PC MAGAZINE

PROGRAMMER'S TECHNICAL REFERENCE:

The Processor and Coprocessor

80x86处理器和80x87协处理器大全

Robert L. Hummel

朱莉　张龙译
朱家维　审校

電子工業出版社

80x86 处理器和 80x87 协处理器

大　全

[美]R. L. Hummel　编著
朱莉　张龙　　译
朱家维　　审校

电子工业出版社

（京）新登字 055 号

内 容 提 要

本书完整地介绍了 Intel 80x86 和 80x87 处理器和协处理器的演变、结构和编程。具体型号包括 8086、80286、80386、80486 处理器及 8087、80287、80387 协处理器。内容既有 CPU 的内部结构，又有处理器出错信息表，还有未写入文案的操作码和指令。

80x86 处理器和 80x87 协处理器大全
〔美〕R. L. Hummel 编著
朱莉 张龙 译
朱家维 审校
责任编辑：魏冬 陈金凤（特约）
*
电子工业出版社出版（北京市万寿路）
电子工业出版社发行 各地新华书店经销
电子工业出版社计算机排版室 排版
北京市顺义县天竺颖华印刷厂印刷
*
开本：787×1092 毫米 1/16 印张：36.5 字数：1190 千字
1994 年 3 月第 1 版 1994 年 3 月第 1 次印刷
印数：5000 册 定价：60.00 元
ISBN 7-5053-2278-8/TP・630

献给我的妻子(我最大的支持者),她曾跟我说,“你骗谁呀?还有件事你肯定做不到。”

献给沙菲,温迪和戈文多琳,她们凭借爱心帮我录入并且从不抱怨。

内容一览表

致　谢

这本书的规模和复杂度是许多人献身和努力工作的结果。我要特别感谢 Ziff-Davis 出版公司的 Cindy Hudson，Sheila McGill 和 Cheryl Holzaepfel。显而易见，他们已打破常规而创作出了一个更好的作品。

我也非常幸运地与在个人计算机领域中杰出的人物认识并共同工作。这些长期从事前沿技术研究的先生们非常慷慨地把他们的时间、专长和优质的软件和硬件产品与我分享。我特别感谢 Frank van Gilluwe（V Communications 公司），Frank Grossman（NuMega Technologies 公司），Rob Larsen（Larsen Computing 公司），Paul Passarelli（Crescent Software 公司），Brett Salter（The Periscope Company 公司）和 Bob Smith（Qualitas 公司）。在附录 D 中有关于他们的产品的更多信息。

我同时也要感谢在犹他州奥莱姆市的 WordPerfect 公司的人们。这本书的手稿是在 DOS 5.0 下运行 WordPerfect 5.1 而写成的。若没有抵制住其他产品和环境优势的诱惑，若没有 WordPerfect 毫不费力地编辑表格和复杂等式的能力，这本书的大部分，特别是附录部分几乎是不可能制作出来的。

引　言

我相信这位先生是个计算机行家。显然他可能有一到两台计算机——谁知道呢？

Henry Darrow, Host of JackPot Bingo
Los Angeles, California 1986

我这本书的目的是提供一个有关 Intel 处理器和协处理器系列的完整的、毫无废话的参考指南。为了达到这一目的，在本书中写入了 CPU 的内部结构的信息，列出处理器的错误以及未公布的操作码和指令。这本书的主题很简单，“这里有所有你所需要的在 80x86 和 80x87 下编程的技术参考资料”。

我的读者对象是使用各种语言和技巧编程的程序员。让我们面对这样一个事实：所有的人最终都用汇编语言编程。几乎所有 BASIC，Pascal 和 C 编译器都提供调用汇编程序或用几行汇编语言助记符的某些方法，这种情况并非偶然。最新的高性能的 C＋＋编译器也在软件包中包容了汇编程序！即使这样，汇编语言还是很流行和有用的。不管编程人员选择何种语言，他们需要关于芯片和它如何工作的好的参考资料。

作为《PC Magazine》的高级技术编辑，我必须熟悉大量的种类繁多的硬件、软件和兼容产品以保证我们的汇编语言工具能在读者的计算要上成功地运行。这常常意味着要翻阅大量参考手册，如硬件指南、OEM 文档和隐藏的错误清单。象这本参考书那样列出各个芯片的细节和错误将使这项工作变得更容易些。

如何使用这本书完全取决于您以及您的编程经验。从头至尾阅读本书以获取处理器及协处理器的工作概念，指令集和操作方式。或者挑选您所感兴趣的章节，或针对您要解决的问题来参阅本书（例如在第 12 章中的公式表示出如何使用 FPTAN 函数来导出其他三角函数）。

对任何编程书来说，快速获取参考资料是很重要的。你会发现这本书包含了整个 Intel 80x86 和 80x87 指令集以及由其他厂商生成的兼容芯片的指令集的综合的指南。我相信附录 A 和 B 的设计会使您很容易查找到一条指令和所需的信息。

当然您必须懂得什么是二进制数、十六进制数和十进制数，并且能在不同的基之间转换简单的数值（例如，二进制 1101＝十进制 11＝十六进制 B）。在引言中，回顾了这些原理。

在这本书中的所有信息在不同的计算机系统上得到证实，包括 80486DX，80386DX（早期和近期和版本），80286，8088，V20，80387SX，80287XL 和 80287。在本书中提到的指令时间是 Intel 芯片最近版本的数据。其他芯片的时间可从它们的制造商处获得。

那么，这本书是为谁编写的呢？它是为你编写的！它是为那些想了解处理器工作原理与方式的人们而写的；为那些爱好创造、乐于学习新技术的人们而写的；为那些能在人群拥挤的房间里敏锐地听到低语“未写入文档”的人们而写的。

常规约定

在这本书中，定义和解释了一些新术语和符号。最后，以下是一些约定，用来表示处理器操作、数据结构、指令编码和数字过程的细节。这些约定在这部分中解释。

数字系统

当讨论处理器的操作时，常常需要用三种不同的数字系统作解释：二进制、十进制、十六进制。对于在本书中出现的所有数值数据用下面的常规约定表示。

十进制数 除非特别指定，在文章中的所有数都是十进制数(基为 10)。例如，如果句子中说有 256 个可用中断向量，数值 256 被认为是十进制数。十进制数用在注释和程序的源代码清单中。由 DEBUG 输入或显示的数总是十六进制的数，不会是十进制数。

十六进制(Hex)数 用大写或小写字母 h 做后缀的数是十六进制数(基为 16)。例如，在指令 MOV AH,21h 中的数值 21h 是十六进制数，它等价于十进制数 33。

如果十六进制数的起始数是字符 A—F 中之一，则许多汇编语言要求在它前面加个 0。如果不这样做，例如指令 MUL BH，可被看作是访问 BH 寄存器或十六进制数 Bh。

一个字节的值可方便地用两个十六进制数表示。中断号，寄存器值和存储器地址通常用十六进制表示。注意在 Intel 文档中用十进制数表示它的中断号。所以，当在 Ineel 文档中指出的通用保护异常，即中断 13，应被作为是中断 Dh 而非 13h。

对十六进制数来讲有两个重要例外。DEBUG 总是操作十六进制数，所以在用 DEBUG 输入或显示时不用加后缀。输入 DEBUG 时用后缀 h 会导致 DEBUG 产生错误而忽略这一输入。(这是因为字母 h 不是一个有效的十六进制数。)表示指令和操作码的程序清单也省略 h 后缀。

以(*段*:*偏移量*)表示的存储器地址仅在偏移量部分的后面加后缀 h，但这两个数都被认为是十六进制数。例如，分段地址 0400：0048h 应被理解为段地址值是 0400h，偏移量是 0048h 的地址。

二进制数 文本中的二进制数通常有一个大写或小写字母 b 做后缀。在许多情况下，用二进制比用其他计数制更方便。例如，指定逻辑操作的位屏蔽值常用二进制数。

注意在表示指令的编码，FLAGS 寄存器和处理器数据结构等二进制数时常不用后缀 b。这些结构通常用各位数字或各位组合在一起表示。

位和字节顺序

80x86 系列采用 *little endian* 数据存储方法(该名称源于斯威夫特的《格列佛游记》一书)。这种方法把字节的高序位放在高地址。操作数的字节，字或双字部分的基地址都是相同的。

这本书中的图，例如寄存器布局，结构和位域总是表示为右边是低存储器地址。在结构中，当你向左或向图顶部移动时，地址就增加。位域的位位置从右向左增加。

位序号，结构和存储器地址采取以 0 为底值的策略识别。在任何结构中的最低有效元素总是编号为 0。因此一个字节中的各位应从 0 到 7 编号。处理器可产生的最低分段存储器地址用 16 位位移量是 0000：0000h，用 32 位位移量是 0000：00000000h。

保留的和未定义的域

作为处理器结构的代表，例如 FLAGS 寄存器，某些位或位域没有特别地定义为一已知

值。这些域被标记为保留的，并且通常被涂暗以表示他们不能被用作任何未公布的用途以及他们的值是不可靠的。保留域中的值在不同的处理器间，甚至在同一处理器的不同版本间不是一直不变的。（每当推出一个新型芯片时，用于制造该芯片的掩膜总是因发现错误而被更改和修正。）为了某种目的而使用保留域，当你把代码转移到不同处理器上时会导致不兼容的问题。

如果一个域由一个操作设置成一个值，但返回值在操作的前后没有什么意义，则这一域被称为未定义的。例如，在 DIV 操作之后，在零标志和操作结果之间没有相互关系。所以，虽然零标志被 DIV 指令改变了，但它的值被称作是未定义的。

操作数顺序

80x86 的 Intel 汇编语言设计者采用模仿一般数学标记法的从右到左的指令语法。首先总是表示目的操作数，随后是源操作数。例如，表达式 A＝5。这用汇编语言语法表示为 MOV A，5。在这两种情况下，首先表示目的操作数 A。这种语法用于许多 80x86 汇编语言中，DEBUG 中和本书中。

目　录

第一章　微处理器家族概述

Intel 对微处理器的定义是:“由一个或多个的集成电路元件组成的系统,它采用半导体技术和数字逻辑能在很小的规模上完成大型计算机的功能”。Intel 已极为成功地把这一定义从理论转变成事实,对于成千上万的个人计算机制造商和用户来说,微处理器的定义即为“由 Intel 生产的芯片”。

相对来讲,微处理器还是很年轻的。但是与大多数技术革新相比,个人计算机的革新速度是非常之快的。本章将简要介绍 Intel 的微处理器系列从一开始到当前技术状态的变革史。对每一种处理器的主要特点和性能都进行了比较,同时概述了与之相关的数字协处理器系列。

§1.1　Intel 微处理器的演变

Intel 开发的第一片微处理器芯片是 4004,主要用于嵌入的应用。在 1971 年年推出 4004 时,Intel 将它销售给生产工业和商业产品的制造厂,并断言,制造商若在他们的产品设计中包含该芯片,则可提高产品的实力和实用性。

用今天的标准来衡量,4004 非常简单。例如,它只有一个 4 位宽的外部数据总线,这就意味着每次只能在处理器与外部存储器之间传送 4 位数据。由此可以想象出,它的速度是非常慢的,因为许多传送操作不得不采取多个步骤才能完成。为了提高性能,Intel 于 1972 年又推出了 8008。8008 芯片仅是 4004 芯片 8 位数据线的版本,但它代表了 8 位微处理器时代的开始。

两年之后,在 1974 年,传奇的 8080 芯片诞生了。8080 保留了 8008 的结构,但已设计成为一个通用的微处理器。这种适应性强的芯片的出现,对工业界来说非常重要。Intel 8080 的成功激发了开发 8 位处理器的竞争,如 Zilog 的 Z80 和 Motorola 的 6800。

§1.1.1　8086 和 8088

1978 年诞生的 8086 成为微处理器发展的又一里程碑,它是 Intel 的第一个 16 位微处理器,并且是 80x86 系列的第一个成员。众所周知,该芯片象 iAPX 86,有充足的 16 位寄存器,一次可传送 16 位数据到外部存储器,且可直接寻址一兆字节的物理存储器。伴随着处理速度的提高,8086 在性能上有极大的增强。对于系统程序员来说,8086 在指令集和结构上与 8080 的类似使得源代码的改变并不是那么费劲。

紧接着 8086 的是 8088。这种版本的芯片保持了在内部采用 16 位寄存器和数据总线,与外部接口采用 8 位数据总线的结构。8088 允许系统设计者在通用的比较廉价的 8 位外设基础上开展他们的新设计。当然,8088 也是 IBM 为它初期 PC 所选用的芯片。它的广泛应用使之不仅成为工业标准,而且也成为世界标准结构。

在谈到 Intel 芯片的同时,提及一下 NEC V20 和 V30 芯片,它们可以引脚兼容地依次替代 8088 和 8086。除了具有透明的兼容性外,它们比 Intel 的同类产品运行速度快且功耗低。通过在它们自身的芯片上包含专用的硬件来执行许多 8086 寻址方式中用到的有效地址计算,使基