新世纪高职高专实用规划教材

• 计算机系列

汇编语言程序设计

HUIBIAN YUYAN CHENGXU SHEJI

杨永生　王立红　编著

清华大学出版社

新世纪高职高专实用规划教材　计算机系列

汇编语言程序设计

杨永生　王立红　编著

清华大学出版社

内 容 简 介

本教材以8086/8088指令系统为基础，阐述和讨论了计算机硬件编程模型。本书共有8章和4个附录。内容包括IBM PC系列兼容机的组成，8086/8088CPU的组成，存储器的组织及分段，8086/8088的指令系统及寻址方式等。同时还介绍了伪指令、汇编语言程序格式及汇编语言的上机过程。接着介绍了汇编语言中的顺序结构、分支结构和循环结构的程序设计基本方法，重点介绍了子程序结构和子程序的设计方法。最后还介绍了宏汇编、重复汇编及条件汇编的设计方法，汇编语言程序与C语言程序的连接技术，输入输出程序设计和中断程序设计方法等。全书所有的汇编语言程序例子都是在实际系统开发中的实例，对读者有非常重要的参考价值。

本书在编写中注重基础，精选内容，增大实例量，并以实例为模板，介绍程序的编写方法，以期使读者在基本理论、基本知识和基本技能方面得到训练。本书针对高职高专院校师生，可作为数控、计算机等相关专业的教材。

图书在版编目(CIP)数据

汇编语言程序设计/杨永生，王立红编著. —北京：清华大学出版社，2004.11
(新世纪高职高专实用规划教材　计算机系列)
ISBN 7-302-09810-7

Ⅰ.汇…　Ⅱ. ①杨…　②王…　Ⅲ.汇编语言—程序设计—高等学校：技术学校—教材　Ⅳ. TP313

中国版本图书馆CIP数据核字(2004)第110142号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　**邮　　编**：100084
社 总 机：010-62770175　**客户服务**：010-62776969
组稿编辑：林章波
文稿编辑：杨作梅
封面设计：陈刘源
印 刷 者：北京市清华园胶印厂
装 订 者：三河市金元装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：13.5　**字数**：315千字
版　　次：2004年11月第1版　2004年11月第1次印刷
书　　号：ISBN 7-302-09810-7/TP·6770
印　　数：1～5000
定　　价：18.00元

《新世纪高职高专实用规划教材》序

编写目的

目前，随着教育改革的不断深入，高等职业教育发展迅速，进入到一个新的历史阶段。学校规模之大，数量之众，专业设置之广，办学条件之好和招生人数之多，都大大超过了历史上任何一个时期。然而，作为高职院校核心建设项目之一的教材建设，却远远滞后于高等职业教育发展的步伐，以至于许多高职院校的学生缺乏适用的教材，这势必影响高职院校的教育质量，也不利于高职教育的进一步发展。

目前，高职教材建设面临着新的契机和挑战：

(1) 高等职业教育发展迅猛，相应教材在编写、出版等环节需要在保证质量之前提下加快步伐，跟上节奏。

(2) 新型人才的需求，对教材提出了更高要求，科学性、先进性和实用性充分体现。

(3) 高职高专教育自身的特点是强调学生的实践能力和动手能力，教材的取材和内容设置必须满足不断发展的教学需求，突出理论和实践的紧密结合。

(4) 新教材应充分考虑一线教师的教学需要和教学安排。

有鉴于此，清华大学出版社在相关主管部门的大力支持下，组织近百所高等职业技术学院的优秀教师以及相关行业的工程师，推出了一系列切合当前教育改革需要的高质量的面向就业的职业技术实用型教材。

系列教材

本系列教材主要涵盖以下领域：

- 计算机基础及其应用
- 计算机网络
- 计算机图形图像处理与多媒体
- 电子商务
- 计算机编程
- 电子电工
- 机械
- 数控技术及模具设计
- 土木建筑
- 经济与管理
- 金融与保险

另外，系列教材还包括大学英语、大学语文、高等数学、大学物理、大学生心理健康等基础教材。所有教材都有相关的配套用书，如实训教材、辅导教材、习题集等。

教材特点

为了完善高等职业技术教育的教材体系，全面提高学生的动手能力、实践能力和职业技术素质，特意聘请有实践经验的高级工程师参与系列教材的编写，采用了一线工程技术人员与在校教师联合编写的模式，使课堂教学与实际操作紧密结合。本系列丛书的特点如下：

(1) 打破以往教科书的编写套路，在兼顾基础知识的同时，强调实用性和可操作性。

(2) 突出概念和应用，相关课程配有上机指导及习题，帮助读者对所学内容进行总结和提高。

(3) 设计了“注意”、“提示”、“技巧”等带有醒目标记的特色段落，让读者更容易得到有益的提示与应用技巧。

(4) 增加了全新的、实用的内容和知识点，并采取由浅入深、循序渐进、层次清楚、步骤详尽的写作方式，突出实践技能和动手能力。

读者定位

本系列教材针对职业教育，主要面向高职高专院校，同时也适用于同等学历的职业教育和继续教育。本丛书以三年高职为主，同时也适用于两年制高职。

本系列教材的编写和出版是高职教育办学体制和动作体制改革下的产物，在后期的推广使用过程中将紧紧跟随职业技术教育发展的步伐，不断吸取新型办学模式，课程改革的思路和方法，为促进职业培训和继续教育的社会需求奉献自己的一份力。

我们希望，通过本系列教材的编写和推广应用，不仅有利于提高职业技术教育的整体水平，而且有助于加快改进职业技术教育的办学模式、课程体系和教学培训方法，形成具有特色的职业技术教育的新体系。

教材编委会

新世纪高职高专实用规划教材
·计算机系列编委会

前　言

本书是为高等教育学历文凭考试计算机及应用专业编写的。按照学历文凭教育的基本要求，本书在编写中注重基础，精选内容，增大实例量，并以实例为模板，介绍程序的编写方法，以期使读者在基本理论、基本知识和基本技能方面得到训练。

本课程是高等院校计算机硬件、软件及应用专业学生必修的核心课程之一。通过本课程的学习，为操作系统、微型机及其接口技术、计算机控制、单片机原理及应用等课程打好理论和编程基础。

汇编语言是计算机能提供给用户的最快而又最有效的语言，是能够利用计算机所有硬件特性并能直接控制硬件的语言，它适用于编写控制和使用计算机硬件和外部设备的系统程序，适用于编制计算机控制系统、仪器仪表和家用电器等的应用程序。

本教材以 8086/8088 指令系统为基础，阐述和讨论了计算机硬件编程模型。读者只要有一种高级语言程序设计作为基础，就可以通过学习本书掌握汇编语言程序设计技术。

本书共有 8 章。第 1 章为汇编语言所需的基础知识，已经学习过计算机基础课程的读者可以跳过这章。第 2 章介绍了 IBM PC 系列兼容机的组成，8086/8088CPU 的组成，存储器的组织及分段。第 3 章介绍了 8086/8088 的指令系统及寻址方式，并给出了各种指令的使用举例。第 4 章介绍了伪指令、汇编语言程序格式及汇编语言的上机过程。第 5 章介绍了顺序结构、分支结构和循环结构程序设计的基本方法。第 6 章介绍了子程序结构，子程序设计方法，宏汇编、重复汇编及条件汇编的设计方法，同时也介绍了汇编语言程序与高级语言程序的连接技术。第 7 章介绍了输入输出程序设计和中断程序设计方法。第 8 章为汇编语言程序在系统开发中的实例。

本书的第 6~8 章由杨永生副教授编写，第 1~5 章由王立红副教授编写。全书由杨永生副教授主审。

因编者经验不足，水平有限，缺点和错误之处，敬请广大读者批评指正，以待改进。

编　者

2004 年 8 月于西安

目　录

第1章 概 述

本章要点

- 进位计数制及不同基数的数之间的转换
- 计算机中数据的表示方式
- 基本的逻辑运算
- 机器语言的概念
- 汇编语言的概念及其重要性

现代计算机有数字电子计算机和模拟电子计算机两大类。目前大量使用的计算机属于数字电子计算机，它只能接受由0和1两种符号构成的二进制数。二进制数在书写时显得过于麻烦，所以在与计算机打交道时还经常使用八进制数、十进制数和十六进制数。同一个数值可以用不同的数制表示，不同数制之间可以相互转换。

1.1 进位计数制及不同基数的数之间的转换

1.1.1 进位计数制

凡是用数字符号排列，按由低位向高位进位计数的方法称为进位计数制，简称为计数制或进位制。在人们的日常生活中，会碰到各种不同的进位计数制，不仅有最常使用的十进制，还有二进制、八进制、十二进制、十六进制、二十四进制和六十进制等。在计算机中，数的表示仅采用二进制计数制。

数据无论采用什么进位制，都包含两个基本要素：基数(Radix)和位权(Weight)。

1. 基数

一种计数制允许选用基本数字符号的个数称为基数。

二进制有两个符号：0和1；基数为2。

八进制有8个符号：0，1，2，3，4，5，6，7；基数为8。

十进制中有10个符号：0，1，2，3，4，5，6，7，8，9；基数为10。

十六进制有16个符号：0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F；基数为16。

在基数为K的进位计数制中，必然包含K个不同的数字符号，每个数位计满K就向高位进1，即“逢K进1，借1当K”。

任意进制的各个计数符号是有顺序的，二进制、八进制和十进制的每个计数符号就是它的序号值，十六进制的后6个计数符A、B、C、D、E、F的序号值依次是10、11、12、

13、14、15。

2. 位权

如果把用 K 进制书写的一个整数从右往左依次记作第 0 位、第 1 位、…、第 n 位，则第 i 位上的数符 a_i 所代表的含义是 $a_i \times K^i$。在此，我们把 K^i 称为 K 进制数第 i 位的权。

例如，十进制数十分位、个位、十位、百位上的权依次是 10^{-1}、10^0、10^1、10^2，213.9 最高位上的 2 代表的数值是数字符号 2 乘以位权 10^2，而最低位上的数字 9 代表的数值是数字符号 9 乘以位权 10^{-1}。

十进制数每位的值等于该位的位权与该位数字符号值的乘积，一个十进制数就可以写成按权展开的多项式和的形式。例如，$213.9=2\times10^2+1\times10^1+3\times10^0+9\times10^{-1}$。对于任意一个十进制数 N，设整数部分有 n 位，小数部分有 m 位，于是可以写出一个十进制数的一般表达式如下：

$$N=K_{n-1}\times10^{n-1}+K_{n-2}\times10^{n-2}+\cdots+K_1\times10^1+K_0\times10^0+K_{-1}\times10^{-1}+K_{-2}\times10^{-2}+\cdots+K_{-m}\times10^{-m}$$

类似地，将一个 J 进制数 N_j 按权展开的多项式和的一般表达式为：

$$N_j=K_{n-1}\times J^{n-1}+K_{n-2}\times J^{n-2}+\cdots+K_1\times J^1+K_0\times J^0+K_{-1}\times J^{-1}+K_{-2}\times J^{-2}+\cdots+K_{-m}\times J^{-m} \quad (1.1)$$

由上可见，J 进制数相邻两个数位的权相差 J 倍，如果小数点向左移一位，数值缩小 J 倍，反之，如果小数点向右移一位，数值扩大 J 倍。

1.1.2 把非十进制数转换成十进制数

从式(1.1)中不难看出，用 J 进制数书写一个数，依照计数符号的顺序把 J 进制的一个符号 K_i 转换成一个十进制数，然后把式(1.1)的右边按照十进制运算规则进行运算，求得的结果就是相应的十进制数。

【例 1.1】将二进制数 110010.1_2 转换成十进制数。

解：$110010.1_2=1\times2^5+1\times2^4+0\times2^3+0\times2^2+1\times2^1+0\times2^0+1\times2^{-1}$

$=32+16+2+0.5$

$=50.5$

【例 1.2】将八进制数 27.4_8 转换成十进制数。

解：$27.4_8=2\times8^1+7\times8^0+4\times8^{-1}$

$=16+7+0.5$

$=23.5$

【例 1.3】将十二进制数 $2B.6_{12}$ 转换成十进制数。

解：$2B.6_{12}=2\times12^1+11\times12^0+6\times12^{-1}$

$=24+11+0.5$

$=35.5$

【例 1.4】将十六进制数 $4F.8_{16}$ 转换成十进制数。

解：$4F.8_{16}=4\times16^1+15\times16^0+8\times16^{-1}$

$=64+15+0.5$

$=79.5$

1.1.3　把十进制数转换成非十进制数的通用方法

十进制数转换成非十进制数时，整数部分和小数部分换算算法不同，需要分别进行。整数部分用除基数取余数法转换，小数部分用乘基数取整数法转换。

1. 除基数取余数法

除基数取余数法是十进制整数转换成非十进制整数的方法。需要转换的整数除以目的进制的基数，取其商的余数就是目的进制数最低位的系数 K_0，将商的整数部分继续除以目的进制的基数，取其商的余数就是目的进制数次低位的系数 K_1，…，这样逐次相除直到商为 0，即得到从低位到高位的余数序列，构成对应的某进制整数。

2. 乘基数取整数法

乘基数取整数法是十进制小数转换成非十进制小数的方法。需要转换的小数乘以目的进制的基数，取其积的整数部分作为对应目的进制小数的最高位系数 K_{-1}，将积的小数部分继续乘以目的进制的基数，新得到的积的整数部分作为目的进制小数的次高位系数 K_{-2}，…，这样逐次相乘，即得到从高位到低位积的整数序列，便构成对应的某进制小数。

【例 1.5】把十进制数 137 转换成二进制数。

解：

十进制整数	余数	系数 K_i	位
2 | 137			
2 | 68	1	K_0	最低位
2 | 34	0	K_1	
2 | 17	0	K_2	
2 | 8	1	K_3	
2 | 4	0	K_4	
2 | 2	0	K_5	
2 | 1	1	K_6	最高位
0			

转换结果：$137=1001001_2$

【例 1.6】把十进制数 233 转换成八进制数。

解：

十进制整数	余数	系数 K_i	位
8 | 233			
8 | 29	1	K_0	最低位
8 | 3	5	K_1	
0	3	K_2	最高位

转换结果：$233=351_8$

【例 1.7】把十进制数 15370 转换成十六进制数。

解：

十进制数	余数	系数 K_i	位
16 | 15370			
16 | 960	10(A)	K_0	最低位
16 | 60	0	K_1	
16 | 3	12(C)	K_2	
0	3	K_3	最高位

转换结果：$15370=3C0A_{16}$

【例 1.8】把十进制数 218.8125 转换成二进制数。

解：

整数部分	余数	系数	小数部分	整数	系数
2 | 218			0.8125		
2 | 109	0	K_0	× 2		
2 | 54	1	K_1	1.6250	1	K_{-1}
2 | 27	0	K_2	0.6250		
2 | 13	1	K_3	× 2		
2 | 6	1	K_4	1.2500	1	K_{-2}
2 | 3	0	K_5	0.2500		
2 | 1	1	K_6	× 2		
0	1	K_7	0.5000	0	K_{-3}
			× 2		
			1.0000	1	K_{-4}

转换结果：$218.8125=11011010.1101_2$

1.1.4 二进制数与八进制、十六进制数的相互转换

在十进制以外的其他数制转换之间，如果要把一个 K 进制数转换成 R 进制数，一般是以十进制作为中间媒介。先把 K 进制数转换成十进制数，再把等值的十进制数转换成 R 进制数。但是，在二进制数和八进制、十六进制数之间有非常简便的转换方法。

1. 二进制数与八进制数的相互转换

(1) 二进制数转换成八进制数的具体方法

以小数点为分界线，整数部分从低位到高位，小数部分从高位到低位，每 3 位二进制数分成一组，不足 3 位的，小数部分在低位补 0，整数部分在高位补 0；把每组 3 位的二进制数转换成 1 位的八进制数，这就是一个相应的八进制数的表示；采用八进制书写二进制，位数约减少到原来的 1/3。

【例 1.9】把二进制数 110110.1011_2 转换成八进制数。

解：　$110110.1011_2=\underline{110}\quad\underline{110}.\quad\underline{101}\quad\underline{100}_2$

$\qquad\qquad\qquad\quad\downarrow\quad\downarrow\quad\downarrow\quad\downarrow$

$\qquad\qquad\qquad\quad 6\quad 6\quad 5\quad 4$

$\qquad\qquad\quad =66.54_8$

(2) 八进制数转换成二进制数的具体方法

八进制数中的 1 位数字对应书写成 3 位的二进制数，即构成二进制数。

【例 1.10】把八进制数 36.248 转换成二进制数。

解：　$36.24_8=3\quad 6.\quad 2\quad 4_8$

$\qquad\qquad\downarrow\quad\downarrow\quad\downarrow\quad\downarrow$

$\qquad\quad 011\quad 110\quad .101\quad 100$

$\qquad\quad =011110.101100_2$

2. 二进制数与十六进制数的相互转换

二进制数与十六进制数的相互转换方法类似二进制数与八进制数的转换方法。采用十六进制书写二进制数，位数可以减少到原来的 1/4。

【例 1.11】把二进制数 10110011.01011110_2 转换成十六进制数。

解：　$10110011.01011110_2=1011\quad 0011.\quad 0101\quad 1110_2$

$\qquad\qquad\qquad\qquad\downarrow\quad\downarrow\quad\downarrow\quad\downarrow$

$\qquad\qquad\qquad\qquad B\quad 3\ .\quad 5\quad E$

$\qquad\qquad\qquad =B3.5E_{16}$

十六进制数转换成二进制数时，把每位十六进制数符写成相应的 4 位二进制数，即构成二进制数。

【例 1.12】把十六进制数 $3BD.A5_{16}$ 转换成二进制数。

解：　$3BD.A5_{16}=3\quad B\quad D.\quad A\quad 5_{16}$

$\qquad\qquad\downarrow\quad\downarrow\quad\downarrow\quad\downarrow\quad\downarrow$

$\qquad 0011\quad 1011\quad 1101\quad 1010\quad 0101$

$\qquad =001110111101.10100101_2$

表 1.1 中列出了 0～15 之间的十进制数在二进制、八进制和十六进制下的对应值。为了加快数制转换的速度，这张表中的内容应该熟记于心。

表 1.1　0～15 在各数制下的表示

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0000	00	0	8	1000	10	8
1	0001	01	1	9	1001	11	9
2	0010	02	2	10	1010	12	A
3	0011	03	3	11	1011	13	B
4	0100	04	4	12	1100	14	C
5	0101	05	5	13	1101	15	D
6	0110	06	6	14	1110	16	E
7	0111	07	7	15	1111	17	F

1.1.5 数的书写方法

由于计算机中经常使用的数制有二进制、八进制、十进制和十六进制，所以单独写出一个数无法分辨到底使用的是哪一种计数制，如 1011。为此，需要在书写形式上加以区分。一般来说，书写方法有 3 种：下标法、前导法和后缀法。

1. 下标法

这是日常书写的一种方法，是在写完表示数值的符号序列之后，在其右下角以角标的形势写上所使用的数制的基数，比如：

1001_2 表示二进制数 1001。

377_8 表示八进制数 377。

377_{16} 表示十六进制数 377。

对于日常使用的十进制数，可以按上述方法加下标表示，也可以不加；反之，如果一个数在书写时没有加上任何标记，则表示是十进制数。

2. 前导法

有些程序设计语言中使用前导法，即在表示数值的数符序列前面加上特定的前导符号以区分不同的数制。比如，在 C 语言中就用 0x 作为十六进制数的前导符号，在 Turbo Pascal 中用$作为十六进制数的前导符号。十六进制数 123 在 C 语言中写作 0x123，在 Turbo Pascal 中则写作$123。

3. 后缀法

与前导法相反的一种做法是，写完表示数值的数符序列后，加上一个特殊的符号表示不同的数制。汇编语言中就采取这种做法。二进制、八进制、十进制和十六进制的后缀符号分别是 B、Q 或 O、D 和 H。比如：

1011B 表示二进制数 1011。

1011H 表示十六进制数 1011。

汇编程序默认的数为十进制数，因而在源程序中出现的没有特别指定的数均看作十进制数，而对其他基数表示的数，应专门加以标记，如二进制数后加字母 B，八进制数后面加 O 或 Q，十进制数后可以加 D 也可以不加，十六进制数后加字母 H，但这种默认的基数可以用 RADIX 伪指令改变。特别需要指明的是，如果一个十六进制数以字母数符开头，会与后面章节中所说的变量、标号等标示符相混淆，为了区分这种情况，在所有的字母数符开头的前面加上一个 0。比如，十六进制数 ABC 必须写成 0ABCH。

1.2 二进制数和十六进制数运算

1.2.1 二进制数的运算

加法规则：0+0=0　　0+1=1　　1+0=1　　1+1=0(进位 1)

减法规则：0–0=0　0–1=1(借位 1)　1–0=1　1–1=0
乘法规则：0×0=0　0×1=0　1×0=0　1×1=1
除法规则：0×1=0　1÷1=1

【例 1.13】

```
  1011 1101 0100 0110B
+ 0101 1111 0001 1010B
-------------------
1 0001 1100 0110 0000B
```

【例 1.14】

```
  1001 0000   1111   1100B
- 0111 1110   0011   1001B
-------------------------
  0001 0010   1100   0011B
```

【例 1.15】

```
                1  1  1  0B
              ×1  0  0  1B
-----------------------------
                1  1  1  0
          0  0  0  0
       0  0  0  0
  +1   1  1  0
-----------------------------
  1  1  1  1  1  1  0B
```

【例 1.16】

```
          10001
      ------------
10110 | 101110110
      -10110
      ------
        10110
       -10110
       ------
            0
```

1.2.2　十六进制数的运算

按照逢十六进一，借一当十六的规则，直接用十六进制数计算。也可以采用先把该十六进制转换数成十进制数，经过计算后再把结果转换成为十六进制数。

【例 1.17】

```
  05 C3H
+3DF2H
-------
  43 B5H
```

【例 1.18】

```
  3D25H
 -05C3H
-------
  3762H
```