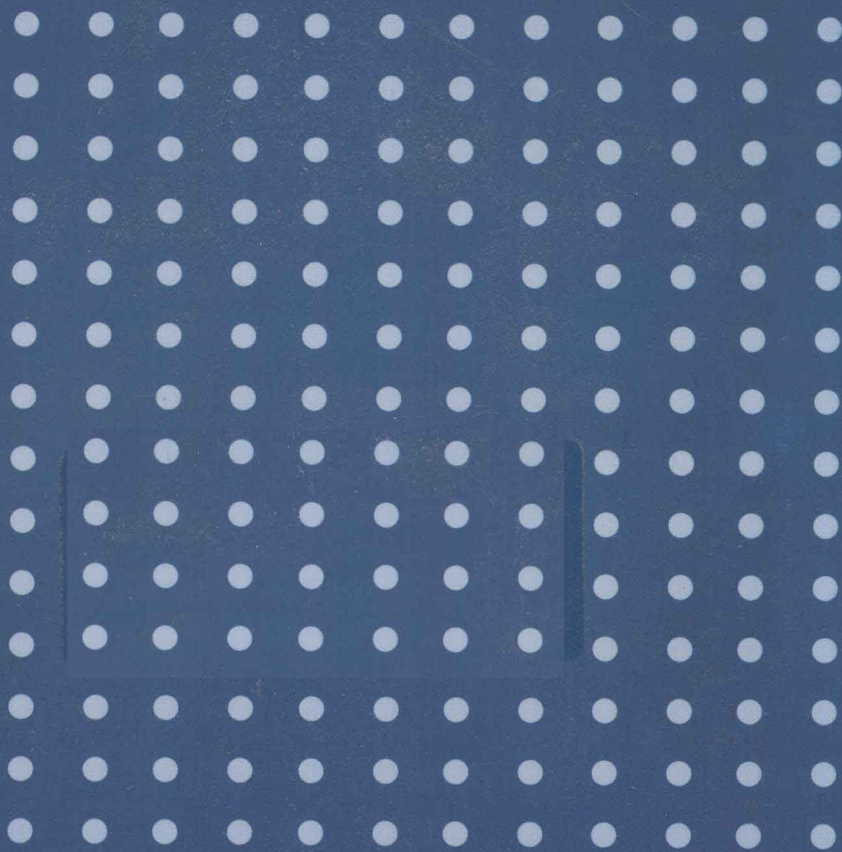


重点大学计算机专业系列教材

# 汇编语言 程序设计及上机指导

钱忠民 编著



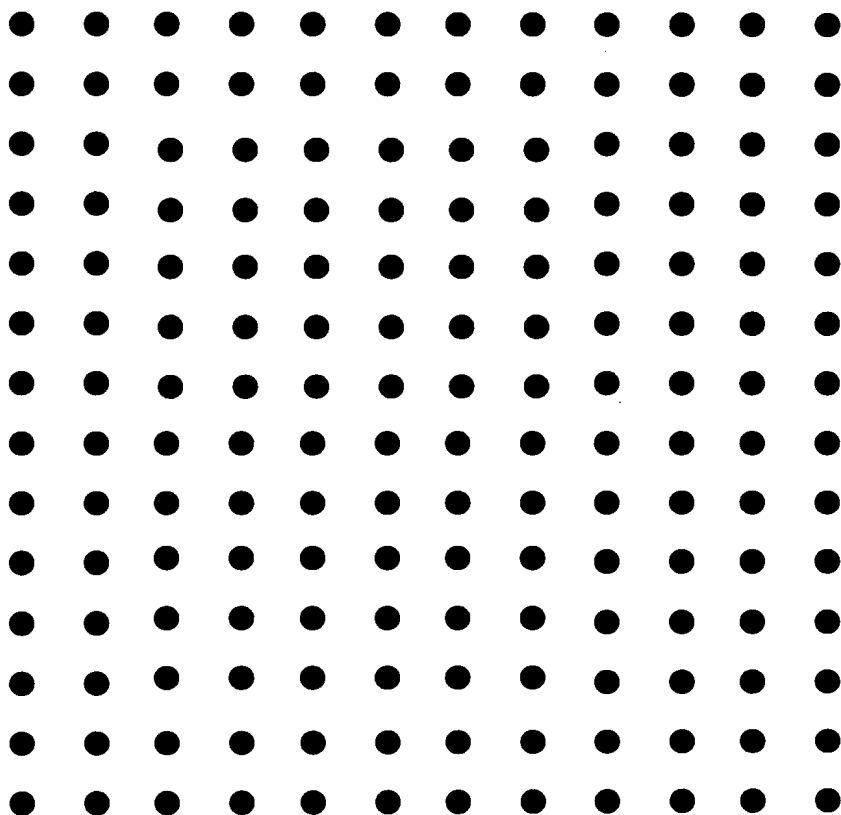
清华大学出版社



重点大学计算机专业系列教材

# 汇编语言 程序设计及上机指导

钱忠民 编著



清华大学出版社

北京

## 内 容 简 介

本书以 Intel 80x86 至 CORE 微处理器指令系统和 MASM 6. X 为主体,介绍了 16 位、32 位以及 64 位定点数指令的汇编语言程序设计方法。全书共分 8 章,前 7 章分别讲述了微处理器的基本组成结构、指令系统和程序设计方法等,并包含了目前最新的 64 位程序设计内容;第 8 章是上机指导,结合前 7 章的知识给出了 10 个实验,介绍了完整的实验步骤和方法;本书还附录了各章习题参考答案。

本书适合作为高等院校相关专业“汇编语言程序设计”课程的教材或参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

## 图书在版编目(CIP)数据

汇编语言程序设计及上机指导/钱忠民编著. —北京:清华大学出版社,2011.6  
(重点大学计算机专业系列教材)

ISBN 978-7-302-24741-8

I. ①汇… II. ①钱… III. ①汇编语言—程序设计 IV. ①TP313

中国版本图书馆 CIP 数据核字(2011)第 021429 号

责任编辑:魏江江 顾 冰

责任校对:李建庄

责任印制:王秀菊

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010 62770175

投稿与读者服务:010-62795954,jsjtc@tup.tsinghua.edu.cn

质 量 反 馈:010 62772015,zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010 62786544

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185×260 印 张:12.25 字 数:304 千字

版 次:2011 年 6 月第 1 版 印 次:2011 年 6 月第 1 次印刷

印 数:1~3000

定 价:21.00 元

---

产品编号:040373-01

## 出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有 16 个国家重点学科、20 个博士点一级学科、28 个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

2. 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略,突出重点,保证质量。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

5. 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

# 前言

“汇编语言程序设计”课程是计算机专业和自动控制专业以及机械类专业的必修课程,是我国高等院校工科类相关专业的基础课程。它不仅是机械控制、计算机接口技术及操作系统等其他课程的先行课,而且对训练学生掌握程序设计技术、加深对计算机工作原理的理解有非常重要的作用。其重点是培养学生汇编语言程序的编程能力,培养学生在实际应用中汇编语言程序的分析和设计能力,进而具备编写直接控制硬件程序的能力。

目前高校相关专业的“汇编语言程序设计”课程计划学时普遍较少,一般为 24~40 学时。再加上该课程内容较抽象,不像高级语言那样容易理解,导致学生普遍反映这门课程比较难掌握。而且现有教材的内容很多都比较陈旧,或者是内容太多,使学生无法掌握重点,已经不能适应人才培养的需要。

本书在总结教学经验的基础上,本着既适合于教师授课又方便学生学习的思想,以 80x86 至 CORE 微处理器的指令系统和 MASM 6.X 为主体,深入浅出地全面叙述了汇编语言程序设计知识。作为一本教材,它具有如下三个方面的特点。

## 1. 内容精炼

本书内容的编写参照国内高校相关专业的教学计划和教学大纲的要求,着重介绍了 80x86 至 CORE 微处理器的定点数指令,对于在较少学时中基本学不到的浮点数指令及多媒体指令没有叙述。程序设计部分主要介绍了在 MASM 高版本环境下的简化段定义结构。

## 2. 知识较新,突出重点

本书讲述的知识兼顾了 8086CPU 的基本指令集,同时对于在 32 位 CPU 及目前最新的 64 位环境下的指令及程序设计都有所涉及,重点是基本指令集,教学中可以根据实际学时和教学侧重点的不同而有选择地讲述。

## 3. 实用性强

本书特别强调上机实践,叙述了与教学内容相关的实验指导环节,并给出了具体操作步骤和完整的源程序。源程序都是在 MASM 环境下汇编通过且功能正确的,非常适合于教学过程中的上机实践参考。另外教材中还给出了丰富的课后习题,并提供了较完整的习题答案,在很大程度上也方便了自学。

本书由 7 个章节、上机指导(包括 10 个实验)以及课后习题答案组成。通过使用本教材,可以使学生掌握 80x86 至 CORE 微处理器的结构特点、寻址方式、指令系统及其汇编语言程序设计基本思想和方法,并通过上机实践熟悉上机操作和程序调试技巧,进一步掌握利用汇编语言程序设计充分发挥计算机效率的基本思想,也是计算机硬件课程体系中最重要的一环。

限于编者的学识水平,本书中难免有疏漏和不当之处,敬请读者和同行批评指正,非常感谢。

作 者

2011 年 4 月

|                        |    |
|------------------------|----|
| <b>第 1 章 汇编语言概述</b>    | 1  |
| 1.1 汇编语言的由来及其特点        | 1  |
| 1.1.1 机器语言             | 1  |
| 1.1.2 汇编语言             | 2  |
| 1.1.3 汇编语言程序           | 2  |
| 1.1.4 汇编语言的主要特点        | 2  |
| 1.1.5 汇编语言的使用领域        | 3  |
| 1.2 数据的表示和类型           | 3  |
| 1.2.1 数值数据的表示          | 3  |
| 1.2.2 非数值数据的表示         | 4  |
| 1.2.3 基本的数据类型          | 5  |
| 习题 1                   | 6  |
| <b>第 2 章 寄存器与存储器</b>   | 7  |
| 2.1 寄存器组织              | 7  |
| 2.1.1 通用寄存器            | 7  |
| 2.1.2 标志寄存器            | 9  |
| 2.2 存储器的管理模式           | 10 |
| 2.2.1 实地址方式内存管理        | 10 |
| 2.2.2 保护方式内存管理模式       | 12 |
| 习题 2                   | 12 |
| <b>第 3 章 指令格式与寻址方式</b> | 14 |
| 3.1 指令格式               | 14 |
| 3.2 操作数类型和有效地址         | 15 |
| 3.2.1 操作数类型            | 15 |
| 3.2.2 有效地址             | 15 |

|                                            |           |
|--------------------------------------------|-----------|
| 3.3 各种寻址方式 .....                           | 16        |
| 3.3.1 立即寻址 .....                           | 16        |
| 3.3.2 寄存器寻址 .....                          | 17        |
| 3.3.3 存储器寻址 .....                          | 17        |
| 3.3.4 端口寻址 .....                           | 22        |
| 习题 3 .....                                 | 24        |
| <b>第 4 章 指令系统</b> .....                    | <b>25</b> |
| 4.1 指令系统概述 .....                           | 25        |
| 4.2 数据传送类指令 .....                          | 26        |
| 4.2.1 通用数据传送指令 .....                       | 26        |
| 4.2.2 堆栈操作指令 .....                         | 29        |
| 4.2.3 地址传送指令 .....                         | 30        |
| 4.2.4 标志位传送指令 .....                        | 31        |
| 4.2.5 输入输出数据传送指令 .....                     | 32        |
| 4.3 算术运算类指令 .....                          | 33        |
| 4.3.1 加减法类指令 .....                         | 33        |
| 4.3.2 乘除法指令 .....                          | 37        |
| 4.3.3 符号扩展指令 .....                         | 39        |
| 4.3.4 BCD(十进制)算术运算指令 .....                 | 41        |
| 4.4 逻辑运算与移位指令 .....                        | 42        |
| 4.4.1 逻辑运算指令 .....                         | 43        |
| 4.4.2 移位和循环移位指令 .....                      | 44        |
| 4.5 串操作类指令 .....                           | 46        |
| 4.5.1 重复前缀 REP/REPE/REPNE/REPZ/REPNZ ..... | 47        |
| 4.5.2 基本串操作指令 .....                        | 48        |
| 4.6 控制转移类指令 .....                          | 51        |
| 4.6.1 无条件转移指令 JMP .....                    | 51        |
| 4.6.2 条件转移类指令 .....                        | 53        |
| 4.6.3 调用与返回指令 .....                        | 54        |
| 4.6.4 循环控制指令 .....                         | 57        |
| 4.6.5 中断指令 .....                           | 58        |
| 4.7 处理器控制类指令 .....                         | 59        |
| 4.7.1 标志位操作类指令 .....                       | 59        |
| 4.7.2 处理器协调类指令 .....                       | 59        |
| 4.8 其他指令 .....                             | 60        |
| 4.8.1 高级语言指令和保护模式系统控制指令 .....              | 60        |
| 4.8.2 条件设置指令 .....                         | 63        |
| 4.8.3 位操作指令 .....                          | 64        |

|                                   |            |
|-----------------------------------|------------|
| 4.8.4    特殊功能实用指令 .....           | 65         |
| 习题 4 .....                        | 66         |
| <b>第 5 章    汇编语言语法 .....</b>      | <b>70</b>  |
| 5.1    汇编语言语法概述 .....             | 70         |
| 5.1.1    汇编语言程序的基本概念 .....        | 70         |
| 5.1.2    汇编语言的基本语法 .....          | 71         |
| 5.2    汇编语言的数据和表达式 .....          | 72         |
| 5.2.1    常量 .....                 | 72         |
| 5.2.2    变量 .....                 | 73         |
| 5.2.3    标号 .....                 | 76         |
| 5.2.4    表达式和运算符 .....            | 78         |
| 5.3    宏汇编语言的伪指令 .....            | 82         |
| 5.3.1    段定义和程序说明伪指令 .....        | 83         |
| 5.3.2    重复汇编和条件汇编伪指令 .....       | 91         |
| 5.3.3    结构定义伪指令 .....            | 94         |
| 习题 5 .....                        | 95         |
| <b>第 6 章    结构化程序设计 .....</b>     | <b>98</b>  |
| 6.1    分支和循环结构程序设计 .....          | 98         |
| 6.1.1    分支程序设计 .....             | 98         |
| 6.1.2    循环程序设计 .....             | 101        |
| 6.1.3    利用中断调用实现简单输入输出 .....     | 104        |
| 6.2    子程序设计 .....                | 105        |
| 6.2.1    过程设计 .....               | 106        |
| 6.2.2    多模块编程 .....              | 108        |
| 6.3    宏结构程序设计 .....              | 109        |
| 6.4    汇编程序和汇编处理过程 .....          | 112        |
| 6.4.1    源程序的汇编 .....             | 112        |
| 6.4.2    目标程序的运行和调试 .....         | 115        |
| 习题 6 .....                        | 117        |
| <b>第 7 章    高级汇编语言程序设计 .....</b>  | <b>119</b> |
| 7.1    保护模式编程介绍 .....             | 119        |
| 7.1.1    保护模式的特征 .....            | 119        |
| 7.1.2    保护模式编程 .....             | 120        |
| 7.2    用汇编语言编写 Windows 应用程序 ..... | 123        |
| 7.2.1    Windows 与 DOS 的区别 .....  | 123        |
| 7.2.2    简单应用程序开发示例 .....         | 124        |

|                                 |            |
|---------------------------------|------------|
| 7.3 汇编语言和 C 语言的简单混合编程 .....     | 129        |
| 7.3.1 汇编指令的嵌入式编程 .....          | 129        |
| 7.3.2 多模块混合编程 .....             | 131        |
| 习题 7 .....                      | 135        |
| <b>第 8 章 汇编语言程序设计实验指导 .....</b> | <b>140</b> |
| 实验一 调试工具 DEBUG 的应用 .....        | 140        |
| 实验二 内存操作数及寻址方法 .....            | 142        |
| 实验三 数据串传送和查表程序 .....            | 143        |
| 实验四 完整段定义程序实现 .....             | 146        |
| 实验五 分支结构程序设计 .....              | 147        |
| 实验六 分支与循环混合结构程序设计 .....         | 149        |
| 实验七 多重循环及过程的应用 .....            | 151        |
| 实验八 结构的应用及宏程序设计 .....           | 153        |
| 实验九 过程调用及模块化程序设计 .....          | 156        |
| *实验十 汇编语言综合实验 .....             | 159        |
| <b>附录 A 汇编语言常见出错信息 .....</b>    | <b>163</b> |
| <b>附录 B 部分习题答案 .....</b>        | <b>168</b> |

# 汇编语言概述

## 第 1 章

**【本章内容提要】** 本章介绍汇编语言的一些基本概念,以 Intel 的 IA32 系列微处理器为例,给出一些用汇编语言编程所需要的基本知识,包括汇编语言的特点、适应领域以及进行汇编语言程序设计所需的各种数据类型等方面的内容。

**【本章学习目标】**

- 掌握汇编语言和机器语言的基本区别;
- 明确汇编语言相对于高级语言的基本特点和应用领域;
- 掌握各种数据类型的用法。

## 1.1 汇编语言的由来及其特点

### 1.1.1 机器语言

机器指令是 CPU 能直接识别并执行的指令,它的表现形式是二进制编码。机器指令通常由操作码和操作数两部分组成,操作码指出该指令所要完成的操作,即指令的功能;操作数指出参与运算的对象,以及运算结果所存放的位置等。

由于机器指令与 CPU 紧密相关,所以,不同种类的 CPU 所对应的机器指令也就不同,而且它们的指令系统往往差别很大。但对同一系列的 CPU 来说,为了满足各型号之间具有良好的兼容性,要做到:新一代 CPU 的指令系统必须兼容先前同系列 CPU 的指令系统。只有这样,先前开发出来的各类程序在新一代 CPU 上才能正常运行。

机器语言是用来直接描述机器指令、使用机器指令的规则等。它是 CPU 能直接识别的唯一一种语言,也就是说,CPU 能直接执行用机器语言描述的程序。用机器语言编写的程序不易读、出错率高、难以维护,也不能直观地反映用计算机解决问题的基本思路,现在几乎没有程序员这样编写程序了。

### 1.1.2 汇编语言

虽然用机器语言编写程序有很高的要求和许多不便,但编写出来的程序执行效率高,CPU 严格按照程序员的要求去做,没有多余的额外操作。所以,在保留“程序执行效率高”的前提下,人们就开始着手研究一种能大大改善程序可读性的编程方法。

为了改善机器指令的可读性,选用了一些能反映机器指令功能的单词或词组来代表该机器指令,而不再关心机器指令的具体二进制编码。与此同时,也把 CPU 内部的各种资源符号化,使用该符号名也等于引用了该具体的物理资源。

如此一来,令人难懂的二进制机器指令就可以用通俗易懂的、具有一定含义的符号指令来表示了,于是就有了汇编语言。现在,称这些具有一定含义的符号为助记符,用指令助记符、符号地址等组成的符号指令称为汇编指令。

汇编语言是汇编指令集、伪指令集和使用它们的规则的统称。伪指令是在程序设计时所需要的一些辅助性说明指令,它不对应具体的机器指令,有关内容在以后的各章节中会有详细叙述,在此不展开介绍。

用汇编语言编写的程序称为汇编语言程序,或汇编语言源程序。汇编语言程序要比用机器指令编写的程序容易理解和维护。

### 1.1.3 汇编语言程序

用汇编语言编写的程序大大提高了程序的可读性,但失去了 CPU 能直接识别的特性。例如用汇编语言书写的指令“MOV AX, BX, CPU”不会知道这几个字符所表达出来的功能,但程序员一看就知道:要求 CPU 把寄存器 BX 的值传送给寄存器 AX。

把机器指令符号化增加了程序的可读性,但引起了如何让 CPU 知道程序员的用意,并按照其要求完成相应操作的问题。解决该问题就需要一个翻译程序,它能把汇编语言编写的源程序翻译成 CPU 能识别的机器指令序列。这里称该翻译程序为汇编程序。

图 1.1 是汇编语言程序翻译过程的示意图。从图中可以看出,汇编程序能把左边汇编语言源程序翻译成右边的机器指令序列。其中,把汇编语言指令“MOV AX, BX”和“ADD AX, 8”分别转换成机器指令 89D8H 和 050800H,而后者都是 CPU 能直接识别和执行的。

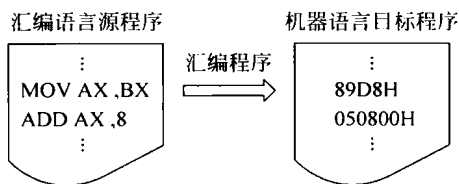


图 1.1 汇编语言程序翻译过程

### 1.1.4 汇编语言的主要特点

汇编语言指令是用一些具有相应含义的助记符来表达的,所以,它比机器语言容易掌握和运用,但是要直接使用 CPU 的资源,相对高级程序设计语言来说,又是很难掌握的。汇编语言程序归纳起来大概有以下几个主要特性。

### 1) 与机器相关

汇编语言指令是机器指令的一种符号表示,而不同类型的 CPU 有不同的机器指令系统,也就有不同的汇编语言,所以,汇编语言程序与机器有着密切的关系。除了同系列、不同型号 CPU 之间的汇编语言程序有一定程度的可移植性之外,其他不同类型(如:小型机和微型机等)CPU 之间的汇编语言程序是无法移植的,也就是说,汇编语言程序的通用性和可移植性要比高级语言程序低。

### 2) 执行效率高

正因为汇编语言有“与机器相关”的特性,程序员用汇编语言编写程序时,可充分发挥自己的聪明才智,对机器内部的各种资源进行合理的安排,让它们始终处于最佳的使用状态,这样做的最终效果就是:程序的执行代码短,执行速度快。

### 3) 编写程序难度大

汇编语言是一种面向机器的语言,其汇编指令与机器指令基本上一一对应,所以,汇编指令也同机器指令一样具有功能单一、具体的特点。由于汇编语言程序要安排运算的每一个细节,这就使得编写汇编语言程序比较烦琐、复杂。一个简单的计算公式或计算方法,也要用一系列汇编指令一步一步来实现。

## 1.1.5 汇编语言的使用领域

汇编语言的优点也直接导致其严重的缺点,其“与机器相关”和“执行的高效率”导致其可移植性差和调试困难。所以,我们在选用汇编语言时要根据实际的应用环境,尽可能避免其缺点对整个应用系统的影响。

汇编语言主要适用于以下几个领域。

- (1) 要求执行效率高、反应快的领域,如:操作系统内核、工业控制、实时系统等。
- (2) 与硬件资源密切相关的软件开发,如:设备驱动程序等。
- (3) 受存储容量限制的应用领域,如:家用电器的计算机控制功能等。
- (4) 没有适当的高级语言开发环境。

其他情况下一般不宜于使用汇编语言开发,如大型软件的整体开发,或者是没有特殊要求的一般应用系统的开发等。但是在高级程序设计开发中,为了解决系统性能瓶颈,或者对于频繁被使用的子程序等,为了提高效率,往往会利用汇编语言来实现部分程序设计,这就涉及高级语言和汇编语言的混合程序设计,在后续的章节中会有叙述。

## 1.2 数据的表示和类型

在用汇编语言进行程序设计时,程序员可以直接访问内存,对数据在存储器内的表示形式要有一个清晰的认识。下面,只简单介绍本课程所要用到的数据表示知识,为后面的学习做一些必要的准备。

### 1.2.1 数值数据的表示

#### 1) 二进制

在计算机内,数值是用二进制来表示的,每个二进制数按权相加就可得到其十进制数

值。在书写二进制时,为了区别,在数据后面紧跟一个字母 B。数据的二进制表示形式简单明了,但它书写起来比较长,所以,通常情况下,在程序中不直接用二进制来书写具体的数值,而改用八进制、十进制或十六进制。

### 2) 八进制

八进制是一种二进制的变形,三位二进制数可变为一位八进制数;反之亦然。八进制的表示元素是:0,1,...,7。在书写时,为了区别,在数据后面紧跟一个字母 Q。如:1234Q、7654Q、54Q 等都是八进制。

### 3) 十进制

十进制是我们最熟悉的一种数据表示形式,它的基本元素是:0,1,...,9。在书写时,为了区别,在数据后面紧跟一个字母 D,如:1234D。在程序中经常用十进制来表示数据。一般情况下,在程序设计中都默认使用十进制,所以可以在数据后面省略字母 D,如 1234、200 等。

### 4) 十六进制

十六进制是另一种二进制的变形,四位二进制数可变为一位十六进制数;反之亦然。十六进制的基本元素是:0,1,...,9,A,B,...,F(字母也可以小写),其中:字母 A,B,...,F 依次代表 10,11,...,15。在书写时,为了区别,在数据后面紧跟一个字母 H。当十六进制数的第一个字符是字母时,在第一个字符之前必须添加一个“0”。如:100H、56EFH、0FFH、0ABCDH 等都是十六进制数。

在汇编语言指令或程序中可以任意使用以上 4 种进制的数据,不管是指令操作数还是内存地址,具体数据在对应的机器指令中一般都是以其补码形式表示的。

## 1.2.2 非数值数据的表示

计算机除了具有进行数值计算能力之外,还具有进行非数值计算的能力。所以,对非数值信息的编码就显得越加重要。这里只介绍字符数据的表示。

### 1. ASCII 码

ASCII(American Standard Code for Information Interchange)码是目前应用极其广泛的一种非数值信息编码,许多计算机系统都是采用它为字符进行编码。它是一种 7 位二进制编码,如表 1.1 所示。在该表中,字符 0~9 是连续编码的,其编码的低 4 位就是该字符在十进制中的数值;小写字母的编码比大写字母的编码大,对应字母的编码之间相差 20H。表中的 000 列和 001 列是控制符号的编码,控制符号是不可显示的。

表 1.1 ASCII 码的编码方案

| 高位<br>低位 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|
| 0000     | NUL | DEL | SP  | 0   | @   | P   |     | p   |
| 0001     | SOH | DC1 | !   | 1   | A   | Q   | a   | q   |
| 0010     | STX | DC2 | "   | 2   | B   | R   | b   | r   |
| 0011     | ETX | DC3 | #   | 3   | C   | S   | c   | s   |
| 0100     | EOT | DC4 | \$  | 4   | D   | T   | d   | t   |
| 0101     | ENQ | NAK | %   | 5   | E   | U   | e   | u   |

续表

| 高位<br>低位 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|
| 0110     | ACK | SYN | &   | 6   | F   | V   | f   | v   |
| 0111     | BEL | ETB | '   | 7   | G   | W   | g   | w   |
| 1000     | BS  | CAN | (   | 8   | H   | X   | h   | x   |
| 1001     | HT  | EM  | )   | 9   | I   | Y   | i   | y   |
| 1010     | LF  | SUB | *   | :   | J   | Z   | j   | z   |
| 1011     | VT  | ESC | +   | ;   | K   | [   | k   | {   |
| 1100     | FF  | FS  |     | <   | L   | \   | l   |     |
| 1101     | CR  | GS  | —   | =   | M   | ]   | m   | }   |
| 1110     | SO  | RS  | .   | >   | N   | ^   | n   | ~   |
| 1111     | SI  | US  | /   | ?   | O   | —   | o   | Del |

## 2. 汉字编码

ASCII 码是针对英文的字母、数字和其他特殊字符进行编码的,它不能用于对汉字的编码。要想用计算机来处理汉字,就必须先对汉字进行适当的编码。我国在 1981 年 5 月对 6000 多个常用的汉字制定了交换码的国家标准,即 GB 2312—80。该标准规定了汉字交换用的基本汉字字符和一些图形字符,它们共计 7445 个,其中汉字有 6763 个。该标准给定每个字符的双字节二进制编码,即国标码,而存储时还要转换为相应的机内码。目前微机操作系统普遍使用的是 Unicode 编码。有关汉字编码的详细信息,可参阅其他有关书籍,在此不再介绍。

### 1.2.3 基本的数据类型

汇编语言所用到的基本数据类型为字节、字、双字等,这些数据类型在以后的章节中都有相应的类型说明符。下面对它们进行最基本的描述。

#### 1. 字节

一个字节由 8 位二进制数组成,其最高位是第 7 位,最低位是第 0 位,如图 1.2 所示。在表示有符号数时,最高位就是符号位。通常情况下,存储器按字节编址,读写存储器的最小信息单位就是一个字节。

#### 2. 字

由两个字节组成一个字,其最高位是第 15 位,最低位是第 0 位。高 8 位称为高字节,低 8 位称为低字节,如图 1.3 所示。字节和字是汇编语言程序中最常用的两种数据类型,也是最容易出错的数据类型。

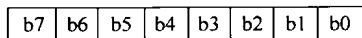


图 1.2 字节数据类型示意图

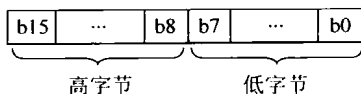


图 1.3 字数据类型示意图

### 3. 双字

用两个字(4个字节)来组成一个双字,其高16位称为高字,低16位称为低字,如图1.4所示。双字有较大的数据表示范围,它通常是为了满足数据的表示范围而选用的数据类型,也可用于存储远指针,在32位程序设计时使用较多。

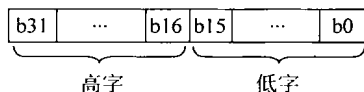


图 1.4 双字数据类型示意图

### 4. 四字

由4个字(8个字节)组成一个四字类型,它总共有64个二进制位,当然,也就有更大的数据表示范围,通常用于64位汇编语言程序设计。

### 5. 十字节

由10个字节组成一个十字节类型,它总共有80个二进制位。在汇编语言中很少使用该数据类型。

### 6. 字符串

字符串是由若干个字节组成的,字节数不定,通常每个字节存储一个字符。该数据形式是汇编语言程序中经常使用的另一种数据形式。

## 习题 1

- 1.1 汇编语言的主要特点有哪些?
- 1.2 汇编语言适用于哪些领域,在哪些领域使用不太方便?
- 1.3 在汇编语言中,如何表示二进制、八进制、十进制和十六进制的数值?
- 1.4 在计算机中,如何表示正、负数?在保持数值大小不变的情况下,如何把位数少的二进制数值扩展成位数较多的二进制数值?
- 1.5 在ASCII表,字符'0'~'9'与数值0~9之间的编码规律是什么?大写字母和小写字母之间的编码规律是什么?
- 1.6 汇编语言中的基本数据类型有哪些?它与高级程序设计语言(如C语言)中的数据类型的对应关系是什么?