

## <<汇编语言程序设计>>

### 图书基本信息

书名：<<汇编语言程序设计>>

13位ISBN编号：9787561123515

10位ISBN编号：7561123515

出版时间：2003-6

出版时间：大连理工大学出版社

作者：雷印胜，秦然，贾萍 编著

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>

## <<汇编语言程序设计>>

### 内容概要

汇编语言是从事计算机系统开发必备的程序开发技术之一，它属于低级语言，能够直接操作计算机硬件，是其他语言不能取代的。

本书内容由基本篇、提高篇和自测篇三部分组成，第一部分讲述了“其工作原理、内部结构、寻址方式和指令系统，并用较大的篇幅介绍了简单程序和复杂程序设计实例。

第二部分为提高篇，以80386微处理器为讲述对象，详细介绍了它的基本工作原理、新增指令集和3种工作模式，并用多个应用例子介绍了基于80386CPU的编程技巧。

第三部分为检验学习效果，本书给出了两套自测试卷及其参考答案。

## &lt;&lt;汇编语言程序设计&gt;&gt;

## 书籍目录

第1章 微型计算机系统概述 1.1 微型计算机系统简介 1.1.1 微型计算机发展史 1.1.2 微型计算机应用领域 1.2 微型计算机系统基本组成 1.2.1 微型计算机软硬件概念 1.2.2 微型计算机结构 1.2.3 CISC机和RISC机 1.3 习题与综合练习第2章 微型计算机系统结构 2.1 8086/8088微处理器 2.1.1 CPU结构 2.1.2 寄存器结构 2.2 工作模式 2.2.1 最小工作模式 2.2.2 最大工作模式 2.3 工作过程 2.4 一个完整的源程序 2.5 习题与综合练习第3章 8086/8088指令系统 3.1 寻址方式 3.1.1 操作数类型 3.1.2 寻址方式 3.2 指令系统 3.2.1 数据传送指令 3.2.2 算术运算指令 3.2.3 逻辑运算指令 3.2.4 移位指令 3.2.5 转移指令 3.2.6 字符串操作指令 3.2.7 处理器控制指令 3.2.8 输入输出指令 3.2.9 中断指令 3.3 高级汇编语言技术 3.3.1 汇编语言语句类型 3.3.2 汇编语言伪指令 3.3.3 汇编语言程序设计 3.3.4 结构和记录 3.3.5 条件汇编与宏操作伪指令 3.4 DOS系统功能调用及程序设计 3.4.1 概述 3.4.2 DOS功能调用分组 3.4.3 常用的DOS INT 21H功能调用 3.4.4 磁盘文件管理 3.5 BIOS功能调用 3.5.1 概述 3.5.2 常用BIOS功能调用 3.5.3 图形显示程序设计 3.6 习题与综合练习第4章 基本程序设计 4.1 顺序程序设计 4.1.1 存储单元内容移位 4.1.2 乘法运算与乘10运算 4.1.3 屏蔽与组合 4.1.4 字节分离 4.1.5 单字节压缩BCD数加法运算 4.1.6 两字节的二进制数加法运算 4.1.7 取数的反码和补码 4.1.8 F方表 4.2 分支程序设计 4.2.1 单重分支结构程序 4.2.2 多重分支结构程序 4.3 循环程序设计 4.3.1 循环程序的结构 4.3.2 单重循环程序 4.3.3 多重循环程序 4.3.4 循环次数未知的循环程序 4.3.5 “位”控制循环程序 4.4 子程序设计 4.4.1 子程序与主程序 4.4.2 子程序段内调用和返回 4.4.3 子程序段间调用和返回 4.4.4 调用程序和子程序间的参数传递 4.4.5 寄存器内容的保护 4.4.6 子程序的嵌套使用 4.4.7 关于递归子程序、可重入子程序 4.5 具有模块结构的程序设计 4.5.1 概述 4.5.2 模块的组合方式 4.5.3 模块间的通信 4.5.4 模块化程序设计的注意点 4.5.5 模块化程序设计举例 4.6 习题与综合练习第5章 复杂程序设计 5.1 定点数算术运算程序 5.1.1 定点数运算的概念 5.1.2 定点数加法运算 5.1.3 定点数减法运算 5.1.4 定点数乘法运算 5.1.5 定点数除法运算 5.2 浮点数算术运算程序 5.2.1 浮点数概念 5.2.2 浮点数的规格化 5.2.3 浮点数加减运算 5.2.4 浮点数乘除运算 5.3 代码转换 5.3.1 二进制数与ASC 码间的相互转换 5.3.2 二进制数与BCD码间的相互转换 5.3.3 二进制数到七段显示码的转换 5.4 字符数据处理 5.4.1 字符串比较 5.4.2 字符串检索 5.4.3 字符删除与插入 5.4.4 字符串统计 5.5 表处理 5.5.1 表的查询 5.5.2 表的插入与删除 5.6 检索 5.6.1 顺序检索 5.6.2 折半检索 5.6.3 散列值检索 5.7 排序 5.7.1 交换排序 5.7.2 选择排序 5.7.3 插入排序 5.8 习题与综合练习第6章 汇编语言的一些特殊命令用法 6.1.EXE文件和.COM文件 6.2 程序段前缀 6.3 汇编程序(ASM, MASM) 6.3.1 汇编程序的类别 6.3.2 汇编过程 6.3.3 运行环境 6.3.4 操作过程 6.3.5 汇编操作举例 6.4 连接程序(LINK) 6.4.1 连接程序的作用 6.4.2 连接过程 6.4.3 LINK的使用与操作 6.5 调试程序(DEBUG) 6.5.1 DEBUG的功能及其启动 6.5.2 DEBUG各命令的用法 6.5.3 应用举例 6.6 符号调试程序SYMDEB简介 6.7 上机操作辅助程序介绍 6.7.1 显示(或打印)单个字符 6.7.2 显示字符串 6.7.3 键入单个字符 6.7.4 键入字符串 6.7.5 程序正常结束 6.8 汇编语言与高级语言的连接 6.8.1 TURBO C调用汇编子程序 6.8.2 TURBO C行间嵌入汇编 6.9 习题与综合练习第7章 80386程序设计基础 7.1 实地址方式 7.2 80386寄存器 7.2.1 通用寄存器 7.2.2 段寄存器和段描述符寄存器 7.2.3 指令指针和标志寄存器 7.3 80386存储器寻址 7.3.1 存储器寻址基本概念 7.3.2 灵活的存储器寻址方式 7.3.3 支持各种数据结构 7.4 80386指令集 7.4.1 数据传送指令 7.4.2 算术运算指令 7.4.3 逻辑运算和移位指令 7.4.4 控制转移指令 7.4.5 串操作指令 7.4.6 高级语言支持指令 7.4.7 条件字节设置指令 7.4.8 位操作指令 7.4.9 处理器控制指令 7.5 实地址方式下的程序设计 7.5.1 说明 7.5.2 实例 7.6 操作系统类指令 7.6.1 实地址方式和任何特权级下可执行的指令 7.6.2 实地址方式及特权级0下可执行的指令 7.6.3 只能在保护虚地址方式下执行的指令 7.6.4 显示关键寄存器内容的实例 7.6.5 被包含文件386SCD.ASM 7.6.6 特权指令 7.7 习题与综合练习-第8章 保护虚地址方式下的80386及其编程 8.1 保护虚地址方式简介 8.1.1 存储管理机制 8.1.2 保护机制 8.2 分段管理机制 8.2.1 段定义和虚拟地址到线性地址转换 8.2.2 存储段描述符 8.2.3 全局和局部描述符表 8.2.4 段选择子 8.2.5 段描述符高速缓冲寄存器 8.3 80386控制寄存器和系统地址寄存器 8.3.1 控制寄存器 8.3.2 系统地址寄存器 8.4 实地址方式与保护虚地址方式切换实例 8.4.1 演示实地址和保护虚地址切换的实例 8.4.2 演示32位代码段和16位代码段切换的实例 8.5 任务状态段和控制门 8.5.1 系统段描述符 8.5.2 门描述符 8.5.3 任

<<汇编语言程序设计>>

务状态段 8.6 控制转移 8.6.1 任务内无特权级变换的转移 8.6.2 演示任务内无特权级变换转移的实例  
8.7 分页管理机制 8.7.1 存储器分页管理机制 8.7.2 线性地址到物理地址的转换 8.7.3 页级保护和虚拟  
存储器支持 8.7.4 页异常 8.7.5 演示分页机制的实例 8.8 输入/输出保护 8.8.1 输入/输出保护 8.8.2 重  
要标志保护 8.8.3 演示输入/输出保护的实例 8.9 80386的中断和异常 8.9.1 80386的中断和异常 8.9.2 异  
常类型 8.9.3 中断和异常的转移方法 8.9.4 演示中断处理的实例 8.10 虚拟8086方式 8.10.1 V86方式  
8.10.2 进入和离开V86方式 8.10.3 演示进入和离开V86方式的实例 8.10.4 V86方式下的敏感指令 8.11 习  
题与综合练习自测试卷(一)自测试卷(二)

## &lt;&lt;汇编语言程序设计&gt;&gt;

## 章节摘录

插图：8086 / 8088也可以按最大工作模式来配置系统。

当MN / MX线接地，则系统就工作于最大工作模式了。

这里我们先简要说明什么是最大工作模式，它和最小工作模式有何区别。

在上面讨论的8086 / 8088最小方式系统中，8086 / 8088CPU的引脚直接提供所有必须的总线控制信号，这种方式适合于单处理器组成的小系统。

在最小工作方式中，作为单处理器的8086 / 8088CPU通常控制着系统总线，但也允许系统中的其他主控设备——DMA控制器占用系统总线。

DMA控制器通过占用系统总线可实现外部设备和存储器之间直接数据传送。

DMA控制器通过向8086 / 8088的HOLD引脚发送一个高电平信号向CPU提出占用系统总线的请求信号，通常在现行总线周期完成后，8086 / 8088CPU作出响应，使HLDA引脚变成高电平，通知DMA控制器可以使用系统总线。

DMA控制器接收到HLDA引脚的高电平信号后，掌握系统控制权，进行外部设备与存储器之间的直接数据传送。

当DMA控制器完成传送任务时，撤销发向HOLD引脚的总线请求信号，CPU重新获得对系统的控制权。

需着重指出的是，DMA控制器虽然通过挪用总线周期实现外部设备与存储器之间的直接数据传送，提高了整个系统的能力，但DMA控制器却不能执行命令，其能力是相当有限的。

假如系统中有两个或多个同时执行指令的处理器，这样的系统就称为多处理器系统。

增加的处理器可以是8086 / 8088处理器，也可以是数字数据处理器8087或I / O处理器8089。

在设计多处理器系统时，除了解决对存储器和I / O设备的控制、中断管理、DMA传送时总线控制权外，还必须解决多处理器对系统总线的争用问题和处理器之间的通信问题。

因为多个处理器通过公共系统总线共享存储器和I / O设备，所以必须增加相应的逻辑电路，以确保每次只有一个处理器占用系统总线。

为了使一个处理器能够把任务分配给另一个处理器或者从另一个处理器取回执行结果，必须提供一种明确的方法来解决两个处理器之间的通信。

多处理器系统可以有效地提高整个系统的性能。

8086 / 8088的最大工作方式就是专门为实现多处理器系统而设计的。

IBMPC系列机系统中的微处理器工作于最大工作方式，系统中配置了一个作为协处理器的数字数据处理器8087。

以提高系统数据处理的能力。

## <<汇编语言程序设计>>

### 编辑推荐

《汇编语言程序设计(第2版)(计算机类)》：新世纪高职高专软件专业系列规划教材

<<汇编语言程序设计>>

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>