微型计算机应用技术知识

8086 / 8088
汇编语言程序设计

陈泽文　编著

北京出版社

微型计算机应用技术知识

8086/8088汇编语言程序设计

陈 泽 文 编著

北 京 出 版 社

微型计算机应用技术知识

8086/8088 汇编语言程序设计

8086/8088 Huibian Yuyan Chengxu Sheji

陈泽文 编著

北京出版社出版

（北京北三环中路6号）

新华书店北京发行所发行

昌平县马池口印刷厂印刷

*

787×1092毫米 32开本 10.75印张 228,000字

1987年12月第1版 1987年12月第1次印刷

印数 1—5,000

ISBN 7-200-00272-0/TP·5

书号：15071·92 定价：3.00元

《微型计算机应用技术知识》丛书编委会

编辑说明

随着新技术革命的到来，微型计算机的应用已逐步深入到国民经济的各个领域。不仅从事计算机硬件、软件的设计和生产人员需要有关的专业知识，而且广大的计算机使用者——工程技术人员和管理干部，也迫切需要了解微型计算机的基本知识和应用技术。为了帮助这些使用者较快地掌握这方面的知识，我们编辑了这套《微型计算机应用技术知识》丛书，由北京出版社出版。

这套丛书主要包括：微型计算机系统、接口技术、操作系统、汇编程序设计、几种常用的高级语言程序设计、实用数值方法、数据库、汉字系统、计算机图形学及其应用、局部网络以及计算机辅助设计等。在内容选材方面，努力体现科学性、实用性、普及性和先进性等特点；在编写上，力求深入浅出，通俗易懂。这套丛书具有较完整的系统性，其中的每一单册又具有相对独立性。它不仅适合具有中专或大专文化水平的广大工程技术人员和管理干部自学参考，而且可作为中专或大专院校一般专业以及各种微型计算机培训班的教材。

由于水平所限，编写工作中容有舛错不妥之处，请广大读者批评指正。

《微型计算机应用技术知识》编委会

前　言

汇编语言是一种使用最广泛的初级语言。它是用符号来表示机器代码的，是计算机机器语言的符号形式。由于它直接与计算机的硬件指令相对应，所以，使用汇编语言编写的程序，执行速度快，效率高，而且能够解决有些高级语言不能解决的问题。计算机的系统程序，大多数都使用汇编语言来编写，因而学习和掌握汇编程序设计的方法是非常有用的。

目前，IBM 公司的个人计算机 IBM PC，PC/XT 以及各种兼容机，在国内拥有不少用户，是较为普及的一类微型计算机。1984 年初，我们开始使用 IBM PC 的汇编语言为美国某软件公司编写电子游戏程序。为了帮助广大微型计算机用户学习和掌握汇编语言程序设计的方法，作者于 1984 年底，通过近一年来的工作实践，编写成《8086/8088 汇编语言程序设计》一书的初稿，曾两次被用作汇编语言程序设计课的教材。经过授课实践，对原稿进行了修改和充实。全书分七章，介绍了汇编语言程序设计的概念和过程，以及 8086/8088 的指令系统和 IBM PC 宏汇编伪指令、汇编方式等内容。为了便于初学者能很快入门，书中还列举了大量的实例，这些例子全部使用 8086/8088 的汇编语言指令，并且大多数都在 IBM PC 机上运行过。本书可供广大读者自学，

了解和掌握IBM PC汇编语言，也可做为工作中的参考手册。

十分荣幸的是，清华大学计算机系朱家维教授对本书内容的修改提出了指导性意见，并审校了全书。同时，复旦大学计算机科学系施伯乐教授、上海应用数学咨询开发中心刘鼎元教授对本书编写工作给予了帮助，我衷心地表示感谢。

复旦大学　陈泽文

1985年12月于上海

目　录

第一章 概 述

1-1 汇编语言介绍

当人们相互间进行通话时，如果双方使用同样的语言，则不必经过翻译而能直接通话；倘若在语言不同的二人之间通话，只要其中一人懂得两种语言，能将另一人的语言翻译成自己的语言，通话也能顺利进行。就象人们相互间通话一样，通过已确定的语言，人也可以与计算机通话。但由于计算机是一种电子设备，它只能理解电信号，而这些电信号就是计算机本身语言的基础。因此，人要与计算机通话，就得学会这些电子语言。使用这些语言来实现人与计算机的通话，需要书写程序。实际上，早期的计算机使用的就是这种方法。那时，程序都是一系列的数码，分别对应计算机的内部信号，因而被人们称之为机器码的程序。

与人类的语言具有一定的规则相类似，计算机语言也具有一定的规则，只是计算机语言的词汇要比人类语言的词汇少得多。计算机的语言基本上可以分为三种：机器语言、汇编语言和高级语言。

用机器语言来编写程序，需要记忆大量的机器指令编码，计算并记住许多指令的地址。由于用机器码编写的程序都是一些数字，程序中包含的错误非常难于发现和定位。因

此，用机器语言编写程序所花费的时间很多，代价很高，而且很容易出现差错，效率非常低。例如，用机器语言输入一条加法指令，可能要打进二进制码10000011，或者十六进制等效码83H；同样，要输入减法指令，要打进二进制码00101101，或者十六进制等效码2DH，这样不仅花费时间，而且也容易出错。

随着时间的推移，以及更加复杂的问题需要计算机去解决(要求编制更大的程序)，而产生了新的思想，那就是用易于记忆的缩写符号来代替数字码。这些缩写符号类似于英语缩写，有明确的意义。例如，加法指令可以用ADD来表示，而不写成10000011；减法指令用SUB来表示，而不写成00101101。这种用来表示计算机指令的符号，通常被称为助记忆符。除此之外，还可以使用标号、变量名等，而不再考虑具体的存储器地址、程序转移的长度及位移量。

用汇编语言写程序比起用机器语言写程序简单得多，也容易理解。但是，为了让计算机懂得汇编语言，能够执行用汇编语言编写的程序，就需要有一个翻译，把用汇编语言写的程序改写成用机器语言写的程序，这个翻译就是汇编程序。通常，我们把用汇编语言写的程序称为源程序；把执行翻译任务的程序叫做汇编程序；把翻译后产生的用机器语言写的程序称为目标程序，有时也称为目标模块；把汇编程序进行翻译的过程称为汇编。当然，汇编程序的工作不仅仅是产生一个目标程序，它还把源程序和目标程序一一对应起来，并给每一条目标程序的指令分配一个正确的地址，将这结果通过列表文件打印出来。这种可以打印出来的表格式的文件，称为列表文件，也称为程序清单。它既包含源程序，也包含

存储地址和目标码，可以通过它了解符号指令及其对应的机器码，也可以将清单装订成册，作为档案保留。

为了便于比较用机器语言和汇编语言所写的程序，现列举一简单的例子，即两个数 1000H 和 2000H 相加，其结果不大于 FFFFH，则再加上 2000H，直至大于 FFFFH 为止。条件满足后，寄存器 CX 中记下了相加的次数。

地址	机器码	指令的含义
08FD:0100	B90000	0 送到寄存器 CX
08FD:0103	B80010	1000 H 送到寄存器 AX
08FD:0106	BB0020	2000H 送到寄存器 BX
08FD:0109	01D8	AX 和 BX 相加，结果送 AX
08FD:010B	73FC	结果大于 FFFFH 吗？不大于再去相加
08FD:010D	EBFE	结果大于 FFFFH，结束

可以看出，要看懂这种用机器语言写的程序有多困难，又是多么乏味。如果用与它相对应的汇编语言程序，则容易被接受得多。

程序设计的发展很快，现在已经有许多程序设计语言，如 BASIC、FORTRAN、COBOL 等等。这些语言称为高级语言。高级语言不同于汇编语言，它的特点之一是一条高级语言的语句通常都产生许多条机器码指令，而汇编语言的一条语句则只产生一条机器码指令。这是汇编语言的一个缺点，但从某个角度来说，它又是汇编语言的优点。因为汇编语言更加接近机器语言，因而能更有效地发挥计算机的作用。用汇编语言编写程序，可以做到不生成没有用的多余的机器码，从而既可节省计算机的存储空间，又能提高程序执

行的速度。因此，在计算机资源和程序运行速度有限制的场合，使用汇编语言更能显示出它的优越性。特别是在某些场合，比如涉及到模数/数模转换控制的场合，高级语言已无能为力，必须使用汇编语言来编写程序。

学习汇编语言是学习其他高级语言的基础之一。汇编语言能够加深对计算机工作过程的认识，加深对其他高级语言的了解。实践证明，学习过汇编语言，能更好地掌握和使用高级语言。特别值得指出的是，汇编语言在程序分析和查错中有着非常重要的作用。

1-2 8086/8088 系列

根据不同的观点，8088 可以说是 8 位处理器，也可以说是 16 位处理器。用标准的工业术语来说，8088 应该属于 8 位微处理器，因为在它的数据总线上一次只传送 8 位信息。不过，8088 内部总线是 16 位的，它可以同样方便地进行 8 位或 16 位数据的操作。因此，也可称之为 16 位微处理器。8088 和 8086 具有完全相同的指令系统，它们的软件是完全兼容的。在 8086 上运行的所有程序可以不加任何修改地在 8088 上运行，反之亦然。Intel 公司的文献通常将 8086 和 8088 称为 iAPX 8086 和 iAPX 8088。

地址	机器码	汇编语言语句
08FD:0100	B90000	MOV CX, 0000
08FD:0103	B80010	MOV AX, 1000
08FD:0106	BB0020	MOV BX, 2000
08FD:0109	01D8	REPEAT:ADD AX, BX
08FD:010B	73FC	JNC REPEAT

08FD:010D EBFE HALT:JMP HALT

显然，对于汇编语言写的程序，只要了解 MOV，CX，AX，BX，REPEAT 等符号的含义，这个程序也就明白了。这种使用汇编语言的规定符号来编写程序，就是汇编语言程序设计。

8088 和 8086 一样，有十二个数据和地址寄存器，一个 16 位的指令指针(类似于大多数微处理器的程序计数器)和一个 16 位的标志寄存器。十二个数据和地址寄存器分成三组，每组四个，分别称为数据寄存器、指针和变址寄存器、段寄存器。

1．编址

因为指令指针及所有的数据和地址寄存器都是 16 位的，故可能被认为 8088 的编址能力不超过 8 位微处理器的编址范围，即：不超过 64K 字节(65536 个字节)。实际上，由于 8088 采用了一种新的编址方法，将段地址寄存器的内容乘上 16，再加上偏移地址，所以产生一个 20 位的地址值，即：

实际地址＝偏移地址＋(16×段地址寄存器的内容)

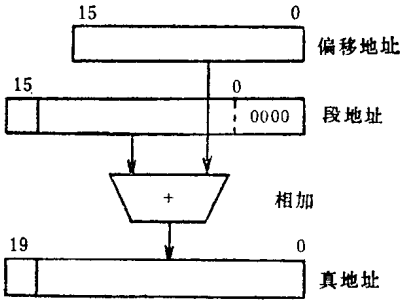

图 1-1 真地址产生示意图

因为每将一个二进制数向左移一位相当于乘上2，2×2×2×2=16，所以将段寄存器的内容左移四位就相当于乘16。因此实际操作时，8088并不是真的将段地址寄存器的内容乘16，而是把段地址寄存器的内容看作后面还跟四个0来使用(见图1-1)。例如，偏移地址值为10H(后缀H表示这是个十六进制数)，段地址寄存器的内容为2000H，真地址计算如下(操作数用二进制表示)：

		0000	0000	0001	0000	偏移地址
+	0010	0000	0000	0000	0000	段地址
	0010	0000	0000	0001	0000	真地址

因此，真正的地址有20位，20010H。

由于8088的地址可以有20位，存储容量可达1M字节(1，048，576个字节)，也就是1024K字节，所以它的编址范围是8080的十六倍。

2．段地址和偏移地址

由于8088是通过两个寄存器(段地址和偏移地址寄存器)来形成存储器地址的，所以它的程序操作就与其他处理器有些不同。一般处理器，只用一个数来给出一个存储单元的地址，而8088却要有两个数，即一个段地址和一个偏移地址。不过，通常只需要给出偏移地址即可，段地址由8088自己提供。

3．软件特点

8088软件具有不少的特色，特别是和早期8位微处理器设计程序相比，8088的软件更能显示出它的特点。在算

术运算中，8088 允许使用带符号或无符号的 8 位或 16 位二进制数，也允许使用压缩(每个字节表示二位)或非压缩（一个字节代表一位）的十进制码。8088 还可以对长达 64 K 字节的串进行以字节或字为单位的操作。

8088 指令系统有 92 种基本指令，提供 7 种不同的编址方式。通过 92 种基本指令和 7 种编址方式(以及可以使用各种各样的操作数)的组合，再加上不同的数据形式，8088 可以执行上千种指令。

8088 主振频率是 4.77 MHz，微处理器的时钟每秒是 4770000 周，每个时钟周期约为 210 ns (毫微秒)。最快的指令，如将一个寄存器的内容写到另一个寄存器，其执行时间为 2 个时钟周期，即 420 ns。最慢的指令，带符号的 16 位数用 16 位数来除，其执行时间多达 206 个时钟周期，大约为 43 μs。

8086 和 8088 的指令系统是一样的，它们的软件相互兼容，只是时钟频率有所不同，外部数据的传送方法也有 8 位和 16 位的区别。

4．输入/输出口

8088 可以编址 64 K 个输入/输出口，口 0 到口 255 是可以使用输入输出指令直接编址的口。使用其他外部口时，可以先将口的识别号置于数据寄存器 DX，然后用输入输出指令来访问。与存储器中的数据一样，外部口的数据可以是 8 位的，也可以是 16 位的。

5．存储器分配

8088 本身使用最高和最低的一些单元做特殊用处，其他 1 M 单元的大多数可供系统及用户使用。存储器的最高

16 个字节，存放一个或多个系统重置指令。在接通电源时，8088 自动执行这些指令。存储器最低的 1024 个字节，用来存放中断向量。这些中断向量，是在产生中断时 8088 转向中断服务程序的入口地址。用户可以用指令修改这些中断向量，从而使用用户自己提供的中断服务程序。

6．中断

在计算机系统中，外围设备的中断要求处理器停止正在执行的程序去为它服务。这种服务，就是执行特定的中断处理程序。由于使用中断系统，大大地提高了计算机系统的效率。如果没有中断系统，处理器将不得不周期性地去逐个询问每个外围设备，以便知道是否有哪一个请求服务。

8088 可以处理两种不同类型的中断：一种是可以屏蔽(不响应)的，一种是一旦有中断必须立即接受的。中断可以由外部设备产生，比如磁盘和其他外围设备。也可以从内部产生，由中断发生指令或(在某种条件下)由 8088 本身产生。

7．中断类型

8088 可以识别 256 种不同的中断，每种中断都有一个唯一的为 8088 所能识别的中断码，这个中断码（一个 0 到 255 之间的数）指向中断向量表中的一个单元，该单元的内容就是处理该中断的服务程序的入口地址。在这 256 种可能的中断中，有五种分配给内部中断。它们是：

① 0 号中断：除法错。如果一条除法指令产生的商太大，其结果使寄存器容纳不下，或者，用 0 来做除数，就产生 0 号中断。

② 1 号中断：单步操作。在使用 8088 的查错方式的单步指令时产生。