

## <<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：9787030139979

10位ISBN编号：7030139976

出版时间：2004-8

出版时间：科学出版社

作者：龚荣武 编

页数：228

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>

## &lt;&lt;汇编语言程序设计&gt;&gt;

## 前言

微型计算机的使用已非常普及，在国内使用80X86处理器的微机占大多数，在这样的微机上进行各种应用软件开发是高职高专计算机专业的学生应该掌握的。

汇编语言是一种低级语言，它与硬件关系最为密切，由于汇编语言具有目标代码短、执行速度快、能够充分利用硬件资源及能够进行精确控制的优点，这些优点是其他高级语言远远不能比拟的。

正是这一优点，使得在某些编程中应该、甚至必须使用汇编语言。

要了解微型计算机是如何工作，各种硬件在微机系统中起什么作用，仅仅掌握高级语言的程序设计是无法达到要求的，只有掌握了汇编语言程序设计才能深入了解微型计算机的工作原理。

为此，高职高专计算机专业学生学习汇编语言，掌握汇编语言基本程序的设计方法，是非常有意义的。

本书以Intel 8086 / 8088系列微机为对象，介绍汇编语言程序设计。

在介绍Intel8086 / 8088微处理器的寻址方式和指令系统的基础上，详细讲解了编写汇编语言程序的基本方法和技巧。

掌握这些方法和技巧，不仅有助于利用Intel 8086 / 8088及Pentium系列微机的汇编语言程序进行程序设计，并为Intel 80X86及Pentium系列微机的汇编语言程序设计奠定了基础。

本书内容的安排力求循序渐近、重点突出、难点分散，通过理论课的课堂讲授和上机实验，学生能对以下问题进行汇编的编程：非压缩BCD码数的加、减、乘、除；压缩BC[ ]数的加、减、乘、除；二进制数（在一个字的长度范围内）的加、减、乘、除；将十进制数转换成二进制数，再变成补码；将二进制数转换成十进制数；各种数、字符的输出等。

在内容的组织和阐述中，我们尽量将理论知识和具体的实际应用联系起来讲述，对相应的章节还配有大量实例。

在每一章中，我们将本章的内容、要点等通过“知识点”、“难点”、“要求”等对读者做出提示，使读者在学习的过程中对每一章的具体知识点、重点难点以及要求掌握和了解的内容有清楚的认识，方便读者对全书内容和组织结构有系统化的了解；书后我们还以附录的方式提供了ASCII码表、80X86指令表、Debug命令表等相应的学习汇编语言必备资料，方便读者自学。

本书由龚荣武任主编，刘萍、蔡英任副主编，杨大千编写了第1、2章，韦坤编写了第3、4、5章，刘萍编写了第6、7章，刘萍、杨大千合编第8章，蔡英编写了第9、10章，杨大千、韦坤、刘萍、尹晓燕整理了附录，全书由龚荣武教授统稿。

在本书的编写过程中，得到科学出版社的热情支持和帮助，在此表示衷心的感谢。

由于编者水平有限，书中难免存在一些缺点和错误，恳请广大读者和同行专家批评指正。

## <<汇编语言程序设计>>

### 内容概要

本书是高职高专计算机规划教材，主要内容包括：微型计算机基础知识、IBM-PC微型计算机系统概述、8086 / 8088指令系统、80X86 / Pentium微处理器指令系统、汇编语言程序设计基本方法、三大结构程序设计（顺序、选择、循环）、子程序设计、汇编语言程序设计示例、模块化程序设计的基本技术、上机实验内容及汇编语言程序调试方法等。

全书共分10章。

内容丰富，系统性强，深入浅出。

在组织和阐述内容时。

我们尽量将理论知识和具体的实际应用联系起来讲述，对相应的章节还配有大量实例。

在每一章中，我们将本章的内容、要点等通过“知识点”、“难点”、“要求”等对读者做出提示，使读者在学习的过程中对每一章的具体知识点、重点难点以及要求掌握和了解的内容有一个清楚的认识，方便读者对全书内容和组织结构有一个系统化的了解；书后以附录的方式提供了ASCII码表、80X86指令表、Debug命令表等相应的学习汇编语言必备资料，以方便读者自学。

本书可作为高职高专计算机、通信、网络等专业的教材。

# <<汇编语言程序设计>>

## 书籍目录

### 第1章 微型计算机基础知识

#### 1.1 微型计算机系统简述

##### 1.1.1 微处理器

##### 1.1.2 微型计算机系统

#### 1.2 微型计算机数和字符的表示

##### 1.2.1 计算机中数的表示

##### 1.2.2 常用数制及其算术运算

##### 1.2.3 数制的转换

##### 1.2.4 数的表示

##### 1.2.5 加减法运算的溢出条件

##### 1.2.6 字符的表示

#### 1.3 汇编语言程序设计概述

##### 1.3.1 汇编语言

##### 1.3.2 汇编语言程序设计

#### 习题

### 第2章 8086 / 8088微型计算机系统概述

#### 2.1 8086 / 8088的功能结构

##### 2.1.1 执行单元EU

##### 2.1.2 总线接口单元BIU

#### 2.2 8086 / 8088的寄存器结构

##### 2.2.1 通用寄存器

##### 2.2.2 段寄存器

##### 2.2.3 状态寄存器

##### 2.2.4 指令指针寄存器

#### 2.3 存储器的组织

##### 2.3.1 存储器的标准结构

##### 2.3.2 存储器的分段

##### 2.3.3 实际地址和逻辑地址

##### 2.3.4 堆栈

#### 2.4 Intel80X86系统高档微处理器简介

##### 2.4.1 80286微处理器

##### 2.4.2 80386微处理器

##### 2.4.3 80486微处理器

##### 2.4.4 Pentium微处理器

#### 习题

### 第3章 8086 / 8088指令系统

#### 3.1 指令的基本格式

##### 3.1.1 指令的构成

##### 3.1.2 8086指令的通用格式

#### 3.2 8086 / 8088的寻址方式

##### 3.2.1 固定寻址

##### 3.2.2 立即寻址

##### 3.2.3 寄存器寻址

##### 3.2.4 存储器寻址

##### 3.2.5 I / O端口寻址

## &lt;&lt;汇编语言程序设计&gt;&gt;

## 3.3 指令的执行时间

## 3.4 8086 / 8088指令系统

## 3.4.1 数据传送类指令及应用

## 3.4.2 算术运算类指令及应用

## 3.4.3 逻辑运算与移位类指令及应用

## 3.4.4 串操作类指令及应用

## 3.4.5 控制转移类指令及应用

## 3.4.6 处理器控制类指令及应用

## 习题

## 第4章 汇编语言程序

## 4.1 汇编语言与汇编程序

## 4.1.1 汇编语言

## 4.1.2 汇编程序

## 4.2 汇编语言程序的格式和组成元素

## 4.2.1 语句类型(指令、伪指令、宏指令三类)

## 4.2.2 汇编语言语句的四个域

## 4.2.3 汇编语言程序的段结构

## 4.3 伪指令及其应用

## 4.3.1 符号定义伪指令

## 4.3.2 数据定义及存储器分配伪指令

## 4.3.3 标号定义伪指令

## 4.3.4 段定义伪指令

## 4.3.5 过程定义伪指令

## 4.3.6 80X86指令集选择伪指令

## 4.4 汇编语言程序的上机过程

## 4.4.1 建立ASM文件

## 4.4.2 生成OBJ文件

## 4.4.3 生成EXE文件

## 4.4.4 快速生成可执行文件的方法

## 4.4.5 程序的执行和调试

## 4.4.6 TASM、TLINK及TurboDebug的使用

## 习题

## 第5章 80x86 / Pentium微处理器指令系统

## 5.1 80286增强和扩充指令

## 5.1.1 80286工作模式

## 5.1.2 有符号整数乘法指令

## 5.1.3 堆栈操作指令

## 5.1.4 移位指令

## 5.1.5 支持高级语言的指令

## 5.1.6 控制保护指令

## 5.2 80386增强和扩充指令

## 5.2.1 数据传送与扩展指令

## 5.2.2 地址传送指令

## 5.2.3 有符号乘法指令

## 5.2.4 符号扩展指令

## 5.2.5 堆栈操作指令

## 5.2.6 移位指令

## <<汇编语言程序设计>>

5.2.7 位操作指令

5.2.8 条件设置指令

5.3 80486新增指令

5.3.1 字节交换指令

5.3.2 互换并相加指令

5.3.3 比较并相加指令

5.3.4 Cache管理指令

5.4 Pentium新增指令

5.4.1 8字节比较交换指令

5.4.2 处理器特征识别指令

5.4.3 读时间标记计数器指令

5.4.4 读模型专用寄存器指令

5.4.5 写模型专用寄存器指令

习题

第6章 顺序程序设计

6.1 汇编语言程序设计的基本步骤

6.1.1 问题分析

6.1.2 确定算法, 编制流程图

6.1.3 编码与调试

6.2 顺序程序设计

6.2.1 输入 / 输出的DOS功能调用

6.2.2 十进制BCD码校正指令

6.2.3 顺序程序设计举例

习题

第7章 分支程序设计

7.1 分支程序结构

7.1.1 单边选择程序设计

7.1.2 多边选择程序设计

7.2 转移指令

7.2.1 条件转移指令及应用

7.2.2 无条件转移指令及应用

7.3 分支程序设计

7.3.1 用比较转移指令实现分支

7.3.2 转移指示表法实现分支

习题

第8章 循环程序设计

8.1 循环程序结构

8.1.1 问题的提出

8.1.2 循环程序结构

8.2 循环指令

8.2.1 重复控制指令(循环控制指令)

8.2.2 串操作指令及重复前缀

8.3 循环程序设计

8.3.1 先执行后判断结构的循环程序设计

8.3.2 先判断后执行结构的循环程序设计

8.3.3 循环程序的控制方法

8.4 多重循环

## <<汇编语言程序设计>>

### 习题

#### 第9章 子程序设计

##### 9.1 子程序与调用程序

###### 9.1.1 子程序设计方法

###### 9.1.2 子程序的调用

##### 9.2 具有子程序的汇编程序设计

###### 9.2.1 含有NEAR过程及过程调用的程序设计

###### 9.2.2 含有FAR过程及过程调用的程序设计

##### 9.3 保护现场与恢复现场

##### 9.4 过程的参数传递方法

###### 9.4.1 约定寄存器法

###### 9.4.2 约定存储单元法

###### 9.4.3 伪指令ExTRN, PUBLIC的使用

###### 9.4.4 用寄存器传送参数地址表的地址

###### 9.4.5 用堆栈传送参数

##### 9.5 过程的嵌套和递归调用

###### 9.5.1 过程的嵌套

###### 9.5.2 过程的递归调用

#### 第10章 汇编语言程序设计示例

##### 10.1 字符处理

##### 10.2 代码转换

##### 10.3 表的处理

##### 10.4 多模块的连接

###### 10.4.1 运行汇编源程序的过程

###### 10.4.2 多模块的连接

#### 附录

##### 附录A ASCII码表

##### 附录B 80X86指令表

##### 附录C TASM5.0宏汇编程序出错信息表

##### 附录D Debug命令表

##### 附录E 上机实验指导

#### 主要参考文献

## &lt;&lt;汇编语言程序设计&gt;&gt;

## 章节摘录

如同高级语言一样，除非遇到分支指令，否则机器语言程序总是依次执行。

指令寄存器（IR）用来保存它当前正在译码和执行的指令。

程序计数器（PC）寄存器用来保存下一条将要取出的指令地址。

当前正在执行的指令完成后，PC就把它保存的指令地址送到地址总线上，从存储器读出下一条指令，存储器把该指令的代码通过数据总线送到CPU。

CPU把指令接收并传到指令寄存器。

每当从内存读出一个字节后PC就自动加1，所以它总是指向下一条指令的地址。

如果在程序的执行中遇到了无条件分支指令，PC的内容就被该指令所给的目的地址所取代，从而改变了程序顺序执行的规律。

如果遇到条件分支指令，当条件满足时，PC的内容被该指令给定的目的地址所取代，否则PC的内容不变，从而实现了条件满足则转移，否则顺序执行的功能。

处理器当前的状态保存在处理器状态寄存器中。

处理器状态寄存器（PSW）包含指明这样一些事件的位：即前面的运算操作产生的结果是正、负，或全0等结果。

例如：如果跟在减法运算指令后面的是一条全0分支指令（Jz），则根据PSW寄存器中ZF（保存的减法结果是否为0的位）的值来决定是否发生分支。

如果ZF=1（表示运算结果为0）则发生分支，否则顺序执行。

发生分支后，程序指令又从新的地址开始顺序执行。



<<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>