

*Parallel
Supercomputing
in
SIMD Architectures*

R. Michael Hord

Parallel Supercomputing in SIMD Architectures

江苏工业学院图书馆
藏书章

R. Michael Hord

Advanced Technology Laboratory

General Electric Company

Moorestown, New Jersey



CRC Press

Boca Raton Ann Arbor Boston

Acquiring Editor: Russ Hall
Production Director: Sandy Pearlman
Coordinating Editor: Suzanne Lassandro
Cover Design: Chris Pearl

Library of Congress Cataloging-in-Publication Data

Hord, R. Michael, 1940—

Parallel supercomputing in SIMD architectures/author, R. Michael Hord.

p. cm.

Includes bibliographical references.

ISBN 0-8493-4271-6

1. Parallel processing (Electronic computers). 2. Computer architecture. I. Title.

QA76.5.H675 1990

004'.35—dc20

89-71253

CIP

This journal represents information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Every reasonable effort has been made to give reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

All rights reserved. This book, or any parts thereof, may not be reproduced in any form without written consent from the publisher.

Direct all inquiries to CRC Press, Inc., 2000 Corporate Blvd., N.W., Boca Raton, Florida 33431.

© 1990 by CRC Press, Inc.

International Standard Book Number 0-8493-4271-6

Library of Congress Card Number 89-71253

Printed in the United States

FOREWORD

We live in a golden age of computer architecture innovation. The advent of parallel processing has produced a flood of architecture paradigms, some more successful than others.

The most well known taxonomy for parallel computers was proposed by M. J. Flynn in 1966. It is based on the multiplicity of the instruction and data streams, which identifies four classes of computers: (1) Single instruction stream single data stream (SISD) computers correspond to regular Von Neumann non-parallel computers that can execute one instruction at a time on one data item at a time. (2) Single instruction multiple data (SIMD) computers are based on a central program controller that drives the program flow and a set of processing elements that all execute the instructions from the central controller on their individual data items. (3) Multiple instruction single data (MISD) computers are based on a pipeline principle. A given data item passes from processor to processor and is acted on differently at each stage of this assembly line. (4) Multiple instruction multiple data (MIMD) computers consist of a number of SISD computers configured to communicate among themselves in the course of a program.

Examples of recent successful MISD computers include the Cytocomputer and the PIPE. Instances of the MIMD category are too numerous to list but include NCUBE, iPSC, PASM, WARP, ZMOB, transputer based machines, Encore, Sequent, Alliant and Datacube.

In this book we explore the SIMD category. After a chapter of background that includes a brief discussion of the PEPE, STARAN and SOLOMON early machines, seven important SIMD machines are examined in depth: Illiac IV, MPP, DAP, GAPP, Connection Machine, GAM and CLIP4. Other significant SIMD machines not covered here include the Adaptive Array Processor, GRID, Non Von, Hughes Wafer Stack, GF11, PAPIA and CLIP7.

All of the machines in these categories have distinct characteristics, differing in connectivity, word size, memory management, and other major parameters. But even then the field of parallel processing is not fully mapped because there are whole classes of architectures that do not neatly fit the Flynn taxonomy. What is one to do with neural network computers, logic enhanced memories, data flow architectures and heterogeneous machines? And Very Long Instruction Word machines? And optical computers? In the face of this abundance, the premise of this book is simple. SIMD computers are a successful category of parallel processors because they provide an effective balance between complexity and simplicity. They are rich in algorithmic opportunity yet conceptually clean and easy to understand. The material presented attempts to offer a rebuttal to both camps of SIMD critics: those that claim the range of applicability is narrow because many problems are not sufficiently parallel to use so many processors efficiently and those that claim that having all of the parallel processors execute the same instruction is too restricted to embody the complexities of real world problems. I say

instead that the range of applicability is enormous and hence broad enough to make good general computing and economic sense. And I say that the conceptual clarity of the architecture provides an ease of programming that keeps software costs within reason. The contents of the book are intended to allow the reader to judge these issues for himself.

Much of the material included here is based on and adapted from work previously published in government reports, journal articles, books and vendor literature. A listing of the major sources is provided following the text.

PREFACE

Parallel Supercomputing in SIMD Architectures is a survey book providing a thorough review of Single-Instruction-Multiple-Data machines, a type of parallel processing computer that has grown to importance in recent years. It was written to describe this technology in depth including the architectural concept, its history, a variety of hardware implementations, major programming languages, algorithmic methods, representative applications, and an assessment of benefits and drawbacks.

The book is intended for a wide range of readers. Computer professionals will find sufficient detail to incorporate much of this material into their own endeavors. Program managers and applications system designers may find the solution to their requirements for high computational performance at an affordable cost. Scientists and engineers will find sufficient processing speed to make interactive simulation a practical adjunct to theory and experiment. Students will find a case study of an emerging and maturing technology. The general reader is afforded the opportunity to appreciate the power of advanced computing and some of the ramifications of this growing capability.

Although there are numerous books on parallel processing, this is the first volume devoted entirely to the massively parallel machines of the SIMD class. The reader already familiar with low order parallel processing will discover a different philosophy of parallelism—the data parallel paradigm instead of the more familiar program parallel scheme.

The contents are organized into nine chapters, rich with illustrations and tables. The first two provide introduction and background covering fundamental concepts and a description of early SIMD computers. Chapters 3 through 8 each address specific machines from the first SIMD supercomputer (Illiac IV) through several contemporary designs to some example research computers. The final chapter provides commentary and lessons learned. Because the test of any technology is what it can do, diverse applications are incorporated throughout, leading step by step to increasingly ambitious examples.

For Susan

THE AUTHOR

R. Michael Hord is presently Manager of the Processing Applications Laboratory at the General Electric Advanced Technologies Laboratories, Moorestown, New Jersey. In this capacity he directs research and development activities employing the Connection Machine, the Butterfly, and the Warp advanced architecture computers. The current emphasis is on acoustic signal analysis, military data/information systems, and future architectures.

Until the end of 1989, Mr. Hord was Head of the Advanced Development Center at MRJ, Inc., Oakton, Virginia, where he directed diverse computer applications using parallel architectures including two Connection Machines. Application areas included image processing, signal processing, electromagnetic scattering, operations research and artificial intelligence. Mr. Hord joined MRJ in 1984 where he also directed the corporate research and development program.

For 5 years (1980 to 1984) Mr. Hord was the Director of Space Systems for General Research Corporation. Under contract to NASA and the Air Force, he and his staff assessed technology readiness for future space systems and performed applications analysis for innovative on-board processor architectures.

SIMD parallel processing was the focus of his efforts as the Manager of Applications Development for the Institute for Advanced Computation (IAC). IAC was the joint DARPA/NASA sponsored organization responsible for the development of the Illiac IV parallel supercomputer at Ames Research Center. Projects included computational fluid dynamics, seismic simulation, digital cartography, linear programming, climate modeling and diverse image and signal processing applications.

Prior positions at Earth Satellite Corporation, Itek Corporation and Technology Incorporated were devoted to the development of computationally intensive applications such as optical system design and natural resource management.

Mr. Hord's five prior books and scores of papers address advanced parallel computing, digital image processing and space technology. He has long been active in the applied imagery pattern recognition community, has been an IEEE Distinguished Visitor, and is a frequent guest lecturer at several universities. His B.S. in physics was granted by Notre Dame University in 1962 and in 1966 he earned an M.S. in physics from the University of Maryland.

Eight men dominate the history of SIMD computer architectures. Their names are listed here to acknowledge their pioneering efforts:

S. H. Unger	First to propose spatially organized architectures (1958)
Daniel Slotnick	SOLOMON and Illiac IV
David Schaefer	TZE, MPP, GAM
Kenneth Batcher	MPP
W. Holsztynski	GAPP
Dennis Parkinson	DAP
Michael J. B. Duff	CLIP4
Danny Hillis	Connection Machine

BRIEF CONTENTS

Chapter 1	
Introduction	1
Chapter 2	
Background	5
2.1 SIMD vs. MIMD	5
2.2 Fine Grain vs. Coarse Grain	6
2.3 Connectivity	7
2.4 Early Machines	7
2.4.1 PEPE	7
2.4.2 STARAN	10
2.4.3 SOLOMON I	11
Chapter 3	
Illiac IV — The First SIMD Supercomputer	17
3.1 History	17
3.2 The Illiac IV System	30
3.3 The CFD Language	39
3.4 Language Review	48
3.5 Performance	56
3.6 Seismic Analysis Application	56
3.7 Landsat Analysis	62
3.8 Synthetic Aperture Radar	65
3.9 Fast Fourier Transform	69
3.10 Linear Programming Image Enhancement	70
3.11 Comments on Some Case Studies	75
3.12 The Effects of the Illiac IV System on Computing Technology	79
Chapter 4	
The Massively Parallel Processor (MPP)	85
4.1 The MPP Design	85
4.2 Parallel Forth Language	90
4.3 Parallel Pascal Design	105
4.4 Ising Spin Exchange Simulation	121
4.5 Stereo Analysis	125
4.6 Sort	132
Chapter 5	
The Distributed Array of Processors (DAP)	143
5.1 The DAP Design	143
5.2 Parallel Data Transforms	156

5.3	Solution of a Large System of Equations	159
5.4	Image Understanding	164

Chapter 6

The Geometric Arithmetic Parallel Processor (GAPP)	181
6.1 The GAPP Design	181
6.2 GAPP Programming	186
6.3 GAPP System	199

Chapter 7

The Connection Machine (CM)	205
7.1 The CM System	206
7.1.1 Overview	208
7.1.2 The Connection Machine	209
7.1.3 The DataVault	219
7.2 Programming	221
7.2.1 The *Lisp Language	223
7.2.2 The C* Language	227
7.2.3 The Paris Language	233
7.3 Applications	236
7.3.1 Physics	237
7.3.2 Operations Research	250
7.3.3 Image Processing	268
7.3.4 Miscellaneous	273

Chapter 8

Research SIMD Computers	301
8.1 GAM	301
8.2 CLIP4	308

Chapter 9

Commentary	329
-------------------------	------------

Appendix

Fault Tolerant SIMD	347
----------------------------------	------------

Acknowledgments	365
------------------------------	------------

Index	367
--------------------	------------

TABLE OF CONTENTS

Chapter 1	
Introduction	1
Chapter 2	
Background	5
2.1. SIMD vs. MIMD	5
2.2. Fine Grain vs. Course Grain	6
2.3. Connectivity	7
2.4. Early Machines	7
2.4.1. PEPE (Parallel Element Processing Ensemble)	7
2.4.2. STARAN	10
2.4.3. SOLOMON I	11
Chapter 3	
Illiacy IV — The First SIMD Supercomputer	17
3.1. History	17
3.1.1. The Design Concept	18
3.1.2. Implementation Difficulties	21
3.2. I4 System	30
3.3. The CFD Language	39
3.4. Language Review	48
3.4.1. PE Variables	50
3.4.2. PE Variables Memory Allocation	51
3.4.3. Array Addressing	51
3.4.4. Routing	52
3.5. Performance	56
3.6. Seismic Analysis Application: A Three-Dimensional Finite Difference Code for Seismic Analysis on the Illiacy IV Parallel Processor	56
3.6.1. Empirical Evidence	56
3.6.2. The TRES Computer Program	58
3.6.3. Implementation on the Illiacy IV	60
3.6.4. Results Obtained with the I4TRES Program	62
3.7. Landsat	62
3.7.1. Landsat Data Analysis	63
3.7.2. Illiacy IV Implementation	63
3.8. Digital Processing of Synthetic Aperture Radar Data on Illiacy IV	65
3.8.1. Synthetic Aperture Radar Concepts	65
3.9. Fast Fourier Transform	69
3.10. Linear Programming Image Enhancement	70
3.10.1. Statement of the Problem	70

3.10.2.	Linear Programming Approach	73
3.10.3.	Practical Issues	73
3.11.	Comments on Some Case Studies	75
3.11.1.	Sparse Matrix Multiply	75
3.11.2.	A Model for Disaster	76
3.11.3.	Monte Carlo Methods on the Illiac	78
3.11.4.	Conclusions	79
3.12.	The Effects of the Illiac IV System on Computing Technology	79
3.12.1.	Component and Manufacturing Technology	80
3.12.1.1.	Major Impetus to ECL Development	80
3.12.1.2.	Test Bed for Design Automation	80
3.12.1.3.	New Contribution to Logic Circuitry	80
3.12.1.4.	First Significant Use of Semiconductor Memory	80
3.12.1.5.	Definitive Contribution to Interconnection Technology	80
3.12.1.6.	A Major Milestone in Multilayer PC Cards	81
3.12.2.	Machine Architecture	81
3.12.2.1.	Definitive Demonstration of Array Approach to Computation	81
3.12.2.2.	Synchronous Control to Focus Research on Efficiency of Computation	81
3.12.2.3.	First Large Scale Computer to be Microprogrammed	81
3.12.2.4.	Synchronizaton of Independent Disk Drives	81
3.12.2.5.	Exhaustive Simulation as a Realistic Diagnostic Tool	81
3.12.2.6.	Test Bed for Future Machines	82
3.12.3.	System Architecture	82
3.12.4.	Applications	83
3.12.4.1.	New Horizons in Solvable Problems	83
3.12.4.2.	Spurring the Development of New Algorithms	83
3.12.4.3.	Rethinking Problems for Parallel Processors Pays Dividends on Other Processors	83
References.		84

Chapter 4

The MPP	85
4.1. The MPP Design	85

4.1.1.	MPP Software	85
4.1.2.	MPP Hardware	87
4.1.2.1.	Array Unit	87
4.1.2.2.	Array Control Unit.....	88
4.1.2.3.	Staging Memory	89
4.1.2.4.	Host Processor	90
4.2.	Parallel Forth Language	90
4.2.1.	Introduction	90
4.2.2.	Vocabulary and Data Definition	91
4.2.3.	Parallel I/O	92
4.2.4.	Memory Operations	92
4.2.5.	Array Stock Operations.....	92
4.2.6.	Arithmetic, Logic, and Comparison Operations	93
4.2.7.	Control Operations	94
4.2.8.	PECU and Mask Stock Primitives.....	95
4.2.9.	MPP Parallel Forth Word Reference	95
4.2.9.1.	Context Changing Words	95
4.2.9.2.	Arithmetic Words	95
4.2.9.3.	Logical Words	98
4.2.9.4.	Comparison Words.....	98
4.2.9.5.	Stack Operation Words.....	99
4.2.9.6.	Memory Operation Words	101
4.2.9.7.	Control Words	102
4.2.9.8.	I/O Words.....	103
4.2.9.9.	Defining Words	104
4.2.9.10.	Compiler Words	104
4.2.9.11.	PECU Primitive Words	104
4.2.9.12.	Mask Stack Operations.....	105
4.3.	Parallel Pascal Design	105
4.3.1.	Motivation.....	105
4.3.2.	Parallel Pascal Specification.....	106
4.4.	Ising Spin Exchange Simulation	120
4.4.1.	Introduction	120
4.4.2.	Algorithms	122
4.4.3.	Vector Machines	124
4.4.4.	Preliminary Results.....	125
4.5.	Stereo Analysis	125
4.5.1.	Introduction	125
4.5.2.	Background	126
4.5.3.	Difficulties in Stereo Matching.....	126
4.5.4.	Matching Technique.....	127
4.5.5.	Matching Algorithm on the MPP.....	128
4.5.6.	Preprocessing of the First Image	128
4.5.7.	Determination of Matches	129

4.5.8.	Removal of “Bad Match” Areas in the Disparity Function.....	129
4.5.9.	Smoothing the Resulting Disparity Function.....	131
4.5.10.	Warping the Test Image.....	131
4.5.11.	Interactive Operations on the MPP.....	131
4.5.12.	Interactive Turnaround Time.....	132
4.6.	Sort.....	132
4.6.1.	Massively Parallel Communications.....	132
4.6.2.	Sorting as a Communication Primitive.....	133
4.6.3.	Sorting Algorithms Implemented on the MPP.....	133
4.6.4.	Sort Aggregation.....	136
4.6.5.	Sort Distribution.....	137
4.6.6.	Merge Aggregation and Merge Distribution.....	139
4.6.7.	The Unmerge Operation.....	140
	References.....	141

Chapter 5

DAP.		143
5.1.	The DAP Design.....	143
5.1.1.	The DAP Software.....	144
5.2.	Parallel Data Transforms.....	156
5.2.1.	Introduction.....	156
5.2.2.	Computation on Processor Arrays.....	156
5.2.3.	A Complementary Approach.....	157
5.2.4.	An Overview of the PDT Approach.....	158
5.2.5.	Implementation.....	158
5.2.6.	Summary.....	159
5.3.	Solution of a Large System of Equation on DAP Using a Hybrid Gauss/Gauss-Jordan Method.....	159
5.3.1.	Introduction.....	159
5.3.2.	Principles of the Equation Solver.....	160
5.3.3.	Implementation Details.....	162
5.3.4.	Performance.....	164
5.4.	An Image Understanding Performance Study on the Distributed Array Processor.....	164
5.4.1.	Introduction.....	164
5.4.2.	Overview.....	164
5.4.3.	Image Capture.....	165
5.4.4.	DOG Convolutions.....	166
5.4.5.	Direct Application of Convolutions.....	167
5.4.6.	Segmentation.....	169
5.4.7.	Segment Labeling.....	172
5.4.8.	Feature Generation.....	173
5.4.9.	Classification.....	176
5.4.10.	Conclusion.....	177

References.....	178
-----------------	-----

Chapter 6

GAPP	181
6.1. The GAPP Design.....	181
6.1.1. Introduction.....	181
6.1.2. Background.....	181
6.1.3. Cell Description.....	182
6.1.3.1. General.....	182
6.1.3.2. Registers.....	183
6.1.3.3. Full Adder/Subtractor.....	183
6.1.3.4. RAM.....	183
6.1.3.5. Control/Clock.....	183
6.1.4. GAPP Chip Descriptions.....	183
6.1.4.1. Control.....	183
6.1.4.2. Shift Register Groups.....	184
6.1.4.3. Chip Performance/Mechanics.....	184
6.1.5. GAPP Array.....	185
6.1.5.1. Assembly.....	185
6.1.5.2. Input/Output.....	185
6.1.5.3. Sizes.....	185
6.2. GAPP Programming.....	186
6.2.1. Convolution and Correlation.....	191
6.2.2. Straightforward Convolutions.....	195
6.3. GAPP System.....	199
6.3.1. Introduction.....	199
6.3.2. Architecture.....	200

Chapter 7

The Connection Machine	205
7.1. The CM System.....	206
7.1.1. Overview.....	208
7.1.2. The Connection Machine.....	209
7.1.2.1. Performance Specifications.....	217
7.1.3. DataVault.....	219
7.2. Programming.....	221
7.2.1. The *Lisp Language.....	223
7.2.2. The C* Language.....	227
7.2.3. The Paris Language.....	233
7.3. Applications.....	236
7.3.1. Physics.....	237
Numerical Computation of Electromagnetic Scattering on the Connection Machine Using the Method of Moments.....	237

	Parallel Implicit Methods for Numerical Physics	
	Using the Connection Machine	241
7.3.2.	Operations Research.....	250
	Nonlinear Network Optimization on a Massively	
	Parallel Connection Machine	250
	Massively Parallel Implementation of Some	
	Common 0/1 Knapsack Approximation	
	Algorithms	264
7.3.3.	Image Processing.....	268
	Automatic Target Detection on the Connection	
	Machine	268
7.3.4.	Miscellaneous	273
	Parallel Free-Text Search.....	274
	Neural Network Implementation Approaches for	
	the Connection Machine.....	279
	Orbit Collision Problem Benchmark Study	291
References.....		298

Chapter 8

Research SIMD Computers.....	301
8.1. The GAM II Pyramid.....	301
8.1.1. Introduction	301
8.1.2. The Pyramid Structure	301
8.1.3. The Processing Element	302
8.1.4. The Daughter Card	302
8.1.5. The Back Plane	302
8.1.6. The Adder Network	303
8.1.7. The Sequencer.....	303
8.1.8. The Program Flow Sub-Unit	304
8.1.9. The Next Address Generator	304
8.1.10. Micro Memory	305
8.1.11. Pipeline Register	305
8.1.12. Condition Code Register	305
8.1.13. The Data Executive Sub-Unit	305
8.1.14. The General Purpose Registers	305
8.1.15. Special Purpose Registers	305
8.1.16. Data Memory.....	306
8.1.17. Array Instruction Generation	306
8.1.18. The Input and Output Unit	306
8.1.19. The Host System	306
8.1.20. Future Control System Expansion	307
8.2. CLIP4	308
8.2.1. Introduction	308
8.2.2. Array Processing	309
8.2.3. Processor Design General Considerations	311