

21 世纪高职高专计算机系列教材

汇编语言 程序设计

主 编 宋彩利
主 审 陈建铎

西安交通大学出版社

21 世纪高职高专计算机系列教材

汇编语言 程序设计

主 编 宋彩利
编 著 宋彩利 杨君锐
李革新 刘 尊
主 审 陈建铎



西安交通大学出版社
· 西 安 ·

内 容 提 要

本书是高等院校面向 21 世纪高职高专计算机系列教材之一。

本书是以 8086/8088 CPU 为背景,系统讲述了汇编语言指令系统、程序格式及程序设计思路、方法和技术。主要包括计算机系统组成和 PC 机硬件特点,指令系统与寻址方式,顺序程序、分支程序、循环程序和子程序的设计方法以及多模块程序设计和混合语言编程。书中指令分散到相关章节讲述,并用大量图形描述指令执行过程,用例题说明程序设计的思路,便于理解和掌握程序设计的方法。本书中介绍了汇编语言的集成上机环境,可加快程序调试速度。

本书可作为高职高专计算机专业的教材和理工类本科非计算机专业的教材,也可作为从事计算机专业的技术人员的参考书,同时适合自学的读者。

图书在版编目(CIP)数据

汇编语言程序设计 / 宋彩利主编. —西安:西安交通大学出版社,2004.2

(21 世纪高职高专计算机系列教材)

ISBN 7-5605-1824-9

I. 汇... II. 宋... III. 汇编语言-程序设计-高等学校:技术学校-教材 IV. TP313

中国版本图书馆 CIP 数据核字(2004)第 007279 号

- 书 名 汇编语言程序设计
主 编 宋彩利
策划编辑 贺峰涛 屈晓燕
文字编辑 曹银强
出版发行 西安交通大学出版社
地 址 西安市兴庆南路 25 号(邮编:710049)
网 址 <http://unit.xjtu.edu.cn/unit/jtupress>
电 话 (029)82668357 82667874(发行部)
(029)82668315 82669096(总编办)
电子信箱 eibooks@163.com
印 刷 西安建筑科技大学印刷厂
版 次 2004 年 2 月第 1 版 2004 年 2 月第 1 次印刷
开 本 787mm×1 092mm 1/16
印 张 13
字 数 304 千字
书 号 ISBN 7-5605-1824-9/TP·376
定 价 18.00 元

21 世纪高职高专计算机系列教材编委会

顾 问：冯博琴

主 编：陈建铎

副 主 编：谢膺白 王四万 何东健 龚尚福

编 委：(以姓氏笔画为序)

王 津 王四万 王佑元 王晓奇 何东健

张水平 张俊兰 张晓云 李银兴 陈建铎

段宏斌 龚尚福 谢膺白 魏玉梅

策划编辑：贺峰涛 屈晓燕

序

随着我国科学技术的发展,全民高等教育已经成为时代的要求。扩大招生规模,发展高等教育,已经成为各级政府和广大教育工作者的共识。为了指导和推动全国高等职业教育的健康发展,教育部先后制定了“高职高专教育基础课程教学基本要求”和“高职高专教育专业人才培养目标和规格”两个文件。在此基础上,许多出版社先后出版了相关的系列教材,对推动我国的高等职业教育起到了积极的作用。

时代在前进,科学技术在发展,尤其是计算机信息技术发展的速度更是惊人,这就要求高等学校的教学内容应能跟上科学技术的发展,应能满足新技术对新型人才的需求;教材应当不断地修改和更新。故此,我们组织高校中长期从事高等职业教育的专家、学者编写了“21 世纪高职高专计算机系列教材”。在编写过程中,我们以教育部的上述两个文件为依据,参阅同类教材,汲取多年来在高等专科教育、成人教育中培养应用型人才的成功经验,充分体现高职高专实用型人才的特征,“以应用为目的,以必须、够用为度”,尽量做到从实际应用的需求出发,减少枯燥乏味的纯理论和概念,使学生理论联系实际,学中有用,边学边用,通过学习提高应用和解决实际问题的能力。在编排顺序方面,尽量做到由浅入深,循序渐进,内容多样,结构合理,语言简练,文字流畅,使学生易学、易懂、易掌握。

这套教材目前已列入选题的有 19 种,既有专业基础知识,又有最新技术,可作为高职高专基础课、专业基础课以及最新技术课的教材,也可供自考和学历文凭教育使用。

在 21 世纪到来的时候,我国高等职业教育迅猛发展的格局已经形成。这就要求教育界的志士仁人奋发努力,以自己的心血和汗水去培养时代所需要的一代有理想、有道德、有知识、有能力的高素质、高水平的应用型专业人才。

陈建铎

2002 年 10 月

前 言

本书是以 8086/8088 CPU 为背景,系统讲述了汇编语言指令系统、程序格式及程序设计思路、方法和技术。全书共分 9 章。第 1 章基础知识,介绍了计算机系统组成和 PC 机硬件特点;第 2 章寻址方式与指令系统,介绍了 8086/8088 CPU 的寻址方式和常用指令;第 3 章汇编语言程序格式,介绍伪指令、源程序书写格式及汇编语言上机过程;第 4 章顺序程序设计,介绍汇编语言程序设计的方法及流程图画法;第 5 章分支程序设计,介绍转移指令及分支程序设计方法;第 6 章循环程序设计,主要介绍循环指令、串操作及循环程序设计方法;第 7 章子程序设计,介绍调用与返回指令、子程序设计方法及 DOS 系统功能调用;第 8 章输入输出和中断,介绍输入输出命令、中断基本概念、中断程序设计及 DOS 和 BIOS 调用;第 9 章多模块程序设计,介绍多模块程序设计方法及混合语言编程。此外,书末附录部分还给出了 DOS 系统功能调用表和 BIOS 调用表。

本书主要特点为:第一,用图形描述指令执行过程,形象直观;第二,将指令分散到各章节讲述,易掌握其用途;第三,利用大量例题说明程序设计,便于理解和掌握程序设计的思路与方法;第四,介绍汇编语言的集成上机环境,有利于掌握程序调试过程,提高应用能力。本书第 1,2 章由西安石油大学计算机学院宋彩利老师编写,第 3,4 章由西北工业大学刘尊老师编写,第 5,6 章由陕西工业职业技术学院李革新老师编写,第 7,8,9 章由西安科技大学杨君锐老师编写。全书由宋彩利老师统稿,陈建铎教授审稿。

目 录

第 1 章 基础知识

1.1 计算机系统概述	(1)
1.1.1 硬件系统	(1)
1.1.2 软件系统	(2)
1.2 8086/8088 中央处理器	(2)
1.2.1 8086/8088 中央处理器的组成	(2)
1.2.2 8086/8088 CPU 的寄存器结构	(3)
1.3 8086/8088 的存储器组织	(5)
1.3.1 存储器的地址和内容	(5)
1.3.2 存储器分段	(6)
1.4 外部设备	(7)
习题	(7)

第 2 章 寻址方式与指令系统

2.1 概述	(9)
2.2 寻址方式	(9)
2.2.1 与数据有关的寻址方式	(9)
2.2.2 与转移地址有关的寻址方式	(16)
2.3 指令系统	(18)
2.3.1 数据传送指令	(18)
2.3.2 算术运算指令	(22)
2.3.3 逻辑运算指令	(30)
2.3.4 移位指令	(32)
2.3.5 处理机控制指令	(34)
习题	(35)

第 3 章 汇编语言程序格式

3.1 汇编语言的语句格式	(37)
3.1.1 指令语句的格式	(37)
3.1.2 伪指令语句的格式	(38)
3.2 伪指令语句	(39)
3.2.1 数据定义及存储器分配伪指令	(39)

3.2.2	表达式赋值伪指令·····	(40)
3.2.3	基数控制伪指令·····	(41)
3.2.4	段定义伪指令·····	(41)
3.2.5	对准伪指令·····	(42)
3.2.6	汇编语句中的标号、常量和变量·····	(43)
3.2.7	汇编语句中的表达式·····	(44)
3.2.8	程序开始和结束伪指令·····	(49)
3.3	宏指令语句·····	(49)
3.3.1	宏的定义、调用和展开·····	(49)
3.3.2	宏定义中的参数·····	(50)
3.3.3	宏定义中的 LOCAL 伪指令·····	(52)
3.4	汇编语言程序格式·····	(53)
3.5	汇编语言上机过程·····	(54)
3.5.1	宏汇编 MASM 系统的安装·····	(54)
3.5.2	宏汇编 MASM 系统的使用·····	(54)
3.5.3	汇编程序的调试·····	(59)
习题	·····	(62)

第 4 章 顺序程序设计

4.1	汇编语言程序设计的基本步骤·····	(63)
4.1.1	汇编语言程序的基本要求·····	(63)
4.1.2	汇编语言程序设计的基本步骤·····	(63)
4.2	流程图的画法·····	(65)
4.3	顺序程序设计·····	(66)
习题	·····	(68)

第 5 章 分支程序设计

5.1	转移指令·····	(69)
5.1.1	无条件转移指令·····	(69)
5.1.2	条件转移指令·····	(70)
5.2	简单分支程序设计·····	(71)
5.3	复合分支程序设计·····	(72)
5.4	多分支程序设计·····	(73)
5.4.1	地址表法多分支程序设计·····	(73)
5.4.2	逻辑分解法分支程序设计·····	(74)
习题	·····	(75)

第 6 章 循环程序设计

6.1	串操作指令·····	(76)
-----	------------	------

6.1.1	与 REP 配合的串指令 MOVS,STOS,LODS	(76)
6.1.2	与 REPE/REPZ 和 REPNE/REPNZ 相配合工作的串操作指令 CMPS,SCAS	(78)
6.2	循环控制转移指令.....	(81)
6.3	循环程序控制方法.....	(83)
6.3.1	计数控制法.....	(83)
6.3.2	条件控制法.....	(84)
6.4	多重循环程序设计.....	(85)
习题		(86)

第 7 章 子程序设计

7.1	子程序的概念与特性.....	(88)
7.2	子程序调用与返回指令.....	(89)
7.2.1	子程序调用指令 CALL	(89)
7.2.2	子程序返回指令 RET	(91)
7.3	子程序设计	(92)
7.3.1	子程序的结构.....	(92)
7.3.2	子程序的定义.....	(93)
7.3.3	子程序的参数传递方法.....	(98)
7.3.4	子程序嵌套	(105)
7.3.5	子程序的递归调用	(107)
习题		(111)

第 8 章 输入输出和中断

8.1	输入输出指令	(113)
8.1.1	输入指令	(113)
8.1.2	输出指令	(114)
8.2	中断概述	(114)
8.2.1	中断的概念	(114)
8.2.2	中断的用途	(115)
8.2.3	中断源及其分类	(115)
8.2.4	中断优先级及中断嵌套	(115)
8.3	IBM-PC 机中断系统	(116)
8.3.1	8086/8088 CPU 的中断结构	(116)
8.3.2	中断类型	(116)
8.3.3	中断矢量及中断矢量表	(118)
8.3.4	中断处理过程	(121)
8.4	中断指令及中断返回指令	(122)
8.4.1	中断指令 INT	(122)

8.4.2 中断返回指令 IRET	(123)
8.5 DOS 和 BIOS 功能调用	(123)
8.5.1 概述	(123)
8.5.2 DOS 功能调用	(124)
8.5.3 BIOS 功能调用	(129)
8.6 中断服务程序的设计	(141)
8.6.1 中断服务程序的设计方法	(141)
8.6.2 中断服务程序的设计举例	(143)
习题	(147)

第 9 章 多模块程序设计

9.1 多模块程序设计	(148)
9.1.1 概述	(148)
9.1.2 多模块程序设计技术	(150)
9.2 汇编语言与高级语言的混合编程	(158)
9.2.1 C 语言与汇编语言程序的连接	(159)
9.2.2 PASCAL 语言与汇编语言程序的连接	(167)
9.2.3 FORTRAN 语言与汇编语言程序的连接	(169)
习题	(170)

附录 1 DOS 系统功能调用表

附录 2 BIOS 调用表

参考文献



第1章

基础知识

1.1 计算机系统概述

计算机系统由硬件系统和软件系统两部分组成。硬件系统包括各种功能部件电路、外部设备和机柜等硬设备,是计算机的物质基础。软件系统是为了运行、管理和维护计算机而编制的各种程序的总和。软件建立在硬件的基础之上,可使计算机性能得到充分的发挥。

1.1.1 硬件系统

硬件系统包括中央处理器(CPU)、存储器和外部设备三个主要部分,这三者由系统总线连接在一起。其典型结构框图如图 1.1 所示。

(1)中央处理器 CPU 包括运算器和控制器两部分。运算器执行所有的算术和逻辑运算指令。控制器负责全机的控制操作,它负责从存储器中逐条取出指令,进行译码分析后向全机发出取数、执行、存数等控制命令,以完成程序所要求的功能。

(2)存储器是计算机的记忆部件,用于存储程序和数据。一般把在机器内部的存储器称为内存或者主存。

(3)外部设备一般包括 I/O 设备和大容量存储器。I/O 设备是指计算

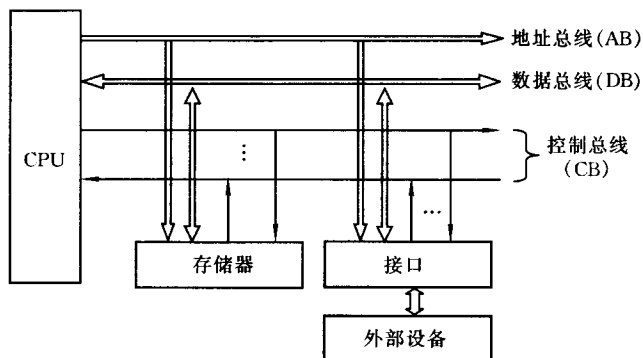


图 1.1 计算机硬件结构

机与外部世界通信的输入/输出设备,如键盘、鼠标、显示器、打印机等;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、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,外部存储器容量比内存容量大得多,但存取信息的速度比内存慢得多。内存中一般只存放正在执行的程序和正在被处理的数据,其它数据和程序都存放在外存中。程序运行时,CPU 把必要的数据和程序调入内存,程序运行结束后,处理结果及程序再从内存送往

外部存储器。

(4) 系统总线连接 CPU、存储器和外部设备,用来传送各部件之间的信息。系统总线包括数据总线(DB)、地址总线(AB)和控制总线(CB),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),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 对存储器或外部设备的控制命令和外部设备的请求及状态信号。

1.1.2 软件系统

用于一台计算机的各种程序和数据统称为软件。计算机软件通常分为两大类:系统软件和应用软件。系统软件是生成、准备和执行其他应用程序所必需的一组程序,通常由计算机厂家提供,是介于用户和硬件之间的界面,用户通过它来使用计算机。系统软件主要包括:①管理程序及操作系统;②语言处理程序,如汇编程序、各种高级语言的编译或解释程序等;③服务性程序,如诊断程序、排错程序等。应用软件是指用户用计算机解决问题时所编制的各种程序。

1.2 8086/8088 中央处理器

1.2.1 8086/8088 中央处理器的组成

Intel 8086/8088 属第三代 CPU,它们均有 20 位地址总线,可直接寻址 1MB 内存单元。8088 具有 8 位与内存或外部设备交换数据的数据总线,而 8086 具有 16 位数据总线,其他方面两种处理器完全相同,为一种 CPU 编写的软件可以不加修改地在另一种 CPU 上运行。

Intel 8086/8088 CPU 就功能而言,内部可分成两大部分:总线接口单元 BIU 和执行单元 EU。如图 1.2 所示。

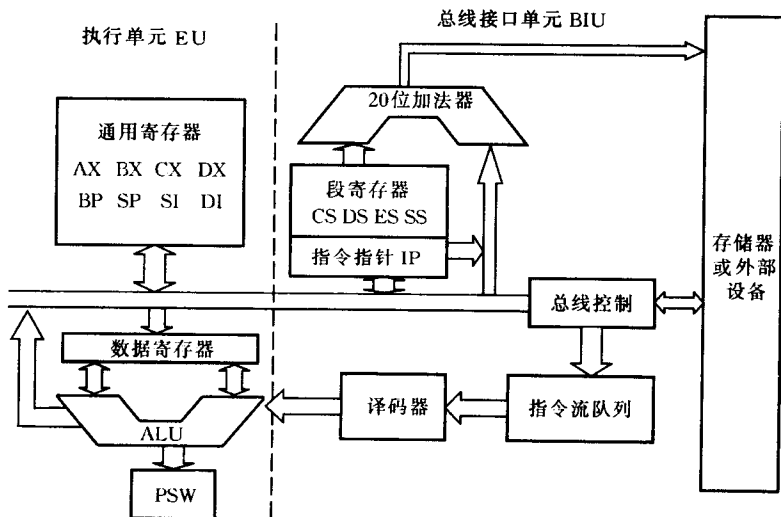


图 1.2 8086/8088 CPU 结构

BIU 负责 CPU 与存储器和外部设备之间的信息传送。具体地说, BIU 负责从内存指定单元取出指令送至指令队列排队。在指令执行时, 所需的操作数也由 BIU 从内存或外部设备的指定区域取出, 送给 EU 部分执行, 执行结果再由 BIU 送至内存或外部设备。

EU 负责指令的执行, 进行算术和逻辑运算。由于取指部分和执行部分是分开的, 所以在一条指令的执行过程中, 就可以取下一条(或多条)指令, 放在指令队列中排队。当一条指令执行完以后就可以立即执行下一条指令, 减少了 CPU 为取指令而等待的时间, 提高了 CPU 的利用率, 加快了系统的运行速度, 另一方面又降低了对存储器存取速度的要求。

1.2.2 8086/8088 CPU 的寄存器结构

8086/8088 CPU 的寄存器结构如图 1.3 所示, 包括通用寄存器、段寄存器和控制寄存器, 通用寄存器又分为数据寄存器和指针及变址寄存器。

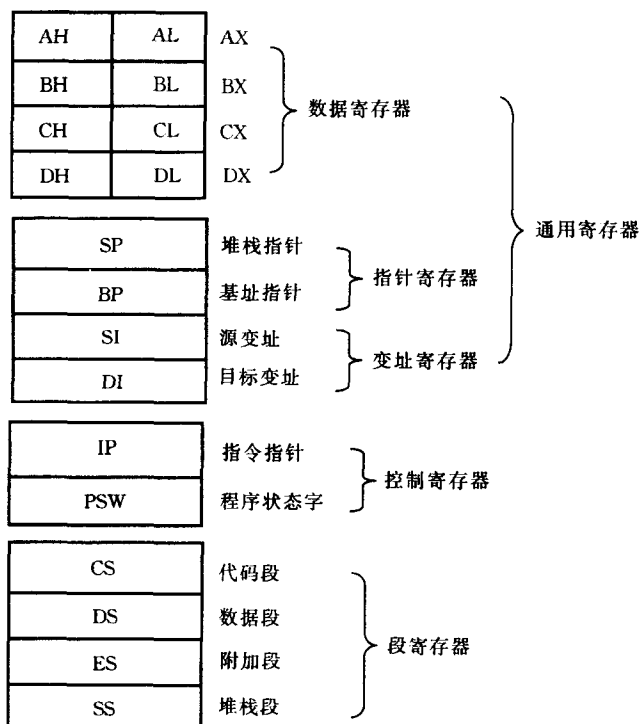


图 1.3 8086/8088 CPU 的寄存器结构

1. 数据寄存器

数据寄存器 AX, BX, CX 和 DX 是四个 16 位的字寄存器, 每个寄存器也可以作为两个 8 位字节寄存器使用, 分别为高字节和低字节, 例如 AH 和 AL 分别代表寄存器 AX 的高字节和低字节。这些寄存器用来暂存运算过程中所用的操作数、结果数据或其它信息。这四个寄存器属于通用寄存器, 但又可以用于各自专用的目的。

AX: 作为累加器使用, 在乘除法运算中用于默认存放被乘数或被除数, 在输入/输出指令中用于与外设传送数据。

BX:基址寄存器,在访问存储器时常用来存放存储器的基地址,与数据段寄存器 DS 一起形成存储器的物理地址。

CX:在循环指令(LOOP)和串指令中隐含存放循环次数,在移位指令中用 CL 存放移位次数。

DX:在乘除法运算中,乘积或被除数为 32 位时,高 16 位操作数存放在 DX 中,在输入/输出指令中,外设端口地址大于 255 时,地址应存放在 DX 中。

2. 指针及变址寄存器

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,但是只能以字(16 位)的形式使用。此外,它们更经常的用途是在段内寻址时提供存储单元的偏移地址,偏移地址也称为有效地址(EA)。

SP:堆栈指针寄存器,用于指示堆栈栈顶的偏移地址,与堆栈段寄存器 SS 一起形成栈顶的物理地址。

BP:基址指针寄存器,在访问存储器时常用来存放存储器的基地址,与堆栈段寄存器 SS 一起形成存储器的物理地址。

SI:源变址寄存器。

DI:目标变址寄存器。

在通用存储器访问指令中,SI, DI 可作偏移地址寄存器,与 DS 一起形成存储器的物理地址。在串处理指令中,SI 和 DI 作为隐含的源变址寄存器和目标变址寄存器,此时,SI 与数据段寄存器 DS 联用指示源操作数,DI 与附加段寄存器 ES 联用指示目标操作数。在执行串处理指令时,SI 和 DI 具有自动增量或自动减量的功能。

3. 段寄存器

8086/8088 CPU 在访问 1MB 的存储空间时采用分段技术,将内存分成四个逻辑段,分别为代码段、数据段、附加段和堆栈段,指向各段的寄存器分别为 CS, DS, ES, SS。各段寄存器长度为 16 位,在访问存储器时,段地址左移 4 位和偏移地址相加,形成存储单元的物理地址。

4. 控制寄存器

包括 IP 和 PSW 两个 16 位的寄存器。

IP:指令指针寄存器,在程序运行时,始终指向下一条指令的偏移地址,它与 CS 联用确定下一条指令的物理地址。IP 具有自增功能,当一条指令从存储器中取出后 IP 立即自增,指向下一条指令的地址。

例 1.1 PC 机启动或复位时,CS 自动置为 0FFFFH,IP 的值为 0000H。因此机器启动后执行的第一条指令的地址为 0FFFF0H。

PSW:程序状态字寄存器,长度为 16 位,只用了其中 9 位,包括 6 位状态标志和 3 位控制标志,如图 1.4 所示。

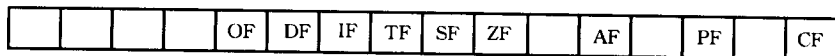


图 1.4 PSW 格式

状态标志用来记录程序运行过程中的状态信息。这些信息由处理器根据运算结果自动设

置,作为后续条件转移指令的转移控制条件,所以状态标志也称为条件码。具体有以下作用:

CF:进/借位标志,加、减法运算结果的最高位出现进位或借位则 $CF=1$,否则 $CF=0$ 。这个标志主要用于多字节或多字的加、减法运算中。在移位类指令中,每次移出位也进入 CF 标志位。

AF:辅助进位标志,加、减法运算结果的低半字节向高半字节产生进位或借位则 $AF=1$,否则 $AF=0$ 。这个标志主要用于十进制加、减法运算中。

OF:溢出标志,运算结果超出机器能表示的数值范围时称为溢出。结果溢出则 $OF=1$,否则 $OF=0$ 。溢出判断最常用的方法有双高位法和双符号法。

ZF:零标志,运算结果为零则 $ZF=1$,否则 $ZF=0$ 。

SF:符号标志,运算结果为负则 $SF=1$,运算结果为正则 $SF=0$ 。

PF:奇偶标志,运算结果中 1 的个数为偶数则 $PF=1$,否则 $PF=0$ 。

例 1.2 从下面算式可以看出,0D1H 和 0BFH 两个字节数求和结果为:90H,状态标志分别是: $CF=1, AF=1, SF=1, ZF=0, OF=0, PF=1$ 。

$$\begin{array}{r} 1101\ 0001 \\ +\ 1101\ 1111 \\ \hline 11001\ 0000 \end{array}$$

控制标志由程序根据需要进行设置,用来改变程序的运行情况,主要包括以下 3 位:

DF:方向标志,在串处理指令中用来控制处理信息的方向。当 DF 置 1 时,每次操作结束后使变址寄存器 SI 和 DI 减量,这样就使串指令从高地址向低地址方向处理;当 DF 置 0 时,每次操作结束后使 SI 和 DI 增量,这样就使串指令从低地址向高地址方向处理。

IF:中断允许标志,当 IF 置 1 时,允许外部中断请求,否则禁止中断请求。

TF:陷阱标志,用于程序的单步跟踪。当 TF 置 1 时,每条指令执行结束后,由系统控制计算机暂停;当 TF 置 0 时,CPU 正常工作,不产生陷阱。

1.3

8086/8088 的存储器组织

1.3.1 存储器的地址和内容

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,位的编号从低到高依次为 $D_0 \sim D_7$,两个字节组成一个字,位的编号从低到高依次为 $D_0 \sim D_{15}$ 。

在存储器中以字节为单位存储信息。为正确的进行信息存取,给每个字节一个编号,即存储单元的地址。地址从 0 开始,后续单元地址依次加 1。一个存储单元中存放的信息称为该单元的内容,图 1.5 表示存储器中存放信息的情况。

从图中可以看出,20000H 单元的内容为 01H,表示为:

$$(20000H) = 01H$$

但是,大部分机器都是以字为单位进行存取,一个字在存储单元中占连续两个字节,低字节存入低地址单元,高字节存入高字节单元。字单元的地址采用低地址表示。图 1.5 中

20000H 字单元的内容为 2F01H,20001H 字单元的内容为 562FH,分别表示为:

(20000H)=2F01H
(20001H)=562FH

存储器的主要特性是:它的内容是取之不尽的。也就是说,从存储单元取出信息后,该单元的内容保持不变,可以重复取出,向存储单元存入新信息后,原来内容被新内容代替。

1.3.2 存储器分段

8086/8088 CPU 有 20 条地址线,可直接访问 1MB 存储单元,地址从 00000H 到 FFFFFH。在访问存储单元时,每个存储单元都有一个惟一的 20 位地址,称为该存储单元的物理地址。然而,8086/8088 CPU 中与地址有关的寄存器如 IP,SP, BP, SI,DI 等都是 16 位,16 位的地址只能访问 64KB 的存储单元。那么,8086/8088 的 20 位地址如何形成呢?它是将整个 1MB 存储器以 64KB 为范围分成若干段,在访问一个具体物理单元时,必须由一个基本地址加上 IP,SP, BP,SI,DI 等寄存器中的 16 位偏移地址,形成 20 位物理地址。这个基本地址就是由段寄存器 CS,DS,ES 或 SS 左移 4 位自动形成的。图 1.6 表示了 20 位地址的形成过程。

当需要产生一个 20 位地址的时候,一个段寄存器会自动被选择,且能自动左移 4 位,再与一个 16 位偏移地址相加,产生所需要的 20 位物理地址。

取指令时,自动选择代码段寄存器 CS,左移 4 位,再加上由 IP 所决定的 16 位偏移地址,得到要取的指令的物理地址。

当涉及到堆栈操作时,自动选择堆栈段寄存器 SS,左移 4 位,再加上由 SP 所决定的 16 位偏移地址,得到堆栈操作所需要的 20 位物理地址。

当涉及到一个操作数时,自动选择数据段寄存器

DS 或附加段寄存器 ES,左移 4 位,再加上 16 位偏移地址,得到操作数所需要的 20 位物理地址。此时的 16 位偏移地址可以是包含在指令中的直接地址,也可以是某个 16 位地址寄存器的值,或者是指令中的偏移量加上 16 位地址寄存器的值,这取决于指令的寻址方式。

在存储器分段技术中,当段寄存器的值确定后,该段的最大寻址空间为 64K。若有一个任务,它的程序长度、堆栈长度以及数据区长度都不超过 64K,则可在程序开始时分别给 CS, DS,ES,SS 赋值,然后在程序段就不用再考虑这些段寄存器的值,程序就可以在各自的区域中正常地进行工作。若某个任务所需的程序长度、堆栈长度以及数据区总长度不超过 64K,则可在程序开始时分别给 CS,DS,ES,SS 赋相同的值,程序也能正常工作。

在这种分段技术中,若某个段超过 64K,可在同一程序中设置多个相同类型的段,然后在程序运行的过程中动态调整段寄存器的内容,就可以保证程序的正常运行。若数据段、附加段或堆栈段超过 64K 时,可以设置多个同类型的段,只要在访问存储器以前,用指令给 DS,ES

00000H	22H
00001H	1EH
00002H	0EH
	⋮
20000H	01H
20001H	2FH
20002H	56H
20003H	C2H
	⋮

图 1.5 存储单元的地址与内容

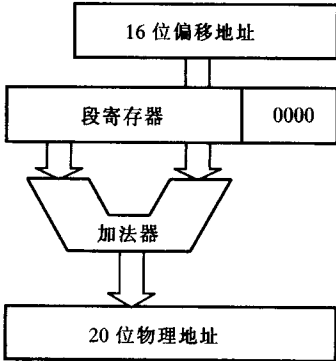


图 1.6 8086/8088 20 位物理地址的形成

或 SS 重新赋值就可以了。若代码段超过 64K, 同一个程序中可以设置多个代码段, 用段间远跳转实现段间转移。

例 1.3 假设 $(DS)=1000H$, $(ES)=2000H$, $(SS)=3000H$, $(CS)=4000H$, 则数据段的范围为 $10000H\sim1FFFFH$, 附加段的范围为 $20000H\sim2FFFFH$, 堆栈段的范围为 $30000H\sim3FFFFH$, 代码段的范围为 $40000H\sim4FFFFH$ 。

在实际使用时, 存储器地址经常用逻辑地址表示, 其格式为:

段地址: 偏移地址

例 1.4 $2000H:1100H$ 表示段地址为 $2000H$, 偏移地址为 $1100H$ 的存储单元, 其物理地址为 $21100H$ 。 $SS:0100H$ 则表示段地址在 SS 中, 偏移地址为 $0100H$ 的存储单元, 假设 $(SS)=2100H$, 其物理地址同样为 $21100H$ 。由此可见, 物理地址相同的存储单元其逻辑地址可以有不同的表示方法。

1.4 外部设备

计算机运行时的程序和数据都要通过输入设备送入计算机, 程序运行的结果要通过输出设备送给用户, 所以输入输出设备是计算机必不可少的组成部分。大容量的外存储器(如磁盘)能存储大量信息, 也是现代计算机不可缺少的一部分。

从图 1.1 可见, 外部设备与主机的通信是通过接口电路完成的。在每个接口电路中都有一组寄存器, 通常称其为端口, 这些端口按照功能分为三种类型:

数据端口: 用来存放外设与主机之间所要传送的数据, 这种端口实际上起缓冲或锁存的作用。

状态端口: 用来保存外部设备或接口电路的状态信息, 以便 CPU 在必要时测试外设或接口电路的状态, 了解它们当前的工作情况。例如, 输入设备的准备好位, 表明输入设备是否准备好数据, 等待 CPU 读取; 输出设备的忙闲位, 表示输出设备是否可以接受 CPU 发送的数据。

控制端口: CPU 给外设或接口电路的控制命令通过控制端口暂存, 以便向接口电路或外设发出各种命令。例如, 打印机的启动命令等。

为使主机访问外设方便, 外设的每个端口都给定一个编号, 叫做端口地址, 这样就组成了 I/O 地址空间。在 8086/8088 CPU 中, I/O 地址空间独立设置, 通过专门的指令访问。8086/8088 CPU 访问外设端口时, 地址线用 16 位, 因此 I/O 地址空间可达 64K, 端口地址范围为 $0000H\sim FFFFH$ 。

习 题

- 1.1 什么是计算机系统? 说明其组成。
- 1.2 8086/8088 CPU 中有哪些寄存器? 说明它们的功能及分组情况。
- 1.3 8086/8088 CPU 内部分为哪两大功能部件? 试述这两个部件的作用。
- 1.4 程序状态字 PSW 中包含几个标志位? 各位的含义及用途如何?