# Table of contents provided by Syndetics

- **Preface** (p. xv)
- **Chapter 1 Computer Abstractions and Technology** (p. 2)
- **1.1 Introduction** (p. 3)
- **1.2 Seven Great Ideas in Computer Architecture** (p. 10)
- **1.3 Below Your Program** (p. 13)
- **1.4 Under the Covers** (p. 16)
- **1.5 Technologies for Building Processors and Memory** (p. 24)
- **1.6 Performance** (p. 28)
- **17 The Power Wall** (p. 40)
- **1.8 The Sea Change: The Switch from Uniprocessors to Multiprocessors** (p. 43)
- **1.9 Real Stuff: Benchmarking the Intel Core i7** (p. 46)
- **1.10 Going Faster: Matrix Multiply in Python** (p. 49)
- **1.11 Fallacies and Pitfalls** (p. 50)
- **1.12 Concluding Remarks** (p. 53)
- **1.13 Historical Perspective and Further Reading** (p. 55)
- **1.14 Self-Study** (p. 55)
- **1.15 Exercises** (p. 59)
- **Chapter 2 Instructions: Language of the Computer** (p. 66)
- **2.1 Introduction** (p. 68)
- **2.2 Operations of the Computer Hardware** (p. 69)
- **2.3 Operands of the Computer Hardware** (p. 72)
- **2.4 Signed and Unsigned Numbers** (p. 79)
- **2.5 Representing Instructions in the Computer** (p. 86)
- **2.6 Logical Operations** (p. 93)
- **2.7 Instructions for Making Decisions** (p. 96)
- **2.8 Supporting Procedures in Computer Hardware** (p. 102)
- **2.9 Communicating with People** (p. 112)
- **2.10 MIPS Addressing for 32-Bit Immediates and Addresses** (p. 118)
- **2.11 Parallelism and Instructions: Synchronization** (p. 127)
- **2.12 Translating and Starting a Program** (p. 129)
- **2.13 A C Sort Example to Put It All Together** (p. 138)
- **2.14 Arrays versus Pointers** (p. 147)
- **2.15 Advanced Material: Compiling C and Interpreting Java** (p. 151)
- **2.16 Real Stuff: ARMv7 (32-bit) Instructions** (p. 151)
- **2.17 Real Stuff: ARMv8 (64-bit) Instructions** (p. 155)
- **2.18 Real Stuff: RISC-V Instructions** (p. 156)
- **2.19 Real Stuff: x86 Instructions** (p. 157)
- **2.20 Going Faster: Matrix Multiply in C** (p. 166)
- **2.21 Fallacies and Pitfalls** (p. 167)
- **2.22 Concluding Remarks** (p. 169)
- **2.23 Historical Perspective and Further Reading** (p. 172)
- **2.24 Self Study** (p. 172)
- **2.25 Exercises** (p. 175)
- **Chapter 3 Arithmetic for Computers** (p. 186)

- **3.1 Introduction** (p. 188)
- **3.2 Addition and Subtraction** (p. 188)
- **3.3 Multiplication** (p. 193)
- **3.4 Division** (p. 199)
- **3.5 Floating Point** (p. 206)
- **3.6 Parallelism and Computer Arithmetic: Subword Parallelism** (p. 232)
- **3.7 Real Stuff: Streaming SIMD Extensions and Advanced Vector Extensions in x86** (p. 234)
- **3.8 Going Faster: Subword Parallelism and Matrix Multiply** (p. 235)
- **3.9 Fallacies and Pitfalls** (p. 237)
- **3.10 Concluding Remarks** (p. 241)
- **3.11 Historical Perspective and Further Reading** (p. 245)
- **3.12 Self Study** (p. 245)
- **3.13 Exercises** (p. 248)
- **Chapter 4 The Processor** (p. 254)
- **4.1 Introduction** (p. 256)
- **4.2 Logic Design Conventions** (p. 260)
- **4.3 Building a Datapath** (p. 263)
- **4.4 A Simple Implementation Scheme** (p. 271)
- **4.5 A Multicycle Implementation** (p. 284)
- **4.6 An Overview of Pipelining** (p. 285)
- **4.7 Pipelined Datapath and Control** (p. 298)
- **4.8 Data Hazards: Forwarding versus Stalling** (p. 315)
- **4.9 Control Hazards** (p. 328)
- **4.10 Exceptions** (p. 337)
- **4.11 Parallelism via Instructions** (p. 344)
- **4.12 Putting It All Together: The Intel Core i7 6700 and ARM Cortex-A53** (p. 358)
- **4.13 Going Faster: Instruction-Level Parallelism and Matrix Multiply** (p. 366)
- **4.14 Advanced Topic: An Introduction to Digital Design Using a Hardware Design Language to Describe and Model a Pipeline and More Pipelining Illustrations** (p. 368)
- **4.15 Fallacies and Pitfalls** (p. 369)
- **4.16 Concluding Remarks** (p. 370)
- **4.17 Historical Perspective and Further Reading** (p. 371)
- **4.18 Self-Study** (p. 371)
- **4.19 Exercises** (p. 372)
- **Chapter 5 Large and Fast: Exploiting Memory Hierarchy** (p. 390)
- **5.1 Introduction** (p. 392)
- **5.2 Memory Technologies** (p. 396)
- **5.3 The Basics of Caches** (p. 401)
- **5.4 Measuring and Improving Cache Performance** (p. 416)
- **5.5 Dependable Memory Hierarchy** (p. 436)
- **5.6 Virtual Machines** (p. 442)
- **5.7 Virtual Memory** (p. 446)
- **5.8 A Common Framework for Memory Hierarchy** (p. 472)
- **5.9 Using a Finite-State Machine to Control a Simple Cache** (p. 479)

- **5.10 Parallelism and Memory Hierarchies: Cache Coherence** (p. 484)
- **5.11 Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks** (p. 488)
- **5.12 Advanced Material: Implementing Cache Controllers** (p. 488)
- **5.13 Real Stuff: The ARM Cortex-A8 and Intel Core i7 Memory Hierarchies** (p. 489)
- **5.14 Going Faster: Cache Blocking and Matrix Multiply** (p. 494)
- **5.15 Fallacies and Pitfalls** (p. 496)
- **5.16 Concluding Remarks** (p. 500)
- **5.17 Historical Perspective and Further Reading** (p. 501)
- **5.18 Self-Study** (p. 501)
- **5.19 Exercises** (p. 506)
- **Chapter 6 Parallel Processors from Client to Cloud** (p. 524)
- **6.1 Introduction** (p. 526)
- **6.2 The Difficulty of Creating Parallel Processing Programs** (p. 528)
- **6.3 SISD, MIMD, SIMD, SPMD, and Vector** (p. 533)
- **6.4 Hardware Multithreading** (p. 540)
- **6.5 Multicore and Other Shared Memory Multiprocessors** (p. 543)
- **6.6 Introduction to Graphics Processing Units** (p. 548)
- **6.7 Domain Specific Architectures** (p. 555)
- **6.8 Clusters, Warehouse Scale Computers, and Other Message-Passing Multiprocessors** (p. 558)
- **6.9 Introduction to Multiprocessor Network Topologies** (p. 563)
- **6.10 Communicating to the Outside World: Cluster Networking** (p. 566)
- **6.11 Multiprocessor Benchmarks and Performance Models** (p. 567)
- **6.12 Real Stuff: Benchmarking the Google TPUv3 Supercomputer and an NVIDIA Volta GPU Cluster** (p. 577)
- **6.13 Going Faster: Multiple Processors and Matrix Multiply** (p. 586)
- **6.14 Fallacies and Pitfalls** (p. 589)
- **6.15 Concluding Remarks** (p. 592)
- **6.16 Historical Perspective and Further Reading** (p. 594)
- **6.17 Self Study** (p. 594)
- **6.18 Exercises** (p. 596)
- **Appendices**
- **A Assemblers, Linkers, and the SPIM Simulator** (p. A-610)
- **A.1 Introduction** (p. A-611)
- **A.2 Assemblers** (p. A-618)
- **A.3 Linkers** (p. A-626)
- **A.4 Loading** (p. A-627)
- **A.5 Memory Usage** (p. A-628)
- **A.6 Procedure Call Convention** (p. A-630)
- **A.7 Exceptions and Interrupts** (p. A-641)
- **A.8 Input and Output** (p. A-646)
- **A.9 SPIM** (p. A-648)
- **A.10 MIPS R2000 Assembly Language** (p. A-653)
- **A.11 Concluding Remarks** (p. A-689)

- **A.12 Exercises** (p. A-690)
- **B The Basics of Logic Design** (p. B-692)
- **B.1 Introduction** (p. B-693)
- **B.2 Gates, Truth Tables, and Logic Equations** (p. B-694)
- **B.3 Combinational Logic** (p. B-699)
- **B.4 Using a Hardware Description Language** (p. B-710)
- **B.5 Constructing a Basic Arithmetic Logic Unit** (p. B-716)
- **B.6 Faster Addition: Carry Lookahead** (p. B-728)
- **B.7 Clocks** (p. B-738)
- **B.8 Memory Elements: Flip-Flops, Latches, and Registers** (p. B-740)
- **B.9 Memory Elements: SRAMs and DRAMs** (p. B-748)
- **B.10 Finite-State Machines** (p. B-757)
- **B.11 Timing Methodologies** (p. B-762)
- **B.12 Field Programmable Devices** (p. B-768)
- **B.13 Concluding Remarks** (p. B-769)
- **B.14 Exercises** (p. B-770)
- **Index** (p. I-1)
- **Online Content**
- **C Graphics and Computing GPUs** (p. C-2)
- **C.1 Introduction** (p. C-3)
- **C.2 GPU System Architectures** (p. C-7)
- **C.3 Programming GPUs** (p. C-12)
- **C.4 Multithreaded Multiprocessor Architecture** (p. C-25)
- **C.5 Parallel Memory System** (p. C-36)
- **C.6 Floating Point Arithmetic** (p. C-41)
- **C.7 Real Stuff: The NVTDIA GeForce 8800** (p. C-46)
- **C.8 Real Stuff: Mapping Applications to GPUs** (p. C-55)
- **C.9 Fallacies and Pitfalls** (p. C-72)
- **C.10 Concluding Remarks** (p. C-76)
- **C.11 Historical Perspective and Further Reading** (p. C-77)
- **D Mapping Control to Hardware** (p. D-2)
- **D.1 Introduction** (p. D-3)
- **D.2 Implementing Combinational Control Units** (p. D-4)
- **D.3 Implementing Finite-State Machine Control** (p. D-8)
- **D.4 Implementing the Next-State Function with a Sequencer** (p. D-22)
- **D.5 Translating a Microprogram to Hardware** (p. D-28)
- **D.6 Concluding Remarks** (p. D-32)
- **D.7 Exercises** (p. D-33)
- **E Survey of Instruction Set Architectures**
- **E.1 Introduction** (p. E-3)
- **E.2 A Survey of RISC Architecture for Desktop, Server, and Embedded Computers** (p. E-4)
- **E.3 The Intel 80×86** (p. E-30)
- **E.4 The VAX Architecture** (p. E-50)
- **E.5 The IBM 360/370 Architecture for Mainframe Computers** (p. E-69)
- **E.6 Historical Perspective and References** (p. E-73)

- **Glossary** (p. G-1)
- **Further Reading** (p. FR-1)