

# 《Intel微处理器》

## 图书基本信息

书名：《Intel微处理器》

13位ISBN编号：9787111304852

10位ISBN编号：7111304853

出版时间：2010年6月

出版社：机械工业出版社

作者：（美）Barry B Brey

页数：708

译者：金惠华,艾明晶,尚利宏

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

## 前言

这本讲述Intel系列微处理器程序设计和接口技术的参考书已经是第8版了。随着技术的发展和进步，作者多次改编增补删减，与时俱进，吐故纳新，在选材、知识点配置和编程环境选择方面充分体现了先进性和实用性。与前几版相比，第8版主要增加了讲解在DOS和Windows环境下如何使用可自由下载的Visual C++ Express编写C/C++与汇编语言的接口，说明了在微处理器和微处理器接口方面出现的新成果。本书的特点是：取材涵盖微机原理、汇编语言和接口通信技术的教学要求和知识点，各部分有机整合，适合国内教学要求。以Intel系列微处理器为背景，以DOS、Windows和Visual C/C++为编程环境，通过示例为读者深入揭示了微机工作原理和最新技术进步。许多示例都可以作为读者开发类似应用的样板或原型，指引读者开发新的应用。每章开头提示本章学习目的，末尾概要总结知识要点，最后有大量习题检验学习成果。编排符合学习规律，适合读者自学。这是一本非常实用的教材，有助于读者彻底掌握Intel系列微处理器程序设计和接口技术，灵活自如地使用微机的各种资源，解决学习和开发工作中的实际问题。本书由金惠华译第16章，艾明晶译第915章，尚利宏译第1619章，高洁译第7章，郝广奇译第8章，李雅倩译附录，崔代锐、尚利荣、邓媛、刘云峰、徐其志参与了部分章节初译、示例习题核对及文稿录入。全书由金惠华统稿审校。译稿对原书中的笔误和疏漏进行了更正。由于译审者水平有限，译文中难免有不妥之处，敬请读者批评指正。

# 《Intel微处理器》

## 内容概要

本书重点讲解Intel系列微处理器（8086/8088、80186/80188、80286、80386、80486、Pentium、Pentium Pro Processor、Pentium 、Pentium 、Pentium 4和Core2）的体系结构、程序设计和接口通信技术，并通过微型计算机原理把三者有机地整合在一起。本书以Intel系列微处理器为背景，以DOS、Windows和Visual C/C++为编程环境，通过示例为读者深入揭示了微型计算机工作原理和最新的技术进步。许多示例都可以作为开发类似应用的样板或原型，用以指引开发新的应用。本书适合作为高等院校计算机、电子通信和自动控制等专业教材，也可供工程技术人员参考。

# 《Intel微处理器》

## 作者简介

德福瑞大学（ DeVry University ）荣誉退休教授。他是美国关于微处理器和汇编语言著作的主要作者，至今著有33部教材。其个人主页为<http://members.ee.net/brey/index.html>。



## 章节摘录

插图：

## 编辑推荐

《Intel微处理器(原书第8版)》：这是一本将微型计算机原理、汇编语言程序设计和PC机接口通信技术有机整合在一起的著作，可作为高等院校计算机、通信和自动控制专业的教材，也可供工程技术人员参考。《Intel微处理器(原书第8版)》主要内容采用200多个相关编程实例(许多用Visual C++及嵌入式汇编语言编写)来阐述基本概念。开发软件，控制应用系统与微处理器接口。用嵌入汇编语言的Microsoft Visual C++程序设计环境编写微处理器程序，控制PC机。开发软件，控制键盘、显示器及其他各种计算机部件。编写算术协处理器程序、MMX程序、SSE部件程序，求解复杂方程式。解释Intel系列各种处理器的区别，明确每一型号的特性。描述微处理器实模式(DOS)和保护模式(Windows)的用途。说明存储器管理操作，控制保护模式和分页机制，分配存储器。设计存储器、I/O系统到微处理器的接口。开发驱动硬件接口和应用系统的软件。解释嵌入式环境中实时操作系统(RTOS)的工作。解释磁盘及视频系统的操作。建立小型系统与PC机的ISA总线、PCI总线、并口或串口、USB总线之间的接口。详述Pentium 4微处理器新的64位扩展(EMT-64)。覆盖最新的Pentium 4和Core2处理器的内容，包括如何基于Pentium Core2及其新的64位体系化结构编程。在DOS或者Windows环境下，如何使用Visual C++Express编写C / C++与汇编程序接口的内容。针对微处理器领域的最新进展进行了更新。





# 《Intel微处理器》

## 精彩书评

1、这本非常适合汇编程序员和需要进行硬件编程的人（PCI,总线，USB）等等。我在图书馆读了一半。总结如下优点：1）一上来就讲了保护模式和真实模式。而其他类似的书一般都是到中间才讲。2）很详细地解释了汇编语言的命令机制。尤其解释了它们如何执行；数据如何从内存到寄存器，再从寄存器到另一个寄存器；计算机中的内存是如何组织的；PUSH和POP指令如何工作的等等。在解释这些时还配了好多的图例。3）内容覆盖了8086至奔腾4硬件结构。4）每章节结束还配有练习，从而强化读者对知识的掌握和理解。不足：某几个章节的顺序安排不是特别合理，要读完后面才能回头理解前面的。不过总的来说这还是一本经典的书，都出到第8版了。

## 章节试读

### 1、《Intel微处理器》的笔记-第359页

有两种不同的方法连接I/O和微处理器：独立编址I/O与存储器映像I/O基本输入设备是一组三态缓冲器，基本输出设备是一组数据锁存器。

“输入”是指数据从I/O设备到微处理器的过程

“输出”是指处理器到I/O设备的过程。输入电路图如图从左往右分别是I/O设备，74ALS244\*（3态8位缓冲器，一般用作总线驱动器）和微处理器数据总线（此图中为8位总线）。当SEL位低电平时，经过非逻辑输入到1和19口为高电平，再经过一次非逻辑，到1G 2G的时候为低电平，则三态模式为正常态，A端与Y端数据相连传输，相反SEL为高电平时，74ALS244为高阻态（电阻极大），相当于缓存器的电路被断开，数据不传输。这里因为1G和2G都是从SEL端电平连接过来，所以三态只有0态和3态，即0态和高阻态，没有1态。每次执行IN指令时，I/O端口地址被译码，SEL变为高电平。

\*74LS244为3态8位缓冲器，一般用作总线驱动器。74LS244没有锁存的功能。地址锁存器就是一个暂存器，它根据控制信号的状态，将总线上地址代码暂存起来。8086/8088数据和地址总线采用分时复用操作方法，即用同一总线既传输数据又传输地址。

74ALS244跟74LS244的功能是一样的。74ALS244的电源工作电压范围为4.5V~5.5V

74LS244的电源电压为5V。

输出电路图输出时，当OUT指令执行，AL，AX或EAX的数据通过数据总线（DB）传送给锁存器（74ALS374），SEL也被激活（通过非变为低电平和OC组合10），电路接通，如果Q端哪个数据为0（低电平）时，LED灯就变亮（VCC和低电平，有电压差）。

### 2、《Intel微处理器》的笔记-第60页

#### MOV指令

除了MOVS指令之外，其他任何指令都不许存储器到存储器（\*存储器和寄存器不同，寄存器是CPU里的存储单元，与CPU离得近，所以CPU在运算时通常都会用寄存器当中转站。存储器是在CPU外部的存储器，分为RAM,ROM。）的传送。

这里数据的传送方向是从右到左

MOV指令只是把源操作数(operand\* 一条运算包括operand和operator，所以这里是指复制操作中被复制的操作数)复制到目标操作数，并不移除源操作数。大多数MOV指令不影响标志寄存器

通常把在立即寻址方式指令中给出的数称为立即数寻址

所以这里不同寻址方式是针对不同的操作，例如

1.把一个变量复制到另一个变量中：寄存器寻址；

\*长度不同的寄存器不能互相转移操作数，目标源操作数里原来存储的数据会被清除替换为指令后的数

\* MOV CS,AX 错误，因为CS是段寄存器，指向存放程序的内存段，由系统指定，如果IP（指令指针，用来存放下一条待执行的指令在该段的偏移量，把它们合在一起可在该内存段内取到下次要执行的指令。）未变，则程序无法正常执行。

2.直接把一个常数（立即数）放到一个寄存器/存储器里：立即寻址；

\*16进制可用0F2H或者h'0F2，或者'A'，或100（十进制）

3.用于AL，AX，EAX和存储单元的数据转移：直接寻址

\* [1234H]表示绝对地址，汇编程序是不人土的，会被解释为MOV AX,DS:[1234H]，即把DS：1234H内

## 《Intel微处理器》

存空间里的信息传进AX，如DS是1000H的数据段存储单元（区别于代码段，堆栈段），则在实模式下转换成（ $1000H * 10H + 1234H = 11234H$ ）。也是三字节指令

位移量寻址：除了指令是4字节而不是3字节以外和直接寻址没有区别，是在80386到奔腾的CPU中

4.计算出某个寄存器的地址，再把变量放到那个地址中：寄存器间接寻址

.....

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)