COMPUTERS FOR ARTIFICIAL INTELLIGENCE PROCESSING

Edited by Benjamin W. Wah and C. V. Ramamoorthy

A WILEY-INTERSCIENCE PUBLICATION
John Wiley & Sons, Inc.
NEW YORK / CHICHESTER / BRISBANE / TORONTO / SINGAPORE

In recognition of the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc. to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

DEC, PDP, VAX, MicroVAX, and VMS are trademarks of Digital Equipment Corporation. IBM is a registered trademark of International Business Machines, Inc. Any use of any other trademarked or registered names in this book is for editorial purposes only and is not intended in any way to infringe on the rights of the trademark holder.

Copyright © 1990 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging in Publication Data:

Computers for artificial intelligence processing / edited by Benjamin W. Wah and C. V. Ramamoorthy.
p. cm.
"A Wiley-Interscience publication."
Includes bibliographical references.

1. Electronic digital computers. 2. Computer architecture. 3. Artificial intelligence. I. Wah, Benjamin W. II. Ramamoorthy, C. V. (Chitoor V.), 192

QA76.5.C6144 1990
006.-dc20 89-77799
ISBN 0-471-84811-5 CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

CONTENTS

Contributors **xvii**

Preface **xxi**

SECTION I. INTRODUCTION

1. "Computers for Symbolic Processing" **1**
by Benjamin W. Wah, Matthew B. Lowrie, and Guo-Jie Li

1 Symbolic Processing / 1
- 1.1 Introduction / 1
- 1.2 Classification of Computations / 3
- 1.3 Characteristics of Symbolic Processing Applications / 5

2 Knowledge Representation and Processing / 8
- 2.1 Knowledge Representation / 8
- 2.1.1 Features of Knowledge Representations / 9
- 2.1.2 Classical Knowledge Representation Schemes / 11
- 2.2 Knowledge Processing / 13
- 2.2.1 Uncertain, Incomplete, and Inconsistent Knowledge Processing / 13
- 2.2.2 Parallel Knowledge Processing / 17

3 Architectural Concepts for Symbolic Processing / 18
- 3.1 Software Architectures / 18
- 3.1.1 Functional Programming Languages / 21
- 3.1.2 Rule-Based Languages / 23

3.1.3 Object-Oriented Languages / 26
3.1.4 Mapping Applications into Software / 27
3.2 Hardware Architectural Support for Symbolic Processing / 28
3.2.1 Microlevel Hardware Features / 29
3.2.2 Subsystem-Level Architectures / 36

4 Complete Systems / 45
4.1 Single-Processor Symbolic Computers / 47
4.2 Parallel Symbolic Processors / 49
4.2.1 Communication and Synchronization / 49
4.2.2 Parallel Functional Programming Computers / 51
4.2.3 Parallel Logic Architectures / 53
4.2.4 Parallel Systems for Production Systems Computations / 53
4.2.5 Parallel Object-Oriented Architectures / 55
4.3 Connectionist Processing / 56
4.4 Summary / 57

5 Research Directions / 59

6 Acknowledgements / 60

References / 60

SECTION II. LANGUAGE-BASED AI ARCHITECTURES

2. "Architectural Features of Lisp Computers" **74**
by Andrew R. Pleszkun and Matthew J. Thazhuthaveetil

1 Run Time Requirements of a Lisp System / 75
1.1 Function Calling / 75
1.2 Dealing with Lists / 76
1.3 Summary / 78

2 Approaches to LISP Machine Design / 78
2.1 Function Calling / 79
2.2 Maintaining the Environment / 81
2.3 Efficient Lisp Representation / 86
2.3.1 Vector-Coded Representations of Lists / 86
2.3.2 Structure-Coded Representations of Lists / 88
2.3.3 Summary / 91
2.4 Heap Maintenance / 91

3 Conclusions / 94

References / 95

3. "Symbolics Architecture" **98**
by David A. Moon

1 System Architecture / 99
2 Instruction Architecture / 104
3 Processor Architecture / 114
4 Conclusion / 117
References / 118
About the Author / 118

4. "Memory Management and Usage in a Lisp System: A Measurement-Based Study" **119**
by Rene L. Llames and Ravi K. Iyer

1 Introduction / 119
2 Related Work / 120
3 Garbage Collection in Virtual Memory / 122
3.1 Incremental Copying Garbage Collection / 122
3.2 Approximately Depth-first Copying / 124
3.3 Generational Garbage Collection / 124
3.4 Tagged Architecture and Special Hardware / 125
4 The Measurement Environment / 125
4.1 The Data / 126
4.2 The Workload / 126
4.3 Software Sampling and Interference Effects / 127
5 Paging and Storage Allocation Activity / 128
5.1 Preliminaries / 128
5.2 Totals and Ratios / 130
5.3 Distributions / 133
5.4 Time Series / 139
5.5 Scatter Plots / 146
6 Garbage Collector Analysis / 148
6.1 Regression Models / 152
6.2 Regression Results / 154
7 Memory Usage / 158
7.1 Dynamic versus Ephemeral versus No Collection / 159
7.2 Allocation Behavior / 162
7.3 Collection Cycle Characteristics / 162
7.4 Static Space / 167
7.5 Memory Usage Summary / 167

8 Conclusions / 168

References / 169

5. **"Multiprocessor Architectural Support for Balanced Lisp Processing"** **171**
by Raymond Chowkwanyun and Kai Hwang

1 Introduction / 171

2 Load Balancing in Multiprocessors / 173

3 O/S Functions for Process Migration / 176

4 The Hybrid Load Balancing System / 179

4.1 Receiver-Initiated Mode / 179

4.2 Sender-Initiated Mode / 180

4.3 The Hybrid Load Balancing Method / 182

4.4 The Macro Data-Flow Execution Model / 182

5 Hardware Support for Hybrid Load Balancing / 184

6 Software Implementation of the Hybrid Load Balancer / 188

7 Parallelism in Executing Lisp Programs on Multiprocessors / 191

8 Translating Sequential Programs for Parallel Execution / 193

9 Parallelization of Lisp Benchmark Programs / 195

9.1 The Tak Program / 195

9.2 The Boyer Program / 196

9.3 The Browse Program / 197

9.4 The Traverse Program / 198

9.5 Translation Effort / 199

10 Experimental Concurrent Lisp Benchmark Results / 200

11 Conclusions / 206

12 Acknowledgements / 207

References / 207

6. **"Data-Flow Computing Models, Logic and Functional Languages, and Data-Flow Machines for Intelligence Computations"** **209**
by Jayantha Herath, Yoshinori Yamaguchi, Susantha Herath, Nobuo Saito, and Toshitsugu Yuba

1 Introduction / 209

1.1 Parallelism in Computations / 209

1.2 Ideal Parallelism / 211
1.3 Overview / 211

2 Data-Flow Computing Models / 213
2.1 Basic Models / 213
2.2 Static Computing / 215
2.2.1 Strictly Static / 216
2.2.2 Acknowledgement Static / 216
2.3 Recursive Dynamic Computing / 216
2.4 Tagged-Token Dynamic / 216
2.5 Eduction Computing / 218
2.6 Data-Flow–Control-Flow Computing / 219
2.7 Eager-Lazy Computing / 220
2.8 Psuedoresult Computing / 221
2.9 Not(operation) Computing / 222

3 Logic and Functional Programming for Data-Flow Computing / 223
3.1 Logic Programming / 224
3.1.1 Unification / 224
3.1.2 Prolog / 228
3.1.3 Relational Language / 228
3.1.4 Parlog / 228
3.1.5 Concurrent Prolog / 228
3.2 Functional Programming Languages / 229
3.2.1 VAL / 230
3.2.2 Id / 230
3.2.3 LUCID / 230
3.2.4 Manchester Languages / 231
3.2.5 VALID / 232
3.2.6 EMLISP / 232
3.3 DCBL Transformations for Data-Flow Computing Languages / 232
3.3.1 Specification of DCBL / 233
3.3.2 Data-Flow Graph Specification Language / 234
3.3.3 DCBL Transformation / 235
3.3.4 Functionality / 236
3.3.5 DCBL Transformations in LISP / 238

4 Data-Flow Computing Machines / 241
4.1 Static Machines / 242
4.1.1 VIM / 242

4.1.2 Texas Distributed Data Processor / 243
4.1.3 LAU System / 245
4.1.4 NEDIPS / 246
4.2 Dynamic Machines / 246
4.2.1 MIT Tagged-Token Data-Flow Machine / 246
4.2.2 Manchester Data-Flow Computer / 247
4.2.3 DDM1 / 248
4.2.4 SIGMA-1 / 249
4.2.5 EM-3 / 251
4.2.6 EDDY / 252
4.2.7 DFM / 252
4.2.8 PIM-D / 254
4.2.9 TOPSTAR / 255
4.3 Other Projects / 255
4.4 Problems in Data-Flow Computing Machines / 255
4.4.1 Matching Bottleneck / 255
4.4.2 Remaining Packet Garbage / 256
4.4.3 Control of Parallelism / 256
4.4.4 Sequential Computing Segments / 257
4.4.5 Parallel Execution of Conditional Computations / 257
4.4.6 Optimisation of Data-Flow Computations / 257

5 Performance Evaluations Using the EM-3 / 257
5.1 EM-3 Operational Model / 257
5.2 EMIL / 258
5.3 Performance Evaluation Measurements / 259
5.3.1 Effectiveness of Psuedoresult Model / 259
5.3.2 Effectiveness of Not(operation) Model / 260

6 Conclusions / 266
6.1 Further Research / 267

7 Acknowledgements / 268

References / 268

7. "Design Decisions in SPUR" **273**
by Mark Hill, Randy Katz, John Ousterhout, David Patterson, et al.

1 System Overview / 274

2 Processor Overview / 276

3 The Memory System / 277

4 The CPU and Floating-Point Coprocessor / 285
4.1 General-Purpose Features / 286

5 Status / 296

6 Acknowledgements / 297

References / 298

8. "What Price Smalltalk?" **300**
by David Ungar and David Patterson

1 Introduction / 300

2 The Demands of Smalltalk-80 / 300

3 The Costs on Traditional Systems / 302

4 Reducing Costs Through Software Innovation / 304
4.1 Interpretation / 304
4.2 Caching Call Targets in Line / 305
4.3 Object-Oriented Storage Management / 306
4.4 Automatic Storage Reclamation / 307

5 Reducing Costs Through Architectural Innovation / 309
5.1 Tags Trap Bad Guesses / 309
5.2 Multiple Overlapping On-Chip Register Windows / 310

6 The Architect's Trap / 311

7 Acknowledgements / 314

References / 314

9. "Special Purpose Chip for Production Systems" **316**
by G. T. Alley, W. L. Bryan, R. O. Eason, D. F. Newport, and D. W. Bouldin

1 Introduction / 316

2 The Architecture / 317

3 The VLSI Implementation / 318

4 Operational Modes / 318

5 Functional Elements / 319

6 Additional Functions / 320

7 Conclusions / 321

References / 322

SECTION III. MULTIPROCESSOR AI ARCHITECTURE

10. "Applications of the Connection Machine" **324**
by David L. Waltz

1 Introduction to the Connection Machine System / 324

2 Data Level Parallelism / 325

2.1 Hardware and Software: System-Level Specifications / 326

3 Document Retrieval / 327

3.1 Document Retrieval by Relevance Feedback / 328

4 Building a Database on the Connection Machine System / 331

5 Programming the Document Retrieval System / 332

5.1 Document Lookup on the Connection Machine System / 333

5.2 Retrieving the Highest-Scoring Documents / 334

5.3 Timing and Performance / 335

6 Memory-Based Reasoning Systems / 336

6.1 The Operation of Memory-Based Reasoning / 337

6.2 A System for Medical Reasoning / 337

6.3 Evaluation of Memory-Based Reasoning / 341

7 Basic Operations: Scanning / 341

8 Bulk Processing of Natural Language / 344

9 Other Applications / 346

10 Conclusions / 348

11 Acknowledgments / 349

References / 349

11. "A Database Machine Based on Concatenated Code Words for Very Large Databases" **352**
by Soon Myoung Chung and P. Bruce Berra

1 Introduction / 352

2 Backend Database Machine / 354

2.1 General Structure / 354

2.2 Surrogate File Processing System / 355

2.3 EDB Processing System / 357

2.4 Processing Mode and the Multiple Backend System / 358

3 Relational Operations in the Backend Database Machine / 358

3.1 Selection Operation / 359

3.2 Join Operation / 360

4 Performance Analysis of the Proposed Database Machine / 362

4.1 Selection Operation / 363

4.2 Join Operation / 365

4.3 Performance of the Proposed Machine with Clustered CCW Surrogate Files / 370

5 Conclusion / 373

References / 373

SECTION IV. CONNECTIONIST ARCHITECTURES AND APPLICATIONS

12. "Connectionist Architectures for Artificial Intelligence" 376
by Scott E. Fahlman and Geoffrey E. Hinton

1 What is Connectionism? / 378

2 Distributed Representations / 379

3 NETL: A Connectionist System for Symbolic Knowledge / 380

4 Layered Value-Passing Networks for Recognition / 383

5 Learning Representations / 386

6 Constraint-Satisfaction in Iterative Networks / 388

7 Hopfield and Boltzmann Networks for Constraint Satisfaction / 388

8 The Boltzmann Machine Learning Procedure / 391

9 Acknowledgments / 393

10 Further Reading / 393

References / 393

13. "Architectures for Strategy Learning" 395
by Pankaj Mehra and Benjamin W. Wah

1 Introduction / 395

1.1 Problem Solving Strategy / 396

1.2 Strategic Knowledge / 397

1.3 Learning / 397

1.4 Nomenclature of Strategy-Learning Paradigms / 398

1.5 Architectures for Strategy-Learning Systems / 398

1.6 Methods for Problem Solving / 401
1.7 Problem Representation / 402
1.8 Problem Solving Environment / 403
1.9 Overview of This Chapter / 404

2 The Need for Studying Strategy-Learning Systems / 405

3 A Taxonomy of Strategy-Learning Problems / 408
3.1 Nature of the Objective Function / 408
3.2 Immediate Feedback versus Delayed Feedback / 411
3.3 Background Knowledge of the Environment / 412
3.4 The Nature of Feedback / 413
3.5 Strategy Selection versus Strategy Construction / 414
3.6 Uncertain and Incomplete Information / 415
3.7 Resource and Time Constraints / 415
3.8 Types of Available Learning Techniques / 415

4 Complexity Classes for Strategy Learning / 417

5 Dynamic Decision Problems / 420
5.1 Characteristics of Dynamic Decision Problems / 422
5.2 An Example of Dynamic Decision Problems / 423

6 Survey of Strategy Learning Systems / 424
6.1 Cognitive Models of Skill Learning / 426
6.2 Strategy-Learning Systems for Making Choices / 427
6.3 Strategy-Learning Methods in Statistical Decision Theory / 428
6.4 Strategy-Learning Methods in Artificial Intelligence / 430
6.4.1 Empirical Learning Techniques / 430
6.4.2 Analytical Learning Techniques / 434
6.4.3 Knowledge-Based Methods / 434
6.4.4 Analogy-Based Methods / 440
6.4.5 Hybrid Methods / 441
6.5 Connectionist Methods / 443

7 A Proposed Model of Learning Systems / 446

8 Conclusion / 451

9 Acknowledgments / 455

References / 455

SECTION V. SOFTWARE ARCHITECTURES FOR AI APPLICATIONS

14. "AI and Software Engineering: A Clash of Cultures?" **469**
by Wei-Tek Tsai, K. Heisler, D. Volovik, and I. A. Zualkernan

1 Introduction / 469

2 AI for SE / 475
2.1 The Role of Automatic Programming / 475
2.2 The Role of Expert Systems / 478
2.3 The Role of AI Languages and Environments / 482
2.4 The Role of Automated Software Design / 485
2.5 The Role of Rapid Prototyping / 488

3 SE for AI / 493
3.1 The Role of the Waterfall Model / 493
3.2 The Role of Specification / 499

4 Summary / 504

References / 505

Appendix: Recent Developments / 510
AI for Software Engineering / 510
Software Engineering for AI / 511
References / 512

15. "Development Support for AI Programs" **513**
by C. V. Ramamoorthy, Shashi Shekhar, and Vijay Garg

1 Introduction / 513

2 Inside an AI Program / 514

3 AI Programming Paradigms and Languages / 516
3.1 OPS-5: Production Systems / 517

4 Development Support for AI Programs / 521
4.1 Tools From Prolog Environments / 523
4.2 The Interlisp Environment / 524
4.3 Life Cycle Support for AI Programming / 526
4.4 Knowledge-Based Tools / 528

5 Conclusions / 529

6 Acknowledgements / 530

References / 530

16. "Reliability of AI Programs" **532**
by Farokh B. Bastani

1 Introduction / 532

2 Review of Hardware Reliability / 534

3 Review of Software Reliability / 538

3.1 Software Reliability Growth Models / 538

3.2 Sampling Models / 540

3.3 Fault Seeding / 542

3.4 System Reliability / 543

3.5 Discussion / 543

4 Characteristics of AI Programs / 544

5 Software Reliability Model / 546

6 System Reliability / 550

6.1 Response Computation Time / 551

6.1.1 Generate and Test Procedure / 551

6.1.2 Branch and Bound Precedure / 552

6.1.3 A* with Unique Solution Paths / 553

6.1.4 A* with Multiple Solution Paths / 554

6.1.5 Planning in a Changing Environment / 556

6.2 Response Execution Time / 558

7 Summary / 559

8 Acknowledgments / 560

References / 560

Index **563**

CONTRIBUTORS

G. T. Alley
Instrumentation & Controls Division
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6006
Chapter 9

Farokh B. Bastani
Department of Computer Science
University of Houston
Houston, TX 77025
Chapter 16

Professor P. Bruce Berra
Department of Electrical and Computer Engineering
Syracuse University
111 Lind Hall
Syracuse, NY 13210
Chapter 11

D. W. Bouldin
Electrical & Computer Engineering
University of Tennessee
Knoxville, TN 37996-2100
Chapter 9

W. L. Bryan
Instrumentation & Controls
Division
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6006
Chapter 9

Dr. Soon Myoung Chung
Department of Computer Science and Engineering
Wright State University
Dayton, OH 45435
Chapter 11

R. O. Eason
Electrical Engineering
University of Maine
Orono, ME 04469
Chapter 9

Scott E. Fahlman
School of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213
Chapter 12

Dr. Vijay Garg
Computer Science Division
University of California/Berkeley
Berkeley, CA 94720
Chapter 15

Jayantha Herath
Department of Electrical
& Computer Engineering
Drexel University
Philadelphia, PA 19104
Chapter 6

Susantha Herath
Department of Electrical
& Computer Engineering
Keio University
Yokohama, JAPAN
Chapter 6

Geoffrey E. Hinton
Computer Science Department
University of Toronto
Toronto M5S 1A4 CANADA
Chapter 12

Kai Hwang
Department of Electrical
Engineering
University of Southern California
Los Angeles, CA 90089-0781
Chapter 5

Ravi K. Iyer
Coordinated Science Laboratory
University of Illinois
1101 W. Springfield Avenue
Urbana, IL 61801
Chapter 4

Guo-Jie Li
Institute of Computing Technology
Academia Sinica
P.O. Box 2704-1
Beijing, PROC
Chapter 1

Rene L. Llames
Coordinated Science Laboratory
University of Illinois
1101 W. Springfield Avenue
Urbana, IL 61801
Chapter 4

Matthew B. Lowrie
926 North Scott
Wheaton, IL 60187
Chapter 1

Pankaj Mehra
Coordinated Science Laboratory
University of Illinois
1101 W. Springfield Avenue
Urbana, IL 61801
Chapter 13

David A. Moon
Symbolics Inc.
8 New England Executive Park East
Burlington, MA 01803
Chapter 3

D. F. Newport
Electrical & Computer Engineering
University of Tennessee
Knoxville, TN 37996-2100
Chapter 9

SPUR Project
C/O Professor David Patterson
Computer Science Division
University of California
Berkeley, CA 94720
Chapters 7 & 8

Andrew R. Pleszkun
Department of Electrical
& Computer Engineering
University of Colorado
Campus Box 425
Boulder, CO 80309
Chapter 2

Professor C. V. Ramamoorthy
Computer Science Division
University of California/Berkeley
Berkeley, CA 94720
Chapter 15

Nobuo Saito
Department of Computer Science
Keio University
Yokohama, JAPAN
Chapter 6

Professor Shashi Shekhar
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
Chapter 15

Matthew J. Thazhuthaveetil
Department of Electrical Engineering
Pennsylvania State University
University Park, PA 16802
Chapter 2

Dr. Wei-Tek Tsai
Department of Computer Science
University of Minnesota
4-192 EE/CSCI Bldg.,
200 Union St. SE
Minneapolis, MN 55455
Chapter 14

David Ungar
Department of Computer Science
Stanford University
Palo Alto, CA 94305
Chapter 8

Benjamin W. Wah
Coordinated Science Laboratory
University of Illinois
1101 W. Springfield Avenue
Urbana, IL 61801
Chapters 1 & 13

David L. Waltz
Thinking Machines Corporation
245 First Street
Cambridge, MA 02142-1214
Chapter 10

Yoshinori Yamaguchi
Electrotechnical Lab
Sakuramura Niiharigun
Tsukubu 305 JAPAN
Chapter 6

Toshitsugu Yuba
Electrotechnical Lab
Sakuramura Niiharigun
Tsukubu 305 JAPAN
Chapter 6

PREFACE

This book addresses the increasing complexity and the growing need for computational power of artificial intelligence (AI) algorithms and programs. These algorithms and software, which share many common features with symbolic processing, are not supported efficiently by conventional von Neumann computers, which are oriented towards numeric processing. Their efficient evaluation requires new architectural designs, languages, algorithms, and representation schemes to be developed.

This book presents fundamentals in architectures, languages, and software designs for supporting AI applications. It provides a comprehensive treatment of the design issues and current state-of-the-art research efforts in this area, and illustrates these solutions with example designs. The discussion spans from hardware architectures to software engineering methods to meta-level strategy designs.

This book represents a collective effort of fifty-one authors, all recognized experts in areas of computer architecture, parallel processing, artificial intelligence, and software engineering. It was developed over a period of three years and reflects some of the leading efforts in this area.

This book can serve as a reference text for researchers and developers working in the area, as well as an introductory text for beginners. It can also serve as a reference text to accompany an advanced course on computer architecture. The topics selected for presentation provide an overview of the area as well as an in-depth discussion of some of the important and difficult problems in the area. The material presented assumes a basic knowledge on computer system design, computer architecture, artificial intelligence, and software design methods. A senior in Computer Science will possess the necessary background for understanding the material presented.

This book is organized into five major sections. Each section delineates a specific aspect of the problem and may have one or more chapters.

Section 1 presents a comprehensive survey on the design issues and examples of computers oriented towards symbolic processing. An extensive bibliography accompanies the discussion.

Section 2 discusses the design and implementation of special-purpose language-oriented computers for supporting AI processing. Special-purpose languages studied include functional languages, Lisp, production systems, and Smalltalk. Three chapters are devoted to sequential Lisp processing, with discussions on the design issues, memory management, performance evaluation, and an example illustrated with the Symbolics Lisp computer. Three chapters are devoted to multiprocessing and parallel processing of Lisp programs and, in general, functional programs. The last two chapters in this section present architectures for supporting Smalltalk-80 and production systems.

Section 3 examines multiprocessor systems for general AI processing. The Connection Machine is drawn as an example of a symbolic multiprocessor with data-level parallelism. Design of large data/knowledge base machines for AI processing is also studied.

Section 4 discusses connectionist architectures and applications. One chapter is devoted to illustrating the benefits and design issues of connectionist systems. A second chapter presents an extensive survey on connectionist architectures, as well as other computing architectures designed for learning strategies.

The last section addresses software architectures for AI applications and the design of AI software as a software engineering project, two important issues that are largely neglected in the literature. Three aspects are examined: AI and software engineering, development tools for AI programs, and reliability of AI programs.

We would like to thank all the authors who participated in this project for their dedication and patience. We are also grateful to the reviewers who provided many constructive criticisms on this work. We would like to acknowledge the partial support of this project by the National Aeronautics and Space Administration under contract NCC 2-481. Lastly, we are indebted to Miss Vickie DeMoss, who spent many late evenings to enter the text and draw the figures using Interleaf's University Publishing System.

Benjamin W. Wah *Urbana-Champaign, Illinois*

C. V. Ramamoorthy *Berkeley, California*