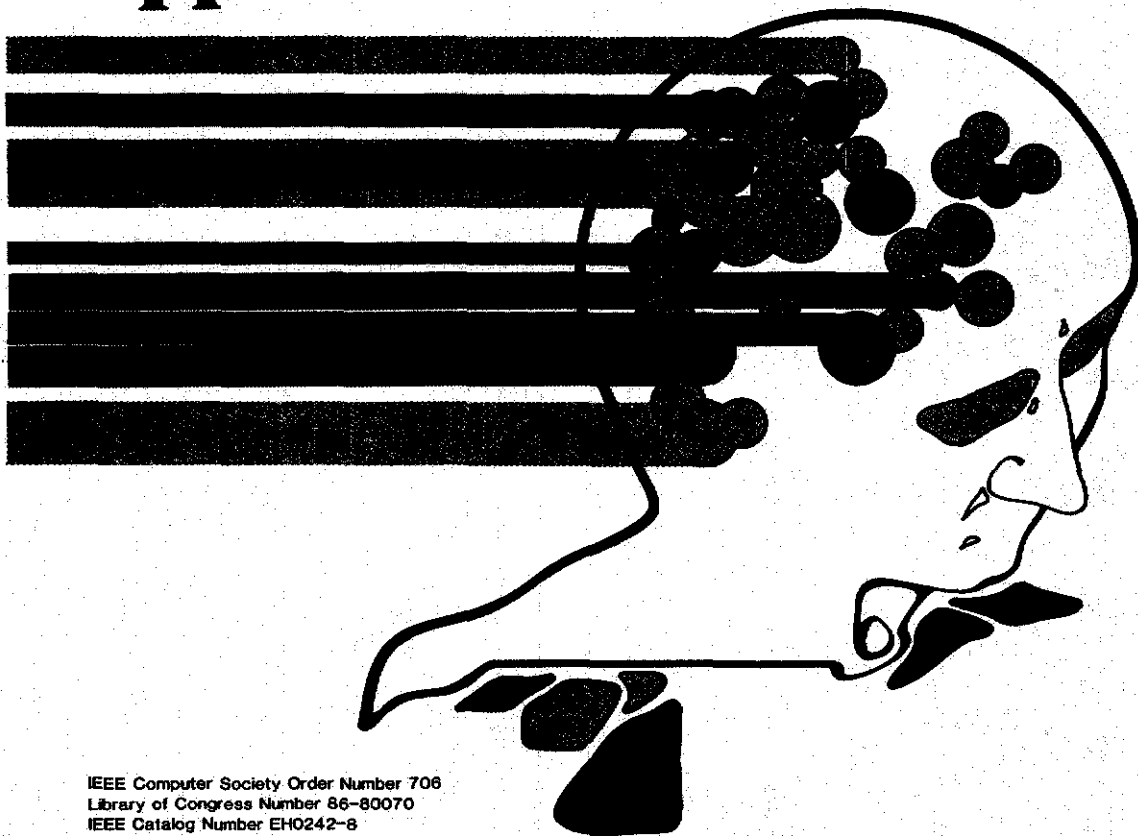


T U T O R I A L

COMPUTERS for artificial intelligence applications

Benjamin Wah and G.-J. Li



IEEE Computer Society Order Number 706
Library of Congress Number 86-80070
IEEE Catalog Number EH0242-8
ISBN 0-8186-0708-8

 **IEEE COMPUTER SOCIETY**


**THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INC.**
IEEE

**THE
COMPUTER
SOCIETY
PRESS** 

Published by IEEE Computer Society Press
1730 Massachusetts Avenue, N.W.
Washington, D.C. 20036-1903

COVER DESIGNED BY JACK I. BALLESTERO

Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress Street, Salem, MA 01970. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint or republication permission, write to Director, Publishing services, IEEE, 345 E. 47 St., New York, NY 10017. All rights reserved. Copyright © 1986 by The Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Order Number 706
Library of Congress Number 86-80070
IEEE Catalog Number EH0242-8
ISBN 0-8186-0706-8 (Paper)
ISBN 0-8186-4706-X (Microfiche)

Order from: IEEE Computer Society
Post Office Box 80452
Worldway Postal Center
Los Angeles, CA 90080

IEEE Service Center
44 Hoes Lane
Piscataway, NJ 08854



THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

This
for art
to the
incre
beca
is as
appl
exp
eve
be
m
g
c

Preface

This tutorial presents fundamentals in language-oriented and knowledge-oriented computer architectures for artificial intelligence applications, in short, AI architectures. As AI applications move from laboratories to the real world and as AI software grows in complexity, computational throughput, and cost are increasingly important concerns. Conventional von Neumann computers are unsuitable for AI applications because they are designed mainly for numerical processing. Although the idea of building intelligent systems is as old as the AI field itself, it is only in recent years that the design of special computer architectures for AI applications becomes feasible with the decreasing hardware costs, the advent in VLSI technology, and the experience gained with AI programming. Opportunities for increased efficiency in AI computations exist at every level, from improved instruction-set designs to parallel computer systems. A great amount of effort has been devoted to investigating and developing efficient AI architectures, and this topic is becoming more and more attractive for researchers and designers in the areas of computers and AI.

Solving a problem in AI, as well as many other problems, can be regarded as evaluating an algorithm. In general, the algorithm for solving any problem on a computer can be coded in software and executed on a conventional von Neumann computer. To cope with the increasing inefficiency and difficulty in coding algorithms in AI by using conventional languages, *declarative languages* have been developed. *Lambda-based* and *logic-based languages* are the two popular classes of declarative languages.

One of the starting points of the computer architect in supporting AI applications is the language rather than the algorithms. Of course, some primitive operations may be designed directly into hardware without first being translated by using a high-level language. This approach has been termed the *language-first approach*. A possible disadvantage of this approach is that each language may lead to a somewhat distinct architecture unsuitable for other languages, a dilemma in high-level-language computer architectures. In AI applications, the lambda-based and logic-based languages have been considered seriously by novel architects. Recent research lies in integrating the logic and lambda languages with procedural languages. The work on lambda- and logic-oriented architectures also provides useful guidelines for designing parallel architectures to support more advanced languages.

Another starting point for the computer architect is to design architectures to support the management of a large volume of knowledge. Recognizing that the major overhead lies in the search and manipulation of knowledge, the architect designs efficient storage mechanisms with distributed intelligence to support queries and updates of knowledge. The resulting architecture is strongly dictated by the knowledge representation schemes. A hybrid of the language-first and the knowledge-first approaches is of growing interest.

In spite of the growing importance of AI applications, the issues of designing these architectures have never been fairly and completely explored in a single publication. In fact, work in this area is so diversified that articles were published in other areas besides AI, ranging from psychology, medicine, manufacturing, computer architecture, software engineering, and database management to industrial engineering, electronics, theory of computation, communications, image processing, speech understanding, operations research, and the list grows. The literature search is also complicated by the fact that with the development of the Fifth-Generation Computer Systems, some work in this area is very recent and was published in many foreign countries. During our literature search to compile this tutorial text, we systematically went through over 60 different journals published in various countries and proceedings from over 50 conferences in the last 20 years and over 70 books. In view of this, the authors believe that a tutorial is urgently needed to promote more fruitful research efforts in this area.

The text of this tutorial is designed for engineers, programmers, graduate students, educators, and those involved in the research, development, and application of special-purpose computers for solving problems in AI. It is a tutorial in nature and provides a state-of-the-art survey. It is expected that this tutorial will serve as a guide for beginners, as well as a major reference for all professionals in computer engineering and AI. It is hoped that after reading the text readers will be able to understand the design issues, realize the limitations, write better programs, promote more research in this area, and trigger further creative designs for architectures and languages in AI.

The text is organized into nine chapters. A tutorial guide is provided at the beginning of each chapter to explain the underlying concepts and to annotate an up-to-date reference list. Each chapter, excluding Chapter 9, includes some reprinted articles to explain the design issues and to exemplify some recent developments.

Chapter 1 provides a general introduction to the problems and objectives in AI.

Chapter 2 discusses the motivations and design issues in designing specialized architectures for AI applications.

Chapter 3 explains the various declarative languages and programming techniques in AI, their developments, and their relationships to each other.

Chapter 4 focuses on the microlevel hardware architectures that include pattern matching, intelligent memories, and hardware unification units.

Chapter 5 investigates the macrolevel hardware architectures. Issues and designs on knowledge database machines and parallel searching are presented.

Chapters 6, 7, and 8 explore hardware architectures on the system level. Chapters 6 and 7 highlight architectures in general, while Chapter 8 presents architectures directly related to the Fifth-Generation Computer-System Projects in various countries.

Chapter 6 focuses on architectures for functional and lambda-based languages.

Chapter 7 concentrates on logic and knowledge oriented architectures. The architectures are divided into three classes and are directed toward logic programs, production systems, and semantic nets and distributed problem solving.

Chapter 8 discusses architectures of the Fifth-Generation Computer-System projects in England, the European Commission, Japan, and the United States.

Last, Chapter 9 exemplifies some architectures designed for applications in AI including speech recognition, image understanding, and computer chess.

We would like to take this opportunity to thank all the authors who have contributed to this tutorial text. We apologize to those authors whose valuable papers could not be included owing to page limitations and we also apologize to those authors whose shorter papers were selected in place of their longer and more comprehensive versions because of the page limitations. We tried to be as complete as possible in compiling the reference lists, which were based on contributions and accessibility of the work. We deeply regret any omissions and hope that they will be made known to us for use in future revisions. We would like to acknowledge the support of CIDMAC, a research unit of Purdue University, sponsored by Purdue; Cincinnati Milicron Corporation; Control Data Corporation; Cummins Engine Company; Ransburg Corporation; TRW; and the National Science Foundation grant DMC-85-19649 in preparing this tutorial text. Finally, we would like to thank sincerely Professor Chuan-lin Wu and the reviewers for their helpful criticisms, suggestions, and encouragement; Jo Johnson for her patient secretarial help; and Margaret Brown and her staff of the IEEE Computer Society Press for editing the text.

Benjamin W. Wah and Guo-Jie Li
University of Illinois, Urbana-Champaign

Table of Contents

Preface	iii
Chapter 1: Introduction to Artificial Intelligence Applications	1
History of LISP	5
<i>J. McCarthy (SIGPLAN Notices, August 1978, pages 217-222)</i>	
Artificial Intelligence: Underlying Assumptions and Basic Objectives	12
<i>N. Cercone and G. McCalla (Journal of the American Society for Information Science, September 1984, pages 280-290)</i>	
Chapter 2: Introduction to Computer Architectures for Artificial Intelligence	23
A Preliminary Survey of Artificial Intelligence Machines	26
<i>H. Boley (SIGART Newsletter, July 1980, pages 21-28)</i>	
Computing Facilities for AI: A Survey of Present and Near-Future Options	34
<i>S. Fahlman (AI Magazine, Winter 1980-1981, pages 16-23)</i>	
The New Generation of Computer Architecture	42
<i>P.C. Treleaven (Proceedings of the 10th Annual International Symposium on Computer Architecture, June 1983, pages 402-409)</i>	
Design Issues in Parallel Architectures for Artificial Intelligence	50
<i>C. Hewitt and H. Lieberman (Proceedings of COMPCON S'84, 1984, pages 418-423)</i>	
Chapter 3: Artificial Intelligence Languages and Programming	57
Function-Level Computing	62
<i>J. Backus (Spectrum, August 1982, pages 22-27)</i>	
Predicate Logic as Programming Language	68
<i>R. Kowalski (IFIP Information Processing 74, 1974, pages 569-574)</i>	
Power Tools for Programmers	74
<i>B. Sheil (Datamation, February 1983, pages 131-144)</i>	
Object Oriented Programming	81
<i>T. Rentsch (SIGPLAN Notices, September 1982, pages 51-57)</i>	
Notes on Systems Programming in PARLOG	88
<i>K. Clark and S. Gregory (Proceedings of the International Conference on Fifth- Generation Computer Systems, 1984, pages 299-306)</i>	
Object Oriented Programming in Concurrent Prolog	96
<i>E.Y. Shapiro and A. Takeuchi (New Generation Computing, Vol. 1, 1983, pages 25-48)</i>	
Chapter 4: Microlevel Architecture	121
Hardware Algorithms for Nonnumeric Computation	124
<i>A. Mukhopadhyay* (IEEE Transactions on Computers, June 1979, pages 384-394)</i>	
New Classes for Parallel Complexity: A Study of Unification and Other Complete Problems for P	135
<i>J.S. Vitter and R.A. Simons (IEEE Transactions on Computers, May 1986, pages 403-418)</i>	

*Professor Amar Mukhopadhyay has changed his name to Amar Mukherjee. His new address is Dept. of Computer Science, University of Central Florida, Orlando, FL 32751.

Chapter 5: Macrolevel Architectures	151
Performance Analysis of Alternative Database Machine Architectures	155
<i>P.B. Hawthorn and D.J. DeWitt (IEEE Transactions on Software Engineering, January 1982, pages 61-75)</i>	
Intelligent Assistants for Knowledge and Information Resources Management	170
<i>C.H. Kellogg (Proceedings of the 8th International Joint Conference on Artificial Intelligence, 1983, pages 170-172)</i>	
Multiprocessing of Combinatorial Search Problems	173
<i>B.W. Wah, G.-J. Li, and C.F. Yu (Computer, June 1985, pages 93-108)</i>	
Garbage Collection of Linked Data Structures	189
<i>J. Cohen (Computing Surveys, September 1981, pages 341-367)</i>	
 Chapter 6: Functional-Programming- Oriented Architectures	 217
A Survey of Proposed Architectures for the Execution of Functional Languages	223
<i>S.R. Vegdahl (IEEE Transactions on Computers, December 1984, pages 1050-1071)</i>	
The Lisp Machine	245
<i>A. Bawden, R. Greenblatt, J. Holloway, T. Knight, D. Moon, and D. Weinreb (Artificial Intelligence: An MIT Perspective, Vol. 1, P.H. Winston and R.H. Brown (eds.), 1979, pages 343-373)</i>	
Lisp and Prolog Machines Are Proliferating	266
<i>T. Manuel (Electronics, November 1983, pages 132-137)</i>	
Lisp Machines Come out of the Lab	272
<i>M. Creeger (Computer Design, November 1983, pages 207-216)</i>	
Architecture of the Symbolics 3600	279
<i>D.A. Moon (Proceedings of the 12th Annual International Symposium on Computer Architecture, 1985, pages 76-83)</i>	
ALPHA: A High-Performance LISP Machine Equipped with a New Stack Structure and Garbage Collection System	287
<i>H. Hayashi, A. Hattori, and H. Akimoto (Proceedings of the 10th Annual International Symposium on Computer Architecture, 1983, pages 342-348)</i>	
Scheme-79—Lisp on a Chip	294
<i>G.J. Sussman, J. Holloway, G.L. Steel, Jr., and A. Bell (Computer, July 1981, pages 10-21)</i>	
EM-3: A LISP-Based Data-Driven Machine	305
<i>Y. Yamaguchi, K. Toda, J. Herath, and T. Yuba (Proceedings of the International Conference on Fifth-Generation Computer Systems, 1984, pages 524-532)</i>	
Architecture of SOAR: Smalltalk on a RISC	314
<i>D. Ungar, R. Blau, P. Foley, D. Samples, and D. Patterson (Proceedings of the 11th Annual International Symposium on Computer Architecture, 1984, pages 188-197)</i>	
Making Parallel Computation Simple: The FFP Machine	324
<i>G. Magó (Proceedings of COMPCON S'85, 1985, pages 424-428)</i>	
Rediflow Multiprocessing	329
<i>R.M. Keller, F.C.H. Lin, and J. Tanaka (Proceedings of COMPCON S'84, 1984, pages 410-417)</i>	
A Multi-Microprocessor System for Concurrent LISP	337
<i>S. Sugimoto, K. Agusa, K. Tabata and Y. Ohno (Proceedings of the International Conference on Parallel Processing, 1983, pages 135-143)</i>	

Chapter 7: Logic and Knowledge Oriented Architectures	347
7.A: Logic Programs	
Execution of Bagof on the OR-Parallel Token Machine	352
<i>A. Ciepielewski and S. Haridi (Proceedings of the International Conference on Fifth-Generation Computer Systems, 1984, pages 551-560)</i>	
Restricted AND-Parallelism	361
<i>D. DeGroot (Proceedings of the International Conference on Fifth-Generation Computer Systems, 1984, pages 471-478)</i>	
AND Parallelism and Nondeterminism in Logic Programs	369
<i>J.S. Conery and D.F. Kibler (New Generation Computing, Vol. 3, 1985, pages 43- 70)</i>	
MANIP-2: A Multicomputer Architecture for Evaluating Logic Programs	392
<i>G.-J. Li and B.W. Wah (Proceedings of the International Conference on Parallel Processing, 1985, pages 123-130)</i>	
Towards a Pipelined Prolog Processor	400
<i>E. Tick and D.H.D. Warren (New Generation Computing, Vol. 2, 1984, pages 323-345)</i>	
Status and Performance of the Zmob Parallel Processing System	423
<i>M. Weiser, S. Kogge, M. McElvany, R. Pierson, R. Post and A. Thareja (Proceedings of COMPCON S'85, 1985, pages 71-73)</i>	
Aquarius—A High Performance Computing System for Symbolic/Numeric Applications	427
<i>A.M. Despain and Y.N. Patt (Proceedings of COMPCON S'85, 1985, pages 376-382)</i>	
Dataflow Computing and Eager and Lazy Evaluations	434
<i>M. Amamiya and R. Hasegawa (New Generation Computing, Vol. 2, 1984, pages 105-129)</i>	
The FAIM-1 Symbolic Multiprocessing System	455
<i>A.L. Davis and S.V. Robison (Proceedings of COMPCON S'85, 1985, pages 370-375)</i>	
7.B: Production Systems	
Initial Assessment of Architectures for Production Systems	461
<i>C. Forgy, A. Gupta, A. Newell, and R. Wedig (Proceedings of the National Conference on Artificial Intelligence, August 1984, pages 116-120)</i>	
Mapping Production Systems into Multiprocessors	466
<i>M.F.M. Tenorio and D.I. Moldovan (Proceedings of the International Conference on Parallel Processing, 1985, pages 56-62)</i>	
DADO: A Parallel Processor for Expert Systems	473
<i>S.J. Stolfo and D.P. Miranker (Proceedings of the International Conference on Parallel Processing, 1984, pages 74-82)</i>	
7.C: Semantic Nets and Distributed Problem Solving	
Design Sketch for a Million-Element NETL Machine	482
<i>S.E. Fahlman (Proceedings of the 1st Annual National Conference on Artificial Intelligence, August 1980, pages 249-252)</i>	

Massively Parallel Architectures for AI: NETL, Thistle and Boltzmann Machines	486
<i>S.E. Fahlman and G.E. Hinton (Proceedings of the Annual National Conference on Artificial Intelligence, 1983, pages 109-113)</i>	
The Connection Machine: A Computer Architecture Based on Cellular Automata	491
<i>W.D. Hillis (Physica, 1984, pages 213-228)</i>	
The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network	507
<i>D.D. Corkill and V.R. Lesser (Proceedings of the 8th International Joint Conference on Artificial Intelligence, 1983, pages 748-756)</i>	
A Framework for Distributed Problem Solving	516
<i>R.G. Smith (Proceedings of the 6th International Joint Conference on Artificial Intelligence, 1979, pages 836-841)</i>	
The LOCO Approach to Distributed Task Allocation in <i>Aida</i> by Verdi	522
<i>V.M. Milutinovic, J.J. Crnkovic, L.Y. Chang, and H.J. Siegel (Proceedings of the 5th International Conference on Distributed Computing Systems, 1985, pages 359-368)</i>	
Chapter 8: Fifth-Generation Computer System	533
Fifth-Generation Computer Systems:	
A Japanese Project	536
<i>T. Moto-oka and H.S. Stone (Computer, March 1984, pages 6-13)</i>	
Hardware Design and Implementation of the Personal Sequential Inference Machine (PSI)	544
<i>K. Taki, M. Yokota, A. Yamamoto, H. Nishikawa, S. Uchida, H. Nakashima, and A. Mitsuishi (Proceedings of the International Conference on Fifth-Generation Computer Systems, 1984, pages 398-409)</i>	
Highly Parallel Inference Engine PIE— Goal Rewriting Model and Machine Architecture	556
<i>A. Goto, H. Tanaka, and T. Moto-oka (New Generation Computing, Vol. 2, 1984, pages 37-58)</i>	
Data-Flow Based Execution Mechanisms of Parallel and Concurrent Prolog	577
<i>N. Ito, H. Shimizu, M. Kishi, E. Kuno, and K. Rokusawa (New Generation Computing, Vol. 3, 1985, pages 15-41)</i>	
A Relational Database Machine with Large Semiconductor Disk and Hardware Relational Algebra Processor	603
<i>S. Shibayama, T. Kakuta, N. Miyazaki, H. Yokota, and K. Murakami (New Generation Computing, Vol. 2, 1984, pages 131-155)</i>	
New Computer Breed Uses Transputers for Parallel Processing	627
<i>K. Smith (Electronics, February 1983, pages 67-68)</i>	
ALICE and the Parallel Evaluation of Logic Programs	629
<i>J. Darlington and M. Reeve</i>	
Chapter 9: Computer Systems for Specialized Artificial Intelligence Applications	645