普通高等教育“十一五”规划教材

汇编语言程序设计

张晓明 等 编著

国防工业出版社
National Defense Industry Press

汇编语言程序设计

张晓明　白凤凤　李雅红　编著

国防工业出版社

·北京·

内容简介

本书以 Microsoft 宏汇编 MASM 为背景，系统讲述了 8086 指令系统及汇编语言程序设计的方法和技术，介绍了 32 位 80x86 系列微处理器指令及其程序设计。全书共分 10 章，第 1 章介绍基础知识；第 2、3 章介绍 8086 微处理器的基本结构、寻址方式及汇编语言程序格式；第 4～7 章叙述基本指令系统及顺序结构、分支结构、循环结构、子程序等设计方法；第 8 章介绍宏与多模块程序设计技术；第 9 章介绍输入/输出和中断程序设计；第 10 章介绍 32 位 80x86 微处理器指令及其程序设计方法。全书提供了大量程序实例，每章后均附有习题。

本书可作为高等院校计算机及相关专业本、专科的汇编语言程序设计课程的教材或参考书，也可供使用汇编语言的工程技术人员参考。

图书在版编目(CIP)数据

汇编语言程序设计/张晓明，白凤凤，李雅红编著. 北京：国防工业出版社，2009.1
ISBN 978-7-118-06008-9

Ⅰ.汇... Ⅱ.①张...②白...③李... Ⅲ.汇编语言－程序设计 Ⅳ.TP313

中国版本图书馆 CIP 数据核字(2008)第 161697 号

※

国防工业出版社出版发行
(北京市海淀区紫竹院南路 23 号 邮政编码 100048)
涿中印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 **印张** 18¼ **字数** 421 千字
2009 年 1 月第 1 版第 1 次印刷 **印数** 1—4000 册 **定价** 29.00 元

国防书店：(010)68428422 发行邮购：(010)68414474
发行传真：(010)68411535 发行业务：(010)68472764

前　言

自从1946年第一台电子数字计算机ENIAC问世以来，计算机技术以令人瞠目的速度迅速地发展起来，并普及渗透到社会生活的各个层面，它已经改变了并且正在继续改变着全人类的生产、生活、交流甚至思维方式。如果在现代社会中不会使用计算机的人被称为“文盲”，那么这种现状必然对计算机技术的专业从业者提出了更高的要求。

作为高等学校中计算机专业的学生，熟练掌握描述计算机工作流程的工具——计算机语言是最基本的要求。计算机语言的发展经历了机器语言、汇编语言、高级语言3个阶段。现在的高级语言种类繁多、功能强大，即使如此，助记符形式的汇编语言仍拥有它不可或缺的地位和作用。其中最重要的原因就是它得天独厚的针对计算机硬件的编程能力。编程人员可以利用汇编语言“无孔不入”地、“渗透式”地对计算机硬件的各个组成部分编程，使得计算机系统的性能得到充分地发挥和利用。随着计算机技术的发展，汇编语言也堪称“与时俱进”，如今已被广泛应用于工业控制、实时系统、计算机网络与通信等众多领域。正因为如此，那些较好地掌握了汇编语言程序设计方法和技巧的计算机专业工作者往往感到“如鱼得水”，对之爱不释手；另一方面，由于汇编语言直接脱胎于“机器语言”，无论是其格式还是使用都较为繁琐、晦涩，这又使得初学者经常“望而生畏”。作为高等学校中多年从事本课程教学的教师，对学生的这种“爱惧交加”的心理体会颇深。我们编写这本教材的初衷，就是将我们多年的教学经验总结出来，与大家交流。

我们在吸收、借鉴其他同类教材精华的基础上，在这本书中融进了多年的教学体会。这主要体现在知识点的渐进展开，例题的由浅入深，编程技巧的点评，教学进度的安排等方面。我们认为只要方法得当，汇编语言不像想象中那么难学，而且，这个学习的过程对于计算机专业的大学生来说，也是非常难得的专业素质的训练过程。过了这一关以后再学习其他编程语言，可能就会觉得不那么难了。

全书共分10章。第1章介绍学习汇编语言所用的基础知识；第2章介绍了8086微处理器的基本结构与寻址方式；第3章介绍了汇编语言程序格式；第4～7章是本书的核心部分，结合具体实例分别讲述了基本指令系统与顺序结构程序设计、分支结构程序设计、循环结构程序、子程序设计等设计方法；第8章介绍了宏与多模块程序设计方法；第9章介绍了输入/输出与中断程序设计；第10章介绍了32位80x86微处理器指令及其程序设计方法。书中提供了大量程序例题，每章后均有习题，便于读者复习和检查学习效果。

由于时间仓促，同时限于编者水平，本书肯定会存在一些缺点和不足，在此恳请专家和读者批评指正。

编者

目　录

第 1 章　基础知识

汇编语言是机器语言的符号表示。用汇编语言设计程序，可以充分利用和发挥计算机硬件的特性与优势。学习汇编语言，不仅可以加深对计算机系统尤其是程序工作原理的理解，而且有助于提高程序设计水平。本章介绍学习汇编语言的预备知识，包括数制与数制之间的转换、汇编语言的基本概念以及计算机中数据的表示方法。

1.1　数制及数制间的转换

1.1.1　数制

在日常生活、生产实践、科学实践中，人们最常用和最熟悉的数制是十进制。然而，计算机内部用二进制表示数据。由于二进制数书写太冗长，而且二进制数与十进制数之间的转换又较为烦琐，因此，在计算机应用特别是汇编语言程序设计中，常采用十六进制。十六进制数简短、易读，与二进制数之间的转换非常容易。人们谈论计算机常用十六进制。此外，有时也使用八进制数。

1. 十进制

十进制数由 0～9 的 10 个数码组成，基数为 10，运算规则是“逢 10 进 1”。同一个数码在十进制数中的不同数位上，它们所代表的数值不同。例如：十进制数 121.12 中，有 3 个数码为 1，但处在不同的数位上，它们表示的值分别为1×10^2、1×10^0、1×10^{-1}，将10^2、10^0、10^{-1}称为相应数位的“权”(或“权值”)，数位不同，其“权”的大小也不同，表示的数值也不一样。

如果 1 个十进制数 D 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为$10^{n-1},10^{n-2},...,10^1,10^0,10^{-1},...,10^{-m}$，所以，任意 1 个十进制数都可以表示为

$$D=\pm\sum_{i=-m}^{n-1}(D_i\times10^i)$$

式中：D_i 表示第 i 位的数码；m, n 为正整数。

2. 二进制

二进制数由 0 和 1 两个数码组成，基数为 2，运算规则是“逢 2 进 1”。

如果 1 个二进制数 B 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为$2^{n-1},2^{n-2},...,2^1,2^0,2^{-1},2^{-2},...,2^{-m}$，所以，任意 1 个二进制数都可以表示为

$$B=\pm\sum_{i=-m}^{n-1}(B_i\times2^i)$$

式中：B_i 表示第 i 位的数码；m, n 为正整数。

3. 十六进制

十六进制数由 0～9 的 10 个数码和 A～F 的 6 个字符组成，基数为 16，运算规则是“逢 16 进 1”。

如果 1 个十六进制数 H 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为$16^{n-1},16^{n-2},...,16^{1},16^{0},16^{-1},...,16^{-m}$，所以，任意 1 个十六进制数都可以表示为

$$H=\pm\sum_{i=-m}^{n-1}(H_i\times16^i)$$

式中：H_i 表示第 i 位的数码；m，n 为正整数。

4. 八进制

八进制数由 0～7 的 8 个数码组成，基数为 8，运算规则是“逢 8 进 1”。

如果 1 个八进制数 O 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为$8^{n-1},8^{n-2},...,8^{1},8^{0},8^{-1},8^{-2},...,8^{-m}$，所以，任意 1 个八进制数都可以表示为

$$O=\pm\sum_{i=-m}^{n-1}(O_i\times8^i)$$

式中：O_i 表示第 i 位的数码；m，n 为正整数。

在汇编语言程序中，为了能表示各种不同进制数，通常在数据的结尾用 1 个英文字母来区分。其中，十进制数用 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：139D，11010011B，130AH 等。当然也可以用这些字母的小写形式。

1.1.2 数制之间的转换

计算机内部处理的数据都是二进制数。然而，在程序设计时，为了方便，常常使用一些其他进制的数据。因此，为了能用不同进制表示数据，需要掌握各种进制数之间的转换方法。下面介绍整数在二进制、八进制、十六进制与十进制之间的转换方法。

1. 二进制、八进制或十六进制整数转换为十进制整数

二进制、八进制或十六进制整数转换为十进制整数比较容易，各位数字与其对应权值乘积之和即为等值的十进制整数。

例如：

$$01011100B=2^6+2^4+2^3+2^2=92$$

$$0F3CH=15\times16^2+3\times16^1+12\times16^0=3900$$

$$255O=2\times8^2+5\times8^1+5\times8^0=173$$

2. 十进制整数转换为二进制、八进制或十六进制整数

设给定的十进制整数为 N。通常，将十进制整数转换为 R 进制整数，可以采用下列两种方法。

1) 除基法

除基法是按照从低位到高位的次序生成等值的 R 进制整数的各位数字。具体步骤是：用商(开始时取 N)不断除以 R，记下余数，将所得余数写成 R 进制数码作为所求的 R 进制数的最低位，而后将除得的商再除以 R，直到商为 0 时止。然后，将所得余数逆序排

列，即可得到等值的 R 进制整数。这种方法可简单地记为“除基取余，逆序排列”。

2) 除权法

除权法是按照从高位到低位的次序生成等值的 R 进制整数的各位数字。具体步骤是：写出所有小于 N 的各位 R 进制权值，将这些权值以从大到小的次序排列，然后不断地用余数(开始时取 N)除以各权值，所得的商即为对应 R 进制整数各位的值。

【例 1-1】 设 N=87D，求 N 对应的二进制数。

方法 1：采用除基法。

具体步骤可以采取图 1-1 所示的纵式的计算过程：

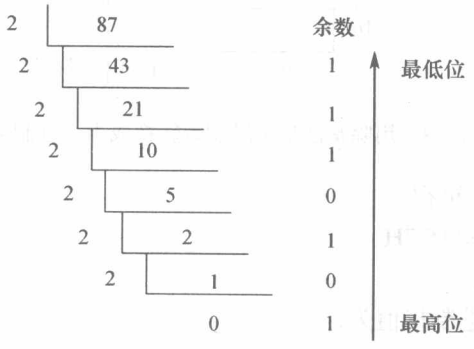

图 1-1 用除基法把十进制数转换成二进制数

将所得的余数逆序排列，可得

N=1010111B

方法 2：采用除权法。

小于 N 的各位二进制权值为

64，32，16，8，4，2，1

具体步骤可以采取图 1-2 所示的纵式的计算过程：

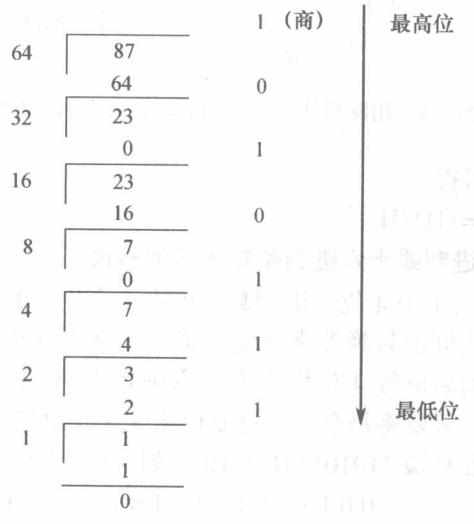

图 1-2 用除权法把十进制数转换成二进制数

将所得的商正序排列，可得

N=1010111B

【例 1-2】 设 N=4567D，求 N 对应的十六进制数。

方法 1：采用除基法。

具体步骤可以采取图 1-3 所示的纵式的计算过程：

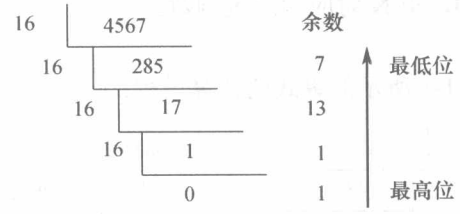

图 1-3 用除基法把十进制数转换成十六进制数

将所得的余数逆序排列，可得

N=(1)(1)(13)(7)=11D7H

方法 2：采用除权法。

小于 N 的各位十六进制权值为：

4096，256，16，1

具体步骤可以采取图 1-4 所示的纵式的计算过程：

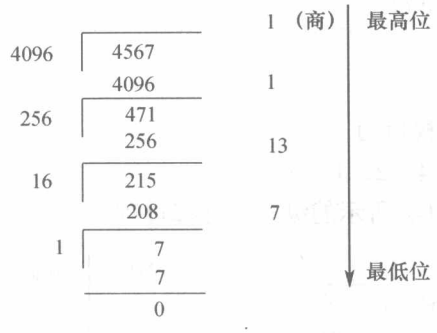

图 1-4 用除权法把十进制数转换成十六进制数

将所得的商正序排列，可得

N=(1)(1)(13)(7)=11D7H

3. 二进制整数与八进制或十六进制整数之间的转换

由于 $2^4=16$，所以任何 1 个 4 位二进制数均可由 1 个 1 位的十六进制数来表示。因此，二进制数与十六进制数之间的转换非常方便。将二进制整数转换为十六进制整数时，只要将二进制整数从低位到高位每 4 位用 1 个十六进制位表示即可；反之，将十六进制整数转换为二进制整数时，只要将每个十六进制位用 4 个二进制位表示即可。例如：

【例 1-3】 将二进制整数 11010110111110B 转换为十六进制整数。

0011 0101 1011 1110

3 5 B E

故 110101101111110B=35BEH

【例 1-4】 将十六进制整数 A19CH 转换为二进制整数。

A 1 9 C

1010 0001 1001 1100

表 1-1 所列为十进制、二进制与十六进制数之间的对应关系。

表 1-1 十进制、二进制与十六进制数之间的对应表

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

同理，由于 2^3=8，因此，将二进制整数转换为八进制整数时，只要将二进制整数从低位到高位每 3 位用 1 个八进制位表示即可；反之，将八进制整数转换为二进制整数时，只要将每个八进制位用 3 个二进制位表示即可。例如：

【例 1-5】 将二进制整数 110101101111110B 转换为八进制整数。

011 010 110 111 110

3 2 6 7 6

故 110101101111110B=32676O

1.1.3 二进制与十六进制的运算规则

在进行二进制数或十六进制数的运算时，虽然可以先转换为十进制数进行运算，然后将结果再转换为二进制数或十六进制数，但这样做比较烦琐。其实，二进制、十六进制运算与十进制类似，只不过是“逢 2 进 1”或“逢 16 进 1”。

基本的二进制算术运算规则如下。

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

【例 1-6】 二进制数的算术运算。

(1) 1101B + 0011B = 10000B

(2) 1101B − 0011B = 1010B

(3) 1101B × 0011B = 1101 + 1101 = 100111B

【例 1-7】 十六进制数的算术运算。

(1) 05C3H + 3D25H = 42E8H

(2) 3D25H − 05C3H = 3762H

(3) 05C3H × 00ABH = 3F61 + 399E = 3D941H

二进制数以及十六进制数的除法可根据其乘法和减法规则处理，这里不再赘述。

1.2 计算机语言

语言是人们用来表达意思，交流思想，抒发感情的工具。若两个人使用同种语言，则可以直接进行交流，否则就需要借助于翻译。要使计算机为人类服务，就必须通过某种工具，告诉计算机“做什么”甚至“怎样做”，这种工具就是计算机语言。人们用计算机语言安排好处理步骤，每一步都是用计算机语言编写的，然后送入计算机内，计算机就按处理步骤，一步步地完成人们所规定的工作。用计算机语言描述的处理步骤称为程序，而编制处理步骤的过程称为程序设计。

计算机语言通常可分为 3 类：机器语言(Machine Language)、汇编语言(Assembly Language)和高级语言(High Level Language)。其中，机器语言和汇编语言是与机器密切相关的，一般称为低级语言。

1.2.1 机器语言

计算机能够直接识别并进行处理的是二进制数 0 和 1 组成的代码，计算机的工作就是对二进制信息进行处理。

1. 机器指令

机器指令是指用二进制编码的指令，指示计算机所要进行的操作以及操作的数据对象。机器指令由指令译码器所识别，并经过一定的时钟周期付诸实现，从而完成指令所规定的操作。

机器指令由操作码(Opcode)和操作数(Operand)构成。操作码表明处理器要进行的操作，如加法、减法、左移、右移等。操作数表明参加操作的数据或数据地址。

2. 指令系统与机器语言

指令系统(Instruction Set)是指计算机机器指令的集合。指令系统及其机器指令的使用规则便构成了机器语言。

机器语言是计算机唯一能够识别的语言。例如，完成两个数据 100 和 256 相加的操作，在 Intel 8086 CPU 上用十六进制表达的代码序列如下：

```
B8 64 00
05 00 01
A3 00 20
```

虽然只有几行，但几乎没有人能直接读懂该程序段的功能，因为机器语言看起来就是一串毫无意义的代码。

3. 机器语言的优缺点

1) 机器语言的优点

(1) 机器语言能被计算机直接认识、执行。

(2) 程序紧凑，占用内存空间少，执行速度高。

(3) 能充分发挥计算机的硬件特性和优势。

2) 机器语言的缺点

(1) 用机器语言设计出的程序难以理解，不易调试，易出错。

(2) 机器语言通用性差，开发效率低。

因此，只是在计算机发展的早期或不得已的情况下，才用机器语言编写程序。现在，除了有时在程序某处需要直接采用机器指令填充外，几乎没有人采用机器语言编写程序了。

1.2.2 汇编语言

为了克服机器语言难以记忆、阅读和理解的缺点，人们采用便于记忆、并能描述指令功能的符号来表示机器指令。表示指令操作码的符号称为指令助记符(Mnemonic)，简称助记符；助记符一般是表明指令功能的英语单词或其缩写。指令操作数同样也可以用易于记忆的符号表示。

用助记符描述的指令称为汇编格式指令或符号指令，通常简称指令。指令和伪指令的集合以及使用它们编写程序的规则便构成了汇编语言。用汇编语言书写的程序称为汇编语言程序，或称为汇编语言源程序。关于伪指令的概念将在第 3 章介绍。

若使用汇编语言来实现 100 与 256 相加，则可用下述程序段表达：

```
MOV   AX,100          ；取一个数据 100
ADD   AX,256          ；实现 100+256
```

可以看出，用汇编语言编写的程序要比机器代码容易理解得多。

然而，汇编语言与机器语言并无本质区别，汇编语言仅仅是机器语言的符号表示，每条汇编语言指令均对应唯一的机器指令。因此，汇编语言同样具有机器语言“与机器硬件密切相关”等特点，只是在直观和记忆方面有了较大改进。

由于计算机只能识别机器语言，因此，用汇编语言编写的源程序不能在计算机上直接执行，必须将其翻译成机器语言。把汇编语言源程序翻译成由机器语言描述的目标程序的过程称为汇编或汇编过程，完成汇编过程的程序称为汇编程序(Assembler)或汇编器。

汇编程序的主要功能是对汇编语言源程序进行语法检查，若源程序没有语法错误，则将符号指令翻译成机器指令，生成相应的目标文件(.OBJ 文件)。

尽管目标文件已经是机器语言程序，但还不能直接执行，需要通过连接程序或称连接器(Linker)将其与其他目标文件或库文件连接在一起，生成可执行文件(.EXE 文件)后，才能在计算机上运行。

连接程序的主要功能是实现多个目标文件及库文件的连接，定位目标文件中可能存在的外部符号，完成浮动地址的重定位，最后生成可执行文件。

1.2.3 高级语言

尽管汇编语言较机器语言在记忆和直观等方面有了很大的改进，但仍然烦琐难记，并无本质上的提高。人们希望有一种接近自然语言或数学表达式的程序设计语言，使程序设计工作能避开与机器硬件相关的细节，而着重于解决问题的算法本身，因此便产生了高级语言。目前，广泛应用的高级语言有十多种，如 BASIC、Pascal、C/C++、Java 等。用高级语言表达 100 与 256 相加，就是通常的数学表达式：100+256。

与汇编语言语言相比，高级语言在程序设计的简易性以及代码的可读性和可移植性等方面有了质的飞跃。当然，用高级语言编写的源程序不会被机器直接执行，而需经过编译或解释程序的翻译才可变为机器语言程序。

1.2.4 学习汇编语言的意义

高级语言简单、易学、程序开发效率高，而汇编语言复杂、难懂、程序开发效率低。那么，是否就没有必要再学习和使用汇编语言了呢？下面先来比较一下汇编语言和高级语言的特点。

汇编语言与处理器密切相关。每种处理器都有自己的指令系统，相应的汇编语言各不相同。所以，汇编语言程序的通用性、可移植性较差。相对来说，高级语言与具体计算机无关，用高级语言编写的程序可在多种计算机上编译后执行。

汇编语言功能有限，又涉及寄存器、主存单元等硬件，所以编写程序比较烦琐，调试较困难，高级语言提供了强大的功能，采用类似自然语言的语法，所以容易被掌握和应用，也不必关心诸如寄存器、标志等问题。

由于汇编语言本质上就是机器语言，可直接、有效地控制计算机硬件，因而，容易产生运行速度快、指令序列短小的高效目标程序。高级语言不易直接控制计算机的各种操作，编译后产生的目标程序往往比较庞大、程序难以优化，所以运行速度较慢。

由此可见，汇编语言的优势在于能直接控制计算机的硬件部件，这样可编写出在“时间”和“空间”两方面均为最优的程序。然而，随着计算机速度的不断提高和内存容量的大幅提升，用高级语言编写的程序就可以满足大多数问题对时间和空间方面的要求，因此，有时可采用高级语言和汇编语言混合编程的方法，相互取长补短，以求更好地解决实际问题。

学习汇编语言的意义主要体现在以下 4 个方面。

(1) 要求程序具有执行速度快，或者占用存储空间少的场合。例如，操作系统的核心程序段，实时控制系统的软件，智能仪器仪表的控制程序等。

(2) 要求程序与计算机硬件密切相关，程序能直接、有效地控制硬件。如，输入/输出(I/O)接口电路的初始化程序段，外部设备的底层驱动程序等。