教育部高等教育司推荐
国外优秀信息科学与技术系列教学用书

Intel 微处理器

——从 8086 到 Pentium 系列 体系结构、编程与接口技术

(第五版 影印版)

THE INTEL MICROPROCESSORS

8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing

(Fifth Edition)

■ Barry B. Brey

高等教育出版社
Higher Education Press
Prentice Hall
Pearson Education
出版集团

教 育 部 高 等 教 育 司 推 荐
国外优秀信息科学与技术系列教学用书

Intel 微处理器

——从 8086 到 Pentium 系列体系结构、编程与接口技术

（第五版 影印版）

THE INTEL MICROPROCESSORS

8086/8088,80186/80188,80286,80386,80486,Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing

(Fifth Edition)

Barry B. Brey

高 等 教 育 出 版 社
Pearson Education 出版集团

图字：01-2001-2167 号

图书在版编目(CIP)数据

Intel 微处理器:从 8086 到 Pentium 系列体系结构、编程与接口技术:5 版/(美)布雷(Brey, B.)(影印本). —北京:高等教育出版社,2001

ISBN 7-04-010098-3

Ⅰ.I… Ⅱ.布… Ⅲ.微处理器,Intel 系列-高等学校-教材-英文 Ⅳ.TP332

中国版本图书馆 CIP 数据核字(2001)第 045090 号

Intel 微处理器—从 8086 到 Pentium 系列体系结构、编程与接口技术(第五版 影印版)

Barry B. Brey

出版发行	高等教育出版社		
社　　址	北京市东城区沙滩后街 55 号	邮政编码	100009
电　　话	010—64054588	传　　真	010—64014048
网　　址	http://www.hep.edu.cn http://www.hep.com.cn		
经　　销	新华书店北京发行所		
印　　刷	北京外文印刷厂		
开　　本	850×1168　1/16	版　　次	2001 年 8 月影印版
印　　张	61.25	印　　次	2001 年 8 月第 1 次印刷
字　　数	1 464 000	定　　价	52.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

前 言

20 世纪末，以计算机和通信技术为代表的信息科学和技术，对世界的经济、军事、科技、教育、文化、卫生等方面的发展产生了深刻的影响，由此而兴起的信息产业已经成为世界经济发展的支柱。进入 21 世纪，各国为了加快本国的信息产业，加大了资金投入和政策扶持。

为了加快我国信息产业的进程，在我国《国民经济和社会发展第十个五年计划纲要》中，明确提出“以信息化带动工业化，发挥后发优势，实现社会生产力的跨越式发展。”信息产业的国际竞争将日趋激烈。在我国加入 WTO 后，我国信息产业将面临国外竞争对手的严峻挑战。竞争成败最终将取决于信息科学和技术人才的多少与优劣。

在 20 世纪末，我国信息产业虽然得到迅猛发展，但与国际先进国家相比，差距还很大。为了赶上并超过国际先进水平，我国必须加快信息技术人才的培养，特别要培养一大批具有国际竞争能力的高水平的信息技术人才，促进我国信息产业和国家信息化水平的全面提高。为此，教育部高等教育司根据教育部吕福源副部长的意见，在长期重视推动高等学校信息科学和技术教学的基础上，将实施超前发展战略，采取一些重要举措，加快推动高等学校的信息科学和技术等相关专业的教学工作。在大力宣传、推荐我国专家编著的面向 21 世纪和“九五”重点的信息科学和技术课程教材的基础上，在有条件的高等学校的某些信息科学和技术课程中推动使用国外优秀教材的影印版进行英语或双语教学，以缩短我国在计算机教学上与国际先进水平的差距，同时也有助于强化我国大学生的英语水平。

为了达到上述目的，在分析一些出版社已影印相关教材，一些学校已试用影印教材进行教学的基础上，教育部高等教育司组织并委托高等教育出版社开展国外优秀信息科学和技术优秀教材及其教学辅助材料的引进研究与影印出版的试点工作。为推动用影印版教材进行教学创造条件。

本次引进的系列教材的影印出版工作，是在对我国高校信息科学和技术专业的课程与美国高校的进行对比分析的基础上展开的；所影印出版的教材均由我国主要高校的信息科学和技术专家组成的专家组，从国外近两年出版的大量最新教材中精心筛选评审通

过的内容新、有影响的优秀教材；影印教材的定价原则上应与我国大学教材价格相当。

教育部高等教育司将此影印系列教材推荐给高等学校，希望有关教师选用，使用后有什么意见和建议请及时反馈。也希望有条件的出版社，根据影印教材的要求，积极参加此项工作，以便引进更多、更新、更好的外国教材和教学辅助材料。

同时，感谢国外有关出版公司对此项引进工作的配合，欢迎更多的国外公司关心并参与此项工作。

教育部高等教育司

二〇〇一年四月

PREFACE

This fifth edition is written for students who require a thorough knowledge of programming and interfacing of the Intel family of microprocessors. It is a very practical reference for anyone interested in all programming and interfacing aspects of this important microprocessor family. Today, anyone involved in a field of study that uses computers must understand assembly language programming and interfacing. Intel microprocessors have gained wide applications in many areas of electronics, communications, and control systems, particularly in desktop computer systems. Major additions to the textbook show how to interface C/C++ with assembly language, the USB interface, MMX technology extension, and the Pentium II. Also included are many updated sections detailing new events in the field of microprocessors and microprocessor interfacing.

ORGANIZATION AND COVERAGE

In order to cultivate a comprehensive approach to learning, each chapter begins with a set of objectives that briefly define its contents. Many programming applications that illustrate the main topics are then presented. A numerical summary, which doubles as a study guide, reviews the information presented. Finally, questions and problems are provided to promote practice and mental exercise with the concepts presented in the chapter.

Many examples using the Microsoft MACRO Assembler program provide an opportunity to learn how to program the Intel family of microprocessors. Operation of the programming environment includes the linker, library, macros, DOS function, BIOS functions, and C/C++ program development. The in-line assembler (C/C++) is illustrated for both the 16- and 32-bit programming environments.

Thorough descriptions of each family member, memory systems, and various I/O systems that include disk memory, ADC and DAC, 16550 UART, PIAs, timers, keyboard/display controllers, arithmetic coprocessors, and video display systems are also provided. Also discussed are the personal computer system buses (AGP, ISA, PCI, VESA, and USB). Through these systems, a practical approach to microprocessor interfacing is learned.

APPROACH

Because the Intel family of microprocessors is quite diverse, this text initially concentrates on real-mode programming, which is compatible with all versions of the Intel family of microprocessors. Instructions for each family member, which includes the 80386, 80486, Pentium, Pentium Pro, and Pentium II, are compared and contrasted with the 8086/8088 microprocessors. This entire series of microprocessors is very similar, which allows more advanced versions to be learned once the basic 8086/8088 is understood. Please note that the 8086/8088 microprocessors are still used in controllers along with their updated counterparts, the 80186/80188 and 80386EX embedded controllers.

This text also explains the programming and operation of the numeric coprocessor (8087/80287/80387/80486/80487/ Pentium/Pentium Pro/Pentium II, which functions in a system to provide access to floating-point calculations that are important in control systems, video graphics, and computer-aided design (CAD) applications. The numeric coprocessor allows a program to access complex arithmetic operations that are otherwise difficult to achieve with normal microprocessor programming.

Also described are the pin-outs and functions of the 8086–80486 and Pentium/Pentium Pro/Pentium II microprocessors. Interfacing is first developed by using the 8086/8088 with some of the more common peripheral components. After learning the basics, a more advanced emphasis is placed on the 80186/80188, 80386, 80486, Pentium, Pentium Pro, and Pentium II microprocessors. Because of its similarity to the 8086 and 80386, coverage of the 80286 is minimized so that the 80386, 80486, Pentium, Pentium Pro, and Pentium II can be covered in complete detail.

By studying the operation and programming of the microprocessor and numeric coprocessors, as well as interfacing all family members, you will have a working and practical background on the Intel family of microprocessors. Upon completing a course based on this text, you should be able to:

- Develop control software to control an application interface to the 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro, and Pentium II microprocessors. Generally, the software developed will also function on all versions of the microprocessor and also includes DOS-based applications.
- Program by using DOS function calls to control the keyboard, video display system, and disk memory in assembly language.
- Use the BIOS functions to control the keyboard, display, and various other components in the computer system.
- Develop software that uses macro sequences, procedures, conditional assembly, and flow control assembler directives.
- Develop software that uses interrupt hooks and hot-keys to gain access to terminate-and-stay-resident software.
- Develop programs that use both C/C++ and assembly language.
- Program the numeric coprocessor (80287/80387/80486/80487/Pentium/Pentium Pro/Pentium II) to solve complex equations.
- Detail the use of the MMX technology extension to the Pentium and Pentium II microprocessors.
- Explain the differences between the family members and highlight the features of each member.
- Describe and use real and protected mode operation of the 80286, 80386, 80486, Pentium, Pentium Pro, and Pentium II microprocessors.
- Interface memory and I/O systems to the microprocessor.
- Provide a detailed and comprehensive comparison of all family members, their software, and their hardware interface.
- Explain the operation of disk and video systems.
- Interface small systems to the ISA, VESA local, PCI bus, and USB in a personal computer system.

CONTENT OVERVIEW

Chapter 1 introduces the Intel family of microprocessors, with an emphasis on the microprocessor-based computer system. This chapter introduces the microprocessor, its history, its operation, and the methods used to store data in a microprocessor-based system. Chapter 2 explores the programming model of the microprocessor and system architecture. Both real- and protected-mode operation are explained in this chapter.

Chapters 3 though 6 explain how each instruction functions with the Intel family of microprocessors. As instructions are explained, simple applications are presented to illustrate the operation of the instructions and to develop basic programming concepts.

After the basis for programming is developed, Chapter 7 provides applications using the assembler program. These applications include programming using DOS and BIOS function calls, and the mouse function calls. Disk files are explained, as well as keyboard and video operation on a personal computer system. This chapter provides the tools required to develop virtually any program on a personal computer system. It also introduces the concept of interrupt hooks and hot-keys.

Chapter 8, new to this edition, shows how to incorporate assembly language procedures with C/C++. It also details the use of the in-line assembler program available in some versions of C/C++. Chapter 9 introduces the 8086/8088 family as a basis for learning the basic memory and I/O interfacing concepts that follow in later chapters. This chapter shows the buffered system and system timing.

Chapter 10 provides complete detail on memory interface, using both integrated decoders and programmable logic devices. Parity is illustrated, as well as dynamic memory systems. The 8-, 16-, 32-, and 64-bit memory systems are provided so that the 8086–80486 and Pentium/Pentium Pro/Pentium II microprocessors can be interfaced to memory.

Chapter 11 provides a detailed look at basic I/O interfacing by discussing PIAs, timers, keyboard/display interfaces, 16550 UART, and ADC/DAC. It also describes the interface of both DC and stepper motors. After these basic I/O components and their interface to the microprocessor are presented, Chapters 12 and 13 provide details on advanced I/O techniques that include interrupts and direct memory access (DMA). Applications include a printer interface, real-time clock, disk memory, and video systems.

Chapter 14 details the operation and programming for the 8087–Pentium II family of arithmetic coprocessors. Today few applications function efficiently without the power of the arithmetic coprocessor. Remember that all Intel microprocessors since the 80486 contain a coprocessor. The MMX technology extension, is also detailed, which is new to this edition.

Chapter 15 shows how to interface small systems to the personal computer through the use of the ISA, VESA, PCI bus, and USB interfaces. This chapter provides a launching point for the many cards being designed for use in the personal computer embedded in control systems in the industry.

Chapters 16 and 17 provide details on the advanced 80186/80188–80486 microprocessors. These chapters explore the differences between these microprocesors and the 8086/8088, as well as their enhancements and features. Cache, interleaved, and burst memory are described with the 80386 and 80486 microprocessors. Also described are memory management and memory paging.

Chapters 18 and 19 detail the Pentium, Pentium Pro, and Pentium II microprocessors. These microprocessors are based upon the original 8086/8088.

Four appendixes are included to enhance the application of the text:

1. Appendix A includes a complete listing of the DOS INT 21H function calls. This appendix also details the use of the assembler program and many of the BIOS function calls including BIOS function call INT 10H.
2. Appendix B includes a complete list of all 8086–Pentium II instructions, including many example instructions and machine coding in hexadecimal, as well as clock timing information.
3. Appendix C provides a compact list of all the instructions that change the flag bits.
4. Appendix D provides answers for the even-numbered questions and problems from the text.

ACKNOWLEDGMENTS

I would like to thank the following reviewers for their useful suggestions: Michael Agina, Texas Southern University; Soo-Young Lee, Auburn University; and Gursel Serpen, University of Toledo.

STAY IN TOUCH

You can stay in touch through the Internet. My Internet site contains information about all of my textbooks and has many important links that are specific to the personal computer, microprocessors, hardware, and software. Also available is a frequently updated lesson that details many of the aspects of the personal computer. My Internet address is

http://members.ee.net/~bbrey

责任编辑 张海波
封面设计 韩　冲
责任印制 陈伟光

CONTENTS

CHAPTER 1

Introduction to the Microprocessor and Computer

INTRODUCTION

This chapter provides an overview of the Intel family of microprocessors. Included is a discussion of the history of computers and the function of the microprocessor in the microprocessor-based computer system. Also introduced are terms and jargon used in the computer field, so **computerese** is understood and applied when discussing microprocessors and computers.

The block diagram and a description of the function of each block detail the operation of a computer system. The chapter also shows how the memory and input/output (I/O) system of the personal computer function. Finally, the way that data are stored in the memory is provided, so that each data type can be used as software is developed. Numeric data are stored as integers, floating-point, and binary-coded decimal (BCD); alphanumeric data are stored by using the ASCII (American Standard Code for Information Interchange) code.

CHAPTER OBJECTIVES

Upon completion of this chapter, you will be able to:

1. Converse by using appropriate computer terminology such as bit, byte, data, real memory system, expanded memory system (EMS), extended memory system (XMS), DOS, BIOS, I/O, and so forth.
2. Briefly detail the history of the computer and list applications performed by computer systems.
3. Provide an overview of the various 80X86 and Pentium—Pentium II family members.
4. Draw the block diagram of a computer system and explain the purpose of each block.
5. Describe the function of the microprocessor and detail its basic operation.
6. Define the contents of the memory system in the personal computer.
7. Convert between binary, decimal, and hexadecimal numbers.
8. Differentiate and represent numeric and alphabetic information as integers, floating-point, BCD, and ASCII data.

1–1 A HISTORICAL BACKGROUND

This first section outlines the historical events leading to the development of the microprocessor and, specifically, the extremely powerful and current 80X86,[1] Pentium, Pentium Pro,[2] and Pentium II microprocessors. Although a study of history is not essential to understand the microprocessor, it furnishes interesting reading and provides a historical perspective of the fast-paced evolution of the computer.

The Mechanical Age

The idea of a computing system is not new—it has been around long before modern electrical and electronic devices were developed. The idea of calculating with a machine dates to 500 B.C. when the Babylonians invented the **abacus,** the first mechanical calculator. The abacus, which used strings of beads to perform calculations, was used by the ancient Babylonian priests to keep track of their vast storehouses of grain. The abacus, which was used extensively and is still in use today, was not improved until 1642, when mathematician Blaise Pascal invented a calculator that was constructed of gears and wheels. Each gear contained 10 teeth that, when moved one complete revolution, advanced a second gear one place. This is the same principal that is used in the automobile's odometer mechanism and is the basis of all mechanical calculators. Incidentally, the PASCAL programming language is named in honor of Blaise Pascal for his pioneering work in mathematics and with the mechanical calculator.

The arrival of the first practical geared, mechanical machines used to automatically compute information dates to the early 1800s. This is before humans invented the light bulb or before much was known about electricity. In this dawn of the computer age, humans dreamed of mechanical machines that could compute numerical facts with a program—not merely calculating facts, as with a calculator.

In 1937 it was discovered through plans and journals that one early pioneer of mechanical computing machinery was Charles Babbage, aided by Augusta Ada Byron, the Countess of Lovelace. Babbage was commissioned in 1823 by the Royal Astronomical Society of Great Britain to produce a programmable calculating machine. This machine was to generate navigational tables for the Royal Navy. He accepted the challenge and began to create what he called his **Analytical Engine.** This engine was a mechanical computer that stored 1000 20-digit decimal numbers and a variable program that could modify the function of the machine to perform various calculating tasks. Input to his engine was through punched cards, much as computers in the 1950s and 1960s used punched cards. It is assumed that he obtained the idea of using punched cards from Joseph Jacquard, a Frenchman who used punched cards as input to a weaving machine he invented in 1801, which is today called Jacquard's loom. Jacquard's loom used punched cards to select intricate weaving patterns in the cloth that it produced. The punched cards programmed the loom.

After many years of work, Babbage's dream began to fade when he realized that the machinists of his day were unable to create the mechanical parts needed to complete his work. The Analytical Engine required more than 50,000 machined parts, which could not be made with enough precision to allow his engine to function reliably.

The Electrical Age

The 1800s saw the advent of the electric motor (conceived by Michael Faraday); with it came a multitude of motor-driven adding machines, all based on the mechanical calculator developed by

---

[1]80X86 is shorthand notation that embodies the 8086, 8088, 80188, 80286, 80386, and 80486 microprocessors.

[2]Pentium, Pentium Pro, and Pentium II are registered trademarks of Intel Corporation.

Blaise Pascal. These electrically driven mechanical calculators were common pieces of office equipment until well into the early 1970s, when the small hand-held electronic calculator, first introduced by Bomar, appeared. Monroe was also a leading pioneer of electronic calculators, but its machines were desktop, four-function models the size of cash registers.

In 1889, Herman Hollerith developed the punched card for storing data. Like Babbage, he too apparently borrowed the idea of a punched card from Jacquard. He also developed a mechanical machine—driven by one of the new electric motors—that counted, sorted, and collated information stored on punched cards. The idea of calculating by machinery intrigued the United States government so much that Hollerith was commissioned to use his punched-card system to store and tabulate information for the 1890 census.

In 1896, Hollerith formed a company called the Tabulating Machine Company, which developed a line of machines that used punched cards for tabulation. After a number of mergers, the Tabulating Machine Company was formed into the International Business Machines Corporation, now referred to more commonly as IBM, Inc. The punched cards used in computer systems are often called **Hollerith cards,** in honor of Herman Hollerith. The 12-bit code used on a punched card is called the **Hollerith code.**

Mechanical machines driven by electric motors continued to dominate the information processing world until the construction of the first electronic calculating machine in 1941 by a German inventor named Konrad Zuse. His calculating computer, the Z3, was used in aircraft and missile design during World War II for the German war effort. Had Zuse been given adequate funding by the German government, he most likely would have developed a much more powerful computer system. Zuse is today finally receiving some belated honor for his pioneering work in the area of digital electronics which began in the 1930s and for his Z3 computer system.

It has recently been discovered (through the declassification of British military documents) that the first electronic computer was placed into operation in 1943 to break secret German military codes. This first electronic computing system, which used vacuum tubes, was invented by Alan Turing. Turing called his machine **Colossus,** probably because of its size. A problem with Colossus was that although its design allowed it to break secret German military codes generated by the mechanical **Enigma machine,** it could not solve other problems. Colossus was not programmable—it was a fixed-program computer system, which today is often called a **special-purpose computer.**

The first general-purpose, programmable electronic computer system was developed in 1946 at the University of Pennsylvania. This first modern computer was called the ENIAC (**Electronics Numerical Integrator and Calculator).** The ENIAC was a huge machine, containing over 17,000 vacuum tubes and over 500 miles of wires. This massive machine weighted over 30 tons, yet performed only about 100,000 operations per second. The ENIAC thrust the world into the age of electronic computers. The ENIAC was programmed by rewiring its circuits—a process that took many workers several days to accomplish. The workers changed the electrical connections on plug-boards that looked like early telephone switchboards. Another problem with the ENIAC was the life of the vacuum tube components, which required frequent maintenance.

Breakthroughs that followed were the development of the transistor in 1948 at Bell Labs, followed by the 1958 invention of the integrated circuit by Jack Kilby of Texas Instruments. The integrated circuit led to the development of digital integrated circuits (RTL, or resistor-to-transistor logic) in the 1960s and the first microprocessor at Intel Corporation in 1971. At that time, Intel and one of its engineers, Marcian E. Hoff, developed the 4004 microprocessor—the device that started the microprocessor revolution that continues today at an ever-accelerating pace.

Programming Advancements

Now that programmable machines were developed, programs and programming languages began to appear. As mentioned earlier, the first programmable electronic computer system was programmed by rewiring its circuits. Because this proved too cumbersome for practical application, early in the evolution of computer systems, computer languages began to appear in order to control the computer. The first such language, **machine language,** was constructed of ones and zeros using binary codes that were stored in the computer memory system as groups of instructions called programs. This was more efficient than rewiring a machine to program it, but it was still extremely time-consuming to develop a program because of the sheer number of codes that were required. Mathematician John von Neumann was the first person to develop a system that accepted instructions and stored them in memory. Computers are often called **von Neumann machines** in honor of John von Neumann. (Remember that Babbage also had developed the concept long before von Neumann.)

Once computer systems such as the UNIVAC became available in the early 1950s, **assembly language** was used to simplify the chore of entering binary code into a computer as its instructions. The assembler allowed the programmer to use mnemonic codes, such as ADD for addition, in place of a binary number such as 01000111. Although assembly language was an aid to programming, it wasn't until 1957, when Grace Hopper developed the first high-level programming language called **FLOW-MATIC,** that computers became easier to program. In the same year, IBM developed FORTRAN **(FORMula TRANslator)** for its computer systems. The FORTRAN language allowed programmers to develop programs that used formulas to solve mathematical problems. Note that FORTRAN is still used by some scientists for computer programming. Another similar language, introduced about a year after FORTRAN, was ALGOL **(ALGOrithmic Language).**

The first truly successful and widespread programming language for business applications was COBOL **(COmputer Business Oriented Language).** Although COBOL usage has diminished somewhat in recent years, it is still a major player in many large business systems. Another once-popular business language is RPG **(Report Program Generator)**, which allows programming by specifying the form of the input, output, and calculations.

Since these early days of programming, additional languages have appeared. Some of the more common are BASIC, C/C++, PASCAL, and ADA. The BASIC and PASCAL languages were both designed as teaching languages, but have escaped the classroom and are used in many computer systems. The BASIC language is probably the easiest of all to learn. Some estimates indicate that the BASIC language is used in the personal computer for 80 percent of the programs written by users. Recently, a new version of BASIC, VISUAL BASIC, has made programming in the WINDOWS environment easier. The VISUAL BASIC language may eventually supplant C/C++ and PASCAL.

In the scientific community, C/C++ and (occasionally) PASCAL appear as control programs. Both languages, especially C/C++, allow the programmer almost complete control over the programming environment and computer system. In many cases, C/C++ is replacing some of the low-level, machine control software normally reserved for assembly language. Even so, assembly language still plays an important role in programming. Most video games written for the personal computer are written almost exclusively in assembly language. Assembly language is also interspersed with C/C++ and PASCAL to perform machine control functions efficiently.

The ADA language is used heavily by the Department of Defense. The ADA language was named in honor of Augusta Ada Byron, Countess of Lovelace. The Countess worked with Charles Babbage in the early 1800s in the development of his Analytical Engine.

The Microprocessor Age

The world's first microprocessor, the Intel 4004, was a 4-bit microprocessor—a programmable controller on a chip. It addressed a mere 4096 4-bit wide memory locations. (A **bit** is a binary digit with a value of one or zero. A 4-bit wide memory location is often called a **nibble.**) The 4004 instruction set contained only 45 instructions. It was fabricated with the then-current state-of-the-art

P-channel MOSFET technology that only allowed it to execute instructions at the slow rate of 50 KIPs **(kilo-instructions per second).** This was slow when compared to the 100,000 instructions executed per second by the 30-ton ENIAC computer in 1946. The main difference was that the 4004 weighed much less than an ounce.

At first, applications abounded for this device. The 4-bit microprocessor debuted in early video game systems and small microprocessor-based control systems. One such early video game, a shuffleboard game, was produced by Balley. The main problems with this early microprocessor were its speed, word width, and memory size. The evolution of the 4-bit microprocessor ended when Intel released the 4040, an updated version of the earlier 4004. The 4040 operated at a higher speed, although it lacked improvements in word width and memory size. Other companies, particularly Texas Instruments (TMS-1000), also produced 4-bit microprocessors. The 4-bit microprocessor still survives in low-end applications such as microwave ovens and small control systems, and is still available from some microprocessor manufacturers. Most calculators are still based on 4-bit microprocessors that process 4-bit BCD **(binary-coded decimal)** codes.

Later in 1971, realizing that the microprocessor was a commercially viable product, Intel Corporation released the 8008—an extended 8-bit version of the 4004 microprocessor. The 8008 addressed an expanded memory size (16K bytes) and contained additional instructions (a total of 48) that provided an opportunity for its application in more advanced systems. (A **byte** is generally an 8-bit wide binary number and a **K** is 1024. Often, memory size is specified in K bytes.)

As engineers developed more demanding uses for the 8008 microprocessor, they discovered that its somewhat small memory size, slow speed, and instruction set limited its usefulness. Intel recognized these limitations and introduced the 8080 microprocessor in 1973—the first of the modern 8-bit microprocessors. About six months after Intel released the 8080 microprocessor, Motorola Corporation introduced its MC6800 microprocessor. The floodgates opened and the 8080—and, to a lesser degree, the MC6800—ushered in the age of the microprocessor. Soon, other companies began to introduce their own versions of the 8-bit microprocessor. Table 1–1 lists several of these early microprocessors and their manufacturers. Of these early microprocessor producers, only Intel and Motorola continue successfully to create newer and improved versions of the microprocessor. Zilog still manufactures microprocessors, but has remained in the background, concentrating on microcontrollers and embedded controllers instead of general-purpose microprocessors. Rockwell has all but abandoned microprocessor development in favor of modem circuitry. Motorola has declined from having nearly 50 percent share of the microprocessor market to a much smaller share.

What Was Special about the 8080? Not only could the 8080 address more memory and execute additional instructions, but it executed them 10 times faster than the 8008. An addition that took 20 μs (50,000 instructions per second) on an 8008-based system required only 2.0 μs (500,000 instructions per second) on an 8080-based system. Also, the 8080 was compatible with TTL (transistor-transistor logic), whereas the 8008 was not directly compatible. This made interfacing

TABLE 1–1 Early 8-bit microprocessors.

Manufacturer	*Part Number*
Fairchild	F-8
Intel	8080
MOS Technology	6502
Motorola	MC6800
National Semiconductor	IMP-8
Rockwell International	PPS-8
Zilog	Z-8