

工科课程提高与应试丛书

- 涵盖课程重点及难点
- 精设典型题详解及评注
- 选配课程考试模拟及全真试卷

马瑞芳 主编

汇编语言程序设计

典型题解析及自测试题



西北工业大学出版社

工科课程提高与应试丛书

汇编语言程序设计 典型题解析及自测试题

主 编 马瑞芳

编著者 王会燃

西北工业大学出版社

【内容简介】 本书是大专院校学生学习“汇编语言程序设计”课程的辅导教材,其中包括计算机基础知识、IBM PC 计算机组织、寻址方式和指令系统、汇编语言程序格式及上机过程、汇编语言程序设计、输入输出与中断程序设计、高级汇编语言技术、模块化程序设计技术、80X86 汇编语言程序设计及 CASL 汇编语言程序的编制。第一部分每章在提出基本要求,简要回顾基本知识点后,给出了典型题解析,最后提供了大量的习题。第二部分给出 3 套自测试题。附录提供了全部习题及自测试题答案。

本书既可作为本科生、专科生、自考者学习“汇编语言程序设计”课程的辅导教材,也可作为参加高级程序员考试人员的复习参考书。

图书在版编目(CIP)数据

汇编语言程序设计典型题解析及自测试题/马瑞芳主编. —西安:西北工业大学出版社,2002.3

(工科课程提高与应试丛书)

ISBN 7-5612-1450-2

I. 汇… II. 马… III. 汇编语言-程序设计-高等学校-解题
IV. TP313-44

中国版本图书馆 CIP 数据核字(2002)第 009905 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号 邮编:710072 电话:(029) 8493844

网 址: <http://www.nwpup.com>

印刷者:西安市向阳印刷厂

开 本:850 mm×1 168 mm 1/32

印 张:12.375

字 数:310 千字

版 次:2002 年 3 月第 1 版 2002 年 3 月第 1 次印刷

印 数:1~5 000 册

定 价:16.00 元

前 言

《汇编语言程序设计》课程是高等院校计算机应用专业本、专科生必修的核心课程。该课程不仅是计算机组成、操作系统等其他核心课程的必要先修课,而且对于训练学生掌握程序设计技术、熟悉上机操作和程序调试技术都有着重要作用。然而,汇编语言与机器内部结构关系密切,对于初学者来说感到概念抽象,难于掌握。本书通过提出基本要求、基本知识点和提供大量的典型题解析、习题、自测试题,把抽象的概念变成实际的操作,加快、加深学生对基本概念的理解和掌握。

本书共分三部分。第一部分包括计算机基础知识、IBM PC 计算机组织、寻址方式和指令系统、汇编语言程序格式及上机过程、汇编语言程序设计、输入输出与中断程序设计、高级汇编语言技术、模块化程序设计技术、80X86 汇编语言程序设计及 CASL 语言程序设计等十章内容。每章首先提出基本要求,使读者明确本章应达到的目标;其次归纳出基本知识点,简要回顾本章的主要知识;再次是典型题解析,对本章的重点、难点内容进行举例分析及详尽的讲解。最后列举了大量习题及自测试题,希望读者通过动手练习,进一步巩固、加深对基本知识和概念的理解。

第二部分给出 3 套全真自测试题,读者可在本课程学完后进行自测,以检查自己是否达到了课程要求。

第三部分附录提供了全部习题、自测试题的参考答案。书中所有程序,作者都亲自在微机上进行了调试运行。

11668 / 1203

本书第一、二、三、四、五、八、十章由马瑞芳编写,第六、七、九章由王会燃编写,全书由马瑞芳统稿。

由于编者的知识水平和时间所限,书中不妥之处在所难免,恳请广大读者提出宝贵意见。

E-mail: rf_ma@263.net

电 话: (029)8093675

马瑞芳

2001年9月于西安交通大学

目 录

第一部分 典型题解析

第一章 基础知识	1
一、基本要求	1
二、基本知识点	1
三、典型题解析	5
四、习题	16
第二章 IBM PC 计算机组织	18
一、基本要求	18
二、基本知识点	18
三、典型题解析	26
四、习题	28
第三章 寻址方式和指令系统	30
一、基本要求	30
二、基本知识点	30
三、典型题解析	41
四、习题	70
第四章 汇编语言程序格式及上机过程	74
一、基本要求	74

二、基本知识点	74
三、典型题解析	78
四、习题	92
第五章 汇编语言程序设计	96
一、基本要求	96
二、基本知识点	96
三、典型题解析	103
四、习题	135
第六章 输入输出与中断程序设计	140
一、基本要求	140
二、基本知识点	140
三、典型题解析	145
四、习题	173
第七章 高级汇编语言技术	174
一、基本要求	174
二、基本知识点	174
三、典型题解析	178
四、习题	190
第八章 模块化程序设计技术	191
一、基本要求	191
二、基本知识点	191
三、典型题解析	198
四、习题	211

第九章 80X86 汇编语言程序设计	213
一、基本要求	213
二、基本知识点	213
三、典型题解析	225
四、习题	237
第十章 CASL 汇编语言程序的编制	239
一、基本要求	239
二、基本知识点	239
三、典型题解析	248
四、习题	300

第二部分 自测试题

自测试题一	301
自测试题二	308
自测试题三	315

附录 习题与自测试题答案

一、习题答案	321
二、自测试题答案	365
参考文献	384

第一部分 典型题解析

第一章 基础知识

本章学习计算机基础知识及计算机中数据的表示与编码。

一、基本要求

通过本章的学习,应达到以下目标:

- (1) 掌握常用的各种进制数的表示、转换规则和运算;
- (2) 掌握带符号数的补码表示方法和补码的运算;
- (3) 熟悉符号扩展的概念;
- (4) 熟悉数据的表示范围;
- (5) 了解计算机存取信息的基本数据类型;
- (6) 了解计算机中字符的表示;
- (7) 了解汇编语言的基本概念。

二、基本知识点

(一) 计算机中常用的数制

计算机中常用的数制包括:二进制、十进制和十六进制数。

二进制数是以 2 为基的表示法,数由 2 个数字构成(0,1),一

个二进制数后缀为 B, 如 10110111B。

十进制数是以 10 为基的表示法, 数由 10 个数字构成(0~9), 一个十进制数后缀为 D, 如 2001D。在大多数情况下, 十进制数的后缀可缺省。

十六进制数是以 16 为基的表示法, 数由 16 个数字构成(0~9), A(10), B(11), C(12), D(13), E(14), F(15)), 一个十六进制数后缀为 H, 如 18ADH。

(二) 不同进制之间的数据转换

把一种进制数转换为另一种进制的数, 其实质是进行基数的转换。基数转换是依据两个有理数相等, 其整数部分与小数部分分别相等的原则。因此在转换时, 其整数部分与小数部分应分别进行转换, 将转换后的结果合并, 整数部分与小数部分之间用小数点隔开, 就得到相应的转换结果。

(1) 二进制数转换为十进制数的转换规则是“按权值相加”。也就是说, 只要把二进制数中数位是“1”的那些位的权值相加, 其和就是等效的十进制数。

(2) 十进制数转换为二进制数, 它的整数和小数部分要分别进行转换。转换结束后将整数转换结果写在左边, 小数转换结果写在右边, 中间点上小数点。

整数部分转换规则:将十进制整数用基数 2 连续去除, 直到商为 0 为止, 将每次除得的余数反向排列, 就可得到十进制数整数部分的转换结果。反向排列是指最后得到的余数排在前边, 作为结果的最高位, 最先得到的余数排在后边, 作为结果的最低位。

小数部分转换规则:将十进制数的小数部分用基数 2 连续去乘, 直到小数部分为 0 或达到要求的精度为止, 将每次所得的乘积的整数部分正向排列, 就可得到十进制小数的转换结果。正向排列是指最先得到的整数为结果的最高位, 最后得到的整数为结果

的最低位。

(3) 二进制数转换为(八进制)十六进制数。将二进制数以小数点为界,向左、向右分别按(3)4位一组划分,不足(3)4位的部分用“0”补足,将每一组数写成对应的(八进制)十六进制数,就可得到转换结果。

(三) 原码

1. 原码表示法

将数的真值形式中的正(负)号,用代码 0(1) 来表示,数值部分用二进制来表示。

2. 原码的特点

(1) “0”的原码有两种表示法。

$$[+0]_{\text{原}} = 00000000\text{B}, [-0]_{\text{原}} = 10000000\text{B}$$

(2) n 位二进制原码所能表示的数值范围为:

$$-(2^{n-1}-1) \sim (2^{n-1}-1)$$

(3) 原码表示一个数时,最高位为符号位。

符号位为 0 时,其后面的 $n-1$ 位为数值部分,这个数为正数。

符号位为 1 时,其后面的 $n-1$ 位为数值部分,这个数为负数。

(四) 补码

1. 补码表示法

正数的补码表示与正数的原码相同;负数的补码表示为将它的原码表示,除符号位以外,其余位按位取反且在最低位加 1 形成。

2. 补码的特点

(1) “0”的补码表示是惟一的。

$$[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000\text{B}$$

(2) n 位二进制补码所能表示的数值范围为:

$$-2^{n-1} \sim 2^{n-1} - 1$$

- (3) 补码表示一个数时,最高位为符号位。
 - (4) 补码运算时符号位无需单独处理。
 - (5) 采用补码运算时,减法可用加法来实现。
3. 补码的加法和减法规则

$$[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$$

$$[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$$

(五) 二进制编码

(1) ASCII 码: 计算机中的字符数据一般用 ASCII(American Standard Code for Information Interchange)码表示, 一个字符占存储器一个字节单元(8 位二进制代码)。

(2) BCD 码: BCD(Binary Code Decimal)码是一种二进制编码的十进制数。常用的 BCD 码是 8421BCD 码。它的每一位十进制数都分别用 4 位二进制数来表示。

BCD 码有 2 种格式:

1) 压缩格式的 BCD 码: 在一个字节内能表示 2 位十进制数。

例 82D=1000 0010BCD (D—表示十进制数, BCD—表示 BCD 码)

2) 非压缩格式的 BCD 码: 在一个字节内只表示 1 位十进制数的 BCD 码, 这个字节的低 4 位表示 8421BCD 码, 高 4 位无意义。

例 82D=××××1000××××0010BCD

(六) 机器语言

CPU 直接识别并遵照执行的指令称为机器指令。机器指令在形式上表现为二进制编码的目标程序。

机器语言是用二进制编码的机器指令的集合及一组使用机器

指令的规则。它是 CPU 能直接识别的惟一语言。由机器语言描述的程序称为目标程序。

机器语言的缺点:不能用人们熟悉的形式来描述计算机要执行的任务;用机器语言编写程序十分繁琐,极易出错;一旦出错,很难发现,调试困难。

(七) 汇编语言

1. 概念

汇编格式指令:用助记符、地址符号等符号表示的指令。

汇编语言:由汇编格式指令、伪指令的集合及表示、使用这些指令的一组规则组成。

汇编语言源程序:用汇编语言书写的程序。

汇编:把汇编语言源程序翻译成机器能够直接识别的目标程序的过程。

汇编程序:完成汇编任务的程序。

2. 汇编语言的优点

(1) 能编写出“时空”(运行速度快、目标程序短)效率高的程序;

(2) 它可直接、有效的操纵机器硬件系统。

3. 汇编语言的缺点

(1) 面向机器,与机器关系密切;

(2) 编写程序繁琐、调试程序困难;

(3) 维护、交流和移植程序困难。

三、典型题解析

例 1.1 以十进制数 666.66 为例说明十进制数的特点。

解 (1) 十进制数的基是 10。

十进制数有 10 个不同的数字符号, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9。

(2) 十进制数逢 10 进位。

十进制数的同一个数字符号在不同的位置(数的排列先后)代表的数值是不同的。例如 666.66 这个数, 小数点左边第 1 位数“6”位于个位, 它的值就是 6 本身, 小数点左边第 2 位数“6”位于 10 位, 它的值就是 $6 \times 10 = 60$, 小数点左边第 3 位数“6”位于百位, 它的值就是 $6 \times 100 = 600$; 而小数点右面第 1 位数“6”位于十分位, 它的值是 $6 \times 10^{-1} = 0.6$, 小数点右面第 2 位数“6”位于百分位, 它的值是 $6 \times 10^{-2} = 0.06, \dots$ 。所以这个数可以写成

$$666.66 = 6 \times 10^2 + 6 \times 10^1 + 6 \times 10^0 + 6 \times 10^{-1} + 6 \times 10^{-2}$$

任意一个十进制数 D(Decimal Number)都可以表示为

$$D = D_{n-1} \cdot 10^{n-1} + D_{n-2} \cdot 10^{n-2} + \dots + D_1 \cdot 10^1 + D_0 \cdot 10^0 + D_{-1} \cdot 10^{-1} + D_{-2} \cdot 10^{-2} + \dots + D_{-m} \cdot 10^{-m} = \sum_{i=-m}^{n-1} D_i \cdot 10^i$$

例 1.2 以二进制数 10110111.11 为例, 说明二进制数的特点。

答 二进制数有 0 和 1 两种取值, 用计算机内部的电路实现时, 比十进制数要方便得多。实际上在计算机电子线路中用 +5 V 表示 1, 用 0 V 表示 0, 若干位排列起来就可表示一个二进制数。若把 0 V~5 V 分成 10 段来表示十进制数, 可靠性随电压波动就会急剧下降, 即使电压稳定, 线路干扰也会降低可靠性。所以, 数据在计算机内部存储、运算及传送均采用二进制数。当把计算机中存放的数据及其运算结果呈现给桌面上的用户时, 系统的翻译转换程序将数据自动转换成各种用户容易理解的数据, 如十进制数值, 图像, 图表, 曲线等。

二进制数的特点:

(1) 具有两个不同的基本符号 0, 1。

(2) 逢二进一, 借一当二。

(3) 一般表达式 $B = \sum_{i=m}^{n-1} B_i 2^i$ 。

以二进制数 10110111.11 为例说明上述 3 个特点。这个二进制数中只有“0”和“1”两种符号,若给这个二进制加“1”,算式如下

$$\begin{array}{r}
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1. \ 1 \ 1 \\
 + \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1. \ 0 \ 0 \\
 \hline
 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0. \ 1 \ 1
 \end{array}$$

给这个二进制数加“1”,相当于给这个数加上“00000001.00”这个数,它是包括 8 位整数和 2 位小数的二进制数。从上述运算中可以看出运用的二进制加法规则如下

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \leftarrow \text{向高位进位为 1 (逢 2 进 1)}$$

对于任意两个二进制数相减,算式如下:

$$\begin{array}{r}
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1. \ 1 \ 1 \\
 - \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0. \ 1 \ 0 \\
 \hline
 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1. \ 0 \ 1
 \end{array}$$

从这个运算中可以看出运用的二进制减法规则如下:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$0 - 1 = 1 \leftarrow \text{向高位借 1, 借 1 当 2}$$

$$1 - 1 = 0$$

另外,二进制数 10110111.11 用如下方法表示。

$$\begin{aligned}
 (10110111.11)_2 &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 \\
 &\quad + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + \\
 &\quad 1 \times 2^{-2} + 0 \times 2^{-3}
 \end{aligned}$$

计算一下这个表达式的和值其和为 183.75, 就是相应的十进制数结果。

$$\text{即}(1011011.11)_2=(183.75)$$

例 1.3 十六进制数与二进制、十进制数之间的对应关系。

答 因为用二进制表示一个数, 与采用其他进制表示同样的数相比, 数位太多。用十六进制数表示一个数, 恰恰克服了这个缺点。十六进制数的 1 位可以表示二进制数的 4 位, 为此, 人们在书写计算机程序时常常采用十六进制数。

十六进制数的特点:

(1) 具有 16 个不同的基本符号 0~9, A~F。

(2) 逢 16 进一, 借一当 16。

(3) 一般表达式 $H = \sum_{i=-m}^{n-1} H_i 16^i$ 。

下面列出十进制数、二进制数和十六进制数的对应关系如表 1-1 所示。

表 1-1 十进制数、二进制数和十六进制数之间的对应关系

十进制数	二进制数	十六进制数	十进制数	二进制数	十六进制数
0	00000000	00	9	00001001	09
1	00000001	01	10	00001010	0A
2	00000010	02	11	00001011	0B
3	00000011	03	12	00001100	0C
4	00000100	04	13	00001101	0D
5	00000101	05	14	00001110	0E
6	00000110	06	15	00001111	0F
7	00000111	07	16	00010000	10
8	00001000	08			

汇编语言中数据的书写规则如下:

(1) 二进制数后缀字母 B(Binary Digit)。

(2) 十进制数后缀字母 D (Decimal Digit)。

(3) 十六进制数后缀字母 H(Hexadecimal Digit)。

(4) 八进制数后缀字母 Q(Octal Digit), 由于字母“O”容易与数字“0”相混淆, 所以八进制数后缀 Q。

说明: 通常十进制可不后缀 D。换一句话说, 后缀为 D 的数是十进制数, 不带后缀的数也是十进制数。一般情况下二进制数必须后缀 B, 十六进制数必须后缀 H, 八进制数后缀字母 Q。

例 1.4 将下列二进制数转换为十进制数。

(1) 101101.10B

(2) 111.111B

解 把二进制数位中的数值是 1 的那些位的权值相加, 其和就是等效的十进制数。由转换规则得

$$\begin{aligned} 101101.11\text{B} &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + \\ &\quad 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 41.75\text{D} \end{aligned}$$

$$\begin{aligned} 111.111\text{B} &= 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + \\ &\quad 1 \times 2^{-2} + 1 \times 2^{-3} = 7.875\text{D} \end{aligned}$$

二进制数转换为十进制数要求读者熟记表 1-2 所示的位—权表($i=-3 \sim 15$)。

例 1.5 将下列十进制数转换为二进制数。

(1) 19.75

(2) 7.875

解 (1) 19.75

• 整数部分转换(将十进制整数用基数 2 连续去除, 直到商为 0 为止, 将每次所得的余数反向排列。)