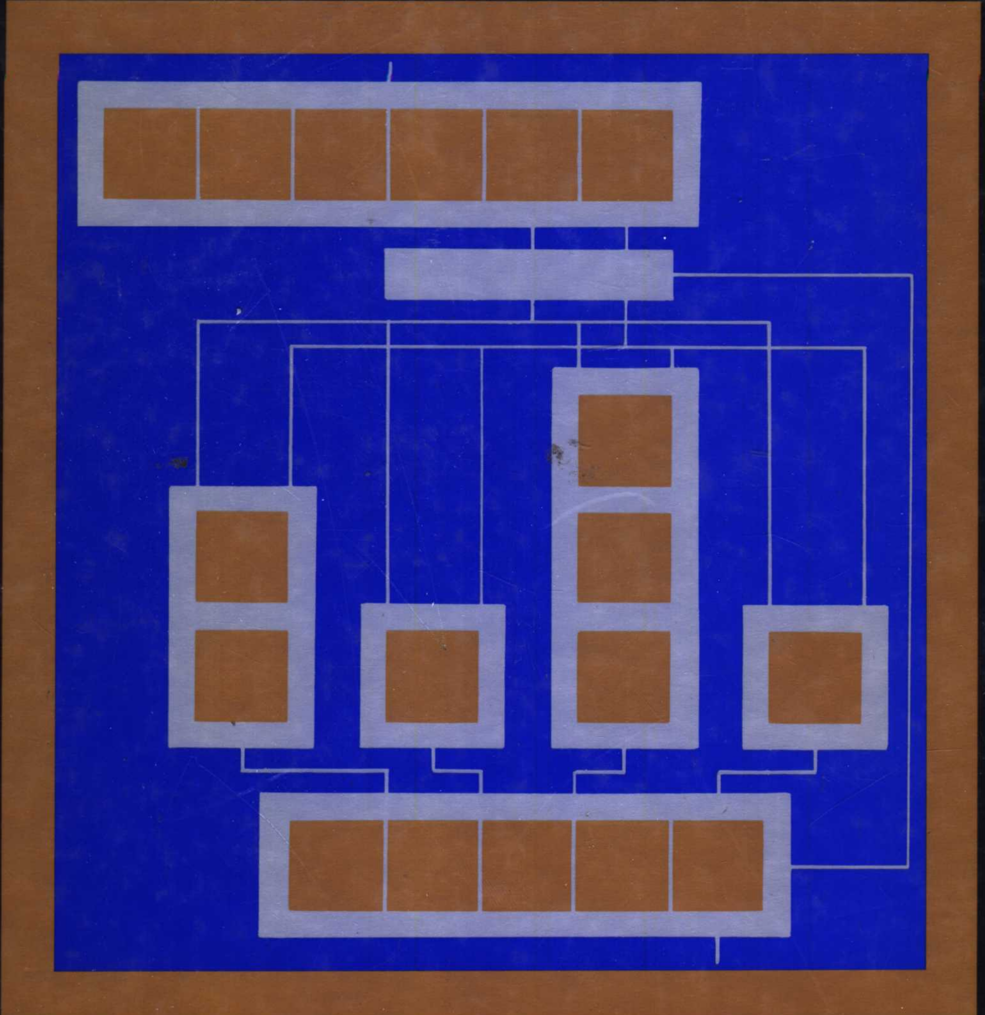
FOURTH EDITION

COMPUTER ORGANIZATION

V. CARL HAMACHER
ZVONKO G. VRANESIC
SAFWAT G. ZAKY



Computer Organization

FOURTH EDITION

V. Carl Hamacher
Queen's University

Zvonko G. Vranesic
University of Toronto

Safwat G. Zaky
University of Toronto

THE McGRAW-HILL COMPANIES, INC.

New York St. Louis San Francisco Auckland Bogotá Caracas
Lisbon London Madrid Mexico City Milan Montreal New Delhi
San Juan Singapore Sydney Tokyo Toronto

McGraw-Hill
A Division of The McGraw-Hill Companies

COMPUTER ORGANIZATION

This book is printed on acid-free paper.

5 6 7 8 9 0 DOC DOC 9 0 9

ISBN 0-07-025883-X

This book was set in Times Roman by Publication Services, Inc.
The editor was Eric M. Munson;
the production supervisor was Elizabeth J. Strange.
The cover was designed by Rafael Hernandez.
Project supervision was done by Publication Services, Inc.
R. R. Donnelley & Sons Company was printer and binder.

INTERNATIONAL EDITION

When ordering this title, use ISBN 0-07-114309-2.

Library of Congress Catalog Card Number:
95-81367

ABOUT THE AUTHORS

V. CARL HAMACHER received his B.A.Sc. degree in engineering physics from the University of Waterloo, Canada, the M.Sc. degree in electrical engineering from Queen's University, Kingston, Canada, and the Ph.D. degree in electrical engineering from Syracuse University, New York. From 1968 to 1990 he was at the University of Toronto, where he was professor in the Departments of Electrical Engineering and Computer Science. He also served as director of the Computer Systems Research Institute during 1984 to 1988, and as chairman of the Division of Engineering Science during 1988 to 1990. Since January 1991 he has been dean of the faculty of applied science at Queen's University. During 1978 to 1979 he was a visiting scientist at the IBM Research Laboratory in San Jose, California: and in 1986 he was a research visitor at the Laboratory for Circuits and Systems at the University of Grenoble, France.

His current research interests are in multiprocessors, multicomputers, and their interconnection networks, including local area networks.

ZVONKO G. VRANESIC received his B.A.Sc., M.A.Sc., and Ph.D. degrees, all in electrical engineering, from the University of Toronto. From 1963 to 1965 he worked as a design engineer with the Northern Electric Co. Ltd. in Bramalea, Ontario. In 1968 he joined the University of Toronto, where he is now a professor in the Departments of Electrical and Computer Engineering and Computer Science. Presently, he serves as chair of the Division of Engineering Science at the University of Toronto. During the 1978 to 1979 academic year, he was a senior visitor at the University of Cambridge, England, and during 1984 to 1985 he was at the University of Paris 6, France.

His current research interests include computer architecture, field-programmable VLSI technology, local area networks, and multiple-valued logic systems. He is a coauthor of two other books, *Microcomputer Structures* and *Field-Programmable Gate Arrays*.

SAFWAT G. ZAKY received his B.Sc. in electrical engineering and B.Sc. in mathematics, both from Cairo University, Egypt, and his M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto. He was with Bell Northern Research, Bramalea, Ontario, Canada, from 1969 to 1972, where he worked on applications of electro-optics and magnetics in mass storage and telephone switching. In 1973, he joined the University of Toronto, where he is currently a professor in the Department of Electrical and Computer Engineering and the Department of Computer Science. At present, he is serving as chair of the Department of Electrical and Computer Engineering. He was a senior visitor at the Computer Laboratory, University of Cambridge, England from 1980 to 1981.

His research interests are in the areas of computer architecture, logic synthesis, and electromagnetic compatibility of digital systems. He is a coauthor of the textbook *Microcomputer Structures*.

PREFACE

This book is intended for use in a first-level course on "computer organization" in electrical engineering and computer science curricula. The book is self-contained, assuming only that the reader has a basic knowledge of computer programming in a high-level language. Many students who study computer organization will have had an introductory course on digital logic circuits. Therefore, this subject is not covered in the main body of the book. However, we have provided an extensive appendix on logic circuits for those students who need it.

Our resolve to write the original version of the book stemmed from our experience in teaching computer organization to three distinct types of undergraduates: electrical and computer engineering undergraduates, computer science specialists, and engineering science undergraduates. We have always approached the teaching of courses in this area from as practical a point of view as possible. Thus, a major choice in shaping the contents of the four editions of the book has been to illustrate the principles of computer organization by using a number of extensive examples drawn from commercially available computers.

Second, we feel that it is important to recognize that digital system design is not a straightforward process of applying "optimal design" algorithms. Many design decisions are based largely on heuristic judgment and experience. They involve cost/performance tradeoffs over a range of alternatives. It is our goal to convey these notions to the reader.

Third, we have endeavored to provide sufficient details to force the student to dig beyond the surface when dealing with ideas that seem to be intuitively obvious. We believe that this is best accomplished by giving real examples that are adequately documented. Block diagrams are a powerful means of describing organizational features of a computer. However, they can easily lead to an oversimplified view of the problems involved. Hence, they must be accompanied by the details of implementation alternatives.

We use several real machines for illustrative purposes. Our main examples are drawn from the following computers: Motorola 680X0 family, PowerPC family, Intel 80X86 family, HP3000, and DEC Alpha AXP. The PowerPC and 68000 are used as detailed examples early in the book. Their manageable size and complexity make them suitable for teaching purposes.

The book is aimed at a one-semester course in engineering or computer science programs. It is suitable for both hardware- and software-oriented students; but there is a greater emphasis on hardware, because we feel that this is the way computer organization should be taught. It is a mistake to describe computer structures solely from the programming viewpoint, particularly for students who will eventually work with systems that involve a variety of equipment, interfacing, and communication facilities. Although the emphasis is on computer hardware, we have addressed a number of software issues and discussed representative instances of software-hardware tradeoffs in the implementation of various components of a computing system. We have also provided quantitative performance evaluations at both the component and system levels, where total time to execute a program is the dominant performance measure.

THE SCOPE OF THE BOOK

We now review the topics covered in sequence, chapter by chapter. The first seven chapters cover the basic principles of computer organization, operation, and performance, using the PowerPC and 68000 as major examples. The remaining three chapters deal with other commercial computer examples, peripheral devices, and large computer systems.

Chapter 1 provides an overview of computer hardware and software and informally introduces terms that are dealt with in more depth in the remainder of the book. This chapter discusses the basic ways in which standard functional units are interconnected to form a complete computing system, and the role of system software. Basic aspects of performance evaluation and a brief treatment of the history of computer development are also provided.

Chapter 2 gives a methodical treatment of addressing techniques and instruction sequencing. The Motorola 68000 and PowerPC are used to illustrate the basic concepts. Numerous program examples at the machine instruction level are used to discuss loops, subroutines, and simple input-output programming. All concepts are first presented in general terms and are then illustrated in separate complete parts for the 68000 and PowerPC. Instructors and students who wish to cover only one of these machine-specific parts will not miss any essential material.

Chapter 3 begins with a register-transfer-level treatment of the implementation of instruction fetching and execution in a processor. This is followed by a discussion of processor implementation by both hardwired and microprogrammed methods.

Input-output organization is developed in Chapter 4. The basics of I/O data transfer synchronization are presented, and a series of increasingly complex I/O structures are explained. Interrupts and direct-memory access methods are described in detail, including a discussion of the role of software interrupts in operating systems. Bus protocols and standards are also presented, with the SCSI and VMEbus standards being used as representative commercial examples.

Semiconductor memories are discussed in Chapter 5. Caches and multiple-module memory systems are explained as ways for increasing main memory bandwidth. Caches are discussed in some detail, including performance modeling. Virtual-memory systems and memory management are also presented.

Chapter 6 treats the arithmetic unit of a computer. It begins with a discussion of fixed-point add, subtract, multiply, and divide hardware, operating on 2's-complement numbers. Lookahead adders and high-speed multipliers are explained, including descriptions of the Booth multiplier recoding and carry-save addition techniques. Floating-point number representation and operation, in the context of the IEEE Standard, are presented.

Chapter 7 provides a detailed coverage of the use of pipelining in the design of high-performance processors. The role of the compiler and the relationship between pipelined execution and instruction set design are explored. The basics of superscalar instruction processing by the utilization of multiple functional units are also presented.

Chapter 8 discusses the Motorola 680X0 and Intel 80X86 families of processors as competing alternatives in the CISC class. The DEC Alpha processor is described as exemplifying the RISC approach to processor design, complementing the PowerPC presentation. The Hewlett-Packard HP3000, although not a modern processor, is described as a clear example of a processor design that is based on the stack structure for holding data operands.

Peripheral devices and communication with remote terminals using the RS-232-C serial standard are dealt with in Chapter 9. Video terminals are described, and secondary storage facilities based on magnetic disks, disk arrays, and magnetic tapes are explained.

Chapter 10 extends the discussion of computer organization to large systems based on the use of many processors operating in parallel. Both array processors, for highly-structured numerical processing computations, and multiprocessors, for more general parallel task execution, are covered. Interconnection networks for multiprocessors are described, and an introduction to cache coherence controls is also presented.

CHANGES IN THE FOURTH EDITION

Major changes in content and organization have been made in preparing the fourth edition of this book. They include the following:

- The PowerPC has been introduced in Chapter 2, replacing the PDP-11, as one of the two example processors used to illustrate the basic principles of addressing methods and machine program sequencing. The Motorola 68000 has been retained as the other example.
- The quantitative aspects of performance evaluation, in terms of program execution time, have been introduced in many places, particularly in Chapter 1 (introductory formulation), Chapter 5 (the effects of cache memory), Chapter 6 (arithmetic unit speedup techniques), and Chapter 7 (pipelining and multiple operation units), as a new feature in the fourth edition.
- Chapter 7 on pipelining is a significantly expanded version of the third edition treatment of this important design topic.
- Chapter 10 on large computer systems is a broadened view of the material on multiprocessors that was contained in the third edition, particularly in the areas of interconnection networks, memory organization, and cache coherence.
- Chapter 3 is a consolidation of hardwired and microprogrammed processor control design that was presented separately in Chapters 4 and 5 in the third edition.
- Chapter 8 is a consolidation and updating of material on several processors (Intel 80X86, DEC Alpha, and HP3000) that provide contrasting examples to the Motorola 680X0 and PowerPC. Comparable material was presented in Chapters 3 and 10 in the third edition.

- Chapter 9 on peripherals combines and updates material from Chapters 9 and 13 of the third edition, omitting much of the material on computer communications that was in the old Chapter 13 and was not central to computer organization.

In addition to these content and organizational changes, significant updating and rearrangement of material in the remaining chapters has also been done. Chapters 1 (basic structure), 4 (I/O), 5 (main memory), and 6 (arithmetic), replace Chapters 1, 6, 8, and 7, respectively, from the third edition.

WHAT CAN BE COVERED IN A ONE-SEMESTER COURSE

This book is intended for use at the university or college level as a text for a one-semester course in computer organization for either engineering or computer science students. There is more than enough material in the book for a one-semester course.

Coverage should include Chapter 1 as a reading assignment, followed by lectures on Chapters 2, 3, 4, 5, 6, and 7. For students who have not had a course in logic circuits, the basic material in Appendix A should be studied at the beginning of the course and certainly prior to covering Chapter 3. It takes a minimum of two weeks, or 5 or 6 lectures, to treat the essential material in Appendix A. Chapters 1 through 7, and the logic appendix, if needed, constitute the core material on computer organization that should be covered by all students.

There may not be enough time to cover all of Chapter 2, in which case we would advise studying only one of the two example processors (Motorola 68000 or the PowerPC) in addition to the machine-independent material in the first part of that chapter. For computer science students, the electronic aspects (memory cell circuits, refresh considerations, and so on) in Chapter 5 may be omitted, and the latter part of Chapter 4 on I/O interface circuits and bus standards can be dropped.

The material in Chapters 8 through 10 is not essential for a basic course in computer organization, but most instructors will probably want to introduce at least the material on magnetic disks and tapes in Chapter 9. It is important to give the student an appreciation for how these peripheral storage devices affect the overall performance of a complete computer system.

Chapters 8 and 10 contain material that is desirable for any student who wishes to go on to a more advanced level of understanding. The treatment of several commercial processors, other than the 68000 and the PowerPC, in Chapter 8 gives the student an appreciation for the variety that exists in practical processor designs. Chapter 10 is an introductory discussion of many aspects that are relevant to large computer systems containing many processors. In cases where the book might be appropriate for a senior level course or for a combined senior level/beginning graduate level course, both of these chapters should be covered and possibly supplemented with selections from other publications listed in the reference sections of those chapters.

Although we have indicated that some chapters or parts of chapters may be omitted, we offer the following suggestion. Any student who has a serious interest in the study of computer systems is urged to read the whole book and try the problems. We have attempted to give a carefully balanced description of what we believe to be the important aspects of computer organization, and we hope that the readers will benefit from our efforts.

ACKNOWLEDGMENTS

We wish to express our thanks to the people who have helped during the preparation of this fourth edition. Gail Dragan and Kelly Chan helped with the technical preparation of the manuscript. Dan Vranesic produced most of the artwork. Our colleagues Tarek Abdelrahman, Stephen Brown, Corinna Lee, and Jonathan Rose made many constructive comments. We are particularly grateful to Tarek for his help with the examples in Chapter 10. The reviewers, James Goodman, University of Wisconsin; Steve Liu, Texas A&M University; Dan Marinescu, Purdue University; Gandhi Puvvada, University of Southern California; Arun Somani, University of Washington; Daniel Tabak, George Mason University; and Philip Wilsey, University of Cincinnati, provided insightful criticism and helpful suggestions for improvements to the presentation. Our editor, Eric Munson, enthusiastically encouraged us and provided full support whenever needed.

V. Carl Hamacher
Zvonko G. Vranesic
Safwat G. Zaky

TABLE OF CONTENTS

CHAPTER 1

Basic Structure of Computer Hardware and Software

This book is about computer organization. It describes the function and design of the various units of digital computers that store and process information. It also deals with the units of the computer that receive information from and that send computed results to the outside world. Most of the material in this book is devoted to *computer hardware* and *computer architecture*. Computer hardware is the electronic circuits and electromechanical equipment that constitutes the computer. Computer architecture is defined as the functional operation of the individual hardware units in a computer system and the flow of information among and the control of those units.

Many aspects of programming and software components in computer systems are also discussed in this book. It is important to consider both hardware and software aspects of the design of various computer components in order to achieve a good understanding of computer systems.

This chapter introduces a number of hardware and software concepts, presents some common terminology, and gives a broad overview of the fundamental aspects of the subject. More detailed discussions follow in subsequent chapters.

1.1 FUNCTIONAL UNITS

Let us first define the term *digital computer,* or simply *computer;* this term is often misunderstood although it is widely used. In its simplest form, a contemporary computer is a fast electronic calculating machine that accepts digitized input information, processes it according to a list of internally stored instructions, and produces the resulting output information. The list of instructions is called a computer *program,* and the internal storage is called computer *memory.*

Many types of computers exist that differ widely in size, speed, and cost. It is fashionable to use more specific words to represent common types of computers. The most common computer is the *personal computer,* which has found wide use in homes,