

★大学计算机**学习指导**系列★

汇编语言程序设计

学习与解题指南

钱晓捷 编著



华中科技大学出版社

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS
E-mail: hustpp@wuhan.cngb.com

大学计算机学习指导系列

汇编语言程序设计

学习与解题指南

钱晓捷 编著

华中科技大学出版社

图书在版编目(CIP)数据

汇编语言程序设计学习与解题指南/钱晓捷 编著
武汉:华中科技大学出版社, 2002年8月
ISBN 7-5609-2760-2

- I. 汇…
- II. 钱…
- III. 汇编语言-程序设计-高等学校-教学参考资料
- IV. TP312

汇编语言程序设计学习与解题指南

钱晓捷 编著

责任编辑:周正国 沈旭日
责任校对:张兴田

封面设计:潘 群
责任监印:张正林

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87545012

录 排:华中科技大学惠友科技文印中心

印 刷:湖北新华印务有限公司

开本:787×960 1/16

印张:10.75

字数:173 000

版次:2002年8月第1版

印次:2002年8月第1次印刷

印数:1—5 000

ISBN 7-5609-2760-2/TP·473

定价:14.00元

(本书若有印装质量问题,请向出版社发行部调换)

前言

“汇编语言程序设计”是计算机科学与技术学科本科、专科的必修课程，同时也是电子、自动控制等专业所必修的“微机原理”课程的主要内容。由于汇编语言是一种低级语言，涉及微型计算机的工作原理，很多学生感到不易掌握；尤其是参加自学考试的学生更是感到不知如何准备考试。本书希望为这些学生提供帮助。

本书共有7章，其中前5章将“汇编语言程序设计”以如下结构展开，指导读者掌握相应内容：

- (1) 概述——提示本章的学习内容；
- (2) 重点与难点——总结本章的知识要点、分析重点与难点内容；
- (3) 例题精选——选择有代表性的例题，并给予详尽的解答和说明；
- (4) 自测练习——提供选择题、填空题、问答题以及程序阅读题和程序设计题，用于读者配合学习内容练习。自测练习题主要是各种试题形式，也有部分是习题形式。这是作者在教学过程中，收集整理并编制的各类试题，近400道。
- (5) 自测练习解答——提供自测练习题的解答和必要的说明，供读者判断对错。

本书第6章提供了两套模拟试题。第1套比较简单，基本对应专科层次的教学要求；而第2套略有难度，基本对应本科层次的教学要求。

第7章是上机指导，主要为学生实际编写和调试程序提供指导意见。

本书没有按照某本教材的章节安排内容，而是遵循共同的要求，或者说是从掌握汇编语言程序设计的基本要求出发来编排的。本书在写作过程中参照了国内高等院校本科、专科的教学要求，以及全国和部分省市自学考试大纲。当然读者可以采用作者主编的教材（参考文献1和参考文献2），也可以使用其他同类教材。本书试图适应较广的范围，从简单入手，使其适合新高职和专科层次；主要面向自学考试和本专科教学；也有部分较高要求的试题，以便同时适合考研要求。

本书虽然冠以“学习与解题指南”，但作者希望读者不要陷入题海之中。

本书的目的是以重难点分析和题解的形式帮助读者更好地掌握汇编语言程序设计；或者说，更明确学习的重点和难点。虽然本书能够帮助读者应付各种考试，但作者并不赞同那种纯粹以考试为目的的学习，因此本书没有收录怪异和刁钻的试题。

另外，提醒各位读者，实际教学中，每位教师对重点或考试要求都可能各有各的看法，即使是全国各地的自学考试（有较详细的考试大纲）内容也不尽相同；当然，其中的基本要求还是明确的。所以，参加各类考试的学生需要清楚你的老师的要求。

钱晓捷

2002年2月

目 录

第 1 章 汇编语言基础知识	(1)
1.1 重点与难点	(1)
1.1.1 汇编语言及其特点	(1)
1.1.2 计算机系统的基本组成	(1)
1.1.3 数据表示	(2)
1.1.4 8086/8088 的寄存器组	(2)
1.1.5 8086/8088 的存储器组织	(2)
1.1.6 8086/8088 的标志	(3)
1.1.7 8086/8088 的寻址方式	(3)
1.2 例题精选	(4)
1.3 自测练习	(7)
1.4 自测练习解答	(11)
第 2 章 8086/8088 指令系统	(15)
2.1 重点与难点	(15)
2.1.1 数据传送类指令	(15)
2.1.2 算术运算类指令	(17)
2.1.3 位操作类指令	(18)
2.1.4 串操作类指令	(19)
2.1.5 控制转移类指令	(20)
2.1.6 处理器控制类指令	(22)
2.1.7 中断指令和系统功能调用	(22)
2.2 例题精选	(23)
2.3 自测练习	(30)
2.4 自测练习解答	(42)
第 3 章 汇编语言程序格式	(51)
3.1 重点与难点	(51)

3.1.1	汇编语言的语句	(51)
3.1.2	源程序的格式	(52)
3.1.3	常量的表达	(54)
3.1.4	变量的定义和使用	(55)
3.1.5	名字和标号的属性	(55)
3.2	例题精选	(55)
3.3	自测练习	(59)
3.4	自测练习解答	(69)
第 4 章	基本汇编语言程序设计	(77)
4.1	重点与难点	(77)
4.1.1	顺序结构	(77)
4.1.2	分支结构	(77)
4.1.3	循环结构	(78)
4.1.4	子程序	(78)
4.2	例题精选	(79)
4.3	自测练习	(89)
4.4	自测练习解答	(101)
第 5 章	高级汇编语言程序设计	(120)
5.1	重点与难点	(120)
5.1.1	宏汇编	(120)
5.1.2	重复汇编和条件汇编	(121)
5.1.3	模块化程序设计	(121)
5.1.4	输入/输出程序设计	(121)
5.2	例题精选	(121)
5.3	自测练习	(124)
5.4	自测练习解答	(128)
第 6 章	模拟试卷及解答	(135)
6.1	模拟试卷 1	(135)
6.2	模拟试卷 1 解答	(140)
6.3	模拟试卷 2	(144)
6.4	模拟试卷 2 解答	(148)

第 7 章 上机指导.....	(154)
7.1 上机环境.....	(154)
7.2 调试程序.....	(155)
7.2.1 用调试程序编写程序段.....	(155)
7.2.2 使用调试程序的注意事项.....	(156)
7.3 开发汇编语言程序.....	(158)
参考文献.....	(163)

第 1 章

汇编语言基础知识

概 述

本章介绍学习汇编语言程序设计所涉及到的基本知识，为展开指令系统和程序设计打下基础。对于已在先修课程中学习过相关知识的学生，这部分也可以不作要求。

为了化解难点，将微处理器的基本结构和寻址方式安排在这里，成为本章的重点。这部分内容同时也是学习指令系统的基础。

1.1 重点与难点

1.1.1 汇编语言及其特点

汇编语言以助记符形式表达处理器指令，是一种面向处理器的低级程序设计语言。利用汇编语言可以编写与计算机硬件密切相关的程序，有效地控制硬件；汇编语言程序产生的可执行程序容量小、效率高、运行速度快。相对于广泛应用的高级程序设计语言来说，利用汇编语言编写程序比较繁琐、调试比较困难，且功能不够强大。

1.1.2 计算机系统的基本组成

计算机系统分成硬件和软件两大部分。

计算机硬件有五大组成部件：运算器、控制器、存储器、输入设备和输出设备。作为计算机核心的中央处理单元 CPU 包含有运算器和控制器，外设则是输入和输出设备的统称，存储器分为主存储器和辅存储器。

计算机软件分成：系统软件和应用软件。

1.1.3 数据表示

十进制是我们习惯使用的数据进制，二进制（0、1）则是计算机内部采用的数据进制，十六进制（0~9、A~F）是二进制的缩略形式，它用一位表达二进制的 4 位。在用汇编语言进行程序设计时，三种进制数据间经常需要相互转换，尤其是整数部分。

对于有符号数据，计算机内部有原码、反码和补码表示方法。补码是计算机缺省使用的表达方式。

BCD 码是用二进制编码的十进制数，经常采用 8421 码：只使用 0000~1001 十个编码。ASCII 码是美国标准信息交换码，前 32 个和最后一个作为控制字符，其他 95 个编码表示可显示和打印的字符，包括数字和大小写英文字母。读者要熟悉其编码规律。

1.1.4 8086/8088 的寄存器组

8086/8088 具有 8 个 16 位通用寄存器：AX、BX、CX、DX、SI、DI、BP 和 SP；每个寄存器都还有各自的专用目的。其中前 4 个数据寄存器又可以分成 8 个 8 位通用寄存器：AH/AL、BH/BL、CH/CL 和 DH/DL。

8086/8088 具有 4 个 16 位段寄存器：CS、SS、DS 和 ES，依次指明代码段、堆栈段、数据段和附加段的首地址。每种逻辑段的作用各自不同。代码段内指令的偏移地址由指令指针 IP 指示，堆栈段的当前栈顶由堆栈指针 SP 指示，数据段和附加段内数据的偏移地址由寻址方式确定。

1.1.5 8086/8088 的存储器组织

8086/8088 的地址总线有 20 条信号线，可以寻址的主存容量是 1MB，这 1MB 主存空间被分段管理；每个逻辑段不超过 64KB，必须从低 4 位全为 0 的地址边界（被称为模 16 地址，或小段 Paragraph）开始。主存中每个存储单元具有一个唯一的编号，就是 20 位物理地址，这是微处理器输出给外部总线的地址信号。无论是在微处理器内部还是在进行程序设计时，都是采用分段的逻辑地址，表达为“段基地址：偏移地址”，将其转换为物理地址的方法是“段基地址左移 4 位加偏移地址”。

1.1.6 8086/8088 的标志

8086/8088 共有 9 个标志位, 保存在 16 位的标志寄存器(程序状态字 PSW) 中, 分成 6 个状态标志和 3 个控制标志。

状态标志用于提供指令执行结果的辅助信息, 有进位 CF、零位 ZF、符号 SF、奇偶 PF、溢出 OF 和辅助进位 AF 标志。其中前 5 个是我们学习的重点, 而溢出标志 OF 及其判断则是难点。

控制标志会影响处理器执行指令的方式, 有方向 DF、中断 IF 和单步 TF 标志, 依次控制串操作指令、能否响应可屏蔽中断、是否进入单步执行方式。

1.1.7 8086/8088 的寻址方式

指令分成操作码和操作数两部分。指令中用于说明操作数位置的方法就是操作数的寻址方式。

1. 立即(数)寻址方式

操作数存放在指令代码中, 紧跟着指令操作码; 这种操作数被称为立即数。在汇编语言中, 立即数用常量表示, 可以是二、十、十六、八进制常数, 可以是字符串, 还可以是数值表达式、符号常量。

2. 寄存器寻址方式

操作数存放在处理器内部的寄存器中。在汇编语言中, 直接利用寄存器名来表达操作数所在的寄存器; 8086/8088 中只有 8 个 8 位和 8 个 16 位通用寄存器、4 个段寄存器。

3. 存储器寻址方式

操作数存放在主存中。在汇编语言中, 用存储器地址表示操作数所在主存的位置。由于 8086/8088 分段管理主存, 所以要注意以下几点。

(1) 存储器地址采用逻辑地址(段地址: 偏移地址)表示。

(2) 段地址在段寄存器中, 采用 BP 寻址主存, 段地址默认在 SS 寄存器中, 即操作数在堆栈段; 其他情况寻址主存, 段地址默认在 DS 寄存器中, 即操作数在数据段。它们都可以利用段超越前缀指令, 改变段地址所在的段寄存器, 也即改变操作数所在的逻辑段。

(3) 偏移地址被称为有效地址 EA, 有多种形式, 可以统一表示成

有效地址 $EA = BX \mid BP + SI \mid DI + \text{位移量}$

该式的各种组合就产生了多种寻址方式：

- ① 直接寻址方式——只取位移量，指令中直接包含了操作数的偏移地址（有效地址）。
- ②（寄存器）间接寻址方式——只用寄存器，有效地址存放在寄存器中。
- ③（寄存器）相对寻址方式——采用寄存器与位移量的组合，有效地址是寄存器内容与位移量之和。
- ④ 基址变址寻址方式——利用基址寄存器（BX、BP）与变址寄存器（SI、DI），有效地址是基址寄存器内容与变址寄存器内容之和。
- ⑤ 相对基址变址寻址方式——使用全部 3 个组成部分，有效地址是基址寄存器、变址寄存器与位移量之和。

说明：在汇编语言中，直接寻址中的有效地址可用常量表达，也常用变量表达；相对寻址中的位移量可用常量表示，也可用符号表示。

1.2 例题精选

例 1-1 将十进制数据 -46 用 8 位(bit, 简称 b)的原码、反码和补码表示。

解 先将 46 转换为二进制数： $(46)_{10} = (2E)_{16} = (00101110)_2$

-46 用原码表示为： $-46 = (10101110)$ 原码

-46 用反码表示为： $-46 = (11010001)$ 反码

-46 用补码表示为： $-46 = (11010010)$ 补码

例 1-2 对于计算机内部的一个编码：01001001，它代表什么含义？

解 对于同一个编码，不同的使用场合将表达完全不同的含义，例如：

01001001 作为二进制整数，表示十进制的 73 ($2^6 + 2^3 + 2^0$)；

01001001 作为 BCD 码，表示十进制的 49；

01001001 作为 ASCII 码，表示字符 I。

例 1-3 已知一个 16 字节(Byte, 简称 B)的数组始于逻辑地址 F000 : 1070，请问最后一个字节所在的物理地址？

解 这是一个常见的逻辑地址转换为物理地址的例子：

物理地址 = 段地址左移 4 位(对十六进制数来说就是一位) + 偏移地址

第一个字节单元的物理地址是 $F000H + 1070H = F1070H$

最后一个字节单元的物理地址是 $F1070H + (16-1) = F107FH$

例 1-4 偏移地址 2000H~2003H 的 4 个字节单元依次存放如下数据 (十六进制): 10 20 30 40。请问偏移地址 2000H 处的字节、字和双字单元内容分别是什么?

解 Intel 80x86 系列微处理器采用“小端方式”存放多字节数据,即在内存中低地址存放低字节、高地址存放高字节,就是所谓“低对低、高对高”原则。因此,结论是:

[2000H] 字节单元 = 10H

[2000H] 字单元 = 2010H

[2000H] 双字单元 = 40302010H

例 1-5 8086/8088 的 1MB 主存空间最多能分成多少个逻辑段,最少分成多少个逻辑段?

解 逻辑段有两个限制:容量小于等于 64KB,段开始于模 16 地址。

如果从最低地址 00000H 开始,每隔 16 个字节就分一个段(最小的逻辑段容量),这样 1MB 主存空间最多就分成 $1\text{MB} \div 16 = 2^{20} \div 2^4 = 2^{16}$ 个逻辑段。

如果从最低地址 00000H 开始,每隔 64KB 分一个段(最大的逻辑段容量),这样 1MB 主存空间最少就分成 $1\text{MB} \div 64\text{KB} = 2^{20} \div 2^{16} = 2^4$ 个逻辑段。

例 1-6 请问两个 8 位数据 8FH 与 36H 相加后,状态标志 CF、ZF、SF、PF 和 OF 的状态分别是什么?

解 直接对这两个数据进行(无符号数据)加法运算:

$$8\text{FH} + 36\text{H} = \text{C5H}$$

进位标志针对无符号数据的加减运算。

- ① 最高位没有进位,所以 $\text{CF} = 0$;
- ② 结果 C5H 不是零,所以 $\text{ZF} = 0$;
- ③ 最高位 $\text{D}_7 = 1$,所以 $\text{SF} = 1$;
- ④ 结果 C5H 中“1”的个数是 4,为偶数,所以 $\text{PF} = 1$ 。

溢出标志针对有符号数据的加减运算。它的判断可以采用一个简单的规则:当两个相同符号的数据相加(含不同符号的数据相减),如果运算结果的符号与原数据符号不同,就产生溢出;其他情况,不会产生溢出。对于 8FH(负数)与 36H(正数)两个不同符号数据相加,根本不会产生溢出,所以 $\text{OF} = 0$ 。

如果 8FH 减去 36H,则结果是: $8\text{FH} - 36\text{H} = 59\text{H}$ 。没有借位,所以 $\text{CF} = 0$ 。“负数减正数”可以认为是“负数加负数”,结果不应是正数(59H),所以有溢出 $\text{OF} = 1$ 。

例 1-7 说明下列各条指令中,源操作数采用的寻址方式?

- ① `mov bx, 2002h`

- ② mov bx, dx
- ③ mov bx, [2002h]
- ④ mov bx, [si]
- ⑤ mov bx, [di+10h]
- ⑥ mov bx, [bp+si]
- ⑦ mov bx, [bp+di+100]

解 注意各种寻址方式的定义，并熟悉各种寻址方式的表达形式。

- ① 立即寻址方式，2002h 是 16 位立即数。
- ② 寄存器寻址方式，操作数在 DX 寄存器中。
- ③ 直接寻址方式，有效地址为 2002h。

注意区别立即数寻址方式和直接寻址方式。

- ④ 寄存器间接寻址方式，有效地址在 SI 寄存器中。

注意区别寄存器寻址方式和寄存器间接寻址方式。

- ⑤ 寄存器相对寻址方式，注意该指令也可以书写成：

```
mov bx, 10h[di].
```

- ⑥ 基址变址寻址方式，该指令也可以表达成：

```
mov bx, [bp][si].
```

- ⑦ 相对基址变址寻址方式，该指令还有其他书写形式：

```
mov bx, 100[bp][di]
```

```
mov bx, 100[bp+di]
```

例 1-8 假设有关寄存器及存储单元内容如下：

DS = 1000h, BX = 0100h, SI = 0002

[10100H] = 12h, [10101H] = 34h, [10102H] = 56h, [10103H] = 78h

[11200H] = A1h, [11201H] = B2h, [11202H] = C3h, [11203H] = D4h

说明如下各条指令执行完后，CX = ?

- ① mov cx, 1200
- ② mov cx, bx
- ③ mov cx, [1200h]
- ④ mov cx, [bx]
- ⑤ mov cx, 1100h[bx]
- ⑥ mov cx, [bx][si]
- ⑦ mov cx, 1100h[bx][si]

解 对后 5 种存储器寻址方式，需要首先计算它的有效地址，然后与段地址组合成物理地址，进而从存储单元取出内容（注意：这是 16 位数据）。

- A 10011011 B 01100100
C 10011101 D 10011100
8. 下列各个 8 位二进制数的补码中, 绝对值最大的是_____。
- A 10001000 B 11111110
C 00000100 D 00000001
9. 16 位有符号数的补码所表示的十进制数的范围是_____。
- A -32767~+32768 B -32768~+32767
C -65535~+65536 D 0~65535
10. 对数值 83A7H 作逻辑非运算后的结果是_____。
- A 83A8H B 73A8H
C 7C59H D 7C58H
11. 当设置标志 TF 为 1, 每条指令执行后 CPU 自动进入_____中断。
- A 断点中断 B 非屏蔽中断
C 单步中断 D 指令中断
12. 若用户堆栈位于存储区 10000H~1FFFFH, 则该堆栈的段地址是_____。
- A 10000H B 1FFFFH
C 01000H D 00FFFFH
13. 指令“mov cx,[bp+16]”的源操作数采用的段寄存器是_____。
- A CS B SS
C ES D ES
14. 指向程序堆栈区的段寄存器是_____。
- A CS B SS
C DS D ES
15. 指令“mov es:[2000h],ax”中, 目的操作数的寻址方式是_____。
- A 立即(数)寻址 B 寄存器寻址
C 直接寻址 D 寄存器间接寻址
16. 有效地址是指_____。
- A 存储器操作数的物理地址 B 存储器操作数的段地址
C 存储器操作数的偏移地址 D 立即数的偏移地址
17. 将 OBJ 文件转换为 EXE 可执行文件的是_____。
- A 汇编程序 B 连接程序
C 编辑程序 D 调试程序
18. 寄存器间接寻址方式中, 操作数在_____中。

