

高等院校精品课程系列教材

# 汇编语言

郑晓薇 ©编著



*Assembly Language*



机械工业出版社  
China Machine Press

高等院校精品课程系列教材

# 汇编语言

郑晓薇 ©编著



*Assembly Language*



机械工业出版社  
China Machine Press

汇编语言是计算机专业的专业基础课程，也是电子、通信及自动控制等相关专业计算机技术课程的内容。本书以 80X86 系列微型计算机为基础，以 MASM5.0 为汇编上机实验环境，重点介绍 Intel8086 指令系统。本书的写作特点为采用实例驱动教学的方法，以丰富的示例和实例在 DEBUG 下的运行结果为依托展开教学和学习；在编写思路，将指令系统分散到相关章节，指令的学习与示例、实例结合。采用启发式设问引导教学，在编写结构上提出一个学习框架，便于读者思考和学习。本书中实验练习贯穿始终，在各章中布置了实验任务模块，并在第 10 章专门安排了综合性设计性实验内容，通过多层次的实验训练来加强各章内容的学习理解、融会贯通。

全书结构清晰，内容丰富，例题多样，练习和习题针对性强。所有程序都经过运行验证，习题和测验附有答案。与本书配套有多媒体 PPT 课件、书中的例题程序及习题答案等，可免费赠送使用本书做教材的教师。

本书可以作为计算机专业汇编语言课程的教材（含实验），或者作为其他专业相关课程的参考书和自学教材。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所



图书在版编目 (CIP) 数据

汇编语言/郑晓薇编著. —北京: 机械工业出版社, 2009.5  
(高等院校精品课程系列教材)

ISBN 978-7-111-26907-6

I. 汇… II. 郑… III. 汇编语言—程序设计—高等学校—教材 IV. TP313

中国版本图书馆 CIP 数据核字 (2009) 第 061218 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 李俊竹

三河市明辉印装有限公司印刷

2009 年 9 月第 1 版第 1 次印刷

184mm × 260mm · 16.75 印张

标准书号: ISBN 978-7-111-26907-6

定价: 29.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换  
本社购书热线: (010) 68326294

# 前言

汇编语言是计算机专业的专业基础课程，也是电子、通信及自动控制等相关专业计算机技术课程的内容。通过汇编语言的学习，可以使學生具有在 CPU 的寄存器级上进行控制和操作的能力，获得直接对计算机硬件底层编程的经验，从而对计算机系统有更深刻的认识。这样，在学习操作系统、微机原理、嵌入式技术等课程时，思路会更开阔，基础会更扎实，分析问题会站在更深的层面，许多问题就会迎刃而解。

本书以 80X86 系列微型计算机为基础，以 MASM5.0 为汇编上机实验环境，重点介绍 Intel8086 指令系统。本书的写作特点：

1) 采用实例驱动教学的方法，讲解汇编语言的基本概念和实用程序设计技术。每章的最后一节都给出一个实例，对本章的学习内容加以归纳，得出一个有特色的论点。在各章中则以丰富的示例为依托展开教学和学习，示例的选择由浅入深，最后归结到实例上。在编写思路上，将指令系统分散到相关章节，指令的学习融化在示例、实例中，避免所有指令集中在一章介绍所产生的记不住、消化不了的现象。每个示例、实例以在 DEBUG 下的运行过程出现，给读者一个直观的印象，便于教师的讲解、学生的自学。

2) 启发式设问引导教学。汇编语言难学的原因有多种，包括指令系统的繁杂、难记，涉及机器硬件层面，要从机器的角度以二进制和十六进制的思维考虑问题，许多工作必须逐条指令构成和执行，不像高级语言用一条语句就解决问题，等等。因此作者从启发式教学的角度，在每章的开始部分都提出了一些问题，使读者在学习本章内容之前，先想到一些问题、提出一些问题，然后带着问题学习，就会收到事半功倍的效果。

3) 构造学习框架。对于学习一门技术而言，模仿是快速掌握技能的一个捷径。只有对一事物有了解、获得了初步的技能，才能对该事物产生兴趣、激发出热情。作者在编写结构上提出一个学习框架，对每一个示例题目，按照分析题意、设计思路、程序框图、程序代码、运行结果显示和结果分析的框架结构编写。对示例程序的分析以 DEBUG 下的操作和运行结果为依据，使读者有样板可学，有结果

可见。同时在每个知识点上增加了练习部分，采用边学习边练习的方式。在每章内容之后有习题和测验题，书后附有参考答案，以加强读者对相关概念的学习与吸收。

4) 实验训练贯穿始终。汇编语言是一门理论与实践相结合的课程，只有在大量的编程训练下，才能很好地掌握基础理论与编程技巧。因此本书特别强调动手训练，在前面各章节中都采用边讲解理论边练习的方式；同时在每章中更安排了实验任务模块，以期通过多层次的实验训练来加强各章内容的学习理解、融会贯通。在第10章专门安排了综合性设计性实验内容，使学生在学习的后期能够达到自己设计一个较大型的综合性设计性实验，也是对汇编语言的学习做一个总结和检验。

全书共分10章。第1章基础知识，重点是数的正确表示。第2章计算机基本原理，主要强调CPU的寄存器和存储单元的概念和使用，实验内容是用DEBUG观察寄存器和存储器。第3章指令系统与寻址方式，重点是数据的寻址方式，实验内容为DEBUG下汇编指令的输入与执行。第4章汇编语言程序设计，以一个公式计算程序的设计为例，给出汇编语言程序的设计过程、伪指令的作用和基本的汇编指令。第5章分支程序设计，穿插了用位操作指令设计分支程序；深入分析转移特征，给出系统启动和程序加载过程。第6章循环程序设计，同时加入串处理的概念和用法。第7章子程序设计，重点是子程序的调用和返回、参数传递，实例中提出了模块化结构概念。第8章宏汇编技术，加入了结构伪操作、重复汇编和条件汇编，实例为多个代码段下的多模块设计。第9章中断程序设计，给出了中断的绝大部分概念，对系统中断作了较深入的分析，教会读者如何定制自己的中断。第10章综合性设计性实验，在介绍端口概念的同时，以读取CMOS时钟为例讲解了I/O接口实验；并给出了六个综合性设计性实验题目和要求。

全书结构清晰，内容丰富，例题多样，练习和习题针对性强。书中语言浅显易懂、详略得当。所有程序都经过运行验证，习题和测验附有答案。与本书配套的多媒体PPT课件，书中的例题程序及习题、测验和答案等教辅资源可在出版社网站上下载。

本书是作者在多年讲授汇编语言课程过程中教学经验的积累，是对汇编语言课程进行教学改革成果。本书得到了辽宁师范大学特色自编教材资助。

在教材编写过程中，得到相关老师和学生的帮助，也参考了其他同行的教材，作者在此深表感谢。同时，还要感谢机械工业出版社华章分社的编辑们，是他们的大力支持才使本书能够很快出版。书中难免有错误和不当之处，敬请读者指正。

郑晓薇

2009年2月

# 教学建议

教学内容	学习要点及教学要求	课时安排	
		计算机类 本科专业	非计算机类 本科专业
第1章 基础知识	<ul style="list-style-type: none"> <li>了解汇编语言基本概念。</li> <li>二进制、十六进制、十进制的相互转换，十六进制的加减乘运算必须熟练掌握。</li> <li>补码表示、补码运算、符号位扩展是汇编指令的基础，要求熟练掌握。</li> <li>理解BCD码、ASCII码的含义和作用。</li> <li>数的正确表示。理解机器中保存的二进制数的含义。</li> </ul>	2~4	4
第2章 计算机 基本原理	<ul style="list-style-type: none"> <li>理解冯·诺依曼计算机的结构和基本特点。</li> <li>理解汇编语言与微型计算机系统的相互关系。了解微处理器的结构。</li> <li>寄存器的概念和用法在汇编语言中十分重要，要求重点掌握8086CPU的寄存器类型及作用。理解标志寄存器中标志位的含义和用法。</li> <li>内存是非常重要的学习内容。重点掌握物理地址和逻辑地址的概念及转换，存储单元的属性，存储器分段概念。</li> <li>初步掌握调试工具DEBUG的使用。上机实验任务为DEBUG常用命令的用法。</li> </ul>	4~6	4~6
第3章 指令系统与 寻址方式	<ul style="list-style-type: none"> <li>理解汇编指令的格式和属性。</li> <li>了解8086指令系统的分类。</li> <li>理解寻址的含义，掌握操作数与寻址的关系。</li> <li>熟练掌握立即寻址方式、寄存器寻址方式、存储器寻址方式的特点及指令表示。</li> <li>了解与数据有关的各种寻址方式的选择特点。能够根据寻址方式的要求，写出相关指令。</li> <li>上机实验任务是在DEBUG下观察和分析存储单元的逻辑地址的表示以及指令的执行结果。</li> <li>熟练掌握DEBUG的R命令、A命令、T命令和D命令的用法。</li> </ul>	4~6	4

(续)

教学内容	学习要点及教学要求	课时安排	
		计算机类 本科专业	非计算机类 本科专业
第4章 汇编语言 程序设计	<ul style="list-style-type: none"> <li>了解汇编语言程序设计的基本步骤。</li> <li>掌握用汇编指令实现设计思路及算法的方法。</li> <li>了解从源程序到生成可执行程序的过程和汇编环境要求。</li> <li>伪指令是汇编语言的重要概念。理解伪指令的用法和功能。</li> <li>重点掌握基本的汇编指令。能够根据题意编写出顺序程序。</li> <li>熟练编写出数值运算程序,了解BCD码十进制运算程序的作用。</li> <li>掌握屏幕显示和键盘输入等DOS功能调用INT 21H指令的用法。</li> <li>上机实验任务为简单的算术运算程序的编写及调试。</li> </ul>	8~10	6~8 选讲
第5章 分支程序设计	<ul style="list-style-type: none"> <li>理解分支的概念及分支结构的特点。</li> <li>理解与转移地址有关的寻址方式。理解CS、IP寄存器与转移地址的关系。重点掌握段内寻址与段间寻址的区别。</li> <li>熟练掌握与分支有关指令的用法。</li> <li>重点掌握条件转移的四种指令的格式与用法。</li> <li>学会用转移指令编写分支程序。</li> <li>熟练掌握逻辑指令和移位指令。学会编写对数字和字母的判断程序,掌握查表程序的设计方法。</li> <li>掌握进制转换程序设计方法,菜单程序设计方法。</li> <li>上机实验任务为简单的字符判断及进制转换程序的编写及调试。</li> </ul>	8~10	6~8 选讲
第6章 循环程序设计	<ul style="list-style-type: none"> <li>了解循环与分支的异同。</li> <li>掌握循环指令的用法。会用循环指令编写简单的循环程序。</li> <li>掌握串处理指令。理解串处理与循环的关系,了解实现循环的几种方式。</li> <li>理解多重循环的概念,重点掌握数组排序程序的编写方法。</li> <li>综合分支与循环的用法,编写出具有判断和循环功能的程序,如多字节数组元素相加、求数组中最大值、删除数组元素、数组分割等程序。</li> <li>上机实验任务为上述有关数组程序的验证及改造。</li> </ul>	6~8	6~8 选讲
第7章 子程序设计	<ul style="list-style-type: none"> <li>了解子程序的概念。</li> <li>理解和掌握子程序调用指令CALL和子程序返回指令RET的作用及执行过程。</li> <li>掌握过程定义伪指令的格式和用法。理解子程序的属性概念。掌握现场保护和子程序参数传递的作用和用法。</li> <li>能够根据题目设计出简单的子程序。</li> <li>初步学会编写主程序调用子程序、子程序的参数传递。</li> <li>了解子程序的嵌套与递归。</li> <li>了解模块化结构的概念与结构特点。</li> <li>熟练掌握用键盘输入的数据进行算术运算及显示结果的多功能程序设计方法。</li> <li>上机实验任务为键盘输入十进制数运算程序的设计与验证。</li> </ul>	6~8	4~6 选讲

(续)

教学内容	学习要点及教学要求	课时安排	
		计算机类 本科专业	非计算机类 本科专业
第8章 宏汇编技术	<ul style="list-style-type: none"> <li>理解宏的概念。掌握宏与子程序的区别。</li> <li>充分理解宏定义、宏调用、宏展开的作用和用法。</li> <li>掌握宏库的概念。会将常用的简单功能程序定义为宏或带有哑元的宏。</li> <li>了解结构、重复汇编和条件汇编的概念。</li> <li>了解多个代码段下多模块程序的编写方式。了解只有一个小段的小型程序设计方法。</li> <li>熟练掌握利用宏来简化程序的方法。</li> <li>上机实验任务为利用宏编写的输入十进制数求其补码或反码，用二进制和十六进制显示的多功能程序的验证和改造。</li> </ul>	4~6	2~4 选讲
第9章 中断程序设计	<ul style="list-style-type: none"> <li>了解中断的概念及 CPU 响应和执行中断处理程序的过程。理解中断程序与子程序的区别。</li> <li>理解中断源的概念。掌握中断类型与中断向量的概念及用法。</li> <li>掌握基本的 BIOS 中断调用和 DOS 调用功能。尤其是键盘输入、光标控制、屏幕显示、时钟中断、系统日期和时间的读取等功能。</li> <li>了解编写自己定制的中断程序和改变系统中断程序的设计方法。</li> <li>熟练掌握清屏、开窗口、置光标、设置字符颜色等中断功能调用方法。</li> <li>上机实验任务为编写具有上述屏幕功能与读取定时时间计数值的中断指令多功能程序的验证和改造。</li> </ul>	6~8	4~6
第10章 综合性 设计性实验	<ul style="list-style-type: none"> <li>了解 I/O 端口的概念。</li> <li>掌握输入输出指令 IN/OUT 的用法。</li> <li>学会通过 I/O 端口读取 CMOS 时间和日期的方法。</li> <li>强调综合性设计性实验的作用和重要性，了解各个实验题目的作用和设计方法。</li> <li>在教学或学习的中后期布置综合性设计性实验题目。</li> <li>综合性设计性实验属于较大型实验题目，可让学生任选一个或分组选择若干个。</li> <li>由于学时有限，布置后需要学生课下配合完成，再由教师统一检查，给出成绩。</li> </ul>	4~6	2~4
教学总学时建议		52~72	42~58

说明：

1. 本书是计算机类本科专业汇编语言课程教材，授课（含实验）学时数为 52~72 学时。有关专业可根据不同的教学要求和计划教学学时酌情对教材内容进行适当取舍。如果学时数较少，可以在各章中少讲几个示例。
2. 非计算机类本科专业、计算机类专科专业使用本教材可适当降低教学要求。
3. 本书授课学时数 52~72 学时包含实验课、课堂讨论等必要的教学环节。讲授与实验的比例约为 3:1。
4. 若某些计算机类本科专业和非计算机类本科专业计划教学学时数少于 52 学时，可舍去教学建议中属于“了解”的内容，“理解”和“掌握”的内容也可部分删减。
5. 汇编语言教学必须强调与实际机器结合，教师讲课时最好能以 DEBUG 环境下的运行过程和结果为例讲解；如有可能尽量增加实验学时，或者布置上机作业让学生课外完成。
6. 综合性设计性实验题目可以由教师指定其一，或者由学生自选。



# 目录

前言

教学建议

## 第1章 基础知识 ..... 1

- 1.1 汇编语言基本概念 ..... 1
- 1.2 计算机中数的表示 ..... 2
  - 1.2.1 进制转换 ..... 2
  - 1.2.2 进制运算 ..... 5
  - 1.2.3 补码运算 ..... 6
  - 1.2.4 编码 ..... 9
- 1.3 实例一 揭开数的面纱 ..... 13
  - 1.3.1 数的正确表示 ..... 13
  - 1.3.2 数的符号问题 ..... 14
- 习题一 ..... 14
- 测验一 ..... 15

## 第2章 计算机基本原理 ..... 16

- 2.1 冯·诺依曼计算机 ..... 16
  - 2.1.1 冯·诺依曼计算机的原理 ..... 16
  - 2.1.2 冯·诺依曼计算机的基本结构 ..... 17
- 2.2 微型计算机系统 ..... 18
  - 2.2.1 微型计算机系统概念 ..... 18
  - 2.2.2 微处理器 ..... 19
- 2.3 80X86 寄存器 ..... 19
  - 2.3.1 8086 寄存器组 ..... 20
  - 2.3.2 80X86 寄存器组 ..... 22
- 2.4 内存储器 ..... 22
  - 2.4.1 物理地址与逻辑地址 ..... 22
  - 2.4.2 存储单元 ..... 23
  - 2.4.3 存储器分段 ..... 24

2.5 实例二 进入计算机 ..... 27

- 2.5.1 调试工具 DEBUG ..... 27
- 2.5.2 实验任务 ..... 34
- 习题二 ..... 34
- 测验二 ..... 35

## 第3章 指令系统与寻址方式 ..... 37

- 3.1 汇编语言指令 ..... 37
  - 3.1.1 机器指令 ..... 37
  - 3.1.2 汇编指令 ..... 38
  - 3.1.3 指令系统 ..... 39
- 3.2 指令的寻址方式 ..... 39
  - 3.2.1 寻址方式 ..... 39
  - 3.2.2 立即寻址方式 ..... 40
  - 3.2.3 寄存器寻址方式 ..... 41
  - 3.2.4 存储器寻址方式 ..... 41
- 3.3 实例三 寻找操作数 ..... 47
  - 3.3.1 寻址方式的选择 ..... 47
  - 3.3.2 实验示例 ..... 48
  - 3.3.3 实验任务 ..... 49
- 习题三 ..... 49
- 测验三 ..... 50

## 第4章 汇编语言程序设计 ..... 52

- 4.1 汇编语言程序设计初步 ..... 52
  - 4.1.1 第一个汇编语言程序 ..... 53
  - 4.1.2 从源程序到可执行程序 ..... 54
- 4.2 伪指令 ..... 58
  - 4.2.1 段定义伪操作 ..... 58
  - 4.2.2 数据定义伪指令 ..... 60
  - 4.2.3 其他伪指令 ..... 63

4.3 基本汇编指令	65	5.5.1 分支的选择	118
4.3.1 数据、栈及查表	66	5.5.2 菜单程序设计	120
4.3.2 逻辑地址的获得	72	5.5.3 用分支表实现多路转移	121
4.3.3 符号位扩展	73	5.5.4 实验示例	123
4.3.4 双精度数运算	73	5.5.5 实验任务	124
4.3.5 多字节数运算	75	习题五	124
4.3.6 混合算术运算	77	测验五	126
4.3.7 十进制数运算	80	<b>第6章 循环程序设计</b>	128
4.4 屏幕显示和键盘输入	82	6.1 循环的概念	128
4.4.1 DOS 功能调用	82	6.1.1 循环结构	128
4.4.2 直接写显存显示字符	84	6.1.2 循环程序例子	129
4.5 实例四 带彩色显示的算术		6.1.3 与循环有关的指令	130
程序	86	6.2 循环指令	130
4.5.1 简化的程序结构	86	6.2.1 LOOP 循环指令	131
4.5.2 实验示例	88	6.2.2 LOOPZ/LOOPE	131
4.5.3 实验任务	89	6.2.3 LOOPNZ/LOOPNE	132
习题四	89	6.3 串处理	132
测验四	91	6.3.1 串的概念	132
<b>第5章 分支程序设计</b>	94	6.3.2 串处理例子	133
5.1 分支的概念	94	6.3.3 串处理指令	134
5.1.1 分支结构	94	6.3.4 串与循环	137
5.1.2 一个分支程序例子	95	6.4 多重循环	138
5.2 与分支有关的指令	97	6.4.1 多重循环结构	138
5.2.1 转移地址的寻址	98	6.4.2 排序程序	139
5.2.2 条件转移方式	100	6.5 循环程序举例	140
5.3 位操作的分支程序	103	6.6 实例六 循环之循环	143
5.3.1 逻辑运算	104	6.6.1 循环的执行	143
5.3.2 测试指令 TEST	106	6.6.2 实验示例	146
5.3.3 移位操作	108	6.6.3 实验任务	147
5.3.4 分支程序举例	110	习题六	147
5.4 深入分析转移特征	114	测验六	148
5.4.1 内存空间分配	114	<b>第7章 子程序设计</b>	150
5.4.2 系统启动	114	7.1 子程序的概念	150
5.4.3 程序的加载	115	7.1.1 主程序和子程序	150
5.4.4 JMP 转移特征	117	7.1.2 一个改造的例子	151
5.5 实例五 走向分支	118	7.2 调用和返回	153

7.2.1 调用指令 CALL .....	153	8.5 多模块结构 .....	187
7.2.2 返回指令 RET .....	154	8.5.1 多个代码段下的模块 .....	187
7.3 过程定义 .....	154	8.5.2 模块的参数设置 .....	188
7.3.1 伪指令 PROC .....	154	8.6 实例八 宏与多模块 .....	188
7.3.2 过程属性 .....	155	8.6.1 多模块设计 .....	188
7.4 现场保护 .....	155	8.6.2 一个段的模块 .....	192
7.5 子程序参数传递 .....	156	8.6.3 实验示例 .....	195
7.5.1 寄存器传参 .....	156	8.6.4 实验任务 .....	197
7.5.2 存储单元传参 .....	158	习题八 .....	197
7.5.3 堆栈传参 .....	160	测验八 .....	199
7.6 嵌套与递归 .....	162	<b>第9章 中断程序设计</b> .....	200
7.6.1 子程序嵌套 .....	162	9.1 中断的概念 .....	200
7.6.2 子程序递归 .....	163	9.1.1 软件中断 .....	200
7.7 实例七 子程序与模块化 .....	163	9.1.2 硬件中断 .....	201
7.7.1 模块化结构 .....	163	9.1.3 中断类型与中断向量 .....	202
7.7.2 实验示例 .....	168	9.1.4 中断过程 .....	204
7.7.3 实验任务 .....	170	9.2 定制自己的中断 .....	205
习题七 .....	170	9.2.1 软件中断子程序的编写 .....	205
测验七 .....	172	9.2.2 中断的设置 .....	205
<b>第8章 宏汇编技术</b> .....	174	9.2.3 软件中断的触发与处理 .....	206
8.1 宏 .....	174	9.2.4 对除0中断的修改 .....	208
8.1.1 宏定义 .....	174	9.3 BIOS中断 .....	210
8.1.2 宏调用 .....	175	9.3.1 屏幕及光标控制 INT 10H .....	211
8.1.3 宏展开 .....	175	9.3.2 键盘中断 INT 16H .....	216
8.1.4 宏与子程序 .....	176	9.3.3 时钟中断 INT 1AH .....	218
8.1.5 宏的参数 .....	177	9.4 DOS中断 .....	219
8.1.6 宏运算 .....	179	9.4.1 DOS显示功能调用 .....	219
8.2 其他宏功能 .....	180	9.4.2 DOS键盘功能调用 .....	220
8.2.1 宏标号 .....	180	9.4.3 DOS日期、时间功能 调用 .....	220
8.2.2 宏删除 .....	181	9.5 实例九 中断程序应用 .....	221
8.2.3 宏嵌套 .....	181	9.5.1 时间与计数 .....	221
8.2.4 宏库建立与调用 .....	182	9.5.2 实验示例 .....	226
8.3 结构伪操作 .....	185	9.5.3 实验任务 .....	229
8.4 重复汇编和条件汇编 .....	186	习题九 .....	229
8.4.1 重复汇编 .....	186	测验九 .....	231
8.4.2 条件汇编 .....	186		

<b>第 10 章 综合性设计性实验</b> .....	233	10.2.4 实验四 设计一个小计 计算器 .....	238
10.1 I/O 端口实验 .....	233	10.2.5 实验五 小学生算术练习 软件 .....	239
10.1.1 I/O 端口地址 .....	233	10.2.6 实验六 进制及编码转换 工具 .....	240
10.1.2 IN 指令和 OUT 指令 .....	234		
10.1.3 读取 CMOS 时钟 .....	235		
10.2 实验项目 .....	236	<b>附录 A 8086 指令系统表</b> .....	241
10.2.1 实验一 CMOS 时间和 日期 .....	236	<b>附录 B 汇编出错提示信息</b> .....	248
10.2.2 实验二 英文打字练习 软件 .....	237	<b>附录 C 汇编语言各章测验答案</b> .....	251
10.2.3 实验三 英文填字游戏 软件 .....	238	<b>参考文献</b> .....	252

# 第 1 章

## 基础知识

设问：

1. 为什么要学习汇编语言？
2. 什么是汇编语言？
3. 为什么要用十六进制数？
4. 怎样区别计算机中数的含义？

学习汇编语言，重要的是掌握如何通过汇编指令和程序来控制计算机各个组成部件工作，完成一系列任务。因此学习汇编语言与学习高级语言的不同之处是要学习如何深入到计算机的内部进行控制，包括控制 CPU 的数据寄存器、地址寄存器、标志寄存器；对存储器的读写操作，对输入设备和输出设备的访问等具体操作。

学会了汇编语言，就能够在 CPU 的寄存器级上进行控制和操作，掌握直接对计算机硬件编程的方法，从而对计算机系统有更深刻的认识，分析问题的角度就会处在更深的层面。而高级语言采用英语单词表示程序语句，使用通常的数学表达式以及专门的语法规则编写程序。通过高级语言，不懂计算机原理和结构的人也可编写出程序来，但是他们有可能只知其一，不知其二。用高级语言编写的程序对计算机的控制不如用汇编语言编写的程序对机器的控制来得更直接、更有效和迅速。

本章介绍学习汇编语言所需的基本知识，并通过具体的例子为读者建立起汇编语言的初步概念。

### 1.1 汇编语言基本概念

众所周知，计算机以二进制数为基础。那么控制计算机工作的机器指令就由二进制数构成，而机器指令的集合称之为机器语言。如果想让计算机工作，就要写出一系列二进制的机器码。计算机获得这些机器指令后，立即而迅速地完成任务。例如计算  $Z = 35 + 27$ ，写成机器指令为：

二进制表示

101110000010001100000000

000001010001101100000000

101000110000010000000000

十六进制表示

B82300

051B00

A30400

可以看出，机器指令又晦涩又难记。在上例中二进制数用了 24 位，写成十六进制也要用 6 位。将这种用二进制表示的机器指令改为便于记忆和理解的助记符表示形式。助记符采用英文单词的缩写表示，容易理解。上例计算用指令助记符表示如下：

```
MOV AX,35
ADD AX,27
MOV Z,AX
```

其中，MOV 代表传送，第 1 条指令表示把 35 传送给 AX 寄存器；ADD 代表加法，该指令表示将 AX 中的值与 27 相加，结果再放到 AX 中；最后一条指令表示将 AX 中的计算结果放入 Z 存储单元中。这样的指令就好理解了，可以容易地将计算程序编写出来。

这些助记符就是汇编指令，汇编指令的集合构成了汇编语言。汇编语言是一种符号化的机器语言。汇编语言既便于程序员编写程序，又保留了机器语言可直接而迅速地控制机器的长处。可以说，汇编语言是直接控制计算机工作的最简便的语言。

但是用汇编语言编写程序时要遵循一定的语法规则，这些规则与高级语言相似，程序员要按照规则编写程序以便于翻译程序进行翻译。对于汇编语言而言，程序员编写的程序称为源程序，翻译程序是一种称为汇编程序的系统软件，“翻译”的过程称为汇编。这是规范的叫法。

汇编语言有三种指令形式：汇编指令、伪指令和宏指令。其中，汇编指令可以翻译成二进制的机器指令代码，而伪指令和宏指令不能翻译成机器指令，它们是在汇编期间为汇编程序提供相关信息使用的。总而言之，用户编写的汇编源程序中必须要有汇编指令和伪指令，宏指令可根据需要设定。

在汇编语言中，涉及的基本概念有：数的表示、寄存器、存储单元、指令格式、语法规则等。要想掌握汇编语言的概念和汇编语言程序设计方法，就要先学习和掌握这些基础知识。

## 1.2 计算机中数的表示

在计算机中，数可以用二进制、十六进制、十进制、八进制等表示。我们日常生活中习惯使用十进制，而编写汇编语言程序时经常用到的是十六进制、二进制数据。因此，三种进制之间的相互转换应该熟练掌握；同时还要使自己尽快习惯用十六进制来思考。由于计算机中常用的进制是二进制、十六进制、十进制，本书主要介绍这三种进制数。

### 1.2.1 进制转换

#### 1. 进制数的三要素

基数、权、进位规则是描述一种进制数的三个要素。在表示数值时，各进制数可以写成多项式展开的形式。在汇编语言中，数值后面分别用字母 B、H、D 代表二进制 (Binary)、十六进制 (Hexadecimal)、十进制数 (Decimal) (十进制数可以省略 D)。

##### (1) 十进制数

数码：0、1、2、3、4、5、6、7、8、9

基数：10

权：10 的  $N-1$  次方

进位规则：逢十进一

例如：十进制数  $257.36 = 2 \times 10^2 + 5 \times 10^1 + 7 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2}$

##### (2) 二进制数

数码：0、1

基数：2

权：2 的  $N-1$  次方

进位规则：逢二进一

例如：二进制数  $1101.01 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$

(3) 十六进制数

数码：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F

基数：16

权：16的N-1次方

进位规则：逢十六进一

例如：十六进制数  $3A6.52 = 3 \times 16^2 + A \times 16^1 + 6 \times 16^0 + 5 \times 16^{-1} + 2 \times 16^{-2}$

各进制数的表示和相互之间的关系应该熟练掌握，尤其是二进制和十六进制数的表示应该脱口而出，形成以二进制和十六进制的思维方式去联想数值关系。

各进制数据对应关系如表 1-1 所示。

表 1-1 各进制数值对照

二进制	十进制	十六进制	八进制
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	8	8	10
1001	9	9	11
1010	10	A	12
1011	11	B	13
1100	12	C	14
1101	13	D	15
1110	14	E	16
1111	15	F	17

在计算机中，数据都是以二进制表示的，因此采用2的n次方形式描述数的权值大小比较方便。

$2^{10} = 1024$ ，称为1K； $2^{20} = 1024K$ ，称为1M； $2^{30} = 1024M$ ，称为1G； $2^{40} = 1024G$ ，称为1T。

二进制权值对照关系如表 1-2 所示。

表 1-2 二进制权值对照表

$2^0 = 1$			
$2^1 = 2$	$2^{11} = 2048 = 2K$	$2^{21} = 2M$	$2^{31} = 2G$
$2^2 = 4$	$2^{12} = 4096 = 4K$	$2^{22} = 4M$	$2^{32} = 4G$
$2^3 = 8$	$2^{13} = 8192 = 8K$	$2^{23} = 8M$	$2^{33} = 8G$
$2^4 = 16$	$2^{14} = 16384 = 16K$	$2^{24} = 16M$	$2^{34} = 16G$
$2^5 = 32$	$2^{15} = 32768 = 32K$	$2^{25} = 32M$	$2^{35} = 32G$
$2^6 = 64$	$2^{16} = 65536 = 64K$	$2^{26} = 64M$	$2^{36} = 64G$
$2^7 = 128$	$2^{17} = 131072 = 128K$	$2^{27} = 128M$	$2^{37} = 128G$
$2^8 = 256$	$2^{18} = 262144 = 256K$	$2^{28} = 256M$	$2^{38} = 256G$
$2^9 = 512$	$2^{19} = 524288 = 512K$	$2^{29} = 512M$	$2^{39} = 512G$
$2^{10} = 1024 = 1K$	$2^{20} = 1048576 = 1024K = 1M$	$2^{30} = 1024M = 1G$	$2^{40} = 1024G = 1T$

## 2. 十进制与其他进制转换

十进制整数部分转换为其他进制数采用“除基取余”法，小数部分转换采用“乘基取整”法。

例1 十进制数 58.125 转换为二进制数。

整数部分，将 58 转换成二进制数，逐次除 2 取余：

$$\begin{array}{r}
 2 \overline{) 58} \\
 \underline{29} \phantom{0} \\
 2 \overline{) 29} \phantom{0} \\
 \underline{14} \phantom{1} \\
 2 \overline{) 14} \phantom{0} \\
 \underline{7} \phantom{0} \\
 2 \overline{) 7} \phantom{0} \\
 \underline{3} \phantom{1} \\
 2 \overline{) 3} \phantom{1} \\
 \underline{1} \phantom{1} \\
 2 \overline{) 1} \phantom{1} \\
 \underline{0} \phantom{1} \\
 0 \phantom{.} \phantom{1}
 \end{array}$$

得到的余数从后至前依次为：1、1、1、0、1、0

可得到：58 = 111010B

小数部分，将 0.125 转换为二进制小数，逐次乘 2 取整（如果最后乘积不能为纯整数，说明此十进制小数不能精确转换成二进制小数，则应该取若干精度位数，例如取到小数点后 3 位）：

$$\begin{array}{r}
 0.125 \\
 \times 2 \\
 \hline
 0.250 \\
 \times 2 \\
 \hline
 0.50 \\
 \times 2 \\
 \hline
 1.0
 \end{array}$$

得到的整数从前至后依次为：0、0、1

可得到：0.125 = 0.001B

即 58.125 = 111010.001 B

例2 十进制数 58.125 转换为十六进制数。

整数部分，将 58 转换成十六进制数，逐次除 16 取余：

$$\begin{array}{r}
 16 \overline{) 58} \\
 \underline{3} \phantom{A} \\
 0 \phantom{3}
 \end{array}$$

得到的余数从后至前依次为：3、A

可得到：58 = 3AH

小数部分，将 0.125 转换为十六进制小数，逐次乘 16 取整：

$$\begin{array}{r}
 0.125 \\
 \times 16 \\
 \hline
 2.000
 \end{array}$$



可得到:  $0.125 = 0.2H$

即  $58.125 = 3A.2H$

**练习:**  $125 = ( \quad ) B = ( \quad ) H$   
 $200 = ( \quad ) B = ( \quad ) H$   
 $33.5 = ( \quad ) B = ( \quad ) H$   
 $68.26 = ( \quad ) B = ( \quad ) H$

### 3. 二进制与其他进制转换

#### (1) 二进制数转换为十进制数

采用按权展开法,也称为多项式展开。

**例** 二进制数  $101101.1B$  转换为十进制数。

$$101101.1B = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = 45.5D$$

#### (2) 二进制数转换为十六进制数

二进制和十六进制之间有一个简单的对应关系,即每4位二进制数可以表示为1位十六进制数。如表1-1所示,二进制数0000对应十六进制数0,二进制数0001对应十六进制数1……二进制数1111对应十六进制数F。

二进制数整数从小数点左边开始每4位一组,小数则从小数点右边开始每4位一组,不够位数以0补齐。

**例** 二进制数  $101101.1B$  转换为十六进制数。

$$101101.1B = \underline{0010} \underline{1101} . \underline{1000} = 2D.8H$$

### 4. 十六进制与其他进制转换

**例1** 十六进制数  $39CH$  转换为十进制数。

按权展开:

$$39CH = 3 \times 16^2 + 9 \times 16^1 + 12 \times 16^0 = 924D$$

**例2** 十六进制数  $39CH$  转换为二进制数。

$$39CH = 0011 \ 1001 \ 1100B$$

**练习:**  $11001101B = ( \quad ) D = ( \quad ) H$   
 $100010001000B = ( \quad ) D = ( \quad ) H$   
 $11111111B = ( \quad ) D = ( \quad ) H$   
 $10000000B = ( \quad ) D = ( \quad ) H$   
 $49C0H = ( \quad ) B$   
 $57H = ( \quad ) B$

## 1.2.2 进制运算

### 1. 二进制运算

二进制只有0和1两个数码,可以使用具有两种稳定状态的元器件来表示。比如器件的输入/输出为高电平时可以表示“1”,为低电平时表示“0”。而逻辑代数同样也使用“真”和“假”两个值,也可以用0、1表示。这就为计算机的逻辑设计提供了便利工具。同时二进制数的运算规则简单。

加法规则:

$$0 + 0 = 0 \quad 0 + 1 = 1$$