# Contents

Preface xi

## Chapters

### 1. Computer Abstractions and Technology 2

1.1 Introduction 3  
1.2 Seven Great Ideas in Computer Architecture 10  
1.3 Below Your Program 13  
1.4 Under the Covers 16  
1.5 Technologies for Building Processors and Memory 25  
1.6 Performance 29  
1.7 The Power Wall 40  
1.8 The Sea Change: The Switch from Uniprocessors to Multiprocessors 43  
1.9 Real Stuff: Benchmarking the Intel Core i7 46  
1.10 Going Faster: Matrix Multiply in Python 49  
1.11 Fallacies and Pitfalls 50  
1.12 Concluding Remarks 53  
1.13 Historical Perspective and Further Reading 55  
1.14 Self-Study 55  
1.15 Exercises 59

### 2. Instructions: Language of the Computer 66

2.1 Introduction 68  
2.2 Operations of the Computer Hardware 69  
2.3 Operands of the Computer Hardware 73  
2.4 Signed and Unsigned Numbers 80  
2.5 Representing Instructions in the Computer 87  
2.6 Logical Operations 95  
2.7 Instructions for Making Decisions 98  
2.8 Supporting Procedures in Computer Hardware 104  
2.9 Communicating with People 114  
2.10 RISC-V Addressing for Wide Immediates and Addresses 120  
2.11 Parallelism and Instructions: Synchronization 128  
2.12 Translating and Starting a Program 131  
2.13 A C Sort Example to Put it All Together 140
5 Large and Fast: Exploiting Memory Hierarchy 386

5.1 Introduction 388
5.2 Memory Technologies 392
5.3 The Basics of Caches 398
5.4 Measuring and Improving Cache Performance 412
5.5 Dependable Memory Hierarchy 431
5.6 Virtual Machines 436
5.7 Virtual Memory 440
5.8 A Common Framework for Memory Hierarchy 464
5.9 Using a Finite-State Machine to Control a Simple Cache 470
5.10 Parallelism and Memory Hierarchy: Cache Coherence 475
5.11 Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks 479
5.12 Advanced Material: Implementing Cache Controllers 480
5.13 Real Stuff: The ARM Cortex-A8 and Intel Core i7 Memory Hierarchies 480
5.14 Real Stuff: The Rest of the RISC-V System and Special Instructions 486
5.15 Going Faster: Cache Blocking and Matrix Multiply 488
5.16 Fallacies and Pitfalls 489
5.17 Concluding Remarks 494
5.18 Historical Perspective and Further Reading 495
5.19 Self-Study 495
5.20 Exercises 499

6 Parallel Processors from Client to Cloud 518

6.1 Introduction 520
6.2 The Difficulty of Creating Parallel Processing Programs 522
6.3 SISD, MIMD, SIMD, SPMD, and Vector 527
6.4 Hardware Multithreading 534
6.5 Multicore and Other Shared Memory Multiprocessors 537
6.6 Introduction to Graphics Processing Units 542
6.7 Domain-Specific Architectures 549
6.8 Clusters, Warehouse Scale Computers, and Other Message-Passing Multiprocessors 552
6.9 Introduction to Multiprocessor Network Topologies 557
6.10 Communicating to the Outside World: Cluster Networking 561
6.11 Multiprocessor Benchmarks and Performance Models 561
6.12 Real Stuff: Benchmarking the Google TPUv3 Supercomputer and an NVIDIA Volta GPU Cluster 572
6.13 Going Faster: Multiple Processors and Matrix Multiply 580
6.14 Fallacies and Pitfalls 583
6.15 Concluding Remarks 585
6.16 Historical Perspective and Further Reading 587
6.17 Self-Study 588
6.18 Exercises 590

A P P E N D I X

The Basics of Logic Design A-2

A.1 Introduction A-3
A.2 Gates, Truth Tables, and Logic Equations A-4
A.3 Combinational Logic A-9
A.4 Using a Hardware Description Language A-20
A.5 Constructing a Basic Arithmetic Logic Unit A-26
A.6 Faster Addition: Carry Lookahead A-37
A.7 Clocks A-47
A.8 Memory Elements: Flip-Flops, Latches, and Registers A-49
A.9 Memory Elements: SRAMs and DRAMs A-57
A.10 Finite-State Machines A-66
A.11 Timing Methodologies A-71
A.12 Field Programmable Devices A-77
A.13 Concluding Remarks A-78
A.14 Exercises A-79

Index I-1

O N L I N E  C O N T E N T

Graphics and Computing GPUs B-2

B.1 Introduction B-3
B.2 GPU System Architectures B-7
B.3 Programming GPUs B-12
B.4 Multithreaded Multiprocessor Architecture B-25
B.5 Parallel Memory System B-36
B.6 Floating-point Arithmetic B-41
B.7 Real Stuff: The NVIDIA GeForce 8800  B-46
B.8 Real Stuff: Mapping Applications to GPUs  B-55
B.9 Fallacies and Pitfalls  B-72
B.10 Concluding Remarks  B-76
B.11 Historical Perspective and Further Reading  B-77

Mapping Control to Hardware  C-2

C.1 Introduction  C-3
C.2 Implementing Combinational Control Units  C-4
C.3 Implementing Finite-State Machine Control  C-8
C.4 Implementing the Next-State Function with a Sequencer  C-22
C.5 Translating a Microprogram to Hardware  C-28
C.6 Concluding Remarks  C-32
C.7 Exercises  C-33

Survey of Instruction Set Architectures  D-2

D.1 Introduction  D-3
D.2 A Survey of RISC Architectures for Desktop, Server, and Embedded Computers  D-4
D.3 The Intel 80×86  D-30
D.4 The VAX Architecture  D-50
D.5 The IBM 360/370 Architecture for Mainframe Computers  D-68
D.6 Historical Perspective and References  D-74

Glossary  G-1
Further Reading  FR-1