



普通高校系列教材·信息技术

汇编语言程序设计

普通高校教材（信息技术）编委会主编

作者：姚君遗



南京大学出版社

全国高等教育自学考试教材
计算机及其应用专业(专科)

汇编语言程序设计

主 编 姚君遗
副主编 张维勇
主 审 佟英泰

南京大学出版社
1995·南京

内

容

简

介

本书详细地阐述和讨论了 8086/8088 汇编语言程序设计的原理、方法和技巧。主要包括:8086/8088 硬件编程模型,8086/8088 寻址方式和指令系统,汇编语言程序格式,顺序、分支、循环和子程序的基本结构形式及程序设计,关于数制转换、算术运算、字符串和表格处理等实用程序的程序设计。

本书可作为高等院校相关专业的教材,也适用于以自学为主的读者和工程技术人员学习汇编语言程序设计的教材和参考书。

图书在版编目(CIP)数据

汇编语言程序设计/姚君遗主编. —南京:南京大学出版社,2001

ISBN 7-305-02681-6

I. 汇... II. 姚... III. 汇编语言—程序设计 IV. TP313

中国版本图书馆 CIP 数据核字(2001)第 063487 号

汇编语言程序设计

姚君遗 主编

佟英泰 主审

*

南京大学出版社出版发行

(南京大学校内 邮政编码:210093)

合肥学苑印刷厂印刷

*

开本:787×1092 1/16 印张:19 字数:456 千

2002 年 1 月第 5 次印刷

ISBN 7-305-02681-6/TP·109

定价:24.00 元

普通高校教材(信息技术)编委会

主 任:孙钟秀 中国科学院院士

副主任:张福炎 南京大学教授

陈国良 中国科学技术大学教授

钱洲胜 中国计算机函授学院院长

委 员(按姓氏笔划排序):

王佩珠	西安交通大学	王文兰	桂林电子工业学院
王蔚韬	重庆建筑大学	伍良富	成都电子科技大学
成松林	东南大学	刘存书	郑州信息工程大学
朱大奇	安徽工业大学	朱宝长	西安电子科技大学
孙德文	上海交通大学	杜象元	上海交通大学
李茂青	厦门大学	李学干	西安电子科技大学
杨来利	兰州大学	何淑兰	北京科技大学
张凤祥	华中科技大学	张绍林	河北行政学院
张维勇	合肥工业大学	张民坤	云南工业大学
张景书	哈尔滨工程大学	邹华跃	中国计算机函授学院
赵良全	新疆大学	皇甫正贤	东南大学
洪志全	成都理工学院	姚君遗	合肥工业大学
高 平	浙江大学	陶世群	山西大学
曹翊旺	湖南省计算机高等专科学校	梁文康	山东大学
韩国强	华南理工大学	舒 洪	南昌大学
葛 燕	中国科学技术大学	解世耀	辽宁大学
谭耀铭	南京大学	黎庆国	合肥工业大学

出版前言

近些年来我国的高等教育事业有了长足的发展,高校招生人数年年递增,越来越多的年轻人有机会接受正规的高等教育。这一举措无疑对我国的社会进步和经济发展有着重要的意义。但是人们也深刻的认识到,高等教育质量的好坏是一个不容忽视的关键性问题,而保证教育质量的一个重要环节就是抓好教材建设。但是教材内容陈旧,教学手段落后的现象一直存在着。尤其像计算机技术这样的新兴领域发展迅猛,知识更新日新月异,教学内容落后于新技术新知识的矛盾显得尤为突出。基于上述两方面考虑,在南京大学出版社的鼎力相助下,一个以组编高校信息、电子类专业教材为主要任务的教材编委会成立了。

针对我国高等教育的现状和信息、电子技术的发展趋势,编委会组织部分高校的专家教授进行了深入的专题研讨。大家一致认为,在当前情况下组编一套紧跟新技术发展、符合高校教学需要、满足大学生求知欲望的系列教材势在必行,这将有助于规范教学体系、更新教学内容、把握教学质量,培养合格人才。专家们还对教材的结构、内容、体例及配套服务等提出了具体要求。为了能使这套教材逐步完善,并促进全国各地高校教学质量的提高,编委会决定在教材之外认真做好三件事:第一、为每本教材配备一本供学生使用的学习参考书,其主体内容为学习方法指导、习题分析与解答、典型题解或课程设计、模拟测试卷及解答、实验指导书;第二、定期对教材内容进行修订,及时补充新技术新知识,并根据具体情况组编新的教材;第三、有计划地组织各地高校教师进行教学交流与研讨,通过这种途径来提高偏远地区的师资水平。我们相信,通过各方面的大力支持和大家的不懈努力,这套教材会逐步被广大师生所接受,并在使用过程中得到完善、充实。

大家都知道,组编这样一套系列教材是个牵涉面很广的大工程。这个工程不仅在起步阶段需要得到各级教育主管部门、各高等院校、出版社的大力支持和协助,而且在使用过程中也离不开各位专家、教授、学生的热心呵护和指导。因此,殷切期待所有的能人志士关心我们,帮助我们,向我们提出好的建议或意见,为我们指出教材中的不足之处。

最后,感谢所有为本套系列教材出版付出辛勤劳动的同志们。

普通高校教材(信息技术)编委会

2001年8月

编 者 的 话

本书是根据国家教委高等教育自学考试指导委员会电类专业委员会组织制定的计算机及其应用专业(专科)《汇编语言程序设计自学考试大纲》而编写的,并被认定为全国高等教育自学考试教材。按照自学考试大纲的基本要求,本书在编写中注重基础,突出重点,精选内容,以期使自学应考者在基本理论、基本知识和基本技能方面得以训练。

本课程是计算机及其应用专业的核心课程之一,通过本课程的学习,为操作系统、微型机及其应用、计算机控制等课程打好汇编语言基本理论和基本编程方法的基础。

汇编语言是一种速度快、功能强的程序设计语言,它适用于编制控制和使用计算机硬件和外部设备的系统程序和应用程序。

本教材是以 8086/8088 指令系统和 ASM_86 汇编语言为基础,阐述和讨论了硬件编程模型,寻址方式和指令系统,汇编语言程序格式,四种基本程序结构形式——顺序程序、分支程序、循环程序和子程序的汇编语言程序设计的原理和方法。最后以大量的实例对汇编语言实用程序,如数制转换,算术运算,字符串和表格处理等进行了详细的分析和讨论,读者可从中得到汇编语言程序设计的方法和技巧的有益知识。

教材中第九章结构化程序设计简介不作为自学考试的必修内容,这是一种较高级的编程方法和技巧,作为知识面的拓宽。

本课程是一门实践性很强的课程,教材内容中增设了大量的实例,每章都有较多的习题,教材中也列出了汇编语言的上机过程等内容。读者在阅读本教材时,可结合实例和习题进行练习和编程,有条件的应上机练习。

本书第一章——第五章由姚君遗同志编写,第六章——第九章由张维勇同志编写。

本书在编写过程中,得到国家教委高等教育自学考试指导委员会电类专业委员会宋文涛教授和陈敏逊教授的支持和指导。得到中国计算机函授学院的钱洲胜院长、牛允鹏教授和有关同志们的大力支持和帮助。

本书由安徽省计算机水平考试委员会佟英泰教授主审,中国科技大学杨寿保副教授和合肥工业大学王定安副教授参加审阅。他们在百忙中仔细审阅了全书,并提出了宝贵的建议和意见。

对他们热情指导和审阅,一并表示衷心的感谢。

因编者经验不足,水平有限,缺点和错误之处定有不少,敬请广大读者指正,以待改进。

编 者

一九九五年元月于合肥

目 次

第一章 基础知识	(1)
§ 1.1 计算机系统概述	(1)
§ 1.2 中央处理单元——微处理器	(4)
§ 1.3 8086/8088 的存储器组织	(7)
§ 1.4 外部设备.....	(11)
小结	(12)
习题	(14)
第二章 8086/8088 的寻址方式和指令系统	(16)
§ 2.1 概述.....	(16)
§ 2.2 8086/8088 的寻址方式	(18)
§ 2.3 8086/8088 指令系统	(29)
小结	(97)
习题	(101)
第三章 8086 汇编语言程序格式	(107)
§ 3.1 汇编语言的语句	(107)
§ 3.2 指令语句	(118)
§ 3.3 指示性语句	(123)
§ 3.4 宏指令语句	(136)
§ 3.5 汇编语言程序的上机过程	(139)
小结.....	(148)
习题	(151)
第四章 顺序程序设计	(155)
§ 4.1 汇编语言程序设计的基本步骤	(155)
§ 4.2 顺序程序的结构形式和程序设计	(160)
小结.....	(165)
习题	(166)
第五章 分支程序设计	(168)
§ 5.1 分支程序的结构	(168)
§ 5.2 双分支程序设计	(170)
§ 5.3 多分支程序设计	(175)
小结.....	(184)
习题	(185)
第六章 循环结构程序设计	(188)
§ 6.1 概述	(188)

§ 6.2	循环程序的基本结构形式与设计方法	(193)
§ 6.3	单重循环	(196)
§ 6.4	多重循环	(199)
§ 6.5	循环程序的控制方法	(203)
小结		(211)
习题		(211)
第七章	子程序设计	(215)
§ 7.1	概述	(215)
§ 7.2	子程序的结构形式	(216)
§ 7.3	子程序设计方法	(218)
§ 7.4	DOS 系统功能调用简介	(232)
§ 7.5	子程序设计举例	(238)
小结		(245)
习题		(245)
第八章	汇编语言实用程序	(248)
§ 8.1	数制和代码转换	(248)
§ 8.2	算术运算	(259)
§ 8.3	字符串操作	(269)
§ 8.4	表格的处理和应用	(273)
小结		(280)
习题		(280)
附录一	8086/8088 指令系统一览表	(282)
附录二	DOS 系统功能调用(INT 21H)	(288)

第一章

基础知识

[内容提要] 本章简要地介绍微型计算机的系统组成,包括系统硬件和系统软件。重点介绍汇编语言程序设计所应具备的编程硬件模型——8086/8088的CPU内部可编程寄存器的组成、分类、名称、字长和功能,内部存储器的组织——分段结构和存储单元物理地址的组成成分及物理地址的形成,外部设备及其端口地址。

§ 1.1 计算机系统概述

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、外部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。

一、硬件

典型的计算机的基本结构框图如图1—1所示。其中包括中央处理器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。

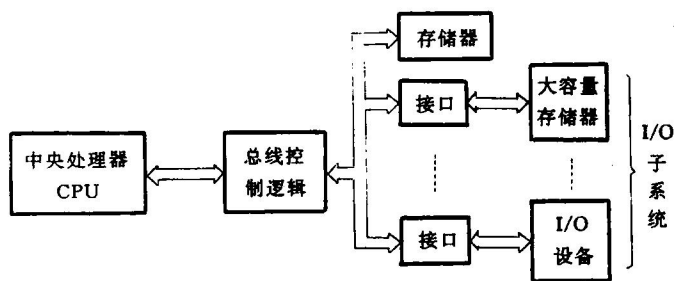


图 1—1 计算机结构框图

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)就存放在这里。它也

可以存放程序中所用的数据(原始数据和结果数据)、信息及中间结果。一般称这种在机器内部的存储器为内存。

中央处理器包括运算器和控制器内部可编程寄存器组。运算器执行所有的算术和逻辑运算指令。控制器则负责全机的控制操作,它负责把指令逐条从存储器中取出,经译码分析后向全机发出取数、执行和存数等控制命令,以保证正确完成程序所要求的功能。

I/O 子系统(输入/输出)一般包括 I/O 设备及大容量存储器两类外部设备。I/O 设备是指与计算机的外部世界通信用的输入/输出设备,如显示终端、键盘输入和打印输出等多种类型的外部设备。大容量存储器则是指可存储大量信息的外部存储器,如磁盘、磁带和光盘等。由于内存容量有限,计算机用外部存储器作为内存的后援设备,它的容量比内存大得多,但存取信息的速度比内存慢得多。除了必要的系统程序(如 DOS 的引导程序及直接和输入/输出设备进行数据交换的程序)是存放在内存中以外,一般程序(包括数据)是存放在外部存储器中的。只有当运行时,才把它们从外部存储器送到内存的某个区域,由中央处理器 CPU 控制执行。

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。数据总线传送数据(包括指令代码、原始数据、中间数据和结果数据),地址总线上的信息(即地址)指出数据的来源和目的地,控制总线传送 CPU 对存储器或 I/O 设备的控制命令和 I/O 设备对 CPU 的请求信号。系统总线的工作由总线控制逻辑(在 CPU 的控制器内)负责指挥。

IBM PC 机是美国 IBM 公司在 1981 年推出的个人计算机,不久就升级换代为 IBM PC/XT、286 微机、386 微机以至于发展到现在的 586 微机。作为汇编语言程序设计的基础,我们还是选用 8086/8088 作为 CPU 的基本机型作个介绍。IBM PC 机内有一片微处理器 8088 作为 CPU,其主要性能是:系统时钟 4.77MHz,基本指令 99 条。机内字长 16 位,外部数据总线只有 8 根。地址总线有 20 根,提供了 1M 字节的寻址能力。平均运算速度为 0.65MPS(兆次/秒)。机内有 1M 字节的随机存取存储器(RAM)和 40K 字节的只读存储器(ROM)。ROM 中存放有 BASIC 解释程序及基本输入/输出系统软件 BIOS。它们是只允许读出而不允许用户写入或修改的。IBM PC 机提供了三种基本外设接口:键盘、扬声器和盒式磁带机。允许用户配置各种外部设备,如 CRT 显示器、硬磁盘驱动器、软磁盘驱动器和打印机等。

二、软件

计算机软件是计算机系统的重要组成部分,它可分为系统软件 and 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行编制的各种程序。图 1—2 表示了计算机软件的分层结构。

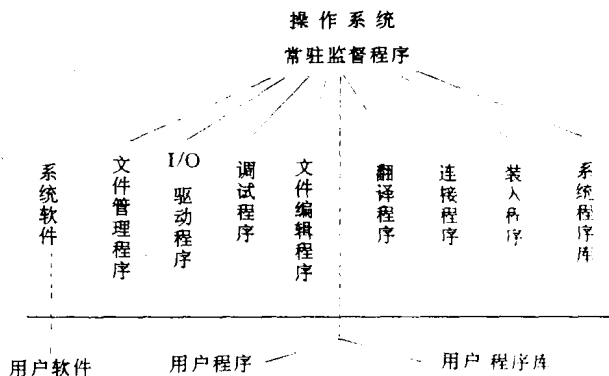


图 1—2 计算机软件层次图

层次。下面我们简要地介绍系统软件的组成。

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。

I/O 驱动程序(I/O driver):用来对 I/O 设备进行控制和管理。当系统程序或用户程序需要使用 I/O 设备,只要发出命令,执行 I/O 驱动程序,完成 CPU 和 I/O 设备之间的信息传送。

文件管理系统(File Management):用来处理存放在外存储器中的大量信息,它可以和外存储器的设备驱动程序相连接,对存放在其中的信息以文件(File)形式进行存取,复制及其它管理操作。

文件编辑程序(Text Editor):文本是指由字母、数字和符号等组成的一组信息,它可以是一个用汇编语言或高级语言编写的程序,也可以是一组数据或一份报告。文本编辑程序用来建立、输入或修改文本,并使它存入存储器或外存储器中(磁盘)。例如 IBM PC 机提供的行编辑程序 EDLIN 可用来建立源文件、修改文本,有插入、删除、编辑和显示行等功能。字处理程序 WORDSTAR 和中西文字处理程序 WPS 等可提供屏幕编辑功能,并能提供各种功能及命令菜单,使文本的建立和修改更加方便。

翻译程序(Translator):计算机是通过逐条执行组成程序的指令来完成人们所给予的任务的,所以指令就是计算机能识别并能直接执行的命令,它是由二进制代码书写的。用这样一种二进制代码编写指令和程序的语言称为机器语言。显然这种语言书写的指令既烦琐又难于记忆,人们使用时很不方便,但这是计算机能识别的唯一的一种语言。在计算机语言的发展过程中就出现了汇编语言和高级语言。汇编语言是一种符号语言,它书写的指令和机器语言书写的指令是一一对应的,它在书写时使用字符串组成的助记符来代表操作的内容,而不是像机器指令那样用二进制编码来代表操作的内容,例如加法在汇编语言中是用助记符 ADD 表示的。而机器语言则用 6 位二进制代码来表示。显然,相对于机器语言来说,汇编语言是易于为人们所理解的,当然计算机却不能直接识别汇编语言。汇编程序是一种翻译程序,它将把用户用汇编语言编写的程序(称为源程序)翻译成机器语言程序(称为目标程序),它是一种系统程序。IBM PC 机中的汇编程序有小汇编程序 ASM 和宏汇编程序 MASM 两种,ASM 占有较小的存储区,功能较弱。MASM 需要在存储区较大,功能较强,具有宏汇编能力,而 ASM 不具备这种能力。

高级语言完全脱离机器指令,用人们更加易于理解的方式编写程序,更接近于科学计算的公式及问题,所以我们有时候说,机器语言和汇编语言是一种面向机器的语言,而高级语言是面向科学计算和实际问题的语言。当然高级语言编写的程序不能直接由计算机来执行,必须由翻译程序把它翻译为机器语言的程序,计算机才能执行。高级语言的翻译程序有二种形式:一种是将高级语言的源程序一边进行解释,一边执行,这种翻译程序称为解释程序(Interpreter),如 BASIC 经常采用这种形式。另一种是先把高级语言编写的源程序翻译成机器语言程序,然后再在机器上执行,这种翻译程序称为编译程序(Compiler),多数高级语言如 FORTRAN、PASCAL 等都采用这种形式,编译程序和解释程序都

属于系统程序。

连接程序(Linker):用来把要执行的程序与库文件(子程序库中的子程序)或其它已翻译过的子程序(能完成一种独立功能的程序模块)连接在一起,形成机器能执行的程序。

装入程序(Loader):用来把程序从外存储器传送到内存,以便机器执行。例如,计算机开机后需要立即启动装入程序把常驻监督程序装入内存,使机器运转起来。又如,用户程序经翻译和连接后,由连接程序直接调用装入程序,把可执行的用户程序装入内存以便执行。

调试程序(Debug):是系统提供给用户的能监督和控制用户程序的一种工具程序,它可以装入、修改、显示或逐条或连续执行一个程序。在 IBM PC 机上,简单的汇编语言程序可以通过调试程序来建立、修改和执行。

系统程序库(System Library)和用户程序库(User Library):各种标准程序、子程序及一些文件的集合称为程序库,它可以被系统程序或用户程序调用。操作系统还允许用户建立程序库,以提高不同类型用户的工作效率。

§ 1.2 中央处理单元——微处理器 8086/8088

一、微处理器 8086/8088 的组成

Intel 8086/8088 是两种第三代微处理器。在汇编语言一级,它们与 8080/8085 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。其它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。IBM PC 机及兼容机上广泛地采用了 8088。

CPU 的任务是执行存放在内存中的指令序列。它由运算器、控制器和内部可编程寄存器组所组成。Intel 8086/8088 微处理器的组成示意图如图 1-3 所示。

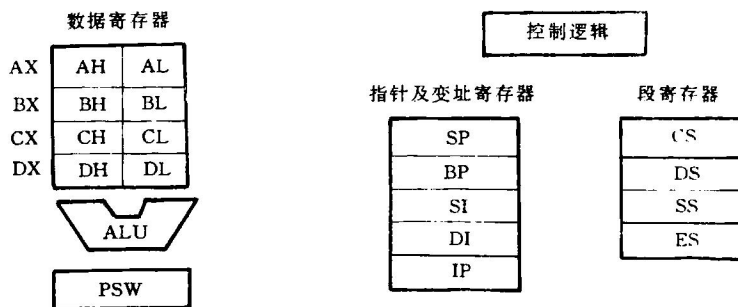


图 1-3 8086/8088 CPU 组成

从图中可见 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 传送的控制等。

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运算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置数或读数。下面我们详细介绍这些工作寄存器。

二、8086/8088 的寄存器组

1. 数据寄存器(或称通用寄存器)

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、结果数据或其它信息。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。

从图 1-3 可知,以字形式使用时,四个通用寄存器分别称为 AX、BX、CX、DX;以字节形式使用时,高 8 位通用寄存器分别称为 AH、BH、CH、DH。低 8 位通用寄存器分别称为 AL、BL、CL、DL。

这四个寄存器都是通用寄存器,但它们又可以用于专用的目的。

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,在作双字长运算时,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

它们包括 SP、BP、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;

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

它们包括 CS、SS、DS、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。

8086/8088 采用存储空间的分段技术来解决寻址 1M 字节的存储空间。这些段寄存器的内容和有效的地址偏移量(称为偏移地址)一起即可确定内存的存储单元的物理地址。通常 CS 划定并控制程序区, DS 和 ES 划定和控制数据区, SS 划定和控制堆栈区, 究竟如何确定内存单元的物理地址详见 § 1.3。

4. 控制寄存器

控制寄存器分为两个 16 位的寄存器 IP 和 PSW。

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。

PSW(Program Status Word)程序状态字寄存器(或称标志寄存器):

这是一个 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。

在调试程序 DEBUG 中提供了测试标志位的手段,它用符号来表示标志位的值。表 1-1 说明每种标志位的符号表示(DEBUG 不提供 TF 的符号,表中未列入)。

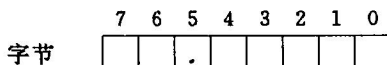
表 1-1 PSW 中标志位的符号表示

标志名	标志为 1	标志为 0
OF 溢出(是/否)	OV	NV
DF 方向(减量/增量)	DN	UP
IF 中断(允许/关闭)	EI	DI
SF 符号(负/正)	NG	PL
ZF 零(是/否)	ZR	NZ
AF 辅助进位(是/否)	AC	NA
PF 奇偶(偶/奇)	PE	PO
CF 进位(是/否)	CY	NC

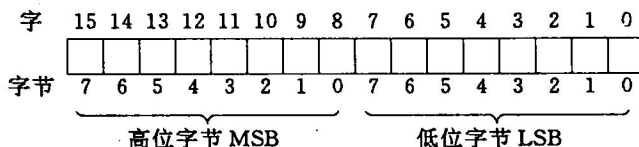
§ 1.3 8086/8088 的存储器组织

一、存储单元的地址和内容

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。位编号如下所示:



8086/8088 的字长为 16 位,由 2 个字节组成,位编号如下所示:



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。注意,它是一个无符号的整数,书写格式通常为十六进制数。

上节已提到,存放存储器地址的寄存器字长为 16 位。因此每个存储单元若用 16 位二进制数表示地址,那么 16 位的二进制数可以表示多少个存储单元呢? 应该是 $2^{16}=65536$ 个。在计算机里,为方便起见,总是以 $2^{10}=1024$ 个存储单元作为存储器容量的一种基本单位,称为 1K(或 1K 字节)。这样 65536 个存储单元的存储器存储容量就是 64K(或 64K 字节,或 64KB)。所以存储单元地址编号用十六进制数表示为 0000H~FFFFH,表示如下:

0000,0001,0002,……,000D,000E,000F

0010,0011,0012,……,001D,001E,001F

⋮

FFE0,FFE1,FFE2,……,FFED,FFEE,FFEF

FFF0,FFF1,FFF2,……,FFFD,FFFE,FFFF

一个存储单元中存放的信息(数据)称为该存储单元的内容。图 1—4 表示了存储器里存放信息的情况。由图可知:4 号存储单元中存放的信息为 34H。也就是说 4 号存储单元中的内容为 34H,可表示为:

$$(0004H)=34H$$

8086/8088 字长是 16 位,大部分数据都是以字为单位表示的。那么一个字(16 位)怎样存入存储器中呢? 一个字存入存储器要占有相继的二个存储单元。存放时,字的低位字节存入低地址的存储单元,字的高位字节存入高地址的存储单元。也就是说,信息的存入次序和书写次序是相反的。一个字数据为 7B86H,那么存储时,先存入 86H,然后再存入 7BH。存储器中二个字节单元组成一个字单元,字单元的地址采用它的低位字节存储单元的地址表示。图 1—4 中 4 号字单元的内容是 1234H,可表示为:

(0004H)=1234H

可以看出,同一个地址既可看作字节单元的地址,又可看作字单元的地址。那么这个地址究竟是字单元还是字节单元,如何区别呢?这在以后的章节中会详细说明,可在指令中定义操作数是字还是字节来规定。

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,访问存储器(取数或存数)都是以字为单元进行的。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 必须访问二次存储器,花费的时间要多一倍。

如上所述,如果用 X 表示某存储单元的地址,则 X 单元的内容可以表示为(X),假如 X 单元中存放着 Y,而 Y 又是一个地址,则可用(Y)=(X)来表示 Y 单元的内容,如图 1—4 中,

(0004H)=1234H

而 (1234H)=2F1EH

则也可表示为

((0004H))=2F1EH

存储器进行取数或存数,即读或写操作时,从某个单元中读出其内容后,该单元的内容不变,可以重复读出该内容;只有当写入新的内容,原来保存的内容就自动丢失。

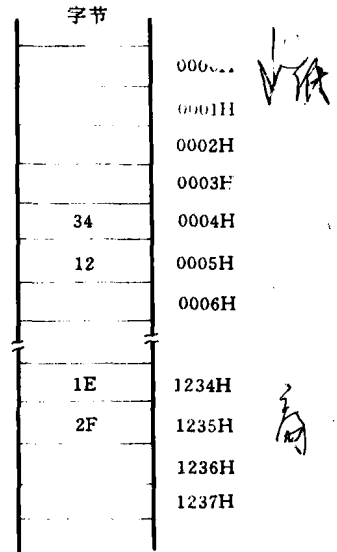


图 1—4 存储单元的
地址和内容

二、存储器地址的分段

8086/8088 微处理器有地址总线 20 根,可访问存储器的最大容量为:

$$2^{20}=1048576=1024K=1M$$

所以用 16 进制数表示 1M 字节的地址范围为 00000H~FFFFFH。那么在 16 位字长的机器里,用什么办法来提供 20 位地址呢?在 IBM PC 机里采用了存储器地址分段的办法。

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 范围内的任意个字节,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(Paragraph)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:

00000、00001、00002,……,0000E,000F

00010、00011、00012,……,0001E,0001F

00020、00021、00022,……,0002E,0002F

其中,第一列就是每小段的首地址,00000H 为第一小段首地址,00010H 为第二小段首地址,00020H 为第三小段首地址。其特征是:在 16 进制数表示的地址中,最低位为 0 (即 20 位地址的最低 4 位为 0000)。在 1M 字节的地址空间中,共有 64K 个小段首地址,可表示如下:

00000H