COMPUTER ARCHITECTURE AND PARALLEL PROCESSING

Kai Hwang

Fayé A. Briggs

COMPUTER ARCHITECTURE AND PARALLEL PROCESSING

Kai Hwang
Purdue University

Fayé A. Briggs
Rice University

McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá Hamburg
Johannesburg London Madrid Mexico Montreal New Delhi
Panama Paris Sào Paulo Singapore Sydney Tokyo Toronto

This book was set in Times Roman.
The editors were Eric M. Munson and Jonathan Palace;
the production supervisor was Leroy A. Young.
The drawings were done by ANCO/Boston.
Halliday Lithograph Corporation was printer and binder.

COMPUTER ARCHITECTURE AND PARALLEL PROCESSING

1 2 3 4 5 6 7 8 9 0 HALHAL 8 9 8 7 6 5 4

ISBN 0-07-031556-6

Library of Congress Cataloging in Publication Data

Hwang, Kai.
Computer architecture and parallel processing.

(McGraw-Hill series in computer organization and architecture)
Bibliography: p.
Includes index.
1. Computer architecture. 2. Parallel processing (Electronic computers) I. Briggs, Faye A. (Faye Alaye) II. Title. III. Series.
QA76.9.A73H88 1984 001.64 83-23867
ISBN 0-07-031556-6

ABOUT THE AUTHORS

Kai Hwang is a professor of computer engineering at the School of Electrical Engineering, Purdue University. He received the Ph.D. in EECS from the University of California at Berkeley in 1972. His current research and teaching interest lies mainly in vector supercomputers, multiprocessors, VLSI and dataflow computing structures, and image understanding systems. Dr. Hwang has published about 60 scientific papers and books. He is the author of *Computer Arithmetic Principles, Architecture and Design* (Wiley, 1979) and the editor of *Supercomputer Design and Applications* (Computer Society Press, 1984). Dr. Hwang is the coeditor-in-chief of the *Journal of Parallel and Distributed Computing.* He has been involved in advanced studies on supercomputer architectures and large-scale scientific computations for U.S. National Science Foundation, IBM Fishkill Facilities, Fujitsu in Japan, Academia Sinica, and NATO Advanced Study Institute. He has been a distinguished visitor of IEEE Computer Society since 1981.

Fayé A. Briggs is an associate professor of electrical engineering at Rice University. He received the Ph.D. degree in electrical engineering from the University of Illinois, Urbana. Prior to joining the faculty at Rice University, Dr. Briggs was an assistant professor and the systems manager of the Advanced Automation and Research Laboratory at Purdue University. He has also served as a consultant to IBM Watson Research Center and Exxon Production Research Company. Since 1983 he has been a distinguished visitor of the IEEE Computer Society. Dr. Briggs' current research interests include performance evaluation of advanced computer architectures such as pipeline computers, multiprocessors, and VLSI computing structures. He has published over 30 technical papers and book chapters in the areas of multiprocessing computer architectures, memory organizations, and performance modeling techniques.

To my parents,
Hwang Yuan-Chung and Liu Cheng Fong,
my wife, Pu Fong,
and my sons, Tony and Andy.

Kai Hwang

黄鎧

To my grandparents,
Rev. P. B. Harry and Mrs. B. P. Harry.

Fayé A. Briggs

Fayéofori Alayé Nimifa-ã

PREFACE

High-performance computers are increasingly in demand in the areas of structural analysis, weather forecasting, petroleum exploration, fusion energy research, medical diagnosis, aerodynamics simulations, artificial intelligence, expert systems, industrial automation, remote sensing, military defense, genetic engineering, and socioeconomics, among many other scientific and engineering applications. Without superpower computers, many of these challenges to advance human civilization cannot be made within a reasonable time period. Achieving high performance depends not only on using faster and more reliable hardware devices but also on major improvements in computer architecture and processing techniques. This book is devoted to studying advanced computer architectures, theories of parallel computing, optimal resource allocation, fast algorithms, efficient programming languages, and application requirements of cost-effective computer systems to meet the above demands.

Advanced computer architectures are centered around the concept of parallel processing. State-of-the-art parallel computer systems can be characterized into three structural classes: *pipelined computers*, *array processors*, and *multiprocessor systems*. The development and application of these computer systems require a broad knowledge of the underlying hardware and software structures and close interactions between parallel computing algorithms and the optimal allocation of machine resources. This book provides the readers with the necessary knowledge to design a new computer system; to improve an existing one; to develop fast computing algorithms; and to allocate hardware-software resources in solving large-scale computing problems.

The book is designed to be used by seniors and/or graduate students in computer science, electrical engineering, industrial engineering, and in any other fields demanding the use of high-performance mainframe computers, attached vector processors, scientific supercomputers, and multiprocessors to solve specific applications problems. Parallel processing can be applied at the hardware/software system level or at the algorithmic and programming level. It demands concurrent execution of many programs in the computer. Parallel-processing computers provide a cost-effective means to achieve high system performance through concurrent activities.

Computer scientists, system designers, advanced programmers, application engineers, computational experimentalists, and computer professionals should find the material useful in their respective lines of work. Results obtained by many researchers, designers, and users of parallel-processing computers have been included in the text. The material being presented is the outgrowth from several courses in Computer Architecture and Advanced Computing taught by both authors, at Purdue University and Rice University. The book is organized into 10 chapters. Chapter 1 introduces the basic concepts of parallel processing and computer structures and prepares readers with an overview of parallelism in computer systems and various application areas. Chapter 2 presents memory hierarchy and input-output subsystems needed in parallel processing computers.

Chapters 3 and 4 are devoted to the design principles and applications of pipelined supercomputers. Systems to be studied include Star-100, TI-ASC, Cray-1, Cyber-205, Fujitsu VP-200, CDC-NASF, and attached scientific processors AP-120B (FPS-164), IBM 3838, and Datawest MATP. Language and compiler requirements are assessed toward optimized vectorization. Pipelined vector processing methods and performance evaluation of pipeline computers are also provided.

Chapters 5 and 6 present the interconnection structures of processor arrays. Such array processors can handle single instruction stream over multiple data streams. Several parallel algorithms are presented for array processors and associative processors. Case study systems include the Illiac-IV, Burroughs Scientific Processor (BSP), the STARAN, and PEPE, and the massively parallel processor (MPP). Performance enhancement methods are also provided for synchronous array processors.

In Chapters 7, 8, and 9, we study hardware system architectures, operating system controls, parallel algorithms, and performance evaluation of multiprocessor systems. Design experiences of three exploratory research multiprocessors, C.mmp, S-1, and Cm*, will be presented. Commercial multiprocessors to be studied include the IBM 370/168 MP, 3033, and 3081, Univac 1100/80 and 1100/90, Cray X-MP, Tandem/16, and Denelcor's HEP. Research issues toward designing tightly coupled multiprocessors are discussed.

Chapter 10 is devoted to studying new computing concepts and their realization issues. Principles of data-driven computations are introduced. Functional languages and existing data flow computer architectures are reviewed. Finally, we study parallel computing algorithms that are suitable for VLSI hardware implementations. Applications of VLSI architectures for image processing are presented.

The prerequisite for reading this book is an introductory undergraduate course in computer organization and programming. We use Fortran and its extensions in Chapters 3 to 6, because most vector supercomputers are Fortran machines. For multiprocessors in Chapters 7 to 9, we use concurrent Pascal as the illustrative language. Sections marked with asterisk (*) are research-oriented topics. Readers are expected to have some background on discrete mathematics and probability theory in studying these research topics. These difficult sections can be skipped in the first reading without loss of continuity. Homework problems are essential to provide readers with in-depth thinking and hands-on experience in the design, application, and evaluation of parallel computers.

Parallel processing and computer architecture are two wide-open areas for research and development. We hope that this book will inspire further advances in these frontier computer areas. Bibliographic notes are attached at the end of each chapter to help interested readers find additional references for extended studies. The authors are fully responsible for any errors or omissions. We apologize to those computer specialists whose original works are not included in this volume. The computer area is changing so rapidly that no book can cover every new progress being made. However, we do welcome inputs and criticisms from our readers. Readers are invited to send their comments directly to the authors, so that improvement can be made in future printings or revisions of the book.

This book can be used as a text when offering a sequence of two courses on computer architecture and parallel processing. Each course contains 45 lectures, each 50 minutes long. We suggest the following materials be covered in the first course of a two-course sequence. The remaining sections are reserved for the second course.

Material suggested for the first course

Chapter	Sections and subsections
1	1.1, 1.2, 1.3.1–3, 1.3.5, 1.4.1, 1.5
2	2.1.1, 2.1.3, 2.2.1–2, 2.4.1–2, 2.5.1
3	3.1.1–2, 3.2.1–2, 3.3.1–2, 3.4.1
4	4.1, 4.2.1, 4.3.1, 4.4.1–3
5	5.1, 5.2.1–2, 5.3.1, 5.4.1
6	6.1, 6.2.1–2, 6.3.1, 6.4.1
7	7.1, 7.2.1–2, 7.4.1, 7.5.1
8	8.1.1, 8.2.1, 8.3.1, 8.4.1
9	9.1, 9.2.1, 9.5.1–2, 9.6.1
10	10.1.1, 10.2.1, 10.3.1

The first course is suitable for senior and first-year graduate students. The second course is mainly for graduate students. The first course is a prerequisite for the second course. If the book is adopted for only one course offering, the instructor can move some sections from the second course to the first one to give more complete coverage of some selected topics which are of special interest to the instructor and students. This may necessitate trading some sections listed above with the added sections from the second course. A Solutions Manual to this book will be available from McGraw-Hill for instructors only. The manual contains solutions to all problems plus a number of design projects suitable for use as term projects. Instructors are welcome to communicate directly with the authors or with McGraw-Hill representatives for suggestions or sharing their experiences in using this book as either a required text or as a research reference.

The authors are grateful to a number of individuals whose professional encouragement and assistance have made the long writing and production process a very pleasant endeavour. In particular, we wish to thank Harold S. Stone, Jean Loup Baer, C. V. Ramamoorthy, Tse-yun Feng, John P. Hayes, King-Sun Fu, Clarance L. Coates, V. Carl Hamacher, Herschel H. Loomis, Jr., Bart Sinclair, J. Robert Jump, Edward S. Davidson, H. J. Siegel, Tom Mowbray, Wolfgang Händler, Kenichi Miura, Lional M. Ni, Michel Dubois, and Shun-Piao Su for their valuable comments and suggestions. Many of our students helped us in improving the manuscript. In particular, we thank Chi-Yuan Chin, Zhiwei Xu, and William Carlson for their assistance. Finally, we wish to thank the typing, drafting, and secretarial assistance from Andy Hughes, Wanda Booth, Linda Stovall, Vicki Johnson, Pat Loomis, Mickey Krebs, Sharon Katz, Nancy Lein, and D. Ringe. Last but not least we appreciate the McGraw-Hill editing staffs and production professionals for their excellent work in producing the book. Without the timely efforts of the above individuals, this book might be still in preparation.

Kai Hwang
Fayé A. Briggs

CONTENTS

CHAPTER

ONE

INTRODUCTION TO PARALLEL PROCESSING

Basic concepts of parallel processing on high-performance computers are introduced in this chapter. We will review the architectural evolution, examine various forms of concurrent activities in modern computer systems, and assess advanced applications of parallel processing computers. Parallel computer structures will be characterized as *pipelined computers*, *array processors*, and *multiprocessor systems*. Several new computing concepts, including data flow and VLSI approaches, will be introduced. The material presented in this introductory chapter will provide an overview of the field and pave the way to studying in subsequent chapters the details of theories of parallel computing, machine architectures, system controls, fast algorithms, and programming requirements.

1.1 EVOLUTION OF COMPUTER SYSTEMS

Over the past four decades the computer industry has experienced four generations of development, physically marked by the rapid changing of building blocks from relays and vacuum tubes (1940-1950s) to discrete diodes and transistors (1950–1960s), to small- and medium-scale integrated (SSI/MSI) circuits (1960–1970s), and to large- and very-large-scale integrated (LSI/VLSI) devices (1970s and beyond). Increases in device speed and reliability and reductions in hardware cost and physical size have greatly enhanced computer performance. However, better devices are not the sole factor contributing to high performance. Ever since the stored-program concept of von Neumann, the computer has been recognized as more than just a hardware organization problem. A modern computer system is really a composite of such items as processors, memories, functional units, interconnection networks, compilers, operating systems, peripheral devices, communication channels, and database banks.

To design a powerful and cost-effective computer system and to devise efficient programs to solve a computational problem, one must understand the underlying hardware and software system structures and the computing algorithms to be implemented on the machine with some user-oriented programming languages. These disciplines constitute the technical scope of *computer architecture*. Computer architecture is really a system concept integrating hardware, software, algorithms, and languages to perform large computations. A good computer architect should master all these disciplines. It is the revolutionary advances in integrated circuits and system architecture that have contributed most to the significant improvement of computer performance during the past 40 years. In this section, we review the generations of computer systems and indicate the general trends in the development of high performance computers.

1.1.1 Generations of Computer Systems

The division of computer systems into generations is determined by the device technology, system architecture, processing mode, and languages used. We consider each generation to have a time span of about 10 years. Adjacent generations may overlap in several years as demonstrated in Figure 1.1. The long time span is intended to cover both development and use of the machines in various parts of the world. We are currently in the fourth generation, while the fifth generation is not materialized yet.

The first generation (1938–1953) The introduction of the first electronic analog computer in 1938 and the first electronic digital computer, ENIAC (Electronic Numerical Integrator and Computer), in 1946 marked the beginning of the first generation of computers. Electromechanical relays were used as switching devices

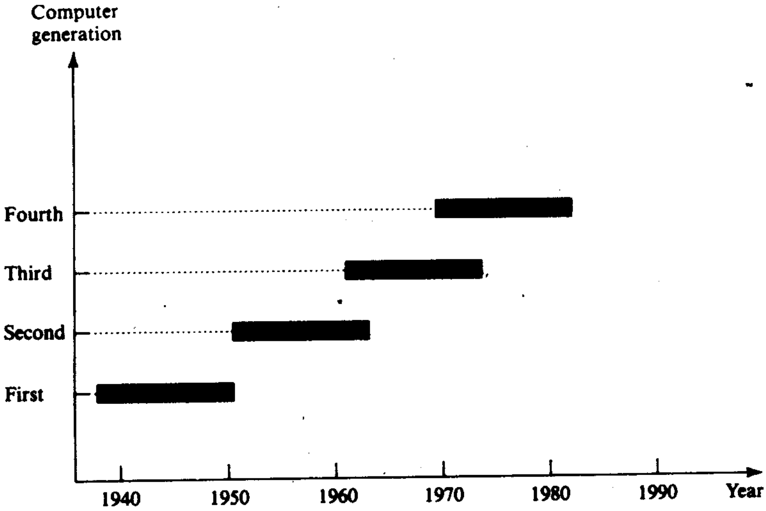

Figure 1.1 The evolution of computer systems.