



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	36
2.3 FLOATING POINT NUMBERS	37
2.3.1 Range and Precision In Floating Point Numbers	38
2.3.2 Normalization, and The Hidden Bit	39

2.3.3	Representing Floating Point Numbers in the Computer—Preliminaries	40
2.3.4	Error in Floating Point Representations	43
2.3.5	The IEEE 754 Floating Point Standard	47
2.4	CASE STUDY: PATRIOT MISSILE DEFENSE FAILURE CAUSED BY LOSS OF PRECISION	51
2.5	CHARACTER CODES	52
2.5.1	The ASCII Character Set	53
2.5.2	The EBCDIC Character Set	54
2.5.3	The Unicode Character Set	54
3	ARITHMETIC	63
3.1	OVERVIEW	63
3.2	FIXED POINT ADDITION AND SUBTRACTION	63
3.2.1	Two's complement addition and subtraction	64
3.2.2	Hardware implementation of adders and subtractors	67
3.2.3	One's Complement Addition and Subtraction	69
3.3	FIXED POINT MULTIPLICATION AND DIVISION	71
3.3.1	Unsigned Multiplication	71
3.3.2	Unsigned Division	73
3.3.3	Signed Multiplication and Division	75
3.4	FLOATING POINT ARITHMETIC	77
3.4.1	Floating Point Addition and Subtraction	77
3.4.2	Floating Point Multiplication and Division	78
3.5	HIGH PERFORMANCE ARITHMETIC	79
3.5.1	High Performance Addition	79
3.5.2	High Performance Multiplication	81
3.5.3	High Performance Division	85
3.5.4	Residue Arithmetic	88
3.6	CASE STUDY: CALCULATOR ARITHMETIC USING BINARY CODED DECIMAL	91
3.6.1	The HP9100A Calculator	92
3.6.2	Binary Coded Decimal Addition and subtraction	92
3.6.3	BCD Floating Point Addition and Subtraction	95
4	MACHINE LANGUAGE AND ASSEMBLY LANGUAGE	103
4.1	HARDWARE COMPONENTS OF THE INSTRUCTION SET ARCHITECTURE	104
4.1.1	The System Bus Model Revisited	104
4.1.2	Memory	105
4.1.3	The CPU	108
4.2	ARC, A RISC COMPUTER	112

4.2.1	ARC Memory	113
4.2.2	ARC Instruction set	114
4.2.3	ARC Assembly Language Format	116
4.2.4	ARC Instruction Formats	118
4.2.5	ARC Data Formats	120
4.2.6	ARC Instruction Descriptions	120
4.3	PSEUDO-OPS	125
4.4	EXAMPLES OF ASSEMBLY LANGUAGE PROGRAMS	126
4.4.1	Variations in machine architectures and addressing	129
4.4.2	Performance of Instruction Set Architectures	132
4.5	ACCESSING DATA IN MEMORY—ADDRESSING MODES	132
4.6	SUBROUTINE LINKAGE AND STACKS	134
4.7	INPUT AND OUTPUT IN ASSEMBLY LANGUAGE	139
4.8	CASE STUDY: THE JAVA VIRTUAL MACHINE ISA	143
5	WORKING WITH ASSEMBLY LANGUAGE	157
5.1	THE ASSEMBLY PROCESS	157
5.2	LINKING AND LOADING	163
5.2.1	Linking	164
5.2.2	Loading	166
5.3	MACROS	169
5.4	CASE STUDY: EXTENSIONS TO THE INSTRUCTION SET – THE INTEL MMX™ AND MOTOROLA ALTIVEC™ SIMD INSTRUCTIONS.	171
5.4.1	Background	171
5.4.2	The Base Architectures	172
5.4.3	VECTOR Registers	173
5.4.4	Vector Arithmetic operations	176
5.4.5	Vector compare operations	178
5.4.6	Summary	179
6	THE CONTROL UNIT	185
6.1	BASICS OF THE MICROARCHITECTURE	186
6.2	A MICROARCHITECTURE FOR THE ARC	187
6.2.1	The Datapath	187
6.2.2	The Control Section	195
6.2.3	Timing	199
6.2.4	Developing the Microprogram	200
6.2.5	Traps and Interrupts	211
6.2.6	Nanoprogramming	213

6.3	HARDWIRED CONTROL	214
6.4	CASE STUDY: A USER-MICROPROGRAMMABLE COMPUTER: THE PDP11/60	223
6.4.1	Microprogrammed vs. hardwired control units	224
6.4.2	Statistics in computer design	224
6.4.3	The PDP 11/60 Control unit	225
6.4.4	Writable control store and user microprogramming	226
7	MEMORY	237
7.1	THE MEMORY HIERARCHY	237
7.2	RANDOM ACCESS MEMORY	239
7.3	CHIP ORGANIZATION	240
7.3.1	Constructing LARGE RAMS FROM SMALL RAMS	243
7.4	COMMERCIAL MEMORY MODULES	244
7.5	READ-ONLY MEMORY	245
7.6	CACHE MEMORY	248
7.6.1	Associative Mapped Cache	250
7.6.2	Direct Mapped Cache	253
7.6.3	Set Associative Mapped Cache	256
7.6.4	Cache performance	257
7.6.5	Hit Ratios and Effective Access Times	259
7.6.6	Multilevel Caches	261
7.6.7	Cache management	261
7.7	VIRTUAL MEMORY	263
7.7.1	Overlays	263
7.7.2	Paging	265
7.7.3	Segmentation	268
7.7.4	Fragmentation	269
7.7.5	Virtual Memory vs. Cache Memory	271
7.7.6	THE TRANSLATION LOOKASIDE BUFFER	271
7.8	ADVANCED TOPICS	273
7.8.1	Tree decoders	273
7.8.2	Decoders for large RAMs	274
7.8.3	Content Addressable (Associative) Memories	275
7.9	CASE STUDY: RAMBUS MEMORY	280
7.10	CASE STUDY: THE INTEL PENTIUM MEMORY SYSTEM	283
8	INPUT, OUTPUT, AND COMMUNICATION	293
8.1	MASS STORAGE	294
8.1.1	Magnetic Disks	294

8.1.2	Magnetic Tape	302
8.1.3	Magnetic Drums	303
8.1.4	Optical Disks	304
8.2	INPUT DEVICES	306
8.2.1	Keyboards	306
8.2.2	Bit Pads	308
8.2.3	Mice and Trackballs	308
8.2.4	Lightpens and Touch Screens	309
8.2.5	Joysticks	310
8.3	OUTPUT DEVICES	311
8.3.1	Laser Printers	311
8.3.2	Video Displays	312
8.4	COMMUNICATION	315
8.4.1	Busses	315
8.4.2	Communication between processors and memories	322
8.4.3	I/O Channels	327
8.4.4	Modems	329
8.4.5	Local Area Networks	331
8.5	COMMUNICATION ERRORS AND ERROR CORRECTING CODES	338
8.5.1	Bit Error Rate Defined	338
8.5.2	Error Detection and Correction	339
8.5.3	Vertical Redundancy Checking	345
8.5.4	Cyclic Redundancy Checking	346
8.6	CASE STUDY: COMMUNICATION ON THE INTEL PENTIUM ARCHITECTURE	350
9	MODERN ARCHITECTURES	363
9.1	THE EMERGENCE OF RISC ARCHITECTURES	363
9.2	QUANTITATIVE ANALYSES OF PROGRAM EXECUTION	364
9.3	OVERLAPPING REGISTER WINDOWS	366
9.4	PIPELINING THE DATAPATH	368
9.5	MULTIPLE INSTRUCTION ISSUE MACHINES	371
9.6	VLIW MACHINES	371
9.7	CASE STUDY: INSPECTING COMPILED CODE	371
9.8	CASE STUDY: SUPERSCALAR ASSEMBLY LANGUAGE PROGRAMMING ON THE INTEL PENTIUM II WITH MMX TECHNOLOGY	377
10	ADVANCED TOPICS	381
10.1	PARALLEL ARCHITECTURE	381
10.1.1	The Flynn Taxonomy	383

10.1.2	Interconnection Networks	385
10.1.3	Mapping an Algorithm onto a Parallel Architecture	391
10.1.4	Fine-Grain Parallelism – The Connection Machine CM-1	396
10.1.5	Course-Grain Parallelism: The CM-5	399
10.2	CASE STUDY: PARALLEL PROCESSING IN THE SEGA GENESIS	402
10.2.1	The SEGA Genesis Architecture	402
10.2.2	Sega Genesis Operation	404
10.2.3	Sega Genesis Programming	404
10.3	NETWORK ARCHITECTURE: THE INTERNET	406
10.3.1	The Internet Model	406
10.3.2	Transmission media	411
10.3.3	Bridges and Routers Revisited, and Switches	416

A APPENDIX A: DIGITAL LOGIC

421

A.1	INTRODUCTION	421
A.2	COMBINATIONAL LOGIC	421
A.3	TRUTH TABLES	422
A.4	LOGIC GATES	424
A.4.1	Electronic implementation of logic gates	427
A.4.2	Tri-STATE Buffers	430
A.5	PROPERTIES OF BOOLEAN ALGEBRA	430
A.6	THE SUM-OF-PRODUCTS FORM, AND LOGIC DIAGRAMS	433
A.7	THE PRODUCT-OF-SUMS FORM	435
A.8	POSITIVE VS. NEGATIVE LOGIC	437
A.9	THE DATA SHEET	439
A.10	DIGITAL COMPONENTS	441
A.10.1	Levels of Integration	441
A.10.2	Multiplexers	442
A.10.3	Demultiplexers	444
A.10.4	Decoders	445
A.10.5	Priority Encoders	447
A.10.6	Programmable Logic Arrays	447
A.11	SEQUENTIAL LOGIC	452
A.11.1	The S-R Flip-Flop	453
A.11.2	The Clocked S-R Flip-Flop	455
A.11.3	The D Flip-Flop and the Master-Slave Configuration	457
A.11.4	J-K and T Flip-Flops	459
A.12	DESIGN OF FINITE STATE MACHINES	460
A.13	MEALY VS. MOORE MACHINES	469

- A.14 REGISTERS 470
- A.15 COUNTERS 471

B APPENDIX B: REDUCTION OF DIGITAL LOGIC **483**

- B.1 REDUCTION OF COMBINATIONAL LOGIC AND SEQUENTIAL LOGIC 483
- B.2 REDUCTION OF TWO-LEVEL EXPRESSIONS 483
 - B.2.1 The Algebraic Method 484
 - B.2.2 The K-Map Method 485
 - B.2.3 The Tabular Method 494
 - B.2.4 Logic reduction: EFFECT ON speed and performance 502
- B.3 STATE REDUCTION 506
 - B.3.1 The State Assignment Problem 510
 - B.3.2 Excitation Tables 514

SOLUTIONS TO PROBLEMS **529**

INDEX **581**

