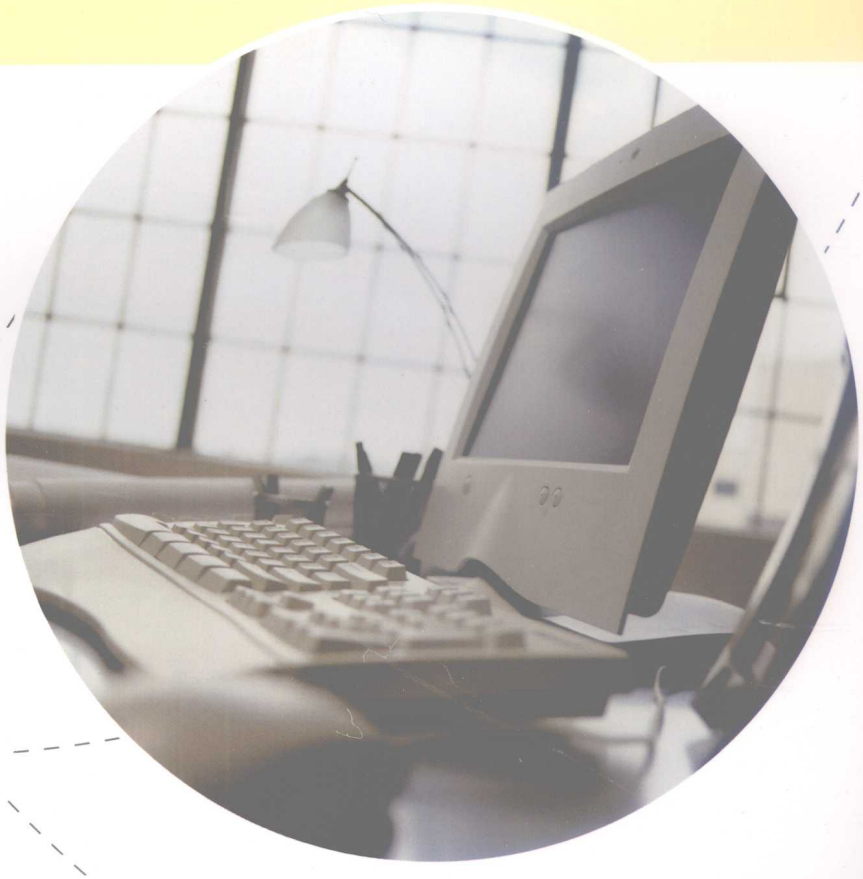
21世纪全国高等院校计算机教育“十一五”规划教材

汇编语言程序设计

主　编　李　静　原永滨
副主编　樊景博　谌洪茂
　　许青林　张慧春



中国计划出版社

21世纪全国高等院校计算机教育“十一五”规划教材

汇编语言程序设计

本书编委会　编著

中国计划出版社

图书在版编目（CIP）数据

汇编语言程序设计 / 《汇编语言程序设计》编委会编著. —北京：中国计划出版社，2008.7
21世纪全国高等院校计算机教育“十一五”规划教材
ISBN 978-7-80242-129-5

Ⅰ. 汇… Ⅱ. 汇… Ⅲ. 汇编语言—程序设计—高等学校—教材 Ⅳ. TP313

中国版本图书馆CIP数据核字（2008）第087965号

内 容 简 介

本书以目前广泛使用的 PC 机为学习平台，系统讲述了 8086 指令系统及汇编语言程序设计的基本方法与技巧，并专门介绍了 32 位 x86 指令及其程序设计。全书共 10 章，主要内容包括宏汇编语言、程序设计的基本技巧、输入/输出程序设计、32 位 CPU 扩展功能、汇编语言上机内容与要求。本书内容详实，叙述细腻易懂，在章节安排上由浅入深，并且在每章后都附有习题，使得知识更易于理解和掌握。

本书可作为高等院校计算机、自动化、电子技术及相关专业“汇编语言程序设计”课程教材，也可作为独立学院、高职高专计算机及相关专业、有关工程技术人员的教学参考书。

21世纪全国高等院校计算机教育“十一五”规划教材

汇编语言程序设计

本书编委会　编著

☆

中国计划出版社出版

（地址：北京市西城区木樨地北里甲11号国宏大厦 C 座4层）

（邮政编码：100038　电话：63906433　63906381）

新华书店北京发行所发行

河北省高碑店市鑫宏源印刷包装有限责任公司印刷

787×1092 毫米　1/16　15.5 印张　377 千字

2008 年 7 月第 1 版　2008 年 7 月第 1 次印刷

印数 1—4000 册

☆

ISBN 978-7-80242-129-5

定价：25.00元

本书编委会

主　编：李　静　原永滨

副主编：樊景博　谌洪茂　许青林　张慧春

参　编：郭团囡　叶喜民　余　超　王会芳　陈天凯

前　言

“汇编语言程序设计”是高等院校计算机硬、软件及应用专业学生都必须学习的核心课程之一，它是计算机组成原理、操作系统及其他核心课程的基础课，也是微机原理、单片机应用等课程的学习基础。

本书的主要特点是突出理论性、实践性、先进性、通俗性，力求自学方便，适用于普通高等院校，使学生在尽量短的时间内熟练掌握最基本的汇编语言的功能、用法和编程技巧。

全书共分 10 章，第 1 章首先对汇编语言程序设计中的基础知识进行讲解，第 2 章着重讲解了 80x86 计算机组织结构，第 3 章详细介绍了 80x86 的指令系统和寻址方式，第 4～8 章用图、例结合的方式分别讲述了汇编语言程序格式及伪指令，3 种程序设计结构，子程序设计，输入/输出程序设计以及汇编语言与 C 语言的混合编程，第 9 章专门介绍了 32 位机新增指令及功能，第 10 章则是汇编语言上机实验，对以上章节所述内容进行了复习巩固。并且在每章后均附有习题，有助于进一步掌握各章内容。

本书由李静、原永滨主编，樊景博、谌洪茂、许青林、张慧春担任副主编，郭团囡、叶喜民、余超、王会芳、陈天凯参与编写。

本书在编写过程中，参阅了大量的图书资料，其中的部分参考资料已列在参考文献中，编者在些向这些图书的作者表示衷心的感谢。

由于编者水平有限，书中难免有不足之处，恳请有关专家和广大读者不吝赐教。

编者

2008年5月

目　录

第 1 章

汇编语言基础知识

汇编语言是唯一能够充分利用计算机硬件特性面向机器的一种低级语言。它以助记符的形式表示每一条计算机指令，并且每一条指令都对应着计算机硬件的一个具体的操作，它随机器结构的不同而不同。因此，汇编语言与计算机有着密不可分的关系。要学习汇编语言，就必须首先了解该机器有关的硬件结构、数据类型及表示方法等。本章将从汇编语言程序设计的角度出发，介绍有关的预备知识，如：什么是汇编语言、Intel 8086 微处理器中的寄存器组、主存储器的编址方式及物理地址的形成方式、数和符号在计算机中的表示方法，同时还以一个源程序作为实例介绍了汇编源程序的基本结构和格式，为以后各章节的学习做好铺垫。

本章主要内容

- 计算机语言
- 数值数据表示形式
- 字符数据表示形式

1.1 计算机语言

计算机不会自动的为人们完成各种工作，它工作的前提就是必须接受人的操纵和控制，人们为了让计算机按自己的意愿工作，就不得不与计算机之间进行信息交流，那么交流信息所使用的工具即称之为计算机语言。目前程序设计所使用的计算机语言大致分为 3 类：机器语言、汇编语言和高级语言。其中，机器语言和汇编语言是靠近机器的语言。相对而言，高级语言更接近于自然语言，易学、易记，容易掌握，便于阅读，并且使用方便，通用性强，又不依赖于具体的计算机。但是计算机并不识别高级语言，它只能识别那些在设计机器时事先规定好的机器指令。

1.1.1 机器语言

机器语言是指一台计算机全部的指令集合。计算机所使用的是由“0”和“1”组成的二进制数，二进制是计算机的语言的基础。计算机发明之初，人们只能降贵纡尊，用计算机的语言去命令计算机干这干那，一句话，就是写出一串串由“0”和“1”组成的指令序

列交由计算机执行，这种计算机能够认识的语言，就是机器语言。使用机器语言是十分痛苦的，特别是在程序有错需要修改时，更是如此。

机器指令是指挥计算机怎样进行操作、并如何完成任务的命令。机器指令的一般形式为：

操作码	地址码

操纵码和地址码均是由 0 和 1 组成的二进制代码，由此可见每一条机器指令都是用一组二进制代码表示的。操纵码指出运算的类型，如加、减、传送、移位等。地址码指出参与运算的操作数和运算结果存放的位置。例如，将偏移地址为 100 的字存储单元中的内容加 2，再送回到原存储单元中去，如果用 Intel 8086 的机器指令来完成该操纵，则相应的机器指令为：

```
1000 0011
0000 0110
0110 0100
0000 0000
0000 0010
```

其中，第一、二行中的两个 8 位二进制数是操作码，表示要进行“加”操作，同时还指明了以何种方式取得两个加数。第三、四行中的两个 8 位二进制数指出了第一个加数（称目的操作数）所存放的地址是 100（或 64H）。相加的结果也送入该存储单元中。第五行的 8 位二进制数指出了第二个加数（称源操作数）是 2。

机器指令通常被称为硬指令，它是面向机器的。也就是说每台计算机都规定了自己特有的、一定数量的基本指令，这批指令的全体就构成了计算机的指令系统，这种机器指令所构成的集合就是机器语言。因此，用机器语言编写的程序自然就被称之为机器语言程序。

1.1.2 汇编语言

汇编语言（Assembly Language）是面向机器的程序设计语言。在汇编语句中，用助记符（Memoni）代替操作码，用地址符号（Symbol）或标号（Label）代替地址码。这样用符号代替机器语言的二进制码，就把机器语言变成了汇编语言。于是汇编语言亦称为符号语言。使用汇编语言编写的程序，机器能直接识别，要由一种程序将汇编语言翻译成机器语言，这种起翻译作用的程序叫汇编程序，汇编程序是系统软件中语言处理系统软件。汇编程序把汇编语言翻译成机器语言的过程称为汇编。汇编语言比机器语言易于读写、易于调试和修改，同时也具有机器语言执行速度快，占内存空间少等优点，但在编写复杂程序时具有明显的局限性，汇编语言依赖于具体的机型，不能通用，也不能在不同机型之间移植。

为了减轻使用机器语言编程的痛苦，人们进行了这种有益的改进：用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，比如，用“ADD”代表加法，“MOV”代表数据传递等，这样一来，就很容易读懂并理解程序，纠错及维护都变得方便了，这种

程序设计语言即为汇编语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。本教材中所介绍的是 Intel 8086 宏汇编语言。例如，一个加法例子，用宏汇编语言书写可表示的形式为：

```
ADD WORD PTR DS:[100],2
```

其中，“ADD”是加法指令的助记符，而“DS:[100]”表示在当前数据段中，偏移地址为 100 单元中的内容是目的操作数，“WORD PTR”则说明这个目的操作数是 16 位二进制数，源操作数是 2，相加的结果送入目的操作数所在的单元。

汇编语言是为了程序编写的方便而设计出的一种符号语言，用它编写出的源程序其实并不能直接的被计算机识别，还必须要将它翻译成由机器指令组成的程序后，计算机才能识别并执行。这种由源程序经过翻译转换成的机器语言程序称之为目标程序。目标程序中的二进制代码（即机器指令）称之为目标代码。这个翻译工作一般都是由计算机自己去完成，但人们必须事先将翻译方法编写成一个语言加工程序作为系统软件的一部分，在需要的时候只要让计算机执行这个程序就可完成对某一汇编源程序的翻译工作。这种把汇编源程序翻译成目标程序的语言加工程序称为汇编程序。汇编程序进行翻译的过程叫做汇编。

汇编程序与汇编源程序、目标程序之间的关系如图 1-1 所示。从关系图可以看出，汇编程序实际上相当于一个翻译器，它加工的对象是人们可以读懂的汇编源程序，而加工的结果是计算机可以识别的目标程序。

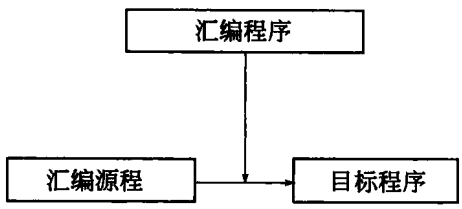


图 1-1　汇编程序与汇编源程序、目标程序之间的关系

为了使汇编程序正确地完成翻译工作，就必须要告诉汇编程序，源程序应从什么位置开始安放，汇编到什么位置结束，数据的类型是什么，并且它应该放在什么位置，留多少内存单元作为临时存储区等相关的信息。这就要求源程序中应该有一套告诉汇编程序如何进行汇编工作的命令，这种命令称伪指令（或称汇编控制命令）。由此可见，指令助记符、语句标号、数据变量、伪指令及它们的使用规则构成了整个汇编语言的内容。由于汇编语句基本上与机器指令对应，因而它的编写也是相当麻烦的。为了简化程序的编写，提高编程的效率，现代的计算机系统一般都提供了宏汇编程序。它允许程序员用一个名字（宏指令名）来代替程序中重复出现的一组语句，然后在源程序中所需要的地方使用宏指令名字及不同的参数进行宏调用。熟练灵活地使用宏调用功能可以使汇编源程序编写得像高级语言一样清晰、简洁，并且容易使算法标准化、处理问题也更加灵活。这样不仅可以加速程序的编写进度，而且还利于阅读、修改和调试源程序。Intel 8086 把用宏汇编语言编写的源程序翻译成目标程序的工作是由 MASM 的宏汇编程序来完成的。

与机器语言相比，汇编语言易于理解和记忆，编写的源程序也容易阅读和调试，所占用的存储空间、执行速度与机器语言相当。与高级语言相比，汇编语言具有直接和简捷的

特点，用它编制程序能精确地描述算法，可以充分发挥计算机硬件的功能。某些用高级语言难以实现的操作，用汇编语言就可以简单实现。目前系统程序的研制虽然已有不少采用高级语言，但其产生的目标往往还是采用汇编语言的形式，而且还有一些系统程序和应用程序仍需要用汇编语言来编写。此外，实用中还有大量的用汇编语言编写的系统程序（如各种编译程序、服务性程序、操作系统、数据库管理系统等）和应用程序（如各种实时控制程序等）需要阅读、分析乃至修改。因此，几乎每一个计算机系统，都把汇编语言作为系统的基本配置，把汇编程序作为系统软件的核心成分之一。而汇编语言程序设计是从事计算机研究与应用，特别是软件研究的基础。对于从事计算机研制和应用的广大科技工作者来说，掌握汇编语言及其程序设计技术是非常重要的。

1.1.3 高级语言

与机器语言和汇编语言相比，高级语言与具体计算机无关，是一种能方便描述算法过程的计算机程序设计语言。高级语言种类千差万别，但一般包含有以下 4 部分：数据用来描述程序所涉及的数据；运算用来描述运算；控制用来表达程序的控制构造；传输用来表达数据的传输。由于高级语言程序主要是描述计算机的解题过程，即描述复杂的加工处理过程，所以也称这种高级语言为面向过程语言。

用高级语言编写的程序称为“源程序”。计算机不能直接执行源程序的语句，通常有解释方式和编译方式两种方法在计算机上执行源程序。解释方式，即让计算机运行解释程序，解释程序逐句取出源程序中的语句，对它作解释执行，输入数据，产生结果。编译方式，即先运行编译程序，从源程序一次翻译产生计算机可直接执行的二进制程序（称为目标程序）；然后让计算机执行目标程序，输入数据，产生结果。

解释方式的主要优点是计算机与人的交互性好，调试程序时，能一边执行一边直接改错，能较快得到一个正确的程序。缺点是逐句解释执行，运行速度慢。编译方式的主要优点是计算机运行目标程序快，缺点是修改源程序后必须重新编译以产生新的目标程序。现在也有将上述两种方式结合起来的，即先编译源程序，产生计算机还是不能直接执行的中间代码，然后让解释程序解释执行中间代码。这样做的好处首先是比直接解释执行快；更大的好处是中间代码独立于计算机，只要有相应的解释程序，就可在任何计算机上运行。

1.1.4 汇编语言的特点与汇编程序

1. 汇编语言的特点

由于汇编语言使用指令助记符和符号地址，所以它要比机器语言容易掌握。与高级语言相比较，汇编语言有如下特点。

（1）汇编语言与机器关系密切

因为汇编格式指令是机器指令的符号表示，所以汇编格式指令与机器有着密切的关系，因此汇编语言也与机器有着密切的关系，确切地说汇编语言与机器所带的 CPU（微处理器）有着十分密切的关系。对于各种不同类型的 CPU，要使用各种不同的汇编语言。于是，对于各种不同类型的 CPU，也就有各种不同的汇编程序。

由于汇编语言与机器关系十分密切，所以汇编语言源程序与高级语言源程序相比，它的通用性和可移植性要差得多。但通过汇编语言可最直接和最有效地控制机器。常常是大多数高级语言难以做到的。

（2）汇编语言程序效率高

用汇编语言编写的源程序在汇编后所得的目标程序效率高。这种目标的高效率反映在时间和空间两个方面：其一是运行速度快；其二是目标程序短。在采用相同算法的前提下，任何高级语言程序在这两方面的效率都不如汇编语言程序，许多情况下更是远远不及。

汇编语言程序能获得“时空”高效率的主要原因：构成汇编语言主体的汇编格式指令是机器指令的符号表示，每一条汇编格式指令都是所对应的某条机器指令的“化身”；另一个重要原因是汇编语言程序能直接并充分利用机器硬件系统的许多特性。高级语言程序在上述两点上要逊色得多。

（3）编写汇编语言源程序烦琐

编写汇编语言源程序要比编写高级语言源程序繁琐得多。汇编语言是面向机器的语言，高级语言是面向过程或面向目标、对象的语言。如下两点突出表现了汇编语言的这一特性。

① 作为机器指令符号化的每一条汇编格式指令所能完成的操作极为有限。例如，Z80指令集中没有乘法指令，8086指令集中没有能够同时完成两次算术运算的指令。

② 程序员在利用汇编语言编写程序时，必须考虑包括寄存器、存储单元和寻址方式在内的几乎所有细节问题。例如：指令执行对标志的影响。堆栈设置的位置等。在使用高级语言编写程序时，程序员不会遇到这些琐碎却重要的问题。

（4）汇编语言程序调试困难

调试汇编语言程序往往要比调试高级语言困难。汇编格式指令的功能有限和程序员要注意太多的细节问题是造成这种困难的两个客观原因；汇编语言提供给了程序员最大的“舞台”，而程序员往往为了追求“时空”上的高效率而不顾程序的结构，这是造成调试困难的主要原因。

2. 汇编程序

用汇编语言编写的程序称为汇编源程序，计算机不能直接识别，必须将其翻译成由机器指令组成的程序后，CPU才能执行，这一过程称为“汇编”。用于将汇编原程序翻译成机器语言的程序称为汇编程序，这种由原程序经过机器翻译转换成的机器语言程序也称为目标程序。目标程序还不能直接交给CPU执行，它还需要通过连接程序装配成可执行的程序才能被执行。它们之间的关系如图1-2所示。

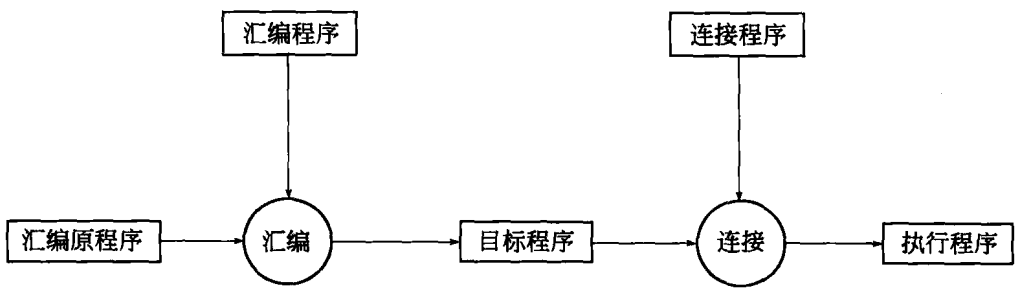


图1-2　汇编程序与目标程序、可执行程序之间的关系

汇编程序 MASM.exe 是一种专门用于将 Intel 8086 的汇编语言源程序翻译成机器语言程序的翻译器，它在执行汇编程序时首先进行语法检查，指出错误的位置及类型，编程人员要一一进行修改，直到源程序无错，汇编程序才执行翻译工作，因此，汇编程序决定了汇编语言的语法。汇编语言的源程序一般以.asm 作为文件的扩展名，目标程序以.obj 作为文件扩展名，执行程序以.exe 作为文件扩展名。

1.2 书中使用符号的说明

在本书的学习过程中，需要引用以下一些符号，下面选择部分符号逐一介绍，有助于更进一步的学习和掌握汇编语言。

（…） 表示地址“…”中的内容。例如，偏移地址为 100 的存储单元中的内容为 50，则表示为（100）=50；寄存器 BX 的内容为 0FFFFH，则表示为（BX）=0FFFFH。

[…] 表示以地址“…”中的内容为偏移地址。例如，（BX）=03A2H，而（03A2H）=100，则（[BX]）表示以 BX 的内容为偏移地址，在该偏移地址中存放的数据。此处（[BX]）=100。

EA 某一存储单元的偏移地址，即指该存储单元到它所在段段首址的字节距离。

PA 某一存储单元的物理地址。

OPD 目的地址，即目的操作数存放的偏移地址。

OPS 源地址，即源操作数存放的偏移地址。

→ 表示传送。例如，300H→BX 表示将操作数 300H 传送到寄存器 BX 中；设（100）=50，（100）→BX 表示将偏移地址为 100 单元中的内容 50 传送到 BX 中。

↑（SP）表示出栈（弹出）。

↓（SP）表示入栈（压入）。

∧ 表示逻辑乘。

∨ 表示逻辑加。

⊻ 表示接位加。

1.3 数值数据在计算机内的表示形式

1.3.1 基本数据类型

计算机数据的最小单位是一个二进制位（Binary Digit），简称位（Bit）。一个二进制位可以由 0 和 1 两个值表示，也可以表示两种对立的状态，如电流的有与无，开关的开与关等。

1. 字节（Byte）

把 8 个二进制位组合在一起构成 1 个字节。字节是计算机内常用的信息长度单位。1 个字节中的 8 位是从右向左编号的，依次为 0～7，如图 1-3 所示。其中第 0 位被称为最低位，第 7 位被称为最高位。

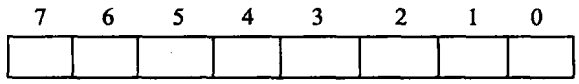


图 1-3　字节的位编号

2．字（Word）

1 个字有 2 个字节（16 位）组成。字也是计算机中常用的数据单位，其表现力比字节强。1 个字节的 16 位是从右向左编号的，依次为 0～15，如图 1-4 所示。其中，第 0 位被称为最低位，第 15 位被称为最高位。第 0～7 位构成低字节（也称低 8 位），第 8～15 位构成高字节（也称高 8 位）。

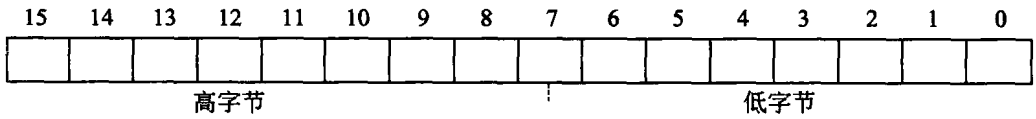


图 1-4　字的位编号

3．双字（Double Word）

1 个双字有 2 个字（32 位）组成。双字的编号从右向左依次为 0～31，如图 1-5 所示。其中第 0 位被称为最低位，第 31 位被称为最高位。第 0～15 位构成低字，第 16～31 位构成高字。在 Intel 8086 处理器中，双字不是存取单位，也不是计算机的数据单位，只有个别的指令（如段间转移指令和段间调用指令）一次能存取一个双字。

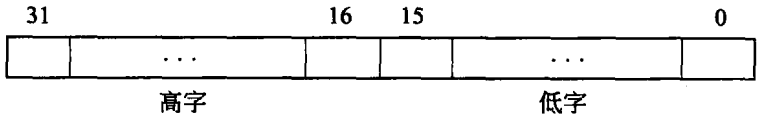


图 1-5　双字的位编码

1.3.2　计算机中数值数据的表示

在计算机中，数值数据有两种表示法：定点表示法和浮点表示法。浮点表示法比定点表示法所表示的数的范围大、精度高，但由于 Intel 8086 微处理器是定点机，它处理的数据小数点位置是固定的，没有处理浮点数的专门指令，对浮点数的运算只能用程序来实现，速度相当慢。为了弥补这方面的不足，专门为 8086 微处理器配置了 8087 微处理器数值协处理器来实现对浮点数的运算。8087 微处理器具有在内部完成算术运算的功能电路，安装简单、使用方便、精度高，速度可达 8086 微处理器的 100 倍，8086 微处理器宏汇编语言中引进浮点数主要供它使用。因此，在本教材中，不再对浮点数进行讨论，感兴趣的读者可参考 Intel 8087 微处理器协处理器的资料。所以可以认为 8086 微处理器宏汇编语言中的数值数据均是指无符号定点数，由于是将小数点固定在第 0 位的后面，因此在不特别说明时所提到的数都是整数，对于有符号数则一律采用 n 位二进制补码表示，n 可以是 16 位，也可以是 8 位。下面讨论有符号数值数据在字节及字中的表示方法及表示范围。