



TABLE OF CONTENTS

PREFACE	iii
1 INTRODUCTION	1
1.1 OVERVIEW	1
1.2 A BRIEF HISTORY	1
1.3 THE VON NEUMANN MODEL	4
1.4 THE SYSTEM BUS MODEL	5
1.5 LEVELS OF MACHINES	7
1.5.1 Upward Compatibility	7
1.5.2 The Levels	7
1.6 A TYPICAL COMPUTER SYSTEM	12
1.7 ORGANIZATION OF THE BOOK	13
1.8 CASE STUDY: WHAT HAPPENED TO SUPERCOMPUTERS?	14
2 DATA REPRESENTATION	21
2.1 INTRODUCTION	21
2.2 FIXED POINT NUMBERS	22
2.2.1 Range and Precision in Fixed Point Numbers	22
2.2.2 The Associative Law of Algebra Does Not Always Hold in Computers	23
2.2.3 Radix Number Systems	24
2.2.4 Conversions Among Radices	25
2.2.5 An Early Look at Computer Arithmetic	31
2.2.6 Signed Fixed Point Numbers	32
2.2.7 Binary Coded Decimal	37
2.3 FLOATING POINT NUMBERS	38
2.3.1 Range and Precision In Floating Point Numbers	38
2.3.2 Normalization, and The Hidden Bit	40

2.3.3	Representing Floating Point Numbers in the Computer—Preliminaries	40
2.3.4	Error in Floating Point Representations	44
2.3.5	The IEEE 754 Floating Point Standard	48
2.4	CASE STUDY: PATRIOT MISSILE DEFENSE FAILURE CAUSED BY LOSS OF PRECISION	51
2.5	CHARACTER CODES	53
2.5.1	The ASCII Character Set	53
2.5.2	The EBCDIC Character Set	54
2.5.3	The Unicode Character Set	55
3	ARITHMETIC	65
3.1	OVERVIEW	65
3.2	FIXED POINT ADDITION AND SUBTRACTION	65
3.2.1	Two's complement addition and subtraction	66
3.2.2	Hardware implementation of adders and subtractors	69
3.2.3	One's Complement Addition and Subtraction	71
3.3	FIXED POINT MULTIPLICATION AND DIVISION	73
3.3.1	Unsigned Multiplication	73
3.3.2	Unsigned Division	75
3.3.3	Signed Multiplication and Division	77
3.4	FLOATING POINT ARITHMETIC	79
3.4.1	Floating Point Addition and Subtraction	79
3.4.2	Floating Point Multiplication and Division	80
3.5	HIGH PERFORMANCE ARITHMETIC	81
3.5.1	High Performance Addition	81
3.5.2	High Performance Multiplication	83
3.5.3	High Performance Division	87
3.5.4	Residue Arithmetic	90
3.6	CASE STUDY: CALCULATOR ARITHMETIC USING BINARY CODED DECIMAL	93
3.6.1	The HP9100A Calculator	94
3.6.2	Binary Coded Decimal Addition and subtraction	94
3.6.3	BCD Floating Point Addition and Subtraction	97
4	THE INSTRUCTION SET ARCHITECTURE	105
4.1	HARDWARE COMPONENTS OF THE INSTRUCTION SET ARCHITECTURE	106
4.1.1	The System Bus Model Revisited	106
4.1.2	Memory	107
4.1.3	The CPU	110
4.2	ARC, A RISC COMPUTER	114

4.2.1	ARC Memory	115
4.2.2	ARC Instruction set	116
4.2.3	ARC Assembly Language Format	118
4.2.4	ARC Instruction Formats	120
4.2.5	ARC Data Formats	122
4.2.6	ARC Instruction Descriptions	123
4.3	PSEUDO-OPS	127
4.4	EXAMPLES OF ASSEMBLY LANGUAGE PROGRAMS	128
4.4.1	Variations in machine architectures and addressing	131
4.4.2	Performance of Instruction Set Architectures	134
4.5	ACCESSING DATA IN MEMORY—ADDRESSING MODES	135
4.6	SUBROUTINE LINKAGE AND STACKS	136
4.7	INPUT AND OUTPUT IN ASSEMBLY LANGUAGE	142
4.8	CASE STUDY: THE JAVA VIRTUAL MACHINE ISA	144
5	LANGUAGES AND THE MACHINE	159
5.1	THE COMPILATION PROCESS	159
5.1.1	The steps of compilation	160
5.1.2	The Compiler Mapping Specification	161
5.1.3	How the compiler maps the three instruction Classes into Assembly Code	161
5.1.4	Data movement	163
5.1.5	Arithmetic instructions	165
5.1.6	program Control flow	166
5.2	THE ASSEMBLY PROCESS	168
5.3	LINKING AND LOADING	176
5.3.1	Linking	177
5.3.2	Loading	180
5.4	MACROS	183
5.5	CASE STUDY: EXTENSIONS TO THE INSTRUCTION SET – THE INTEL MMX™ AND MOTOROLA ALTIVEC™ SIMD INSTRUCTIONS.	185
5.5.1	Background	186
5.5.2	The Base Architectures	186
5.5.3	VECTOR Registers	187
5.5.4	Vector Arithmetic operations	190
5.5.5	Vector compare operations	191
5.5.6	Case Study Summary	193
6	DATAPATH AND CONTROL	199
6.1	BASICS OF THE MICROARCHITECTURE	200

6.2	A MICROARCHITECTURE FOR THE ARC 201	
6.2.1	The Datapath 201	
6.2.2	The Control Section 210	
6.2.3	Timing 213	
6.2.4	Developing the Microprogram 214	
6.2.5	Traps and Interrupts 225	
6.2.6	Nanoprogramming 227	
6.3	HARDWIRED CONTROL 228	
6.4	CASE STUDY: THE VHDL HARDWARE DESCRIPTION LANGUAGE 237	
6.4.1	Background 238	
6.4.2	What is VHDL? 239	
6.4.3	A VHDL specification of the Majority FUNCTION 240	
6.4.4	9-Value logic system 243	
7	MEMORY	255
7.1	THE MEMORY HIERARCHY 255	
7.2	RANDOM ACCESS MEMORY 257	
7.3	CHIP ORGANIZATION 258	
7.3.1	Constructing LARGE RAMS FROM SMALL RAMS 261	
7.4	COMMERCIAL MEMORY MODULES 262	
7.5	READ-ONLY MEMORY 263	
7.6	CACHE MEMORY 266	
7.6.1	Associative Mapped Cache 268	
7.6.2	Direct Mapped Cache 271	
7.6.3	Set Associative Mapped Cache 274	
7.6.4	Cache performance 275	
7.6.5	Hit Ratios and Effective Access Times 277	
7.6.6	Multilevel Caches 279	
7.6.7	Cache management 279	
7.7	VIRTUAL MEMORY 281	
7.7.1	Overlays 281	
7.7.2	Paging 283	
7.7.3	Segmentation 286	
7.7.4	Fragmentation 287	
7.7.5	Virtual Memory vs. Cache Memory 289	
7.7.6	THE TRANSLATION LOOKASIDE BUFFER 289	
7.8	ADVANCED TOPICS 291	
7.8.1	Tree decoders 291	
7.8.2	Decoders for large RAMs 292	

- 7.8.3 Content-Addressable (Associative) Memories 293
- 7.9 CASE STUDY: RAMBUS MEMORY 298
- 7.10 CASE STUDY: THE INTEL PENTIUM MEMORY SYSTEM 301

8 INPUT AND OUTPUT

311

- 8.1 SIMPLE BUS ARCHITECTURES 312
 - 8.1.1 Bus Structure, Protocol, and Control 313
 - 8.1.2 Bus Clocking 314
 - 8.1.3 The Synchronous Bus 314
 - 8.1.4 The Asynchronous Bus 315
 - 8.1.5 Bus Arbitration—Masters and Slaves 316
- 8.2 BRIDGE-BASED BUS ARCHITECTURES 319
- 8.3 COMMUNICATION METHODOLOGIES 321
 - 8.3.1 Programmed I/O 321
 - 8.3.2 Interrupt-driven I/O 322
 - 8.3.3 Direct Memory Access (DMA) 324
- 8.4 CASE STUDY: COMMUNICATION ON THE INTEL PENTIUM ARCHITECTURE 326
 - 8.4.1 System clock, bus clock, and bus speeds 326
 - 8.4.2 Address, data, memory, and I/O capabilities 327
 - 8.4.3 Data words have soft-alignment 327
 - 8.4.4 Bus cycles in the Pentium family 327
 - 8.4.5 Memory read and write bus cycles 328
 - 8.4.6 The burst Read bus cycle 329
 - 8.4.7 Bus hold for request by bus master 330
 - 8.4.8 Data transfer rates 331
- 8.5 MASS STORAGE 332
 - 8.5.1 Magnetic Disks 332
 - 8.5.2 Magnetic Tape 341
 - 8.5.3 Magnetic Drums 342
 - 8.5.4 Optical Disks 343
- 8.6 INPUT DEVICES 346
 - 8.6.1 Keyboards 346
 - 8.6.2 Bit Pads 347
 - 8.6.3 Mice and Trackballs 348
 - 8.6.4 Lightpens and TouchScreens 349
 - 8.6.5 Joysticks 350
- 8.7 OUTPUT DEVICES 351
 - 8.7.1 Laser Printers 351
 - 8.7.2 Video Displays 352

9	COMMUNICATION	361
9.1	MODEMS 361	
9.2	TRANSMISSION MEDIA 364	
9.2.1	Two-Wire Open Lines 365	
9.2.2	Twisted-Pair Lines 366	
9.2.3	Coaxial Cable 366	
9.2.4	Optical Fiber 366	
9.2.5	Satellites 367	
9.2.6	Terrestrial Microwave 368	
9.2.7	Radio 368	
9.3	NETWORK ARCHITECTURE: LOCAL AREA NETWORKS 368	
9.3.1	The OSI Model 369	
9.3.2	Topologies 371	
9.3.3	Data Transmission 372	
9.3.4	Bridges, Routers, and Gateways 374	
9.4	COMMUNICATION ERRORS AND ERROR CORRECTING CODES 375	
9.4.1	Bit Error Rate Defined 375	
9.4.2	Error Detection and Correction 376	
9.4.3	Vertical Redundancy Checking 382	
9.4.4	Cyclic Redundancy Checking 383	
9.5	NETWORK ARCHITECTURE: THE INTERNET 386	
9.5.1	The Internet Model 386	
9.5.2	Bridges and Routers Revisited, and Switches 392	
9.6	CASE STUDY: ASYNCHRONOUS TRANSFER MODE 393	
9.6.1	Synchronous vs. Asynchronous Transfer Mode 395	
9.6.2	What is ATM? 395	
9.6.3	ATM Network Architecture 396	
9.6.4	Outlook on ATM 398	
10	TRENDS IN COMPUTER ARCHITECTURE	403
10.1	QUANTITATIVE ANALYSES OF PROGRAM EXECUTION 403	
10.1.1	quantitative performance analysis 406	
10.2	FROM CISC TO RISC 407	
10.3	PIPELINING THE DATAPATH 409	
10.3.1	arithmetic, branch, and load-store instructions 409	
10.3.2	Pipelining instructions 411	
10.3.3	Keeping the pipeline Filled 411	
10.4	OVERLAPPING REGISTER WINDOWS 415	
10.5	MULTIPLE INSTRUCTION ISSUE (SUPERSCALAR) MACHINES – THE POWERPC 601	

423	
10.6	CASE STUDY: THE POWERPC™ 601 AS A SUPERSCALAR ARCHITECTURE 425
10.6.1	Instruction Set Architecture of the PowerPC 601 425
10.6.2	Hardware architecture of the PowerPC 601 425
10.7	VLIW MACHINES 428
10.8	CASE STUDY: THE INTEL IA-64 (MERCED) ARCHITECTURE 428
10.8.1	background—the 80x86 Cisc architecture 428
10.8.2	The merced: an epic architecture 429
10.9	PARALLEL ARCHITECTURE 432
10.9.1	The Flynn Taxonomy 434
10.9.2	Interconnection Networks 436
10.9.3	Mapping an Algorithm onto a Parallel Architecture 442
10.9.4	Fine-Grain Parallelism – The Connection Machine CM-1 447
10.9.5	Course-Grain Parallelism: The CM-5 450
10.10	CASE STUDY: PARALLEL PROCESSING IN THE SEGA GENESIS 453
10.10.1	The SEGA Genesis Architecture 453
10.10.2	Sega Genesis Operation 455
10.10.3	Sega Genesis Programming 455

A APPENDIX A: DIGITAL LOGIC

461

A.1	INTRODUCTION 461
A.2	COMBINATIONAL LOGIC 461
A.3	TRUTH TABLES 462
A.4	LOGIC GATES 464
A.4.1	Electronic implementation of logic gates 467
A.4.2	Tri-STATE Buffers 470
A.5	PROPERTIES OF BOOLEAN ALGEBRA 470
A.6	THE SUM-OF-PRODUCTS FORM, AND LOGIC DIAGRAMS 473
A.7	THE PRODUCT-OF-SUMS FORM 475
A.8	POSITIVE VS. NEGATIVE LOGIC 477
A.9	THE DATA SHEET 479
A.10	DIGITAL COMPONENTS 481
A.10.1	Levels of Integration 481
A.10.2	Multiplexers 482
A.10.3	Demultiplexers 484
A.10.4	Decoders 485
A.10.5	Priority Encoders 487
A.10.6	Programmable Logic Arrays 487
A.11	SEQUENTIAL LOGIC 492

A.11.1	The S-R Flip-Flop	493	
A.11.2	The Clocked S-R Flip-Flop	495	
A.11.3	The D Flip-Flop and the Master-Slave Configuration	497	
A.11.4	J-K and T Flip-Flops	499	
A.12	DESIGN OF FINITE STATE MACHINES	500	
A.13	MEALY VS. MOORE MACHINES	509	
A.14	REGISTERS	510	
A.15	COUNTERS	511	
B	APPENDIX B: REDUCTION OF DIGITAL LOGIC		523
B.1	REDUCTION OF COMBINATIONAL LOGIC AND SEQUENTIAL LOGIC	523	
B.2	REDUCTION OF TWO-LEVEL EXPRESSIONS	523	
B.2.1	The Algebraic Method	524	
B.2.2	The K-Map Method	525	
B.2.3	The Tabular Method	534	
B.2.4	Logic reduction: EFFECT ON speed and performance	542	
B.3	STATE REDUCTION	546	
B.3.1	The State Assignment Problem	550	
B.3.2	Excitation Tables	554	
	SOLUTIONS TO PROBLEMS		569
	INDEX		623