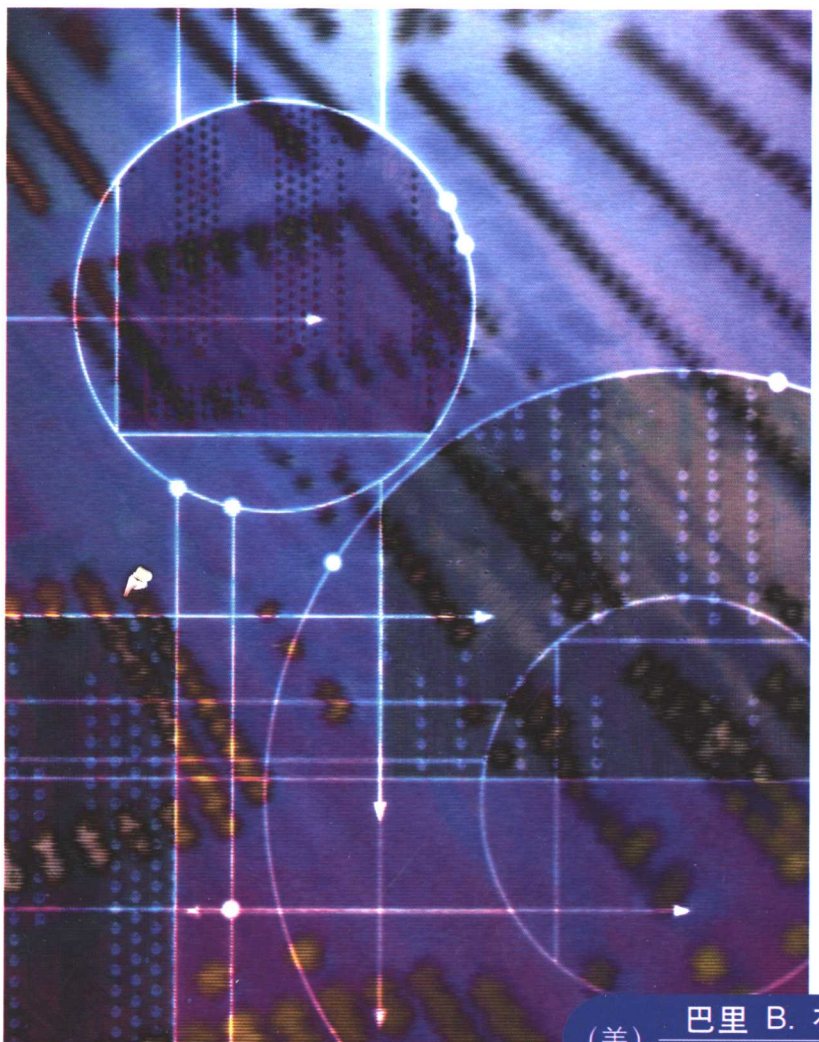
经 典 原 版 书 库

Intel系列微处理器体系结构、编程与接口

（英文版 · 第6版）



本书只供在中国大陆销售

（美）巴里 B. 布雷 著
DeVry理工学院

机械工业出版社
China Machine Press

经 典 原 版 书 库

Intel系列微处理器体系结构、编程与接口

（英文版 · 第6版）

（美） 巴里 B. 布雷
DeVry理工学院 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2005-1619

图书在版编目（CIP）数据

Intel系列微处理器体系结构、编程与接口（英文版·第6版）/（美）布雷（Brey, B. B.）著. -北京：机械工业出版社，2005.4

（经典原版书库）

书名原文：The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, and Pentium 4: Architecture, Programming, and Interfacing, Sixth Edition

ISBN 7-111-16052-5

I. I… II. 布… III. 微处理器-英文 IV. TP332

中国版本图书馆CIP数据核字（2005）第017460号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：迟振春

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2005年4月第1版第1次印刷

787mm×1092mm 1/16·64印张

印数：0 001-3 000册

定价：99.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：(010) 68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周立柱	周克定	周傲英	孟小峰	岳丽华
范　明	郑国梁	施伯乐	钟玉琢	唐世渭
袁崇义	高传善	梅　宏	程　旭	程时端
谢希仁	裘宗燕	戴　葵		

秘　书　组

武卫东　　温莉芳　　刘　江　　杨海玲

前　言

这本实用的教材全面介绍了Intel系列微处理器程序设计和接口技术。当今，任何在计算机应用领域工作或将要为之奋斗的人，都必须懂得汇编语言程序设计和接口技术，因为Intel微处理器在电子、通信和控制系统，特别是桌面计算机系统等众多领域得到了广泛的应用。本书第6版增加的主要内容是如何在DOS和Windows环境下，使C/C++与汇编语言接口，更新的章节增加了微处理器和微处理器接口技术的新成果。

本书编排和范围

为了使学生更好地掌握本书内容，每章开始都有一组学习目标，扼要地确定其内容，每章都包括许多阐明主题的程序设计应用，并以总结结束。总结部分归纳了刚学到的内容。最后是习题部分，读者通过实践和练习可进一步掌握每章中的概念。

本书使用微软宏汇编程序和Visual C++环境作为实例环境，为学习编写Intel系列微处理器程序提供了实践机会。程序设计环境的内容包括连接器、库、宏、DOS功能调用、BIOS功能调用和Visual C/C++程序开发环境等。本书说明了各种版本Visual C++的16位和32位两种内嵌式汇编的编程环境。

本书也提供了对系列成员、存储器系统以及各种I/O系统的全面描述，I/O系统包括磁盘存储器、ADC和DAC、16550 UART、PIA、计时器、键盘/显示控制器、算术协处理器以及视频显示系统。讨论了个人计算机系统总线（AGP、ISA、PCI、VESA和USB）。通过这些系统，可以学到有关微处理器接口技术的实用方法。

使用本书的方法

由于Intel系列微处理器各不相同，本书一开始集中讨论实模式下的程序设计。实模式程序设计与Intel系列微处理器的所有版本兼容，系列的每个成员，包括80386、80486、Pentium、Pentium Pro、Pentium Ⅱ、Pentium Ⅲ和Pentium 4微处理器，都是与8086/8088微处理器相对照进行说明的，所有的微处理器非常类似，一旦理解了基本的8086/8088就可以学习更高版本的微处理器。请注意，8086/8088仍然用于控制器以及它们的升级产品，即80186/80188和80386EX嵌入式控制器。

本书也解释了算术协处理器的程序设计和操作。算术协处理器在系统中提供了浮点计算的能力。浮点计算在控制系统、视频图像和计算机辅助设计（CAD）应用中非常重要。算术协处理器使程序能够完成复杂的算术运算，而用普通的微处理器编程方法是很难完成这些算术运算的。

本书也描述了8086~80486以及Pentium所有微处理器的引脚和功能。首先研究了用于8086/8088的一些通用外围接口部件。在解释了基本微处理器之后，强调了更高级的80186/80188、

80386、80486和Pentium~Pentium 4微处理器。由于80286与8086和80386类似，所以减少了80286的内容，从而可以全面详细地介绍80386、80486和Pentium版本微处理器。

通过学习各种先进微处理器的运算和程序设计以及系列所有成员的接口技术，读者将会具有在Intel系列微处理器环境下工作和实践的能力。通过本书的学习，读者可以掌握以下内容：

1) 开发控制软件，控制微处理器应用接口。通常，开发的软件应能在微处理器的所有版本上运行，也包括基于DOS和Windows的应用。

2) 使用汇编语言，通过DOS功能调用来编写控制键盘、视频显示系统和磁盘存储器的程序。

3) 使用BIOS功能控制计算机系统中的键盘、显示器和各种其他部件。

4) 使用宏指令、过程、条件汇编和流控制汇编伪指令开发软件。

5) 使用中断钩连和热键开发软件，使其能够获得中断并驻留内存程序。

6) 设计算术协处理器以求解复杂方程。

7) 解释Intel系列成员之间的区别及其特征。

8) 描述并使用微处理器的实模式和保护模式进行操作。

9) 设计微处理器到存储器和I/O系统的接口。

10) 比较Intel系列微处理器及其软件和硬件接口。

11) 解释在嵌入式应用中实时操作系统的功能。

12) 解释磁盘和视频系统的操作。

13) 使用ISA、VESA local、PCI、并行端口和USB总线接口，建立小型系统与个人计算机系统的连接。

内容概览

第1章介绍了Intel系列微处理器的历史、操作以及基于微处理器系列存储数据的方法。在本书第6版中还包括数字系统。第2章揭示了微处理器程序设计模型和系统体系结构，解释了实模式和保护模式操作。

介绍了基本的计算机系统之后，第3~6章解释了Intel系列微处理器每条指令的功能。在解释指令的同时，用简单的应用程序说明指令的操作，并阐述程序设计的基本概念。

在讲解了程序设计的基础后，第7章介绍汇编语言应用程序。这些应用程序包括使用DOS和BIOS功能调用以及鼠标功能调用的编程。该章也解释了在个人计算机系统中的磁盘文件、键盘和视频操作，还提供了在个人计算机系统上实际开发程序所需的工具，介绍了中断钩连和热键的概念。

第8章介绍了如何使用C/C++和嵌入式汇编或分离汇编语言混合编写程序模块。

第9章介绍了8086/8088系列微处理器，为后续章节介绍存储器和I/O接口技术提供了基础。该章还提到了缓冲系统以及系统定时器。

第10章解释了使用集成解码器和可编程逻辑器件的存储器接口，说明了奇偶校验和动态存储器系统，介绍了为8086~80486和Pentium~Pentium 4微处理器提供8位、16位、32位和64位的存储器系统接口。

第11章通过讨论PIA、计时器、键盘/显示器接口、16550 UART和ADC/DAC详细说明了基

本I/O接口技术。这一章还描述了直流电机和步进电机的接口。

在介绍微处理器的基本I/O部件和它们与微处理器的接口后，第12、13章详细描述了包括中断和直接存储器存取的高级I/O技术，还讲解了这些技术的应用，包括打印机接口、实时钟、磁盘存储器和视频显示系统。

第14章详细叙述了针对8087~Pentium 4系列算术协处理器的操作和编程设计以及MMX指令。目前几乎没有不利用算术协处理器就能高效运行的应用程序。记住，自80486以来的所有Intel微处理器都有协处理器。

第15章阐明了如何使用并行端口、ISA、VESA和PCI总线接口，将小型系统与个人计算机连接。该章是新增加的一章，讨论了许多正在设计的、用于将个人计算机嵌入在工业控制系统中的板卡。

第16、17章讨论了高级80186/80188~80486微处理器、它们与8086/8088微处理器的区别以及它们的增强功能和特征，讲述了用于80386和80486微处理器的高速缓冲存储器、交叉存储器和猝发存储器。第17章还描述了存储器管理和存储器分页。

第18章详细阐述了Pentium和Pentium Pro微处理器。这些新的微处理器是基于最初的8086/8088微处理器开发的。

第19章介绍了Pentium Ⅱ、Pentium Ⅲ和Pentium 4微处理器，涉及了一些新的特征、封装以及增加到原指令集的SIMD指令。

本书包括4个附录，增加了本书的实用性。

附录A提供了DOS INT 21H功能调用的完全列表，还详细讨论了汇编程序的使用和许多BIOS功能调用，包括BIOS功能调用INT 10H以及鼠标功能调用和DPMI调用。

附录B包括所有8086~Pentium 4指令的完全列表，包括许多指令实例和十六进制机器码以及时钟周期数信息，还提供了SIMD指令以及它的应用实例。

附录C提供了改变标志位的所有指令的列表。

附录D提供了各章偶数编号习题的答案。

致谢

感谢以下专家为本书提供的宝贵评论和建议：美国田纳西州曼菲斯大学的Robert L. Douglas；加利福尼亚州立大学的Isaac Ghansah；路易斯安那州Southern University and A&M College的Raynaud F. Henton；犹他州立大学的Paul A. Wheeler。

联络方式

读者可以通过互联网与本书作者保持联系。作者已经出版教材的信息可以在http://members.ee.net/brey上找到。该网站还提供与个人计算机微处理器硬件和软件相关的信息。每周登载一期详细介绍个人计算机的讲座内容，请特别关注“Technical Stuff”部分，其中包含许多本书中未涉及的技术主题。

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 4 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, Pentium III, and Pentium 4[2] 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 includes the 8086, 8088, 80188, 80286, 80386, and 80486 microprocessors.

[2]Pentium, Pentium Pro, Pentium II, Pentium III, and Pentium 4 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 Z3 calculating computer, as pictured in Figure 1–1, 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.

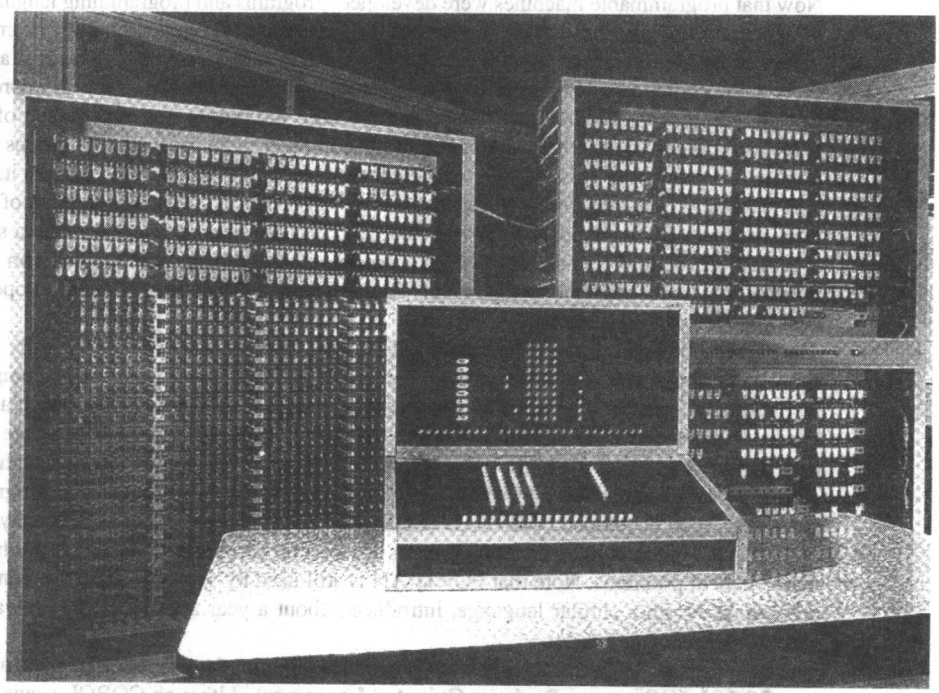


FIGURE 1–1 The Z3 computer developed by Konrad Zuse used a 5.33 Hertz clocking frequency. (Photo courtesy of Horst Zuse, the son of Konrad.)

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 weighed 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,