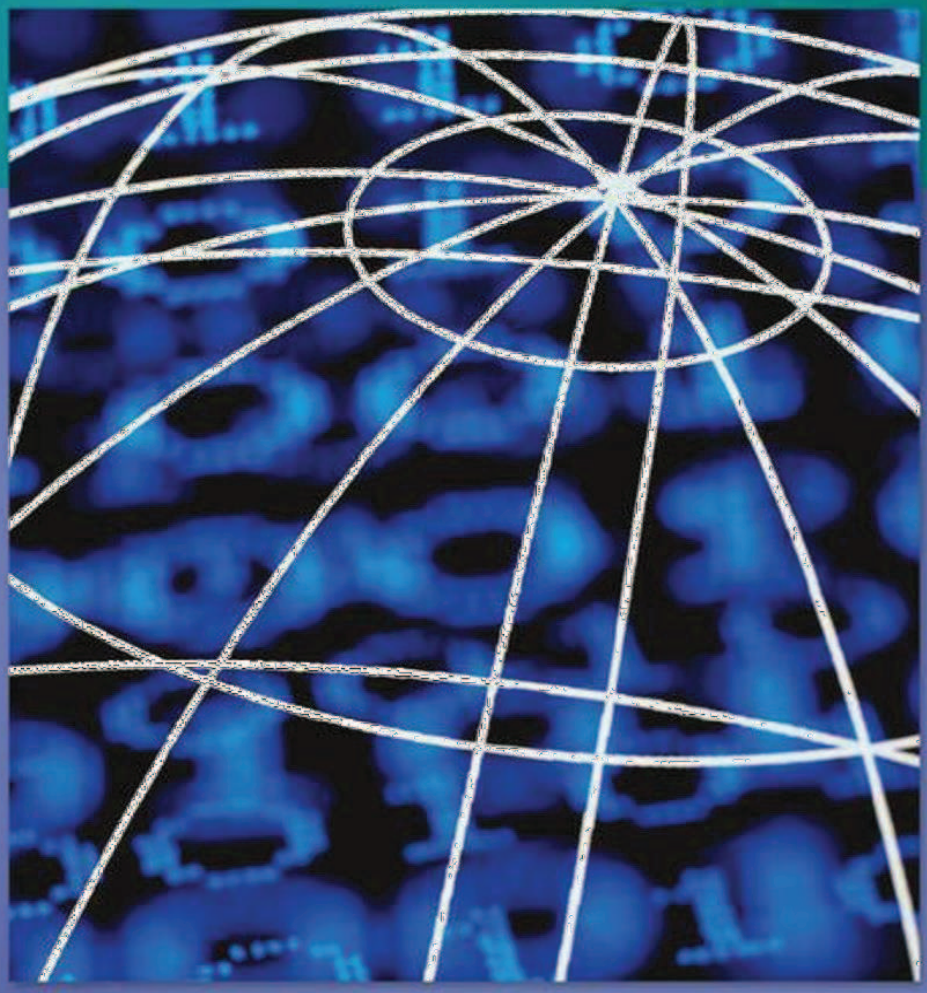
Carl Hamacher · Zvonko Vranesic · Safwat Zaky · Naraig Manjikian



COMPUTER ORGANIZATION
AND EMBEDDED SYSTEMS

Sixth Edition

This page intentionally left blank

COMPUTER ORGANIZATION
AND EMBEDDED SYSTEMS

This page intentionally left blank

COMPUTER ORGANIZATION
AND EMBEDDED SYSTEMS

SIXTH EDITION

Carl Hamacher
Queen's University

Zvonko Vranesic
University of Toronto

Safwat Zaky
University of Toronto

Naraig Manjikian
Queen's University

The McGraw-Hill Companies

COMPUTER ORGANIZATION AND EMBEDDED SYSTEMS, SIXTH EDITION

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 DOC/DOC 0 9 8 7 6 5 4 3 2 1

ISBN 978–0–07–338065–0
MHID 0–07–338065–2

Vice President & Editor-in-Chief: *Marty Lange*
Vice President EDP/Central Publishing Services: *Kimberly Meriwether David*
Publisher: *Raghothaman Srinivasan*
Senior Sponsoring Editor: *Peter E. Massar*
Developmental Editor: *Darlene M. Schueller*
Senior Marketing Manager: *Curt Reynolds*
Senior Project Manager: *Lisa A. Bruflodt*
Buyer: *Laura Fuller*
Design Coordinator: *Brenda A. Rolwes*
Media Project Manager: *Balaji Sundararaman*
Cover Design: *Studio Montage, St. Louis, Missouri*
Cover Image: © *Royalty-Free/CORBIS*
Compositor: *Techsetters, Inc.*
Typeface: *10/12 Times Roman*
Printer: *R. R. Donnelley & Sons Company/Crawfordsville, IN*

Library of Congress Cataloging-in-Publication Data

Computer organization and embedded systems / Carl Hamacher ... [et al.]. – 6th ed.
p. cm.
Includes bibliographical references.
ISBN-13: 978-0-07-338065-0 (alk. paper)
ISBN-10: 0-07-338065-2 (alk. paper)
1. Computer organization. 2. Embedded computer systems. I. Hamacher, V. Carl.
QA76.9.C643.H36 2012
004.2'2–dc22

2010050243

www.mhhe.com

To our families

This page intentionally left blank

About the Authors

Carl Hamacher received the B.A.Sc. degree in Engineering Physics from the University of Waterloo, Canada, the M.Sc. degree in Electrical Engineering from Queen's University, 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, Canada, where he was a Professor in the Department of Electrical Engineering and the Department of 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. In 1991 he joined Queen's University, where is now Professor Emeritus in the Department of Electrical and Computer Engineering. He served as 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, France. During 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.

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, 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 Emeritus in the Department of Electrical & Computer Engineering. During the 1978–79 academic year, he was a Senior Visitor at the University of Cambridge, England, and during 1984-85 he was at the University of Paris, 6. From 1995 to 2000 he served as Chair of the Division of Engineering Science at the University of Toronto. He is also involved in research and development at the Altera Toronto Technology Center.

His current research interests include computer architecture and field-programmable VLSI technology.

He is a coauthor of four other books: *Fundamentals of Digital Logic with VHDL Design*, 3rd ed.; *Fundamentals of Digital Logic with Verilog Design*, 2nd ed.; *Microcomputer Structures*; and *Field-Programmable Gate Arrays*. In 1990, he received the Wighton Fellowship for "innovative and distinctive contributions to undergraduate laboratory instruction." In 2004, he received the Faculty Teaching Award from the Faculty of Applied Science and Engineering at the University of Toronto.

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

This page intentionally left blank

magnetics in mass storage and telephone switching. In 1973, he joined the University of Toronto, where he is now Professor Emeritus in the Department of Electrical and Computer Engineering. He served as Chair of the Department from 1993 to 2003 and as Vice-Provost from 2003 to 2009. During 1980 to 1981, he was a senior visitor at the Computer Laboratory, University of Cambridge, England.

He is a Fellow of the Canadian Academy of Engineering. His research interests are in the areas of computer architecture, digital-circuit design, and electromagnetic compatibility. He is a coauthor of the book *Microcomputer Structures* and is a recipient of the IEEE Third Millennium Medal and of the Vivek Goel Award for distinguished service to the University of Toronto.

Naraig Manjikian received his B.A.Sc. degree in Computer Engineering and M.A.Sc. degree in Electrical Engineering from the University of Waterloo, Canada, and his Ph.D. degree in Electrical Engineering from the University of Toronto. In 1997, he joined Queen's University, Kingston, Canada, where he is now an Associate Professor in the Department of Electrical and Computer Engineering. From 2004 to 2006, he served as Undergraduate Chair for Computer Engineering. From 2006 to 2007, he served as Acting Head of the Department of Electrical and Computer Engineering, and from 2007 until 2009, he served as Associate Head for Student and Alumni Affairs. During 2003 to 2004, he was a visiting professor at McGill University, Montreal, Canada, and the University of British Columbia. During 2010 to 2011, he was a visiting professor at McGill University.

His research interests are in the areas of computer architecture, multiprocessor systems, field-programmable VLSI technology, and applications of parallel processing.

PREFACE

This book is intended for use in a first-level course on computer organization and embedded systems 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 three distinct groups of students: electrical and computer engineering undergraduates, computer science undergraduates, and engineering science undergraduates. We have always approached the teaching of courses on computer organization from a practical point of view. Thus, a key consideration in shaping the contents of the book has been to carefully explain the main principles, supported by examples drawn from commercially available processors. Our main commercial examples are based on: Altera's Nios II, Freescale's ColdFire, ARM, and Intel's IA-32 architectures.

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.

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 relevant software issues.

McGraw-Hill maintains a Website with support material for the book at http://www.mhhe.com/hamacher.

SCOPE OF THE BOOK

The first three chapters introduce the basic structure of computers, the operations that they perform at the machine-instruction level, and input/output methods as seen by a programmer. The fourth chapter provides an overview of the system software needed to translate programs written in assembly and high-level languages into machine language and to manage their execution. The remaining eight chapters deal with the organization, interconnection, and performance of hardware units in modern computers, including a coverage of embedded systems.

Five substantial appendices are provided. The first appendix covers digital logic circuits. Then, four current commercial instruction set architectures—Altera's Nios II, Freescale's ColdFire, ARM, and Intel's IA-32—are described in separate appendices.

Chapter 1 provides an overview of computer hardware and informally introduces terms that are discussed in more depth in the remainder of the book. This chapter discusses

the basic functional units and the ways they interact to form a complete computer system. Number and character representations are discussed, along with basic arithmetic operations. An introduction to performance issues and a brief treatment of the history of computer development are also provided.

Chapter 2 gives a methodical treatment of machine instructions, addressing techniques, and instruction sequencing. Program examples at the machine-instruction level, expressed in a generic assembly language, are used to discuss concepts that include loops, subroutines, and stacks. The concepts are introduced using a RISC-style instruction set architecture. A comparison with CISC-style instruction sets is also included.

Chapter 3 presents a programmer's view of basic input/output techniques. It explains how program-controlled I/O is performed using polling, as well as how interrupts are used in I/O transfers.

Chapter 4 considers system software. The tasks performed by compilers, assemblers, linkers, and loaders are explained. Utility programs that trace and display the results of executing a program are described. Operating system routines that manage the execution of user programs and their input/output operations, including the handling of interrupts, are also described.

Chapter 5 explores the design of a RISC-style processor. This chapter explains the sequence of processing steps needed to fetch and execute the different types of machine instructions. It then develops the hardware organization needed to implement these processing steps. The differing requirements of CISC-style processors are also considered.

Chapter 6 provides coverage of the use of pipelining and multiple execution units in the design of high-performance processors. A pipelined version of the RISC-style processor design from Chapter 5 is used to illustrate pipelining. The role of the compiler and the relationship between pipelined execution and instruction set design are explored. Superscalar processors are discussed.

Input/output hardware is considered in **Chapter 7**. Interconnection networks, including the bus structure, are discussed. Synchronous and asynchronous operation is explained. Interconnection standards, including USB and PCI Express, are also presented.

Semiconductor memories, including SDRAM, Rambus, and Flash memory implementations, are discussed in **Chapter 8**. Caches are explained as a way for increasing the memory bandwidth. They 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 9 explores the implementation of 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. Carry-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.

Today, far more processors are in use in embedded systems than in general-purpose computers. **Chapters 10 and 11** are dedicated to the subject of embedded systems. First, basic aspects of system integration, component interconnections, and real-time operation are presented in Chapter 10. The use of microcontrollers is discussed. Then, Chapter 11 concentrates on system-on-a-chip (SoC) implementations, in which a single chip integrates

the processing, memory, I/O, and timer functionality needed to satisfy application-specific requirements. A substantial example shows how FPGAs and modern design tools can be used in this environment.

Chapter 12 focuses on parallel processing and performance. Hardware multithreading and vector processing are introduced as enhancements in a single processor. Shared-memory multiprocessors are then described, along with the issue of cache coherence. Interconnection networks for multiprocessors are presented.

Appendix A provides extensive coverage of logic circuits, intended for a reader who has not taken a course on the design of such circuits.

Appendices B, C, D, and E illustrate how the instruction set concepts introduced in Chapters 2 and 3 are implemented in four commercial processors: Nios II, ColdFire, ARM, and Intel IA-32. The Nios II and ARM processors illustrate the RISC design style. ColdFire has an easy-to-teach CISC design, while the IA-32 CISC architecture represents the most successful commercial design. The presentation for each processor includes assembly-language examples from Chapters 2 and 3, implemented in the context of that processor. The details given in these appendices are not essential for understanding the material in the main body of the book. It is sufficient to cover only one of these appendices to gain an appreciation for commercial processor instruction sets. The choice of a processor to use as an example is likely to be influenced by the equipment in an accompanying laboratory. Instructors may wish to use more that one processor to illustrate the different design approaches.

CHANGES IN THE SIXTH EDITION

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

- The basic concepts of instruction set architecture are now covered using the RISC-style approach. This is followed by a comparative examination of the CISC-style approach.
- The processor design discussion is focused on a RISC-style implementation, which leads naturally to pipelined operation.
- Two chapters on embedded systems are included: one dealing with the basic structure of such systems and the use of microcontrollers, and the other dealing with system-on-a-chip implementations.
- Appendices are used to give examples of four commercial processors. Each appendix includes the essential information about the instruction set architecture of the given processor.
- Solved problems have been included in a new section toward the end of chapters and appendices. They provide the student with solutions that can be expected for typical problems.

DIFFICULTY LEVEL OF PROBLEMS

The problems at the end of chapters and appendices have been classified as easy (E), medium (M), or difficult (D). These classifications should be interpreted as follows:

- Easy—Solutions can be derived in a few minutes by direct application of specific information presented in one place in the relevant section of the book.
- Medium—Use of the book material in a way that does not directly follow any examples presented is usually needed. In some cases, solutions may follow the general pattern of an example, but will take longer to develop than those for easy problems.
- Difficult—Some additional insight is needed to solve these problems. If a solution requires a program to be written, its underlying algorithm or form may be quite different from that of any program example given in the book. If a hardware design is required, it may involve an arrangement and interconnection of basic logic circuit components that is quite different from any design shown in the book. If a performance analysis is needed, it may involve the derivation of an algebraic expression.

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 the first course that students will take on computer organization.

There is more than enough material in the book for a one-semester course. The core material on computer organization and relevant software issues is given in Chapters 1 through 9. For students who have not had a course in logic circuits, the material in Appendix A should be studied near the beginning of a course and certainly prior to covering Chapter 5.

A course aimed at embedded systems should include Chapters 1, 2, 3, 4, 7, 8, 10 and 11.

Use of the material on commercial processor examples in Appendices B through E can be guided by instructor and student interest, as well as by relevance to any hardware laboratory associated with a course.

ACKNOWLEDGMENTS

We wish to express our thanks to many people who have helped us during the preparation of this sixth edition of the book.

Our colleagues Daniel Etiemble of University of Paris South and Glenn Gulak of University of Toronto provided numerous comments and suggestions that helped significantly in shaping the material.

Blair Fort and Dan Vranesic provided valuable help with some of the programming examples.

Warren R. Carithers of Rochester Institute of Technology, Krishna M. Kavi of University of North Texas, and Nelson Luiz Passos of Midwestern State University provided reviews of material from both the fifth and sixth editions of the book.

The following people provided reviews of material from the fifth edition of the book: Goh Hock Ann of Multimedia University, Joseph E. Beaini of University of Colorado Denver, Kalyan Mohan Goli of Jawaharlal Nehru Technological University, Jaimon Jacob of Model Engineering College Ernakulam, M. Kumaresan of Anna University Coimbatore,

Kenneth K. C. Lee of City University of Hong Kong, Manoj Kumar Mishra of Institute of Technical Education and Research, Junita Mohamad-Saleh of Universiti Sains Malaysia, Prashanta Kumar Patra of College of Engineering and Technology Bhubaneswar, Shanq-Jang Ruan of National Taiwan University of Science and Technology, S. D. Samantaray of G. B. Pant University of Agriculture and Technology, Shivakumar Sastry of University of Akron, Donatella Sciuto of Politecnico of Milano, M. P. Singh of National Institute of Technology Patna, Albert Starling of University of Arkansas, Shannon Tauro of University of California Irvine, R. Thangarajan of Kongu Engineering College, Ashok Kunar Turuk of National Institute of Technology Rourkela, and Philip A. Wilsey of University of Cincinnati.

Finally, we truly appreciate the support of Raghothaman Srinivasan, Peter E. Massar, Darlene M. Schueller, Lisa Bruflodt, Curt Reynolds, Brenda Rolwes, and Laura Fuller at McGraw-Hill.

Carl Hamacher
Zvonko Vranesic
Safwat Zaky
Naraig Manjikian

McGraw-Hill Create™ Craft your teaching resources to match the way you teach! With McGraw-Hill Create, www.mcgrawhillcreate.com, you can easily rearrange chapters, combine material from other content sources, and quickly upload content you have written like your course syllabus or teaching notes. Find the content you need in Create by searching through thousands of leading McGraw-Hill textbooks. Arrange your book to fit your teaching style. Create even allows you to personalize your book's appearance by selecting the cover and adding your name, school, and course information. Order a Create book and you'll receive a complimentary print review copy in 3-5 business days or a complimentary electronic review copy (eComp) via email in minutes. Go to www.mcgrawhillcreate.com today and register to experience how McGraw-Hill Create empowers you to teach your students your way.

McGraw-Hill Higher Education and Blackboard® have teamed up.

Blackboard, the Web-based course management system, has partnered with McGraw-Hill to better allow students and faculty to use online materials and activities to complement face-to-face teaching. Blackboard features exciting social learning and teaching tools that foster more logical, visually impactful and active learning opportunities for students. You'll transform your closed-door classrooms into communities where students remain connected to their educational experience 24 hours a day.

This partnership allows you and your students access to McGraw-Hill's Create right from within your Blackboard course - all with one single sign-on. McGraw-Hill and Blackboard can now offer you easy access to industry leading technology and content, whether your campus hosts it, or we do. Be sure to ask your local McGraw-Hill representative for details.

CONTENTS