

Contents

	<i>Preface</i>	<i>page xi</i>
1	Computing and the Transformation of Society	1
	1.1 Computing Transforms Society and Economy	1
	1.1.1 Home	2
	1.1.2 Automobiles and Transportation	2
	1.1.3 Commerce	3
	1.2 Computing Transforms Science and Discovery	5
	1.3 Extraordinary Characteristics of Computing	6
	1.4 What is Computer Architecture?	8
	1.5 Four Pillars of Computer Performance: Miniaturization, Hidden Parallelism, Dynamic Locality, and Explicit Parallelism	9
	1.6 Expected Background	10
	1.7 Organization of the Book	11
	1.8 Summary	13
	1.9 Problems	13
2	Instructions Sets, Software, and Instruction Execution	16
	2.1 Computer Instruction Sets	16
	2.2 Computer Systems Architecture	17
	2.3 Instruction Set Architecture: RISC-V Example	18
	2.3.1 Computation Instructions	19
	2.3.2 Conditional Control and Procedure Linkage Instructions	20
	2.3.3 Memory Instructions	22
	2.4 Machine Instructions and Basic Software Structures	23
	2.4.1 Implementing Basic Expressions	23
	2.4.2 Implementing Data Structures: Structs and Objects	23
	2.4.3 Implementing One- and Multi-dimensional Arrays	26
	2.4.4 Implementing Conditional Iterative Constructs: Loops	27
	2.4.5 Implementing Procedure Call and Return and the Stack	30
	2.5 Basic Instruction Execution and Implementation	33
	2.5.1 Sequential State and Instruction Execution	35
	2.5.2 Hardware Implementation of Instruction Execution	35
	2.6 Speeding Up Instruction Execution and Program Performance	36

2.7	Summary	37
2.8	Digging Deeper	38
2.9	Problems	39
3	Processors and Scaling: Small is Fast!	48
3.1	Miniaturization and Information Processing	48
3.2	What Is the Natural Size of a Computer?	50
3.2.1	Example: Bit Size and Speed	52
3.2.2	Shrinking Computers	53
3.3	Computer Size and Speed	53
3.3.1	Smaller Computers Are Faster	54
3.3.2	Example: Applying the Size and Clock Period Model	56
3.3.3	Size Scaling Computers from Room-Sized to a Single Chip	57
3.3.4	Size Scaling Single-Chip Computers: The Power Problem and Dennard's Solution	59
3.3.5	The End of Dennard Scaling	62
3.4	Computer Size and Power Consumption	62
3.5	Size in Other Technologies	64
3.6	Tiny Computers Enable an Explosion of Applications	65
3.7	Summary	67
3.8	Digging Deeper	68
3.9	Problems	70
4	Sequential Abstraction, But Parallel Implementation	76
4.1	Sequential Computation Abstraction	76
4.1.1	Sequential Programs	76
4.1.2	Instruction-Level Parallelism: Pipelining and More	79
4.1.3	Data Dependence and the Illusion of Sequence	80
4.2	The Illusion of Sequence: Renaming and Out-of-Order Execution	83
4.2.1	Variable and Register Renaming	85
4.2.2	Implementing Register Renaming: The Reorder Buffer	88
4.2.3	Limits of Out-of-Order Execution	89
4.3	Illusion of Causality: Speculative Execution	90
4.3.1	Branch Prediction	91
4.3.2	Speculative Execution	93
4.3.3	Accurate Branch Predictors	97
4.3.4	Security Risks of Speculation: Spectre and Meltdown	99
4.4	Summary	101
4.5	Digging Deeper	102
4.6	Problems	102
5	Memories: Exploiting Dynamic Locality	113
5.1	Memory Technologies, Miniaturization, and Growing Capacity	113
5.2	Software and Applications Demand Memory Capacity	118

5.3	Memory System Challenges: The Memory Wall	122
5.4	Memory Latency	122
5.4.1	Warping Space–Time (Caches)	123
5.4.2	Dynamic Locality in Programs	126
5.4.3	Address Filters (Caches)	128
5.4.4	The Effectiveness of Filters (Caches)	131
5.4.5	Implementing Caches (Warping and Filtering)	132
5.4.6	Recursive Filtering (Multi-level Caches)	133
5.4.7	Modeling Average Memory Hierarchy Performance	135
5.5	Why Caches Work so Well and Programming for Locality	137
5.6	Measuring Application Dynamic Locality and Modeling Performance	142
5.6.1	Measuring Dynamic Locality: Reuse Distance	142
5.6.2	Reuse Distance and Dynamic Locality	143
5.6.3	Modeling an Application’s Memory Performance Using Reuse Distance	145
5.6.4	Tuning a Program for Dynamic Locality	146
5.7	Access Rate and Parallel Memory Systems	148
5.8	Summary	150
5.9	Digging Deeper	151
5.10	Problems	152
6	The General Purpose Computer	161
6.1	A Commercial Processor: Intel Skylake	161
6.2	A Commercial Memory Hierarchy: Intel Skylake	164
6.2.1	Caches and Power	165
6.3	CPUs Are General Purpose Computers	168
6.4	Perspective: Mathematical Universality and Complexity	170
6.5	Summary	171
6.6	Digging Deeper	172
6.7	Problems	173
7	Beyond Sequential: Parallelism in MultiCore and the Cloud	176
7.1	The End of Dennard Scaling and the Shift to Parallelism	176
7.2	Parallel Single-Chip Computers: Multicore CPUs	179
7.2.1	Example: AMD Ryzen Multicore Chip and System	181
7.3	Programming Multicore Computers: OpenMP and pthreads	182
7.3.1	OpenMP: Pragma-Based Parallelism	183
7.3.2	pthreads: Explicit Thread-Parallelism	184
7.3.3	Challenging Parallelism in a Single Multicore CPU	185
7.3.4	Simpler Use of Multicore: Libraries and Servers	186
7.4	Million-Way Parallelism: Supercomputers and the Cloud	187
7.5	Efficient Parallelism: Computation Grain Size	189
7.6	Programming Cloud Computers: Coarse-Grained Parallelism	192
7.6.1	Three-Tier Web: Scalable Web Services	192

	7.6.2	Scale-Out Map–Reduce (Hadoop and Spark)	193
	7.6.3	Microservices: Modular Reliability and Evolution	194
	7.6.4	Serverless (Function-as-a-Service)	195
	7.7	Summary	196
	7.8	Digging Deeper	198
	7.9	Problems	198
8		Accelerators: Customized Architectures for Performance	205
	8.1	The Emergence of Accelerators	205
	8.1.1	Accelerator Hardware Opportunities	206
	8.1.2	Programming and Software Challenges	206
	8.2	Parallelism Accelerators	208
	8.2.1	The Architecture of GPUs	208
	8.2.2	Diverse GPUs and Performance	210
	8.3	Machine Learning Accelerators	212
	8.3.1	Google’s Tensor Processing Unit	213
	8.3.2	Cerebras CS-2: A Wafer-Scale Machine Learning Accelerator	215
	8.3.3	Small Machine Learning Accelerators (Edge)	216
	8.4	Other Opportunities for Acceleration	217
	8.5	Limitations and Drawbacks of Accelerated Computing	217
	8.6	Summary	219
	8.7	Digging Deeper	219
	8.8	Problems	220
9		Computing Performance: Past, Present, and Future	226
	9.1	Historical Computer Performance	226
	9.2	Future Computer Performance: Opportunities for Performance Increase	228
	9.2.1	Hardware Scaling and Opportunities	228
	9.2.2	Resulting Programming and Software Challenges	230
	9.3	New Computing Models	231
	9.3.1	Higher-Level Architecture	232
	9.3.2	Quantum Computing	233
	9.3.3	Neuromorphic Computing	233
	9.4	Summary	234
	9.5	Digging Deeper	235
	9.6	Problems	235
Appendix		RISC-V Instruction Set Reference Card	239
		<i>References</i>	241
		<i>Index</i>	247