Algorithmically Specialized Parallel Computers

Edited by

Lawrence Snyder, Leah H. Jamieson,
Dennis B. Gannon, and Howard Jay Siegel

ALGORITHMICALLY SPECIALIZED PARALLEL COMPUTERS

EDITED BY

LAWRENCE SNYDER
Department of Computer Science
University of Washington
Seattle, Washington

LEAH H. JAMIESON
School of Electrical Engineering
Purdue University
West Lafayette, Indiana

DENNIS B. GANNON
Department of Computer Sciences
Purdue University
West Lafayette, Indiana

HOWARD JAY SIEGEL
School of Electrical Engineering
Purdue University
West Lafayette, Indiana

1985

ACADEMIC PRESS, INC.
(Harcourt Brace Jovanovich, Publishers)
Orlando San Diego New York London
Toronto Montreal Sydney Tokyo

ACADEMIC PRESS, INC.
Orlando, Florida 32887

United Kingdom Edition published by
ACADEMIC PRESS INC. (LONDON) LTD.
24–28 Oval Road, London NW1 7DX

Library of Congress Catalog Card Number: 85-71568

ISBN 0-12-654130-2

PRINTED IN THE UNITED STATES OF AMERICA

85 86 87 88 9 8 7 6 5 4 3 2 1

ALGORITHMICALLY SPECIALIZED PARALLEL COMPUTERS

ACADEMIC PRESS RAPID MANUSCRIPT REPRODUCTION

CONTRIBUTORS

Numbers in parentheses indicate the pages on which the authors' contributions begin.

GREGORY D. ABRAM (79), Computer Science Department, University of North Carolina, Chapel Hill, North Carolina 27514

T. P. BARNWELL III (119), School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

KENNETH E. BATCHER (59), Digital Technology Department, Goodyear Aerospace Corporation, Akron, Ohio 44315

EDWARD C. BRONSON (139), School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907

B. L. BUZBEE (223), Computing Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

MICHAEL J. CAREY[1] (27, 37), Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720

JANICE E. CUNY[2] (7), Computer Sciences Department, Purdue University, West Lafayette, Indiana 47907

RENATO DE MORI (149), Department of Computer Science, Concordia University, Montreal, Quebec, Canada H3G 1M8

YASUNORI DOHI[3] (47), Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213

ROBERT J. DOUGLASS (105), Computer User Services, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

ALLAN L. FISHER (47), Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213

[1]Present address: Computer Science Department, University of Wisconsin–Madison, Madison, Wisconsin 53706.

[2]Present address: Computer and Information Science Department, Graduate Research Center, University of Massachusetts, Amherst, Massachusetts 01003.

[3]Present address: Department of Computer Engineering, Yokohama National University, 156 Tokwadai, Hodgaya, Kanheawa 240, Japan.

HENRY FUCHS (79), Computer Science Department, University of North Carolina, Chapel Hill, North Carolina 27514

DANIEL D. GAJSKI (249), Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, Illinois 61801

R. J. GAL-EZER[4] (201), Department of Electrical Engineering–Systems, University of Southern California, Los Angeles, California 90089

PAUL M. HANSEN (27), Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720

DON HELLER[5] (233), Computer Science Department, Pennsylvania State University, University Park, Pennsylvania 16802

C. J. M. HODGES (119), School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

LEAH H. JAMIESON (139), School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907

JAMES T. KUEHN (69), School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907

H. T. KUNG (47), Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213

S. Y. KUNG (201), Department of Electrical Engineering–Systems, University of Southern California, Los Angeles, California 90089

R. M. LEA (159), Electrical Engineering and Electronics, Brunel University, Uxbridge, Middlesex UB8 3PH, England

G. JACK LIPOVSKI (247), Department of Electrical Engineering, The University of Texas at Austin, Austin, Texas 78712

DAVID D. LOENDORF (213), Department of Mechanical Engineering, and Department of Computer Science, Duke University, Durham, North Carolina 27706

LOUIS M. MONIER[6] (47), Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213

AHMED SAMEH (191), Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, Illinois 61801

JOHN E. SAVAGE (17), Department of Computer Science, Brown University, Providence, Rhode Island 02912

[4]Present address: Haifa, Israel.
[5]Present address: Physics and Computer Science Department, Bellaire Research Center, Shell Development Company, Houston, Texas 77001.
[6]Present address: Xerox PARC, Palo Alto, California 94304.

HOWARD JAY SIEGEL (69), School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907

HARVEY F. SILVERMAN (129), Laboratory for Engineering in Man/Machine Systems, Division of Engineering, Brown University, Providence, Rhode Island 02912

LAWRENCE SNYDER[7] (7), Computer Sciences Department, Purdue University, West Lafayette, Indiana 47907

STANLEY R. STERNBERG[8] (169), CytoSystems Corporation, Ann Arbor, Michigan, and Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, Michigan 48109

HAROLD S. STONE[9] (89, 251), Electrical and Computer Engineering Department, University of Massachusetts, Amherst, Massachusetts 01003

CLARK D. THOMPSON (27, 37), Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720

LEONARD UHR (177), Department of Computer Sciences, University of Wisconsin–Madison, Madison, Wisconsin 53706

WAYNE T. WILNER (95), Bell Telephone Laboratories, Holmdel, New Jersey 07733

[7]Present address: Department of Computer Science, University of Washington, Seattle, Washington 98195.

[8]Present address: Machine Vision International, Ann Arbor, Michigan 48104.

[9]Present address: IBM Corporation, Yorktown Heights, New York 10598.

PREFACE

The concept of an algorithmically specialized computer, as defined in Chapter 1, has been fermenting in the computer science/engineering research community for the past few years. Bernard Chern of the National Science Foundation grasped its significance and suggested to us the idea of holding a workshop to solidify the concept and to explore some of its characteristics. The result was the Purdue Workshop on Algorithmically-specialized Computer Organizations, held in West Lafayette, Indiana, 29 September through 1 October 1982. This book is based on that meeting.

The workshop was characterized by spirited and stimulating discussions. In order to relate (our interpretation of) the content of those discussions, we have prepared an introductory chapter on the topic of algorithmically specialized computers, chapter introductions, and a synopsis of the panel discussion, Does General Purpose Mean Good for Nothing (in Particular)? Of course we are exercising our editorial license here, but we have reviewed the taped transcript of the panel in an effort to be faithful to the thrusts of the discussion; we apologize if we have misconstrued anyone's comments.

It is a pleasure to thank Bernard Chern of the National Science Foundation for inspiring us to hold the workshop in the first place. We are also grateful for the support of Richard L. Lau and David W. Mizell of the Office of Naval Research. The Workshop was funded by NSF Grant ECS-8206181 and ONR Contract N00014-81-K-0360.

One of the significant features of the workshop was the extensive discussion. Much of the stimulus for this came from the session chairs: Jon Bentley, Jack Lipovski, Franco Preparata, John Savage, Leonard Uhr, and Robert Voigt. We thank them for this invaluable contribution.

Julie K. Hanover, without whom the workshop would not have been possible, deserves the highest praise and sincerest thanks for attending to all of the organizational details. We would also like to thank Carol Edmundson, Pat Kerkhoff, and Mike Hope for their help.

L.S., L.H.J., D.B.G., H.J.S.

CONTENTS

CHAPTER 1

ALGORITHMICALLY SPECIALIZED COMPUTERS

The architectural means of speeding up particular computations has traditionally been to build faster general purpose computers. But speeding up general purpose computers is becoming more and more difficult and it is commonly believed that, for general purpose sequential computers at least, we are approaching the speed limit. It is time to take advantage of more information.

Algorithmically specialized computers are machines whose architecture provides efficient execution for a class of problems by exploiting characteristics of the problem solving method. The concept carries with it the implication that some efficiency, or perhaps even functionality, is lost for problems not in the class, since otherwise there would be no reason to specialize for the particular class and the computer would be an improved architecture for general purpose computing. What is gained by algorithmic specialization is improved performance on a class of problems. What is lost is generality: the machine is only "good" for that class. The thesis, then, is that algorithmic specialization represents a trade-off in which improved performance is gained at the expense of generality.

The last paragraph provided a definition for algorithmically specialized computers and identified the two protagonists of our plot, but further amplification is required.

Notice that the costs and benefits of algorithmic specialization are defined with respect to a particular set S of problems. Although it is not illogical to take S to be a singleton set consisting of only one problem, or the universal set consisting of all effectively computable problems, neither case is terribly interesting. When S is a singleton set then all generality has been traded for speed and the algorithmically specialized computer is a one function, non-programmable circuit that does not even qualify as a computer. When S is the universal set, then no generality is being traded and any performance improvement amounts to an improvement in general purpose computation. Excluding these two extreme cases does not, however, solve the problem of selecting a suitable problem domain.

Often one thinks of the problem domain as being determined by an application area, e.g., S = {image processing problems} or S = {numerical linear algebra}. It may be the case that a number of problems from a given application area will often exhibit similar algorithmic properties. On the other hand, there is apparently no necessary requirement that the algorithms solving a naturally occuring set of problems will have anything in common, so it may be the case that not one property, but a set of algorithmic properties, is needed to span an entire application area. Since it is the common properties of *algorithms* that we wish to exploit in our architectures, we expect the problem domains to be expressed in terms of algorithmic properties or sets of algorithmic properties.

It is important to stress that the algorithmic properties used to characterize the problem domain must be "structural," i.e., reflect a fundamental characteristic of the problem solving method. An example of such a fundamental property is the algorithm's communication structure. The pattern of data movement is a characteristic for which an architecture can be optimized and for which the optimization likely engenders a corresponding improvement in performance and a loss of generality. Other features of algorithms that can often be exploited in the design of a special purpose architecture include data formats and structures, data set size, types of operations, and patterns of control flow.

In addition to algorithmic characteristics, there are often important constraints imposed on the system design by the structure of the applications area. These include processing speed requirements, power consumption, physical size, the accuracy of computed answers, and the cost and cost effectiveness. Although these are attributes of the task or problem domain rather than of the algorithm, they will also play a roll in the design of specialized computer architectures.

The problem, of course, is not so much in defining the algorithm class S as it is in finding an architectural optimization to exploit algorithmic features. The communication structure property mentioned above might, as a general rule, be supported by an architecture with dedicated data paths. Another obvious generic optimization is the use of parallel processing on the independent subcomputations of an algorithm. In general, however, the problem of how best to exploit an algorithmic property is a difficult one, and it is further complicated by the possibility of having to balance conflicting optimizations. The challenge in designing algorithmically-specialized computers, then, is to find algorithmic properties that are amenable to architectural specialization.

The papers in this volume, having been selected to maximize the diversity of viewpoint, treat many different aspects of algorithmic specialization. A substantial fraction of the papers are directly motivated by an application area such as speech understanding or numerical computation. Some papers are motivated by the potential benefits of VLSI technology. Others describe machines which are specialized to particular kinds of data motion. In addition there are papers describing theoretical models, software issues, and automated implementation techniques. Although there is much progress reported here, these papers represent early work in a newly identified research area. Much remains to be explored along the algorithms/architectures frontier.

CHAPTER 2

ALGORITHMIC SPECIALIZATION USING VLSI

Very large scale integration provides the technological impetus for algorithmic specialization, because high density, low cost chips make the implementation feasible. The lower the hardware costs become the wider is the range of problems for which specialized hardware is economically justified. Although the availablity of VLSI technology may make it practical to have specialized systems, the medium is not perfectly malleable. Many problems must be solved before an algorithmically specialized processor is implemented in VLSI. For example, VLSI technology favors planar architectures with much geometric locality, but these characteristics have not typically been considered. Moreover, they are often difficult to achieve.

In this chapter we consider VLSI related topics ranging from models and implementation techniques, through specific algorithms, to a programmable system for a family of algorithms.

In the first paper we are given a model of computation to be used as a tool for developing algorithmically specialized systems. The model abstracts different interprocessor communication strategies. The advancement that Cuny and Snyder have made is that their one model abstracts many communication protocols – synchronous, data driven, etc. – within one consistent framework: The model has a parameter which defines the protocol. Fair comparisons can thus be made between communication mechanisms. Examples and preliminary results are given.

Savage presents in the second paper of the chapter a comparison of general VLSI implementation schemes based on inputs of Boolean equations. The schemes considered are PLAs, Weinberger arrays, and SLAP. SLAP is a new system that is described in the paper and was demonstrated at the Workshop. From the demonstration it was clear that SLAP is a convenient flexible system. The paper shows SLAP to be more area-efficient for certain classes of problems than the other two approaches.

The next two papers give specific designs for the VLSI implementation of algorithmically specialized processors supporting data base operations. The sorting paper by Carey, Hansen, and Thompson gives