

<<高级汇编语言设计及应用>>

图书基本信息

书名：<<高级汇编语言设计及应用>>

13位ISBN编号：9787564119218

10位ISBN编号：7564119217

出版时间：2009-12

出版时间：东南大学出版社

作者：傅德胜，孙文静 著

页数：318

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：<http://www.tushu007.com>

<<高级汇编语言设计及应用>>

前言

汇编语言是一种面向机器的程序设计语言。

汇编语言直接映射系统硬件，使程序设计人员可以像使用自己的工具那样，随心地控制和使用计算机的基本资源，设计各种期望的软件。

同时，汇编语言占用的内存小、执行速度快、效率高，因而是一种强有力的计算机语言。

在撰写本书的过程中，笔者仔细研究了国家教育部颁发的计算机科学与技术专业的教学大纲要求，深入分析了当前计算机技术的发展，结合自己30多年来讲授该课程的体会，以及笔者1999年3月出版的《宏汇编语言程序设计及应用》、2002年修订出版的《80X86汇编语言程序设计及应用》两书读者的中肯建议，本着由浅入深、循序渐进、考虑专业、兼顾普及、注重实用的精神，反复酝酿和推敲。

全书由上、下篇组成。

上篇为理论篇，其中第一部分为预备环节（第1，2章），介绍计算机中用到的数制、码制和CPU的结构。

第二部分为基础环节（第3，4，5章），主要讨论汇编语言的寻址方式、指令系统、伪指令、汇编语言源程序结构和汇编语言程序调试过程，为后面汇编语言程序设计的学习作准备。

第三部分为程序设计环节（第6，7，8，9，10章），阐述汇编语言的顺序结构程序、分支结构程序、循环结构程序及子程序与宏指令的设计方法，让读者通过这方面的学习顺利掌握基本的汇编语言程序设计。

在此基础上，研究CPU与外设之间数据传送程序设计和模块化程序设计，为读者用汇编语言开发大型软件做铺垫。

下篇为应用篇，展示了汇编语言在数据处理、图像处理和接口及通信中的近30个应用示例，供读者借鉴和移植。

同时给出了上机实习指导，给出了实习目的、实习内容、实习要求、实习准备及参考程序。

本书的出版得到了东南大学出版社的大力支持与帮助，在此致以深深的谢意。

本书可作为各种不同层次和类型高等院校计算机及其相关专业的教材，亦可作为计算机研究、生产和应用开发等人员的参考书。

因水平有限，书中错误在所难免，敬请读者指正。

<<高级汇编语言设计及应用>>

内容概要

《高级汇编语言程序设计及应用》上篇全面介绍汇编语言程序设计的原理、方法和技巧，其中包括CPU结构、指令、伪指令、宏指令、汇编语言源程序结构、基本程序（顺序，分支，循环）设计、子程序设计、输入/输出程序设计和模块化程序设计等，并配有例题200余道。下篇为应用篇，给出了汇编语言在数据处理、图像处理以及接口与通信中的应用。同时，配有实习指导，给出了实习目的、实习内容、实习要求、实习准备及参考程序。全书具有明显的特色和较强的实用性。

《高级汇编语言程序设计及应用》取材力求跟踪计算机新技术的发展，具有重点突出、承上启下、由浅入深、简明易懂、示例丰富、方便自学等特点。

《高级汇编语言程序设计及应用》可作为各种不同层次和类型高等院校计算机及相关专业的教材，亦可作为从事计算机研究、生产和开发等有关人员的培训教材和参考书。

<<高级汇编语言设计及应用>>

书籍目录

上篇 理论篇1 计算机中数的表示1.1 计算机数据的表示1.1.1 数制、基数与“位权”1.1.2 二进制数1.1.3 十六进制数1.1.4 不同数制之间的转换1.2 原码、反码与补码1.2.1 无符号数与有符号数1.2.2 原码1.2.3 反码1.2.4 补码1.3 数据存取方式1.3.1 字节数1.3.2 字数据1.3.3 双字数据1.4 ASCII码1.5 BCD码1.5.1 压缩BCD码1.5.2 非压缩BCD码1.6 扩展的键盘代码习题12 Intel微处理器2.1 8086 / 8088微处理器2.1.1 8086 / 8088微处理器的结构2.1.2 8086 / 8088内存存储器的组织2.1.3 段概念的引入2.1.4 内存存储器的地址2.1.5 堆栈2.2 80386微处理器2.2.1 80386微处理器的主要性能2.2.2 80386CPU的结构2.2.3 80386寄存器及其功能2.3 80486微处理器2.3.1 80486的特点2.3.2 80486CPU的结构2.3.3 80486寄存器组2.4 80586 (Pentium) 微处理器2.5 80x86的工作方式习题23 寻址方式与指令系统3.1 8086 / 8088的寻址方式3.1.1 立即寻址3.1.2 直接寻址3.1.3 寄存器寻址3.1.4 寄存器间接寻址3.1.5 基址寻址3.1.6 变址寻址3.1.7 基址变址寻址3.1.8 相对基址变址寻址3.1.9 段默认与段跨越3.2 80x86扩展寻址方式3.3 汇编指令语句格式3.3.1 标号3.3.2 指令助记符3.3.3 目的操作数3.3.4 源操作数3.3.5 注释3.4 8086 / 8088指令系统3.4.1 数据传送指令3.4.2 算术运算指令3.4.3 逻辑运算与移位指令3.4.4 串操作指令3.4.5 控制转移指令3.4.6 处理器控制指令3.4.7 十进制运算调整指令3.5 实方式32位指令3.5.1 32位数据传送指令3.5.2 32位算术运算指令3.5.3 32位逻辑运算与移位指令3.5.4 32位串操作指令3.5.5 32位控制转移指令3.5.6 32位位操作指令3.5.7 32位保护方式指令3.5.8 32位交换与比较指令3.6 Pentium微处理器新增指令习题34 操作数运算符与常用伪指令4.1 操作数运算符4.1.1 算术运算符 (单目+, 单目-, +, -, *, /, MOD) 4.1.2 逻辑运算符 (NOT, AND, OR, XOR) 4.1.3 关系运算符 (EQ, NE, LT, LE, GT, GE) 4.1.4 数值回送运算符 (sEG, OFFSET, TYPE, LENGrrH, SIZE) 4.1.5 属性运算符 (PTR, SHORT, THIS, HIGH, LOW) 4.1.6 移位运算符 (SHL, SHR) 4.1.7 运算符优先级4.2 常用伪指令4.2.1 数据定义伪指令 (DB, DW, DD, DF, DQ, DT, ?, DuP) 4.2.2 符号定义伪指令 (EQU, =, LABEL) 4.2.3 段定义伪指令 (sEGMENT, ENDS, ASSUME) 4.2.4 过程定义伪指令 (PROC, ENDP) 4.2.5 汇编控制伪指令 (END, ORG, NAME) 4.2.6 列表伪指令 (TITLE, PAGE, SUBTTL, %OUT) 4.3 80X86扩展伪指令4.3.1 方式选择伪指令4.3.2 段定义伪指令习题45 汇编语言源程序结构与调试运行5.1 8086 / 8088汇编源程序结构5.1.1 EXE文件的汇编格式5.1.2 COM文件的汇编格式5.2 80X86汇编源程序结构5.3 汇编语言程序调试过程5.3.1 基本过程5.3.2 Windows环境下的上机步骤5.3.3 汇编列表文件、交叉引用文件、连接列表文件说明5.4 COM文件的建立习题56 程序的基本结构与汇编语言程序设计6.1 程序设计的基本过程6.1.1 准备工作6.1.2 程序设计6.1.3 程序评价6.2 程序的基本结构6.3 顺序结构程序设计6.4 分支结构程序设计6.4.1 利用比较和条件转移指令实现程序分支6.4.2 利用跳转表实现多路分支6.5 循环结构程序设计6.5.1 循环程序控制方法6.5.2 单重循环程序设计6.5.3 多重循环程序设计习题67 系统功能调用7.1 BIOS和DOS中断7.2 调用BIOS和DOS中断程序的基本方法7.3 BIOS中断调用7.3.1 文本方式属性7.3.2 INTIOH主要功能调用7.4 DOS中断功能调用7.4.1 DOS中断基本功能7.4.2 DOS系统功能 (INT21H) 调用示例习题78 子程序与宏指令8.1 子程序设计中的基本要素8.2 主程序与子程序的连接8.3 子程序调用中的数据保护与恢复8.4 主程序与子程序之间的参数传递8.4.1 寄存器传递8.4.2 存储单元传递8.4.3 堆栈传递8.5 子程序嵌套与递归8.6 宏指令8.6.1 宏定义8.6.2 宏调用与宏扩展8.7 宏定义中的伪指令8.8 宏嵌套8.9 宏库的建立与使用习题89 输入 / 输出控制方式及程序设计9.1 I / O端口的编址方式..... 下篇 应用篇11 汇编语言在数据处理中的应用12 汇编语言在图像处理中的应用13 汇编语言在接口与通信中的应用14 实习指导附录A 8086 / 80X86指令集附录B MASM5.0出错信息附录C 系统中断参考文献

<<高级汇编语言设计及应用>>

章节摘录

2) 内部寄存器阵列 8086 / 8088内部寄存器阵列共有4组：通用数据寄存器（简称数据寄存器）、指针及变址寄存器、段寄存器和控制寄存器。

(1) 数据寄存器数据寄存器包括4个16位的寄存器（Ax、BX、CX、DX）或8个8位寄存器（AH、AL、BH、BL、CH、cL、DH、DL）。

这些寄存器都可以用来暂时存放操作数、运算结果及其他信息，但同时又具有某些专用用途。

Ax (Accumulator) 累加器。

是算术运算中的主要寄存器，在乘除运算中用来指定被乘数和被除数，也是乘、除运算后积和商的默认存储单元。

另外I/O指令均使用该寄存器与I/O设备传送信息。

BX (Base Register) 基址寄存器。

指令寻址时存入偏移量或偏移量构成成分。

CX (Count Register) 计数寄存器。

在循环指令操作或串处理指令中隐含计数。

DX (Date Register) 数据寄存器。

在双字长运算时，与Ax组合构成32位操作数，DX为高16位。

在某些I/O指令中，DX被用来存放端口地址。

(2) 指针及变址寄存器这一组寄存器共有4个：SP, BP, SI, DI。

它们都是16位寄存器，可以用来存放16位的操作数或中间结果，但更经常的用途是存放偏移量或位移量。

SP (Stack Pointer) 堆栈指针寄存器。

始终指示栈顶位置，与ss寄存器一起组成栈顶数据的物理地址。

BP (Base Pointer) 基址指针寄存器。

系统默认其指向堆栈中某一单元，即提供栈中该单元的偏移量。

加段前缀后，BP可作为非堆栈段的地址指针。

SI (Source Index) 源变址寄存器。

与DS联用，指示数据段中某操作数的偏移量。

在作串处理时，SI指示源操作数地址，并有自动增量（正反）或自动减量（反向）功能。

变址寻址时，SI与某一位移量共同构成操作数的偏移量。

DI (Destination Index) 目的变址寄存器。

与DS联用，指示数据段中某操作数的偏移量，或与某一位移量共同构成操作数的偏移量。

串处理操作时，DI指示附加段中的目的地址，并有自动增量（正向）或减量（反向）功能。

(3) 段寄存器 8086 / 8088微处理器将1MB的内存空间分成若干逻辑段（详见2.1.2小节），程序设计时，用户最多可同时使用其中的四个段。

这四个段分别为：代码段 (Code Segment) 存放当前程序的指令代码。

数据段 (Data Segment) 存放程序所涉及的源数据或结果。

堆栈段 (stack Segment) 以“先入后出”为原则的数据区。

附加段 (Extra Segment) 辅助数据区，存放串或其他数据。

在实地址方式下，代码段寄存器CS、数据段寄存器DS、堆栈段寄存器SS、附加段寄存器Es分别保存当前四个段的段基址（20位的段的起始地址的高16位）。

(4) 控制寄存器控制寄存器含指令指针寄存器IP (Instruction Pointer) 和标志寄存器FR (Flags Register)。

IP寄存器始终指向当前将要执行指令在代码段中存放的偏移量。

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:<http://www.tushu007.com>