8086 汇编语言程序设计

王鉴泉　宋立彤

高等教育出版社

高等学校教材

8086 汇编语言程序设计

王鉴泉　宋立彤

高等教育出版社

(京)112 号

内 容 提 要

本书是根据国家教委高等学校计算机软件专业汇编语言程序设计教学基本要求编著。主要内容包括：8086 的系统结构和指令系统；汇编语言程序的结构和设计方法；输入/输出程序和软中断处理程序的设计；MSDOS 文件管理功能的调用；与高级语言程序的连接方法。最后简要地介绍 80286 和 80386 的指令。每章配有习题和上机指导。

本书可作为计算机有关专业的教材或参考书，也可供有关技术人员参考。

作者备有宏指令库软盘，读者若需要请与作者直接联系。

8086 汇编语言程序设计

王鉴泉　宋立彤

*

高等教育出版社出版

新华书店总店北京发行所发行

北京市顺新印刷厂印装

*

开本 787×1092　1/16　印张 19.5 字数 480 000

1994 年 8 月第 1 版　1994 年 8 月第 1 次印刷

印数0001—5 102

ISBN7-04-005018-8/TP·138

定价 8.50 元

前　言

汇编语言是计算机系统最基本的程序语言。使用汇编语言编写的程序具有如下特点：

- 执行速度最快；
- 占用存储空间最少；
- 充分利用计算机的所有硬件特性和软件资源。

尽管高级程序设计语言功能越来越强，使用越来越方便。但是使用高级语言编写的主要程序模块，调用少量关键的汇编语言程序模块，往往可以接近上述的目标。所以，作为一名程序设计员，了解甚至是熟练使用汇编语言是最基本的要求。

汇编语言的语法简单，指令的意义直观，但是比较深入的学习和灵活的使用还是有一定难度的，能够编写有良好表现的程序，不是轻而易举的。使用汇编语言编制程序，要比用高级语言编程花费更多的时间。在某些情况下高级语言还无能为力，只能使用汇编语言。另外，如果需要高效的程序，即使多花费一些编程时间，这也是划得来的。

本书的适用对象：

本书是按高等学校计算机软件专业汇编语言程序设计教学基本要求编著，可作为大学计算机专业和相近专业"汇编语言程序设计"的教材或参考书。本书尽可能从实用角度介绍应用程序的开发技术，并给出可实际运行的子程序和实用程序，用于应用程序的开发。

如果读者有高级语言程序设计经验，并了解数字编码知识，那么对阅读本书肯定有益处。

本书的内容和组织：

基于上述的目标，编著本书时特别考虑以下内容：

- 在讲述最基本的程序设计方法的同时，尽可能地介绍较新的程序设计技术。由浅入深、循序渐进，结合实用程序阐述 8086 汇编语言程序的结构和设计方法；
- 每章都有"上机指导"，说明如何上机操作和应达到什么目标。每章都附有最基本的习题，不仅作为深入学习和进一步理解正文的手段，有的题目还是正文的补充。
- 考虑到阅读本书的读者大多已有了某种高级语言的基础，故本书采用类 PASCAL 的程序描述语言，说明较复杂的算法；
- 实际应用系统开发工作较多地使用汇编语言实现输入/输出和中断程序设计，为此专门设置两章，详细讲述这方面的技术，并给出了实用程序。

用 * 标识的章节，可作为课外阅读材料。根据作者经验，大约 1/3 的学生有能力阅读带 * 号的章节，甚至探讨更深入的课题。

鸣谢：

在编写过程中，吉林大学计算机系的同仁给予支持和帮助，复旦大学计算机系施伯乐教授审阅了全书并提出宝贵意见，在此一并表示感谢。

本书的主要章节是第一作者在供校内使用的讲义的基础上，并结合多年的教学实践经验编写的，第二作者编写了本书的第2章和第10章。尽管再三斟酌正文和精细地调试和测试程序，还可能有谬误，望读者不吝指正。

王 鉴 泉
于长春吉林大学
1994年5月

目　录

第 0 章　8086 微计算机结构

在过去的 20 年里，微处理机和微计算机的发展令人瞩目，微计算机在计算机的应用领域内占有举足轻重的地位。无论在硬件方面，还是在软件方面，都推动着计算机应用事业的飞速发展。由于其功能和速度可以满足多数情况的需要，不仅单独应用于商业、工农业、科学、教育等行业，甚至家庭也大量地使用各种档次的个人计算机，而且还可以联接成网络、构成分布式系统，满足更高的需求，也可以作为大型计算机的智能终端。目前，IBM PC 系列的个人计算机在微计算机市场中占有最大比例。IBM 系列个人计算机包括 IBM PC/XT、IBM PC/AT 和各种兼容机以及各种 80386 和 80486 个人计算机。IBM PC/XT 及其兼容机使用 Intel 8088 或 Intel 8086 CPU，IBM PC/AT 及其兼容机使用 Intel 80286。较多的 80286/ 80386/ 80486 计算机采用与 IBM PC/AT 兼容的 AT 总线，使用 MSDOS 操作系统或者 XENIX 操作系统。此外，IBM PS/2 个人系统也占有一定的比例，其使用全 32 位的 OS/2 操作系统，提供虚拟存储并允许用户运行多进程，比较充分地发挥了 80386 的优良性能。

Intel 8086 CPU 出现于 1978 年，其后续的处理机分别是 80286、80386、80486 等，虽然功能提高了，但是保持了向上兼容的特点，即高档机可以运行低档机的程序。学习该系列高档机的程序设计，往往可以从 8086 程序设计开始，读者完全可以在读完本书和操作系统的有关文献后，进而了解 80286/80386 等 CPU 的结构和程序设计。

0.0　8086 的历史

1971 年美国 Intel 公司发表了世界上第一个微处理器 4004，它仅是 4 位微处理器，稍后推出世界上第一台微计算机，它使用 4004 作为 CPU。但是，4004 更多地用于计算器。虽然如此，4004 开创了微处理器的新时代。其后，该公司于 1973 年推出用于显示终端的 8 位微处理器 8008。这就是被人们称为第一代的微处理器。

1974 年 Intel 公司推出 8080 微处理器，它是 8 位微处理器，内部包含多个 8 位通用寄存器，能实现 8 位的整数加减法运算，具有 64K Bytes 的寻址空间，并有一级中断的能力。其后，于 1975 年推出性能更好的 8085 微处理器。与此同时，还有 Motorola 公司的 MC6800、Commondare 公司的 6502 以及 Zilog 公司的 Z80 等。这些 8 位微处理器就被人们称为第二代的微处理器，并被设计成不同功能的微计算机。

随着微处理器及其相应的微计算机的发展，8 位的运算和 64K Bytes 的寻址空间，远不能满足实际需要。1978 年 Intel 公司推出 16 位的微处理器 8086，内部包含多个 16 位的通用寄存器，能实现 16 位的整数算术运算，具有 1M Bytes 的寻址空间，并有多级中断的能力。8086 的设计极为成功，采用很多先进的技术和概念，如先进的集成电路工艺和技术、地址分段的概念等。为了与已经大量使用的 8 位数据总线兼容，在 1979 年 Intel 公司推出准 16 位的微处理器 8088，除了 CPU 与内存储器采用 8 位传送外，其它结构与 8086 完全相同。由于 8086 和

8088 的指令系统和中断结构相同，本书以后部分在涉及 8086 和 8088 时，就统一用 8086 表示二者。

由于微处理器具有较高的性能价格比，IBM 公司为它的个人计算机选用 8088 作为其 CPU。软件行业的巨人—— Microsoft 公司开发了它的操作系统，并被称为 MSDOS。该系统是一个开放系统，吸引成千上万的公司为之开发软件。IBM PC 的成功，有力地推动了计算机各个领域的发展。1982 年 Intel 公司推出 80286 微处理器，虽然它仍是 16 位微处理器，但是它的速度比 8086 高 6 倍以上；而且具有大批量数据、实时任务和多道程序处理能力；具有存储保护能力，可对程序和数据保密；可以实现虚拟存储系统，每个用户任务的运行空间可达 16M Bytes。

与此同时，还有 Motorola 公司的 MC68000、National Semiconductor 公司的 16032、Zilog 公司的 Z8000 等 16 位微处理器，就是 80 年代的主流微处理器，被人们称为第三代微处理器。

1985 年 Intel 公司推出 80386 微处理器，它是全 32 位微处理器，它的速度比其前辈 8086 高几十倍；有大批量数据、实时任务和多道程序处理能力；它也同样具有存储保护能力，可对程序和数据保密；能实现虚拟存储系统，而且每个用户任务的运行空间可达 4096M Bytes。

与此同时，还有 Motorola 公司的 MC68020、National Semiconductor 公司的 32032、Zilog 公司的 Z80000 等。这些 32 位微处理器就被人们称为第四代的微处理器。

图 0-0 显示了 Intel 公司 8086 系列微处理器的发表年代和性能对比。

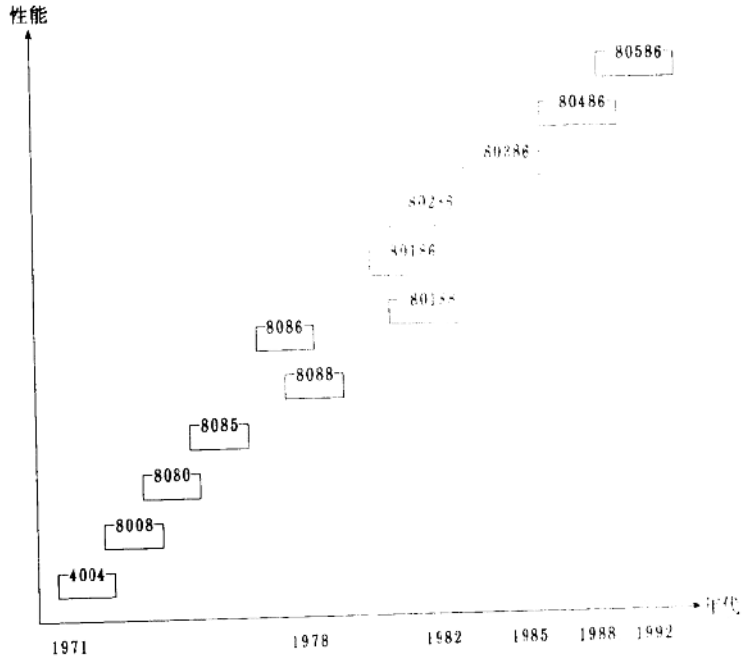

图 0-0 Intel 微处理器发展示意图

8086 系列的一个重要特点是软件兼容。80286 或 80386 在实方式下，可以直接运行 8086 程序；80386 在 286 方式下，可以直接运行 80286 的程序。

微处理器的内部运算指令不外乎执行整数的加、减、乘和除法的运算，不能满足科学计算和复杂的图形处理的需求。为此 8086 系列有相应的协处理器，分别为 8087、80187、80287 和 80387，用来处理浮点数的算术运算和更为复杂的超越函数的计算。而 80486 芯片内部包含协处理器，从程序设计的角度看，它分为两部分：80386 微处理器和 80387 协处理器。

0.1 8086 系统结构

在这一节，我们仅介绍 8086 系统结构的那些与本书前 7 章有关的内容，关于输入/输出（以后将简称为 I/O）的描述见第 7 章，关于中断机制的描述见第 8 章。有关硬件方面的更详细的说明见参考文献[2]、[11]。

0.1.0 8086 微计算机结构

60 年代后期某些小型计算机采用总线结构，70 年代出现的微型计算机也都使用总线结构。计算机的总线是通信线，用于 CPU、内存储器、外部设备传输信息。总线分为三类：地址总线、控制总线和数据总线。8086 微计算机也使用总线结构，CPU、内存储器、外部设备都连结到总线上。图 0-1 给出了 8086 微计算机逻辑结构示意图，其中 AB 表示地址总线，CB 表示控制总线，DB 表示数据总线。有必要指出，这仅是示意图，例如在 80286 系统，80286 CPU 通过 82288 总线控制器、8259 中断控制器、8286 收发器、8282 锁存器等辅助芯片与总线相连接，并接有外部的 82284 时钟发生器。

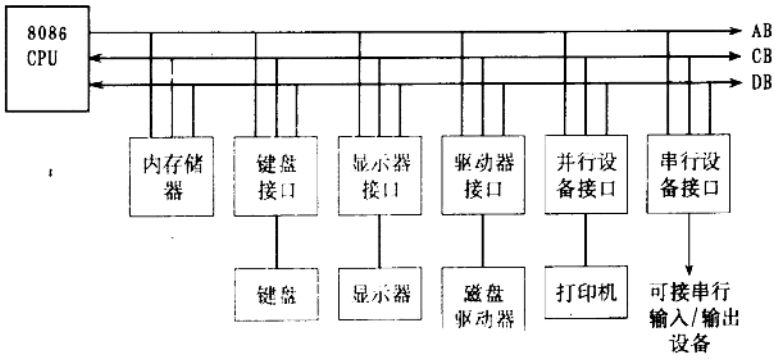

图 0-1 8086 微计算机逻辑结构的示意图

0.1.1 CPU 内部结构

8086 CPU 芯片有四个独立的处理部件，分别是地址部件 AU、总线部件 BU、指令部件 IU 和执行部件 EU。图 0-2 表示了各个部件的逻辑关系和数据流向。

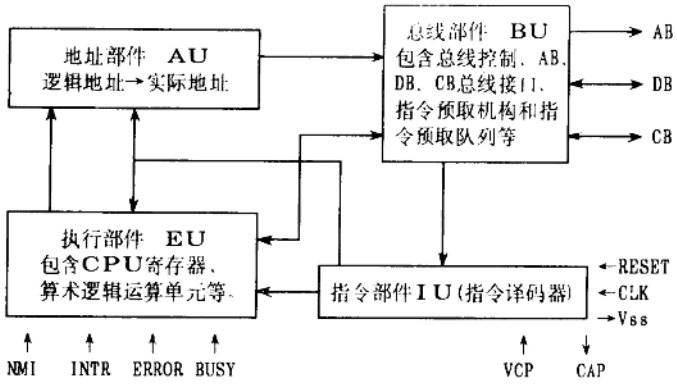

图 0-2 8086 CPU 内部结构示意图

各部件的作用是：

(1)地址部件 AU 从 EU 接收段前缀和段内偏移，把逻辑地址转换为实际地址。

(2)总线部件 BU 把实际地址送到 AB；从 EU 接收数据送入 DB；从 DB 取数据(和指令)及从 CB 接收控制信号，送往 IU 或 EU；转送 IU 发出的控制信号。

(3)指令部件 IU 从 BU 接收指令，经过译码，向 EU 发出命令或经过 BU 向 CB 发出控制信号。

(4)执行部件 EU 执行指令规定的运算，接收中断信号等，并做出相应的反应。

0.1.2 CPU 寄存器

执行部件内部包含多个寄存器，直接受 CPU 控制，有快速的处理能力。8086/80286 是 16 位微处理机，通用寄存器、指令指针寄存器、标志寄存器和段寄存器皆为 16 位。80386 是 32 位的微处理机，把通用寄存器、指令指针寄存器、标志寄存器扩展为 32 位，增加 2 个 16 位的段寄存器。对于应用程序设计来说，我们关心的是上述的 CPU 寄存器，在本书以后的所有各章，将频繁地与这些寄存器打交道。另外 80286/80386 有控制寄存器、段描述寄存器，80386 还有系统地址寄存器、调试寄存器和测试寄存器。由于本书不涉及系统程序设计，所以也很少使用这些寄存器，仅在使用时给出必要的说明。

图 0-3 至图 0-6 分别给出了 8086 四类 CPU 寄存器的描述，以下分别说明各个寄存器的作用。

0.1.2.0 通用寄存器

8086/80286 有 4 个 16 位的通用寄存器被命名为 AX、BX、CX、DX，每个可以存放 16 位的数据。还可以分为 8 个能够单独使用的 8 位寄存器，被命名为 AH、BH、CH、DH 和 AL、BL、CL、DL，每个都可以存放 8 位的数据。另外 4 个 16 位的通用寄存器，名字分别为 SP、BP、SI、DI，用于保存 16 位的数据或偏移。

80386 有 4 个 32 位的通用寄存器，每个可以存放 32 位数据；低位部分命名为 AX、BX、CX、DX，可以作为 4 个独立使用的 16 位寄存器；AX、BX、CX、DX 还可以分为 8 个能够单独使用的 8 位寄存器，被命名为 AH、BH、CH、DH 和 AL、BL、CL、DL。另外还有 4 个 32 位通用寄存器，其低位部分命名为 SP、BP、SI、DI，可以作为独立使用的 16 位寄存器。

每个寄存器保存一定长度的数据。例如 8 位寄存器 AL 包含 8 位数据，最低位是 0 位，最高位为 7 位。又如 16 位寄存器 AX 保存 16 位的数据，最低位是 0 位，最高位为 15 位。图 0-3 上部所示数字是边界的位号，位号较小的位在左边。注意，单独使用 8 位寄存器 AL～DL 和 AH～DH 时，其位号都是从 0 开始，最高位为 7；图中标明的 8 和 15 仅仅表示 8 位寄存器 AH～DH 占用 16 位寄存器 AX、BX、CX、DX 中的 8～15 位。

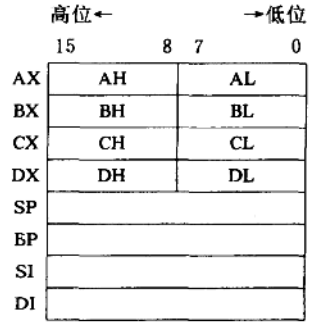

图 0-3 8086 通用寄存器

在以后的章节，经常把 8 位寄存器称为字节寄存器，把 16 位寄存器称为字寄存器。

通用寄存器还有一些独特作用：

AX，AL：**累加器** 在乘除法、某些串操作、I/O 指令中，均要使用累加器。LAHF 指令使用 AH，XLAX 指令使用 AL。

BX：**基址寄存器** 在 8086 系统中，只有 BX 和 BP 可作为基址寄存器。

CX：**计数器** 在 LOOP、JCXZ 指令中使用该寄存器。

DX：**数据寄存器** 在 16 位的乘除法指令中使用 DX。在某些 I/O 指令中用 DX 保持 I/O 地址。

SP：**栈指针寄存器** 在 8086 系统中，SP 作为栈指针寄存器，有其它寄存器所不能取代的特殊作用。

BP：**基址指针寄存器** 8086 系统使用 BP 作为基址指针寄存器，有时还称为栈框架指针，用于方便地存取栈内的数据。

SI：**源指针寄存器** 8086 系统使用 SI 作为源指针寄存器。

DI：**目的指针寄存器** 8086 系统使用 DI 作为目的指针寄存器。

如果可能，把这些作用延伸到允许使用其它寄存器的场合，将有益于阅读程序。例如在很多情况下，可以使用任意一个通用寄存器作为计数器，但应该优先采用 CX。

0.1.2.1 指令指针寄存器

指令指针寄存器用于保存下一条要执行的指令的段内偏移，改变该寄存器就是改变指令

的执行顺序。

8086/80286 IP:**指令指针寄存器** 16 位,段的最大长度为 64K 字节。

0.1.2.2 段寄存器

8086 系统的地址(是内存的逻辑地址,而不是物理地址)由两部分构成:段前缀和段内偏移。段前缀存在于段寄存器,段内偏移存在于指针寄存器、变址寄存器或其它的某个地方。指令地址的段前缀一定在代码段寄存器,栈地址的段前缀一定在栈段寄存器。而数据地址的段前缀可以在任意的段寄存器。

图 0-4 8086 指令指针寄存器

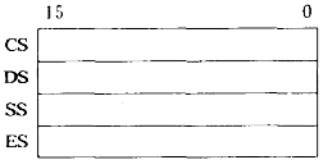

图 0-5 8086 段寄存器

8086 有 4 个段寄存器,其作用是:

CS:**代码段寄存器** 装有代码地址的段前缀

DS:**数据段寄存器** 装有数据的段前缀

SS:**栈段寄存器** 装有栈的段前缀

ES:**扩展数据段寄存器** 装有扩展数据的段前缀

由段前缀和段内偏移共同组成内存逻辑地址,而逻辑地址经过转换才能转化为物理地址,CPU 执行的转换方法为:

步骤 1. 先将 16 位的段前缀扩充为 20 位数,低 4 位均补 0。

步骤 2. 将步骤 1 得到的 20 位数加 16 位的段内偏移,得到 20 位的实际物理地址。

例如:段前缀和段内偏移分别为 0F111H 和 1234H,则实际地址为 0F1110H+1234H,即 0F2344H。

一般来说,在编制应用程序时,无需考虑转换的过程,只需关注逻辑地址即可。

0.1.2.3 标志寄存器

标志寄存器存放微处理机的有关信息,8086 标志寄存器名字为 FLAGS,16 位。

根据用途分为两类:

(1) 状态标志,共 6 位:

CF:**进位标志** 如果加减指令执行时有进位或借位,则 CF=1;否则为 CF=0。移位和循环指令也要用到该位。

PF:**奇偶标志** 如果逻辑指令执行结果中 1 的个数等于偶数,则 PF=1;否则为 PF=0。

AF:**半进位标志** 如果 BCD 数加减指令执行时最低 4 位有进位或借位,则 AF=1;否则为 AF=0。AF 亦称为辅助进位标志。

ZF:**零标志** 如果算术逻辑指令执行时结果为零,则 ZF=1;否则为 ZF=0。

SF:**符号标志** 算术指令执行的结果的最高位送入 SF,表示结果的符号。

OF:**溢出标志** 有符号数的算术运算的结果,若溢出则 OF=1;否则 OF=0。

状态标志表示算术或逻辑指令执行结果的特征,在讲述算术逻辑指令时给出更详细的描述,在讲述条件转移指令时将详细说明如何使用这些状态标志,在以后的程序设计中将频繁地使用它们,尤其是 CF 和 ZF。