99
TP313
90
2

中等专业学校
电子信息类　规划教材

80X86 汇编语言程序设计

唐延玲　王小刚

35

电子科技大学出版社

3 0033 0606 9

声　明

本书无四川省版权防盗标识，不得销售；版权所有，违者必究，举报有奖、举报电话：(028) 6636481　6241146　3201496

中等专业学校
电子信息类　规划教材

80X86 汇编语言程序设计

唐延玲　王小刚

出　版：电子科技大学出版社　(成都建设北路二段四号)　邮编 (610054)
责任编辑：吴艳玲
发　行：新华书店发行
印　刷：成都青羊福利东方彩印厂
开　本：787×1092　1/16　印张 21　　字数 508 千字
版　次：1998 年 6 月第一版
印　次：1998 年 6 月第一次印刷
书　号：ISBN 7-81043-937　5/TP·413
印　数：1　4000 册
定　价：23.00 元

内容简介

全书共分十章，主要内容包括，IBM PC 系列计算机中80X86 CPU 的指令系统、宏汇编语言、程序设计的方法和技术、系统中断处理技术、定时器程序设计、显示器程序设计、键盘程序设计、磁盘文件管理程序设计、汇编语言与高级语言的连接等。同时集习题、实验、试题库系统为一体，便于教学、自学、自查、测评。每章程序举例完整实用并全部在微机上运行通过，实用性较强。

本书可供各类中等专业学校、职校、技校计算机及非计算机专业作为教材，亦可供广大计算机爱好读者自学、参考。

出版说明

为做好全国电子信息类专业“九五”教材的规划和出版工作，根据国家教委《关于“九五”期间普通高等教育教材建设与改革的意见》和《普通高等教育“九五”国家级重点教材立项、管理办法》，我们组织各有关高等学校、中等专业学校、出版社，各专业教学指导委员会，在总结前四轮规划教材编审、出版工作的基础上，根据当代电子信息科学技术的发展和面向21世纪教学内容和课程体系改革的要求，编制了《1996—2000年全国电子信息类专业教材编审出版规划》。

本轮规划教材是由个人申报，经各学校、出版社推荐，由各专业教学指导委员会评选，并由我部教材办协商各专指委、出版社后，审核确定的。本轮规划教材的编制，注意了将教学改革力度较大、有创新精神、特色风格的教材和质量较高、教学适用性较好、需要修订的教材以及教学急需，尚无正式教材的选题优先列入规划。在重点规划本科、专科和中专教材的同时，选择了一批对学科发展具有重要意义，反映学科前沿的选修课、研究生课教材列入规划，以适应高层次专门人才培养的需要。

限于我们的水平和经验，这批教材的编审、出版工作还可能存在不少缺点和不足，希望使用教材的学校、教师、同学和广大读者积极提出批评和建议，以不断提高教材的编写、出版质量，共同为电子信息类专业教材建设服务。

电子工业部教材办公室

前　言

本教材根据电子工业部《1996－2000年全国电子信息类专业教材编审出版规划》精神，由中专计算机专业教学指导委员会编审并推荐出版。

“80X86汇编语言程序设计”课程是计算机及其应用专业人才必修的专业基础课，它不仅作为后续专业课的基础，而且还是专业人员作为与硬件系统密切联系的编程手段，是从事计算机使用、维护、组装、调试等工作所必备的重要技能技术知识。

本书作为中专层次所用计算机系列教材之一，为充分体现汇编语言程序设计的特色，力求注重实用性、实践性，在内容的编排上既突出了80X86系统汇编程序设计的一般方法，又突出其控制硬件的灵活高效的特点。全书后面六个章节中结合应用安排了丰富的例题，而且所列程序完整实用，即读者可按照原样键入某程序后便能够独立运行，同时可举一反三，即学即用。学生可一边学习，一边上机操作，便于从实践中巩固、理解理论知识，掌握编程技术和技巧。

为遵循中专教学规律，在内容的选取、概念的引入、文字的叙述以及例题和习题（含实验题）的选择等方面，力求内容通俗易懂，由浅入深，结合实际，学以致用，便于自学，够用就好的原则。同时为适应计算机的快速发展，及时补充了80386/80486的先进的技术知识。其次教材为适应各类学校及非计算机专业的选用，可方便地选择各章节的内容，进行压缩或扩充。

本课程是一门实践性很强的课程，既包含复杂的脑力劳动，又是一种极富有创造性的活动。因此，教学安排上若为80总学时左右，建议实验时数不得少于26学时。

本书每章后附有习题，并有配套的实验，同时建有完善的试题库系统，以提供现代化教学手段。

本书训练学生从掌握计算机系统的全面知识着手，使学生在较短时间内学会编制较高质量的实用程序，为迅速胜任计算机的使用、维护、维修等工作打下扎实基础。

本书由常州无线电工业学校唐延玲主编，常州无线电工业学校王小刚参编，由山东省信息工程学校张学金高级讲师担任主审，由常州无线电工业学校凌林海高级讲师担任责任编委。

唐延玲编写第一章～第五章，王小刚编写第六章～第十章。

在编写本书的过程中，山东省信息工程学校计算机系的有关领导和老师及本校的有关领导和老师提出了许多宝贵意见，给予了大力支持和帮助，在此谨致以诚挚的谢意。

由于作者水平有限，书中不妥或错误之处在所难免，敬请广大读者批评指出。

编　者

一九九八年三月十日

试读结束：需要全本请在线购买：　www.ert

目　录

第一章 基础知识

汇编语言是唯一能够充分利用计算机硬件特性的、一种面向机器的低级语言，它随机器结构的不同而不同。因此，要学习汇编语言，就必须首先了解机器有关的硬件结构、数据类型及表示方法等。本章将从程序设计者的角度出发，介绍有关的基础知识，叙述 Intel 80X86 系列主要微处理器的结构特征、数据通路、汇编语言所能寻址的寄存器以及对存储器的寻址能力等。

1.1 汇编语言

1.1.1 机器语言

人们为了让计算机按自己的意愿工作就必须与计算机之间交流信息，这个交流信息的工具就是计算机语言。目前所使用的计算机语言分三类：机器语言、汇编语言和高级语言。高级语言接近于自然语言，易学、易记，便于阅读，容易掌握，使用方便，通用性强，且不依赖于具体的计算机。但是计算机并不能直接识别高级语言，它只能直接识别在设计某机器时事先规定好的机器指令。

机器指令即为指挥计算机完成某一基本操作的命令。机器指令的一般格式为：

操作码	地址码	…	地址码

机器指令均由0 和1 二进制代码组成，通常分为两部分，操作码字段部分指出了运算的种类，如加、减、传送、移位等，地址码字段部分指出了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3，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作，则相应的机器指令为：

```
10000011
00000110
01100100
00000000
00000011
```

其中，第一、二行中的两个8 位二进制代码是操作码，表示要进行“加”操作，还指明了以何种方式取得两个加数；第三、四行中的两个8 位二进制代码指出了第一个加数(称目的操作数)所存放的偏移地址是100(或64H)，相加的结果也送入该存储单元中；第五行的8 位二进制代码指出了第二个加数(称源操作数)是3。

机器指令也常常被称为硬指令，它是面向机器的，即每台计算机都规定了自己所特有的、一定数量的基本指令，这批指令的全体即为这台计算机的指令系统；描述指令系统的语

试读结束：需要全本请在线购买： www.ert

言就称为机器语言。用机器语言编写的程序称为机器语言程序(或目标程序)。

机器语言程序能被计算机直接识别、理解并执行。

1.1.2 汇编语言

由于机器指令是用二进制表示的,编写程序相当麻烦,写出的程序也难以阅读和调试。为了克服这些缺点,人们想出了用助记符表示机器指令的操作码,用变量代替操作数的存放地址,还可以在指令前冠以标号,用来代表该指令的存放地址等。这种用符号书写的、与机器指令基本上一一对应的、并遵循一定语法规则的符号语言就是汇编语言;用汇编语言编写的程序称为汇编源程序。由此可见,汇编语言也是面向机器的语言。本教材中所介绍的是Intel 80X86 宏汇编语言。例如,对于前面的例子,用宏汇编语言来书写:

ADD WORD PTR [100],3

其中,ADD 为加指令的助记符;[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"说明了这个目的操作数是16 位二进制数;源操作数是3,相加的结果送入目的操作数所在的单元中。

由于汇编语言是为了方便用户而设计的一种符号语言,因此,用它编写出的源程序并不能直接被计算机识别,必须要将它翻译成由机器指令组成的程序,计算机才能执行。这种由源程序经过翻译转换成的机器语言程序也称为目标程序。目标程序中的二进制代码(即机器指令)称为目标代码。这个翻译工作一般都由计算机自己去完成,但人们事先必须将翻译方法写成一个语言加工程序作为系统软件的一部分,在需要时,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程序称为汇编程序。汇编程序进行翻译的过程叫做汇编。在这里,汇编程序相当于一个翻译器,它加工的对象是汇编源程序,而加工的结果是目标程序。它们三者之间的关系如图1.1 所示。

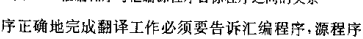

图1.1 汇编程序与汇编源程序目标程序之间的关系

为了能让汇编程序正确地完成翻译工作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,数据应放在什么位置,数据的类型是什么,留多少内存单元作临时存储区等。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使汇编源程序编写得像高级语言一样清晰、简洁,且容易使算法标准化、处理问题灵活,这样,不仅加速了程序的编写进程,而且有利于阅读、修改和调试源程序。Intel 80X86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.EXE 等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,所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仿。

与高级语言相比，汇编语言具有直接和简洁的特点，用它编制程序能精确地描述算法，充分发挥计算机硬件的功能。用它来编制过程控制、媒体接口、通信等方面的程序时，其目标程序简短，占用存储空间小，执行速度快，效率高。特别是有些用高级语言难以实现的操作，却能简单地使用汇编语言实现。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，汇编程序成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。

1.2　80X86 系列CPU 处理器简介

1.2.1　8086/8088 处理器

Intel 公司于1978 年推出了第一种高性能的十六位微处理器——8086。在以后的几年相继推出80286，80386，80486 等微处理器，以及与之相配套的支持芯片，形成了一个80X86 系列。其中Intel 8086 与8088 除硬件接口有些差别外，从软件设计角度看没有多大不同。因此，本节主要介绍Intel 8086 逻辑结构。其内部逻辑结构如图1.2 所示。

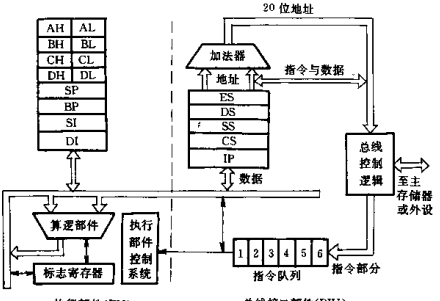

图1.2　8086 CPU 的内部逻辑结构

Intel 8086 功能上可分成总线接口部件 BIU（Bus Interface Unit）和执行部件 EU（Execution Unit）两大部分。

一、执行部件(EU)

执行部件负责指令的执行。它主要由寄存器组、算术逻辑部件、标志寄存器组成。从学习汇编语言程序设计的角度看,了解寄存器组的分工及标志寄存器的作用是很重要的,下面介绍主要寄存器组。关于标志寄存器将在1.5 节中专门介绍。

执行部件中含有8 个16 位的寄存器,这些寄存器属于CPU 的专用存储器,按其作用可将它们分为两组:数据寄存器组和指示器变址寄存器组。

1. 数据寄存器组(AX,BX,CX,DX)

数据寄存器主要用来保存操作数或运算结果等信息。它们的存在减少了为存取操作数所需访问总线和内存储器的时间,加快了机器的运行速度。其中AX 称累加器,BX 称基址寄存器,CX 称计数寄存器,DX 称数据寄存器。它们既可作16 位寄存器使用,又可按高8 位和低 8 位作8 位寄存器使用,因此,又可将这4 个寄存器分为两小组:H 组(AH,BH,CH,DH)和L 组(AL,BL,CL,DL)。除此之外,这些寄存器对于某些指令还有规定的作用。

2. 指示器变址寄存器组(SI,DI,SP,BP)

这4 个寄存器均为16 位寄存器,它们一般用来存放操作数的偏移地址,用作指示器或变址寄存器。其中,SP 称为堆栈指示器,BP 称为对堆栈操作的基址寄存器。SP 中存放的是当前堆栈段中栈顶的偏移地址,BP 中存放的是该段中某一存储单元的偏移地址。SI 和DI 除作一般指示器和变址寄存器外,在串操作指令中,SI 往往被规定用来作取源操作数的指示器,而DI 被规定用来作送目的操作数的指示器,因此,SI 称源变址寄存器,DI 称目的变址寄存器,它们中的内容是当前数据段或当前附加数据段中某一存储单元的偏移地址。当SI、DI、和BP 不用作指示器和变址寄存器时,也可以将它们当作数据寄存器使用,用来保存操作数和运算结果,但这时它们只能用来作16 位寄存器而不能是8 位。由于SP 是专用的堆栈指示器,它不能作数据寄存器使用。

二、总线接口部件(BIU)

BIU 负责与存储器、外部设备的接口。即8086 CPU 与存储器或外部设备之间的信息交换均由BIU 完成。其功能为负责从存储器的指定区域内取出指令,送到六个字节(8088 为四个字节)的指令流队列中排队,并负责从内存指定区域中取出指令所规定的操作数,传送给EU 去执行。这样,8086 的取指令与执行指令是分开进行的,其优点是在 EU 执行指令的过程中,BIU 可以预取多条指令放进指令流队列中排队,当EU 执行一条指令后,可立即执行下一条指令,从而减少了CPU 取指令的等待时间,提高了整个系统的运行速度。

BIU 中主要含有段寄存器组和指令指示器。

1. 段寄存器组(CS,DS,ES,SS)

这4 个段寄存器均为16 位。由于EU 所能提供的存储器地址是16 位的,而8086 访问1MB 存储空间却需要20 位地址,为了形成这20 位地址,在BIU 中设立了4 个段寄存器(CS,DS,ES,SS)。CPU 当前某一时刻可以直接访问4 个存储段:一个代码段、一个堆栈段、一个数据段、一个附加数据段,每个段最大可达64KB。由于这4 个段在当前可以接受CPU 的访问,为了与当前CPU 不可访问的段相区别,常将这4 个段称为当前代码段、当前堆栈段、当前数据段和当前附加数据段。它们的起始基地址分别保存在4 个段寄存器中,其中:CS 寄存器指示当前代码段,即它规定了现行程序段所在的存储区首址;SS 寄存器指示当前堆栈段;DS 寄存器指示当前数据段;ES 指示当前附加数据段,即DS 和ES 规定了现行程序中所用数

据的存储区首址。关于怎样形成20位物理地址的方式将在1.4.3小节中介绍。

2. 指令指示器(IP)

在BIU中,IP是一个很重要的寄存器,它总是保存着下一次将要从内存中取出的指令偏移地址(简称EA),其值为该指令到所在段段首址的字节距离。在目标程序运行时,IP的内容由微处理器硬件自动设置,程序不能直接访问IP,但有些指令却可使其改变,如转移指令、子程序调用指令等。

BIU根据IP的内容与CS寄存器的内容形成指令的物理地址,并根据该地址从内存中取出指令,送入指令队列机构排队,然后IP自动增量,形成下一个字节的偏移地址。EU按序从队列机构中取出一条条指令顺序执行。如果碰到要改变IP的指令(如转移指令),则清队列,直接按IP的新内容与CS寄存器的内容形成指令的物理地址从内存中取出指令,立即送EU执行,然后再从该地址开始,重新继续取指令填满排队机构。

* 1.2.2 80286微处理器

80286是Intel公司于1984年推出的新一代高性能的微处理器。它具有集成在芯片内部的存储器管理机构操作系统和任务程序与任务数据保密。并具有多用户多任务系统的较完善的处理能力。在10MHz的80286上的处理速度比5MHz的8086快6倍左右。

80286具有很强的寻址能力,并能以两种不同的方式运行(实地址方式和虚拟地址保护方式)。在实地址方式下,具有1MB的寻址能力;在虚拟地址保护方式下,能将每个任务的2^{30}字节的虚拟地址空间映射到2^{24}字节的物理地址空间中。即在这种方式下,具有16MB的寻址能力。

80286与现有的8086/8088具有软件向上兼容的特点。即在实地址方式下,80286的目标代码与已有的8086/8088的软件兼容;在虚拟地址方式下,80286与8086/8088保持源代码相兼容,这些代码可升级使用由80286集成在芯片内的存储器管理和保护机构所支持的虚地址。图1.3为80286 CPU的内部逻辑结构图。

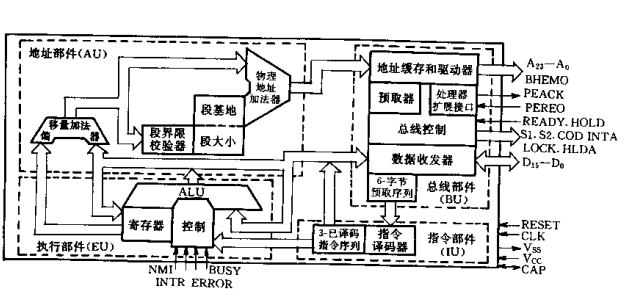

图1.3 80286 CPU的内部逻辑结构

80286 CPU 从功能上看可分成四个部分：地址部件AU(Address Unit)、总线部件BU(Bus Unit)、指令部件IU(Instruction Unit)和执行部件EU(Execution Unit)。前三个部件相当于8086/8088 的总线接口部件(BIU)。这四个部件并行操作提高了系统的吞吐率。80286 内部并行操作的情况如图1.4 所示。

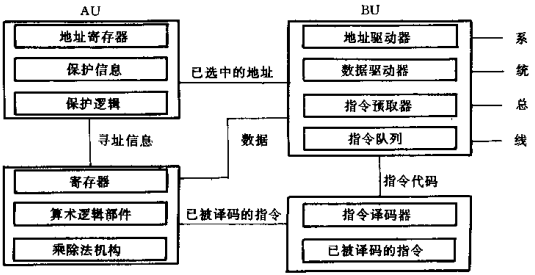

图1.4　80286 内部四个部件并行操作框图

与8086/8088CPU 中所设置的寄存器其数目、名字、功能类同。

1. 关于 4 个 16 位的段寄存器，在实地址方式下，该 4 个段寄存器的解释与使用同8086/8088，它们中包含了实际的物理地址。在虚地址方式下段寄存器中包含了虚拟存储器的地址，它用来指出虚拟空间的16 000 个64KB 段中的一个。这时，存储器的寻址操作由两个分量来指示即一个16 位的段选择子(段寄存器的内容)和一个16 位的偏移量。其中段选择子用来选择存储器中所需要的段，偏移量则表示在该段内的字节地址。虚地址到实地址之间的转换由80286 内部的存储器管理部件自动完成。

2. 增加了一个16 位的机器状态字寄存器MSW。它是一个特殊用途的处理器控制寄存器，它并不用作数据存储或操作。MSW 表示了处理器的构造和并为系统编程专用。16 位的机器状态字(MSW) 只用了最低四位(见图1.5)。

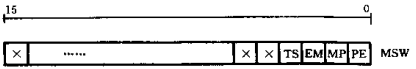

图1.5　机器状态字寄存器

MSW 中各位的含义：

PE 保护方式允许位。当PE 为1 时,使80286 处于保护工作方式,否则就处于实地址工作方式。

MP 监控协处理器。它表示是否存在协处理器。如果存在一个协处理器,则MP 为1,否则MP 为0。

EM 仿真协处理器。表示将在软件中仿真协处理器的功能。当EM=1 时,会引起所有协处理器操作码都产生一个协处理器不可用故障;若EM=0 时,则所有协处理器操作码都将在实际的80287 或80387 协处理器上执行。

TS——任务切换。每当任务切换操作时自动置位(硬件实现)。随着TS 的置位(为1),协处理器操作码将导致协处理器不可用陷阱。

* 1.2.3 80386 微处理器

80386 微处理器与80286 保持向上兼容,它包含了80286 的全部功能。在运算速度与虚存空间上均有了较大的提高与扩充。特别是80386 有了32 位的寄存器与数据通路,它从而突破了16 位微处理机CPU 的性能。它支持32 位的地址和数据类型。它和80286 一样,也有两种操作方式:实地址方式和虚地址保护方式。最大寻址能力可达4 千兆字节的物理存储空间和64 兆兆(2^{46})字节的虚拟存储空间,有内部集成的存储器管理部件与保护机构,从而更有力地支持了多任务及多用户的操作系统,成为新一代32 位的超级微处理机的CPU 之一。

8086/8088 的软件可以在80386 的实地址方式下运行。但这样使用不能充分发挥80386 CPU功能。80386 的指令流水线、32 位的总线和片内地址转换机构极大地缩短了指令的平均执行时间,形成了很高的系统吞吐量。80386 能以每秒3 百万到 4 百万条指令的速度执行指令。80386 的流水线结构如图1.6 所示。

80386 包括了8086/8088 的全部功能,并在8086 基础上增加了寄存器:

1.8 个32 位通用寄存器:EAX,EBX,ECX,EDX,ESI,EDI,EBP,ESP。

2.32 位指针EIP。

3.32 位标志寄存器EFLAGS 或EPSW。

4.2 个附加数据段寄存器:FS 和GS(仍为16 位)。

上述32 位寄存器的低16 位可以单独访问,它们相当于8086/8088 中的16 位寄存器AX,BX,CX,DX,SI,DI,BP,SP,IP 和FLAGS。其中,AX,BX,CX 和DX 仍可以分为高8 位和低8位两个字节访问,它们相当于8086/8088 中的8 位寄存器AH,AL,BH,BL,CH,CL,DH 和DL。实际上,上述32 位寄存器是由8086/8088 中相应的16 位寄存器扩展而成的。例如:EAX由AX 扩展而成;EDI 由DI 扩展而成,如图1.7 所示。

5.3 个32 位控制寄存器:CR_0,CR_2,CR_3。(CR_1 为Intel 公司保留)

其中CR_0 的0～15 位为机器状态字,与80286 相比,新增加了ET(处理器扩展类型)位,其他位的含义同80286。它的PG(位31)和PE(位0)控制页目录表和段表的操作。CR_2 中包含了造成最后一次页故障的线性地址。CR_3 中包含了目录表的物理基地址。

试读结束:需要全本请在线购买: www.ert