COMPUTER ORGANIZATION

FIFTH EDITION

Carl Hamacher
Queen's University

Zvonko Vranesic
University of Toronto

Safwat Zaky
University of Toronto

Boston Burr Ridge, IL Dubuque, IA Madison, WI New York San Francisco St. Louis
Bangkok Bogotá Caracas Kuala Lumpur Lisbon London Madrid Mexico City
Milan Montreal New Delhi Santiago Seoul Singapore Sydney Taipei Toronto

The McGraw-Hill Companies

Mc Graw Hill

COMPUTER ORGANIZATION, Fifth Edition
International Edition 2002

20 19 18 17 16 15 14 13 12 11
20 09 08
CTF BJE

Library of Congress Control Number: 2001030712

When ordering this title, use ISBN 007-120411-3

Printed in Singapore

www.mhhe.com

ABOUT THE AUTHORS

Carl Hamacher received his B.A.Sc. degree in engineering physics from the University of Waterloo, Canada, an M.Sc. degree in electrical engineering from Queen's University, Kingston, Canada, and a 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 a Professor in the Departments of Electrical Engineering and Computer Science. He 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 a Professor of Electrical and Computer Engineering at Queen's University. He served as the dean of the Faculty of Applied Science from 1991 to 1996. During 1978 to 1979, he was a visiting scientist at the IBM Research Laboratory in San Jose, California. In 1986, he was a research visitor at the Laboratory for Circuits and Systems associated with the University of Grenoble in France. In 1996 to 1997, he was a visiting professor in the Computer Science Department at the University of California at Riverside and in the LIP6 Laboratory of the University of Paris VI, France.

His research interests are in multiprocessors and multicomputers, focusing on their interconnection networks.

Zvonko Vranesic received his B.A.Sc., M.A.Sc., and Ph.D. degrees, 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 Department of Electrical and Computer Engineering and the Department of Computer Science. During 1978 to 1979, he was a senior visitor at the University of Cambridge, England, and during 1984 to 1985 he was at the University of Paris VI, France. In 2000 to 2001, he was a principal software engineer at Altera Corporation in Toronto. From 1995 to 2000, he served as chair of the Division of Engineering Science at the University of Toronto.

His current research interests include computer architecture, field-programmable VLSI technology, and multiple-valued logic systems. He is a coauthor of three other books: *Fundamentals of Digital Logic with VHDL Design*, *Microcomputer Structures*, and *Field-Programmable Gate Arrays*. In 1990, he received the Wighton Fellowship for "innovative and distinctive contributions to undergraduate laboratory instruction."

Safwat Zaky received his B.Sc. degree 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. From 1969 to 1972 he was with Bell Northern Research, Bramalea, Ontario, 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 now a Professor in the Department of Electrical and Computer Engineering and the Department of Computer Science. Presently, he

serves as chair of the Department of Electrical and Computer Engineering. From 1980 to 1981, he was a senior visitor at the Computer Laboratory, University of Cambridge, England.

His research interests are in the areas of computer architecture, reliability of digital circuits, and electromagnetic compatibility. He is a coauthor of the book *Microcomputer Structures* and is a recipient of the IEEE Third Millennium Medal.

To Liz, Anne, and Shirley

CONTENTS

PREFACE

This book is intended for use in a first-level course on computer organization in electrical engineering, computer 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.

The book reflects our experience in teaching computer organization to three distinct groups 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 a practical point of view. Thus, a key consideration in shaping the contents of the book has been to illustrate the principles of computer organization using examples drawn from commercially available computers. Our main examples are based on the following processors: ARM, Motorola 680X0, Intel Pentium, and Sun UltraSPARC.

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 and hardware/software tradeoffs over a range of alternatives. It is our goal to convey these notions to the reader.

We have endeavored to provide sufficient details to encourage 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.

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. Even though the emphasis is on hardware, we have addressed a number of software issues, including basic aspects of compilers and operating systems related to instruction execution performance, coordination of parallel operations at the system level, and real-time applications. An understanding of hardware/software interaction and tradeoffs is necessary for computer specialists.

THE SCOPE OF THE BOOK

We now review the topics covered in sequence, chapter by chapter. The first eight chapters cover the basic principles of computer organization, operation, and performance.

The remaining four chapters deal with embedded systems, peripheral devices, processor family evolution patterns, 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 functional units and the ways they are interconnected to form a complete computer system. The role of system software is introduced and basic aspects of performance evaluation are discussed. A brief treatment of the history of computer development is also provided.

Chapter 2 gives a methodical treatment of machine instructions, addressing techniques, and instruction sequencing. Basic aspects of 2's-complement arithmetic are introduced to facilitate the discussion of the generation of effective addresses. Program examples at the machine instruction level, expressed in a generic assembly language, are used to discuss loops, subroutines, simple input-output programming, sorting, and linked list operations.

Chapter 3 illustrates implementation of the concepts introduced in Chapter 2 on three commercial processors — ARM, 68000, and Pentium. The ARM processor illustrates the RISC design style, the 68000 has an easy-to-teach CISC design, while the Pentium represents the most successful commercial design that combines the elements of both the CISC and RISC styles. The material is organized into three independent and complete parts. Each part includes all of the examples from Chapter 2 implemented in the context of the specific processor. It is sufficient to cover only one of the three parts to provide the continuity needed to follow the rest of the book. If laboratory experiments using one of the three processors are associated with the course, the relevant part of Chapter 3 can be covered in parallel with Chapter 2.

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 PCI, SCSI, and USB standards being used as representative commercial examples.

Semiconductor memories, including SDRAM, Rambus, and Flash memory implementations, 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, memory management, and rapid address translation techniques are also presented. Magnetic and optical disks are discussed as components in the memory hierarchy.

Chapter 6 treats the arithmetic unit of a computer. Logic design for fixed-point add, subtract, multiply, and divide hardware, operating on 2's-complement numbers, is described. 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 operations, in the context of the IEEE Standard, are presented.

Chapter 7 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 control.

Chapter 8 provides a detailed coverage of the use of pipelining and multiple function units in the design of high-performance processors. The role of the compiler and the relationship between pipelined execution and instruction set design are explored. Superscalar processors are discussed, and the Sun Microsystems UltraSPARC II processor organization is used to illustrate the concepts.

Today there are many more processors in use in embedded systems than in general-purpose computers. This increasingly important subject, where a single chip integrates the processing, I/O, and timer functionality needed in a wide range of low-cost applications, is treated in Chapter 9. System integration issues, interconnections, and real-time software are discussed.

Chapter 10 presents peripheral devices and computer interconnections. Typical input/output devices are described and hardware needed to support computer graphics applications is introduced. Commonly used communication links, such as DSL, are discussed.

The evolution of the ARM, Motorola, and Intel processor families is discussed in Chapter 11. This chapter highlights the design changes that led to higher performance. The PowerPC, SPARC, Alpha, and Intel IA-64 families are also discussed.

Chapter 12 extends the discussion of computer organization to large systems that use many processors operating in parallel. Interconnection networks for multiprocessors are described, and an introduction to cache coherence controls is presented. Shared-memory and message-passing schemes are discussed.

CHANGES IN THE FIFTH EDITION

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

- Chapter 2 of the fourth edition has been split into two chapters — Chapters 2 and 3 — in the fifth edition. An expanded treatment of basic issues, explained using generic instructions, is presented in Chapter 2. More programming examples for typical tasks, both numeric and non-numeric, are provided. Chapter 3 uses the instruction sets of ARM, 68000, and Pentium processors to show how the basic concepts of instruction set design have been implemented in both the RISC and CISC design styles.
- The discussion of the role of pipelining and multiple functional units in processor design has been extended significantly. The UltraSPARC architecture is used to provide specific examples of performance-enhancing design features.
- A new chapter on embedded-processor systems has been added. A generic design of a typical system is used as the basis for detailed discussion of example applications.

In addition to these main changes, many recent technology and design advances have been added to a number of chapters.

WHAT CAN BE COVERED IN A ONE-SEMESTER COURSE

This book is suitable for use at the university or college level as a text for a one-semester course in computer organization. It is intended for use in the first course on computer organization that the students will take.

There is more than enough material in the book for a one-semester course. The core material is given in Chapters 1 through 8. 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 4.

Chapters 9 through 12 contain a variety of useful material that the instructor may choose from if time permits. Particularly suitable are the discussion of embedded systems in Chapter 9 and the description of hardware found in most personal computers given in Chapter 10.

ACKNOWLEDGMENTS

We wish to express our thanks to many people who have helped during the preparation of this fifth edition. Gail Burgess and Kelly Chan helped with the technical preparation of the manuscript. Alex Grbic, Frank Hsu and Robert Lu provided valuable help with a number of programming examples. Our colleagues Tarek Abdelrahman, Stephen Brown, Paul Chow, Glenn Gulak and Jonathan Rose offered constructive comments. We are particularly grateful to Stephen and Tarek for their help with important details. The reviewers, Gojko Babic of The Ohio State University, Nathaniel Davis of Virginia Polytechnic Institute and State University, Jose Fortes of Purdue University, John Greiner of Rice University, Sung Hu of San Francisco State University, Ali Hurson of Pennsylvania State University, Lizy Kurian John of University of Texas at Austin, Stefan Leue of Albert Ludwigs Universitat in Freiburg, Fabrizio Lombardi of Northeastern University, Wayne Loucks of University of Waterloo, Prasant Mohapatra of Iowa State University, Daniel Tabak of George Mason University, and John Valois of Rensselaer Polytechnic Institute gave us many excellent suggestions and provided constructive criticism. We want to thank Eli Vranesic for permission to use his painting "Fall in High Park" on the front cover; he created it using the computer as a paint brush. Finally, we truly appreciate the support of our editor, Catherine Fields Shultz, and her McGraw-Hill associates: Kelley Butcher, Michelle Flomenhoft, Kalah Graham, Betsy Jones, Rick Noel, Heather Sabo, and Christine Walker.

Carl Hamacher
Zvonko Vranesic
Safwat Zaky