普通高等教育“九五”国家教委重点教材

清华大学计算机系列教材

IBM-PC 汇编语言程序设计

第2版

沈美明
温冬婵 编著

清华大学出版社
http://www.tup.tsinghua.edu.cn

清华大学计算机系列教材

普通高等教育“九五”国家教委重点教材
清华大学计算机系列教材

IBM-PC 汇编语言程序设计

（第 2 版）

沈美明　温冬婵　编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书主要阐述 80x86 汇编语言程序设计方法和技术。全书共分四部分：第 1 章和第 2 章为基础知识部分；第 3 章和第 4 章为编程工具部分，主要内容为 80x86 的指令系统与寻址方式，以及包括伪操作在内的汇编语言程序格式；第 5 章～第 9 章说明编程方法，内容包括循环、分支、子程序等基本程序结构，程序设计的基本方法和技术，多模块连接技术，宏汇编技术，以中断为主的输入输出程序设计方法，以及 BIOS 和 DOS 系统功能调用；第 10 章和第 11 章为实际应用部分，说明图形显示、发声和磁盘文件存取技术。全书提供了大量程序实例，每章后均附有习题。

本书是在 1991 年第 1 版的基础上，融会了 10 年来教学与科研的新成果改编成的。其第 1 版曾先后获得过原电子工业部工科电子类专业优秀教材一等奖、教育部科技进步一等奖、国家科技进步三等奖等；受到广大读者的欢迎，累计发行达 130 多万册。

本书适用于高等院校以及大、中专院校作为"汇编语言程序设计"课程的教材。本书也适于初学者使用，只要具有一种高级语言程序设计基础的读者，都可通过学习本书掌握汇编语言程序设计技术。

书　名：IBM-PC 汇编语言程序设计(第 2 版)
作　者：沈美明　温冬婵　编著
出版者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
　　http://www.tup.tsinghua.edu.cn
印刷者：世界知识印刷厂
发行者：新华书店总店北京发行所
开　本：787×1092 1/16　印张：30.75　字数：711 千字
版　次：2001 年 8 月第 1 版　2001 年 8 月第 1 次印刷
书　号：ISBN 7-302-04664-6/TP・2769
印　数：0001～8000
定　价：34.80 元

再 版 前 言

汇编语言是计算机能够提供给用户使用的最快而又最有效的语言，也是能够利用计算机所有硬件特性并能直接控制硬件的惟一语言。因而，对程序的空间和时间要求很高的场合，汇编语言的应用是必不可少的。至于很多需要直接控制硬件的应用场合，则更是非用汇编语言不可了。

《IBM-PC 汇编语言程序设计》(第 2 版)是高等院校计算机科学与技术专业"汇编语言程序设计"必修课所用教材。它的第 1 版(1991 年版)曾被评为 1992 年第四届全国科技类优秀畅销书；获得 1996 年电子工业部第三届工科电子类专业优秀教材一等奖；1999 年教育部科技进步一等奖；以及 1999 年国家科技进步三等奖。

在本书的第一版中，我们选用了以 8086 为 CPU 的 PC 机作为基础机型来组织教学。第 2 版是在第 1 版的基础上增加了有关当今计算机技术发展的新内容，其中包括 8086 后继机型(80x86)所提供的指令及寻址方式；汇编程序 MASM 新版本所提供的伪操作及高级汇编语言技术；保护模式的编程基础等，以便满足广大读者使用高档微机的需要。由于第 2 版内容的进一步充实，因此书的篇幅也有明显增加。为照顾不同需求，本书对第 1 版中某些部分作了删减，以便突出重点，达到课程的基本要求。本书可供高等院校及大、中专院校作为"汇编语言程序设计"课程的教材使用。但本书也适合于初学者使用，只要有一种高级语言程序设计基础，都可以通过学习本书掌握汇编语言程序设计技术。

全书由"基础理论、编程工具、编程方法和实际应用"四部分(共 11 章)组成。第 1 章和第 2 章为基础理论部分。包括数制、码制等基础知识，计算机组成及基本原理。第 3 章和第 4 章介绍编程工具。包括指令系统、寻址方式、伪操作和汇编语言格式。第 5 章～第 9 章说明编程方法。包括循环、分支、子程序等基本程序结构，宏汇编技术，中断等输入输出程序设计方法，BIOS 和 DOS 系统功能调用方法，以及多个模块的连接技术。第 10 章和第 11 章为实际应用。包括图形显示、发声和磁盘文件存取技术。这四个组成部分构成一个完整的系统。书中提供了大量程序例题，每章之后均附有若干习题，便于读者复习及检查学习效果。同时为了能适应各种类型院校的不同要求，各章之间相互配合而又自成体系，易于为不同类型院校按其要求适当加以裁剪，所以本教材的适用面比较宽。

本书为清华大学计算机科学与技术系"汇编语言程序设计"课程的教材。该课程课内 80 学时，其中讲课 48 学时，上机实践 32 学时，课内外学时比例为 1 ∶ 1.5。讲课内容为第 1 章～第 9 章，第 10 章和第 11 章则结合实验由学生自学并上机。选用本教材的各校均可根据教学计划规定的学时灵活安排。为便于查阅，本书把指令系统集中在第 3 章，因此所占篇幅较大。在讲课过程中，为使学生尽可能早些上机，开始编程训练，可把有关指令分散到其后各章讲述。例如，把转移类指令放在循环与分支程序设计一章，把转子与返回

指令放在子程序结构一章，把中断指令放在输入输出程序设计一章，等等。课程的上机安排可参考与本书配套的《IBM-PC 汇编语言程序设计实验教程》一书，根据课程上机时数及学生的水平，选用相应的实验。

本书的第 1 章～第 7 章由沈美明编写，第 8 章～第 11 章由温冬婵编写。书中如有错误或不当之处，欢迎读者不吝批评指正。

编著者

2001 年 8 月

目　　录

第1章 基础知识

1.1 进位记数制与不同基数的数之间的转换

1.1.1 二进制数

进位记数制是一种计数的方法，习惯上最常用的是十进制记数法。一个任意的十进制可以表示为：

$$a_n a_{n-1}\cdots a_0 \cdot b_1 b_2 \cdots b_m$$

其含义是

$$a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \cdots + a_i \cdot 10^i + \cdots + a_0 \cdot 10^0 + b_1 \cdot 10^{-1} + b_2 \cdot 10^{-2} + \cdots + b_j \cdot 10^{-j} + \cdots + b_m \cdot 10^{-m}$$

其中 $a_i(i=0,1,\cdots,n)$，$b_j(j=1,2,\cdots,m)$是 0,1,2,3,4,5,6,7,8,9 十个数码中的一个。

十进制数的基数为 10，即其数码的个数为 10，且遵循逢十进一的规则。上式中相应于每位数字的 10^k称为该位数字的权，所以每位数字乘以其权所得到的乘积之和即为所表示数的值。例如：

$$12345.67 = 1\times10^4 + 2\times10^3 + 3\times10^2 + 4\times10^1 + 5\times10^0 + 6\times10^{-1} + 7\times10^{-2}$$

十进制数是人们最熟悉、最常用的一种数制，但它不是惟一的数制。例如，计时用的时、分、秒就是按 60 进制计数的。基数为 r 的 r 进制数的值可表示为：

$$a_n \cdot r^n + a_{n-1} \cdot r^{n-1} + \cdots + a_0 \cdot r^0 + b_1 \cdot r^{-1} + b_2 \cdot r^{-2} + \cdots + b_m \cdot r^{-m}$$

其中 a_i，b_j可以是 0,1,…,r−1 中的任一个数码，r^k则为各位数相应的权。

计算机中为便于存储及计算的物理实现，采用了二进制数。二进制数的基数为 2，只有 0，1 两个数码，并遵循逢 2 进 1 的规则，它的各位权是以 2^k表示的，因此二进制数 $a_n a_{n-1}\cdots a_0 \cdot b_1 b_2 \cdots b_m$的值是：

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_0 \cdot 2^0 + b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \cdots + b_m \cdot 2^{-m}$$

其中 a_i，b_j为 0,1 两个数码中的一个。例如：

$$101101_2 = 1\times2^5 + 1\times2^3 + 1\times2^2 + 1\times2^0 = 45_{10}$$

其中数的下角标表示该数的基数 r，即二进制的 101101 与十进制的 45 等值。

n 位二进制数可以表示 2^n个数。例如 3 位二进制数可以表示 8 个数，它们是：

二进制数	000	001	010	011	100	101	110	111
相应的十进制数	0	1	2	3	4	5	6	7

而 4 位二进制数则表示十进制的 0～15 共 16 个数，如下所示：

二进制数	0000	0001	0010	0011	0100	0101	0110	0111
相应的十进制数	0	1	2	3	4	5	6	7
二进制数	1000	1001	1010	1011	1100	1101	1110	1111
相应的十进制数	8	9	10	11	12	13	14	15

为便于人们阅读及书写，经常使用八进制数或十六进制数来表示二进制数。它们的基数和数码如表 1.1 所示。

表 1.1　几种常用的进位计数制的基数和数码

进位计数制	基　数	数　码
十六进制数	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
十进制数	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
八进制数	8	0, 1, 2, 3, 4, 5, 6, 7
二进制数	2	0, 1

按同样的方法，读者可以很容易地掌握八进制和十六进制数的表示方法。可以看出，23_{10}可以表示为17_{16}、27_8及10111_2，1.375_{10}可以表示为1.6_{16}、1.3_8及1.011_2等。在计算机里，通常用数字后面跟一个英文字母来表示该数的数制。十进制数一般用 D(decimal)、二进制数用 B(binary)、八进制数用 O(octal)、十六进制数用 H(hexadecimal)来表示。例如：117D，1110101B，0075H，…。当然也可以用这些字母的小写形式，本书的后面就采用了这种表示方法。

1.1.2　二进制数和十进制数之间的转换

1. 二进制数转换为十进制数

各位二进制数码乘以与其对应的权之和即为该二进制数相对应的十进制数。例如：

$$1011100.10111B = 2^6+2^4+2^3+2^2+2^{-1}+2^{-3}+2^{-4}+2^{-5} = 92.71875D$$

2. 十进制数转换为二进制数

十进制数转换为二进制数的方法很多，这里只说明比较简单的降幂法及除法两种。

(1) 降幂法

首先写出要转换的十进制数，其次写出所有小于此数的各位二进制权值，然后用要转换的十进制数减去与它最相近的二进制权值，如够减则减去并在相应位记以 1；如不够减则在相应位记以 0 并跳过此位；如此不断反复，直到该数为 0 为止。

例 1.1　N=117D，小于 N 的二进制权为：

	64	32	16	8	4	2	1
对应的二进制数是	1	1	1	0	1	0	1

计算过程如下：

$$117-2^6=117-64=53 \qquad (a_6=1)$$
$$53-2^5=53-32=21 \qquad (a_5=1)$$
$$21-2^4=21-16=5 \qquad (a_4=1)$$
$$(a_3=0)$$
$$5-2^2=5-4=1 \qquad (a_2=1)$$
$$(a_1=0)$$
$$1-2^0=1-1=0 \qquad (a_0=1)$$

所以 N=117D=1110101B。

例 1.2 N=0.8125D,小于此数的二进制权为：

0.5	0.25	0.125	0.0625
1	1	0	1

对应的二进制数是

计算过程如下：

$$0.8125-2^{-1}=0.8125-0.5=0.3125 \qquad (b_1=1)$$
$$0.3125-2^{-2}=0.3125-0.25=0.0625 \qquad (b_2=1)$$
$$(b_3=0)$$
$$0.0625-2^{-4}=0.0625-0.0625=0 \qquad (b_4=1)$$

所以 N = 0.8125D = 0.1101B。

(2) 除法

把要转换的十进制数的整数部分不断除以 2,并记下余数,直到商为 0 为止。

例 1.3 N =117D

$$117/2=58 \qquad (a_0=1)$$
$$58/2=29 \qquad (a_1=0)$$
$$29/2=14 \qquad (a_2=1)$$
$$14/2=7 \qquad (a_3=0)$$
$$7/2=3 \qquad (a_4=1)$$
$$3/2=1 \qquad (a_5=1)$$
$$1/2=0 \qquad (a_6=1)$$

所以 N =117D =1110101B。

对于被转换的十进制数的小数部分则应不断乘以 2,并记下其整数部分,直到结果的小数部分为 0 为止。

例 1.4 N=0.8125D

$$0.8125\times2=1.625 \qquad (b_1=1)$$

$$0.625\times2=1.25 \quad (b_2=1)$$

$$0.25\times2=0.5 \quad (b_3=0)$$

$$0.5\times2=1.0 \quad (b_4=1)$$

所以 N=0.8125D=0.1101B。

1.1.3 十六进制数及其与二进制、十进制数之间的转换

我们知道，在计算机内部，数的运算和存储都是采用二进制的。但是，二进制数对于人们阅读、书写及记忆都是很不方便的。十进制数虽然是人们最熟悉的一种进位计数制，但它与二进制数之间并无直接的对应关系。为了便于人们对二进制数的描述，应该选择一种易于与二进制数相互转换的数制。显然，使用 2^n 作为基数的数制是能适合人们的这种要求的，常用的有八进制数和十六进制数，这里主要介绍十六进制数。

1. 十六进制数的表示

计算机中存储信息的基本单位为一个二进制位(bit)，它可以用来表示 0 和 1 两个数码。此外，由于计算机中常用的字符是采用由 8 位二进制数组成的一个字节(byte)来表示的，因此字节也成为计算机中存储信息的单位。计算机的字长一般都选为字节的整数倍，如 16 位、32 位、64 位等。一个字节由 8 位组成，它可以用两个四位组(又称半字节)来表示，所以用十六进制数来表示二进制数是比较方便的。

十六进制数的基数是 16，共有 16 个数码，它们是 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F。其中 A 表示十进制的 10，余类推。它们与二进制和十进制数的关系如下：

二进制数	十进制数	十六进制数
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

对应于十六进制数中各位的权是 16^k。

2. 十六进制数和二进制数之间的转换

由于十六进制数的基数是 2 的幂，所以这两种数制之间的转换是十分容易的。一个二进制数，只要把它从低位到高位每 4 位组成一组，直接用十六进制数来表示就可以了。

例 1.5

0 011	0 101	1 011	1 111
3	5	B	F

亦即 0011010110111111B=35BFH

反之，把十六进制数中的每一位用 4 位二进制数表示，就形成相应的二进制数了。

例 1.6

A	1	9	C
1010	0001	1001	1100

亦即 A19CH=1010000110011100B

3. 十六进制数和十进制数之间的转换

各位十六进制数与其对应权值的乘积之和即为与此十六进制数相对应的十进制数。

例 1.7

$$\begin{aligned} N &= BF3CH \\ &= 11\times16^3+15\times16^2+3\times16^1+12\times16^0 \\ &= 11\times4096+15\times256+3\times16+12\times1 \\ &= 48956D \end{aligned}$$

十进制数转换为十六进制数也可使用降幂法和除法。

(1) 降幂法

首先写出要转换的十进制数，其次写出小于该数的十六进制权值，然后找出该数中包含多少个最接近它的权值的倍数，这一倍数即对应位的值，用原数减去此倍数与相应位权值的乘积得到一个差值，再用此差值去找低一位的权值的倍数，如此反复直到差值为 0 为止。

例 1.8 N=48956D

小于 N 的十六进制权值为	4096	256	16	1
对应的十六进制数	B	F	3	C

计算过程如下：

$$\begin{aligned} 48956-11\times4096&=3900 \\ 3900-15\times256&=60 \\ 60-3\times16&=12 \\ 12-12\times1&=0 \end{aligned}$$

所以

$$\begin{aligned} N &= 48956D=(11)(15)(3)(12) \\ &= BF3CH \end{aligned}$$

(2) 除法

把要转换的十进制数的整数部分不断除以 16，并记下余数，直到商为 0 为止。

例 1.9 N=48956D

$$48956/16=3059 \qquad (a_0=12)$$

$$3059/16=191 \qquad (a_1=3)$$

$$191/16=11 \qquad (a_2=15)$$
$$11/16=0 \qquad (a_3=11)$$

所以 N＝48956D＝BF3CH。

对于要转换的十进制数的小数部分，则应不断地乘以 16，并记下其整数部分，直到结果的小数部分为 0 为止。由于其方法与二、十进制数的转换方法是相同的，这里不再举例说明。显然，为把一个十进制数转换为二进制数，可以先把该数转换为十六进制数，然后再转换为二进制数，这样可以减少计算次数；反之，要把一个二进制数转换为十进制数，也可采用同样的办法。

1.2 二进制数和十六进制数运算

1.2.1 二进制数运算

加法规则：

0＋0＝ 0
0＋1＝ 1
1＋0＝ 1
1＋1＝ 0(进位 1)

乘法规则：

0×0＝ 0
0×1＝ 0
1×0＝ 0
1×1＝ 1

1.2.2 十六进制数运算

十六进制数的运算可以采用先把该十六进制数转换为十进制数，经过计算后再把结果转换为十六进制数的方法，但这样做比较繁琐。其实，只要按照逢十六进一的规则，直接用十六进制数来计算也是很方便的。

十六进制加法：

当两个一位数之和 S 小于 16 时，与十进制数同样处理；如果两个一位数之和大于或等于 16 时，则应该用 S－16 及进位 1 来取代 S。

例 1.10

$$\begin{array}{rr} & 05C3H \\ + & 3D25H \\ \hline & 42E8H \end{array}$$

十六进制数的减法也与十进制数类似，够减时可直接相减，不够减时服从向高位借 1 为 16 的规则。

例 1.11

$$\begin{array}{rr} & 3D25H \\ - & 05C3H \\ \hline & 3762H \end{array}$$

十六进制数的乘法可以用十进制数的乘法规则来计算，但结果必须用十六进制数来