



21世纪高职高专规划教材·计算机系列

汇编语言 程序设计

卜艳萍 吴玉萍 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

21 世纪高职高专规划教材·计算机系列

汇编语言程序设计

卜艳萍 吴玉萍 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 IBM PC 机作为背景系统,深入浅出地介绍了微机原理的基础知识及汇编语言程序设计的方法和技术。

全书共分为 8 章,第 1 章和第 2 章介绍微机基础知识;第 3 章详细介绍了 IBM PC 机的寻址方式和指令系统;第 4 章介绍伪指令、汇编语言程序格式及设计思想、宏指令语句等;第 5 章讲述顺序程序设计、分支程序设计、循环程序设计及子程序(过程)设计的方法;第 6 章分析了程序查询传送方式和中断传送方式的基本方法;第 7 章讲述了 DOS 系统功能调用和 BIOS 系统功能调用知识;第 8 章是汇编语言程序设计实验指导。

本书可作为高职高专计算机专业及相关专业师生的教学用书,也可作为从事计算机应用的工程技术人员的技术参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

汇编语言程序设计/卜艳萍等编著. —北京:电子工业出版社,2002.2

21 世纪高职高专规划教材·计算机系列

ISBN 7-5053-7425-7

I. 汇… II. ①卜… ②吴… 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(2002)第 004521 号

责任编辑:刘宪兰 特约编辑:程清源

印 刷:北京李史山胶印厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:13.75 字数:339 千字

版 次:2002 年 2 月第 1 版 2002 年 2 月第 1 次印刷

印 数:5 000 册 定价:18.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077

前 言

《汇编语言程序设计》是高等职业技术教育计算机专业学生的必修课,是计算机专业语言类最基础的课程之一。学好这门课程,可为学习相关的专业课打下基础,它是《微机原理及接口技术》、《操作系统》、《计算机控制技术》和《单片机原理及应用》的前导课程。学习本书的读者需要有一定的微机基本知识。由于汇编语言是一门实践性很强的课程,因此本课程的开设,对于训练学生掌握指令的操作过程、程序设计的基本规则和方法以及上机调试程序的能力都有重要作用。本书是选择目前国内最广泛使用的 IBM PC 机作为基础机型而编写的。

本书共有 8 章。第 1 章为微型计算机基础知识,第 2 章为 8088 微处理器介绍,这两章为后面的讲解提供必要的微机系统软件、硬件知识。第 3 章详细介绍 IBM PC 机的寻址方式和指令系统,并给出各种指令的使用范例。第 4 章介绍伪指令、汇编语言程序格式及设计思想、宏指令语句等。第 5 章是汇编语言程序设计,包括顺序程序设计、分支程序设计、循环程序设计及子程序(过程)设计。第 6 章叙述输入/输出程序设计的基本方法,主要分析了程序查询方式和中断传送方式的程序实例。第 7 章讲述系统功能调用。包括 DOS 系统功能调用和 BIOS 系统功能调用,并通过程序设计实例分析系统功能调用的实现。第 8 章是汇编语言程序设计实验指导,在介绍了 DEBUG 和 MASM 的命令之后,提出了 8 个程序设计实验要求并进行分析。书中提供了大量的程序例题,每章后均有习题,便于学生复习。

本书的第 8 章和附录 A、附录 B 由吴玉萍编写,其余部分由卜艳萍编写。由于编者水平有限及时间仓促,书中错漏之处,敬请读者批评指正。

编 者

2001 年 8 月

目 录

第 1 章 微型计算机基础知识	(1)
1.1 微型计算机的基本结构与组成	(2)
1.1.1 计算机的基本组成	(2)
1.1.2 微机硬件系统结构	(3)
1.1.3 存储器	(4)
1.1.4 输入/输出设备	(7)
1.2 计算机中的数制与编码	(8)
1.2.1 进位计数制及数制转换	(8)
1.2.2 定点数与浮点数	(11)
1.2.3 机器数的编码表示	(12)
1.2.4 ASCII 码与 BCD 码	(14)
第 2 章 16 位微处理器	(17)
2.1 概述	(17)
2.2 8088 微处理器的功能结构	(17)
2.2.1 8088 CPU 的内部寄存器	(18)
2.2.2 8088 CPU 的两大功能结构	(21)
2.3 8088 微处理器的引脚及其功能	(23)
2.4 系统组成	(27)
第 3 章 8086/8088 指令系统	(29)
3.1 寻址方式	(29)
3.1.1 与数据有关的寻址方式	(29)
3.1.2 与转移地址有关的寻址方式	(33)
3.2 8088/8086 指令系统	(34)
3.2.1 数据传送指令	(34)
3.2.2 算术运算指令	(40)
3.2.3 逻辑运算指令	(46)
3.2.4 移位指令	(47)
3.2.5 转移指令	(49)
3.2.6 字符串操作指令	(52)
3.2.7 处理器控制指令	(55)
3.2.8 输入/输出指令	(56)
3.2.9 中断指令	(57)
第 4 章 8088/8086 汇编语言	(61)
4.1 汇编程序功能	(61)
4.2 常数	(62)

4.3	伪指令	(62)
4.3.1	表达式赋值伪指令 EQU	(63)
4.3.2	数据定义伪指令	(63)
4.3.3	LABEL 伪指令	(65)
4.3.4	段定义伪指令	(66)
4.3.5	过程定义伪指令	(69)
4.3.6	模块命名、程序开始和程序结束伪指令	(70)
4.3.7	模块通信伪指令	(71)
4.3.8	结构伪指令 STRUC	(72)
4.4	汇编语言程序格式	(73)
4.4.1	名字项	(73)
4.4.2	操作项	(74)
4.4.3	操作数项	(75)
4.4.4	注释项	(77)
4.5	宏指令语句	(77)
第 5 章	汇编语言程序设计	(83)
5.1	顺序程序设计	(84)
5.2	分支程序设计	(86)
5.3	循环程序设计	(91)
5.3.1	循环程序的基本结构	(91)
5.3.2	循环程序设计	(91)
5.3.3	多重循环程序设计	(97)
5.4	子程序(过程)设计	(100)
第 6 章	输入/输出程序设计	(108)
6.1	CPU 与外设传送数据的控制方式	(108)
6.2	程序查询传送方式程序举例	(110)
6.3	中断传送方式	(112)
6.3.1	中断的概念	(112)
6.3.2	8088/8086 中断系统	(114)
6.4	中断传送方式程序举例	(121)
第 7 章	系统功能调用	(124)
7.1	DOS 系统功能调用	(125)
7.1.1	DOS 系统功能调用概述	(125)
7.1.2	DOS 系统调用程序实例	(126)
7.2	BIOS 系统功能调用	(131)
7.2.1	BIOS 系统功能调用概述	(131)
7.2.2	BIOS 系统功能调用举例	(133)
7.3	显示程序设计	(136)
7.3.1	显示器 BIOS 中断服务	(136)
7.3.2	文本方式下的字符及字符图形的显示	(139)

7.3.3 字符图形显示举例	(141)
第 8 章 汇编语言程序设计实验指导	(150)
8.1 预备知识	(150)
8.2 上机实验指导	(154)
8.2.1 DEBUG 命令的使用实验	(154)
8.2.2 顺序程序设计实验	(161)
8.2.3 分支程序设计实验	(166)
8.2.4 循环程序设计实验	(172)
8.2.5 子程序定义及调用实验	(177)
8.2.6 DOS 功能调用实验	(183)
8.2.7 BIOS 功能调用实验	(189)
8.2.8 综合程序设计实验	(192)
附录 A 8086/8088 指令系统一览表	(197)
附录 B DEBUG 命令一览表	(202)
附录 C DOS 功能调用一览表	(203)
附录 D BIOS 功能调用一览表	(207)
参考文献	(210)

第 1 章 微型计算机基础知识

本章要点:

- 微型计算机的基本结构
- 微型计算机系统的组成
- 微型计算机系统为数制
- 微型计算机系统为数制

电子计算机的产生、发展和应用是 20 世纪重大的科技成就之一,它标志着人类文明已进入了一个新的历史阶段。电子计算机自 1946 年 2 月诞生以来,在 50 多年的时间里得到了迅速的发展。作为 20 世纪一个伟大的发明,其应用已广泛而深入地渗透到社会的各个领域,成为信息化社会的基础。

50 多年来,虽然计算机制造技术发生了很大的变化,其性能有了极大的提高,计算机取得了科学史上最惊人的发展速度,但计算机的基本组成原理仍遵循着 50 多年前美籍匈牙利数学家冯·诺依曼提出的基本计算机的工作原理。冯·诺依曼机应具有以下基本功能。

- 计算机由运算器、控制器、存储器、输入设备和输出设备 5 个部分组成。
- 用二进制形式表示数据和指令。
- 采用存储程序的方式,程序和数据可以在机器中长期记忆保存。
- 机器的工作应当在指令的控制下协调进行,指令由操作码和地址码组成。
- 机器具有数据的运算和处理功能。
- 可进行数据的输入、输出。

随着构成计算机的主要电子元件从电子管、晶体管、中小规模集成电路发展到大规模集成电路,计算机也经历了 4 个发展阶段。计算机的应用领域也由单纯的数值计算发展到科研、工业、农业、国防和社会生活的各个领域。如实时控制系统、数据库管理系统、计算机辅助设计、智能模拟与人工智能系统等。

计算机具有如下特点。

- 运算速度快:计算机具有极高的运算速度,世界上第一台计算机的运算速度为每秒 5000 次,目前普通的微型计算机每秒就可以执行上亿条指令。

- 运算精度高:计算机的运算精度随着数字运算设备的技术发展而提高,加上先进的算法,可得到很高的运算精度。例如 π 的计算,在计算机诞生前的 1500 多年的时间里,虽经人们不懈努力,也仅计算到其小数点后 500 位。而使用计算机后,目前已达到小数点后上亿位。

- 具有记忆能力:计算机的存储器具有存储、记忆大量信息的功能。不但可以存放计算的原始数据、中间数据以及最后结果,还能存放人们事先编好的各种程序,这是计算机能进行自动处理的原因之一。

- 极高的准确性:由于采用了数字化信息编码,使计算机的运算、控制及信息处理具

有极大的准确性。在计算机中，所有的数值和符号、文字、图形、图像、语音等非数值信息均采用数字化的编码形式表示，从而保证了准确性。

- 高度自动化：由于程序和数据存储在计算机中，一旦向计算机发出指令，它就能自动按规定的步骤完成指定的任务。计算机不但具有计算能力，还具有逻辑判断能力。能在程序的指引下，根据比较的结果，自动地确定下一步该做什么，表现出初步的智能化。

- 极强的通用性：计算机采用存储程序原理，程序可以是各个领域中的用户自己编写的应用程序，也可以是厂家提供的供众多用户共享的程序。丰富的软件，多样的信息，使计算机具有相当大的通用性。

1.1 微型计算机的基本结构与组成

本教材以 Intel 8086/8088 指令系统为主，介绍汇编语言程序设计的方法。

汇编语言是与机器的指令系统有关的面向机器的语言。由于各类计算机使用的微处理器（或 CPU）的指令系统是不同的，因而相应的汇编语言也就不同。在学习汇编语言之前，应对计算机的基本结构与组成有大致地了解，知道计算机是如何工作的，了解计算机执行指令的过程，然后再通过指令系统的学习去掌握汇编语言程序设计的方法。

1.1.1 计算机的基本组成

一个完整的计算机系统是由硬件系统（hardware system）和软件系统（software system）两大部分组成。

所谓硬件系统是指构成计算机系统的物理实体或物理装置。它由控制器、运算器、存储器、输入设备和输出设备等部件构成，如图 1.1 所示。

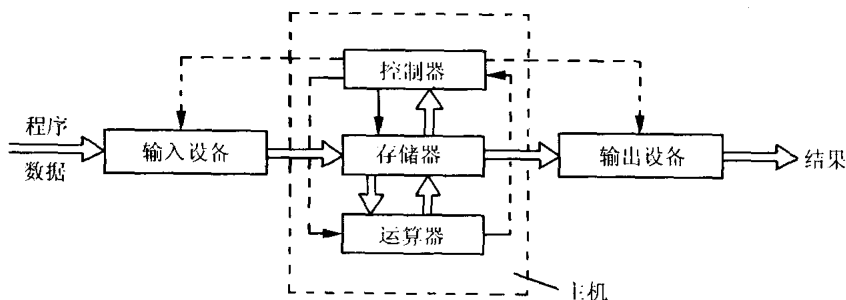


图 1.1 计算机硬件组成

运算器是对信息或数据进行处理和运算的部件，可以实现各种算术运算和逻辑运算。运算器也称算术逻辑单元 ALU（Arithmetic and Logic Unit），它由执行算术及逻辑运算的算术逻辑运算部件和用于暂存数据的寄存器两部分组成，是计算机实现高速运算的核心。它按照指令，在控制器的控制下，对信息进行算术运算、逻辑运算、移位运算等操作。

控制器（Control Unit）主要用来实现计算机本身运行过程的自动化，即实现程序的自动执行，是计算机的管理机构和指挥中心。它的功能是识别翻译指令代码，安排操作次序并向计算机各部件发出适当的控制信号，以便执行机器指令，使计算机能自动地、协调一致地工作。执行程序时，控制器首先从内存中按顺序取出一条指令，并对指令进行分析，然后根据

指令的内容向有关部件发出控制命令，控制它们执行规定的任务。这样逐一执行指令，就能使计算机按照这些指令组成的程序要求自动运行。

在微型计算机中，通常将运算器和控制器集成在一片大规模集成电路中，称之为微处理器（或中央处理器 CPU）。

存储器用来存放程序和数据，是计算机各种信息的存储和交流中心。计算机的存储器分为内存储器和外存储器两大类。内存储器主要采用半导体存储器，其中可以随机进行读和写的存储器称为随机存储器（RAM），只能读出不能写入信息的存储器称为只读存储器（ROM）。内存储器是主机的一部分，而外存储器如磁盘（硬磁盘和软磁盘）、磁带机、光盘等属于输入/输出设备。外存储器也叫辅助存储器，外存用来存储那些暂时不用的信息。外存的存储容量大，价格便宜，但存取速度相对较慢。

输入设备用来输入原始数据和处理这些数据的程序。输入的信息有数字、字母和控制符号等。输入设备是计算机从外部世界获取信息的入口，常用的输入设备有键盘、鼠标、扫描仪等。

输出设备用来输出计算机的处理结果，可以是数字、字母、表格、图形等。它是计算机向外部世界输出信息的出口。常用的输出设备有打印机、CRT 显示器和绘图仪等。输入/输出（I/O）芯片是实现计算机与输入/输出设备之间交换数据信息、控制信息、状态信息的接口。

计算机软件是指运行、维护、管理、应用计算机所需要的各种程序及其有关的文档资料。即指计算机系统所用的各种程序的集合，包括系统软件和应用软件两大类。

系统软件一般是由计算机厂家提供的，为最大限度地发挥计算机作用，充分利用计算机资源以及便于用户使用、管理和维护计算机而编制的程序的总称。它一般包括：操作系统、监控程序、输入/输出驱动程序、文本编辑程序、编译和解释程序、调试程序、连接定位程序等。

系统软件的核心称为操作系统（Operating System）。操作系统是系统软件的指挥中枢，它的主要作用是统一管理计算机的所有资源，包括处理器、存储器、输入/输出设备以及其他的系统软件和应用软件。它能为用户创造方便、有效和可靠的计算机工作环境。

语言加工软件主要是对用户使用的各种高级语言进行加工处理，使计算机最终完成用户使用各种语言所描述的任务。主要包括编辑程序、汇编程序、编译程序、解释程序和连接程序等。

用户软件是用户在自己的业务范围内为解决特定的问题而自行编制、开发的程序。例如计算程序、信息管理程序、过程控制程序等。目前，计算机的应用领域不断扩大，越来越多的计算机应用软件不断涌现。计算机应用在渗透到许多新的工业领域的同时，也使传统产业部门的面目焕然一新。

1.1.2 微机硬件系统结构

硬件系统的结构是指由各部件构成系统时的连接方式。一种典型的微型计算机硬件系统结构如图 1.2 所示。

这种硬件系统结构是单系统总线的系统结构。系统总线是进行信息传递的公共信号线，它由地址总线 AB（Address Bus）、数据总线 DB（Data Bus）和控制总线 CB（Control Bus）组成。系统中各部件均连接在系统总线上。这种系统结构简单，易于扩充，所以目前绝大多数

数微机硬件系统采用这种结构。

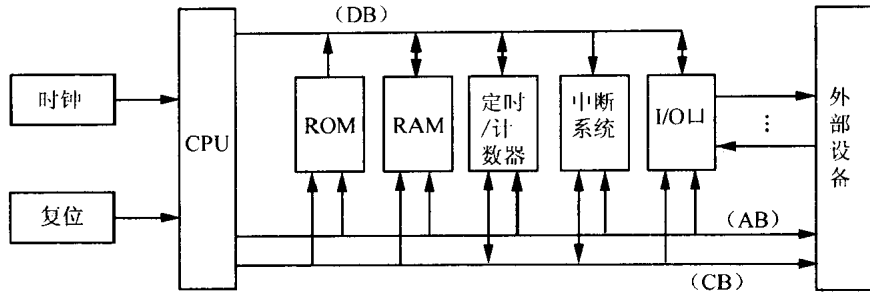


图 1.2 微型计算机硬件系统结构

系统总线把处理器、存储器、输入/输出接口和外围设备连接起来，用来传送它们之间的信息。数据总线负责传送数据，数据包括指令代码、原始数据、中间数据和结果数据；地址总线用来指出数据的来源地和目的地；控制总线负责控制总线的动作，传送处理器对存储器或输入/输出设备的控制命令和输入/输出设备对处理器的请求信号。系统总线的工作由总线控制逻辑负责指挥。

1.1.3 存储器

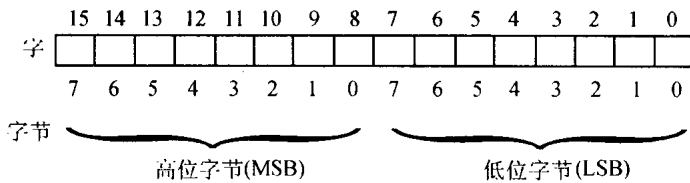
1. 存储单元的地址

在计算机系统中，正在运行的程序和数据存放在存储器中，并且所有的信息都是以二进制代码的形式表示的。那么，这些信息在存储器内是如何存储以及如何识别的呢？下面从 1 位二进制位着手讲述这个问题。

在微机系统中，1 位二进制位用 bit 来表示，8 位二进制数用 Byte（即一个字节）来表示。它们的关系为：1 Byte=8 bit。16 位二进制位或两个字节组成一个字 Word，即：

$$1 \text{ Word} = 2 \text{ Byte} = 16 \text{ bit}$$

一个字的位编号如下所示：



将两个字组成一个双字 Double Word 或 DW。微机的数据总线宽度有 8 位、16 位和 32 位等，8086/8088 CPU 的数据总线宽度为 16 位。计算机系统程序和数据等信息就是以二进制位组合的方式来表达的，当它们被存储在存储器中时，就构成了存储单元中的内容。

对于存储器的每一个存储单元给出一个惟一的编号，这个编号通常是顺序排列的，称做该存储单元的地址。存储器的读/写操作是按地址进行的，因此“地址”是非常重要的概念。存储器地址的编码范围取决于给定的二进制地址码的位数。若地址码为 16 位，地址编码的范围为 0000H~FFFFH，存储容量为 $2^{16}=64 \text{ K}$ （字节）。所以，若 n 为地址码的位数， P 为存储器的容量，则 $P=2^n \text{ Byte}$ 。

应当注意，对于每一个存储单元，有两组二进制信息与其对应：一组为内存单元的地址码；另一组为存储单元的内容。这两组二进制信息的意义是完全不同的，绝对不可混淆。而且应注意到微型机系统基本是按字节编址的，即一个存储单元的长度为一个字节，每一个存储单元有一个地址码相对应。

存储器所容纳的二进制信息总量称为存储器的容量，通常存储器的容量是用所能存储的字数乘以字长来表示：存储容量=字×数×字长。

微机中存储单元的长度大多选一个字节，即存储器按字节编址，所以存储容量常用存储器所占有的总字节数来表示，而且规定 $2^{10}=1024=1\text{ K}$ 。 $1024\text{ K}=1\text{ M}$ （1 兆）。若某存储器有 65536 个字节，则存储容量为 64 KB，简称 64 K。存储器编址范围取决于地址的位数 N ，可能的最大容量为 2^N ，地址范围为 $0\sim 2^N-1$ 。显然存储器的容量越大，存储的信息越多，其功能就越强。

一个存储单元中存放的信息称为该存储单元的内容，图 1.3 表示存储器里存放信息的情况。1000H 单元中存放的信息是 22H，也就是说 1000H 单元中的内容是 22H。表示为：(1000H)=22H。

由于机器字长是 16 位，大部分数据都是以字为单位表示的。一个字存入存储器时，占有连续的两个字节，存放时，低位字节存入低地址，高位字节存入高地址。这样两个字节单元就构成了一个字单元，字单元的地址采用它的低地址来表示。如：

$$(1000\text{H})=1122\text{H}$$

所以同一个地址既可看做字节单元的地址，又可看做字单元的地址，这要根据使用情况确定。

存储器某单元的内容被取出后，该单元仍然保持原来的内容不变，也就是说，可以重复取出。只要存入新的信息，原来保存的内容就自动丢失了。

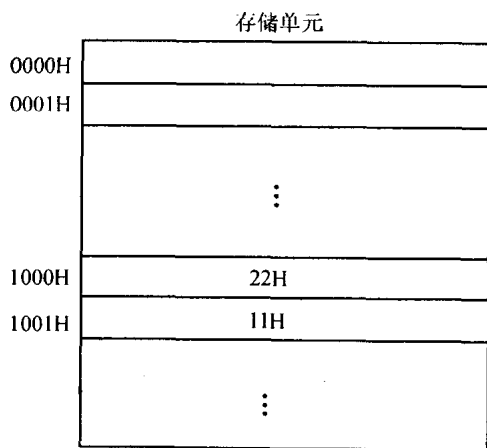


图 1.3 存储单元的地址和内容

2. 存储器地址的分段

8086/8088CPU 芯片引脚中有 20 根地址线，故具有 1M 的内存寻址能力，即 $2^{20}=1048576=1024\text{ K}=1\text{ M}$ 。存储器按字节编址，地址范围为 00000H~FFFFFH。8086/8088 微处理器可处理 16 位二进制数据，即一个字。存储器中的两个连续字节可定义为一个字，其中每个字

的低 8 位存放在低地址单元，高 8 位存放在高地址单元。低 8 位的地址代表字的地址。

由于 CPU 内部提供的存放地址的寄存器都是 16 位的，可寻址能力为 64 K 字节，因此必须设法解决寻址 1 MB 地址空间的问题。8086/8088 系统中是采用地址分段的方式解决这一问题的，即把 1 M 字节空间分为若干逻辑段，段的大小按实际需要确定，最大为 64 KB，实际应用中可能小于 64 KB，其未用的空间可以再定义为其他段的空间。

1 MB 是 16 个 64 KB 存储器的总和，但这并不意味着 1 MB 只能包括 16 个逻辑段。因为这些段既可以首尾相连，也可以相互间隔开，也可以相互重叠或者部分重叠（但需注意段的实体——被实际使用的段空间是不能重叠的），所以只要不影响程序的正常执行，逻辑段的个数可能多于 16 个也可能少于 16 个。

每个段是一个可独立寻址的逻辑单位。每个段中存储的代码或数据，可存放在段内任意单元中。每个段在物理存储器中有一个段的起始地址，称为段基地址，它的低 4 位二进制码必须是 0000，即段基地址是能被 16 整除的地址。

对于 1 MB 的物理存储空间来说，每个存储单元都有一个 20 位的物理地址。物理地址是指存储单元的实际编码，对存储器的读写操作须用物理地址进行按址存取，而采用分段方法可直接形成逻辑地址，逻辑地址表示格式如下所示。

段地址：偏移地址

其中段地址一般存放在工作寄存器中的段寄存器中，而偏移量是表示段内某一存储单元相对于段起始地址的空间位置，即段内偏移地址，也称有效地址 EA。因一个段最大存储空间是 64 K 字节，所以偏移地址也是 16 位。

有了段地址和偏移地址，就可以按下式计算出 20 位物理地址，其中 D 表示十进制数的字母。

$$\text{物理地址} = \text{段地址} \times 16D + \text{偏移地址}$$

例如，3000:1000 表示逻辑地址中段地址为 3000H，段内偏移地址为 1000H，则按上式，其对应的物理地址为：30000H+1000H=31000H。其中 30000H 是段地址 3000H 左移 4 位后得到的。

这样，物理地址的计算方法可以如图 1.4 表示：

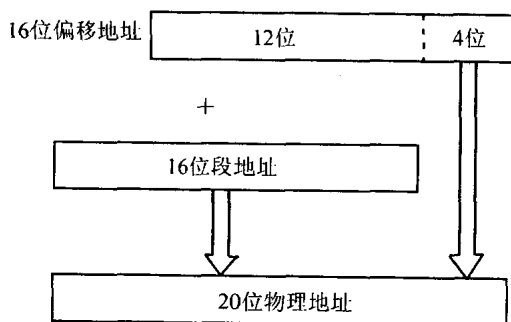


图 1.4 20 位物理地址形成示意图

8086/8088 设置了 4 个段寄存器，每个段寄存器可以确定一个段的起始地址。它们分别是 CS，SS，ES 和 DS，这 4 个段寄存器的具体定义和用途将在下章详细介绍。

8086/8088 中允许字地址从任何单元开始，当要访问的 16 位字的低 8 位字节存放在偶地址时，称字的存储是对准的，也称为对界的，这是一种规则的存放字；当要访问的 16 位字

的低 8 位字节存放在奇地址时, 则称字存储为未对准的, 又称为未对界的, 这是一种非规则的存放字。

8086 CPU 数据总线是 16 位, 每读或写一个字节的数, 就需要一个总线周期, 而访问一个字的数据, 对准的字需要一个总线周期, 未对准的字需要两个总线周期, 都由 CPU 自动完成。

对于 8088 系统, 因其数据总线是 8 位的, 故无论是字数据的读写操作, 还是字节数据的读写操作, 无论 16 位数据低 8 位是奇地址还是偶地址, 一个总线周期都只能完成一个字节的存取操作。这是因为与 8088 对应的 1M 字节空间可以是单一的存储体。在 8086 系统中, 一个存储体由奇地址存储单元 (高字节) 组成, 另一个存储体则由偶地址的存储单元 (低字节) 组成, 两个存储体的 8 位数据总线分别与 CPU 16 位数据总线的高 8 位 ($D_{15} \sim D_8$) 和低 8 位 ($D_7 \sim D_0$) 连接。在控制信号配合下, 既可同时对两个存储体进行读写操作, 也可对其中一个存储体单独进行读写操作。

1.1.4 输入/输出设备

计算机的输入/输出设备简称 I/O 设备 (Input/Output Device) 或者外设。它们位于主机之外, 用于实现计算机与外部设备或者计算机与人进行信息交换, 所以又称为外围设备 (Peripheral Device)。输入设备的任务是将人们要求计算机处理的程序、数据、文字、图形、语音等各种形式的信息转化为计算机能接受的二进制编码形式并输入主机中, 等待计算机处理; 输出设备的作用则是将经过计算机处理后的结果或者中间结果以人们所希望的形式 (例如显示、打印、绘图、印刷、照相、语言等形式) 表达。

人们是通过各种外设而有效地使用计算机的, 因而外设在功能、数量、成本等方面在计算机系统中占有越来越重要的地位。

典型的 I/O 设备分类如下:

- 常规的输入/输出设备: 键盘、鼠标、显示器、打印机、扫描仪等。
- 辅助存储器: 磁盘、磁带、光盘等。
- 数据通信设备: 计算机网络终端设备, 调制解调器等。
- 专用控制设备: 多用于计算机控制系统, 如各种开关、继电器、步进电机、数/模和模/数转换器等。

计算机与外部设备之间由 I/O 接口连接, 接口的功能如下:

- 实现数据缓冲, 使主机和外部设备的速度达到匹配。
- 实现数据格式的变换, 完成串行数据与并行数据格式之间的转换, 或者进行字节之间装配与拆卸。
- 进行 CPU 与 I/O 设备之间通信的控制。包括设备地址的选择, 操作时序的协调, 中断申请与响应, DMA 请求与响应等。还可完成电平、负载的匹配等功能。

I/O 接口可以从不同角度进行分类:

- 按接口与外设之间数据传送的形式分, 有串行接口和并行接口。
- 按主机与外设的传送信息定时关系分, 有同步控制接口和异步控制接口。
- 按信息的交换方式分, 有程序中断方式和直接存储器传送方式。

主机 (CPU 和存储器) 与外部设备的通信是通过外设接口进行的。每个接口包括一组寄存器。这些寄存器通常具有如下功能:

- 数据缓冲寄存器：用来暂时存放主机输出给外设的数据，或外设输入给主机的数据。即用来存放要放在外设和主机间传送的数据，这种寄存器实际上起缓冲器的作用。
- 状态寄存器：用来保存外部设备或接口的状态信息，以便 CPU 在必要时测试外设状态，了解外设的工作情况。
- 控制或命令寄存器：用于接收、存放主机发来的各种命令及控制信息。CPU 给外设或接口的控制命令通过此寄存器送给外部设备。

各种外部设备都有以上 3 种类型的寄存器，只是每个接口所配备的寄存器数量是根据设备的需要确定的。例如：工作方式较简单，速度又慢的键盘只有一个 8 位的数据寄存器，并把状态寄存器和命令寄存器合为一个控制/状态寄存器。又如，工作速度快、工作方式又比较复杂的磁盘则需要多个数据、状态及命令寄存器。

为使主机访问外设方便起见，外设中的每个寄存器都给予了一个端口（Port）地址，这样就组成了一个独立于内存储器的 I/O 地址空间。IBM PC 机的 I/O 地址空间可达 64KB，用 16 位二进制代码表示的地址范围是 0000H~FFFFH。

IBM PC 机提供了两种类型的系统功能调用以方便用户使用外设。一种是 BIOS（Basic Input/Output System）功能调用，另一种是 DOS（Disk Operating System）功能调用。它们都是系统编制的子程序，通过中断方式转入要执行的子程序，执行完后再返回到原来的程序继续执行。这些程序有的完成一次简单的外设信息传送，如从键盘输入一个字符，或送一个字符至显示器等；也有的完成相当复杂的一次外设操作，如从磁盘读写一个文件等。

1.2 计算机中的数制与编码

计算机所处理的信息，必须先经过信息数字化处理，即数据、文字符号、图形等各种信息都要经过编码成为计算机可以识别和处理的数字信息。因此，计算机选择哪种数字系统，如何表示数据，将直接影响机器的性能和结构。数字在机器内部是用电子器件的物理状态表示的，故在工艺条件允许情况下，应尽量选择简单的数据表示形式，以提高机器效率和通用性。第一台计算机奠定了二进制数系基础并沿用至今，就是因为二进制数只有“0”与“1”两个数码，不仅易于物理实现，数据存储、传送和处理简便可靠，而且运算规则简单，节省设备，特别是采用二进制后，能方便地使用逻辑代数这一数学工具进行逻辑电路的设计、分析、综合，并使计算机具有逻辑性。因此有必要了解二进制表示及二进制与其他常用进位计数制间的转换方法。

1.2.1 进位计数制及数制转换

人类在长期的生产劳动实践中创造了人们最熟悉的十进制，而生活中还有十二和十六进制等多种数系，其共同之处是采用进位计数制。

1. 进位计数制

进位计数制是采用位置表示法，即处于不同位置的同一个数字符号，所表示的数值不同。一般说来，如果数制只采用 R 个基本符号，则称为基 R 数制， R 称为数制的“基数”，或简称“基”（Radix）；而数制中每一固定位置对应的单位值称为“权”（Weight）。

进位计数制的编码符合“逢 R 进位”的规则，各位的权是以 R 为底的幂，一个数可按权

展开成多项式，例如，“逢十进一”的十进制数 1992.5 可写为：

$$1992.5=1\times10^3+9\times10^2+9\times10^1+2\times10^0+5\times10^{-1}$$

对 R 进制数 N ，若用 $n+m$ 个代码 D_i ($-m\leq i\leq n-1$) 表示，从 D_{n-1} 到 D_{-m} 自左至右排列，其按权展开多项式为：

$$\begin{aligned} N &= D_{n-1}R^{n-1}+D_{n-2}R^{n-2}+\cdots+D_0R^0+D_{-1}R^{-1}+\cdots+D_{-m}R^{-m} \\ &= \sum_{i=-m}^{n-1} D_iR^i \end{aligned}$$

其中： D_i 为第 i 位代码，它可取 $0\sim(R-1)$ 之间的任何数字符号； m 和 n 均为正整数， n 表示整数部分的位数， m 表示小数部分的位数； R^i 表示 D_i 位的权，它是以 R 为底的幂。

下面是计算机常用的进位计数制：

- 二进制 $R=2$ 基本符号 0,1
- 八进制 $R=8$ 基本符号 0,1,2,3,4,5,6,7
- 十进制 $R=10$ 基本符号 0,1,2,3,4,5,6,7,8,9
- 十六进制 $R=16$ 基本符号 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

其中，十六进制中的数符 A~F 字母，分别对应十进制的 10~15，例如一个十六进制数 8AE6 可写成：

$$8AE6H=8\times16^3+10\times16^2+14\times16^1+6\times16^0$$

在数 8AE6 后加 H 是为了标示此数为十六进制数。

二进制数、八进制数、十六进制数和十进制数之间的对应关系见表 1.1。

表 1.1 二、八、十六进制数和十进制数的对应关系

二进制数	八进制数	十六进制数	十进制数
0000	00	0	0
0001	01	1	1
0010	02	2	2
0011	03	3	3
0100	04	4	4
0101	05	5	5
0110	06	6	6
0111	07	7	7
1000	10	8	8
1001	11	9	9
1010	12	A	10
1011	13	B	11
1100	14	C	12
1101	15	D	13
1110	16	E	14
1111	17	F	15

由于二进制数书写长，难读难懂，为书写方便，计算机中经常使用八进制或十六进制数。人们又习惯于十进制数，而计算机内必须采用二进制数，故上面 4 种进制是经常要用到的。为识别起见，二进制数尾加 B 做标识；十进制数加 D 或省略；八进制数加 Q；十六进制数则加 H。使用 4 种进制必然存在各种进制间的相互转换问题。

2. 进位计数制间的相互转换

不同进位计数制之间的数据转换,其实质是进行基数的转换,转换原则是根据如果两个有理数相等,其整数部分与小数部分则分别相等。故整数部分和小数部分要分别进行转换。

(1) 二进制数、八进制数、十六进制数转换为十进制数

其转换规则为“按权相加”。

例 1.1: 将 10101101.101B 转换为十进制数。

$$\begin{aligned}10101101.101\text{B} &= 2^7 + 2^5 + 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-3} \\&= 128 + 32 + 8 + 4 + 1 + 0.5 + 0.125 \\&= 173.625\text{D}\end{aligned}$$

例 1.2: 将 311Q 和 2A.61H 转换为十进制数。

$$\begin{aligned}311\text{Q} &= 3 \times 8^2 + 1 \times 8^1 + 1 \times 8^0 \\&= 3 \times 64 + 8 + 1 \\&= 201\text{D}\end{aligned}$$

$$\begin{aligned}2\text{A}.61\text{H} &= 2 \times 16^1 + \text{A} \times 16^0 + 6 \times 16^{-1} + 1 \times 16^{-2} \\&= 32 + 10 + 0.375 + 0.0039 \\&= 42.3789\text{D}\end{aligned}$$

(2) 十进制数转换为二进制数、八进制数、十六进制数

整数部分转换——将十进制数连续用基数 R 去除,直到商数到零为止,每次除得的余数依次为二进制数由低到高的各位值,简称“除 R 取余”法。

例 1.3: 将 47D 转换为二进制数。

方法: 把要转换的十进制数的整数部分不断除以 2,并记下余数,直到商为 0 为止。

$$\begin{array}{ll}47 / 2 = 23 & (a_0 = 1) \\23 / 2 = 11 & (a_1 = 1) \\11 / 2 = 5 & (a_2 = 1) \\5 / 2 = 2 & (a_3 = 1) \\2 / 2 = 1 & (a_4 = 0) \\1 / 2 = 0 & (a_5 = 1)\end{array}$$

所以 $47\text{D} = 101111\text{B}$ 。

例 1.4: 将十进制数 36521 转换成十六进制数。

方法: 把要转换的十进制数的整数部分不断除以 16,并记下余数,直到商为 0 为止。

$$\begin{array}{ll}36521 / 16 = 2282 & (a_0 = 9) \\2282 / 16 = 142 & (a_1 = 10) \\142 / 16 = 8 & (a_2 = 14) \\8 / 16 = 0 & (a_3 = 8)\end{array}$$

所以 $48956\text{D} = 8\text{EA}9\text{H}$ 。

(3) 二进制数与八进制数、十六进制数之间的转换

由于八进制数和十六进制数是从二进制数演变而来的,由 3 位二进制数组成 1 位八进制数,4 位二进制数组成 1 位十六进制数。即八进制数、十六进制数与二进制数有内在联系: $2^3 = 8$, $2^4 = 16$ 。因此,八进制数、十六进制数与二进制数之间的转换很方便。