计算机及应用专业 专科

全国高等教育自学考试

汇编语言程序设计习题详解

黄明 梁旭 常征 编著

机械工业出版社
CHINA MACHINE PRESS

全国高等教育自学考试

汇编语言程序设计习题详解

（计算机及应用专业 专科）

黄 明 梁 旭 常 征 编著

本书是根据“全国自学考试（计算机及应用专业 专科）考试大纲”以及历年考题编写的。本书共分 4 部分：第 1 部分是笔试应试指南；第 2 部分是笔试题解；第 3 部分是模拟试卷及参考答案；最后是附录，包括考试大纲和 2002 年下半年试卷及参考答案。

本书紧扣考试大纲，内容取舍得当，叙述通俗易懂，附有大量与考试题型类似的习题及答案，以检查读者对考点的掌握程度。

本书适用于准备参加全国自学考试（计算机及应用专业 专科）的考生，也可作为大专院校和培训班的教学参考书。

图书在版编目（CIP）数据

汇编语言程序设计习题详解/黄明等编著. —北京：机械工业出版社，2004.6
（全国高等教育自学考试）
ISBN 7-111-14424-4

Ⅰ. 汇… Ⅱ. 黄… Ⅲ. 汇编语言—程序设计—高等教育—自学考试—解题
Ⅳ. TP313-44

中国版本图书馆 CIP 数据核字（2004）第 041628 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策　　划：胡毓坚
责任编辑：孙　业
责任印制：洪汉军
三河市宏达印刷有限公司印刷·新华书店北京发行所发行
2004 年 6 月第 1 版·第 1 次印刷
787mm × 1092mm 1/16 · 13 印张 · 317 千字
0 001—5 000 册
定价：20.00 元

出 版 说 明

全国高等教育自学考试指导委员会推出面向社会的高等自学考试，经过 10 多年的实践，已建立起一整套较为完善的规章制度和操作程序，考试组织严密规范，考试纪律严格；坚持考试标准，实行教考分离，确保了毕业生的质量。它为没有机会进入高等学校的中国公民提供了接受高等教育的机会，并以严格的国家考试保证了毕业生的质量，获得了普遍赞誉。国家自考中心于 2002 年开始执行新的考试计划。新计划中开设的专业共 224 个，其中专科 141 个占 63%，独立本科段 61 个占 27%，专本衔接专业 22 个占 10%。为帮助、指导广大自学考生深入理解计算机及相关专业考试的基本概念，灵活运用基本知识，掌握解题方法和技巧，熟悉考试模式，进一步提高应试能力和计算机水平，特编写了以下专业的基础课与专业课主要课程的习题详解。

- ✧ 计算机及应用专业 独立本科段
- ✧ 计算机信息管理专业 独立本科段
- ✧ 计算机网络专业 独立本科段
- ✧ 计算机及应用专业 专科

丛书特点：

1．以 2002 年最新考试大纲为基准

本丛书是根据 2002 年最新考试大纲，为参加全国高等教育自学考试考生编写的一套习题详解教材。

2．例题反映了历届考试中的难度和水平

书中对大量的例题进行了分析，所选例题都是在对最近几年考题深入研究的基础上精心筛选的，从深度和广度上反映了历届考试中的难度和水平。

3．作者经验丰富

本丛书的作者都是多年从事全国高等教育自学考试辅导的高等院校的教师。

读者对象：

- ✧ 准备参加全国高等教育自学考试的考生。
- ✧ 计算机及相关专业的本专科生。

前言

自学考试是对自学者进行以学历考试为主的国家高等教育学历考试。本书是为帮助和指导广大考生深入理解考点涉及的基本概念，灵活运用基本知识，掌握解题方法和技巧，熟悉考试模式，进一步提高应试能力和计算机水平而编写的。

全书共分 4 部分，即笔试应试指南、笔试题解、模拟试卷及参考答案和附录。书中所选试题均是在对历年真题深入研究的基础上经过精心筛选的，从深度和广度上反映了考试的难度和水平。模拟试卷的题型分配与真题一致，这些题目是考试指导教师的多年积累，且在辅导班中多次使用过。

书中附录给出了“全国自学考试（计算机及应用专业 专科）汇编语言程序设计考试大纲”，以及“2002 年下半年全国自学考试（计算机及应用专业 专科）汇编语言程序设计笔试试卷及参考答案”。

本书由大连铁道学院黄明、梁旭、常征编写。

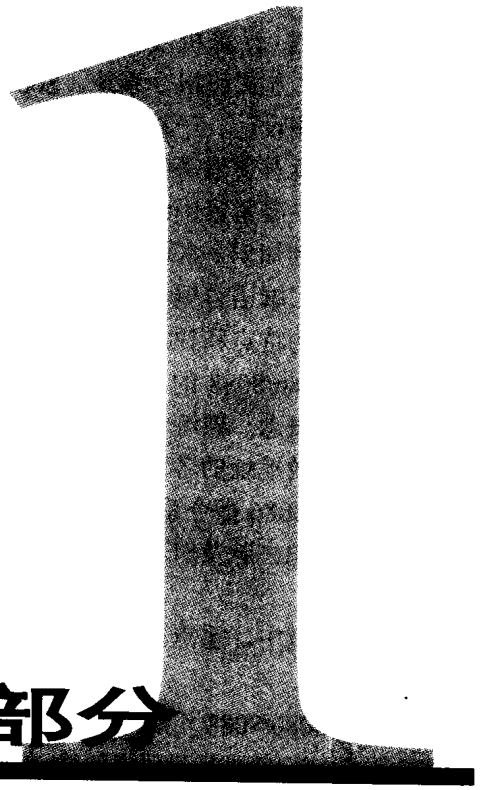
由于编者水平有限，书中错误和不妥之处在所难免，请读者和专家批评指正。

读者在使用本书的过程中如有问题，可通过 E-mail 与我们联系：dlhm@263.net

编 者

目　录

第3部分 模拟试卷及参考答案



第1部分

笔试应试指南

笔试应试策略

笔试考点归纳

1.1 笔试应试策略

全国自学考试（计算机及应用专业 专科）汇编语言程序设计考试大纲涵盖了基础知识、8086/8088 的寻址方式和指令系统、8086/8088 汇编语言程序格式、顺序程序设计、分支程序设计、循环程序设计、子程序设计等 7 章内容。使用的教材是由全国高等教育自学考试指导委员会组编，姚君遗编著的《汇编语言程序设计》，2000 年 3 月由经济科学出版社出版。考试复习的过程中要紧紧围绕大纲的知识点，首先对大纲涉及的各章基本概念应熟练掌握。

第 1 章为基础知识，本章总的要求是：熟练掌握有关计算机系统的基本概念、基本组成。熟练掌握 8086 汇编语言编程的基本硬件模型。理解学习汇编语言程序设计的目的，了解汇编语言程序设计的特点和作用。重点是 8086 汇编语言编程的硬件模型。

第 2 章为 8086/8088 的寻址方式和指令系统，本章总的要求是：深刻理解寻址方式、指令系统和指令的概念。熟练掌握 8086 的各种寻址方式的含义和书写格式。着重在理解存储器操作数形成有效地址的各种方法和规定，段地址的约定和段超越的规定及书写格式。深刻理解和熟练掌握 8086 指令系统中各种常用指令的操作内容和参加操作的源操作数和结果（目的）操作数的来龙去脉及对程序状态字寄存器的状态位的影响。会使用各种常用指令分析和编写程序。

本章的知识点中，重点是数据传送指令、算术和逻辑运算指令。难点是控制转移指令和串处理指令。

第 3 章为 8086/8088 汇编语言程序格式，本章总的要求是：了解和掌握汇编语言语句种类及格式要求、汇编语言源程序的格式要求。熟悉汇编语言程序上机过程。理解和掌握各类伪指令的助记符、操作数的规定、书写格式和用处。了解宏指令的概念、书写规定和用处。

本章的知识点中，重点是符号定义语句和数据定义语句。难点是汇编语言语句种类和格式，汇编语言源程序格式。

第 4 章为顺序程序设计，本章总的要求是：深刻理解程序的基本概念，充分认识程序设计的基本步骤的重要性和必要性，理解和掌握程序设计的基本步骤和基本方法。理解和掌握顺序程序的结构形式和程序设计方法。

本章知识点中，重点是顺序程序的基本结构和顺序程序设计。难点是算法和程序流程图。

第 5 章为分支程序设计，本章总的要求是：熟知各种形式的分支程序的结构特点。理解和掌握双分支程序中产生条件和判断条件的程序段的设计方法和技巧。理解和掌握三种多分支程序设计原理、设计方法和技巧。

本章的知识点中，重点是双分支程序设计。难点是三种多分支程序设计。

第 6 章为循环程序设计，本章总的要求是：熟悉循环程序的基本结构形式及各组成部分的内容和功能。熟练掌握单重循环和多重循环程序设计的方法和技巧。熟练掌握控制循环的四种设计方法和技巧。

本章知识点中，重点是单重循环程序设计和控制循环的方法。难点是多重循环程序设计。

第 7 章为子程序设计，本章总的要求是：深刻理解子程序的概念、基本结构形式和在程序设计中的作用。熟悉子程序文件说明的重要性及书写子程序文件说明的规定。弄清主程序和子程序的调用关系，熟记子程序调用和返回过程所完成的操作。

能熟练掌握子程序设计方法和技巧。初步掌握子程序参数传递的方法和子程序嵌套方法。了解 DOS 系统调用的基本内容及调用方法。

本章知识点中，重点是子程序设计和子程序参数传递。难点是子程序嵌套。

在复习时应根据大纲提供的考核点和考核要求来进行复习，这样就能抓住重点，进行有效复习。在做练习时，要根据考试的题型进行练习，在掌握基本概念的基础上，掌握一定的解题技巧。汇编语言程序设计的考试题型有：单选题、填空题、简答题、程序分析题、程序填空题和程序设计题。对于不同题型，要采用不同的答题方法。

单选题：这种题型可考查考生的理解、推理分析，综合比较，评分客观。在答题时，如果可以，直接得出正确答案，对于没有太大把握的试题，也可以采用排除法，经过分析比较加以逐步排除错误答案，最终选定正确答案。

填空题：这种题型常用于考核考生观察能力与运用有关概念、原理的能力。在答题时，无论有几个空，回答都应明确、肯定，考生在复习中最好的应对办法是对学科知识中最基本的知识、概念、原理等要牢记。

简答题：这种题型着重考核考生对基本知识点理解的全面性和概括性，在复习的过程中对考试大纲涉及的一些基本概念和原理要熟练掌握。

程序分析题：程序分析题目在自考试卷中占有较大的比例，主要考查阅读程序、分析程序的能力以及对指令的掌握。这种题型灵活性比较大，着重考核考生对概念、知识、原理的掌握和分析问题的能力。

程序填空题：程序填空题要求考生必须具备熟练的程序分析和设计能力。而且考查解题的准确性。

程序设计题：这种题型着重考核考生分析、解决实际问题的能力，考核考生综合应用能力和创见性。在答题时，要综合运用所学知识进行分析和设计。

考生在复习时在掌握知识点的同时也应抓住这些题型的特点，这样才能达到好的应试效果。

1.2 笔试考点归纳

1.2.1 基础知识

1. 计算机系统基本组成

计算机系统包括硬件和软件两部分。硬件包括各种功能部件电路、外围设备和机柜等硬设备。软件则是为了运行、管理和维护计算机而编制的各种程序的总和。

（1）硬件

包括中央处理器 CPU（Central Processing Unit）、存储器（Memory）和输入/输出（Input/Output）子系统三个主要组成部分，它们三者由系统总线连接在一起。

中央处理器包括运算器和控制器、内部可编辑寄存器组。

存储器是计算机的记忆部件，人们编写的程序（由指令序列组成）就存放在里面。

系统总线把 CPU、存储器和 I/O 设备连接起来，用来传送各部分之间的信息。系统总线包括数据总线、地址总线和控制总线，简称三总线。

（2）软件

计算机软件是计算机系统的重要组成部分，它可以分为系统软件和用户软件两大类。系统软件是由计算机厂商提供给用户的一组程序，这些程序是用户使用机器时为产生、准备和执行用户程序所必需的。用户软件则是用户自己编制的各种程序。

系统软件的核心称为操作系统（Operating System）。操作系统是系统程序的集合，它的主要作用是对系统的软、硬件资源进行合理的管理，为用户创造方便、有效和可靠的计算机工作环境。

操作系统的主要部分是常驻监督程序（Monitor），只要一开机它就开始运行，它可以接受用户命令，并使操作系统执行相应的动作。

2．8086 汇编语言编程的硬件模型

Intel 8086/8088 是两种第三代微处理器。在汇编语言一级，它们与 8080/8095 微处理器是兼容的。它具有 20 条地址总线，直接寻址内部存储器能达到 1MB（1 兆个字节存储单元）。

8088 具有 8 位数据总线可与内存或输入 / 输出设备交换数据，而 8086 则有 16 位数据总线。其他方面两个微处理器都是相同的，为其中一个 CPU 编写的软件，可以不加修改地在另一个 CPU 上执行。

CPU 由三部分组成：

① 算术逻辑部件 ALU（Arithmetic Logic Unit），用来进行算术和逻辑运算。

② 控制逻辑，负责对全机的控制工作，包括从存储器取出指令，对指令进行译码分析，从存储器取得操作数，发出执行该指令的所有命令，把结果存入存储器，以及对总线及 I/O 传送控制等。

③ 工作寄存器，在计算机中起着重要的作用，每一个寄存器相当于存储器中的一个存储单元，但它的存取速度比存储器（内存）要快得多。它用于存放计算过程中所需要的或所得到的信息，包括操作数地址，操作数据（原始数据、中间结果和结果数据）等。这些工作寄存器有时我们也习惯称它们为可编程寄存器，原因是这些工作寄存器绝大多数可用 8086/8088 指令在编程时对它们写数或读数。

（1）8086 微处理器内部数据寄存器组和段寄存器组的各寄存器名称、符号、位数和功能，指令寄存器 IP 的位数和功能

1）数据寄存器。数据寄存器包括 AX、BX、CX 和 DX 四个通用寄存器，它们用来暂时存放运算过程中所用到的操作数、结果数据或其他信息。它们既可以以字（16 位）的形式使用，也可以以字节（8 位）的形式使用。

AX（Accumulator）作为累加器用，它是乘法运算中存放参加运算的一个操作数及存放运算结果数据——乘积或乘积的低 16 位部分。另外，所有的输入 / 输出指令都使用这一寄存器与外部设备传送信息。

BX（Base）可以用作通用寄存器。此外，在计算存储器地址时，它经常用作基地址寄存器，所以又称基址寄存器。

CX（Count）可以用作通用寄存器。此外，在循环（Loop）和串处理指令中用作隐含的计数器。

DX（Data）可以用作通用寄存器，在做双字长运算时，把 DX 和 AX 组合在一起存放一个双字长数。DX 存放高位字（高 16 位）。此外，对某些输入 / 输出操作，DX 用来存放 I/O

端口地址。

2）指针及变址寄存器。指针及变址寄存器包括 SP、BF、SI、DI 四个 16 位寄存器。它们可以像数据寄存器一样在运算过程中存放操作数，但它们只能以字（16 位）为单位使用。此外，它们更经常用于在段内寻址时提供偏移地址。

SP（Stack Pointer）堆栈指针寄存器：用来指示堆栈的栈顶的偏移地址，与 SS 堆栈段寄存器一起形成栈顶存储单元的物理地址。

BP（Base Pointer）基址指针寄存器：用来指示堆栈中某个数据区的偏移地址—基地址，与 SS 堆栈段寄存器一起形成堆栈中某个存储单元的物理地址。

SI（Source Index）源变址寄存器和 DI（Destination Index）目的变址寄存器：这两个寄存器与 DS 数据段寄存器一起用来确定数据段中某一存储单元的物理地址。这两个寄存器都有自动增量和自动减量功能，用于变址是很方便的。在串处理指令中，SI 和 DI 作为隐含的源变址寄存器和目的变址寄存器，此时 SI 和 DS 联用，DI 和 ES 附加段寄存器联用，分别达到在数据段中和在附加段中寻址的目的。

3）段寄存器。段寄存器包括 CS、SS、DS 和 ES 四个 16 位的段寄存器。

CS（Code Segment）代码段寄存器；

SS（Stack Segment）堆栈段寄存器；

DS（Data Segment）数据段寄存器；

ES（Extra Segment）附加段寄存器。

4）IP（Instruction Pointer）指令指针寄存器。它用来存放代码段中的偏移地址。在程序运行过程中，它始终指向下一条指令的首地址。它与 CS 寄存器联合确定下一条指令的物理地址。当这一地址送到存储器后，控制器可以取得下一条要执行的指令，而控制器一旦取得这条指令，马上修改 IP 的内容，使它指向下一条指令的首地址。计算机就是用 IP 寄存器来控制指令序列的执行流程的，因此 IP 寄存器是计算机中很重要的一个控制寄存器。

（2）8086 的程序状态字寄存器 PSW 的状态标志位和控制标志位的名称、符号和功能，状态标志的状态符号的表示

控制寄存器分为两个 16 位的寄存器 IP 和 PSW。

PSW（Program Status Word）程序状态字寄存器（或称标志寄存器）：是一个 16 位寄存器，由状态（或称条件码）标志（flag）和控制标志所构成，如下所示：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
×	×	×	×	OF	DF	IF	TF	SF	ZF	×	AF	×	PF	×	CF

其中状态标志用来记录程序运行结果的状态信息。由于这些状态信息往往用作后续条件转移指令的转移控制条件，所以又可称为条件码。它包括以下 6 位：

OF（Overflow Flag）溢出标志：在运算过程中，如运算结果已超出了机器能表示的数值范围（指有符号数），称为溢出。此时 OF=1；否则，OF= 0。

SF（Sign Flag）符号标志：记录运算结果的符号，结果为负时置 1；否则置 0。

ZF（Zero Flag）零标志：运算结果为零时，ZF= 1；否则，ZF= 0。

CF（Carry Flag）进位标志：记录运算时从最高有效位产生的进位值或借位值。最高有效位有进位或借位时，CF=1；否则，CF= 0。

AF（Auxiliary Carry Flag）辅助进位标志：记录运算时第 3 位（半个字节）产生的进位值或借位值。第 3 位有进位或借位时，AF=1；否则，AF= 0。这个标志用于十进制算术运算指令 DAA、DAS、AAA 和 AAS。

PF（Parity Flag）奇偶标志：用来为计算机串行传送时可能产生的代码出错情况提供检验条件。当结果操作数（一个字节数据或字数据的低 8 位）中“1”的个数为偶数时，PF= 1；否则，PF= 0。

控制标志位有三个：

DF（Direction Flag）方向标志：在串处理指令中控制处理信息的方向用。当 DF= 1 时，每次操作后使 SI 和 DI 减量，这样就使串处理从高地址向低地址方向进行；当 DF= 0 时，则使 SI 和 DI 增量，使串处理从低地址向高地址方向进行。

IF（Interrupt Flag）中断标志：当 IF = 1 时，允许中断；否则，关闭中断。

TF（Trap Flag）跟踪标志：用于程序调试时进行单步方式工作。当 TF = 1 时，每条指令执行完后产生一个内部中断，允许程序在每条指令执行后，可以进行检查；TF = 0 时，CPU 正常工作不产生内部中断。

以上就是 PSW 中各种状态和控制标志的含义。其中控制标志是由系统程序或用户程序根据需要用指令来设置的。8086/8088 指令系统中提供了设置部分状态和控制标志的指令，给用户在程序中使用这些指令来设置某些标志。

（3）8086 存储器的组织形式和特点，存储器地址的分段，存储单元物理地址的形成方法

计算机存储信息的基本单位是一个二进制位，一位可存储一个二进制数 0 或 1。每 8 位组成一个字节。

8086/8088 的字长为 16 位，由 2 个字节组成。

在存储器里以字节为单位存储信息。为了正确地存放和取得信息，每一个字节单元给予一个存储器地址，地址从 0 开始编号，顺序地每隔一个单元加上 1。在计算机中，地址也是用二进制数来表示的。它是一个无符号的整数，书写格式通常为十六进制数。

一个存储单元中存放的信息（数据）称为该存储单元的内容。

一般来说，字单元的地址可以是偶数，也可以是奇数。但是，在 8086 中，访问存储（取数或存数）都是以字单元进行的。也就是说，机器是以偶地址访问存储器的，对于要取一个奇地址的字单元中的数据，CPU 必须访问两次存储器，花费的时间要多一倍。

8086/8088 的存储器组织是将存储器划分为段，每个段的大小可在 64K 范围内选取任意个字节，段内地址可以用 16 位表示（64K 范围内）。对段的起始地址有所限制，段不能起始于任意地址，而必须从任意一小段（Paragragh）的首地址开始。机器规定，从 0 地址开始，每 16 个字节为一小段，下面列出了存储器最低地址区的三个小段的地址区间，每一行写为一小段中的 16 个地址。

存储单元的五位物理地址（16 进制数）是由 16 位段地址和 16 位偏移地址组成。段地址是指每一段的起始地址，由于它必须是小段的首地址，所以段的起始地址的低四位一定是 0000；因此，就规定了段地址只取段起始地址的高 16 位值。偏移地址则是指在段内相对于段起始地址的偏移量，此偏移地址也是 16 位值。因此，存储单元的物理地址的计算方法表示如下：

物理地址=段地址×16+偏移地址

3. 汇编语言程序设计的特点和作用

（1）学习汇编语言程序设计的目的和意义

1）计算机的指令系统是计算机的重要性能特征；

2）学习与计算机硬件有密切关系的课程，必须具备汇编语言及汇编语言程序设计方面的知识；

3）汇编语言及汇编语言程序设计在微型机应用中占有重要地位。

（2）汇编语言程序设计的特点和作用

汇编语言用操作内容的英文词的缩写符号代替二进制编码，用符号代替地址或操作的数据。汇编语言书写的指令与机器语言书写的指令仍然是一一对应的。显然，汇编语言书写的指令易于为人们所理解和便于书写，但是汇编语言书写的指令和程序必须经过翻译程序，汇编程序翻译成二进制代码的指令和程序——目标程序，计算机对目标程序才能识别和执行。本课程的主要内容就是介绍汇编语言的指令、指令系统和汇编语言程序设计的一系列问题。

汇编语言编写的程序可直接被计算机硬件识别和执行（当然，必须翻译为目标程序）。也就是说，汇编语言编写的程序是面向机器的。

1.2.2 8086 的寻址方式和指令系统

1. 寻址方式的定义

（1）寻址方式的含义和实质

计算机中的指令由操作码字段和操作数字段两部分组成。操作码字段指出计算机所要执行的操作，而操作数字段则指出在指令操作过程中所需的操作数据。

从程序设计的通用性来看，操作数或操作数存放的地址在指令中的指定应具有易于改变的灵活性，需要有多种方式来指定操作数或操作数地址。指令中用于说明操作数所在地址的方法，称为寻址方式（Addressing mode）。

当操作数是存放在存储器中时，存储器的存储单元的物理地址由两部分组成。一部分是偏移地址；一部分是段地址。在 8086/8088 的各种寻址方式中，寻找存储单元所需的偏移地址可由各种成分组成，称为有效地址，用 EA 表示。

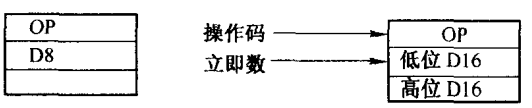
（2）指令中寻址的操作数可分为三种：寄存器操作数，立即数，存储器操作数

指令中操作数字段实质上指出参加操作运算的操作数存放于何处。一般说来，操作数可以存放在指令代码中，称为立即数。操作数也可存放在 CPU 的内部寄存器中，称为寄存器操作数。操作数绝大部分是存放在内部存储器中，称为存储器操作数。

2. 与数据有关的寻址方式

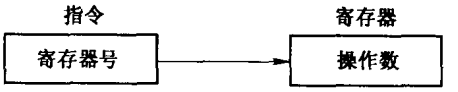
（1）8086 中与数据有关的六种寻址方式的名称和含义

1）立即寻址方式（Immediate Addressing）。操作数直接存放在指令中，紧跟在操作码之后，它作为指令的操作数字段存放在指令代码中，这种操作数称为立即数。立即数可以是 8 位的或 16 位的。如果是 16 位立即数，则低位字节数存放在低地址单元中，高位字节数存放在高地址单元中。机器码存放形式如下所示：

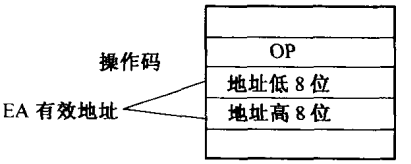


立即寻址方式的操作数用来表示常数，它经常用于给寄存器赋初值，并且只能用于源操作数字段，不能用于目的操作数字段。

2）寄存器寻址方式（Register Addressing）。操作数在寄存器中，指令指定寄存器号。对于 16 位操作数，寄存器可以是 AX、BX、CX、DX、 SI、 DI、SP 和 BP 等；对于 8 位操作数，寄存器可以是 AH、AL、BH、BL、CH、 CL、DH 和 DL 等。这种寻址方式由于操作数在寄存器中，不需要访问存储器来取得操作数，因而可以取得较高的运算速度。这种寻址方式寻找操作数的示意图如下所示：



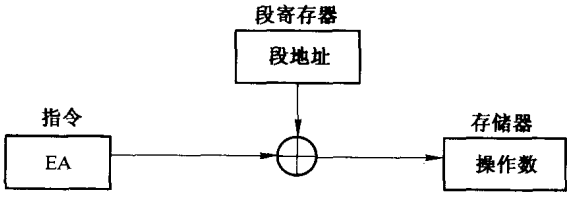
3）直接寻址方式（Direct addressing）。在直接寻址方式中，有效地址 EA 就在指令的代码段中，它存放在代码段中指令操作码后面的操作数字段，机器码可表示如下：



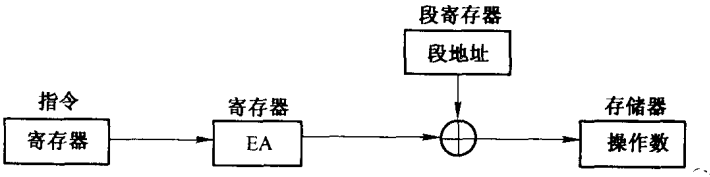
通常操作数的物理地址为：

$$物理地址 = DS \times 16 + EA$$

这种寻址方式寻找寄存器的操作示意图如下图所示：



4）寄存器间接寻址方式（Register Indirect Addressing）。操作数的有效地址在基址寄存器 BX、BP 或变址寄存器 SI、DI 中，操作数则在存储器中，此种寻址方式如下图所示：



如果指令中指定的寄存器是 BX、SI 和 DI，则操作数必定在数据段中，以 DS 段寄存器的内容作为段地址，操作数的物理地址为：

$$物理地址 = 16 \times DS + \begin{matrix} BX \\ SI \\ DI \end{matrix}$$

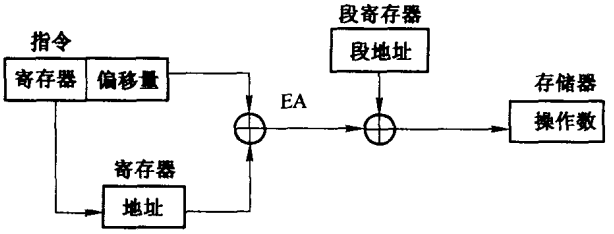
如果指令中指定的寄存器是 BP，则操作数必定在堆栈段中，以 SS 段寄存器的内容作为段地址，操作数的物理地址为：

物理地址= 16×SS+BP

5）寄存器相对寻址方式（Register Relative Addressing）（或称直接变址寻址方式）。操作数的有效地址是一个基址或变址寄存器的内容和指令中指定的 8 位或 16 位偏移量（displacement）之和。即：

$$EA=\left\{\begin{array}{l}BX\\BP\\SI\\DI\end{array}\right\}+\left\{\begin{array}{l}8\text{ 位偏移量}\\16\text{ 位偏移量}\end{array}\right\}$$

这种寻址方式如下图所示：



其物理地址为：

$$\text{物理地址}=16\times DS+\left\{\begin{array}{l}BX\\SI\\DI\end{array}\right\}+\left\{\begin{array}{l}8\text{ 位偏移量}\\16\text{ 位偏移量}\end{array}\right\}$$

或

$$\text{物理地址}=16\times SS+BP+\left\{\begin{array}{l}8\text{ 位偏移量}\\16\text{ 位偏移量}\end{array}\right\}$$

6）基址变址寻址方式（Based Indexed Addressing）。操作数的有效地址是一个基址寄存器和一个变址寄存器的内容之和。两个寄存器均由指令指定。如基址寄存器为 BX，段寄存器使用 DS；基址寄存器为 BP，段寄存器使用 SS，因此物理地址为：

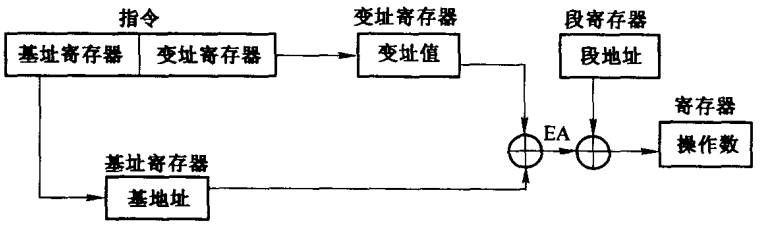
物理地址= 16×DS + BX + SI

或　物理地址= 16×DS + BX + DI

或　物理地址= 16×SS + BP + SI

或　物理地址= 16×SS + BP + DI

这种寻址方式如下图所示：



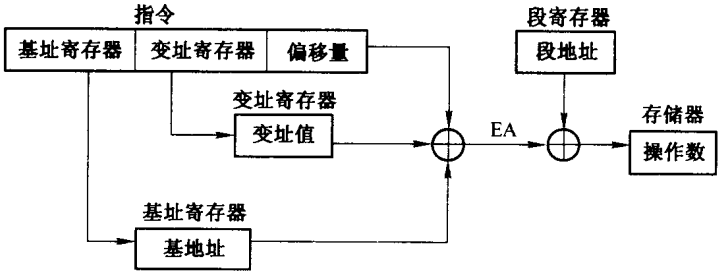
7）相对基址变址寻址方式（Relative Based Indexed Addressing）。操作数的有效地址是

一个基址寄存器的内容，一个变址寄存器的内容和 8 位或 16 位偏移量之和。同样，当基址寄存器为 BX 时，段寄存器用 DS；当基址寄存器为 BP 时，则段寄存器使用 SS。因此操作数的物理地址为：

$$\text{物理地址}=16\times DS+BX+\begin{Bmatrix}SI\\DI\end{Bmatrix}+\begin{Bmatrix}8\text{位偏移量}\\16\text{位偏移量}\end{Bmatrix}$$

$$\text{物理地址}=16\times SS+BP+\begin{Bmatrix}SI\\DI\end{Bmatrix}+\begin{Bmatrix}8\text{位偏移量}\\16\text{位偏移量}\end{Bmatrix}$$

这种寻址方式的示意图如下图所示：



（2）各种寻址方式的操作数书写格式，各种寻址方式的存储器操作数的有效地址形成的方法和书写格式。

内容包含在（1）中。

（3）分析指令中各种寻址方式操作数据的出处和去处，根据要求在指令中写出各种寻址方式的操作数据。

内容包含在（1）中。

3. 与转移地址有关的寻址方式

这种寻址方式用来确定转移指令或 CALL 指令的转移地址。转移地址是由各种寻址方式得到的有效地址（即偏移地址）和段地址相加而成。有效地址存入 IP 寄存器中，段地址指定为 CS 段寄存器内容。

（1）8086 的指令系统中与转移地址有关的四种寻址方式的名称和含义，各种寻址方式转移地址的书写格式和转移范围的书写格式。

1）段内直接寻址（Intrasegment Direct Addressing）。转移地址的有效地址是当前 IP 寄存器内容和指令中指定的 8 位或 16 位偏移量之和。当偏移量为 8 位时，称为短跳转，如：

JMP SHORT QUEST

2）段内间接寻址（Intrasegment Indirect Addressing）。转移地址的有效地址是一个寄存器的内容或存储单元的内容。存储单元内容可以用数据有关的寻址方式中寻找存储器操作数的任何一种寻址方式取得。所取得的有效地址替代 IP 寄存器的内容，达到转移的目的。如：

JMP BX

3）段间直接寻址（Intersegment Direct Addressing）。指令中直接提供了转移地址的段地址和偏移地址，所以只要用指令中指定的偏移地址取代 IP 寄存器内容，用指令中指定的段