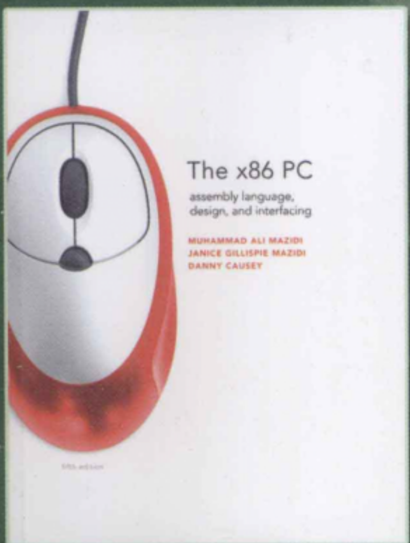
国外计算机科学教材系列

PEARSON

x86 PC 汇编语言、设计与接口（第五版）

The x86 PC Assembly Language, Design, and Interfacing
Fifth Edition



英文版

Muhammad Ali Mazidi
[美] Janice Gillispie Mazidi 著
Danny Causey

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

权威经典

x86 PC 汇编语言、设计与接口（第五版）

The x86 PC
Assembly Language, Design, and Interfacing
Fifth Edition

英文版

本书深入介绍了x86体系结构、总线、接口技术、系统编程、IEEE浮点数学、USB、高速缓存以及RISC和Harvard体系结构，逐步并系统地介绍了x86汇编语言编程和PC体系结构的基础知识，为读者提供了有趣且方便的学习经验，并用大量的示例及习题来加深读者对概念的理解，是高校计算机专业学生及x86嵌入式设计人员的理想教材和参考书。

本书特点

- 覆盖了从8086至64位Itanium的所有x86微处理器。
- 使用汇编和C程序示例来深入介绍x86 PC体系结构。
- 使用示例介绍了x86指令的用法。
- 介绍了IEEE浮点数和数学协处理器的基本知识。
- 探讨和分析了16位、32位和64位处理器（如Pentium和Itanium芯片）间的硬件不同点。
- 讨论了x86微处理器的8位、16位和32位接口。
- 通过使用来自IBM PC技术参考中的程序片断，显示了PC系统编程的真实方法。
- 简单介绍了USB端口以及如何使用C#来访问它的方法。
- 对x86 CPU和RISC处理器进行了比较。
- 考察了x86高速缓存及其组织结构。
- 覆盖了从Intel至AMD的x86处理器的64位新特性。
- 探讨了x86处理器的超标量架构及其多核特性。

说明：为降低图书厚度，本书中删除了英文版原书中的附录A至附录F。附录A至附录F的内容放在华信教育资源网（www.huaxin.edu.cn）上本书的宣传页面，需要的读者可自该页面上下载。

PEARSON
www.PearsonEd.com

ISBN 978-7-121-09770-6

策划编辑：谭海平
责任编辑：谭海平
责任美编：喻　晓

定价：75.00 元

国外计算机科学教材系列

x86 PC 汇编语言、设计与接口

（第五版）

（英文版）

The x86 PC Assembly Language, Design,

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

内容简介

汇编语言与程序设计是计算机类学科专业的重要的专业技术基础课程，是以应用为主的工程技术基础类课程。本书深入介绍了x86体系结构、总线、接口技术、系统编程、IEEE浮点数学、USB、高速缓存以及RISC和Harvard体系结构。全书包括两大部分内容：(1) 汇编语言程序设计；(2) IBM PC兼容计算机接口设计。在介绍汇编语言程序的章节中，以汇编方法为引导，逐步加入各种语句及指令，给出了很多程序实例，并通过Debug实用工具展示出程序指令执行的具体动作。在接口设计方面，从PC所应用的芯片到设备、从电路设计到编程都进行了详细的阐述。

本书内容系统，示例丰富，可作为高校计算机专业学生的教材，也是x86嵌入式设计人员的理想参考书。

图书在版编目（CIP）数据

x86 PC汇编语言、设计和接口：第5版 = The x86 PC: Assembly Language, Design, and Interfacing, 5e: 英文 /（美）马兹迪（Mazidi, M. A.),（美）马兹迪（Mazidi, J.),（美）科西（Causey, C.）著. —北京：电子工业出版社，2009.11

（国外计算机科学教材系列）

ISBN 978-7-121-09770-6

Ⅰ. x…　Ⅱ. ①马…　②马…　③科…　Ⅲ. 汇编语言 – 程序设计 – 教材 – 英文　Ⅳ. TP313

中国版本图书馆CIP数据核字（2009）第195811号

策划编辑：谭海平

责任编辑：谭海平

印　　刷：北京市天竺颖华印刷厂

装　　订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱　邮编：100036

开　　本：787 × 980　1/16　印张：44.75　字数：1145千字

印　　次：2009年11月第1次印刷

印　　数：3000册　定价：75.00元

凡所购买电子工业出版社的图书有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

出 版 说 明

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

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

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

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

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

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

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

电子工业出版社

教材出版委员会

PREFACE

Purpose

This book is intended for use in college-level courses in which both Assembly language programming and x86 PC interfacing are discussed. It not only builds the foundation of Assembly language programming, but also provides a comprehensive treatment of x86 PC design and interfacing for students in engineering and computer science disciplines. This volume is intended for those who wish to gain an in-depth understanding of the internal working of the x86 PC. It builds a foundation for the design and interfacing of microprocessor-based systems using the real-world example of the x86 PC. In addition, it can also be used by practicing technicians, hardware engineers, computer scientists, and hobbyists who want to do PC interfacing and data acquisition.

Prerequisites

Readers should have taken an introductory digital course. Knowledge of other programming languages would be helpful, but is not necessary.

Although a vast majority of current PCs use x86 such as Pentium microprocessors, their design is based on the IBM PC/AT, an 80286 microprocessor system introduced in 1984. A good portion of the features of the PC/AT, hence its limitations, are based on the original IBM PC, an 8088 microprocessor system, introduced in 1981. In other words, one cannot expect to understand fully the architectural philosophy of the x86 PC and its internal architecture unless the 80286 PC/AT and its subset, the IBM PC/XT, are first understood. For this reason, we describe the 8088 and 80286 microprocessors in Chapter 9.

Contents

A systematic, step-by-step approach has been used in covering various aspects of Assembly language programming. Many examples and sample programs are given to clarify concepts and provide students an opportunity to learn by doing. Review questions are provided at the end of each section to reinforce the main points of the section. We feel that one of the functions of a textbook is to familiarize the student with terminology used in technical literature and in industry, so we have followed that guideline in this text.

Chapter 0 covers concepts in number systems (binary, decimal, and hex) and computer architecture. Most students will have learned these concepts in previous courses, but Chapter 0 provides a quick overview for those students who have not learned these concepts, or who may need to refresh their memory.

Chapter 1 provides a brief history of the evolution of x86 microprocessors and an overview of the internal workings of the 8086 as a basis of all x86 processors. Chapter 1 should be used in conjunction with Appendix A (a tutorial introduction to DEBUG) so that the student can experiment with concepts being learned on the x86 PC. The order of topics in Appendix A has been designed to correspond to the order of topics presented in Chapter 1. Thus, the student can begin programming with DEBUG without having to learn how to use an assembler.

Chapter 2 explains the use of assemblers to create programs. Although the programs in the book were developed and tested with Microsoft's MASM assembler, any Intel-compatible assembler such as Borland's TASM may be used.

Chapter 3 introduces the bulk of the logic and arithmetic instructions for unsigned numbers, plus bitwise operations in C.

Chapter 4 introduces DOS and BIOS interrupts. Programs in Assembly allow the student to get input from the keyboard and send output to the monitor. In addition, macro programming in assembly is described.

Chapter 5 describes how to program the keyboard and mouse.

Chapter 6 covers arithmetic and logic instructions for signed numbers as well as string processing instructions.

Chapter 7 discusses modular programming and how to develop larger Assembly language programs by breaking them into smaller modules to be coded and tested separately. In addition, in-line Assembly language within C programs is explained.

Chapter 8 introduces some 32-bit concepts of x86 programming. Although this book emphasizes 16-bit programming, the 386 is introduced to help the student appreciate the power of 32-bit CPUs.

Chapter 9 describes the 8088 and 286 microprocessors and supporting chips in detail and shows how they are used in the original IBM PC/XT/AT. In addition, the origin and function of the address, data, and control signals of the ISA expansion slot are described.

Chapter 10 provides an introduction to various types of RAM and ROM memories, their interfacing to the microprocessor, the memory map of the x86 PC, the timing issue in interfacing memory, and the checksum byte and parity bit techniques of ensuring data integrity in RAM and ROM.

Chapter 11 is dedicated to the interfacing of I/O ports, the use of IN and OUT instructions in the x86, and interfacing and programming of the 8255 programmable peripheral chip. We describe I/O programming in several languages, as well.

Chapter 12 covers the interfacing of PCs to devices for data acquisition such as LCDs, stepper motors, ADC, DAC, and sensors.

Chapter 13 discusses the use of the 8253/54 timer chip in the x86 PC, as well as how to generate music and time delays.

Chapter 14 is dedicated to the explanation of hardware and software interrupts, the use of the 8259 interrupt controller, the origin and assignment of IRQ signals on the expansion slots of the ISA bus, and exception interrupts in 80x86 microprocessors.

Chapter 15 is dedicated to direct memory access (DMA) concepts, the use of the 8237 DMA chip in the x86 PC, and DMA channels and associated signals on the ISA bus.

Chapter 16 covers the basics of video monitors and various video modes and adapters of the PC, in addition to the memory requirements of various video boards in graphics mode.

Chapter 17 discusses serial communication principles and programming of the PC COM port in Assembly and C.

Chapter 18 covers the interfacing and programming of the keyboard in the x86 PC, in addition to printer port interfacing and programming. In addition, a discussion of various types of parallel ports such as EPP and ECP is included.

Chapter 19 discusses hard disk storage organization and terminology.

Chapter 20 examines the x87 math coprocessor, its programming, and IEEE single- and double-precision floating point data types.

Chapter 21 explores the programming and hardware of the 386 microprocessor, contrasts and explains real and protected modes, and discusses the implementation of virtual memory.

Chapter 22 is dedicated to the interfacing of high-speed memories and describes various types of DRAM, including EDO and SDRAM, and examines cache memory and various cache organizations and terminology in detail.

In Chapter 23 we describe the main features of the 486, Pentium, and Pentium Pro and compare these microprocessors with the RISC processors. Chapter 23 also provides a discussion of MMX technology and how to write programs to detect which CPU a PC has.

Chapter 24 examines the new generation of 64-bit microprocessors from Intel and AMD.

Chapter 25 provides an overview of the IC technology and failure analysis, describes IC interfacing and system design issues, and covers error detection and correction.

Chapter 26 is dedicated to the discussion of the various types of PC buses, such as ISA and PC104, their performance comparisons, and features of the PCI bus.

Chapter 27 describes the USB port in detail and shows how to use the C programming language to access USB devices connected to the USB port of x86 PCs.

Appendices[1]

The appendices have been designed to provide all reference material required for the topics covered in this combined volume so that no additional references should be necessary.

Appendix A provides a tutorial introduction to DEBUG. Appendix B provides a listing of Intel's 8086 instructions. Appendix C describes assembler directives with examples of their use. Appendix D lists commonly used interrupt fuction calls and legacy software. Appendix E lists the I/O maps of x86 PCs. Appendix F provides a table of ASCII codes.

Lab Manual

The lab manual contains some very basic labs and can be found at the **www.MicroDigitalEd.com** website. The more advanced and rigorous lab assignments are left up to the instructor depending on the course objectives, class level, and whether the course is graduate or undergraduate. The support materials for this and other books by the authors can be found on this website, too.

Solutions Manual/PowerPoint® Slides

The end-of-chapter problems cover some very basic concepts. The more challenging and rigorous homework assignments are left up to the instructor depending on the course objectives, class level, and whether the course is graduate or undergraduate. The solutions manual was produced with the help of Mr. Sepehr Naimi. The solutions manual and PowerPoint® slides for the drawings are available online for instructors only.

Online Instructor Resources

To access supplementary materials online, instructors need to request an instructor access code. Go to **www.prenhall.com**, click the **Instructor Resource Center** link, and then click **Register Today** for an instructor access code. Within 48 hours after registering you will receive a confirming e-mail including an instructor access code. Once you have received your code, go to the site and log on for full instructions on downloading the materials you wish to use.

Acknowledgments

This book is the result of the dedication, work, and love of many individuals. Thanks must go to the many students and professors around the world whose comments have helped to shape this book since its original publication in 1992. Special thanks to Dimitri Moonen for his detailed reading of the book and pointing out errors. Thanks to Pedran Mazidi for redoing the tables and figures for this new edition.

We would like to thank the people at Prentice Hall, in particular our editor Wyatt Morris, who continues to support and encourage our writing, and our project manager Rex Davidson, who made the book a reality. We were lucky to get the best copy editor in the world, Bret Workman. Thank you for your fantastic job, as usual.

Finally, we would like to sincerely thank Dr. Roger S. Walker of the Computer Science Engineering Department, University of Texas at Arlington for his constant encouragement. We enjoyed writing this book, and hope you enjoy reading it and using it for your courses and projects. Please let us know if you have any suggestions or find any errors.

---

① 为降低图书的厚度，本书中删除了英文原书中的附录A至附录F。附录A至附录F的内容放在华信教育资源网（www.huaxin.edu.cn）本书的宣传页上，需要的读者可自该页面上下载。——编者注

ABOUT THE AUTHORS

Muhammad Ali Mazidi went to Tabriz University and holds Master's degrees from both Southern Methodist University and the University of Texas at Dallas. He is currently a.b.d. on his Ph.D. in the Electrical Engineering Department of Southern Methodist University. He is co-author of some widely used textbooks, including *The 8051 Microcontroller and Embedded Systems, The PIC Microcontroller and Embedded Systems,* and *The HCS12 Microcontroller and Embedded Systems,* also available from Prentice Hall. He teaches microprocessor-based system design at DeVry University in Dallas, Texas. He is the founder of MicroDigitalEd.com, which provides support for this book.

Janice Gillispie Mazidi has a Master of Science degree in Computer Science from the University of North Texas. She has several years experience as a software engineer in Dallas. She is currently teaching computer science and math courses in a local high school.

Danny Causey is a graduate of DeVry University and holds a Bachelor's of Science in Computer Engineering Technology. He is co-author of *The PIC Microcontroller and Embedded Systems* and *The HCS12 Microcontroller and Embedded Systems*. He is a partner in MicroDigitalEd.com. He works in the software development field, providing tools to accelerate daily business.

The authors can be contacted at the following email addresses if you have any comments or suggestions, or if you find any errors.

mdebooks@yahoo.com
mmazidi@microdigitaled.com
dcausey@microdigitaled.com

CONTENTS

CHAPTER 0

INTRODUCTION TO COMPUTING

OBJECTIVES

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

>> Convert any number from base 2, base 10, or base 16 to either of the other two bases
>> Add and subtract hex numbers
>> Add binary numbers
>> Represent any binary number in 2's complement
>> Represent an alphanumeric string in ASCII code
>> Describe logical operations AND, OR, NOT, XOR, NAND, NOR
>> Use logic gates to diagram simple circuits
>> Explain the difference between a bit, a nibble, a byte, and a word
>> Give precise mathematical definitions of the terms *kilobyte*, *megabyte*, *gigabyte*, and *terabyte*
>> Explain the difference between RAM and ROM and describe their use
>> Describe the purpose of the major components of a computer system
>> List the three types of buses found in computers and describe the purpose of each type of bus
>> Describe the role of the CPU in computer systems
>> List the major components of the CPU and describe the purpose of each

To understand the software and hardware of a microcontroller-based system, one must first master some very basic concepts underlying computer design. In this chapter (which in the tradition of digital computers is called Chapter 0), the fundamentals of numbering and coding systems are presented. After an introduction to logic gates, an overview of the workings inside the computer is given. Finally, in the last section we give a brief history of CPU architecture. Although some readers may have an adequate background in many of the topics of this chapter, it is recommended that the material be scanned, however briefly.

SECTION 0.1: NUMBERING AND CODING SYSTEMS

Whereas human beings use base 10 (*decimal*) arithmetic, computers use the base 2 (*binary*) system. In this section we explain how to convert from the decimal system to the binary system, and vice versa. The convenient representation of binary numbers in base 16, called *hexadecimal,* also is covered. Finally, the binary format of the alphanumeric code, called *ASCII,* is explored.

Decimal and binary number systems

Although there has been speculation that the origin of the base 10 system is the fact that human beings have 10 fingers, there is absolutely no speculation about the reason behind the use of the binary system in computers. The binary system is used in computers because 1 and 0 represent the two voltage levels of on and off. Whereas in base 10 there are 10 distinct symbols, 0, 1, 2, ..., 9, in base 2 there are only two, 0 and 1, with which to generate numbers. Base 10 contains digits 0 through 9; binary contains digits 0 and 1 only. These two binary digits, 0 and 1, are commonly referred to as *bits.*

Converting from decimal to binary

One method of converting from decimal to binary is to divide the decimal number by 2 repeatedly, keeping track of the remainders. This process continues until the quotient becomes zero. The remainders are then written in reverse order to obtain the binary number. This is demonstrated in Example 0-1.

Example 0-1

Convert 25_{10} to binary.

Solution:

```
           Quotient  Remainder
25/2  =    12        1      LSB (least significant bit)
12/2  =    6         0
6/2   =    3         0
3/2   =    1         1
1/2   =    0         1      MSB (most significant bit)
```

Therefore, $25_{10} = 11001_2$.

Converting from binary to decimal

To convert from binary to decimal, it is important to understand the concept of weight associated with each digit position. First, as an analogy, recall the weight of numbers in the base 10 system, as shown in the diagram. By the same token, each digit position in a number in base 2 has a weight associated with it:

740683_{10} =

3×10^0	=	3
8×10^1	=	80
6×10^2	=	600
0×10^3	=	0000
4×10^4	=	40000
7×10^5	=	700000
		740683

110101_2 =				**Decimal**	**Binary**
1×2^0 =		1×1	=	1	1
0×2^1 =		0×2	=	0	00
1×2^2 =		1×4	=	4	100
0×2^3 =		0×8	=	0	0000
1×2^4 =		1×16	=	16	10000
1×2^5 =		1×32	=	32	100000
				53	110101

Knowing the weight of each bit in a binary number makes it simplc to add them together to get its decimal equivalent, as shown in Example 0-2.

Example 0-2

Convert 11001_2 to decimal.

Solution:

Weight:	16	8	4	2	1
Digits:	1	1	0	0	1
Sum:	16 +	8 +	0 +	0 +	1 = 25_{10}

Knowing the weight associated with each binary bit position allows one to convert a decimal number to binary directly instead of going through the process of repeated division. This is shown in Example 0-3.

Example 0-3

Use the concept of weight to convert 39_{10} to binary.

Solution:

Weight:	32	16	8	4	2	1
	1	0	0	1	1	1
	32 +	0 +	0 +	4 +	2 +	1 = 39

Therefore, $39_{10} = 100111_2$.