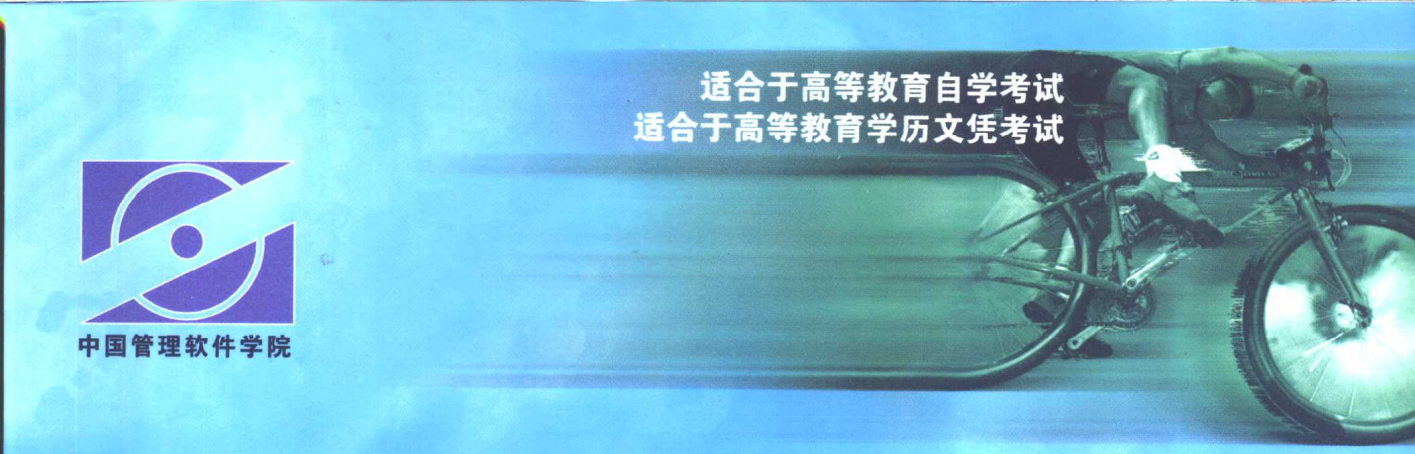
适合于高等教育自学考试
适合于高等教育学历文凭考试



中国管理软件学院

汇编语言考前冲刺

田军峰　编著

国防工业出版社

汇编语言考前冲刺

田军峰　编著

国防工业出版社
·北京·

内 容 简 介

本书介绍了“IBM-PC 汇编语言程序设计”课程的学习要点和重点，收集和编写了大量的习题，并给出了详尽的解答。主要内容有：汇编语言程序设计基础知识、IBM-PC 计算机组织、寻址方式与指令系统、汇编语言程序格式、分支与循环程序设计、子程序结构、高级汇编语言技术、输入/输出程序设计、BIOS 和 DOS 中断、单色和彩色图形显示、磁盘文件访问技术等，书中还收集了自测题和历年的高等教育学历文凭考试、高等教育自学考试试题和部分答案。

读者对象：普通高等学校在校学生，以及社会上准备参加高等教育自学考试和高等教育学历文凭考试的广大读者。

图书在版编目(CIP)数据

汇编语言考前冲刺 / 田军峰编著 .—北京：国防工业出版社，2003.1

ISBN 7-118-03063-5

Ⅰ.汇… Ⅱ.田… Ⅲ.汇编语言－程序设计－高等教育－自学参考资料 Ⅳ.TP313

中国版本图书馆 CIP 数据核字(2002)第 101066 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 11¾ 275 千字

2003 年 1 月第 1 版 2003 年 1 月北京第 1 次印刷

印数：1—6000 册 定价：21.00 元

序

中国管理软件学院成立于1984年，在北京市教委的正确领导下，在全国许多著名的计算机软件和电子信息专家的亲切关怀和指导下，面向现代化、面向世界、面向未来，培养高科技高级专业技术人才和信息管理人才，按照“强专业，高质量”的方针，在教学改革、严格管理、加强建设和探索高级人才培养模式等方面取得了出色的成绩。1993年，中国管理软件学院经国家教委审定，被批准为第一批国家学历文凭考试试点院校；1994年被批准为计算机等级培训点之一；1995年，北京市高等教育自学考试委员会为我院单独开设通信工程专业；1996年，学院获得海淀区民办高校“香港迎回归知识竞赛”一等奖；1997年，被中国成人教育协会民办高等教育委员会评为“民办高校先进单位”；1998年，被北京市教委评为民办高校“优良学校”；2001年，被教育部信息管理中心批准为远程教育培训点；2001年，被少年文摘报社评为百万读者“信得过民办院校”；2001年北京市教委对100所民办高校评估中，我院被评为首批合格24所院校之一；2002年北京市教委和高等教育自学考试委员会唯一批准我院开设计算机网络专业（文凭考试），北京市教委和北京市考试院又在我院设立了北京市英语口语等级证书培训点、考试点。这些成绩的取得是与教委的领导和著名专家的指导分不开的，也是我们全体师生努力奋斗、顽强拼搏、极力创新、创建特色的结果。

我院以计算机控制及应用专业而闻名，以通信工程为名牌，以计算机网络专业为特色，以计算机软件及应用为优势。这四个专业代表了学院的特点，使全国的莘莘学子不远千里，慕名前来求学。随着高科技日新月异的发展，我院不断调整专业的深度和广度，并加强各专业的外语学习，使这些专业长盛不衰，符合社会发展的需要，为国家培养了许多急需的新型的有专业知识的技能型高科技人才，为国家的现代化建设作出了应有的贡献。

实践告诉我们，教育质量是民办高校的生命线，而好的教材是提高教学质量的一个重要方面。通过加强基本理论和基本技能的训练，使学生基础理论扎实，动手能力强，真正成为过硬的高科技应用型人才。

我院一向注重教材建设，编写了一套适合国家高等教育文凭考试和高等教育自学考试的系列教材，受到了许多读者的欢迎和赞扬，也得到了许多同仁的支持和帮助，在此表示深深的感谢。为了更好地为学生服务，我们把一些教材系列出版。希望使用这套书的读者提出更多的宝贵意见，以便我们今后能编写出更好更适用的教材，为我国的民办教育作出更大的贡献。

中国管理软件学院院长

朱忠才

2002年10月

前　言

为了帮助高等教育学历文凭考试和高等教育自学考试“计算机应用”专业(专科)的应试者学习《IBM-PC 汇编语言程序设计》教材的需要,我们编写了这本汇编语言辅导书。编写本书的宗旨是:

1.按照高等教育学历文凭考试大纲的要求,我们从学习方法上、从各知识点如何使用的角度上,对各章的考核知识点进行阐述、分析、比较、归纳,以帮助读者理解和掌握教材内容。

2.考虑到本课程的特点,我们通过大量的例题、习题及题解的分析,不但帮助读者提高解题能力和程序设计能力,更重要的是使读者加深对概念、原理和方法的理解和掌握。

本书包括如下四大部分内容。

第一部分　学习要点。

首先根据高等教育学历文凭考试“‘8086 汇编语言’课程考试大纲”的要求,对历年考试真题和每章考核知识点进行分析,并加以提炼和总结,使读者一目了然每章的主要内容是什么。然后对每章的考核知识点,特别是重点、难点进行分析、比较和归纳,指出知识的关键所在,指出各知识点之间的异同之处,指出使用知识点时的注意事项。

第二部分　自测题及参考答案。

提供了五套汇编语言程序设计自测题。

第三部分　历年高等教育学历文凭考试试题汇编及参考答案。

收集了从 1996 年到 2001 年间每一年的汇编语言学历文凭考试试卷。

第四部分　历年高等教育自学考试试题汇编及部分参考答案。

收集了从 1996 年到 2001 年间每一年北京市和全国汇编语言自学考试试卷。

参加编写本书习题和试题解答的还有:隽青龙、张宝武,审校的有陆广、刘庆涛、吴东、杨生喜、王庆、陈嘉珺、王书玲。此外,不少同志对本书提出了许多宝贵意见,在此,我们一并表示衷心的感谢。

由于时间仓促和限于编者的水平,书中错误和不妥之处,敬请读者批评指正。

编　者

2002 年 12 月于北京

目　录

第一部分　学习要点

第二部分　自测题及参考答案

第三部分　历年高等教育学历文凭考试试题汇编及参考答案

第四部分　历年高等教育自学考试试题汇编及部分参考答案

第一部分　学习要点

第一章　基础知识

本章小结

本章简略介绍了二进制数、十进制数及十六进制数的特点及各种数制之间的转换方法。正确地理解数制之间的关系和转换方法，是学习汇编语言编程的基础。

ASCII 码和 BCD 码是两种常见的数值编码。ASCII 码是 IBM-PC 输入/输出时采用的编码。输入、计算、再输出的过程往往是要进行多次的“码→值”、“值→码”以及数制之间的转换。正确地理解这一过程，对于编写输入、输出用的汇编语言程序很重要。

1.1　数制及数制间的转换

1.1.1　二进制数、十进制数及十六进制数

二进制数	0000	0001	0010	0011	0100	0101	0110	0111
十进制数	0	1	2	3	4	5	6	7
十六进制数	0	1	2	3	4	5	6	7

二进制数	1000	1001	1010	1011	1100	1101	1110	1111
十进制数	8	9	10	11	12	13	14	15
十六进制数	8	9	A	B	C	D	E	F

1.1.2　二进制数与十进制数之间的转换

1. 二进制数转换成十进制数

例：11111100B = $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2$ = 252D

2. 十进制数转换成二进制数

例：将十进制数 350D 转换为二进制数

350/2 = 175　　余 0
175/2 = 87　　余 1
87/2 = 43　　余 1
43/2 = 21　　余 1
21/2 = 10　　余 1
10/2 = 5　　余 0
5/2 = 2　　余 1
2/2 = 1　　余 0
1/2 = 0　　余 1

将余数从后往前排列，得到 101011110B，这就是转换的结果。

1.1.3 二进制数、十进制数及十六进制数之间的转换

1. 二进制数转换成十六进制数

例:1000001B=0100 0001B=41H

2. 十六进制数转换成二进制数

例:56EH=0101 0110 1110B

3. 十六进制数转换成十进制数

例:$0FFFFH = 15\times16^3 + 15\times16^2 + 15\times16^1 + 15\times16^0$

$= 15\times4096 + 15\times256 + 15\times16 + 15\times1$

$= 65535D$

4. 十进制数转换成十六进制数

将十进制数转换成十六进制数的方法主要有除法和降幂法。

例:65534D ⟶ 65534/16=4095 余 14

4095/16=225 余 15

255/16=15 余 15

15/16=0 余 15

将余数从后往前排列,得到 0FFFEH,这就是转换的结果。

1.2 二进制数和十六进制数的算术运算

1.2.1 二进制数和十六进制数的算术运算

1. 二进制数的减法运算,实质上是被减数与减数的补码相加。

2. 二进制负数补码的基本规则是:“各位取反,末位再加 1”。

1.2.2 二进制数和十六进制数的逻辑运算

常见的逻辑运算有:“与”、“或”、“非”、“异或”四种。

1.“与(AND)”运算

$0\wedge0=0$　　$0\wedge1=0$

$1\wedge0=0$　　$1\wedge1=1$

2.“或(OR)”运算

$0\vee0=0$　　$0\vee1=1$

$1\vee0=1$　　$1\vee1=1$

3.“异或(XOR)”运算

$0\forall1=1$　　$1\forall0=1$

$0\forall0=0$　　$1\forall1=0$

4.“非(NOT)”运算

$\overline{0}=1$　　$\overline{1}=0$

1.3 ASCII 码和 BCD 码

在实践中,用一个字节来表示 BCD 码时,有两种方式:压缩 BCD 码和非压缩 BCD

码。标准的ASCII码共有128个字符,分为可打印的ASCII码字符和非打印的ASCII码字符两大类,其中可打印的ASCII码字符共有95个,主要有:

26个大写字母A~Z	41H~5AH
26个小写字母a~z	61H~7AH
10个数字0~9	30H~39H
专用字符:*,$,空格等	2AH,24H,20H

非打印的ASCII码字符用于控制代码,共有33个。

1.4 数值与编码的转换

考核知识点

例题1:计算机中存储信息的最小单位是_____。

【解答】"位"或"比特"

例题2:计算机存储器的基本单位是_____。

【解答】字节

例题3:ASCII码的中文意思是_____。

【解答】美国信息交换标准码

例题4:大写字母A对应的十六进制数是_____,小写字母a对应的十六进制数是_____,数字0对应的十六进制数是_____。

【解答】41H,61H,30H

第二章 IBM-PC计算机组织

本章小结

本章主要介绍了IBM-PC计算机组织,内容有:CPU、内存及堆栈。对编程者来说,主要是与CPU的寄存器打交道。CPU的寄存器组中有8个通用寄存器AX,BX,CX,DX,SP,BP,SI,DI;下面简单介绍它们在实际编程中的作用。

2.1 IBM-PC微型计算机的基本组成

(1) 一台电子计算机通常由三大部分组成:中央处理器、存储器、输入/输出设备。

(2) CPU包括运算器和控制器两部分。运算器负责执行所有的算术和逻辑运算。控制器则负责指令执行时的控制工作。

(3) PC机的CPU最早是由Intel8088和Intel8086组成。8088的外部数据总线是8位的,8086的外部数据总线是16位的。

(4) Intel8088的CPU内部共有14个16位寄存器,共有20根地址线,可直接寻址的空间为1MB。用于访问外部设备的I/O端口地址空间为64KB。

2.2 IBM-PC 上的软件与汇编语言

根据软件的工作性质,其软件可分为两大类:系统软件和应用软件。

2.2.1 系统软件和应用软件

公认的系统软件有:基本输入/输出系统(BIOS)、操作系统、系统工具或其他专用系统、软件开发环境。

2.2.2 高级语言、汇编语言、机器语言

高级语言的翻译程序分为两类:一类是编译程序或编译器;另一类是解释程序。

2.2.3 汇编语言的开发环境

典型的汇编语言开发环境都包含四种工具:编辑器、汇编器、连接程序及调试程序。

(1) 编辑器用于输入汇编语言程序的源程序。

(2) 汇编器的作用是将汇编源程序翻译成"可重新定位的"目标文件(以.OBJ 为扩展名)。

(3) 连接程序的作用是将一个或多个目标文件(.OBJ 文件)连接成一个可执行文件。

(4) 调试程序的作用是以一步步的跟踪执行方式来找出编译好的程序在运行时的状况,从而发现可能的逻辑错误。

2.3 Intel 8086/8088 CPU 的寄存器结构

2.3.1 通用寄存器

Intel 8086/8088 的通用寄存器共有 8 个,它们分别是:

(1) AX 表示累加器(ACCUMULATOR),"A"便取自此单词的首字母。算术运算时,主要用 AX 来存放操作数和运算结果,这种使用有时是强制的,例如乘、除法指令。在输入输出指令中也要用它来存放数据。

(2) BX 一般用作通用寄存器,但在间接寻址时,也用它来存放基地址。

(3) CX 常在循环指令及串循环指令中用做计数器;移位指令则使用 CL 为计数器,每次循环时,它会自动地减 1 或减 2,在没有上述需要时,它也可以用作通用寄存器。

(4) DX 一般当作通用寄存器,但在乘法、除法进行 32 位的双字运算时,它是 AX 的扩展(因为 AX 只有 16 位)。在输入/输出指令中,则用它来存放 I/O 端口的地址,从而实现间接寻址。

(5) SP 堆栈指针(STACK POINTER)的作用是指示堆栈栈顶的偏移地址。

(6) BP 基址指针(BASE POINTER)是一种基地址寄存器,所不同的是它相对于堆栈段而不是数据段。

(7) SI 源变址寄存器(SOURCE INDEX)通常用作地址指针。

(8) DI 目的变址寄存器(DESTINATION INDEX)通常用作地址指针。

2.3.2 段寄存器

有 4 个段寄存器 CS,DS,SS,ES;下面简单介绍其作用。

(1) CS 代码段寄存器(CODE SEGMENT)存放的是代码段的段地址。程序执行时必须有代码段。

(2) DS 数据段寄存器(DATA SEGMENT)存入的是数据段的段地址。一般程序执行时都有一个或多个数据段。使用不同的数据段时,必须相应地修改 DS。

(3) ES 附加段寄存器(EXTRA SEGMENT)存放的是附加段的段地址。当一个程序同时用到一个以上的数据段时,附加段就是必不可少的。

(4) SS 堆栈段寄存器(STACK SEGMENT)存放的是堆栈段的段地址。设置堆栈段是为了给入栈操作和出栈操作保留临时数据区。

2.3.3 指令指针和标志寄存器

Intel 8086/8088 有一个指令指针 IP,一个程序状态字 PSW。每个寄存器都有它独特的用途。

2.4 PC 机的内存组织

2.4.1 内存地址与字节、字的存放

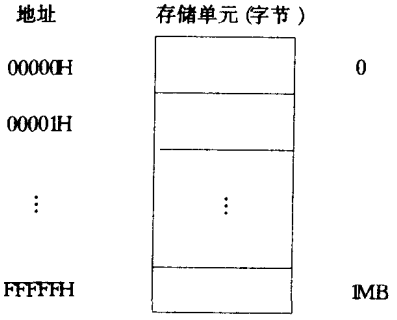


图 2.4.1 PC 机内存地址与内存单元的对应关系

2.4.2 物理地址和逻辑地址

基于 Intel8086/8088 的 PC 机,其内存空间为 1MB,要采用分段的方式来寻址。逻辑地址由段地址和偏移地址组成,它们都是 16 位的,形成物理地址时,由段地址左移 4 位与偏移地址相加。往内存中存入一个字时,低字节在前,高字节在后。

2.5 堆　　栈

什么是堆栈呢? 其实堆栈是内存中的一块特殊区域,它由 SS,SP 寄存器确定。进栈时 SP-2,然后将要压入的字放入(SP)处;出栈时,先将(SP)处的字送入目的操作数处,然后 SP+2。堆栈的操作遵守"先进后出"的规则。

2.5.1 堆栈

堆栈的一端是固定的,另一端是浮动的。信息存取在浮动的一端。

2.5.2 堆栈操作

堆栈的操作遵守"先进后出"的规则。

堆栈的操作主要有两种:压入和弹出。

1. 执行 PUSH AX 指令的过程

(1) SP←SP-2　　;首先将堆栈指针减 2。

(2) (SP)←AX　　;将 AX 中的一个字送入堆栈段内由 SP 指向的偏移地址处。

2. 执行 POP AX 指令的过程

(1) AX←(SP)

(2) SP←SP+2

学习汇编语言必须有一套开发环境,它一般由汇编器、连接程序及调试程序组成。

考核知识点

1. 选择题

例题 1:SI 寄存器是属于(　　)。

A. 通用寄存器　　B. 变址寄存器　　C. 段寄存器　　D. 控制寄存器

【解答】A.B

例题 2:CX 寄存器是属于(　　)。

A. 通用寄存器　　B. 变址寄存器　　C. 段寄存器　　D. 控制寄存器

【解答】A

2. 填空题

例题 3:一台电子计算机通常由三大部分组成,它们分别是:_____,_____,_____。其中_____包括运算器和控制器。

【解答】中央处理器,存储器,输入/输出设备,中央处理器

例题 4:Intel 8088CPU 内部共有_____个 16 位寄存器,共有_____根地址线,可直接寻址的空间为_____,用于访问外部设备的 I/O 端口地址空间为_____。

【解答】14,20,1MB,64KB

例题 5:典型的汇编语言开发环境都包含四种工具,分别是:_____,_____,_____,_____。

【解答】编辑器,汇编器或汇编程序,连接程序,调试程序

例题 6:我们把与内存单元一一对应的用 20 位二进制数(或 4 位十六进制数)表示的地址称为_____。

【解答】物理地址

例题 7:堆栈的一端是_____的,另一端是_____的。信息存取在_____的一端进行。

【解答】固定,浮动,浮动

例题 8:存放各种运算操作的数据寄存器是_____。

【解答】通用寄存器

例题 9:查看堆栈中当前正在进行出、入栈的存储单元的地址寄存器是_____。

【解答】SP 寄存器

例题 10:查看运算结果是否等于零的寄存器是_____。

【解答】PSW 寄存器

例题 11:汇编语言中源程序文件的扩展名是_____。

【解答】.ASM

3. 问答题

例题 12:为什么要学习汇编语言?

【解答】之所以学习汇编语言是因为:

(1) 通过汇编语言可以理解计算机运行软件的过程。

(2) 某些低层系统软件必须用汇编语言来编写。例如主板上的 BIOS 等。

(3) 汇编语言程序的执行效率高,运行代码占用内存少。

第三章　寻址方式与指令系统

本章小结

本章介绍了 Intel8086/8088 的指令系统及寻址方式。

指令的基本形式为:OP dst,src

根据需要,也可以只有一个操作数 src 或没有操作数,特别是,许多指令有隐藏的操作数。

获取指令操作数的方式称为寻址方式。根据用途可分为数据寻址方式、转移指令寻址方式、IN/OUT 指令寻址方式三大类。本章介绍了六种数据寻址方式及四种转移指令寻址方式。

对于数据寻址方式,特别是内存操作数寻址方式,要注意寻址的数据段,形成偏移地址的过程。

对转移指令寻址方式,要结合 JMP 及 CALL 指令,弄清楚转移地址的形成。

Intel8086/8088 的指令系统共分为六大类,本章介绍了其中的大多数常用指令。每条指令的寻址方式、缺省时使用的寄存器及用途,是初学汇编语言者感到难于记忆的内容,需要通过例子和习题,特别是在以后的编程中灵活运用去掌握。

3.1　指 令 格 式

3.1.1　指令的汇编语言格式

Intel8086/8088 指令一般都由操作码和操作数两部分组成。

Intel8086/8088 指令的格式如下:

操作码　　目的操作数　　源操作数

或表示成

OP　　DST,SRC

这里的操作数是该指令的操作对象。

3.1.2　指令的机器语言格式

1. 操作码字节

操作码的基本格式有如下两种:

(1) op　　d　　w

(2) op　　s　　w

2. 寻址方式字节

mod　　reg　　r/m

其中,reg 是指对应的操作数是哪个寄存器,它需要与指令操作码的 w 位结合,mod(寻址方式)字段要与 r/m(寄存器/内存)字段结合确定另一个操作数的寻址方式。

3. 段超越字节

当使用寄存器 BP 寻址时,隐含的段寄存器是 SS,使用其他寄存器(BX,SI,DI)寻址

时隐含的寄存器是DS。

3.2 寻址方式

3.2.1 与数据有关的寻址方式

1. 立即寻址

立即寻址是指指令所需的操作数直接包含在指令代码中,它通常是一个常量或常数,我们称它为立即数。例如:

```
MOV    AL, 05H
MOV    AX, 502
```

2. 寄存器寻址

寄存器寻址是指指令中的操作数是CPU的某个寄存器,例如:

```
MOV    AX,BX
MOV    AX,3412H
ADD    X, AX
PUSH   DS
```

3. 直接寻址

直接寻址是指操作数的偏移地址直接在指令中指出的寻址方式。例如:

```
MOW    AX,[3000H]
```

4. 寄存器间接寻址

寄存器间接寻址方式是指操作数的有效地址EA不是位于指令中,而是位于基址寄存器BX,BP或变址寄存器SI,DI中。例如:

```
MOV   AX, [BX]
MOV   BH, [BP]
```

其中BX,SI,DI,是相对于DS段的偏移地址,而BP中是相对于SS段的偏移地址。例如:

```
MOV    AX,DS: [BX]
MOV    BH,SS: [BP]
```

5. 寄存器相对寻址

寄存器相对寻址是指操作数的有效地址EA是一个基址寄存器或变址寄存器的内容和指令中指定的8位和16位位移量之和。即

EA=间址寄存器的值+8位或16位常量

例如:

```
MOV    AX, [SI+10H]
MOV    AX, 10H [SI]
```

注意上面第2条指令的写法,不是10H乘以SI的内容形成偏移地址,[SI+10H]与10H[SI]的结果是一样的。

6. 基址变址寻址方式

基址变址寻址方式指的是:操作数的有效地址EA等于一个基址寄存器和一个变址寄存器的内容之和。其中基址寄存器为BX或BP,变址寄存器为SI或DI。例如:

```
MOV    AX, [BX][SI]
```

```
MOV   AX, [BX+SI]
MOV   AX, [BX+SI+200]
MOV   ARRAY [BP+SI],AX
```

注意,基址变址寻址方式的基址寄存器只能是 BX 或 BP ,而变址寄存器也只能是 SI 或 DI。

3.2.2 与转移地址有关的寻址方式

与转移地址有关的寻址方式主要运用于转移指令(JMP)和过程调用指令(CALL)。根据其寻址特点,我们将它总结为以下四种寻址方式:

(1) 段内直接寻址。

(2) 段内间接寻址。

(3) 段间直接寻址。

(4) 段间间接寻址。

1. 标号与过程名

在代码段中,程序会随着 IP 值的增加顺序执行,如想改变执行的顺序,就得使用转移指令(无条件转移指令和有条件转移指令)及 CALL 等指令。JMP 或 CALL 指令要转移到的地方,必须是位于同一个代码段内的某个偏移地址,或另一个代码段中的某个偏移地址处。前一种转移称段内转移,后一种称段间转移。

如何确定同一代码段内或另一代码段的偏移地址——通过标号或过程名来定义。标号的定义类似于下面的形式:

```
⋮
L1:MOV   AX, ARRAY[SI]
⋮
```

其中 L1 是标号。

过程名的定义形式如下:

```
⋮
P1   PROC   near  (或 far)
⋮
     RET
P1   ENDP
```

2. 段内直接寻址

段内直接寻址是指要转向(由 JMP,条件转移,CALL 等)指令实现的有效地址是当前 IP 寄存器的内容和指令中指定的 8 位或 16 位位移量之和。

下面的指令是段内直接寻址方式

```
JMP    l1
CALL   p1
```

3. 段内间接寻址

段内间接寻址是指转向的有效地址是一个寄存器或一个存储单元的内容。寄存器可以是 AX,BX,CX,DX,SI,DI,BP,SP 中的任何一种。存储单元位于数据段中,它可以是任何一种寻址方式,如直接寻址、寄存器间接寻址、寄存器相对寻址等。例如:

```
MOV   AX,OFFSET p1
```

CALL AX

4. 段间直接寻址

与段内直接寻址不同的是,段间间接寻址在指令中给出了要转移至(由 JMP 和 CALL 完成的)地址的代码段和偏移值内容。如:

CALL FAR PTR p2

要转移至的标号或过程名必须具备 FAR 属性。

5. 段间间接寻址

段间间接寻址方式类似于段内间接寻址方式,只是在间接寻址时不能将要转移至的地址直接放入寄存器,而必须放入内存单元中,且是一个双字,例如:

JMP DWORD PTR [BX + INTERS]

3.3 指令系统

3.3.1 数据传送指令

1. 数据通路与类型匹配(重点)

数据传送指令中各源操作数和目的操作数之间的匹配遵照图 3.3.1 所示。

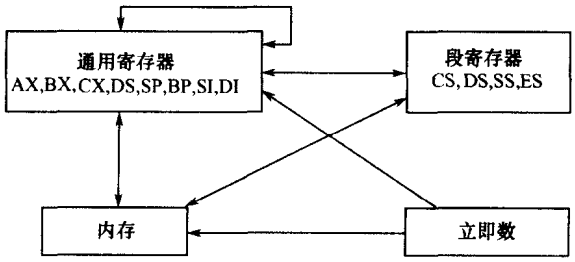


图 3.3.1 指令的数据通路

2. MOV 和 XCHG 指令

MOV 指令或其他大多数指令必须遵守的规则:

(1) 源和目的必须类型匹配(8 位对 8 位,16 位对 16 位)。

(2) 目的操作数不能为立即数。

(3) 源和目的操作数不能同时为内存操作数(串指令除外)。

(4) 源和目的操作数不能同时为段寄存器。

(5) 当使用段寄存器时目的操作数不允许使用 CS。

3. PUSH,POP,PUSHF,POPF 指令

(1) 作用: PUSH SRC ;SP = SP - 2,SS:[SP]←SRC

PUSHF ;SP = SP - 2,SS:[SP]←PSW

POP DST ;DST←SS:[SP],SP = SP + 2

POPF ;PSW←SS:[SP],SP = SP + 2

(2) 堆栈指令的规定如下:

①堆栈的存取只能以字为单元,不允许以字节为操作数。

②PUSH 指令可以使用 CS 寄存器,但 POP 指令不可以使用 CS 寄存器。除此以外,