What is a zyBook?

New to zyBooks? Check out a short video to learn how zyBooks uses concise writing, interactive activities, and research-backed approaches to help students learn.

Watch now

1.1 Introduction

> "Civilization advances by extending the number of important operations which we can perform without thinking about them."
> *Alfred North Whitehead, An Introduction to Mathematics, 1911*

Welcome to this book! We're delighted to have this opportunity to convey the excitement of the world of computer systems. This is not a dry and dreary field, where progress is glacial and where new ideas atrophy from neglect. No! Computers are the product of the incredibly vibrant information technology industry, all aspects of which are responsible for almost 10% of the gross national product of the United States, and whose economy has become dependent in part on the rapid improvements in information technology promised by Moore's Law. This unusual industry embraces innovation at a breath-taking rate. In the last 30 years, there have been a number of new computers whose introduction appeared to revolutionize the computing industry; these revolutions were cut short only because someone else built an even better computer.

This race to innovate has led to unprecedented progress since the inception of electronic computing in the late 1940s. Had the transportation industry kept pace with the computer industry, for example, today we could travel from New York to London in a second for a penny. Take just a moment to contemplate how such an improvement would change society—living in Tahiti while working in San Francisco, going to Moscow for an evening at the Bolshoi Ballet—and you can appreciate the implications of such a change.

Computers have led to a third revolution for civilization, with the ***information revolution*** taking its place alongside the agricultural and the industrial revolutions. The resulting multiplication of humankind's intellectual strength and reach naturally has affected our everyday lives profoundly and changed the ways in which the search for new knowledge is carried out. There is now a new vein of scientific investigation, with computational scientists joining theoretical and experimental scientists in the exploration of new frontiers in astronomy, biology, chemistry, and physics, among others.

The computer revolution continues. Each time the cost of computing improves by another factor of 10, the opportunities for computers multiply. Applications that were economically infeasible suddenly become practical. In the recent past, the following applications were "computer science fiction."

- *Computers in automobiles:* Until microprocessors improved dramatically in price and performance in the early 1980s, computer control of cars was ludicrous. Today, computers reduce pollution, improve fuel efficiency via engine controls, and increase safety through blind spot warnings, lane departure warnings, moving object detection, and air bag inflation to protect occupants in a crash.
- *Cell phones:* Who would have dreamed that advances in computer systems would lead to more than half of the planet having mobile phones, allowing person-to-person communication to almost anyone anywhere in the world?
- *Human genome project:* The cost of computer equipment to map and analyze human DNA sequences was hundreds of millions of dollars. It's unlikely that anyone would have considered this project had the computer costs been 10 to 100 times higher, as they would have been 15 to 25 years earlier. Moreover, costs continue to drop; you will soon be able to acquire your own genome, allowing medical care to be tailored to you.
- *World Wide Web:* Not in existence at the time of the first edition of this book, the web has transformed our society. For many, the web has replaced libraries and newspapers.
- *Search engines:* As the content of the web grew in size and in value, finding relevant information became increasingly important. Today, many people rely on search engines for such a large part of their lives that it would be a hardship to go without them.

Clearly, advances in this technology now affect almost every aspect of our society. Hardware advances have allowed programmers to create wonderfully useful software, which explains why computers are omnipresent. Today's science fiction suggests tomorrow's killer applications: already on their way are glasses that augment reality, the cashless society, and cars that can drive themselves.

PARTICIPATION ACTIVITY 1.1.1: The information revolution.

1) The computing industry has not improved quite as rapidly as the transportation industry.
   - ○ True
   - ○ False

2) The agricultural and industrial revolutions each transformed society. Computers have led to a relatively recent information revolution.
   - ○ True
   - ○ False

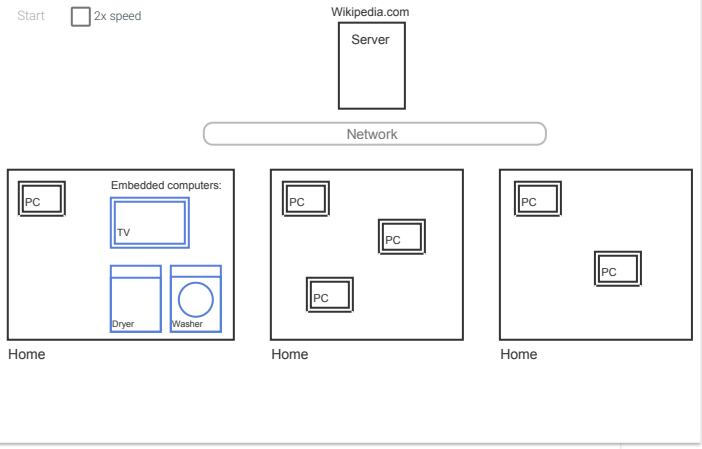
3) Computer improvements have led to previously undreamt applications like cell phones, but most signs suggest the improvements are now coming to an end.
   - ○ True
   - ○ False

Traditional classes of computing applications and their characteristics

Although a common set of hardware technologies (see COD Sections 1.4 (Under the covers) and 1.5 (Technologies for building processors and memory)) is used in computers ranging from smart home appliances to cell phones to the largest supercomputers, these different applications have different design requirements and employ the core hardware technologies in dissimilar ways. Broadly speaking, computers are used in three different classes of applications: personal computers, servers, and embedded computers.

PARTICIPATION 1.1.2: Classes of computing: Personal computers, servers, and embedded

ACTIVITY computers.



Personal computers (PCs) are possibly the best-known form of computing, which readers of this book have likely used extensively. Personal computers emphasize delivery of good performance to single users at low cost and usually execute third-party software. This class of computing drove the evolution of many computing technologies, which is only about 35 years old!

Personal computer (***PC***): A computer designed for use by an individual, usually incorporating a graphics display, a keyboard, and a mouse.

Figure 1.1.1: Personal computers: Desktop and laptop computers.

Source: zyBooks

Servers are the modern form of what were once much larger computers, and are usually accessed only via a network. Servers are oriented to carrying sizable workloads, which may consist of either single complex applications—usually a scientific or engineering application—or handling many small jobs, such as would occur in building a large web server. These applications are usually based on software from another source (such as a database or simulation system), but are often modified or customized for a particular function. Servers are built from the same basic technology as desktop computers, but provide for greater computing, storage, and input/output capacity. In general, servers also place a greater emphasis on dependability, since a crash is usually more costly than it would be on a single-user PC.

Server: A computer used for running larger programs for multiple users, often simultaneously, and typically accessed only via a network.

Servers span the widest range in cost and capability. At the low end, a server may be little more than a desktop computer without a screen or keyboard and cost a thousand dollars. These low-end servers are typically used for file storage, small business applications, or simple web serving (see COD Section 6.10 (Multiprocessor benchmarks and performance models)). At the other extreme are *supercomputers*, which at the present consist of tens of thousands of processors and many *terabytes* of memory, and cost tens to hundreds of millions of dollars (see terabytes discussion near this section's end).

Figure 1.1.2: Servers: Each rack contains multiple servers.

Source: Florian Hirzinger / GFDL or CC-BY-SA-3.0 via Wikimedia Commons

Supercomputers are usually used for high-end scientific and engineering calculations, such as weather forecasting, oil exploration, protein structure determination, and other large-scale problems. Although such supercomputers represent the peak of computing capability, they represent a relatively small fraction of the servers and thus a proportionally tiny fraction of the overall computer market in terms of total revenue.

Supercomputer: A class of computers with the highest performance and cost; they are configured as servers and typically cost tens to hundreds of millions of dollars.

Figure 1.1.3: Supercomputers: Cray Y 190A supercomputer.

Source: NASA / Public domain via Wikimedia Commons

Embedded computers are the largest class of computers and span the widest range of applications and performance. Embedded computers include the microprocessors found in your car, the computers in a television set, and the networks of processors that control a modern airplane or cargo ship. Embedded computing systems are designed to run one application or one set of related applications that are normally integrated with the hardware and delivered as a single system; thus, despite the large number of embedded computers, most users never really see that they are using a computer!

Embedded computer: A computer inside another device used for running one predetermined application or collection of software.

Embedded applications often have unique application requirements that combine a minimum performance with stringent limitations on cost or power. For example, consider a music player: the processor need only be as fast as necessary to handle its limited function, and beyond that, minimizing cost and power is the most important objective. Despite their low cost, embedded computers often have lower tolerance for failure, since the results can vary from upsetting (when your new television crashes) to devastating (such as might occur when the computer in a plane or cargo ship crashes). In consumer-oriented embedded applications, such as a digital home appliance, dependability is achieved primarily through simplicity—the emphasis is on doing one function as perfectly as possible. In large embedded systems, techniques of redundancy from the server world are often employed. Although this book focuses on general-purpose computers, most concepts apply directly, or with slight modifications, to embedded computers.

Figure 1.1.4: Embedded computer: Thermostat.

Source: zyBooks

PARTICIPATION ACTIVITY 1.1.3: The three classes of computing applications.

Indicate to which class each computing application belongs.

1) A home computer kept on a desktop and used by family members for emails, web browsing, social networking, and movie watching.
   - ○ Embedded
   - ○ PC
   - ○ Server

2) A computer in an Amazon building accessed by thousands of people for online shopping.
   - ○ Embedded
   - ○ PC
   - ○ Server

3) A computer in a cardiac pacemaker, which delivers electric shocks to keep a human's heart beating properly.
   - ○ Embedded
   - ○ PC
   - ○ Server

4) A computer at a federal laboratory that

continually executes sophisticated algorithms on massive amounts of data from various weather stations to develop accurate weather forecasts.

- ○ Embedded
- ○ PC
- ○ Server

Elaboration

Elaborations are short features used throughout the text to provide more detail on a particular subject that may be of interest. Disinterested readers may skip over an elaboration, since the subsequent material will never depend on the contents of the elaboration.

*Many embedded processors are designed using **processor cores**, a version of a processor written in a hardware description language, such as Verilog or VHDL (see COD Chapter 4 (The Processor)). The core allows a designer to integrate other application-specific hardware with the processor core for fabrication on a single chip.*

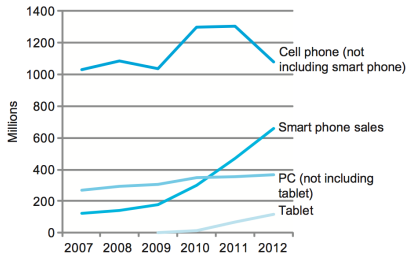
Welcome to the Post-PC era

The continuing march of technology brings about generational changes in computer hardware that shake up the entire information technology industry. Since the last edition of the book, we have undergone such a change, as significant in the past as the switch starting 30 years ago to personal computers. Replacing the PC is the *personal mobile device (PMD)*. PMDs are battery operated with wireless connectivity to the Internet and typically cost hundreds of dollars, and, like PCs, users can download software ("apps") to run on them. Unlike PCs, they no longer have a keyboard and mouse, and are more likely to rely on a touch-sensitive screen or even speech input. Today's PMD is a smart phone or a tablet computer, but tomorrow it may include electronic glasses. The following figure shows the rapid growth over time of tablets and smart phones versus that of PCs and traditional cell phones.

Personal mobile devices (***PMDs***) are small wireless devices to connect to the Internet; they rely on batteries for power, and software is installed by downloading apps. Conventional examples are smart phones and tablets.

Figure 1.1.5: The number manufactured per year of tablets and smart phones, which reflect the post-PC era, versus personal computers and traditional cell phones (COD Figure 1.2).

Smart phones represent the recent growth in the cell phone industry, and they passed PCs in 2011. Tablets are the fastest growing category, nearly doubling between 2011 and 2012. Recent PCs and traditional cell phone categories are relatively flat or declining.



Taking over from the conventional server is *Cloud Computing*, which relies upon giant datacenters that are now known as ***Warehouse Scale Computers*** (WSCs). Companies like Amazon and Google build these WSCs containing 100,000 servers and then let companies rent portions of them so that they can provide software services to PMDs without having to build WSCs of their own. Indeed, Software as a Service (SaaS) deployed via the Cloud is revolutionizing the software industry just as PMDs and WSCs are revolutionizing the hardware industry. Today's software developers will often have a portion of their application that runs on the PMD and a portion that runs in the Cloud.

Cloud computing refers to large collections of servers that provide services over the Internet; some providers rent dynamically varying numbers of servers as a utility.

Software as a Service (SaaS) delivers software and data as a service over the Internet, usually via a thin program such as a browser that runs on local client devices, instead of binary code that must be installed, and runs wholly on that device. Examples include web search and social networking.

PARTICIPATION ACTIVITY 1.1.4: Post-PC era.

1) "Post-PC" era refers to today's situation of today's most widely-used computers being things other than PCs.
   - ○ True
   - ○ False

2) The number of PCs sold annually continues to increase at a dramatic rate.
   - ○ True
   - ○ False

3) A smart phone isn't actually a computer.
   - ○ True
   - ○ False

4) Cloud computing often involves thousands of servers in giant warehouses.
   - ○ True
   - ○ False

5) Software as a Service refers to installing large software applications on a PC, such as Microsoft Office.
   - ○ True
   - ○ False

6) In 2012, about 600 million smart phones were sold.
   - ○ True
   - ○ False

What you can learn in this book

Successful programmers have always been concerned about the performance of their programs, because getting results to the user quickly is critical in creating popular software. In the 1960s and 1970s, a primary constraint on computer performance was the size of the computer's memory. Thus, programmers often followed a simple credo: minimize memory space to make programs fast. In the last decade, advances in computer design and memory technology have greatly reduced the importance of small memory size in most applications other than those in embedded computing systems.

Programmers interested in performance now need to understand the issues that have replaced the simple memory model of the 1960s: the parallel nature of processors and the hierarchical nature of memories. We demonstrate the importance of this understanding in COD Chapters 3 (Arithmetic for Computers) to 6 (Parallel Processors from Client to Cloud) by showing how to improve performance of a C program by a factor of 200. Moreover, as we explain in COD Section 1.7 (The power wall), today's programmers need to worry about energy efficiency of their programs running either on the PMD or in the Cloud, which also requires understanding what is below your code. Programmers who seek to build competitive versions of software will therefore need to increase their knowledge of computer organization.

We are honored to have the opportunity to explain what's inside this revolutionary machine, unraveling the software below your program and the hardware under the covers of your computer. By the time you complete this book, we believe you will be able to answer the following questions:

- How are programs written in a high-level language, such as C or Java, translated into the language of the hardware, and how does the hardware execute the resulting program? Comprehending these concepts forms the basis of understanding the aspects of both the hardware and software that affect program performance.
- What is the interface between the software and the hardware, and how does software instruct the hardware to perform needed functions? These concepts are vital to understanding how to write many kinds of software.
- What determines the performance of a program, and how can a programmer improve the performance? As we will see, this depends on the original program, the software translation of that program into the computer's language, and the effectiveness of the hardware in executing the program.
- What techniques can be used by hardware designers to improve performance? This book will introduce the basic concepts of modern computer design. The interested reader will find much more material on this topic in our advanced book, *Computer Architecture: A Quantitative Approach.*
- What techniques can be used by hardware designers to improve energy efficiency? What can the programmer do to help or hinder energy efficiency?
- What are the reasons for and the consequences of the recent switch from sequential processing to parallel processing? This book gives the motivation, describes the current hardware mechanisms to support parallelism, and surveys the new generation of *"multicore" microprocessors* (see COD Chapter 6 (Parallel Processors from Client to Cloud)).
- Since the first commercial computer in 1951, what great ideas did computer architects come up with that lay the foundation of modern computing?

Multicore microprocessor: A microprocessor containing multiple processors ("cores") in a single integrated circuit.

Without understanding the answers to these questions, improving the performance of your program on a modern computer or evaluating what features might make one computer better than another for a particular application will be a complex process of trial and error, rather than a scientific procedure driven by insight and analysis.

This first chapter lays the foundation for the rest of the book. It introduces the basic ideas and definitions, places the major components of software and hardware in perspective, shows how to evaluate performance and energy, introduces integrated circuits (the technology that fuels the computer revolution), and explains the shift to multicores.

In this chapter and later ones, you will likely see many new words, or words that you may have heard but are not sure what they mean. Don't panic! Yes, there is a lot of special terminology used in describing modern computers, but the terminology actually helps, since it enables us to describe precisely a function or capability. In addition, computer designers (including your authors) love using *acronyms*, which are easy to understand once you know what the letters stand for! To help you remember and locate terms, we include a definition of each key term in a colored box near the first place a key term appears in the text. After a short time of working with the terminology, you will be fluent, and your friends will be impressed as you correctly use acronyms such as BIOS, CPU, DIMM, DRAM, PCIe, SATA, and many others.

Acronym: A word constructed by taking the initial letters of a string of words. For example: *RAM* is an acronym for Random Access Memory, and *CPU* is an acronym for Central Processing Unit.

PARTICIPATION ACTIVITY 1.1.5: What can be learned from this book.

1) Today's programmers of PCs and servers emphasize improving program performance by using a minimal amount of memory.
   - ○ True

- False

2) Today's programmers also emphasize energy efficiency of programs.
   - True
   - False

3) The performance of a program depends on many things, like the original program, how the program is translated to a computer's language, and the hardware.
   - True
   - False

4) A multicore microprocessor is a single processor capable of switching between multiple programs.
   - True
   - False

5) Acronyms are rarely used in the field of computers.
   - True
   - False

To reinforce how the software and hardware systems used to run a program will affect performance, we use a special example called *Understanding program performance* throughout the book to summarize important insights into program performance. The first one appears below.

Understanding program performance

The performance of a program depends on a combination of the effectiveness of the algorithms used in the program, the software systems used to create and translate the program into machine instructions, and the effectiveness of the computer in executing those instructions, which may include input/output (I/O) operations. This table summarizes how the hardware and software affect performance.

Hardware or software component	How this component affects performance	Where is this topic covered?
Algorithm	Determines both the number of source-level statements and the number of I/O operations executed	Other books!
Programming language, compiler, and architecture	Determines the number of computer instructions for each source-level statement	COD Chapters 2 (Instructions: Language of the Computer) and 3 (Arithmetic for Computers)
Processor and memory system	Determines how fast instructions can be executed	COD Chapters 4 (The Processor), 5 (Large and Fast: Exploiting Memory Hierarchy), and 6 (Parallel Processor from Client to Cloud)
I/O system (hardware and operating system)	Determines how fast I/O operations may be executed	COD Chapters 4 (The Processor), 5 (Large and Fast: Exploiting Memory Hierarchy), and 6 (Parallel Processor from Client to Cloud)

To demonstrate the impact of the ideas in this book, as mentioned above, we improve the performance of a C program that multiplies a matrix times a vector in a sequence of chapters. Each step leverages understanding how the underlying hardware really works in a modern microprocessor to improve performance by a factor of 200!

- In the category of *data level parallelism*, in COD Chapter 3 (Arithmetic for Computers) we use *subword parallelism via C intrinsics* to increase performance by a factor of 3.8.
- In the category of *instruction level parallelism*, in COD Chapter 4 (The Processor) we use *loop unrolling to exploit multiple instruction issue and out-of-order execution hardware* to increase performance by another factor of 2.3.
- In the category of *memory hierarchy optimization*, in COD Chapter 5 (Large and Fast: Exploiting Memory Hierarchy) we use *cache blocking* to increase performance on large matrices by another factor of 2.0 to 2.5.
- In the category of *thread level parallelism*, in COD Chapter 6 (Parallel Processor from Client to Cloud) we use *parallel for loops in OpenMP to exploit multicore hardware* to increase performance by another factor of 4 to 14.

A "terabyte" and other sizes

The above discussion on supercomputers used the term "terabyte".

Terabyte (TB): Originally 1,099,511,627,776 (2^{40}) bytes, although communications and secondary storage systems developers started using the term to mean 1,000,000,000,000 (10^{12}) bytes. To reduce confusion, we now use the term ***tebibyte (TiB)*** for 2^{40} bytes, defining *terabyte* (TB) to mean 10^{12} bytes. The figure below shows the full range of decimal and binary values and names.

Table 1.1.1: Size ambiguity resolved with binary notation for common size terms (COD Figure 1.1).

The 2^X vs. 10^Y bytes ambiguity was resolved by adding a binary notation for all the common size terms. In the last column we note how much larger the binary term is than its corresponding decimal term, which is compounded as we head down the chart. These prefixes work for bits as well as bytes, so *gigabit* (Gb) is 10^9 bits while *gibibits* (Gib) is 2^{30} bits.

Decimal	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

PARTICIPATION ACTIVITY 1.1.6: Terms for common sizes.

1) A terabyte is one _____bytes.

Check Show answer

2) 10^{15} bytes is a _____byte.

Check Show answer

3) A gibibyte is close, but not equal, to a _____byte.

Check Show answer

Check yourself

Check yourself sections are designed to help readers assess whether they comprehend the major concepts introduced in a chapter and understand the implications of those concepts. Some *Check yourself* questions have simple answers; others are for discussion among a group.

1. The number of embedded processors sold every year greatly outnumbers the number of PC and even Post-PC processors. Can you confirm or deny this insight based on your own experience? Try to count the number of embedded processors in your home. How does it compare with the number of conventional computers in your home?
2. As mentioned earlier, both the software and hardware affect the performance of a program. Can you think of examples where each of the following is the right place to look for a performance bottleneck?
   - The algorithm chosen
   - The programming language or compiler
   - The operating system
   - The processor
   - The I/O system and devices

Answer: Discussion questions: many answers are acceptable.

Provide feedback on this section