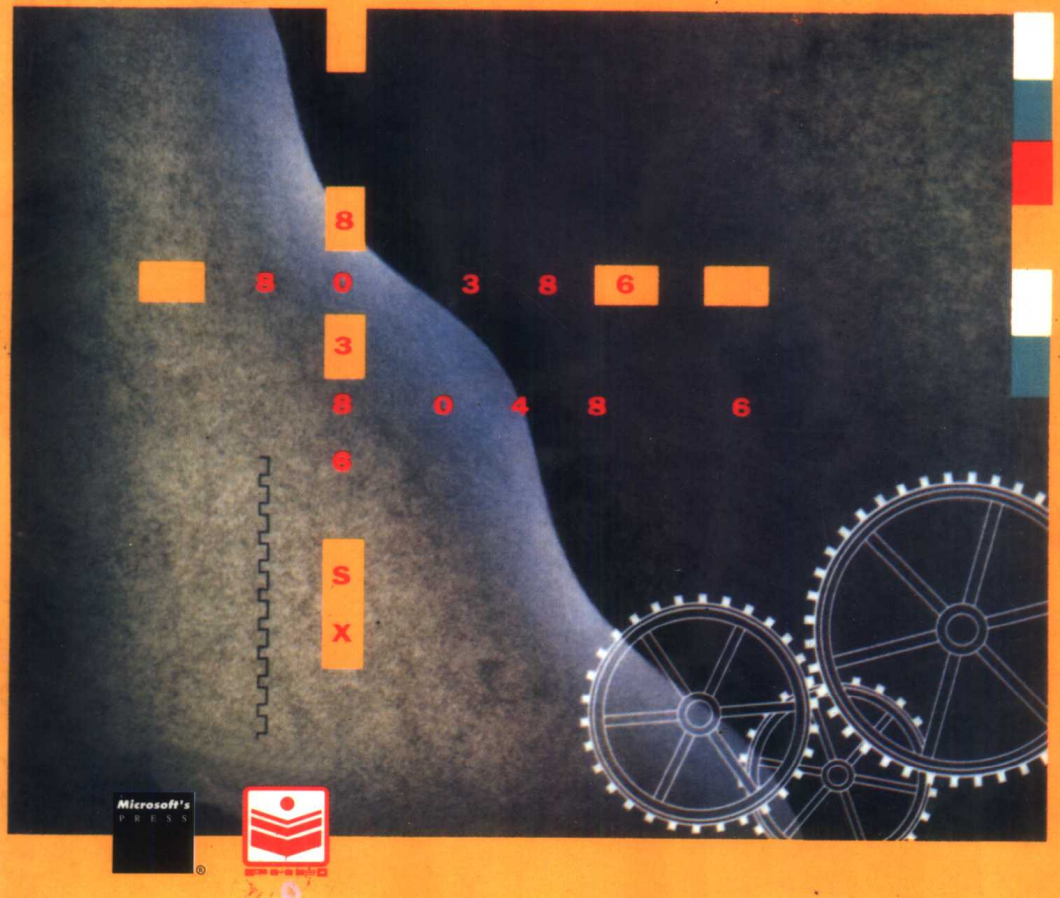
Microsoft's

80386/80486 编程指南

〖美〗Ross P.Nelson 著
田学锋 周豫滨 等 译



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

80386/80486 编程指南

[美] Ross P. Nelson 著

田学锋 周豫滨等 译

电 子 工 业 出 版 社

(京)新登字 055 号

内 容 简 介

本书共分三部分，第一部分包括第一至七章。首先概述了 x86 微处理器家族的历史，后继各章讲述了 80386/80486 处理器的各部分结构。第二章讨论了 CPU 的组织，第三章描述了基本存储器结构。第四章介绍了基本指令集和浮点指令集。第五章解释了保护方式操作，第六、七章讲述了分页扩展存储器系统的原理以及工作于 80486 下的高速缓存原理，讨论了 80286 在实址方式下与前期微处理器的兼容性，同时还包括了虚拟 8086 方式和保护方式的兼容性。第二部分——第八章提供全套的指令集引用。

第三部分——附录提供了机器码与指令集的对照表。本书的重点在于编程方面并且相当多的内容讨论了保护方式操作。这将帮助读者了解各种操作系统的设计。

80386/80486 编程指南

[美] Ross P. Nelson 著

田学锋 周豫滨等 译

责任编辑：赵 平

特约编辑：王 东

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

北京市顺新印刷厂印刷

*

开本：787×1092 毫米 1/16 印张：27.75 字数：710 千字

1994 年 11 月第一版 1994 年 11 月第一次印刷

印数：5000 册 定价：43.00 元

ISBN 7-5053-2675-9/TP・829

译 者 的 话

《80386/80486 编程指南》是系统设计者以及编制底层程序的用户所必不可少的指南。本书着重于 80386/80486 处理器的编程，并且花费相当多的内容讨论了在保护方式下的程序编制。书中提供了指令集参考以及机器码与指令的对照表，这将有助于读者理解和掌握 80386/80486 处理器的底层编程。

本书由田学锋翻译第 1、2 章；代桂玉翻译第 3、4 章；梁洁翻译第 5、6 章；苏宝文翻译第 7 章；周豫滨翻译第 8 章；刘冀伟翻译附录 A、B；章世松翻译附录 C～F。

本书由田学锋校阅。

由于我们水平有限，时间仓促，译文中难免有不准限之处，诚挚地希望读者予以指正。

译者

引　言

自从在个人计算机发展初期引入 8080 芯片以来，Intel 80386 可能是讨论最广泛的中央处理单元(CPU)芯片。本书的第一版探索了 80386 的功能。从那时起，Intel 公司使用同样的基本结构引入了三种附加的处理器。现在，处理器的 80386 家族包括原始的 80386、80386SX、80376 和家族的最新、最快的成员——80486，在本书中将讨论这些处理器之间的差异。

第一章描述了 x86 微处理器家族的历史。后继的各章讲述了 80386/80486 处理器结构的各个部分。在第二章中讨论 CPU 的组织。第三章描述了基本的存储器结构。第四章介绍了基本指令集和浮点指令集。第五章解释了保护方式操作。第六章讲述了分页扩展存储器系统的原理以及工作在 80486 下的高速缓存原理。第七章主要描述了 80286 在实址方式下与前期微处理器的兼容性，同时还讨论了虚拟 8086 方式和保护方式的兼容性。最后，在第八章中提供了全套的指令集引用。

本书的焦点完全集中在编程方面。这里不讨论处理器的硬件功能，除非这些功能与具体的指令相关，如果您对这些处理器中某个硬件特性感兴趣，那么可以从 Intel 公司获取对应的数据表和参考手册。

为了从本书获得最大的收益，您应该熟悉计算机系统。特别是，理解二进制和十六进制以及某些处理器的机器语言编程将会对您阅读本书有所帮助。

本书的相当多的内容讨论保护方式。尽管编写应用程序不必理解这个功能，但是，了解保护方式以掌握系统设计者在实现 OS/2、Microsoft Windows、PC－MOS/386 和 UNIX 操作系统时所做出选择十分重要。

本书中使用的规范在下面概述。如果您熟悉其它 Intel 微处理器，那么或许已经熟悉了这些概念。

数据格式

以三种进制格式表示数字：二进制(以 2 为底数)，十进制(以 10 为底数)，以及十六进制(以 16 为底数)。

假定除了带后缀“B”(表示二进制)或“H”(表示十六进制)的数字以外，其它所有数字都是以 10 为底数的。例如：

1AH＝26＝0011010B

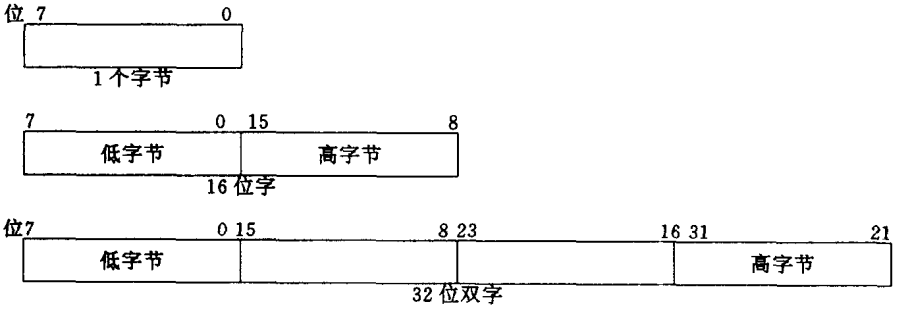
数据类型

最常用的数据类型是 8 位、16 位和 32 位量。在本书中，一个 8 位量称为一个字节，一个 16 位量称为一个字，一个 32 位量称为一个双字。这种术语表示不常用，因为计算机的标准数据项尺寸通常称为一个字。例如，在 Digital Equipment 公司的 VAX 计算机中，一个 32 位量是一个字，一个 16 位量是半个字。这同样适用于 Motorola 68000 系列和 IBM 370 和 390

大型机。

尽管标准 80386/80486 操作数尺寸是 32 位，但是，由于 32 位处理器是 16 位处理器 8086 和 80286 的后代产品，所以 Intel 保持了早期处理器的命名规范。这简化了来自 8086 或 80286 的运行软件的编程并且允许用户使用同样的汇编程序生成适用于四种处理器的代码。

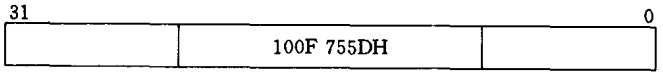
在 x86 家族中最小的可寻址项是字节。所有其它数据项都可以细分成字节。处理器以先存低字节、后存高字节的方式存储大的数据项，如下图所示：



假定在存储器中存储 32 位值 100F755DH，起始地址是 10。那么，单个存储器字节是：

地址	10	11	12	13
内容	5DH	75H	0FH	10H

把字和双字分成有序字节并且说明按单元处理量的方面并不复杂。例如，可以如下表示上面给出的值：



当完成小于一个单个字节项的操作时——例如，对应一个单个位或位段——处理器总是最少从存储器中取出 1 个字节量。

汇编程序记法

一条可执行指令是可由 CPU 内逻辑进行译码的二进制格式。一条指令长度可以从 8 位至 128 位。因为使用二进制格式编制一个程序冗长乏味，所以编程人员使用了一种称为汇编程序的程序。最简单的汇编程序是一组关键字和符号，并且汇编语言把这些关键字和符号翻译成一条指令。关键字和符号集称为汇编语言。典型地，汇编语言中一条指令和一个实际的机器指令间存在一一对应的映射关系。汇编程序可以按如下方式使用一条指令：

```
ADD     EBX,5
```

其含义是“EBX 寄存器中值加 5 并且结果存回 EBX 中”，然后将其翻译成位模式：

```
10000000110000110000010 1B
```

指令的名称，称为助记符，它通常占用指令行的第一个字段。后继字段是指令的操作数

并且采用了多种格式。最简单的是一个数值，如上例中的 5。寄存器名称是操作数的另一种格式，在中括号中的一个表达式，如[EBP+2]，意指操作数是一个存储器访问。

在本书，我们使用了标准 Intel 助记符。可是，请注意，一个助记符不必指定一条指令的准确编码。例如，“增量”指令拥有一个任意操作数可以被编码的通用格式，并且指令 INC EAX 将编码为 FFH 00H。同时也存在一个完成通用寄存器加 1 操作的单字节指令。在这种情况下，INC EAX 指令编码为 40H，汇编程序对应任何给定的助记符通常选择指令的最紧凑格式，而且它们的执行效果是相同的。

设定位的讨论中也使用了一种通用规范。当分配数值 1 给一个存储单元的位时使用术语“置位”，当分配数值 0 给一个存储单元的位时使用术语“复位”。

句法

本书使用了下列句法：

操作符	含义
+	加
−	减
×	乘
/	除
~	非
=	等于
! =	不等于
|	或
^	异或
&	布尔与
>	大于
<	小于
>>	右移
<<	左移
⩽	小于或等于
⩾	大于或等于
←	赋值

32 位指令集

80386、80386SX 和 80486 支持与早期 Intel 处理器(16 位 8086 和 80286)兼容的几种方式。可是，本书主要关注新的功能并且不讨论 8086 和 80286 的 16 位结构，尽管它们是 80386/80486 处理器的功能子集。使用 80386 或 80486 做为早期处理器替代产品的编程人员可以继续使用 8086 或 80286 的参考资料。

操作系统服务

80386 家族结构相当复杂，并且不可能希望一个杰出的程序发挥出 CPU 所有功能。在不同时间，将讲述“操作系统将…”或“在此处，操作系统…”。在这些情况下，我们将不涉及具体的操作系统；代之的，我们将突出讨论由操作系统而不是由应用程序实现的一个功能。

目　录

第一章　80x86家族结构的发展历史

尽管我已经使用个人计算机十多年了，但是“计算机系统”这个词带给我的大脑的是马里兰州立大学中Campus实验室的硬件设备的印象。在那里，位于玻璃墙后面、空调屋内的是大学计算机系统Siggie(一台Xerox Sigma 7)。装在几个冰箱大小的机柜内，Siggie满足整所大学的计算需求。

1986年，当Siggie仍然看做珍品时，源于早期技术的80386微处理器已成为桌面计算机的心脏，其计算能力远远大于Siggie。现在，更快的80486仅是未来处理器家族中的一员，Intel公司声称至2000年为止它们将继续推出几种改进型芯片成员。

1.1　第一代成员

80486是由Intel公司制作的微处理器家族的最新成员。Intel公司于1971年开始向微处理领域投资，当时，主要为一家日本公司开发一个用于新型计算器“大脑”的用户电路。Intel设计人员Ted Hoff设计了一个可编程、通用计算电路，并且使4004芯片成为了现实。不久又推出了4040和8008芯片，可是，这些芯片缺乏许多今天我们所了解的微处理器特性。

1.2　8080

按照流行说法，导致个人计算机工业诞生的芯片是8080，它由Intel公司于1974年引入。1975年9月在“今日电子”(Popular Electronice)中的一篇文章首次向市场引入了“个人”计算机的设想，并且，正如文中指出的，这已成为历史。8080是Altair和IMSAI系统的CPU(中央处理单元)，这些系统是计算机系统的先驱。可是，Intel并没有长期独占市场；Motorola公司引入了6800，MOS Technology引入了6502，8080的两位设计人员离开Intel来到Zilog公司并且为之设计了Z80。与6800和6502不同(其结构完全不同于Intel处理器)，Z80与8080兼容，但拥有一个扩展指令集并且速度快两倍。争夺CPU霸权的战斗从此开始了。

8080是一个8位机器——也就是说，它一次处理8位数据。它拥有一个单累加器(A寄存器)和六个辅助寄存器(B、C、D、E、H和L，如图1-1所示)。这六个寄存器可以用于8位算术运算操作或组合成对(BC、HL)保存16位存储器地址。一个16位地址允许8080访问2^{16}位，或64个字节(KB)的存储器。

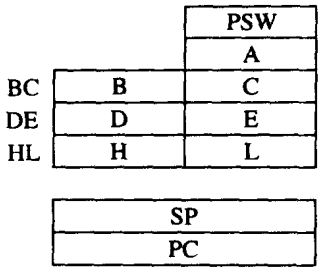


图 1-1　8080 寄存器集

Intel 也开发了 8080 的改进型芯片——8085，它与 8080 兼容并且性能更好、硬件接口更简单。

1.3　8086

在 1978 年，在其它制造商生产更快、功能更强的芯片压力下，Intel 转向一个 16 位结构。8086 做为 8080 微处理器的替代产品出现，并且，尽管指令集是新的，但它仍然保留与 8080 指令集的兼容性。图 1-2 说明了 8086 新寄存器是如何映射成 8080 寄存器组的。

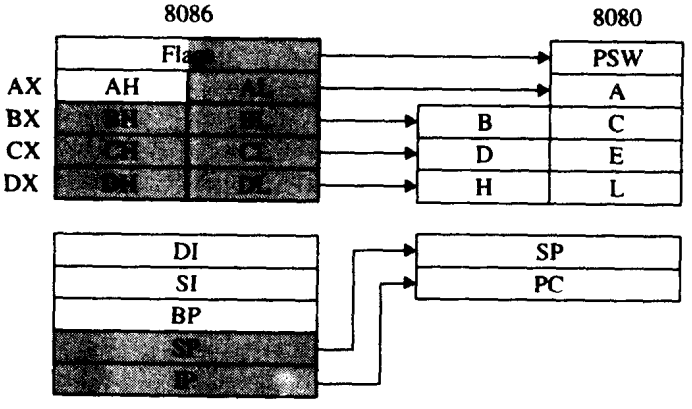


图 1-2　8080－8086 寄存器组映射

对应 8080 编写的程序不能在 8086 上运行；可是，几乎每条 8086 指令都对应一条 8080 指令。最坏的情况是一条 8080 指令可以由两个或三个 8086 操作模拟。一个 Intel 翻译程序可以把 8080 汇编程序转换为 8086 汇编程序，对应 8086 的 Microsoft 公司的 BASIC 和 MicroPro International 公司的 WordStar 的第一版是经 Intel 翻译程序从 8080 系统中移植的。兼容性方面的注意已经特征化了个人计算机市场的现状。微处理器的每代新产品都可以运行对应上代产品编写的软件。

除了提供软件兼容性以外，Intel 也对高级语言支持感兴趣。在 Intel 公司，几乎所有编程工作都是在 PL/M 语言下完成的。Intel 认为 PL/M 或 Pascal 语言将成为占支配地位的个人计算机开发语言，因此，Inel 分配许多 8086 寄存器用于特殊用途，如图 1-3 所示。

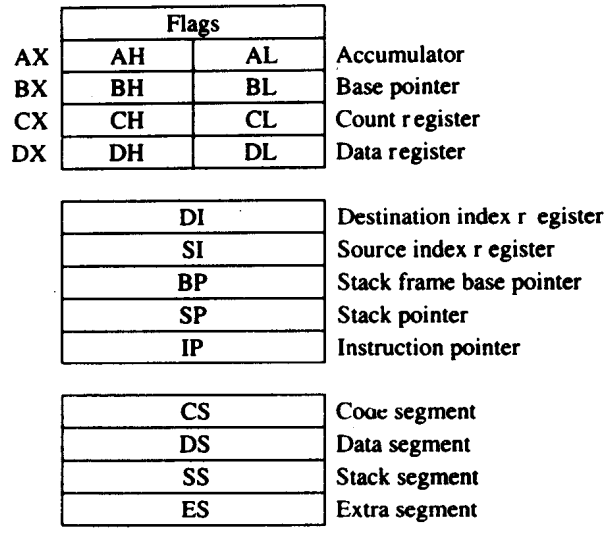


图 1-3　8086 寄存器组

下面两个例子显示使用的专用寄存器。图 1-4 给出了 Pascl 高级语言使用堆栈指针(SP)和基指针(BP)寄存器的方法。

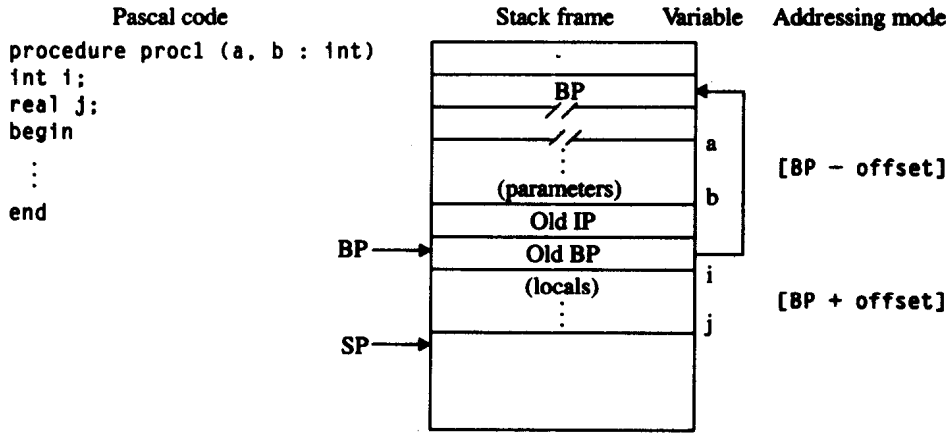


图 1-4　子程序前后关系

在 Pascal 程序中，当前执行的子程序前后关系在堆栈中保持。通过调用例程提供给子程

序的值(参数)在堆栈中位于第一,调用例程的存储 IP 是位于第二,调用例程的存储 BP 位于第三。前后关系也包含由子程序使用的任何临时或局部变量的堆栈空间。访问参数或局部变量与 BP 的当前值相关联。

Pascal 赋值在图 1-5 中陈述。因为必须拷贝一个完整的记录,所以编译程序生成使用 SI、DI 和 CX 寄存器的一个块移动指令。

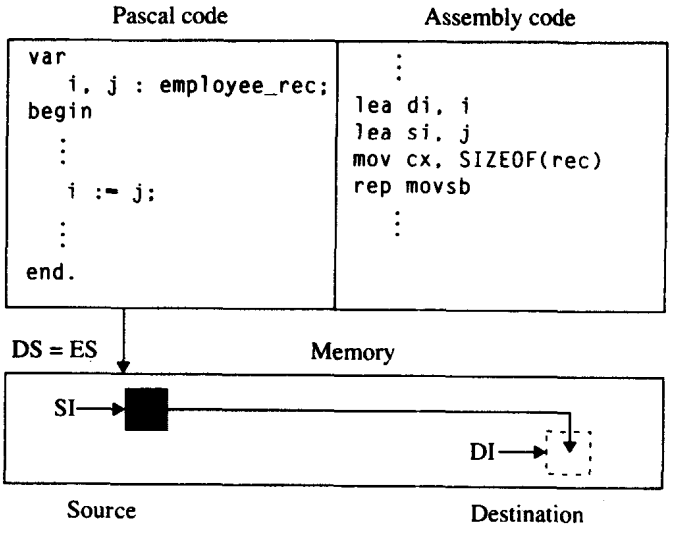


图 1-5 块移动

专用寄存器的优点是它允许 Intel 以紧凑和有效存储器方式编码指令。操作码准确地指定所发生的操作,例如,MOVSB 指令中,不必指定三个操作数(源、目的和计数)。因而,MOVSB 操作码仅为一个字节。专用寄存器的缺点是:如果使用 SI 或 DI 并且想要完成一个 MOVSB 指令,那么不能使用另一个寄存器。

8086 也把段概念引入微处理器世界。段是一个以一个固定地址开始的存储器块,这个地址由段寄存器中的值决定。这个概念结合了 8080 兼容性,由于强加的限制,所以它或许是 8086 最藐视的功能;即每段 64KB,等同于一个 8080 地址空间。使用段,当扩展(使用多个段)芯片可以寻址的存储器时,软件可以保持适用于 8080 的 16 位寻址方式。8086 提供四个段寄存器,它们可以指向 1 兆字节(MB)地址空间内的任何地址。它们如下定义:

CS——程序段寄存器:所有调用和跳转均涉及程序段内的位置。

DS——数据段寄存器:大多数存储器访问指令涉及数据段内一个偏移量。

SS——堆栈段寄存器:所有 PUSH 和 POP 指令访问堆栈段内数据。此外完成任何与 BP 寄存器相关的存储器访问也引向堆栈段。

ES——附加段寄存器:本段指定某个字符串处理指令中的目的段。

应用程序管理存储器的方法(存储器模型)在一个程序中通常是一致的。当 Intel 引入 8086 时,假定了三个存储器模型,如图 1-6 所示。

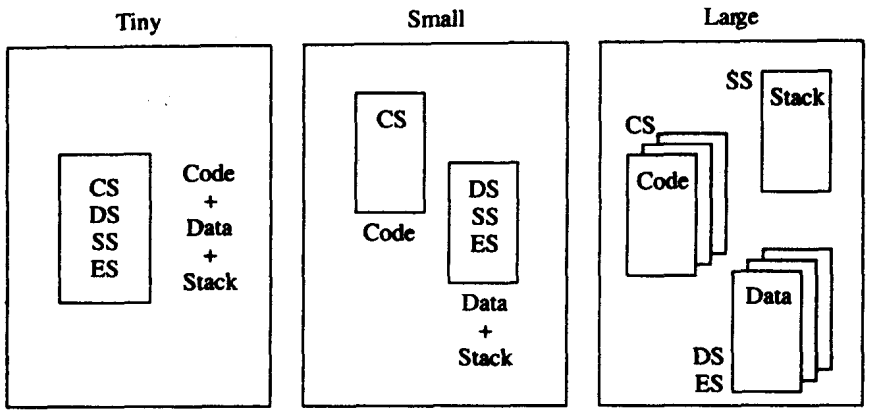


图 1-6　存储器模型

微模型模仿 8080 地址空间。程序段和数据段在同一存储器区域，并且程序限为 64KB。小模型由于允许程序尺寸加倍，所以是流行的模型。通过使用独立的程序和数据段，程序可以扩展至 128KB 并且仍然保留 16 位寻址。大存储器模型允许使用多个程序块和数据段，在这种模型中，可以使用完整的 1MB 处理器地址空间。

当 1978 年推出 8086 时，大多数个人计算机存储器段仍限于 64KB，几乎没有人认识到这么快 64KB 段限制将成为一个严重的问题。尽管大模型允许程序填充完整的 1MB 8086 地址空间，但使用大模型仍然意味着使用 32 位指针。在一台 16 位机器上，32 位指针需要一个尺寸上和大多数编程人员不愿意赋出的性能上的不利后果。至 80 年代初期，1MB 限制已造成混乱。带有“紧凑”和“中”名称的附加存储器模型引入以便最优化特殊编程需求的性能。

8086 家族的其它处理器成员是 8088、80186 和 80188。在 8086 出现一年以后推出的 8088 拥有同样的 16 位内部结构，但拥有一个严格的 8 位外部总线。8088 可以同 8086 一样运行同一程序，但是通常运行速度要慢 30%。当 IBM 在其 PC 和 PC/XT 中选择了 8088 时，它获得了广泛的成功。80186 和 80188 于 1982 年推出。这些处理器保留了同样基础，并且包含了直接存储器访问(DMA)控制器、片上计数器/定时器和一个简化的硬件接口等功能，它们操作比 8086/8088 快得多，并且成为控制器应用程序的流行选择。

1.4　8087

CPUs8086 家族的创新部分是协处理器。ESC 或指令的协处理器换码类仅在 8086 上生成一个存储器地址。此外，可以生成专用 CPUs 监视指令流并且观看 ESC 序列，如图 1-7 所示。当检测到 ESC 时，协处理器可以把换码译码为其本身的一条指令并且完成一个 8086 在其本身不能有效完成的功能。

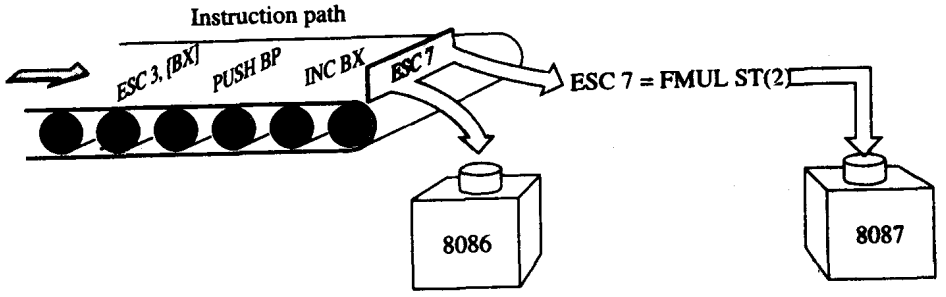


图 1-7　8086 协处理器接口

对应 8086 开发的协处理器仅为 8087。8087 实现一个浮点指令集，其精度为 80 位。Intel 与 IEEE(Institute of Electrical and Electronics Engineers)及加州大学伯克利分校合作设计了一个浮点算法表示；它既灵活精度又高。由于经过 IEEE－734 标准正式公布，所以这种表示和其数字特征已经被正式采用。

8087 专属于 8086，并为其流行做出了贡献。包括 8086 和 8087 在内的桌面计算机比单独使用 8086 的机器可以完成更多的科技领域的工作。在硬件中实现浮点功能改善了基于现有软件例程的数学计算性能。可是，8087 是产生 64KB 段尺寸问题的一个例子。当科学家和工程师拥有处理实际世界的计算能力时，经常需要处理大型数组。64KB 段限限制了双精度浮点数的矢量不超过 1024 单元。这种限制对软件功能的影响不久就显现出来，但大存储器模型很难为程序使用并且运行很慢。

1.5　80286

Intel 于 1982 年推出下一代主要产品——80286，80286 与 8086 家族兼容，并且提供了重大的性能改进。它推出两种操作方式：实址方式和保护方式。实址方式模拟 8086，它为缺省方式。新方式称做保护方式。在保护方式中，80286 支持 8086 指令集，但是对于控制存储器访问的段寄存器内容拥有一个新解释。

尽管在保护方式下实现的操作系统不同于实址方式下的操作系统，但是应用程序可以按照在两方式下运行而开发。这些两种方式的应用程序的设计需要应用程序了解某个存储器限制。

不幸的是，在基于 8086 的机器上占统治地位的操作系统是 MS-DOS，它没有设置应用程序寻址存储器的限制，并且保护方式证实了与大多数 MS-DOS 应用程序的不兼容性。因而，许多年来 80286 通常按一个快速 8086 处理，因为没有人知道如道有效使用新功能——保护方式。这是非常不幸的。因为保护方式把物理可寻址存储器的数量从 1MB 扩展到 16MB，允许实现虚拟存储器并且提供在多任务或多用户环境下任务的独立性。

UNIX 版本运行在保护方式下，但 UNIX 在 80286 上并不成功，因为竞争产品通常在更