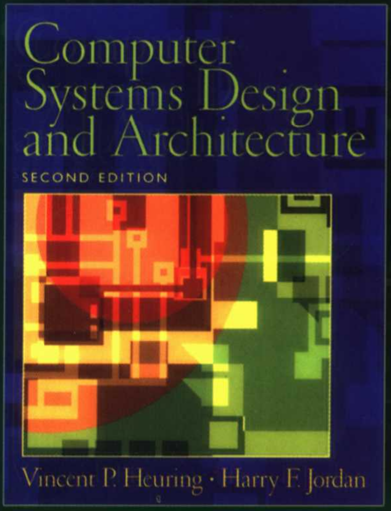
国外计算机科学教材系列

计算机系统设计与结构

（第二版）

Computer Systems Design and Architecture

Second Edition


Computer Systems Design and Architecture

SECOND EDITION

Vincent P. Heuring · Harry F. Jordan

英文版

[美] Vincent P. Heuring
Harry F. Jordan 著

PEARSON
Prentice Hall

PHEI
電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

国外计算机科学教材系列

计算机系统设计与结构

（第二版）

（英文版）

Computer Systems Design and Architecture

Second Edition

［美］ Vincent P. Heuring
Harry F. Jordan 著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书从系统结构设计师、汇编程序员和逻辑设计师的角度介绍了计算机系统结构的设计。本书从计算机系统结构设计的综述入手，讲解了机器和机器语言之间的关系，引入了有代表性且容易理解的SRC模型和RTN结构功能描述语言，并讨论了相关的逻辑设计问题；接下来作者用实例说明了CISC和RISC的区别，深入剖析了指令集和硬件之间的接口关系，介绍了CPU流水线、多指令发射计算机、微代码控制单元的设计以及算术逻辑处理单元的设计；之后作者详细介绍了存储器的层次化结构设计，并且讨论了机器输入输出系统和外围设备；最后作者讨论了一些计算机互连方面的论题。此外本书提供了一个专门介绍数字逻辑的附录，行文深入浅出，相信对阅读本书很有帮助。

本书可作为高校计算机、电子等相关专业本科生和研究生微机原理、系统结构和计算机设计等方面课程的教材，对相关专业人士和研发人员也很有裨益。

版权贸易合同登记号　图字：01-2004-1485

图书在版编目（CIP）数据

计算机系统设计与结构 = Computer Systems Design and Architecture：第2版/（美）休林（Heuring, V.P.）等著. -北京：电子工业出版社，2004.6
（国外计算机科学教材系列）
ISBN 7-120-00112-4

Ⅰ.计… Ⅱ.休… Ⅲ.①电子计算机-系统设计-教材-英文 ②计算机体系结构-教材-英文 Ⅳ. TP30

中国版本图书馆CIP数据核字（2004）第052970号

责任编辑：赵红燕　贺瑞君
印　　刷：北京兴华印刷厂
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787 × 980　1/16　　印张：38.5　字数：863千字
印　　次：2004年6月第1次印刷
定　　价：55元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

出版说明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

Preface

To the Instructor

appendix on digital logic design

This book is suitable for an introductory course on computer design at the junior, senior, or introductory graduate level. We assume that the student has had at least an introductory course in some high-level programming language such as C or Pascal, and a semester of logic design. However, a comprehensive appendix on digital logic design, written by Professor Miles Murdocca of the Internet Institute USA, provides sufficient background material for teaching the course to students without previous digital design experience.

Appropriate topics for such a book have changed considerably in recent years, as desktop computers have evolved from simple, stand-alone units into complex systems attached to high-speed networks and internetworks. Earlier generations of microprocessors had almost trivial internal structure. Present designs contain multiple pipelined functional units with support for multiple processors and memories. Areas of computer design and architecture that were barely touched upon in the not-so-distant past have become major topics for discussion. Introductory compiler courses now routinely discuss optimization for pipelined processors. Users worry about whether they should add level-2 cache memory to their PCs. Support personnel wearily try to explain to the computer user how to configure the subnet mask for their network slip connection.

pipelined processors

memory hierarchy

networking and the Internet

The topics of pipelined processor design, the memory hierarchy, and networks and internetworking are moving to center stage in the arena of computer design and architecture. Therefore we devoted the major parts of three chapters to treatment of these subjects.

three views of the general purpose machine

Given the focus on computer design and computer architecture, we approach the study of the computer from three viewpoints: the view of the assembly/machine language programmer, the view of the logic designer, and the view of the system architect.

In covering the topic of gate-level computer design, we follow a model architecture through the design process, from the instruction set design level to the processor design level. Given the choice of using either a commercial machine with all of the complicating features that are necessary to make such a machine commercially successful, or using a model design that introduces enough practical features to make it both interesting and relevant to the subject, we chose the latter. The model machine, Simple RISC Computer (SRC), is a 32-bit machine with an instruction set architecture that is similar to, but simpler than the current generation of RISCs.

model machine: Simple RISC Computer (SRC)

formal description language: Register Transfer Notation (RTN)

We adopt the view that it is best to use a formal description language in describing machine structure and function. There are many languages from which to choose. We selected a variant of the ISP language, Register Transfer Notation (RTN). We chose this from many alternatives, because most of the languages used by practitioners are aimed more at hardware description and less at machine behavior and function. RTN is simple, easy to learn, and is at the appropriate description level.

To the Student

The computer ushers us into the information age. Born a mere fifty years ago, it now exerts a profound influence on almost every facet of our lives. What is the nature of this machine? How does it work inside? How is it programmed internally? What is the nature of each of its connections to the outside world? These are the questions that this book will help you answer, and we hope that when you have mastered it you will be left with no mysteries about how computers work. We feel that one of the best ways to learn how a computer works is to design one, so throughout most of the book we have taken the perspective of the designer rather than that of the observer or the critic.

Computers are arguably the most complex systems humankind has ever built, and like all complex systems they can be appreciated from many viewpoints. A building can be examined for its overall architectural design, and for the way its design affects its overall function. A building also can be examined from the viewpoint of how the size and shape of its rooms and halls relate to the design of its heating and air conditioning systems. Likewise a computer can be examined from the viewpoint of *its* overall structure and function, referred to as its architecture. A computer also can be examined from the viewpoint of one who is using machine or assembly language to program it. And it can be examined from the viewpoint of its lowest abstract logical structure—its design at the logic gate level.

All of these viewpoints are interrelated, and therefore important for mastery of the subject; thus in this book we adopt all three: the viewpoint of the computer architect, the viewpoint of the assembly language programmer, and the viewpoint of the logic designer. We believe that the synthesis of these three views will give you a depth and richness of understanding of the subject that will serve you well, whether your main interest is in computer design, computer science, or computer programming.

We assume that you have had experience with computers as an end-user, and that you have written programs in some high-level language such as Pascal, C, or FORTRAN. We also assume, in the body of the text, that you have had exposure to digital logic circuits. A knowledge of logic circuits is necessary for the understanding of the material in this book. For those who have not had such exposure, or who are uncertain about whether their background in this area is adequate, we have included Appendix A: Digital Logic Circuits, which should provide you with sufficient background for understanding this text.

Preface to the Second Edition

It is a truism that computer design and architecture textbooks are out of date before the printer's ink is dry. Such is always the case in a field as dynamic and growing as ours. While this places more burden on the authors to revise, update, and release new editions, it also presents opportunities to introduce new material, new tools, and new or expanded discussions of old topics. All of these factors contributed to the desire to develop a second edition of *Computer Systems Design and Architecture* (CSDA). We have continued to adhere to the principle of "no mysteries" in this edition, especially as it pertains to the fundamentals of computer design.

no mysteries

This approach has meant sacrificing ancillary or overly descriptive materials relating to modern processors to keep the textbook to a reasonable size. It was, and continues to be our philosopy that a thorough grounding in the fundamental principles of computer design rather than a more superficial discussion of the architectures *du jour* will better meet the needs of students to master the architectures of the future.

New in the second edition:

Tools Java-based assemblers and simulators with GUI-based interfaces are now available for SRC, the MC68000, and ARC, a SPARC subset. The simulators run on the PC and Macintosh OSX platforms, and all Unix and Linux implementations that run the Java Virtual Machine. VHDL and LogicWorks models of SRC are also available.

The following new or expanded topics have been added:

All chapters have been brought up to date with current information. There are many new and modified Exercises and Examples, including a new feature, Classic Example.

Chapter 2 Expanded discussion of the timing of bus operations during register transfers, including a timing diagram.

Chapter 2 A greatly expanded discussion of performance measurement and estimation. New material on the Pentium 4 architecture.

Chapter 3 New section on calculating performance/speedup.

Chapter 4 Details of the SRC ALU design. New material on SRC control signals.

Chapter 5 Expanded discussion of instruction-level parallelism, including a VLIW design for SRC.

Chapter 7 Discussion and example of temporal and spatial locality. Discussion and examples of SDRAM and DDR RAM.

Chapter 8 Summary of SRC I/O ports. New section using Venn Diagrams to motivate the discussion of Hamming and SECDED encoding.

Chapter 9 New section on Disk System reliability, including RAID and SMART disk drives.

Chapter 10 New section on Gigabit Ethernet. New section on Modern Serial Buses: USB and Firewire. New section on Wasted IP address space: CIDR classless routing, NAT, and DHCP.

Appendix C: Rewritten to target the SRC assembler instead of the 68K assembler. More detailed discussion of the symbol table and the assembly process.

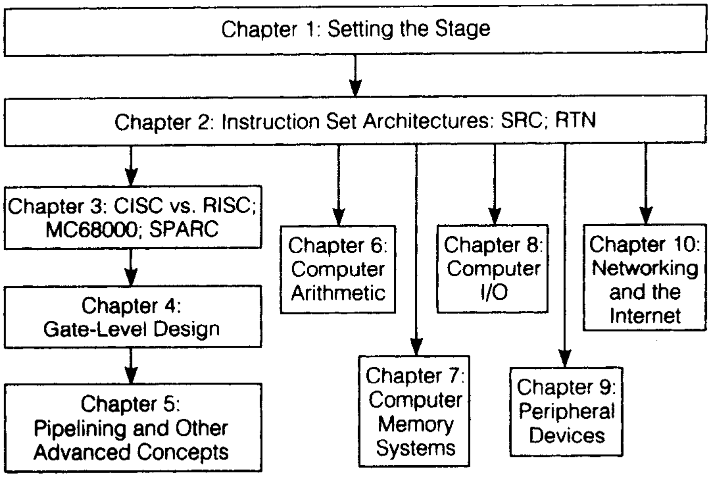
Appendix D: Due to popular student demand, Appendix D, Selected Problems and Solutions, was added. This appendix contains a selection of "classic" problems in computer design and architecture.

Using This Book with Your Curriculum and Syllabus

There are probably almost as many different curricula and syllabi for computer design and architecture as there are schools and instructors in which the subject is taught. As you evaluate the material in this book you may discover the following:

- Some topics may have been covered in a prerequisite course, and so can be omitted.
- Some topics need not be covered because they are covered in a subsequent course or courses.
- Some topics can be omitted or emphasized because of the aims of the course or the instructor. For example, computer science curricula may stress architectural topics and deemphasize the gate-level design material, whereas computer engineering curricula would stress gate-level design.
- Some instructors will choose to emphasize or deemphasize certain topics or change the topic ordering because of their particular philosophy of how the material should be presented.

We have tried to make the topic coverage sufficiently inclusive and self-contained so that you can adapt it to suit your curricula and syllabi. The following figure presents a chart of chapter dependences to aid you in selecting the topics and topic ordering to suit your particular needs. The first two chapters set the stage for the rest of the book, and should be covered first. Chapter 3 treats real machine designs and is essential for any student without a solid background in CISC and RISC instruction sets. Chapters 4 and 5 pursue gate-level design of the CPU. Each of Chapters 6–10 is virtually stand-alone, and can be selected to suit the needs of your particular curriculum and syllabus.


Chapter 1: Setting the Stage
Chapter 2: Instruction Set Architectures: SRC; RTN
Chapter 3: CISC vs. RISC; MC68000; SPARC
Chapter 4: Gate-Level Design
Chapter 5: Pipelining and Other Advanced Concepts
Chapter 6: Computer Arithmetic
Chapter 7: Computer Memory Systems
Chapter 8: Computer I/O
Chapter 9: Peripheral Devices
Chapter 10: Networking and the Internet

Chapter Descriptions

Chapter 1 takes a brief look at the machine from all three levels, and provides an overview on the subject.

Chapter 2 begins with a discussion of the relationship between machines and machine languages—how the nature of the instruction set influences the rest of the CPU structure. At this point we introduce a language for *formally* describing machine structure and function. That is, we provide a way to unambiguously describe both the machine hardware and how instructions run on it. That language, RTN, is a simple one, and it is at the "just right" level to describe the running of instructions on hardware. We introduce RTN by using it to describe a generic model machine, SRC, that has a 32-bit ISA similar to the current crop of 32-bit commercial machines, but without many of the complicating factors. The chapter concludes by switching to the logic circuit level and putting a computer design spin on conventional logic circuit design by a brief discussion of how RTN operations translate into circuit designs.

CISC versus RISC machines

CISC: MC68000

RISC: SPARC

Instruction Set Architectures (ISA)

The goal of Chapter 3 is to provide the student with a concrete understanding of the difference between CISC and RISC machines. It discusses two commercial machines at the ISA (and RTN) level: the CISC Motorola MC68000 and the RISC SPARC. The chapter provides familiarity with several different machine architectures at the ISA level, so that students develop an appreciation for two philosophically different approaches to machine design. We also discuss some practical issues, such as upward compatibility, and how those issues influence machine design.

keystone chapter

Chapter 4 is in many ways the keystone chapter of the book. It describes the interface between the instruction set and the hardware that it runs on, at the gate level. This description unfolds by developing a 1-bus design for the SRC introduced in Chapter 2, with RTN descriptions of machine functioning as a guide. Beginning with the ISA registers, the RTN description adds additional registers that are hidden at the ISA level as it treats how instructions are fetched, decoded, and executed, again at gate level. The discussion proceeds with the soul of the machine

design of the "soul of the machine," the control unit, and its heartbeat, the system clocking and timing. This chapter also begins the discussion of how hardware design influences instruction execution speed by examining alternative 2- and 3-bus designs. Two other aspects of processor design are covered in this chapter: the hardware reset, and the exception process. Our goal in Chapter 4 is to leave no mysteries. At the end of the chapter, the student should understand exactly how the central processing unit works at the gate level.

pipelining and microcoding

instruction-level parallelism

VLIW

superscalar machines

Chapter 5 covers pipelining of the CPU, multiple-instruction-issue machines, and microcoded control unit design. Nearly every current and planned processor design employs pipelining in its CPU, and a thorough understanding of how pipelining works is a necessity for everyone from compiler writers to machine programmers and architects. We first present an overview of the important issues in pipelining and then show the pipeline design process by way of a pipelined design for SRC. A discussion of instruction-level parallelism follows. We treat superscalar operation, where there are multiple functional units in the CPU that are capable of parallel operation, and VLIW, very long instruction word machines, whose instruction words contain a number of processing steps that are executed in parallel. We include an example of a VLIW implementation of SRC. The chapter concludes with a discussion of microcoding. Microcoding is not used much in general purpose microprocessor chips at present, but in addition to its use in fast-turnaround and adaptable special purpose designs, it presents an interesting perspective on computer design.

ALU

integer arithmetic

semi-numeric aspects

floating-point arithmetic

Chapter 6 covers the design of the arithmetic and logic unit (ALU) of the computer. The design of this very important CPU component has a major impact on overall system performance. Since much of the ALU's performance is based on the underlying algorithms that are implemented in the hardware design, the discussion proceeds from data type (e.g., integer), to algorithm (e.g., addition), to the hardware implementation (e.g., a carry lookahead fast adder). Both integer and floating-point data types are covered. A section is included on how branch instructions use the ALU with a discussion of logic operations and overall ALU design. The chapter concludes with a discussion of floating-point arithmetic.

RAM and ROM

chips, boards, and modules

cache memory

the memory hierarchy
virtual memory

Chapter 7 presents the design of the memory hierarchy in detail. Beginning with the simplest 1-bit RAM and ROM cells, the chapter builds those cells into chips, chips into boards, and boards into modules. We discuss the general nature of the relationship between two adjacent levels in the hierarchy, and following that, we discuss cache design and the interaction between cache and main memory. This is followed by a discussion of virtual memory, the process of allowing the memory space to spill out from main memory onto the disk. The chapter concludes with a discussion of memory as a system.

buses and timing
serial and parallel I/O

programmed and interrupt-driven I/O

Chapter 8 discusses the details of the machine's I/O system. It begins with a treatment of several kinds of buses, treating both bus signals and bus timing, and proceeds with a discussion of the two principal kinds of generic I/O interfaces, serial and parallel. The chapter then covers the relationship between the machine interrupt structure and the I/O system, and between these two and DMA, direct memory access, by which an external device can access main memory without CPU intervention.

disk drives

video and other display devices

analog interfaces

Chapter 9 covers the other end of the I/O system: peripheral devices. It treats the structure and functioning of disk drives, video and other interactive display devices, printers, mice, and the interfaces to analog devices. The emphasis is on how these devices actually work, and the nature of the interface between them, the CPU, and the outside world. Peripheral device performance is covered as a main aspect of interest.

Chapter 10 concludes the book with a discussion of computer-to-computer communications. No current treatment of computer systems design and architecture is complete without a fairly in-depth discussion of computer communications and networking. We begin with a discussion of network structure and topology. Following that we present three examples of contemporary machine communications. The first example is the RS-232 serial data communications protocol that permits point-to-point communications between two computers or between a computer and a terminal. The second example is the Ethernet local area network (LAN). We discuss the Ethernet communications protocol at both the physical level and the data link layer, including the Ethernet packet structure, and higher-speed implementations. We also discuss USB and Firewire. The final example of a communications system is the Internet—probably the largest and most important computer communications system on the planet. We discuss the TCP/IP Internet protocol, and Internet addresses and addressing. Also discussed is the wasting of IP addresses by the class A, B, C system, and the use of CIDR, DHCP, and NAT to reclaim them. The chapter, and the book conclude with a very brief discussion of Internet applications and Internet futures.

computers and networking

RS-232 data communications

the Ethernet LAN

the Internet

TCP/IP

Internet addresses and addressing

Instructional Support Materials

For the latest information on these supplements and how to obtain instructional support materials, contact your Prentice-Hall sales representative or visit the Prentice-Hall web site at http://www.prenhall.com.

Solutions Manual The Solutions Manual contains solutions to virtually all end-of-chapter exercises in *Computer Systems Design and Architecture.*

Electronic Lecture Slides This set of approximately 600 slides is available in two formats, Adobe Acrobat and Microsoft PowerPoint. The slides include the book's main points presented in a lecture outline format, and nearly all figures and tables from the text. Using the free Acrobat Reader, the transparencies in the Acrobat format can be viewed and printed in various ways from PC, Macintosh, and UNIX platforms. Instructors who have access to PowerPoint can modify the slides in the PowerPoint format. The slides are available at ftp://schof.colorado.edu/pub/CSDA.

Software Support Tools A growing collection of software support tools are available to adopters by ftp. These tools include Java-based SRC, MC68000, and SPARC subset assemblers and simulators that will run on the PC, Macintosh OSX, and Unix and Linux platforms. VHDL and LogicWorks implementations of SRC are also available. The tools and several other resources are available at ftp://schof.colorado.edu/pub/CSDA.

If You Find an Error

In spite of the good efforts of the authors, editors, reviewers, and class testers, this book undoubtedly contains errors. Please send reports of errors to csdabugs@colorado.edu

Acknowledgments

We also wish to thank our first edition reviewers, who were most helpful in suggesting changes to the original manuscript of this book: George Adams (School of Electrical and Computer Engineering, Purdue University), Dr. Hassan Barada (Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia), Thomas L. Casavant (Electrical and Computer Engineering, University of Iowa), Dr. Peter Drexel (Department of Computer Science, Plymouth State College), Michael A. Driscoll (Department of Electrical Engineering, Portland State University), J. Kelly Flanagan (Computer Science Department, Brigham Young University), Richard Heuring (Cinch Enterprises), William Tsun-yuk Hsu (Computer Science Department, San Francisco State University), Mariusz Jankowski (Department of Engineering, University of Southern Maine), Mike Keenan (Computer Science Department, Virginia Tech), Professor Steven P. Levitan (Department of Electrical Engineering, University of Pittsburgh), Jyh-Charn (Steve) Liu (Computer Science Department, Texas A&M University), Norman Matloff (Department of Computer Science, University of California at Davis), Walid Najjar (Computer Science Department, Colorado State University), Bernhard Weinberg (Computer Science, Michigan State University), D. Scott Wills (Electrical and Computer Engineering, Georgia Institute of Technology), Craig M. Wittenbrink (Board of Computer Engineering, University of California, Santa Cruz), David Wood (Computer Science Department, University of Wisconsin). Also to be thanked are the faculty and students who used the Fall 1995 and Spring 1996 Class Test Editions for their many helpful suggestions and error reports. Faculty were Douglas J. Fouts (Department of Electrical and Computer Engineering, U.S. Naval Postgraduate School), John F. Passafiume (Department of Computer Science, Clemson University). They and their students contributed greatly to the improvement of the book.

We thank Craig Meuller and the Martinique Team at Maxtor for permission to photograph the 1.3 gigabyte Laramie disk drive, and Dr. James Avery for permission to photograph his disk-platter-made-into-a-coffee table (see page 410).

We especially wish to thank Professor Miles Murdocca of the Internet Institute USA for contributing Appendix A: Digital Logic.

Harry Jordan's contributions to CSDA were many and important. He will be greatly missed, both as a colleague and a friend.

VINCENT P. HEURING
Boulder, Colorado

heuring@colorado.edu

Contents