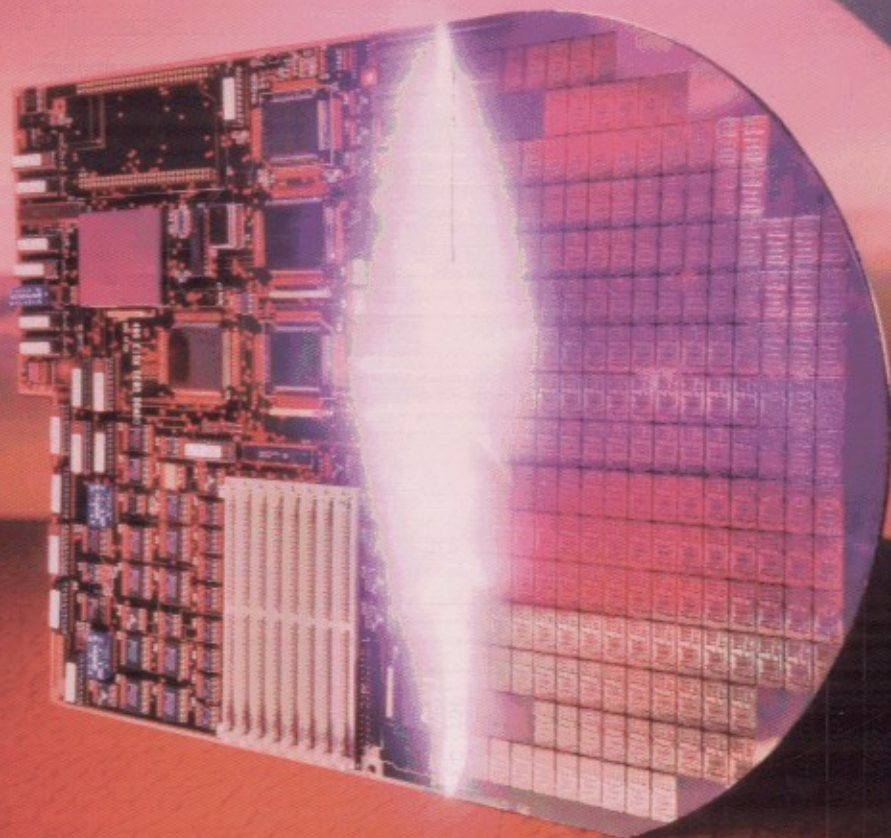


COMPUTER ARCHITECTURE AND ORGANIZATION

*An Integrated
Approach*



MILES MURDOCCA
VINCENT HEURING

CONTENTS

PREFACE ix

CHAPTER 1 INTRODUCTION 1

- 1.1 A Brief History of Computing 1
- 1.2 The Von Neumann Model 9
- 1.3 The System Bus Model 10
- 1.4 Levels of Machines 11
 - 1.4.1 Upward Compatibility 12
 - 1.4.2 The Levels 12
- 1.5 A Typical Computer System 16
- 1.6 Role of the Network 18
- 1.7 Organization of the Book 20
- 1.8 Case Study: What Happened to Supercomputers? 20

CHAPTER 2 DATA REPRESENTATION 27

- 2.1 Fixed-Point Numbers 28
 - 2.1.1 Range and Precision in Fixed-Point Numbers 28
 - 2.1.2 The Associative Law of Algebra Does Not Always Hold in Computers 29
 - 2.1.3 Radix Number Systems 29
 - 2.1.4 Conversions Among Radices 30
 - 2.1.5 An Early Look at Computer Arithmetic 35
 - 2.1.6 Signed Fixed-Point Numbers 36
- 2.2 Floating-Point Numbers 40
 - 2.2.1 Range and Precision in Floating-Point Numbers 40
 - 2.2.2 Normalization and the Hidden Bit 41
 - 2.2.3 Representing Floating-Point Numbers in the Computer—Preliminaries 42
 - 2.2.4 Error in Floating-Point Representations 44
 - 2.2.5 The IEEE 754 Floating-Point Standard 47
- 2.3 Case Study: Patriot Missile Defense Failure Caused by Loss of Precision 51
- 2.4 Character Codes 52
 - 2.4.1 The ASCII Character Set 53
 - 2.4.2 The EBCDIC Character Set 54
 - 2.4.3 The Unicode Character Set 56

CHAPTER 3 ARITHMETIC 61

- 3.1 Fixed-Point Addition and Subtraction 61
 - 3.1.1 Two's Complement Addition and Subtraction 61
 - 3.1.2 Sign Extension 64
 - 3.1.3 Hardware Implementation of Adders and Subtractors 65
 - 3.1.4 One's Complement Addition and Subtraction 67
- 3.2 Fixed-Point Multiplication and Division 68
 - 3.2.1 Unsigned Multiplication 69
 - 3.2.2 Unsigned Division 70
 - 3.2.3 Signed Multiplication and Division 73
- 3.3 Floating-Point Arithmetic 73
 - 3.3.1 Floating-Point Addition and Subtraction 73
 - 3.3.2 Floating-Point Multiplication and Division 76
- 3.4 High-Performance Arithmetic 77
 - 3.4.1 High-Performance Addition 77
 - 3.4.2 High-Performance Multiplication 81
 - 3.4.3 High-Performance Division 84
 - 3.4.4 Residue Arithmetic 86

CHAPTER 4 THE INSTRUCTION SET ARCHITECTURE 93

- 4.1 Hardware Components of the Instruction Set Architecture 93
 - 4.1.1 The System Bus Model Revisited 94
 - 4.1.2 Memory 94
 - 4.1.3 The CPU 98
- 4.2 ARC, A RISC Computer 102
 - 4.2.1 ARC Memory 103
 - 4.2.2 ARC Registers 103
 - 4.2.3 ARC Assembly Language Format 105
 - 4.2.4 The ARC Instruction Set 109
 - 4.2.5 ARC Instruction Formats 112
 - 4.2.6 SPARC and ARC Data Formats 114
 - 4.2.7 ARC Instruction Descriptions 114
- 4.3 Pseudo-Operations 123
- 4.4 Synthetic Instructions 125
- 4.5 Examples of Assembly Language Programs 125

4.5.1	Variations in Machine Architectures and Addressing	128	6.3	Linking and Loading	213
4.5.2	Performance of Instruction Set Architectures	131	6.3.1	Linking	213
4.6	Accessing Data in Memory—Addressing Modes	131	6.3.2	Resolving External References	214
4.7	Subroutine Linkage and Stacks	132	6.3.3	Loading	216
4.8	Input and Output in Assembly Language	137	6.4	Macros	217
4.9	Case Study: The Java Virtual Machine ISA	140	6.5	Quantitative Analyses of Program Execution	224
CHAPTER 5 DATAPATH AND CONTROL 151			6.5.1	Quantitative Performance Analysis	226
5.1	Basics of the Microarchitecture	151	6.6	From CISC to RISC	227
5.2	The Datapath	152	6.7	Pipelining the Datapath	229
5.2.1	Datapath Overview	153	6.7.1	Arithmetic, Branch, and Load-Store Instructions	229
5.3	The Control Section—Microprogrammed	160	6.7.2	Pipelining Instructions	231
5.3.1	Timing	164	6.7.3	Keeping the Pipeline Filled	231
5.3.2	Developing the Microprogram	164	6.8	Overlapping Register Windows	234
5.3.3	Traps and Interrupts	172	6.9	Low-Power Coding	241
5.3.4	Nanoprogramming	175	CHAPTER 7 MEMORY 249		
5.4	The Control Section—Hardwired	176	7.1	The Memory Hierarchy	249
5.5	Case Study: The VHDL Hardware Description Language	183	7.2	Random-Access Memory	250
5.5.1	Background	183	7.3	Memory Chip Organization	252
5.5.2	What is VHDL?	184	7.3.1	Constructing Large RAMs From Small RAMs	255
5.5.3	A VHDL Specification of the Majority Function	185	7.3.2	Commercial Memory Modules	256
5.5.4	Nine-Value Logic System	188	7.3.3	Read-Only Memory	257
5.6	Case Study: What Happens When a Computer Boots Up?	188	7.3.4	Flash Memory	260
CHAPTER 6 LANGUAGES AND THE MACHINE 197			7.4	Case Study: Rambus Memory	262
6.1	The Compilation Process	197	7.5	Cache Memory	266
6.1.1	Steps in Compilation	197	7.5.1	Associative Mapped Cache	267
6.1.2	The Compiler Mapping Specification	198	7.5.2	Direct-Mapped Cache	270
6.1.3	How the Compiler Maps the Three Instruction Classes into Assembly Code	199	7.5.3	Set-Associative Mapped Cache	273
6.1.4	Data Movement	200	7.5.4	Cache Performance	274
6.1.5	Arithmetic Instructions	203	7.5.5	Hit Ratios and Effective Access Times	276
6.1.6	Program Control Flow	203	7.5.6	Multilevel Caches	277
6.2	The Assembly Process	205	7.5.7	Cache Management	279
6.2.1	Assembly and Two-Pass Assemblers	206	7.5.8	Cache Coherency	280
6.2.2	Assembly and the Symbol Table	209	7.6	Virtual Memory	281
6.2.3	Final Tasks of the Assembler	211	7.6.1	Overlays	281
6.2.4	Programs for Embedded vs. Virtual Memory Systems	211	7.6.2	Paging	282
			7.6.3	Segmentation	286
			7.6.4	Protection	288
			7.6.5	Fragmentation	288
			7.6.6	The Translation Lookaside Buffer	290
			7.6.7	Putting It All Together	291
			7.7	Advanced Topics	292
			7.7.1	Content-Addressable (Associative) Memories	292
			7.8	Case Study: Associative Memory in Routers	295

- 7.9 Case Study: The Intel Pentium 4 Memory System 297

CHAPTER 8 BUSES AND PERIPHERALS 303

- 8.1 Parallel Bus Architectures 304
 - 8.1.1 Bus Structure, Protocol, and Control 304
 - 8.1.2 Bus Clocking 305
 - 8.1.3 The Synchronous Bus 306
 - 8.1.4 The Asynchronous Bus 307
 - 8.1.5 Bus Arbitration—Masters and Slaves 308
- 8.2 Bridge-Based Bus Architectures 310
- 8.3 Internal Communication Methodologies 311
 - 8.3.1 Programmed I/O 312
 - 8.3.2 Interrupt-Driven I/O 313
 - 8.3.3 Direct Memory Access (DMA) 314
- 8.4 Case Study: Communication on the Intel Pentium Architecture 316
 - 8.4.1 System Clock, Bus Clock, and Bus Speeds 316
 - 8.4.2 Address, Data, Memory, and I/O Capabilities 316
 - 8.4.3 Data Words Have Soft Alignment 317
 - 8.4.4 Bus Cycles in the Pentium family 317
 - 8.4.5 Memory Read and Write Bus Cycles 318
 - 8.4.6 The Burst Read Bus Cycle 319
 - 8.4.7 Bus Hold for Request by Bus Master 320
 - 8.4.8 Data Transfer Rates 321
- 8.5 Serial Bus Architectures 321
 - 8.5.1 RS-232 322
 - 8.5.2 Universal Serial Bus (USB) 322
 - 8.5.3 Firewire 323
- 8.6 Mass Storage 323
 - 8.6.1 Magnetic Disks 323
 - 8.6.2 Magnetic Tape 330
 - 8.6.3 Optical Disks 332
- 8.7 RAID—Redundant Arrays of Inexpensive Disks 334
- 8.8 Input Devices 339
 - 8.8.1 Keyboards 339
 - 8.8.2 Tablets 339
 - 8.8.3 Mice and Trackballs 340
 - 8.8.4 Touch-Sensitive Pen-Based Display 341
 - 8.8.5 Joysticks 342
- 8.9 Output Devices 342
 - 8.9.1 Laser Printers 342

- 8.9.2 Video Displays 343
- 8.9.3 Liquid Crystal Displays (LCDs) 345

- 8.10 Case Study: Graphics Processing Unit 346
- 8.11 Case Study: How a Virus Infects a Machine 347

CHAPTER 9 NETWORKING AND COMMUNICATION 353

- 9.1 A Few Modulation Schemes 353
- 9.2 Transmission Media 355
 - 9.2.1 Two-Wire Open Lines 356
 - 9.2.2 Twisted-Pair Lines 356
 - 9.2.3 Coaxial Cable 357
 - 9.2.4 Optical Fiber 357
 - 9.2.5 Satellites 358
 - 9.2.6 Terrestrial Microwave 358
 - 9.2.7 Radio 359
- 9.3 Error Detection and Correction 359
 - 9.3.1 Bit Error Rate Defined 360
 - 9.3.2 Hamming Codes 360
 - 9.3.3 Vertical Redundancy Checking 365
 - 9.3.4 Cyclic Redundancy Checking 366
- 9.4 Networking and Network Device Architectures 368
 - 9.4.1 The OSI Model 368
 - 9.4.2 Topologies 370
 - 9.4.3 Ethernet 371
 - 9.4.4 Hubs, Bridges, Switches, Routers, and Gateways 376
- 9.5 Storage Area Networks 378
- 9.6 Case Study: Cisco Router Architecture 379

CHAPTER 10 ADVANCED COMPUTER ARCHITECTURE 385

- 10.1 Parallel Architecture 385
 - 10.1.1 Measuring Performance 386
 - 10.1.2 The Flynn Taxonomy 387
 - 10.1.3 Interconnection Networks 388
 - 10.1.4 Mapping an Algorithm Onto a Parallel Architecture 393
- 10.2 Superscalar Machines and the PowerPC 398
 - 10.2.1 Instruction Set Architecture of the PowerPC 399
 - 10.2.2 Hardware Architecture of the PowerPC 399
- 10.3 VLIW Machines, and the Itanium 401
 - 10.3.1 Case Study: The Intel IA-64 (Itanium) Architecture 402
 - 10.3.2 Background—the 80x86 CISC Architecture 402

10.3.3	The Itanium: an Epic Architecture	402
10.4	Case Study: Extensions to the Instruction Set—The Intel MMX/SSE and Motorola AltiVec SIMD Instructions	406
10.4.1	Background	407
10.4.2	The Base Architectures	407
10.4.3	VECTOR Registers	408
10.4.4	Vector Arithmetic Operations	410
10.4.5	Vector Compare Operations	411
10.4.6	Case Study Summary	412
10.5	Programmable Logic Devices and Custom ICs	413
10.5.1	The Role of CAD Tools in PLD Design	414
10.5.2	PLAS and PALS	414
10.5.3	Complex Programmable Logic Devices	415
10.5.4	Field-Programmable Gate Arrays	415
10.5.5	Application-Specific Integrated Circuits	416
10.6	Unconventional Architectures	417
10.6.1	DNA Computing	417
10.6.2	Quantum Computing	419
10.6.3	Multi-valued Logic	420
10.6.4	Neural Networks	422

APPENDIX A: DIGITAL LOGIC 429

A.1	Introduction	429
A.2	Combinational Logic	429
A.3	Truth Tables	430
A.4	Logic Gates	431
A.4.1	Electronic Implementation of Logic Gates	434
A.4.2	Tri-State Buffers	437
A.5	Properties of Boolean Algebra	438
A.6	The Sum-of-Products Form and Logic Diagrams	440
A.7	The Product-of-Sums Form	442
A.8	Positive vs. Negative Logic	443
A.9	The Data Sheet	445
A.10	Digital Components	447
A.10.1	Levels of Integration	447
A.10.2	Multiplexers	447
A.10.3	Demultiplexers	449
A.10.4	Decoders	450

A.10.5	Priority Encoders	452
A.10.6	Programmable Logic Arrays	453
A.11	Sequential Logic	457
A.11.1	The S-R Flip-Flop	457
A.11.2	The Clocked S-R Flip-Flop	459
A.11.3	The D Flip-Flop and the Master-Slave Configuration	460
A.11.4	J-K and T Flip-Flops	462
A.12	Design of Finite State Machines	464
A.13	Mealy vs. Moore Machines	471
A.14	Registers	472
A.15	Counters	474
A.16	Reduction of Combinational Logic and Sequential Logic	474
A.17	Reduction of Two-Level Expressions	475
A.17.1	The Algebraic Method	475
A.17.2	The K-Map Method	476
A.17.3	The Tabular Method	481
A.17.4	Logic Reduction: Effect on Speed and Performance	488
A.18	State Reduction	492

APPENDIX B: USING ARCTools 505

B.1	Introduction	505
B.2	Accessing and Launching ARCTools	505
B.2.1	Launching ARCTools	506
B.3	The ARC Assembler	506
B.3.1	Loading, Assembling, and Examining a File	508
B.3.2	Saving Files	510
B.3.3	Loading Files into the Simulator	510
B.4	The ARC Simulator	510
B.5	Instructions and Pseudo Instructions Recognized by ARCTools	512
B.5.1	Instructions-Actual, Synthetic, and Pseudo	512
B.6	The Macroprocessor	514
B.7	Measuring Program Performance	514
B.7.1	The TimeModel configuration Editor	515
B.7.2	Memory / IO Parameters	516
B.7.3	TimeModel's Statistics Window	516
B.7.4	The Cache Simulator View	517

INDEX 519