 59
1.13 Historical Perspectives and References 61
Case Studies and Exercises by Diana Franklin 61

Chapter 2 **Memory Hierarchy Design**

2.1 Introduction 72
2.2 Ten Advanced Optimizations of Cache Performance 78
2.3 Memory Technology and Optimizations 96
2.4 Protection: Virtual Memory and Virtual Machines 105
2.5 Crosscutting Issues: The Design of Memory Hierarchies 112
2.6 Putting It All Together: Memory Hierachies in the ARM Cortex-A8 and Intel Core i7 113
2.7 Fallacies and Pitfalls 125

2.8 Concluding Remarks: Looking Ahead 129
2.9 Historical Perspective and References 131
Case Studies and Exercises by Norman P. Jouppi, Naveen Muralimanohar, and Sheng Li 131

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

3.1 Instruction-Level Parallelism: Concepts and Challenges 148
3.2 Basic Compiler Techniques for Exposing ILP 156
3.3 Reducing Branch Costs with Advanced Branch Prediction 162
3.4 Overcoming Data Hazards with Dynamic Scheduling 167
3.5 Dynamic Scheduling: Examples and the Algorithm 176
3.6 Hardware-Based Speculation 183
3.7 Exploiting ILP Using Multiple Issue and Static Scheduling 192
3.8 Exploiting ILP Using Dynamic Scheduling, Multiple Issue, and Speculation 197
3.9 Advanced Techniques for Instruction Delivery and Speculation 202
3.10 Studies of the Limitations of ILP 213
3.11 Cross-Cutting Issues: ILP Approaches and the Memory System 221
3.12 Multithreading: Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput 223
3.13 Putting It All Together: The Intel Core i7 and ARM Cortex-A8 233
3.14 Fallacies and Pitfalls 241
3.15 Concluding Remarks: What's Ahead? 245
3.16 Historical Perspective and References 247
Case Studies and Exercises by Jason D. Bakos and Robert P. Colwell 247

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

4.1 Introduction 262
4.2 Vector Architecture 264
4.3 SIMD Instruction Set Extensions for Multimedia 282
4.4 Graphics Processing Units 288
4.5 Detecting and Enhancing Loop-Level Parallelism 315
4.6 Crosscutting Issues 322
4.7 Putting It All Together: Mobile versus Server GPUs and Tesla versus Core i7 323
4.8 Fallacies and Pitfalls 330
4.9 Concluding Remarks 332
4.10 Historical Perspective and References 334
Case Study and Exercises by Jason D. Bakos 334

Chapter 5 **Thread-Level Parallelism**

5.1 Introduction 344
5.2 Centralized Shared-Memory Architectures 351
5.3 Performance of Symmetric Shared-Memory Multiprocessors 366

5.4 Distributed Shared-Memory and Directory-Based Coherence 378
5.5 Synchronization: The Basics 386
5.6 Models of Memory Consistency: An Introduction 392
5.7 Crosscutting Issues 395
5.8 Putting It All Together: Multicore Processors and Their Performance 400
5.9 Fallacies and Pitfalls 405
5.10 Concluding Remarks 409
5.11 Historical Perspectives and References 412
Case Studies and Exercises by Amr Zaky and David A. Wood 412

Chapter 6 **Warehouse-Scale Computers to Exploit Request-Level and Data-Level Parallelism**

6.1 Introduction 432
6.2 Programming Models and Workloads for Warehouse-Scale Computers 436
6.3 Computer Architecture of Warehouse-Scale Computers 441
6.4 Physical Infrastructure and Costs of Warehouse-Scale Computers 446
6.5 Cloud Computing: The Return of Utility Computing 455
6.6 Crosscutting Issues 461
6.7 Putting It All Together: A Google Warehouse-Scale Computer 464
6.8 Fallacies and Pitfalls 471
6.9 Concluding Remarks 475
6.10 Historical Perspectives and References 476
Case Studies and Exercises by Parthasarathy Ranganathan 476

Appendix A **Instruction Set Principles**

A.1 Introduction A-2
A.2 Classifying Instruction Set Architectures A-3
A.3 Memory Addressing A-7
A.4 Type and Size of Operands A-13
A.5 Operations in the Instruction Set A-14
A.6 Instructions for Control Flow A-16
A.7 Encoding an Instruction Set A-21
A.8 Crosscutting Issues: The Role of Compilers A-24
A.9 Putting It All Together: The MIPS Architecture A-32
A.10 Fallacies and Pitfalls A-39
A.11 Concluding Remarks A-45
A.12 Historical Perspective and References A-47
Exercises by Gregory D. Peterson A-47

Appendix B **Review of Memory Hierarchy**

B.1 Introduction B-2
B.2 Cache Performance B-16
B.3 Six Basic Cache Optimizations B-22

	B.4	Virtual Memory	B-40
	B.5	Protection and Examples of Virtual Memory	B-49
	B.6	Fallacies and Pitfalls	B-57
	B.7	Concluding Remarks	B-59
	B.8	Historical Perspective and References	B-59
		Exercises by Amr Zaky	B-60
Appendix C	**Pipelining: Basic and Intermediate Concepts**		
	C.1	Introduction	C-2
	C.2	The Major Hurdle of Pipelining—Pipeline Hazards	C-11
	C.3	How Is Pipelining Implemented?	C-30
	C.4	What Makes Pipelining Hard to Implement?	C-43
	C.5	Extending the MIPS Pipeline to Handle Multicycle Operations	C-51
	C.6	Putting It All Together: The MIPS R4000 Pipeline	C-61
	C.7	Crosscutting Issues	C-70
	C.8	Fallacies and Pitfalls	C-80
	C.9	Concluding Remarks	C-81
	C.10	Historical Perspective and References	C-81
		Updated Exercises by Diana Franklin	C-82
	Online Appendices		
Appendix D	**Storage Systems**		
Appendix E	**Embedded Systems**		
	By Thomas M. Conte		
Appendix F	**Interconnection Networks**		
	Revised by Timothy M. Pinkston and José Duato		
Appendix G	**Vector Processors in More Depth**		
	Revised by Krste Asanovic		
Appendix H	**Hardware and Software for VLIW and EPIC**		
Appendix I	**Large-Scale Multiprocessors and Scientific Applications**		
Appendix J	**Computer Arithmetic**		
	by David Goldberg		
Appendix K	**Survey of Instruction Set Architectures**		
Appendix L	**Historical Perspectives and References**		
	References		**R-1**
	Index		**I-1**