CHAPMAN & HALL/CRC
TEXTBOOKS IN COMPUTING

The TAO of COMPUTING

SECOND EDITION

Henry M. Walker

CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

Copyrighted material

CHAPMAN & HALL/CRC
TEXTBOOKS IN COMPUTING

The TAO of COMPUTING

SECOND EDITION

Henry M. Walker

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

Copyrighted material

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2013 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed in the United States of America on acid-free paper
Version Date: 20120409

International Standard Book Number: 978-1-4398-9251-0 (Paperback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Walker, Henry M., 1947-
The tao of computing / Henry M. Walker. -- 2nd ed.
p. cm. -- (Chapman & Hall/CRC textbooks in computing)
Includes bibliographical references and index.
ISBN 978-1-4398-9251-0 (pbk.)
1. Electronic data processing. 2. Computers. 3. Internet. I. Title.

QA76.W1855 2012
004--dc23 2012010207

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Copyrighted material

Contents

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Section II Software/Problem-Solving Questions

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Preface

WHAT SHOULD EVERY CITIZEN know about computers and computer technology? For better or worse, answers typically depend on who is asked:

- Students commonly are practical: How can one get the computer to perform a specific task? Why does a machine act in the way it does? What is involved in getting computers to interact? And so on.
- Computing faculty typically emphasize the need to understand concepts and problem solving.
- The National Research Council (NRC) has identified appropriate general knowledge and skill as computer fluency and summarized its perspective in a 1999 report, *Being Fluent with Information Technology.*

On the surface, such perspectives may seem unrelated or contradictory. The practical details of interest to students may seem quite different from the high-level concepts and abstractions highlighted by faculty. Some textbooks written by faculty may do a fine job in covering foundational material but often omit the practical issues that motivate students. Other books may describe pragmatic elements about how to run specific software (e.g., how one can utilize a bold type font in a word-processing document), thus satisfying some students, but such books rarely provide adequate coverage of more general and lasting concepts.

Practical questions with thoughtful answers

This book takes a different approach. The starting point is the observation that thoughtful answers to common, practical questions often require an understanding of ideas and principles. Simple, narrow, superficial answers can avoid deep issues, but these are rarely satisfying in the long run. Students want serious answers to their genuine questions, and such responses require the material that faculty wish to cover and the NRC has identified.

Question-based

The second principle motivating this book is that students learn best when material connects with their experiences, backgrounds, and perspectives. To capture student interest, this book is organized around questions typically asked by general computer users. Common general questions provide the title and theme for each chapter, whereas more detailed questions provide a focus for each section within a chapter.

Copyrighted material

The style reflects the circumstances of a conversation. The reader has many questions about computing and information technology, and the book seeks to provide answers. Throughout, the starting place for discussion is a question—often a real question heard from a computer user or raised by actual students in class.

Computer fluency

A third principle behind this book concerns the need to cover adequate material to allow readers to function effectively in today's computer-based society. Such background sometimes is described as "computer fluency." An initial description of relevant topics appeared in the 1999 report *Being Fluent with Information Technology*, which addressed what every informed citizen should know about computers and technology. The report, by a study group of the NRC, identified 10 high-level "intellectual capacities," 10 "information technology concepts," and 10 practical "information technology skills" that cover basic computer fluency.

In the years since the original NRC report, technology and computing have continued to play an important and expanding role within contemporary society. For example, advances in communication, hardware, and applications have had a significant impact on the uses and roles of computers in everyday life, and the concept of computer fluency is at least as important today as it was when the NRC report first appeared.

Need for a second edition

Although the first edition of *The Tao of Computing* (2005) covered vital topics of computer fluency at that time, technology and computing have evolved in the intervening years, and the needs of general citizens similarly have progressed. Considering contemporary themes and perspectives, three main factors are the motivation for this revision:

- *Technological advances:* Technology, particularly computing technology, evolves at a very rapid rate; many details and some concepts have changed since the first edition appeared. Computers are several times more powerful now than in 2005; data storage for audio and video has expanded dramatically; and networking is much more extensive.

- *New applications and shifting popular culture:* Many new applications, examples, and experiences have emerged in recent years. For example, in 2005, facebook.com had been operating for less than a year; both myspace.com and twitter.com would not appear until the following year; and RSS feeds (Really Simple Syndication for Web news reports) were just achieving widespread use.

- *Student feedback:* Since the first edition appeared, others and I have used *The Tao of Computing* in a range of courses. Often, in preparation for a class, I asked students to propose additional questions and comments for discussion. This feedback and brainstorming have suggested new questions for consideration and helped identify topics for expanded and refined discussion.

Copyrighted material

Overall, although many underlying concepts continue today from 2005, numerous details, applications, and examples from the first edition are now dated. This second edition provides an opportunity to update the earlier material. At the same time, this revision allows for expansion of teaching and learning aids throughout the book.

Key features of this book

This second edition of *The Tao of Computing* utilizes the style and format of the first edition while updating content and expanding pedagogical features.

- *Thoughtful answers to practical questions:* Discussions in this second edition follow from practical questions. Chapters and sections within a chapter begin with a question such as "how does this work?," "why does this application behave this way?," "how might my use of this technology impact me?," "what risks might be found with this application?," and so on. Although many of these questions seem simple to state, the answers often require background knowledge and exploration of technical concepts and approaches.
- *Extensive updating of material:* This second edition reflects many of the technical advances in the field since the first edition appeared.
- *Modest expansion and reordering of some topics:* Student interests and feedback suggested expanded coverage of images and data formats, and this material is now covered in two chapters. Also, the evolution of technology motivates some restructuring of several topics from the first edition to this second edition, and several chapters have been reordered.
- *New and expanded examples:* Even when modern concepts and techniques come from past understandings, many applications are new. Thus, each chapter contains updated descriptions, examples, applications, and technical details.
- *Expanded discussion questions and exercises:* The number of questions for class discussion and the number of exercises have almost doubled for most chapters.
- *New research exercises:* In recent years, many people have become reasonably proficient in using search engines to find information about various topics. However, much information on the Web is unfiltered and not reviewed. Thus, the reliability and accuracy of Web-based materials can be rather spotty, and people need to practice analyzing materials as well as finding them. To help promote this critical review of materials, this second edition contains Research Exercises at the end of each chapter. Each Research Exercise asks a question that the reader is encouraged to research online, in a library, or through other sources. Although questions may appear to be straightforward, many include twists or subtleties that challenge readers. Thus, reporting and analyzing answers can be a particularly valuable mechanism to help sharpen research skills.

Copyrighted material

Overall, although many underlying concepts continue today from 2005, numerous details, applications, and examples from the first edition are now dated. This second edition provides a significant expansion and updating of the earlier material. At the same time, this revision allows for expansion of teaching and learning aids throughout the book.

Online materials supporting this second edition

Color versions of many pictures in this book are available on the World Wide Web at http://www.cs.grinnell.edu/~walker/fluency-book-2/. Some laboratory exercises also are available at this site.

Credits

Many passages in this book are edited versions of sections of my book *The Limits of Computing*, published by Jones and Bartlett in 1994. Part of Chapter 12 is a revised version of material from *Abstract Data Types: Specifications, Implementations, and Applications* by Nell Dale and me, published by Jones and Bartlett in 1996.

Except as noted, I took all photographs in this book. Also, unless other credits are given, all figures in this book were redrawn in Adobe Illustrator by Allison Vosburg, Academic Support Assistant at Grinnell College.

I gratefully acknowledge the permissions received from individuals and publishers for the following figures and passages.

Chapter 1

Figure 1.5 has been taken from B. Burgess et al., "The PowerPC 603 Microprocessor," *Communications of the ACM* 37(5) (June 1994), 35, Figure 1. © 2002 ACM, Inc. Reprinted with permission.

Figure 1.6a was photographed for this book by Ed Dudak, Physics Technical Assistant at Grinnell College.

Figure 1.7 has been taken from Intel. Reprinted with permission.

Chapter 3

Figure 3.1 was developed for this book by Fredrick Hagemeister, curricular technology specialist for the sciences at Grinnell College.

Figures 3.5, 3.6, 3.8, and 3.10 are pictures in the public domain. Figure 3.5 was taken by C. Clark, NOAA Photo Library, NOAA Central Library; OAR/ERL/National Severe Storms Laboratory (NSSL). Figure 3.6 was photographed by John and Karen Hollingsworth, U.S. Fish and Wildlife Service. Figure 3.8 was taken by Joseph O'Brien, plant pathologist, Forest Health Protection Unit, U.S. Forest Service. Figure 3.10 was taken by Richard B. Mieremet, senior advisor, NOAA OSDIA—dated 1997 from the NOAA Photo Library, America's Coastlines Collection, National Oceanic and Atmospheric Administration/Department of Commerce.

Copyrighted material

Chapter 6

Quotations on computer fluency come from *Being Fluent with Information Technology* by the Computer Science and Telecommunications Board of the National Research Council, National Academy Press, 1999.

Chapter 7

Figure 7.2 was created by Doug Peterson, technical support assistant at Grinnell College, on the basis of data supplied by the author.

Chapter 10

Figure 10.4, the RSS logo for syndication, has been made available for general use by the Mozilla Foundation (http://www.mozilla.org/foundation/feed-icon-guidelines/).

Chapter 12

The photograph in the sidebar of Bellcore's Telephone Chords and the text of the first MIME message come from Nathaniel Borenstein. Used with permission.

Chapter 13

Figure 13.3 is a screen shot of an application created by Grinnell College.

Chapter 15

The text of the Resolution on the Use of Filtering Software in Libraries and the CIPA Questions and Answers passage comes from the American Library Association.

Chapter 16

Quotations on finding and evaluating information comes from materials written by Kevin Engel, Science Librarian, Grinnell College, and are used with permission.

The statement of "Categories and Levels of prepublication evaluation" comes from http://www.acm.org/publications/policies/prepub_eval by ACM, the Association for Computing Machinery, and is used with permission.

Chapter 17

Figure 17.1 was edited by Stephanie Peterson, Curricular Technology Assistant at Grinnell College, on the basis of a photograph supplied by the author.

Copyrighted material

Copyrighted material

Acknowledgments

THIS BOOK GROWS OUT of countless conversations and communications with computing faculty and students over the years. Computing educators frequently share ideas, collaborate on projects, cooperate in planning curricula and course, and brainstorm refinements to pedagogy. Special thanks to members of the Special Interest Group on Computer Science Education (SIGCSE) of the Association for Computing Machinery (ACM) and to members of the Liberal Arts Computer Science Consortium (LACS). Both groups have had a profound impact on my understanding of computing and computing education.

Naturally, this second edition began with the development of the first edition. I thank Jones and Bartlett Publishers for their encouragement and professional assistance in bringing the first edition to completion. I also thank Tim Anderson, editor, for his assistance in facilitating the transition from the original edition at Jones and Bartlett to this second edition with Chapman & Hall/CRC Press. Special thanks to Allison Vosburg for merging the first edition with the updates I prepared for this second edition. In the process, she typed most of the entire revised manuscript, reformulated many of the figures, and provided invaluable assistance in moving this project along.

Thanks also are due to Sara Adams for providing detailed feedback on several chapters; her comments have helped in the refinement and polishing of the text. Additional thanks are due to reviewers Kevin McDonnell, Dowling College, and Steven Huss-Lederman, Beloit College, for their feedback on a first-revised manuscript and for their insights for refining both the content and the visual format of this second edition. Through both editions, members of Grinnell College's Information Technology Services Department provided substantial assistance in collecting and preparing cables and computing equipment that I could photograph for many of the images in this book.

I also greatly appreciate the support of Grinnell College for both editions of this book. The first edition benefited from the college's senior leave program and its designation of the author as Frank and Roberta Furbush Faculty Scholar in Mathematics and Computer Science for 2002–2003. The second edition benefited from the college's sabbatical leave program.

In developing and producing this second edition, I also extend great appreciation to Chapman & Hall/CRC Press for their enthusiasm, energetic support, helpful feedback, and thorough professionalism. Special thanks to Randi Cohen, acquisition editor, and John Impagliazzo, Textbooks in Computing Series editor, for their ongoing support for this project. Thanks also to Kevin Craig for his inspired design of the cover, and

Copyrighted material

additional thanks to all those who worked in the production of this book, especially Ed Curtis, project editor for the Taylor & Francis Group, and Syed Mohamad Shajahan, project manager for Techset Composition Ltd.

Finally, I thank my family for their ongoing encouragement, support, patience, and perspectives. Over the years, I have tried to balance professional activity with family and personal relationships, and any balancing act has its strong and weak moments. Throughout, my wife Terry, daughter Donna with Jeff, Jackson, and Hannah, and daughter Barbara have provided continual support while tolerating my bursts of enthusiasm and disappointments—they even understand my own unique perspectives on words, circumstances, and philosophy! Overall, family provides a framework and anchor for my life, and that context provides meaning, focus, and context for all my endeavors.

Copyrighted material

I

Underlying Building-Block Questions

We begin our study of the tao of computing by cutting through the complexity of modern computers, so we can focus on the foundational elements that underlie much computer processing. Section I addresses the basic question, "What are the basic building blocks of a computer?" The overall answer involves four pieces:

- Hardware
- Data
- Computer programs
- Operating systems

Within this section, Chapter 1 considers questions about hardware, the physical components of a computer that we can hold in our hands: electrical components, video monitors, keyboards, audio speakers, etc.

Information includes numbers, text, images, and sound; but in what form are these data stored? In many applications, we may be able to focus on the information itself, and let the computer store data in whatever format seems convenient. However, sometimes the way data are stored can impact what we can see, hear, or do. Chapters 2 and 3 address questions related to the way information is represented; Chapter 2 focuses on nonpictorial information, whereas Chapter 3 considers the representation and storage of images. These two chapters also consider questions about how the representation of data can impact our work with those data.

In addition to the format or representation of information, questions arise regarding where that data might be found. For example, we may edit a document and turn the machine off. When we return, our changes seem not to have been made, and we wonder, "What happened?" Chapter 4 addresses this situation by considering questions about how data and applications might be stored and how storage might impact our use of computers.

Copyrighted material

When we use computers for email or word processing or accessing the World Wide Web, we typically click a few buttons or type some characters to get started. After some work, we may want to save some information or print a document. But how does the computer know how to interpret our clicks or typing, where to store or retrieve information, or how to move the paper or form the characters when printing a document? The basic answer is that many such details are handled by something called an "operating system." Chapter 5 addresses questions about what an operating system is, what it does, and how it works.

Altogether, Section I considers questions about the underlying framework of a computer: what happens behind the scenes to make computers work. Later sections will consider how high-level activities and processing build upon these basic components.

Copyrighted material

CHAPTER 1

How Are Computers Organized?

Computers pervade many of our everyday activities. We use a computer when we withdraw money from an ATM, make purchases with a credit card, and check our voicemail. In the world around us, computers assist people in guiding airplanes, designing cars, stocking store shelves, and maintaining school records. We use computers to communicate with e-mail, to find information through the World Wide Web, and to type papers. We may hear people blame the computer for errors, delays, or foolishness.

In much of our contact with computers, we simply may follow directions or accept whatever we encounter. However, from time to time, we also may wonder how all of this technology works. For example, when we insert CDs or DVDs into disk drives or drag our mouse over icons displayed on our monitor screen, how does a computer know what to do?

This book is designed to answer many of your questions. After describing the book's question–answer style, we move on to establish some common terminology. For better or worse, conversations on any subject require all parties to have a common frame of reference, and this includes an understanding of common words. With this understanding, we will look at computers from the most basic level: their organization.

As an interested computer user, how will this book help address my curiosity about how computers work, what they can do, and what they cannot do?

As a person who uses computers, you naturally are curious; your experience likely has generated many questions regarding computers. You also may have overheard others talking or heard various claims or read advertisements, but wondered what they were talking about or whether their claims were actually true. The purpose of this book is to address your questions. Think of sitting down with a tutor/friend for a series of conversations. You ask questions, and your friend supplies the answers. Sometimes your questions may seem simple, but the answers actually require considerable background. At such times, your tutor/friend may need to provide more information than you expected. Often the

Copyrighted material

underlying answers will require your friend to introduce computer concepts and show how these concepts work in practice. At other times, answers may be short and simple. Such is the nature of genuine discussion. Of course, because this book is printed rather than in an interactive, oral format, I needed to anticipate your questions and to organize questions and answers into a logical framework. In what follows, the questions often were suggested by contact with computer users or during classes with students. Thus, the discussions aim to address real concerns by real people—even though the ordering of questions may be different from what you might generate yourself.

What is meant by "computer programs," "software," "data," "computer applications," and "processing," and how do they differ?

Each of these terms arises frequently in conversations about computing, so it is particularly helpful to understand these words at an early stage.

For a computer to perform any work, it needs instructions to follow. Contrary to popular images from science-fiction movies, such as C3PO in *Star Wars* or David in *A.I.*, a computer has no insight or intuition. Instead, it must be told how to accomplish every task, and these instructions must be given in great detail. These instructions are called **programs**; a program is simply a detailed sequence of instructions that tells a computer how to perform a task. As an example, when we use a computer to help type a paper, the computer utilizes a program that contains instructions to place words on a screen and to move the cursor from one line to another. As a non-computing example, consider what happens when an inexperienced cook follows a detailed recipe in a cookbook to prepare a dish (e.g., cookies, a meal's main course, a stew, etc.). In food preparation, the recipe indicates exactly what work needs to be done; these step-by-step instructions correspond to a program that the novice cook will follow. Since the cook has little experience, the cook will not deviate from the instructions; rather the cook will proceed step by step, and if the recipe was well constructed and complete, the cook would conclude with a culinary treat.

In contrast to programs, **data** are the pieces of information that are handled according to the instructions in a program. As you will see in Chapter 2, data may appear in a variety of types, including numbers, characters, and graphical images. To continue the document-typing example, the characters typed by a user constitute one type of data. When the user types on a keyboard, the user is entering data. In addition, a program, such as Microsoft Word or LibreOffice, may include further information or instructions about how to interpret that user data. Other data might include font descriptions that give information about the form of each character—perhaps in normal appearance, italics, or bold. As this example illustrates, data may come from either of two initial sources: the user or the application itself. To explain these sources further, data coming from a user consists of information exactly as typed. Application data includes data supplied by a program. This might include information about initial fonts or spacing in a document. If the user specifies what type of font to use, then obviously that information has the user as its source. However, if the user does not supply that information, then the program must make some choices regarding font and margins, and that default information may be added to information supplied by

Copyrighted material

the user. As it runs, a program also generates additional data to create the desired final results.

Combining these ideas, **processing** is the term used when a computer follows programs in working with data. Thus, an application that allows us to type and format documents is sometimes called a word-processing program, because the application contains a range of instructions to format our data as we designate.

Interestingly, over the years the distinction between a program and application data has blurred somewhat. Often in early computer work, special application data were written directly into a program. A program was written specifically for a certain task. For example, the act of processing text might have involved the use of only one type of font, and its details may have been an explicit part of the program. Further, the length of a line on the page might have been specified as 80 characters. Other type fonts and font sizes were not available, so there was no need for a program to accommodate other possibilities.

However, as applications have evolved, they have become more flexible. To accommodate various alternatives (e.g., different type fonts), various details are now stored as data. Program instructions continue to tell the computer what to do, but processing may be adjusted according to the data actually encountered. Thus, in today's computing environment, a **computer application** includes both the program(s) and the application data that it relies upon. When such an application is purchased or installed, it typically contains several files, including both the programs and the special application data. For example, when you download and install Adobe Acrobat Reader (to read PDF files from the World Wide Web), you download the basic viewing software, plus quite a variety of special-purpose programs to handle such tasks as working with Web browsers, sending e-mail, interacting with movies, reviewing spelling, and using security options in electronic commerce. Such packages are common when an application includes many functions, each of which come with one or more files.

How do computers work?

This is one of those short, simple-sounding questions that requires an extremely long answer. Consider the next few paragraphs a first, preliminary answer; consider the entire book as a more complete response.

Conceptually, computers may be considered as simple, elegant machines, despite the web of wires and circuits you might see if you open up the body of your computer. When computers perform a task or process information, they rely on only a few fundamental parts of their system. The connections among those components are simple, and data flow easily among the various pieces.

In practice, this simple picture of a computer is complicated by the need to address serious constraints, such as cost and speed. To be useful, a computer must be affordable; expensive technology might allow the construction of fancy or powerful computers, but such machines would not be helpful if they cost too much. Similarly, computers must work fast enough to meet our needs. We may not care if computers take a few minutes to print our papers, but we would not use them if the printing took weeks or months.

Copyrighted material

Real machines use a large range of technological tricks to gain high speed and capability at a relatively low cost. Although this technology improves performance for many applications, the resulting computers no longer achieve conceptual simplicity and elegance. For example, as we will explain in more detail in Chapters 2 and 3, the technology for storing information within a computer usually either facilitates fast processing or holds large quantities of data—but not both. Thus, real computers combine technologies for data storage; this hybrid approach accommodates reasonably large collections of information, but also has inherent processing delays. Such trade-offs are a constant challenge to computer developers, and sometimes simplicity is sacrificed for the sake of performance.

To further address this basic question of how computers work, the next questions and answers discuss a conceptual view of computers by looking at their most simple and basic structure. Later questions and answers review various technological additions that have been made to improve the cost and speed of computers.

What are the main components inside a machine?

Whenever you use a computer, the results you see come from a collaboration of many behind-the-scenes pieces. For example, when you open or save a document, you may hear the whirring sound of a disk drive or CD used for long-term storage. Other pieces generally are quiet in their operation, but their contributions are essential in getting work done.

The basic operation of a simple computer involves only three components:

- The **Central Processing Unit (CPU)**, which directs all operations in the computer and performs the processing steps for the data.
- The **main memory**, which holds the data during most of the processing.
- **Input/Output (I/O) devices**, such as keyboards, monitors, printers, and disks, which allow data to move in and out of the main memory.

Example: To clarify how these basic components interact, consider an application that adds two numbers entered at a keyboard and displays both numbers and their sum on our computer monitor.

Both our keyboard and our monitor are I/O devices. To get our numbers into the machine:

1. We type a number on our keyboard.
2. After determining the numbers have been entered, the CPU tells the keyboard to store the numbers at a designated place in the main memory, and the CPU also directs the numbers to our monitor so we can see what we have typed.

To summarize so far, we type our numbers on an I/O device (the keyboard), and the CPU guides the values toward the main memory and our monitor. The next task in our calculator example involves processing our data, and all such processing is done in the CPU itself. To perform such processing within the CPU, data must be stored in special, high-capability

Copyrighted material

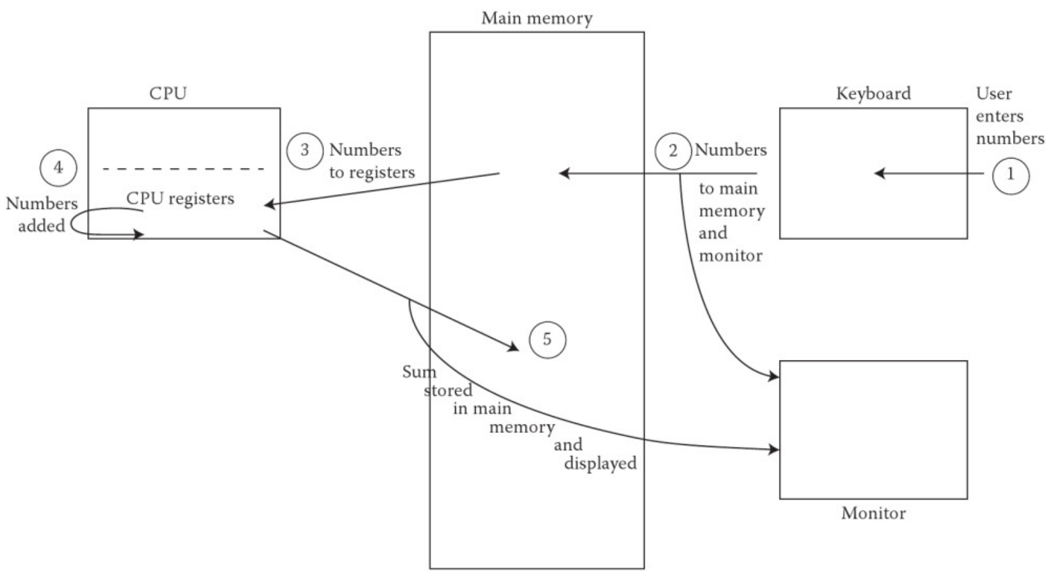

FIGURE 1.1 Reading, adding, and displaying numbers.

storage locations called **registers** within the CPU. Although details vary somewhat from machine to machine, data processing generally follows these steps.

3. The values are loaded from the main memory into CPU registers.
4. The computer then adds the two values in the CPU with the result going to a CPU register.
5. The CPU oversees sending the result from the CPU register to a location in the main memory and then onward to the monitor for display.

In steps 3–5, the numbers move from the main memory to CPU registers, so that processing can take place, and the results then move back to the main memory and our monitor. In practice, each of these main steps requires several small, detailed operations. Throughout, except for the physical operation of input (typing) or output (forming characters on the screen or on paper), the CPU controls every activity. Further, the CPU does the actual data processing (e.g., addition). To support such processing, the main memory holds data both before and after the CPU does the actual work. Schematically, this entire processing of entering, adding, and displaying numbers is shown in Figure 1.1.

Sometimes I hear people talk about circuits and computer chips. How do circuits and chips relate to components, such as a CPU, main memory, and I/O devices?

At a basic level, computers consist of millions (or billions or more) of electrical circuits. An **electrical circuit** is a path that electricity follows through a device. For example, when you plug in a coffee maker, you can think of electricity flowing from the wall outlet, through a

Copyrighted material

wire, to the coffee maker, back through a second wire, and to the outlet. This path of the electricity is a circuit.

Next, consider plugging a lamp into an electrical outlet. Again, there is a circuit; electricity flows from the outlet along one wire through the light bulb and back along the other wire to the outlet. However, in many cases, this circuit has an additional element: the light switch. Electricity flows (and the bulb radiates light) when the switch is on; no electricity flows (and the bulb is dark) when the switch is off.

Overall, basic computer hardware is made up of circuits, and much processing within computers comes about by controlling these circuits through switches. When electricity flows along one collection of circuits, processing proceeds in one way; when electricity flows along other circuits, processing proceeds in another way. One of the breakthroughs of early computers, therefore, was to create small, simple mechanisms to perform switching. In modern computers, an important type of simple switch is called a **transistor**; a transistor provides a way to control the flow of electricity within a circuit.

Of course, by itself, a single transistor cannot do much: it only controls one simple circuit. The power of computers arises because many transistors can be placed together into a single package for various purposes. This combining of multiple transistors into a single entity is called an **integrated circuit (IC)** or an **integrated circuit chip (IC chip)** or just a **chip**. That is, an IC chip is a collection of circuits, controlled by transistors, and designed for various purposes.

With this background, we may consider a CPU as one type of integrated circuit. The main memory often has several different elements that are made out of chips. For example, several chips may be mounted on a single board to constitute a block of main memory; each chip can store some information, and the collection of chips can store a substantial amount of information.

How are the components connected within a computer?

As Figure 1.1 suggests, each component of a computer has circuitry to perform a specific task. These individual components, built from chips, need to communicate on at least two levels:

1. All components must be able to send signals or status reports to the CPU and receive commands from it.
2. Data must be able to move between components.

Through this process, one can think of a signal or status report to the CPU as being analogous to raising your hand in class when you want to participate in a discussion or ask a question. In the case of a keyboard, the signal or status report may indicate that the user has typed something—perhaps after a long wait. For a printer, the signal may indicate that the paper has moved to the right place for actual printing to occur. Another type of signal or status report might indicate that the printer is out of paper. The details of signals or status reports can be tricky, but the idea is fairly simple.

Copyrighted material

In practice, individual wires, called **control lines**, may be adequate to handle simple status reports and CPU commands. However, to promote speed in data communication, multiple wires are used. For example, several characters may be sent from the main memory to the CPU at once. For convenience, both control lines and data-communication wires often are packaged together—sometimes in a configuration that looks something like a ribbon (Figure 1.2).

Continuing the analogy of raising your hand in class, the instructor might ask students to raise their left hands if they want to participate in discussion and raise their right hands if they want to ask a question. Students who have nothing to say would not raise either hand, and students who want to discuss one topic and also have a question on something else might raise both hands. As another example, in planning a future meeting, committee members might raise their left hands if they are available to meet the next morning and their right hands if they could meet the next afternoon. In these examples, an instructor or committee leader could ask two rounds of questions—one to determine who wants to participate in discussion and a second to determine who has questions, or one to determine availability in the morning and another to determine availability in the afternoon. However, both situations can be identified at the same time, if the raising of the left and right hands have different meanings.

Another practical consideration is how to make the physical connection between the parts of a computer. For example, initially, one might consider connecting the CPU to the main memory and to each I/O device by a separate wire ribbon. However, even with only a few such devices, the collection of such ribbons would become unwieldy. Instead, most computers connect several components to a single ribbon of wire, called a **bus** (Figure 1.3).

As a rough analogy, consider how roads are laid out for travel among cities. Commonly, a limited access highway (e.g., a freeway, interstate highway, toll road) connects several cities, and there is an exchange for vehicles to enter and leave the highway at each city. In this setting, one road (like the bus in a computer) serves as the common mechanism that connects the cities; cars enter the highway at one place and exit at another place, There are rarely separate roads between every pair of cities but rather cars share the same road in

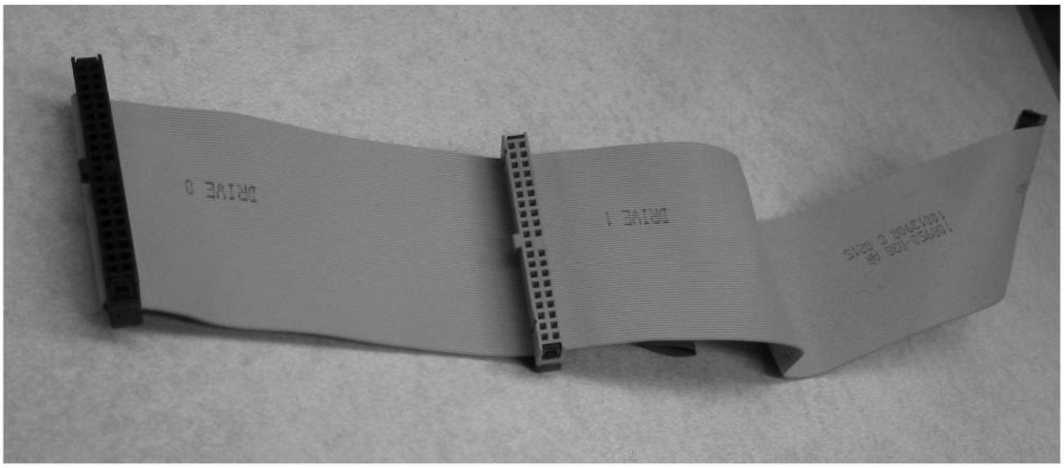

FIGURE 1.2 A ribbon of parallel wires with a connector at one end.

Copyrighted material

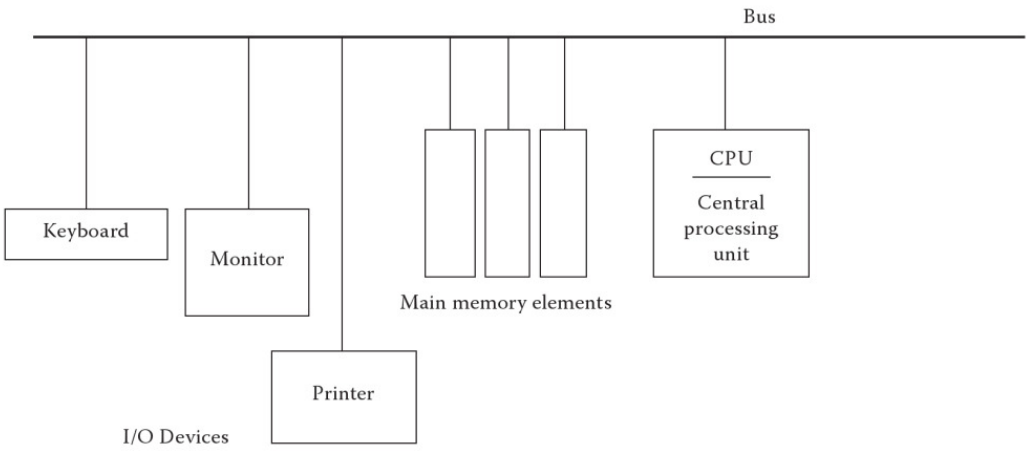

FIGURE 1.3 Diagram of multiple components connected via a bus.

their travels. This analogy does not work completely, in that cars only use the highway from one exit to the next whereas electrical signals along a bus can be detected by all components connected to the bus. However, this analogy is useful in envisioning how a common medium (the highway or the computer's bus) might be used to help move entities (vehicles or data) from some places to others.

Utilizing a bus, each component requires only a single connection point. For a data movement, one component copies the relevant data onto the bus, and the second component stores this information. In addition, a bus likely contains control lines to facilitate the sending of signals between components, so each component knows what to do when.

For the most part, the use of a bus provides several important benefits:

- A single collection of wires can be used for all data and control communication.
- Each component connected to a bus need know only how to interpret its own commands and how to move data to and from the bus.
- Numerous components can be combined in the same machine with minimal interference.
- New components can be added to a machine over time—as long as there is a place to connect them on the bus.

As an aside, note that the ability to add new components to a bus comes into play when considering the upgrade of a current machine or the purchase of a new one. Many computers come with **expansion slots** that provide additional connecting places to an external bus, allowing connections for future components. Figure 1.4 shows one common type of expansion slot, called a **peripheral component interconnect** or **PCI expansion slot**; the parallel wires of the bus are located underneath the slots, and thus are not visible. When a computer has such connections available, adding new capabilities can be easy—perhaps as

Copyrighted material

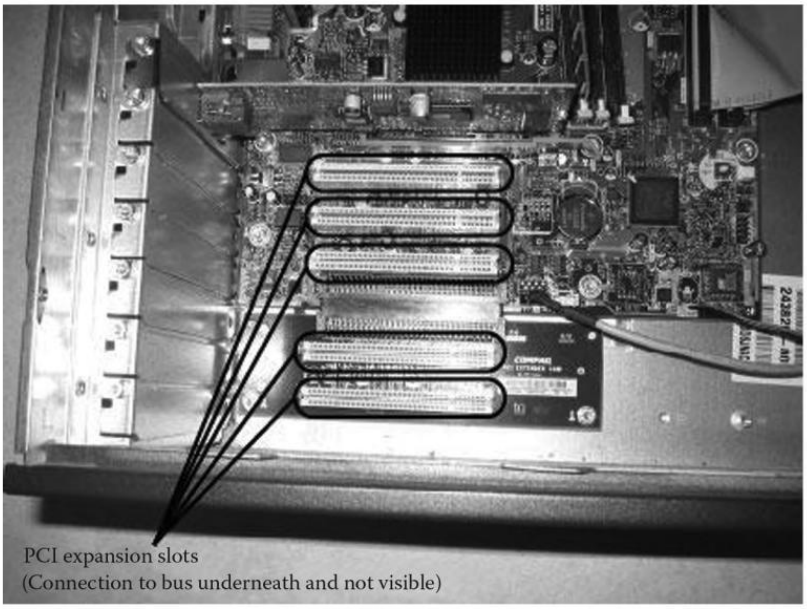

FIGURE 1.4 Peripheral component interconnect (PCI) expansion slots connected (underneath) by parallel wires.

simple as just plugging in the new hardware. However, when all bus connections are already in use, there may be no free connecting points, making expansion difficult or impossible.

Beyond expansion slots, some types of equipment can be plugged into a computer through sockets in the computer's case. For example, a digital camera, additional memory, or an additional disk drive might be connected to a computer through a USB socket. (Chapter 9 discusses USB sockets and other connectors in more detail.) Some modern computers have relatively few of these sockets, in which case the addition of new components outside the computer case may be difficult or impossible. However, other modern computers have numerous sockets, and these may provide considerable flexibility in attaching multiple devices to the machine.

Overall, when purchasing a machine, a buyer may want to consider what expansion is possible. Typically, one can expect new devices and capabilities to enter the market over the lifetime of a computer. These can be added to an existing machine if it has some appropriate free expansion slots or sockets—perhaps extending the useful lifetime of an existing machine.

Can we connect a CPU, main memory, and some I/O devices using a bus to obtain a good computer?

Although the basic CPU–memory–I/O–bus model is sufficient to run most computer programs, the applications may run very slowly. In many cases, this poor performance has three basic causes:

1. The movement of data from one location (such as the main memory) to another (such as the CPU) takes time.

Copyrighted material

2. Connection of components with a single bus limits data movements to one at a time.

3. The CPU coordinates all activities and performs all processing, so any processing must wait until the CPU can monitor or handle it.

To clarify each of these factors, let us revisit the processing outlined in Figure 1.1. In that example, we entered data through our keyboard, the data flowed from the keyboard to the main memory and our monitor, on to the CPU, then back to the main memory with the processing results, and finally to our monitor where we could view the answer. Each of these data movements takes time, and the first possible cause of poor performance is data movement. Second, we entered the two numbers through the keyboard, but these numbers likely had to be sent over a single bus. Thus, time was needed to send one number and then more time was needed for sending the second number. This need to share a bus is the second possible cause listed for poor performance. Finally, behind the scenes through the entire example, the CPU was monitoring and coordinating activity. If the CPU were also coordinating the printing of a document and helping us explore the World Wide Web with our browser, then the CPU might be slowed down in its supervision of the task in the example of reading, adding, and displaying the numbers. This need for the CPU to monitor many different tasks is a third possible reason for poor performance.

With these three potential causes of poor performance within a computer, let us examine each element separately. The first cause, movement of data, is related to the more general topic of latency.

Latency

All activities within a computer take some time to complete. For example, a CPU may request a piece of data from the main memory, but it takes time for the main memory to retrieve that information and make it available on the bus. As a different example, the addition of two numbers within the CPU requires time. We might think of these time delays as being insignificant by themselves—and they may be in some circumstances—but in other cases, such delays combine to consume considerable amounts of time. More generally, any work within a computer takes some time, and this time delay is called **latency**.

Perhaps the most obvious approach to improve performance involves reducing latency in the various components. For example, the main memory is accessed frequently during most processing. Thus, speeding up how quickly memory stores and retrieves information (reducing latency) would likely make overall processing go faster. Computer equipment is sometimes called **hardware**, and hardware developers constantly seek ways to reduce the time required to move data and process information.

A second approach to improve performance takes advantage of new technology that makes circuits small and allows a high density of circuits in a small area. With this technology, modern technology can cram much more circuitry into a space than was possible with earlier technology. Pragmatically, this means that more data can be stored on a single

Copyrighted material

integrated circuit or chip, and more processing capability can be packed into a CPU chip. Further, modern techniques allow the reliable manufacture of chips that are substantially larger than has been possible in the past. In practice, this miniaturization has at last two important consequences that reduce latency:

- Data are physically close together, so travel time from one location to another can be short.
- Much data can be stored on a single chip, so storage of data can involve fewer chips than was possible in the past; often data movements can be within a chip and do not require an external bus from one chip to another.

A third approach to increasing computer performance builds on the observation that although processing may require much data over time, relatively few items are usually needed at any given moment. The idea is to keep current data very close to the CPU, where it can be recalled quickly when needed. Such high-speed memory near the CPU is called **cache**. Cache normally is rather small, but quite effective. (If you have worked with a Web browser, you might have encountered the notion of cache for keeping track of recent Web pages. Although that disk cache follows a similar philosophy as cache for high-speed memory, the discussion here involves different technology than a Web cache.)

SEVEN APPROACHES TO BOOST HARDWARE PERFORMANCE

1. Utilizing new technology to speed up circuits.
2. Moving data elements close together to minimize travel time.
3. Supplementing the main memory with high-speed memory, called **cache memory**.
4. Revising the nature of a CPU to minimize data movements.
5. Combining elements within a single CPU chip.
6. Using multiple busses, sometimes called **channels**.
7. Developing alternative mechanisms for CPU processing.

The idea of main-memory cache is that the first time a piece of data is needed for processing, that information must come from the main memory—but a copy is placed in cache. When the same information is needed next time, it may be retrieved from the relatively fast cache rather than the slow main memory. As this scheme may suggest, cache can work only when coupled with an effective bookkeeping system, so the location of a piece of data (in cache or main memory) can be identified quickly. Without good bookkeeping, identifying the location of data could take as long as getting the information directly from the main memory—thus undermining any advantage cache might have. In practice, such matters can be handled well most of the time, and a CPU will usually find most of the data it needs in the cache memory. Thus, one section of the PowerPC 603 chip includes a cache memory containing a few thousand data items. The newer Intel i7 processor chip carries this idea further, with three levels of cache memory for a core processor. In particular, an i7 core processor maintains 32 thousand high-speed memory cells (called *cache level 1*) for

Copyrighted material

recent instructions it has used, 256 thousand memory cells (called *cache level 2*) for recent data, and 8 million memory cells (called *cache level 3*) for fast general access. Although data still must move between memory and the cache areas of either the PowerPC 603 or i7, the availability of the cache is very effective in allowing these CPU chips to process information efficiently.

As a fourth approach, one might try to increase the amount of memory (the number of registers) within a CPU, so there is less need to move data repeatedly between the main memory and the CPU. Unfortunately, providing full processing capabilities to memory cells in the main part of a CPU is quite expensive and technologically complex. For example, even as sophisticated a processor as the Pentium CPU chip has fewer than 40 full-capability storage locations.* Some other types of processors (called **Reduced Instruction Set Computers** or **RISC** processors) can achieve as many as 64 or so such storage locations, but even then the numbers are modest.

Yet another approach to increase performance is to combine components or processing elements within a single CPU chip. Already, we have noted that the Intel i7 chip contains several different areas for cache memory. In addition, the i7 chip contains four CPUs and some areas for controlling the bus and other I/O tasks. Although we will discuss this in more detail shortly, the main point for now is that modern CPU chips may integrate several components, allowing them to communicate without relying upon the main bus that connects the CPU chip with the main memory and other devices.

Busses

As data flow from one component of a computer to another over busses, improvement of bus technology can have a substantial impact on the flow of data and the corresponding processing. Hardware developers utilize at least two approaches:

- *Maximizing the number of parallel circuits on a bus*: In the early 1980s, early personal computers utilized busses with 8 parallel wires. By the mid-1980s, busses commonly contained 16 parallel wires to transfer 16 different circuits of information concurrently. By the 1990s, busses contained 32 parallel wires, and by the early 2000s, 64 parallel wires were common. With more wires, a bus can transfer more data in a given time interval.

- *Maintaining several busses to allow independent movement of several non-related data elements at once*: Technically, this approach can be quite challenging, because activity along one bus might need to be coordinated with other activity along another bus. However, recent advances now group together three 64-wire busses under the guidance of a single memory controller. In this context, the overall bus for the Intel i7 chip contains 192 parallel wires, organized into three busses (now called *channels*) with a single memory controller.

* Technically, a complex CPU like the Pentium has different types of registers for various specialized tasks, so a count of registers depends on which tasks you want to include. For the Pentium, most register counts would be 8, 16, 24, or 32.

Copyrighted material

In addition, hardware designers may use specialized busses to connect a few (two or three) isolated components. This works particularly well if those elements are close to each other—perhaps parts of the same CPU chip. For example, busses within the Intel i7 chip connect the full-capability registers with several levels of high-speed cache. This enables data to move between the cache and registers without interfering with data transfer between other components over a primary bus, such as might be needed between an I/O device and the main memory.

Alternative mechanisms for CPU processing

Traditionally, a computer utilized a single CPU to coordinate all processing activity within a machine. Within this framework, processing speed depended directly on the speed and functionality of the central CPU. In modern computers, this CPU bottleneck is resolved, at least partially, in two basic ways:

1. Computers may contain multiple CPUs to share processing tasks.
2. Specialized tasks may be delegated to separate components, so a CPU need not devote much attention to some types of work.

Let us examine each of these two approaches in turn.

Sharing processing among multiple CPUs

Conceptually, we might expect that two CPUs could handle about twice the processing load that might be possible with just one CPU, and four CPUs could handle substantially more processing. For example, suppose we want to use a workstation to search the Web with a browser, send and receive e-mail, balance our checkbook, and write a paper or article. In this context, each task seems completely separate from the others, and we might want to assign a separate processor to each job. With four processors working independently, we might have the impression that we are working with four computers.

In practice, however, dividing work from several tasks among processors can create many challenges—just what work should be done by which processor when. To the extent that we can solve this problem of coordination, however, we could expect that processing could proceed much faster if we had several processors rather than just one.

To allow such increases in performance, some modern processing chips contain multiple CPUs within a single chip (sometimes called **multi-core chips**). Other computers contain several CPUs on separate chips, organized in various configurations. The details of these processing environments vary substantially, and each requires great attention to the coordination of processing among the CPUs.

Moving work outside the CPU

Another approach to increase performance is to move some work outside the CPU. In this environment, a CPU still serves as the coordinator of all activities in a computer, but common tasks may be delegated to specialized components. The idea is that when the CPU decides that a task is required, a separate component could be instructed to perform that work rather than requiring the CPU to do it. For coordination, a component usually signals

Copyrighted material

when the task is done by sending a signal, called an **interrupt**, back to the CPU. This entire approach is similar to the way a site manager may assign skilled workers at a construction site. At the start of a day, one electrician may be asked to install wiring in a room, a plumber might run a pipe to a drain, and a wall finisher might install sheetrock in a room. At various times in the day, each worker may report that the required task is done, and the manager then assigns the next job for that specialty.

As an example involving a computer, consider printing: the CPU may tell a specialized processing CPU chip located in an I/O device that data in a specified area of the main memory should be printed on a designated printer. This separate I/O CPU chip then can handle the details of moving the data to the printer, and the printer can follow the steps required to shape the relevant characters or graphical elements on the paper. When the printing job is complete, the separate I/O CPU chip generates an interrupt, indicating the work is done. Thus, instead of taking time to interact with the printer and move data, the CPU simply issues one command to the I/O CPU chip and then moves on to other tasks. The main CPU decides what documents are printed when, but the mechanics are off-loaded to specialized CPU chips in other components.

The CPU also delegates tasks to specialized CPU chips located in I/O devices when reading from the keyboard, moving data between the main memory and a disk, and interacting with outside computer networks. In this way, the CPU saves its own time for processing other data.

What does the "brain" of a traditional computer look like?

The "brain" of a traditional computer was a single CPU, such as the PowerPC 603 that was first manufactured in 1995 by both Motorola and IBM. An expanded version, called the PowerPC 603e, was used in several Apple Macintosh machines in the mid-1990s. As with many traditional CPUs, the PowerPC 603 is a combination of several smaller components; typically registers, circuitry for various processing tasks (e.g., addition of numbers), and cache are packaged together within a single chip.

The actual PowerPC 603 is built on a wafer of silicon and measures just 7.4 × 11.5 mm. Circuits etched within the CPU involve some 1.6 million transistors. Although such circuits are much too small for an unaided eye to distinguish, Figure 1.5 shows a greatly enlarged picture of the basic PowerPC 603 circuitry.

Although many details of this CPU are well beyond the scope of this book, we can relate many of the main sections of the PowerPC 603 to the basic components already discussed.

First, conceptually, information is divided into three basic types: instructions telling the computer what to do, general data (normally integers without decimal points), and real (or floating-point) numbers with decimal points. As Figure 1.5 suggests, the floating-point (FP) registers are the full-capacity storage locations for real numbers, and the general-purpose (GP) registers hold other data. The CPU contains special circuitry associated with each type of register to perform basic operations, such as addition and multiplication. As the names suggest, a Floating Point Unit contains the basic circuitry for such capabilities as arithmetic operations and the comparison of floating-point numbers, whereas the Integer Execution Unit performs the analogous functions for integers and other data.

Copyrighted material

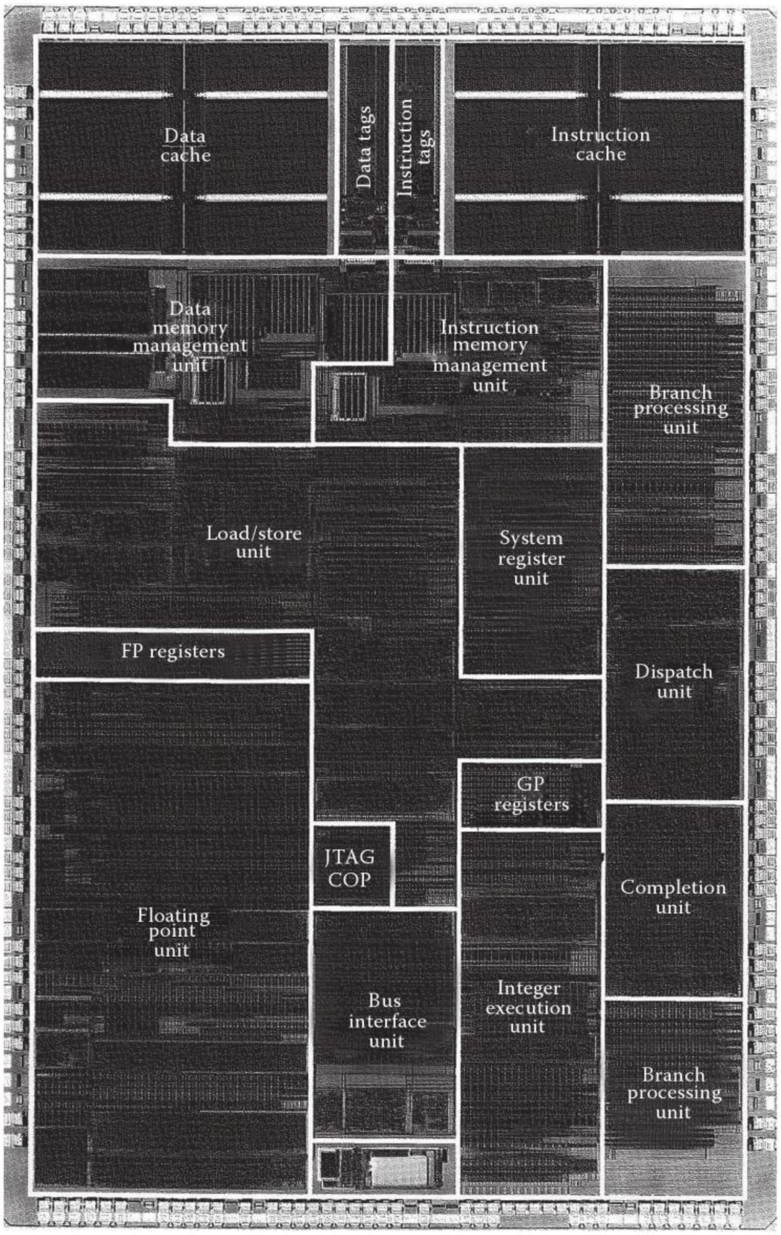

FIGURE 1.5 The PowerPC 603 microprocessor. (From Burgess et al., *Communications of the ACM*, 37(5), 35, 1994, Figure 1B. Copyright 2002 ACM, Inc. Reprinted with permission.)

Other, more specialized processing operations are handled by circuits on the right side of the PowerPC 603, including the Branch Processing Units, the Dispatch and Completion Units, and the System Register Unit. (Unfortunately, details of these specialized operations and units are beyond the scope of this text.)

The PowerPC CPU contains a small, but high-speed, cache for both data and instructions. Movement of data is initiated by the Load/Store Unit that determines what

Copyrighted material

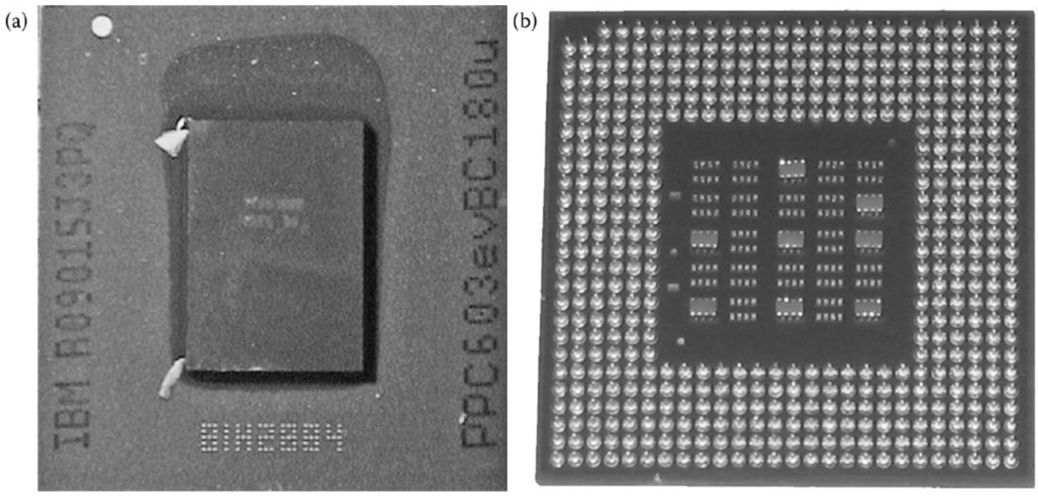

FIGURE 1.6 A ceramic casing for (a) the PowerPC 603 (photographed for this book by Ed Dudak, physics technical assistant at Grinnell College) and (b) the Pentium chip.

information will be needed next. The Data and Instruction Memory Management Units determine whether the desired information is already in cache. (The various tag fields help in this determination.) If the needed information is already located in cache, it can be obtained quickly and directly from the appropriate location on the PowerPC 603. If not, the Bus Interface Unit can be used to interact with the bus and the main memory.

The outside edge of the PowerPC 603 contains about 240 small rectangular pieces or pins. These serve as locations to make electrical connections with the chip. However, because the entire chip is well under a half-inch square, making such connections directly would be a significant practical challenge. As a result, most chips are housed within a relatively large **chip casing** that connects the chip itself to wires or pins that are of a manageable size. In the case of the PowerPC 603, a typical casing is shown in Figure 1.6a. For comparison, a typical casting for the IBM Pentium chip, show in Figure 1.6b, utilizes 296 pins for its connections.

For each example, the casing secures connecting wires to the actual chip in a fixed and workable configuration. The result still is not huge (measuring only three quarters of an inch or so on a side), but the wire connectors are much easier to handle than the extremely small pad areas on the chip itself.

Are modern computer "brains" different from traditional ones?

[Based on information and figures at http://techreport.com/articles.x/15818 by Scott Wasson, dated November 3, 2008 and accessed June 15, 2010.]

A new generation of "brains" or processing chips appears every few years, and some changes are reasonably predictable:

- New chips pack more circuitry into a given amount of space—individual circuits are smaller with each new generation.

Copyrighted material

- New chips are often larger in physical size, allowing even more circuitry to be packed into a single chip.
- New chips contain more memory and/or specialized circuitry to handle various tasks.
- New chips have more pins or electrical connections for communication with the rest of the computer.

In addition, recent processing chips often contain several separate CPUs.

- Each CPU on a processing chip is called a **core**, and multicore chips allow several parts of processing to proceed concurrently.
- A **dual-core** processing chip contains two CPUs.
- A **quad-core** processing chip contains four CPUs.

The Intel i7 processing chip, manufactured initially in 2009, illustrates these recent trends. A comparison of the PowerPC 603 and Intel i7, shown in Table 1.1, gives some indication of how technology has progressed between 1995 and 2009.

In reviewing Table 1.1, this chapter already has discussed transistors, the number of CPUs on a chip, cache, and connections to one or more busses. The following comments provide some background for additional parts of this table.

1. *Size of chip*: The overall dimensions of a chip determine directly how many circuits can be packed into an integrated circuit chip. Naturally, the larger the chip, the more room there is for any desired circuitry. However, over the years, the underlying technology of chip production has limited this size. Manufacturers must be able to reliably fabricate chips that meet required specifications, and factors of production

TABLE 1.1 A Comparison of the PowerPC 603 (1995) and the Intel i7 (2009)

	PowerPC 603	**Intel i7**
Year introduced	1995	2009
Transistors	1.6 million	731 million
Size of chip	85 mm^2	263 mm^2
Central Processing Units (CPUs)	1	4
Cache for instructions (per CPU)	8 thousand storage locations	32 thousand storage locations
Cache for data (per CPU)	8 thousand locations	256 thousand storage locations
Shared cache	—	8 million storage locations
Pins to connect to socket	240	1366
Bus connection(s)	1 bus with 64 parallel wires	192 parallel wires, divided into three 64-wire channels
Width of underlying circuitry	5 micrometers (millionths of a meter)	32 nanometers (billionths of a meter)
Clock speed	80 MHz (80 million cycles per second)	2.66 GHz (2.66 billion cycles per second)
Maximum electrical power	3 watts	18–35 watts

Copyrighted material

significantly influence the size of chips that can be manufactured at a reasonable cost. The comparison of the PowerPC 603 and i7 chips highlights advances in chip production that took place between 1995 and 2009.

2. *Width of underlying circuitry*: The size of a circuit is another important factor in packing circuits onto a chip; the smaller the wires and other electrical components, the more circuits that can be packed into a given space. Overall, the table shows that not only did technology advance to reliably produce larger chips between 1995 and 2009, but advances in technology also allowed the size of wires and circuits in a chip to shrink substantially over the same time. The result is the vast increase in the number of transistors (and other elements) that are packed onto an i7 chip, in contrast to the Power PC 603.

3. *Clock speed*: Yet another type of advance in technology involves the speed at which processing can be done. Within a computer, a clock transmits signals on a regular basis, and these signals serve to coordinate processing. CPUs vary as to just what happens from one clock signal to the next, so direct comparisons of processing speeds are difficulty. However, the speed of the clock can provide some general sense of how quickly processing can proceed. In comparing the PowerPC 603 and the i7 chip, not only are more circuits being utilized, but also the speed of processing within those circuits has increased substantially from 1995 to 2009.

4. *Maximum electrical power*: Processing in computers works by switching electricity among circuits. As the number of circuits and the speed of processing increases, switching also increases for processing, and this additional activity uses more electrical power. The evolution of technology also allows individual circuits to draw less electrical current than in previous years, and this trend toward energy efficiency helps reduce power consumption. Putting these pieces together, Table 1.1 illustrates that the number of circuits (transistors) increased about 450 times from the PowerPC 603 to the i7 chip, the speed of processing (the clock speed) has increased by about 30 times, but advances in energy efficiency limited the increase in power consumption to a factor of about 6–10.

The overall layout of the i7 processing chip is shown in Figure 1.7. Although the cache for each separate core is not shown in this figure, the large shared cache is seen to occupy a substantial portion of the processor chip. With this configuration, data can flow from one core to the shared cache and then to another processor without going outside the chip. Thus, the shared cache serves as a mechanism for the separate CPUs to communicate and transfer data without the need to access the main memory.

As the i7 processing chip has substantial processing capability, the hardware designers paid much attention to how to move data into and out of this chip. Figure 1.7 shows that one part of the i7 chip (at the top of the figure) contains a memory controller. This component oversees connections to the main bus that contains 192 parallel wires (arranged into three independent busses or channels of 64 wires each). Other parts of the i7 processing chip interact with I/O circuits connected to other components within the computer, and these are controlled by other components of the chip—shown in Figure 1.7 on the left and right sides. Overall the Intel i7 chip allows a remarkable number of connections, in that the

Copyrighted material

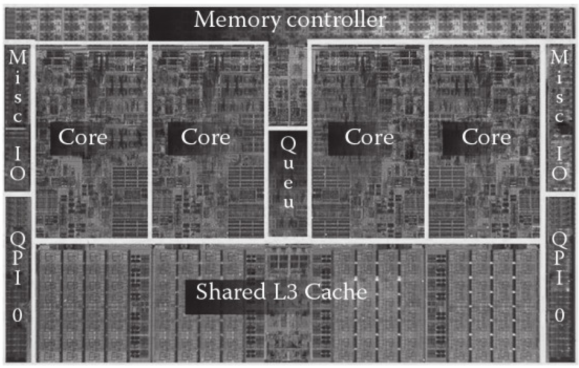

FIGURE 1.7 The Intel i7 processing chip. (Reprinted from Intel. With permission.)

i7 chip plugs into a socket (technically, the FCLGA 1366 socket) that establishes 1366 different electrical connections. Thus, electrical power and data can flow via many paths to keep data moving where needed.

Altogether, modern computer "brains" or processing chips pack many more circuits than in earlier chips, and modern processors may contain multiple CPUs or cores. With this great expansion in capability, modern processing chips also may require considerably more electrical power than earlier chips. As a result, modern computers may contain dedicated fans mounted upon the processing chips to help cooling.

How does this description of components relate to what I can actually see within a computer?

As with CPU chips, many components within real computers may be combined in several ways. First, a single chip may contain several logically separate pieces, as illustrated with both the PowerPC 603 and Intel i7 chips. These chips are then placed in chip casings to facilitate making connections. Sometimes you can look within a computer to see these chip casings directly, with chips embedded inside.

However, modern processing chips often generate significant heat, so the chips and/or chip casings are connected to structures, called **heat sinks**, that act as radiators to help dissipate the heat. One example of a heat sink is shown in Figure 1.8. As this figure suggests, a heat sink is a large piece of metal, and air passes over the fins of the heat sink to create a cooling effect. In this figure, the small rectangle at the bottom shows the area that is actually attached to a processing chip or chip casing. When heat sinks are used, it often is hard to see the main chips at all—heat sinks may obscure the chips themselves.

Figure 1.9 shows an HP Compaq 6000 Pro computer, with the outer cover removed. Some main components visible in this picture are

1. Upper left: a DVD/CD player (plain silver in color).

2. Upper right: a power supply (with a paper label attached) to supply electrical power of the correct voltages to the various components.

Copyrighted material

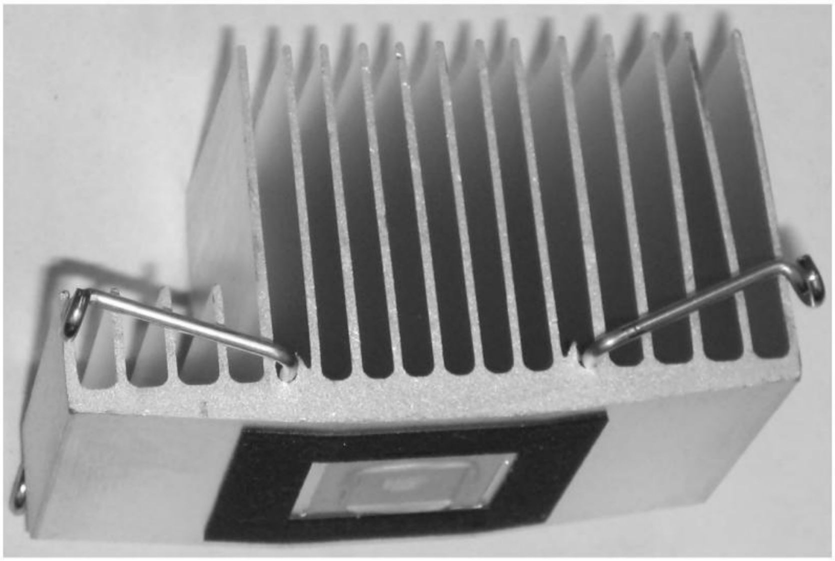

FIGURE 1.8 A heat sink to cool a chip.

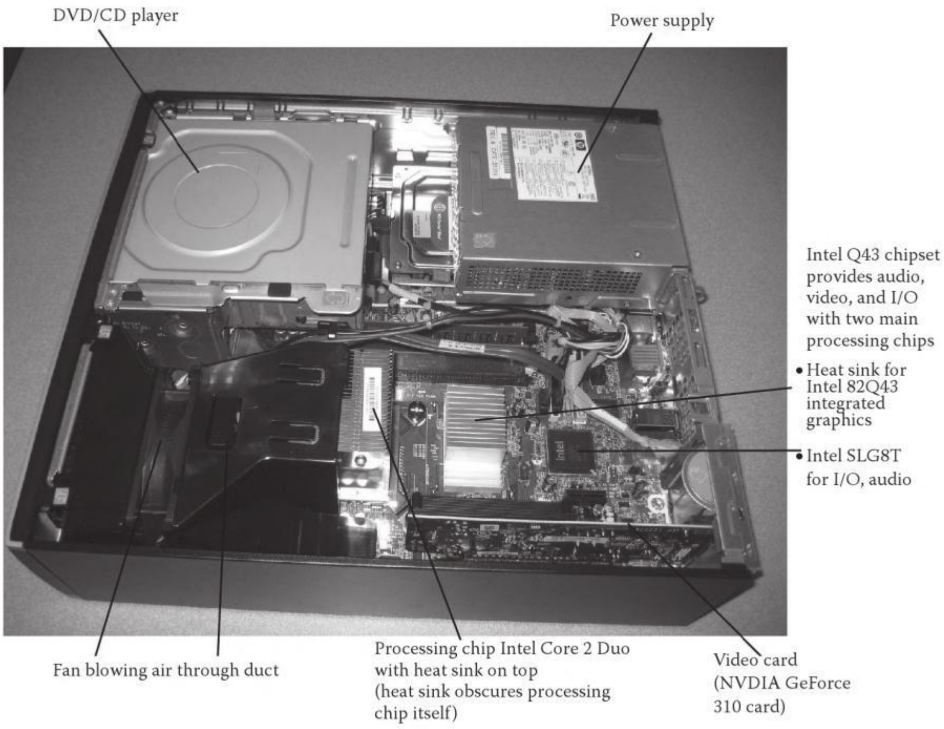
DVD/CD player
Power supply
Intel Q43 chipset provides audio, video, and I/O with two main processing chips
• Heat sink for Intel 82Q43 integrated graphics
• Intel SLG8T for I/O, audio
Fan blowing air through duct
Processing chip Intel Core 2 Duo with heat sink on top (heat sink obscures processing chip itself)
Video card (NVDIA GeForce 310 card)

FIGURE 1.9 Looking inside an HP Compaq 6000 Pro computer.

Copyrighted material

3. Lower left: a fan blowing air through a large, black duct.
4. Left center: the main processing chip (in this case, an Intel Core 2 Duo chip with two core processors), with a heat sink mounted above (the heat sink obscures the processing chip itself).
5. Center and right: to boost performance, several components handle processing tasks related to I/O, audio, and video. Intel bundles these components into a general package (its Q43 chipset), with one chip (an Intel 82Q43 chip) for graphics and another chip (an Intel SLG8T chip) for I/O and audio. A heat sink is attached to the Intel 82Q43 chip to help dissipate heat.
6. Bottom right: a narrow video card (a NVDIA GeForce 310 card) to perform specialized video processing.

Figure 1.10 provides a close-up of the lower part of the cabinet area from Figure 1.9. The top of Figure 1.10 shows the Intel Core 2 Duo chip, topped with a large heat sink. The middle and bottom of Figure 1.10 show the Intel Q43 Chipset with its Intel 82Q43 Integrated Graphics chip (topped with the heat sink from Figure 1.8) and its Intel SLG8T chip for I/O and audio. The NVDIA video card stands vertically on the left of Figure 1.10.

Within Figure 1.10, it is interesting to observe that the fins of the heat sinks at the top and middle are aligned. A fan just above the top of the figure blows air that passes from the top to the bottom through this picture, cooling the fins and the chips that lie beneath.

The NVDIA GeForce 301 video card at the left of Figure 1.10 also illustrates that collections of chips often are combined onto flat plastic cards, with the chips connected by thin, flat wires. Cards are large enough that they can be handled fairly easily. (Note, however, that chip circuitry works with low voltages and very low electric current. Static electricity

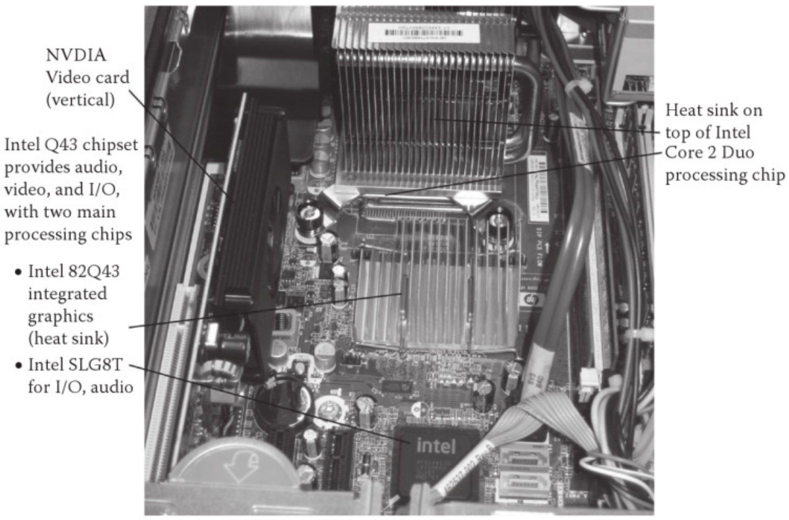
NVDIA Video card (vertical)
Intel Q43 chipset provides audio, video, and I/O, with two main processing chips
• Intel 82Q43 integrated graphics (heat sink)
• Intel SLG8T for I/O, audio
Heat sink on top of Intel Core 2 Duo processing chip
intel

FIGURE 1.10 A close-up view of the processing chips for the HP Compaq 6000 Pro computer.

Copyrighted material

in one's fingers may be sufficient to disrupt chip circuits, and one must be very careful while handling cards.)

To provide access to various internal components, both the DVD drive and the power supply are hinged, allowing the computer to be opened up. Figure 1.11 shows this open configuration and views the internal components from the top direction of Figure 1.9. Here, the power supply (left) has been rotated back, showing an internal disk drive (a Western Digital WE2500AAJS Disk Dive holding 250GB storage); this component (with the blue/white/black sticker) provides much of the long-term storage for this computer. Just above this disk drive, three cards extend horizontally. This machine provides slots to plug in four memory cards, and three of these slots are used for this machine. The top right of this picture also provides another view of the large air duct that carries air from the fan (obscured at the right edge of the case) to the processing chip, the graphics chips, and their heat sinks (top middle and left).

The left of Figure 1.11 shows components mounted on the side of the computer cabinet. These elements provide connections that allow users to plug in various types of audio, video, and other electronic equipment. Figure 1.11 also shows a variety of small wires running from one component to another. Throughout all of these figures, one can see the tops of components, but many elements, including the busses, are located underneath the visible pieces.

Turning from the basic hardware, Figures 1.9 and 1.11 illustrate the overall structural framework of a computer. For example, one section may hold a disk or CD unit. Another

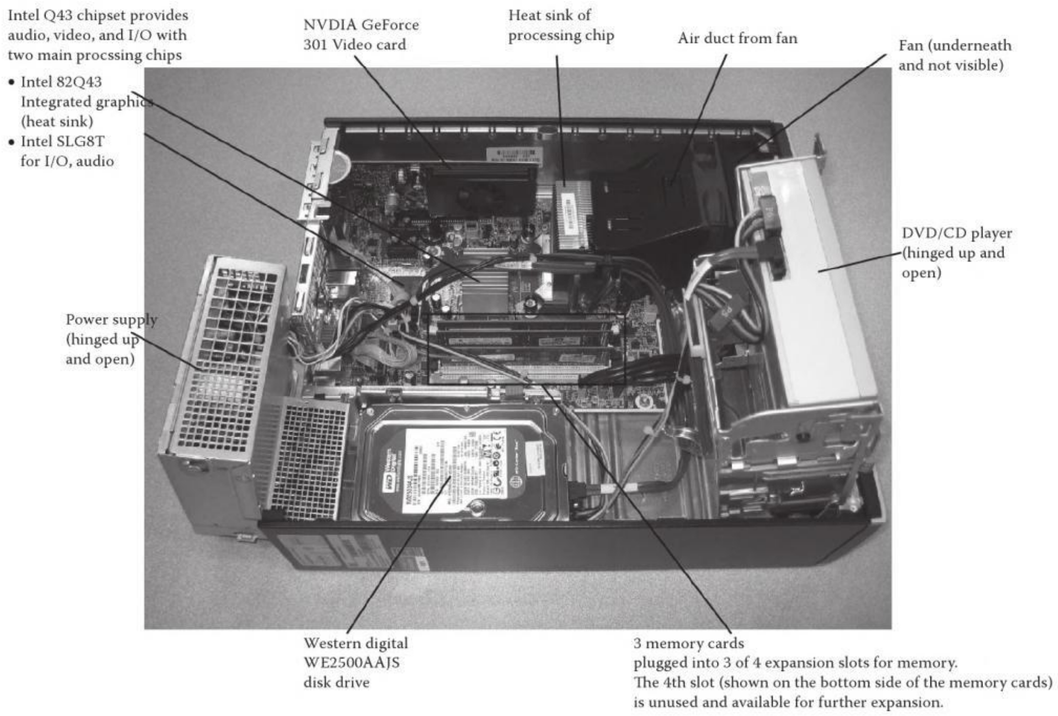
Intel Q43 chipset provides audio, video, and I/O with two main procssing chips
• Intel 82Q43 Integrated graphics (heat sink)
• Intel SLG8T for I/O, audio
NVDIA GeForce 301 Video card
Heat sink of processing chip
Air duct from fan
Fan (underneath and not visible)
DVD/CD player (hinged up and open)
Power supply (hinged up and open)
Western digital WE2500AAJS disk drive
3 memory cards plugged into 3 of 4 expansion slots for memory. The 4th slot (shown on the bottom side of the memory cards) is unused and available for further expansion.

FIGURE 1.11 An HP Compaq 6000 Pro computer with the DVD and disk drive opened for access.

Copyrighted material

section provides several slots, often connected by a bus (underneath) for cards holding the CPU or main memory. In some cases, such as Figure 1.11, more memory can be added just by plugging another card into a free slot. Yet another section holds the power supply and electrical transformer, so that electricity coming from a wall plug can be converted into the form needed by components (typically low-voltage DC rather than high-voltage AC current). Laptop computers also will contain one or more batteries to provide electrical power even if the computer is not plugged in.

Beyond the components themselves, the internal location of elements allows air to circulate. With electricity being used throughout the machine, heat is generated. Dissipation of this heat is vital, because chips and other components tend to malfunction (or even melt) when subjected to excessive heat. The space that separates the parts of the computer helps the computer stay cool by channeling air flow.

Figure 1.12 further highlights the issue of air flow. This figure is largely the same as Figure 1.11, except that the heat sink (from Figure 1.8) has been removed from the Intel 42Q43 Integrated Graphics chip in the upper left section. The casing for the graphics chip is now visible as a small white region located under some connecting wires.

Overall, the internals of real computers provide a clean and logical layout of the various components discussed in this chapter.

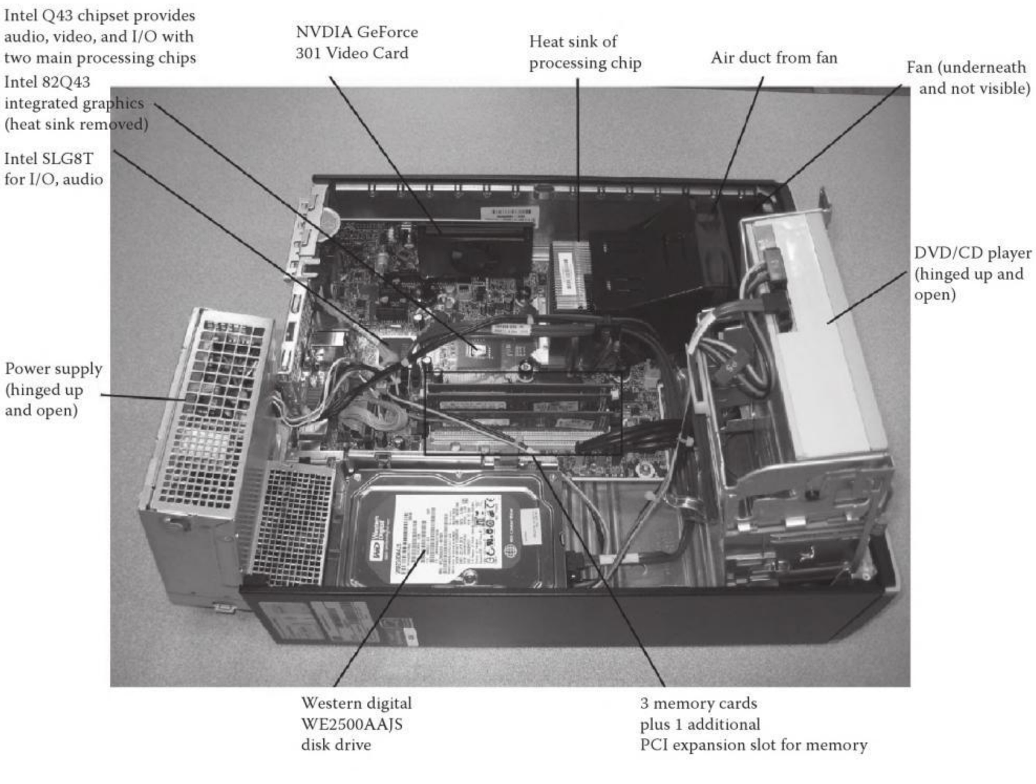
Intel Q43 chipset provides audio, video, and I/O with two main processing chips
• Intel 82Q43 integrated graphics (heat sink removed)
• Intel SLG8T for I/O, audio
NVDIA GeForce 301 Video Card
Heat sink of processing chip
Air duct from fan
Fan (underneath and not visible)
DVD/CD player (hinged up and open)
Power supply (hinged up and open)
Western digital WE2500AAJS disk drive
3 memory cards plus 1 additional PCI expansion slot for memory

FIGURE 1.12 An HP Compaq 6000 Pro computer with the DVD and disk drive opened and a heat sink removed.

Copyrighted material

Summary

Computers contain several primary components: the CPU, main memory, and I/O devices. These are connected by wires, often called busses. Within a computer, many circuits are combined into integrated circuits or IC chips, and chips in turn are placed on cards. You can see the cards and chips when you look inside modern computers.

Latency measures the time required to accomplish a task, such as the movement of data from one place to another. Performance within computers can be improved in several ways, such as

- Several pieces of data may move from place to place in parallel.
- Small, high-speed memory (cache memory) may store frequently used information.
- Processors other than the CPU may handle specialized processing tasks, informing the CPU through interrupts when the tasks are done.

Information within computers can be divided into three general categories: instructions, general data, and real (or floating point) numbers. CPUs are divided into pieces, each of which usually handles a specific type of information and processing.

Terminology from this chapter

Bus	Data	Multicore CPU chip
Cache memory	Expansion slot (or PCI expansion slot)	Peripheral component interconnect
Card	Hardware	Processing
Channel	Heat sink	Program
Chip	Input/Output (I/O) devices	Register
Chip casing	Integrated circuit (IC) or IC chip	Software
Circuit	Latency	Transistor
Computer applications	Main memory	
Control line		
CPU		

Discussion questions

1. In a barber shop or hair salon with just one barber or stylist, customers come in the door, take a seat in a waiting room until called, move to the barber/stylist chair for their haircuts, go to the cash register (also in the waiting room) to pay, and leave through the door.

   a. To what extent does the above flow correspond to processing in a simple computer? (If the barber or stylist plays the role of a CPU, what roles correspond to the door, the customers, and the waiting room?)

   b. Does anything in the barber shop or salon correspond to the cache memory? Why or why not?

Copyrighted material

c. Suppose the salon has a greeter/cashier as well as a stylist. Does this second position correspond with anything within a computer? Explain your answer.

2. Consider the following model related to a student preparing one or more papers. The student finds appropriate books for her work in the library, but does all of her reading and writing in her dorm room. The dorm room has a desk and a small bookcase. The desk has room for only two books, in addition to the word processor where she writes her papers. The bookcase has room for eight books. Due to fire regulations and a general tendency for neatness, the student never uses the floor or other places in her room to store books from the library.

   a. Suppose the student is working on a small paper for which she wants to use two books from the library. Write an outline of the steps she would follow in using the books. For example, she would need to check out the books and use them at her desk.

   b. Suppose the student is working on a paper of moderate size, for which she wants to use seven books from the library. Again, write an outline of what steps she might follow in using the books. Remember that only two books can be open on her desk at one time.

   c. Suppose the student wants to consult 12 books for a large paper. Again, write an outline of what steps she might follow. Even using both her desk and her bookshelf, her dorm room can store only 10 books.

   d. Suppose the student wishes to work on all three papers, for parts a, b, and c at once. After doing some work on one paper, she wants to put it aside and think about it while she works on the other papers. Again, outline how she might proceed in using the books while she makes progress on the three papers.

   e. Compare the work of the student in parts a through d with the workings of a computer. To what extent does the model of student activity have an analog to the CPU, registers, cache memory, or main memory? Are there any ways in which this analogy breaks down? Explain your answer.

3. When a customer calls a service technician regarding a problem with a television set, the technician travels to the customer's home. The technician keeps a collection of commonly used tools in her truck, and uses these as needed in troubleshooting the problem. Often, the technician can fix the problem during the initial call. However, sometimes the technician must take the television set to the shop for more extensive repairs. And sometimes the technician must order parts from a factory or warehouse before the problem can be fixed. To what extent does this hierarchy of equipment and service (house, tools in truck, equipment in shop, parts from a warehouse) correspond to the storage of data within a computer? Explain any similarities and clarify any differences.

4. Check the warranty requirements regarding a computer that is available to you. If not prohibited by the warranty or by school/company policy, remove the outside cover of

Copyrighted material

a computer. Looking inside, try to locate the CPU, main memory, expansion slots, etc. What other components can you identify?

5. Technological considerations typically limit how much circuitry can be jammed into a single chip. Thus, if one part of a chip is enlarged to increase its capacity, then another part often must be decreased with corresponding loss of function. Looking at the PowerPC 603 CPU chip in Figure 1.5, and the Intel i7 chip in Figure 1.7, estimate what fraction of each chip is devoted to each of the main components:

   a. Operation processing

   b. Identifying and locating information/coordinating memory

   c. Cache

   d. Interactions with the bus

   Then determine the relative sizes of the different sections of this CPU chip.

6. A meal is to be prepared for 10 people. The meal consists of three courses (an appetizer, main course, and dessert), and a separate beverage is served with each course (fruit punch with appetizer, water or wine with the main course, coffee or tea with dessert). Preparing each course involves cutting ingredients, blending, cooking in various pans, placing food on dishes, and serving—all according to separate recipes.

   a. Identify a specific menu for the meal, consider specific recipes, and outline the time needed for each step. Assume there is one stove that has four burners and two ovens.

   b. Suppose each meal is to be prepared by a single chef (i.e., a single processor). Estimate how long it might take to prepare and serve the meal.

   c. Suppose there are four chefs (i.e., suppose four cores comprise a processor). Which tasks in the meal preparation might be done in parallel, and which would have to be done in a specific sequence? What coordination might have to be done to ensure the meal preparation follows the appropriate steps? Estimate how long it might take to prepare and serve the meal with four chefs, including time for coordination.

   d. Compare times for meal preparation and service for the one and four chefs. How much faster might the process proceed with four chefs? If four chefs can finish a full four times faster, what properties of this problem allow all chefs to work at full capacity all of the time. If four chefs cannot work four times faster, what elements of the problem slow them down?

   e. This question has mentioned that multiple chefs might be analogous to multiple cores in a processor. To what extent does this analogy hold up under analysis?

   f. Now, suppose counter space was limited, so only three cutting boards or bowls can be placed on the counter at one time. How might this constraint affect the preparation time for one chef and for four chefs?

Copyrighted material

g. Extending the analogy between meal preparation and computer processing, what limitations within a computer might be analogous to restricted counter space within a kitchen?

h. How might your conclusions differ if you changed the menu in step a?

7. Suppose the outside of a house is to be painted, including walls, windows, doors, and trim; and suppose the project should apply two coats of paint to all exterior surfaces.

   a. Identify some constraints regarding what elements should be painted before other elements. (E.g., when painting trim on upper-story windows, the paint might spatter on the walls below, so the final coat of paint for the lower walls should be applied after the upper-story walls and trim.)

   b. One way to utilize four painters would be to have each painter focus on one side of the house (north, east, south, and west). What constraints might this organization impose on the work by the four painters?

   c. Another way to utilize the painters would be for one to focus on windows, one on trim, and two on walls. What constraints might this organization impose on the work by the four painters?

   d. Consider the overall time to paint a house by one painter and the time for four painters following approaches b and c. Which approach would take the longest? Which would finish quickest? In each case, justify your conclusions.

   e. To what extent might painting a house be analogous to processing within a computer with one or four CPUs? Does the example of house painting provide any lessons for computer processing? If so, what are they? If not, why are the two environments so different?

8. Ask the Information Technology Services (ITS) department within your school or business to open up a desktop computer and a laptop computer, so you can see various components within the case.

   a. Can you find the CPU (perhaps with help from ITS)?

   b. What components are most prominent?

   c. Can you find the main memory, busses, I/O devices?

   d. How are memory drives (e.g., CDs or DVDs) connected to the rest of the computer?

   e. Can you find any fans inside the computers? If so, where are they?

   f. Is the inside of a desktop computer packed with components, or is there much air space? Why do you think this is the case?

9. Consider a person who lives in the suburbs but works in a city, so the person must commute between home and work.

Copyrighted material

a. To what extent is commute time related to the concept of latency in computers?

b. Identify several ways that a commuter might reduce commute time.

c. Do any of the efficiencies identified in part b have an analog in reducing latency within a computer?

10. Laptop computers usually have less processing capability and less storage space than larger desktop computers. Considering topics discussed in this chapter, list at least six different reasons why this might be the case. Can you find still more possible reasons why laptops have less computing capacity than desktop models?

Exercises

1. Using your own words, give a careful definition for each of the words or phrases identified in the "Terminology from this chapter."

2. This chapter introduces basic concepts of hardware and software.

   a. Give a careful definition of these two terms. Be sure that your definitions clarify how these two ideas differ.

   b. Consider "hardware," "software," and "other" as three main categories for topics discussed in this chapter. Then place each of the other words or phrases listed in the "Terminology from this chapter" under its appropriate category. Briefly justify your classification.

3. In cities, busses transport people, with specific stops along a fixed route. Consider the extent to which busses within a computer fill a similar role. How are the two types of busses similar? Are there any differences? Explain.

4. Make a field trip to a computer store to inquire about the availability of expansion slots within computers. What can you say about the availability of such slots? Do such slots come at an additional cost? Explain your conclusions.

5. The text indicates that the actual PowerPC 603 chip is just 7.4×11.5 mm^2, although the picture of this chip is magnified significantly in Figure 1.5. Use a ruler to determine the approximate size of a connection area on Figure 1.5, and then scale the dimensions down to the actual chip size to estimate the actual size for an actual connection to the chip.

6. A chip contains many electrical circuits and components. One such component is called a transistor—a type of switch to turn a circuit on and off. The text notes that the PowerPC 603 chip contains about 1.6 million transistors with an area of 85 mm^2. The text also indicates that the Intel i7 chip contains 731 million transistors in a chip with an area of 263 mm^2. Estimate the size of one transistor, ignoring other electrical components on each chip.

Copyrighted material

7. Intel advertises that a transistor for its i7-620M processor is 32 nanometers wide (a nanometer is one billionth of a meter).

   a. How wide is a silicon atom?

   b. Give an estimate for the width of an i7 circuit in atoms.

8. Modern computers use one or more busses, and each component within the computer connects to the bus. Adding a new component therefore requires just one new connection—to the bus.

   a. Suppose a computer was to be developed with a main memory, CPU, disk drive, CD drive, keyboard, and monitor. Suppose further that, instead of a bus, each component needed to be connected separately to the other components (instead of the one bus). How many wires would be needed to connect each of these components to each of the others?

   b. Suppose a new component (e.g., a digital camera) were to be added to this alternative computer, making it necessary to add separate wires from the new component to each of the existing components. How many new wires would be needed to connect this new component?

   c. Suppose yet another component (e.g., a printer) were to be added in addition to the component from part b. How many new wires would be needed to connect the new component?

   d. Using your answers to parts a, b, and c as a base, explain why you think busses have become the standard way to connect components within computers.

9. The chapter outlines a hierarchy of locations to store data, including registers, cache (often several levels of cache), main memory, and disk.

   a. Sometimes a computer can be made to process faster by adding more cache memory or more main memory. Describe some situations in which adding more memory might speed processing—what types of applications might benefit from this additional memory?

   b. For some other situations, adding more cache or main memory might have little impact on processing speed. Rather processing is limited by other factors. What types of applications might *not* benefit from additional memory?

10. A classical type of algebra problem goes something like this: If Hannah can paint a house in four days and Jackson can paint the house in six days, how long will it take them to paint the house working together?

    This exercise asks similar questions in the context of computers and printers. Supposed a printer, connected to a computer, can print a page in 15 seconds.

Copyrighted material

a. How long will it take the printer/computer to print 20 pages?

b. How long would it take 20 printers, each with its own computer, to print 20 pages?

c. How long would it take 40 printers, each with its own computer, to print 20 pages? (Careful!)

d. Explain what assumptions might be made for the house-painting problem and how the assumptions for printing a page on a printer might be different.

Research exercises

1. Traditional processor chips have a single core, whereas several modern processor chips have two, four, or more cores. How much faster will a desktop computer perform with multicore chips over the traditional one-core chip?

2. A computer may or may not have a separate graphics processor as part of its internal hardware. Will a graphics processor always increase computer performance? If so, by how much? If not, why not?

3. The Intel i7 processing chip has an area of about 263 mm^2.

   a. What are the length and width for this chip?

   b. How large is the socket that the i7 plugs into?

   c. The i7 chip has 1366 pins or connections to the socket that holds it. What are all these pins or connections used for?

4. The transistor is the basic circuit for much computer technology.

   a. How has the size of transistors decreased over the years? (Draw a rough graph.)

   b. What is the smallest size reported for a working transistor?

5. This problem asks you to compare processing power in modern appliances with the power of past computers.

   a. How large are the CPU and memory in a modern microwave oven?

   b. How do the sizes in part a compare with a typical minicomputer (e.g., an IBM 1130) of the 1960s? (Note that the IBM 1130 was about the size of a desk.)

   c. How do the sizes in part a compare with a typical personal computer of the 1970s or 1980s?

6. Repeat research exercise 5, making the comparison with modern cell phones.

Copyrighted material

CHAPTER 2

How Are Numbers and Characters Represented in a Computer (and Who Cares)?

When we use computer applications, we interact with the computer at the user level, normally focusing surprisingly little attention on the technology that allows us to accomplish our work at hand. For example, when browsing the Internet, we likely think about the topic we are researching and the information on the screen rather than the servers and networks that make our communication possible. When writing using word processors, we concentrate on the content we want to convey and consider the application's role only in such matters as type font, type size, and page layout. As users, we work with what we "see"; rarely do we stop and think about how the computer processes and organizes our data so that we can see them. Because the machine handles so many behind-the-scenes tasks, we can easily ignore the technical details of how data are represented and stored. We let the machine make the technical decisions, and our work can progress smoothly. The computer's technical decisions about data representation, however, can affect our work in several ways. It is to our advantage to understand how the storage of information impacts:

- The accuracy of our results
- The speed of processing
- The range of alphabets available to us
- The size of the files we must store
- The appearance of the graphics we see on the screen or printed on a page
- The time it takes for materials to download on the Internet

Copyrighted material