386/486汇编语言精要

邓洪涛　编著
沈官林　　审

清 华 大 学 出 版 社

(京) 新登字 158 号

内 容 简 介

汇编语言程序设计历来都是计算机专业人员的基本功，人们往往在了解、掌握了汇编语言之后，对计算机的理解才发生了质的飞跃。但遗憾的是，由于汇编语言比较难学，令不少计算机爱好者们望而生畏。

本书是一本可读性极强的汇编语言教科书，不仅深入浅出地介绍了 80386 和 80486 的宏汇编程序设计技术，而且生动地讲解了计算机的一些基本原理，内容编排新颖别致，是一本很好的汇编语言入门书。

本书适于广大电脑爱好者学习，也可作为各高等院校和各类计算机应用培训班学习汇编语言的教材。

图书在版编目 (CIP) 数据

386/ 486 汇编语言精要/ 邓洪涛编著 . —北京: 清华大学出版社，1995.3

ISBN 7-302-01828-6

Ⅰ.38… Ⅱ. 邓… Ⅲ. 微型计算机-汇编语言 Ⅳ. TP312

中国版本图书馆 CIP 数据核字 (95) 第 04993 号

出版者: 清华大学出版社 (北京清华大学校内，邮编 100084)
责任编辑: 姜峰
印刷者: 北京密云胶印厂
发行者: 新华书店总店北京科技发行所
开 本: 787×1092 1/ 16 印张: 17.25 字数: 408 千字
版 次: 1995 年 6 月第 1 版 1995 年 6 月第 1 次印刷
书 号: ISBN 7-302-01828-6/ TP·820
印 数: 0001—8000
定 价: 16 . 00 元

引　　言

第二次世界大战中，美国阿伯丁弹道研究室应陆军的要求，每天需要提供 6 张炮击表，而每张炮击表都要计算几百条弹道，实验室已聘请了 200 多名技术员，但人手仍远远不够。1942 年 8 月，埃克特等人写了一份《高速电子管计算装置的使用》的备忘录，它实际上成为第一台电子计算机 ENIAC 的初步方案。

1944 年夏的某一天，ENIAC 设计组的戈德斯坦在阿伯丁火车站邂逅著名科学家冯·诺伊曼。当时冯·诺伊曼正在等车，两个人闲聊起来，谈话非常轻松，可是，当戈德斯坦谈到他们正在制造每秒运算 330 次的电子计算机时，气氛顿时严肃起来。当时，冯·诺伊曼正遇到原子核裂变反应过程的大量困难计算，涉及到几十亿次初等算术运算，几百名女计算员加班演算仍然不能满足需要。身为弹道研究所和制造原子弹工程顾问的冯·诺伊曼马上理解到戈德斯坦等人工作的深远意义。形势的需要以及用数学方法解决科学问题的强烈欲望，使冯·诺伊曼迅速投入到紧张的计算机制造的工作中去。

由于天赋以及在科学实践中积累起来的丰富经验，使冯·诺伊曼具有非凡的洞察力。他首先想到，设计计算机的核心是正确提出它的逻辑结构。为此，他提出了存储程序的重要思想。之后，他又和布克斯、戈德斯坦一起，于 1946 年 6 月提出了更完善的设计报告：《电子计算机逻辑结构初探》。他们认为，计算机必须具有保存初始数据、程序及中间和最后结果的存储器，具有算术运算单元，具有控制单元，以及输入和输出数据和程序的媒介，即输入输出设备。这便提出了这个我们十分熟悉的计算机逻辑功能图。

与此同时，冯·诺伊曼把崭新的科学思想马上讨诸实践。1946 年，世界上第一台真正的电子计算机 ENIAC（恩尼亚克，即电子数字积分计算机）正式运行。它占地 170 平方米，用了 18000 个电子管，总重量 30 吨，功率 140 千瓦，速度为每秒运算 5000 次。更为重要的是，它能按照人所编好的程序自动地进行计算。

第一台电子计算机的建成以及冯·诺伊曼等人的理论思想轰动了科学界，从此，人类的科技史进入了一个新时代。

当时，不少有识之士都认为，计算机是本世纪最重要的发明之一，将对人们的生活产生重大影响。可是，即便是最具创新精神、最富有想象力的人，也没有预料到计算机在几十年的时间里会发展得这么快，甚至深入到千家万户。而 1946 年曾有人大胆地估计：整个英国只需要 5 台计算机就绰绰有余了。

如今，地球上几乎每诞生 1 个人，就会诞生 1 台计算机。它们对人们的生产和生活起着不可估量的作用，每年都有几千万台新型计算机投入使用。

进入 90 年代，随着微型计算机事业的迅速发展，大批 80 年代的 Intel 8086/ 8088 微处理器已被淘汰，而新一代的 Intel 80386/ 80486 微处理器正在全世界广泛应用。考虑到新的微处理器在功能上已大大超过 8086/ 8088，而我国目前与之相适应的新教材还不多，因此，作者深感编写这么一本教材的必要性，这就直接导致了本书的产生。

在编写本书的过程中，得到了很多人的帮助，要想一一列举是不可能的。但是，有五个人却不能不提到：一个是中国科学院计算中心的李山林同志，他为作者提供了无私的帮助，没有他的帮助，本书是不可能完成的。另外四个是国防科工委的邢志杰和齐力峰同志及清华大学的沈官林、刘卫东同志，他们帮助作者详细地审阅了全部稿件，没有他们的大力帮助，本书同样是不可能完成的。作者在此对他们以及其他帮助过作者的朋友表示深深的感谢。

因作者水平有限，经验不足，因此在编写过程中一定存在不少错误，欢迎大家批评指正。

另外，书中源程序代码在清华大学出版社软件部有售。

邓洪涛

1994.11 于中国人民大学信息学院

目　　录

第一章　从 8086 到 80486

1.1　8086 以前的时代

本世纪 50 年代，所有的电子设备（无论是收音机还是电视机）都是笨重的电子管设备，那个时代的计算机被称为第一代计算机。50 年代末期，晶体管和其它固态电路开始取代电子管，利用这种技术生产的计算机被称为第二代计算机，运算速度已达每秒几百万次，体积、重量、耗电以及造价都大为减少。

第三代是中小规模集成电路计算机。第一台这样的机器制成于 1962 年，1965 年开始批量生产。这时已有操作系统，小型机广泛应用，有了终端与网络，运算速度已达每秒千万次。

由大规模集成电路组成的计算机称为第四代。一般认为是从 1972 年开始。这时计算机的体积与成本大幅度地减少，可靠性大为提高，速度每秒已超过 1 亿次。这时，许多部件可以集成到一块很小的硅晶片上，制造微型计算机的条件终于成熟了。

早在 1969 年初，在美国加利福尼亚州的硅谷，刚刚营业几个月，只有 12 人的 Intel 公司得到一家日本计算器公司为计算器生产芯片——集成电路的委托，公司的工程师霍夫和梅泽共同设计了一组芯片，Intel 公司将这种装置称为 4004，因为 4004 是这一装置能替代的晶体管的大致数目，是其复杂程度的一种衡量标准。实际上，这就是人类历史上第一种微处理器，它能处理 4 位二进制数据。

那么，什么是微处理器呢？一个微处理器是指一个单一的硅晶片，它通常由控制部件、算术逻辑部件、寄存器、标志、输入/输出接口等部件组成，也就是通常称为 CPU (Central Processing Unit 中央处理单元) 的部分，故微处理器就是微型计算机的 CPU，是微型计算机的核心部分。将微处理器加上电源部件、控制板、输入输出设备如磁盘驱动器、显示屏、键盘等等，就构成一台微型计算机了。但这只是微型计算机的硬件部分，完整的微机还应包括软件，如操作系统、各种编译系统等等。有时我们把上述硬件系统和软件系统的总和称为微型计算机系统。

1971 年，小具规模的 Intel 公司又推出 8008 微处理器。4004 和 8008 就是第一代微处理器，它们都是为专门的应用而设计的，4004 主要用于计算器，8008 则主要用于计算机终端设备。但是，Intel 公司虽然开创了微处理器的时代，却没有得到足够的重视。直到 1974 年，当 8008 成长为第二代微处理器 8080 时，才引起了计算机工业界的震动。8080 是第一个精心设计，可以广泛地在各方面应用的微处理器，可以同时处理 8 位二进制数据，它很快风靡世界。美国各地的电脑爱好者纷纷成立业余计算机小组，他们买不起昂贵的“传统的”计算机，只能买微型计算机，并为微机编制软件。当时年轻的比尔·盖茨和保罗·艾伦曾为微机编写了第一种 BASIC 语言，在业余爱好者中广为流行。许多 Intel 以外的公司开始制造 8080 片子，而且有一些公司制造出了 8080 的增强型产

品，如 Zilog 公司的 Z - 80。Intel 也在 1976 年推出了增强型产品 8085。但这些微处理器的基本特点并没有多大改变。

尽管微处理器已发展到第二代，但是不少传统的计算机大公司仍然认为微处理器是不值得发展“小玩意”，计算机界的超级巨人 IBM 公司就是这么看的。所以，微型计算机工业的规模仍不大，每年只能卖出几千台微机。

但是，在 1976 年秋天，刚刚成立不久、推出几十台苹果—I 型微机的苹果电脑公司，令人震惊地推出了功能卓越、外表美观的苹果—Ⅱ型机，引起了计算机工业界的又一次极大震动。苹果—Ⅱ型电脑极为畅销，每三到四个月，产量就翻一番，1977 年一共卖出创纪录的 35000 台，超过前一年总销售量的四倍。苹果公司声名大震。

苹果公司的巨大成功，在计算机工业界引发了两件大事。第一，他们使 Intel 公司感到了强烈的竞争压力。于是，Intel 公司不久于 1978 年推出了第三代微处理器——8086，而后又推出了介于 8086 和 8080 之间的芯片——8088。第二，IBM 公司终于注意到了微型计算机广阔的发展前景，开始在这一领域倾注巨资。

1981 年 8 月，IBM 公司宣布它的第一台个人计算机问世，称为 IBM PC 机。它采用 Intel 公司主频为 4.77MHz 的 8088 作为中央处理器，64KB 的内存，以后又采用 8086 作为中央处理器，采用比尔·盖茨和保罗·艾伦成立不久的 Microsoft 软件公司研制的 MS - DOS 作为操作系统（IBM 称为 PC - DOS)。IBM PC 机的问世，使微机制造者、软件编制者、零售商以及微机购买者市场发生了彻底的、无可挽回的变化。从此，IBM 公司成为微型计算机工业的领导者，Intel 公司、Microsoft 公司则在这棵大树下迅猛成长壮大，经过十多年的发展，成为今天敢和 IBM 分庭抗礼的巨人。

以下，就三个方面较详细地讲述微型计算机的知识，使读者对微机的软、硬件及工作原理有较详细了解，为今后的学习做准备。作者假定：读者已经具有数制（二进制、八进制、十进制、十六进制)、编码（原码、反码、补码）等有关知识，对 DOS 也有初步了解，会简单使用。

1.2 8086

IBM - PC 机从它诞生的第一天起，就显示出令人激动的魅力，PC 机的问世标志着“个人计算机”时代的到来。

1.2.1 微型计算机的组成

描述一台计算机的一种方法是描述组成该计算机的功能部件，这些部件和它们之间的相互作用称为该计算机的结构。微型计算机虽然在具体的结构上比大中型计算机简单得多，但仍然是计算机，仍然是按照冯·诺伊曼提出的逻辑结构设计的。今天的微型计算机有五个关键部分：中央处理器 CPU、存储器（即内存)、输入/ 输出系统（即 I-uput/ Ouput 系统，简称为 I/ O 系统)、磁盘存储器和程序。微机中还有其他部件象电源、母板、总线（也是 I/ O 接口的一部分）及插件框架等。图 1.1 显示了基本部分之间的逻辑关系。

图 1.1　微机的逻辑结构

所有的微型计算机都是如此，相信读者对提到的有关概念并不陌生。我们的学习重点是 CPU 和存储器（内存）的结构，它们的联系非常紧密。要想讲述 80386/ 80486 的结构特性，我们必须从 8086 及有关基本概念开始，这是学习 386/ 486 的基础。

1.2.2　8086 成功的奥秘

8086 究竟提供了什么，使它能获得如此成功？要想了解答案，我们必须先了解 8080/ 8085 的不足之处。

首先，8080/ 8085 只能处理 8 位字长的数据，数据处理范围只有 0～255 或 - 128～127，远远不能满足需要，而要处理比 8 位字长更长的数据，必须由几个 8 位组来构成，而每一组又必须分开操作，这样就增加了处理时间。8086 则是基于十六位字长操作的芯片，数据处理范围从 0～65535（2^{16} - 1）或 - 32768～32767，比 8080/ 8085 大大增加了。而且 8086 也保留了处理 8 位数据的能力，从而使短数据仍能被 8086 有效地处理，具有良好的向上兼容性（即 8080/ 8085 上编制的程序仍然可以在 8086 上运行）。

其次，8080/ 8085 的直接寻址范围是 64K 字节（1K = 1024），而 8086 的直接寻址能力达 1M（M 表示兆，1M = 2^{20} = 1048576）字节，这在当时是不可想象的，因为在 1980 年，一台相当贵的小型机只有 512～1024KB 内存，价值数百万美元的大型机至多只有 2MB 内存。内存的大幅度增加，有利于编制更复杂的程序。

第三，8080/ 8085 缺少乘法和除法指令，还缺少带符号数操作，因而使用很不方便，8086 则提供了以前缺少的这些指令。

第四，8080/ 8085 的寻址方式不能支持把高级语言写的程序转换成有效的 8080/ 8085 代码，而 8086 的寻址方式则适合于高级语言的处理。

第五，随着系统日益复杂，单靠一个处理器来执行系统全部功能的方法越来越行不通，但是，8080/ 8085 并不能与其他处理器合作，而 8086 则可以在多处理器环境下运行，尤其是可以与两个出色的协处理器，即数据处理器 8087 和 I/ O 处理器 8089 紧密配合，大大提高了 8086 的性能，其中 8087 专门处理数据尤其是实数运算，运算速度比 8086 快一个数量级，8089 则专门用于 I/ O 处理，较好地解决了输入/ 输出的“瓶颈”

问题，大大提高了 8086 的效率。

总之，8086 的设计相对于 8 位机有了很大的突破，这个大约有 29000 个晶体管的集成电路的性能较 8 位芯片大约提高了 10 倍。

1.2.3 8086 的存储器结构

8086 的存储器是一个多至 2^{20} 的 8 位数量的字节（byte）序列。每一个字节单元分配一个唯一的地址，地址用无符号数表示，二进制从 0000 0000 0000 0000 0000 到 1111 1111 1111 1111 1111，用十六进制表示即从 00000 到 0FFFFFH，如图 1.2 所示。

十六进制地址	2 进制地址	存储器
00000	0000 0000 0000 0000 0000	
00001	0000 0000 0000 0000 0001	
00002	0000 0000 0000 0000 0010	
00003	0000 0000 0000 0000 0011	
.	.	
.	.	
.	.	
FFFFF	1111 1111 1111 1111 1111	

图 1.2 存储器地址

在存储器中，任何两个相邻的字节称为字（Word），任何四个相邻的四字节被称为双字（Double Word），即：一个字是 16 位，一个双字是 32 位。

在一个字中，每一个字节都有各自的地址，那么，字的地址该如何确定呢？我们规定两个字节中地址较小的一个做为该字的地址。因此，字的地址有奇地址和偶地址之分，如图 1.3 所示。

一个字有 16 位，具有较高存储器地址的字节含有该字的高 8 位，具有较低存储器地址的字节含有该字的低 8 位。比如，当把字 83A5（十六进制）存入存储器中时，先把 A5 存在较低的地址上，再把 83 存在较高的地址上。也就是先存低位字节，再存高位字节。这种情况可用图 1.4 来说明。

图 1.3 字在存储器中的例子

双字的情况与字类，如图 1.5 所示。

8086 有些指令是访问（即读或写）字节的，有些则是访问字的。在同一时间，8086 传送到存储器或从存储器中取出的信息数量总是 16 位，并且该 16 位总是在存储

图 1.4　字的存储情况

图 1.5　双字的存储情况

器中以偶地址开头的两个字节的内容。在字节指令的情况下，这些位中只有 8 位是有用的，其余 8 位被忽略。由于 8086 读写字时总是从偶地址开始，因此，将字的地址选成偶数，可以提高处理器的效率，这样读写一个偶地址的字时，可以用一个存储器访问周期来实现。否则，对开始于奇地址的字操作指令，就必须对两个连续的偶地址作两次存储器访问，忽略各自不需要的一半，对剩下的一半还要作一些字变换才能得到。各种字和字节的读出例子如图 1.6 所示。

图 1.6　各种字和字节的读出例子

同样，读写双字时，双字的地址最好能被 4 整除，否则也会需要额外的总线周期。

实际上，我们编制程序时并不涉及到这些细节。这些访问，处理器都自动实现。提到这些只是为了增加读者对存储器的了解。

1.2.4 存储器的分段

由于 8086 可以寻址 2^{20} 个存储器字节，所以，字节和字的地址必须表示成 20 位的二进制数。但是，8086 是设计来执行 16 位运算的，它只能处理 16 位长的字，所以，就用了一个巧妙的机构来表示地址，这就引出了存储器分段的概念。

我们可以把 1 兆字节的存储器想象为任意数量的段，其中每一段最多可含有 2^{16} (65536) 个字节，而且，每一段必须开始于一个能被 16 整除的字节地址（即该字节二进制地址的最低 4 位全是 0)，这个地址被称为段地址。这样，一个段的段地址具有十六进制数 XXXX0 的形式。段中某一字节或字的绝对地址减去段地址，称为这一字节或字在这个段中的偏移地址。可以看出，段地址虽然是 20 位二进制数，但低 4 位全是 0，而偏移地址由于小于 2^{16}，所以可以用 16 位二进制数表示，这样，我们就有办法用两个 16 位二进制数来表示段和段中操作数的地址，这就又引出寄存器 (Register) 的概念。

所谓寄存器，就是这样的存储单元：和存储器 (内存) 不一样，它们在 CPU 中，直接参与运算，存取速度比存储器更快。CPU 在处理数据过程中往往用寄存器存放中间结果，运算的初始值也必须从内存中调入寄存器，运算完毕后再将终值返回到内存，也可以用寄存器表示系统的状态和内存中段的基地址及段中数据的偏移地址。寄存器的设置可以大大提高运算速度。

在 8086 CPU 中，有 13 个内部寄存器，这些寄存器都是 16 位的。其中有 4 个寄存器：CS、DS、SS 和 ES，用来表示段的基地址。由于 20 位段地址的低 4 位全为 0，所以把段地址的高 16 位存入它们即可。在 8086 中，有三种类型的段：用来存放程序码的代码段、用来存放程序所需有关数据的数据段和用来存放由于子程序调用等原因需要保存特殊数据的堆栈段。CS 是 Code Segment 的缩写，表示代码段；DS 是 Data Segment 的缩写；SS 则表示 Stack Segment；ES 表示 Extra Segment，即附加段。在 8086 中，CS 用于存放当前代码段的段地址，SS 则存放当前堆栈段的段地址，DS 和 ES 可以存放两个数据段的地址，一个是“数据段”，一个是“附加段”，本质上都是数据段，只是为了提高速度，为了编程需要才设立了两个数据段寄存器。

图 1.7 所示的是 4 个段寄存器分别指示着存储器中 4 个当前段的例子，在这个例子中，假定每 1 个逻辑段的长度都 64KB。例如，假定代码段寄存器 CS 中含有的十六进制数值是 D018，就表示代码段的基地址是 0D0180H（在表示十六进制数时，若第一个数字是表示十进制数 10 到 15 的 A、B、C、D、E、F，则常在它们之前加一个 0，以便把数字和字符区别开，数字末尾的 H 表示它是一个十六进制数)，这样，由于该代码段的长度为 2^{16} (10000H) 个字节，所以该代码段的最后一个字节的地址将是 0D0180H + 0FFFFH = 0E017FH。注意段与段之间是可以重叠的。

那么，如何通过段地址和偏移地址来寻找该段里的字节和字呢？由于 16 位的段地

图 1.7　8086 的段寄存器

址实际上表示低 4 位是 0000 的 20 位物理地址，所以 8086 处理器实际上是把 16 位偏移地址加上附加有最低 4 位 0 的 16 位段寄存器的内容来产生 20 位的物理地址，操作的原理如图 1.8 所示。

图 1.8　20 位物理地址的形成原理

例如，一个字在数据段 DS 中的偏移地址为 18H，DS 中的十六进制数是 7000H，这样，这个字的物理地址就是 70018H。数学运算如下:

```
   7000 0H
+   001 8H
-----------
   7001 8H
```

那么，我们如何通过计算机具体“看到”存储器中的情况呢？

打开计算机电源，计算机启动完毕之后显示提示符 C >，等待用户的下一步动作(如果使用软盘，则出现提示符 A > 或 B >)。检查在 C 盘中或 C 盘的 DOS 子目录中是否有文件 debug.exe。debug 调试程序是 DOS 系统盘所提供的程序之一，假设 debug.exe 是在 C 盘的根目录下。

在 DOS 提示符后，键入“ debug”，然后按回车键: