汇编语言基础

张国良　编译

国防工业出版社

汇编语言基础

张国良 编译

国防工业出版社

内 容 简 介

本书是一本介绍 IBM360、370 系统（OS/VS 及 DOS/VS）汇编语言的教学参考书。全书共分 16 章。本书在讲解汇编语言的基本概念的前提下，通过一些实用程序的例子，系统地介绍了使用汇编语言编写程序的各种技术。本书取材精炼，深入浅出，详尽地阐明了汇编语言中最基本的内容。每章后附有习题。

本书不仅适用于从事小型事务处理方面工作的程序员学习，而且可作为大专院校有关专业的师生参考书。对想了解 IBM 370 系统的读者也有一定的参考价值。

汇编语言基础

张国良 编译

*

国防工业出版社出版

新华书店北京发行所发行　各地新华书店经售

北京昌平兴华印刷厂印装

*

787×1092 1/16　印张 30 1/2　708千字

1987年11月第一版　1987年11月第一次印刷　印数：0,001—4,920册

ISBN 7—118—00187—2/TP18　定价：6.20元

前　言

本书向读者介绍如何使用汇编语言来编写计算机程序的基本原理。众所周知，和其他几种高级语言相比，汇编语言和计算机的结构及机器语言的关系甚为密切。正因为这一点，汇编语言才显示出和其他高级语言所不同的特点。尽管各种不同的计算机系统的汇编语言也有所不同，但由于基本原理大同小异，所以在介绍汇编语言时总是以某一具体系统为背景来讲解。本书就是以 IBM360、370 系统的 OS/VS 及 DOS/VS 操作系统为背景介绍汇编语言的。IBM370 系统是世界市场上用户最多的计算机系统之一，因此，以它为背景是不失其一般性和通用性的。读者只要掌握了本书的内容，对其他计算机系统的汇编语言也就不难掌握了。经验证明，要想成为一个优秀的程序设计者，只掌握一些高级语言（如 FORTRAN、COBOL 等）是不够的，还必须懂得汇编语言，这对更好理解其他各种高级语言是很有帮助的。

本书共有十六章，可分为六部分。

第一部分包括第一、二、三章，介绍汇编语言的基本概念。

第二部分包括第四、五、六章，主要介绍十进制指令系统，讨论了十进制整数、小数的算术运算及简单的编辑操作。

第三部分包括第七、八、九章，主要介绍标准指令系统，讨论了二进数制、十六进数制以及二进制整数、非整数的运算操作。

第四部分包括第十、十一章，介绍了复杂的编辑功能以及程序的转移和循环控制。

第五部分包括第十二、十三、十四章，介绍操作系统的概念，讨论了子程序及其连接的标准约定，还讨论了虚存储器概念。最后介绍了汇编语言程序的调试方法。

第六部分包括第十五、十六章。第十五章简单地介绍科学计算中常用的浮点操作指令。第十六章讨论了大型系统中磁带和磁盘文件的使用。

本书前十四章译自 RINA·YARMISH 等编著的《Assembly Language Fundamentals 360/370 OS/VS DOS/VS》一书。第十五、十六章是根据下列几本书编译的：Shan S·Kuo 的《Assembler Language for FORTRAN, COBOL and PL/1 Programmer》；Ned Chapin 的《360/370 Programming in Assembly Language》以及 Naney Stern 等的《370/360 Assembler Language programming》。

在本书的编译中得到许多同志的热情帮助和鼓励，在此谨表示衷心的感谢！由于译者水平不高，难免有错误和不妥之处，望读者给予指正。

目　录

第一章　计算机简介

本章简单地介绍一下计算机的性能、结构以及同它通信的方法。同时，还将讨论有助于理解本书所需要的一些基本术语。

本章扼要地复习需要几学期才能讲授完的，介绍数据处理课程的某些内容。虽然本章所介绍的内容，对理解本书来说是足够的，但对想更深入地研究某些具体范畴的读者来说，却需要参看已有的许多更好的入门教程。这里所介绍的都是一些实质性的必需的内容，可作为读者阅读过程中的参考。

1.1　什么是计算机？

在生活中，无处不谈到和用到“计算机”。在银行，出纳员使用计算机存款和取款；在实验室，用计算机处理实验结果；军事上使用计算机灵活地把导弹引向目标；计算机能控制很多工厂的操作，甚至用于交通指挥灯的管制等等。这些只是现代计算机的许多应用中的一些个别事例。

究竟什么是计算机呢？概括地说，它是一种特殊类型的“数据处理器”。它是把进来的数据（输入），在事先给它的一组指令（被存储的程序）的控制下转换为输出信息(输出）的一种电子机器。

如果机器完全是电子的，它就没有传动部件。这就意味着，数据的任何传送和处理，从它进入计算机（输入），直到它离开（输出），都是在计算机内以极大速度运动的电信号（实际上就是以极大的速度通过导线传输的电子）来进行的。没有传动的部件，也说明这种机器所需的维修工作量很小。

但是，使计算机具有如此功能的真正原因，是它能按照一组指令（程序）的控制来进行工作。我们可以用改变程序的办法，使计算机也跟着改变它正在进行的工作。因为能够编写的程序的数目可以无限多，所以，计算机也就能完成无限多种作业。计算机本身并不能指挥自己；“思维”必须由人来进行。计算机没有“智力”，即没有思维能力；它之所以重要，原因在于它是一种精确、快速、操作可控的高可靠通用工具。

如果你读了这本书，你就有可能学会如何通过编写程序来指挥计算机的工作。

1.2　计算机系统

一个典型的计算机系统包括有六个基本部分。有时可以将前三个（图 1.1）安装在同一个物理部件中，它们是：

（1）存储器（有时称为主存储器，内存储器或基本存储器）；

（2）算术和逻辑部件，通常简写为运算器（ALU）；

（3）控制器。

这三个部分通常总称为中央处理机(简写为CPU)。(按照美国国家标准学会(ANSI)的建议，CPU 只包括 ALU 及控制器，而存储器则作为一个独立的部件)。数据的实际

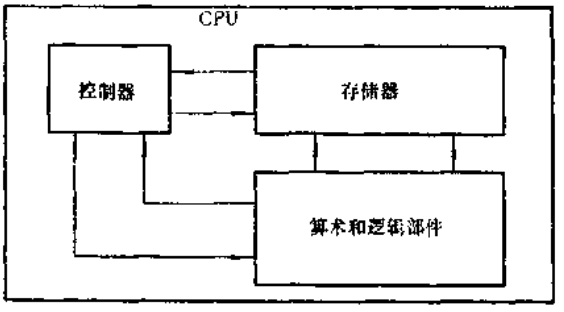


图1.1 中央处理机（CPU）。图中箭头表示信息通道

处理是在CPU中进行的。在本书中，术语“CPU”和“计算机”是可互换的。

另外的三个部分是：

- 输入设备
- 输出设备

→ 经常用I/O表示输入/输出

- 辅助存储器

上述六个部分构成了计算机系统。现在简要地研究一下每一部分。

1.2.1 存储器

正像人们在按照某些信息行动之前，必须先把这些信息存放在大脑中一样，计算机在操作之前，也必须事先将数据，以及用来告诉计算机做什么的指令存放在存储器中。这些指令构成一个存储程序。物理上，一些计算机的存储器是由磁心组成的。这些磁心以阵列方式穿在导线上。因此，有时把计算机的存储器和磁心相提并论。通过这些导线的电流可以有选择地按两个方向（顺时针或逆时针）之一，磁化单个磁心。

实际上，计算机工艺目前正趋向用更快的固态线路及单片线路构成存储器。激光技术也开始用到许多现代计算机的设计和结构中。但是，由于磁心存储器仍作为主存储器来使用，所以仍然保留着这个术语。

二进制数

电灯开关可以处于打开（连通）或关闭（断开）状态；电子线路可以导通或断路；插头也可以连通或断开。在这些情况中，都存在两种可能的状态。技术上可以把这两种状态分别看作为“是”或“否”，或者更一般讲，“1”或“0”。当数据被存储在计算机存储器中时，它是用一列磁心的状态来表示的。用磁心的两个磁化方向，来表示数字0和1。如图1.2所示。因此，一个磁心可看作是一个二进制位（二进制数字）。正像莫尔斯（Morse）电码中每个字母符号都用唯一的点划线序列组成一样，计算机中的符号由唯一的二进制序列组成。不过在莫尔斯编码中，各符号的长度有

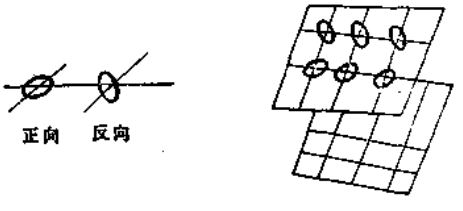


图 1.2

所不同；而计算机的存储器是被分成相等长度的段，称之为字节。每个字节可放一个符号〔诸如字母、数字、专用符号（如“&”）等等〕。在IBM计算机上，每个字节长度为8位。

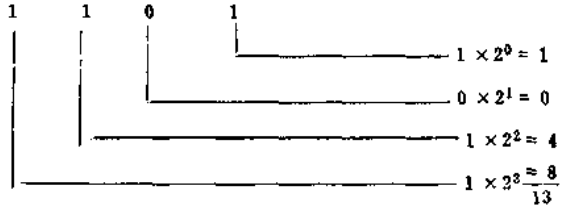
计算机中采用二进制来表示数值，即以2的幂次值作为权，由数字0和1组成的序列来表示一个数值。

二进制中各位的权为：

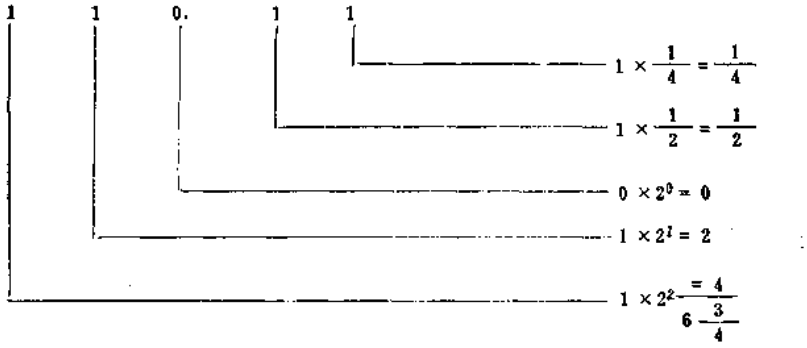
$$\cdots \underline{2^3=8} \quad \underline{2^2=4} \quad \underline{2^1=2} \quad 2^0=1 \quad \underline{2^{-1}=1/2} \quad \underline{2^{-2}=1/4} \quad \underline{2^{-3}=1/8} \cdots$$

二进制数列中的数字“1”是对该列值本身有“价值”的量。数字“0”没有对该列的值作出贡献，只是占据一个位置。

例如，二进制数1101等于十进制13，对应关系如下所示：



类似地，二进制数110.11等于十进制数6 $\frac{3}{4}$，对应关系如下所示：



十六进制表示法

用二进制数进行计算的一个主要缺点是，数目太长和阅读不便。为此，经常采用十六进制，作为表示0和1数字串的一种简洁的表示法。

十六进制表示法，以一个等值的十六进制数字代替4个二进制数字，其对应关系如下：

十六进制	二进制	十六进制	二进制
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

因此，

二进制数　1101 0101 1010 1111 可表示为

十六进制　B　5　A　F

在第七章中还要详细讨论二进制和十六进制的问题。

注意，用 4 位所能表示的最大二进制数是 1111 或 15(即 2^4-1)；8 位所能表示的最大二进制数是 11111111 或 255 (即 2^8-1)。一般讲，n 位所能表示的最大二进制数是 2^n-1。因此，能够用 n 位表示的二进制数的个数等于 n 位的最大二进制数加 1 (包括 0)，即 $(2^n-1)+1$ 或 2^n个。

EBCDIC 编码

用 8 位 ($2^8=256$) 能够表示 256 个 0 和 1 的不同组合。计算机厂家已把许多符号同这些组合一一对应起来。本书中所使用的是一种特殊组合，即 EBCDIC 码 (扩充的二进制编码的十进制交换代码)。通常，将 8 位的字节分为两部分，各位的次序是从左到右，从 0 开始，以 7 结束。现图示如下：

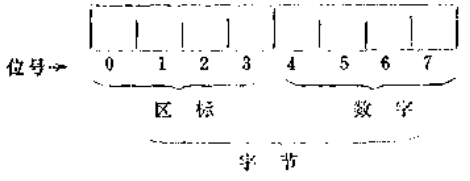


图1.3给出了数字和英文字母的EB CDIC 码

符 号	EBCDIC	编 码
A	1100	0001
B	1100	0010
C	1100	0011
D	1100	0100
E	1100	0101
F	1100	0110
G	1100	0111
H	1100	1000
I	1100	1001
J	1101	0001
K	1101	0010
L	1101	0011
M	1101	0100
N	1101	0101
O	1101	0110
P	1101	0111
Q	1101	1000
R	1101	1001
S	1110	0010
T	1110	0011
U	1110	0100
V	1110	0101
W	1110	0110
X	1110	0111
Y	1110	1000
Z	1110	1001

字母A～I的区标总是1100，而数字部分由4位二进制数1～9组成

字母J～R的区标总是1101，而数字部分由4位二进制数1～9组成

字母S～Z的区标总是1110，而数字部分由4位二进制数2～9组成

十进制数字0～9的区标总是1111，而数字部分由4位二进制数0～9组成

0	1111	0000
1	1111	0001
2	1111	0010
3	1111	0011
4	1111	0100
5	1111	0101
6	1111	0110
7	1111	0111
8	1111	1000
9	1111	1001

图1.3 字母和数字的EBCDIC表示法

例如，单词“RAIN”的EBCDIC码存储时，需要4个连续的字节，即：

1101 1001	1100 0001	1100 1001	1101 0101
R	A	I	N

一个典型的计算机存储器可以存储成千上万个EBCDIC字符。

存储器的寻址

邮局中的信箱号是唯一且连续的编号。类似地，计算机厂家把主存储器中的各字节位置，同程序员用来识别某一特殊字节或存储单元的编号一一对应起来，这个编号称为地址。

因此，我们可以在概念上把计算机存储器看作由许多编了号的小盒子组成的一个大箱子，而每个小盒子的大小等于一个字节。

000	001	002	003	004	005	006	007	008	009
		R	A	I	N				
010	011	012	013						
020	021	022	023						

上图中，单词“RAIN”在存储器中，从002地址开始共占用了4个字节。

IBM公司习惯于以k为单位来表示存储器容量的大小，每个k等于1024个字节（$1024=2^{10}$）。一个典型的计算机存储器容量有128k个字节。如果需要，可以从卖主那里购买附加存储器。磁心存储器通常以4 k为单位出售。

1.2.2 算术及逻辑部件

算术及逻辑部件是由电子线路组成的。这些电子线路的功能是执行诸如加、减、乘、除等操作。计算机把除法看成是减法的扩展。ALU还可以比较两个值，以确定一个值是否大于、等于或小于另一个值，并且它还能判定逻辑状态是“真”还是“假”。

为了便于操作，还可以利用几个32位的寄存器作为存储器。这些寄存器在操作期

间用来暂时存储少量的信息，如计算中产生的一些中间结果。

1.2.3 控制器

控制器扮演着“总经理”的角色。它控制着计算机的各种操作。例如，控制计算机的输入和输出，以及计算机六个部分之间的通信（输入、输出、控制器、ALU、存储器及辅助存储器）；控制器解释程序指令、启动I/O设备、启动计算机程序的执行，以及保证每件事都能以正确的时间和正确顺序来进行。在这方面，它很像是在繁忙的公路交叉路口指挥来往车辆的交通警的工作。

1.2.4 输入设备

输入和输出

把信息送进（输入）计算机以及从计算机中取出（输出）信息的方法很多。它们的区别仅在于所使用的介质、速度、价格、方便性，以及它们的适用性。

在本书中，我们使用穿孔卡片及读卡机分别作为输入介质及输入设备。使用打印机作为输出设备，它的介质是连续的纸。之所以选择这些设备，主要是由于它们是最常用的设备，并且特别适合于初学者使用。

穿孔卡片

图1.4中所示的穿孔卡，是由H. Hollerith在1880年末发明的。此后，它被命名为Hollerith卡片。而近年来多称为IBM卡片，简称为穿孔卡片。

如果从横向看，该卡片上有12个横行，从顶部开始，前三行分别称为12-行、11-行（或X-行）及0-行，这三行总称为区标行。在这些行上的任何穿孔都称为区标孔。接下去，在0-行之下是1-行、2-行等等，一直到9-行。0-行到9-行称为数字行。并且这几行中所有的孔都称为数字孔。注意，0-行既属于区标行也属于数字行。

现在让我们研究一下卡片的纵向划分。图上卡片的最下边印刷行标有“列号”（这一行和0-行下面印刷的数字相同），从1到80。每一个数字标一列；全卡片共有80列，故俗称为80列卡片。在想看某一列时，可使用这些列号。

还可以从这个图上看到，每一个符号都有一列唯一的孔数来表示它。但是，每一个符号只能占用一列，正像打字机上的符号一样，一个典型的符号只能占据一个打印位置。

对于数字0～9的表示，在各数字行的穿孔处印出这个数字。例如，卡片的第8列中印有4的位置，就意味着表示数字4。只需要一个孔就能表示0～9中的某一数字。

在一列中要用两个孔表示一个字母符号，即一个区标孔和一个数字孔。字母A～I分别对应12-行和数字行1～9的穿孔；字母J～R分别对应11-行和数字行1～9的穿孔；字母S～Z分别对应0-行和数字行2～9的穿孔。可以看出，这种编码和计算机存储器中的EBCDIC编码是不同的。

在穿孔卡片上印刷的标识行和列的数值或其它印刷信息，以及卡片的颜色等都是和卡片的存储信息不相干的，仅用孔的位置或卡片上的穿孔来确定表示什么信息。

读卡机

只有用读卡机，才能把穿孔卡片上的内容传送到存储器中。典型的读卡机的“阅读”

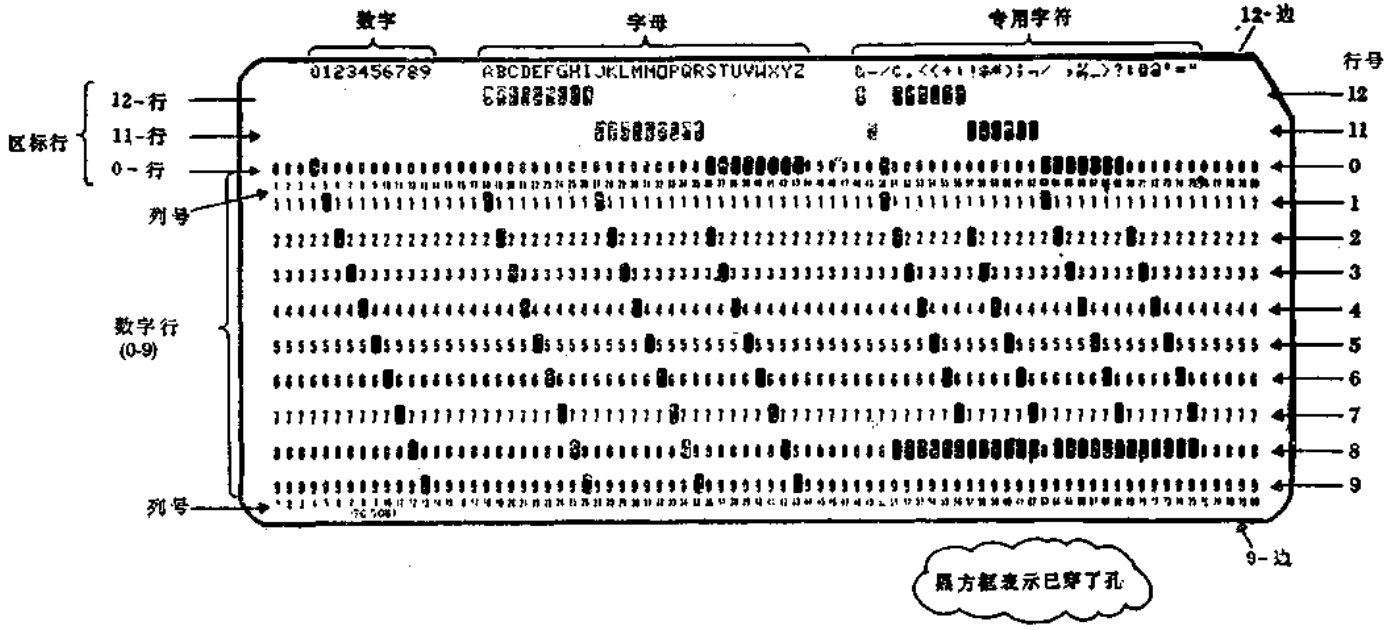


图1.4 Hollerith卡片