

 普通高等教育计算机规划教材

汇编语言与接口技术

第 2 版

叶继华 主编

 提供电子教案

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS



普通高等教育计算机规划教材

汇编语言与接口技术

第2版

叶继华 主编

甘登文	罗贤海	周琪云	杨志文	胡全连
郑 燊	王仕民	江爱文	刘晓东	王懿华 编著
高明华	刘长江	张光河		



机械工业出版社

本书兼顾汇编语言与接口技术两方面知识,阐述了 8086 微型机和 ARM 嵌入式微处理器 PXA270 的结构、工作原理等。在讲解有关硬件和必备知识的基础上,兼顾 16 位和 32 位汇编语言的知识,以 MASM 和 ADS 等工具为主讲解汇编语言的概念和设计知识,然后以典型的可编程接口芯片为载体,讲解汇编语言在接口技术方面的应用,既使学生掌握一般的汇编语言编程方法,又使学生学会如何用汇编实现接口设计。

本书知识体系结构合理、层次分明、逻辑严密、内容丰富,深入浅出,涵盖了汇编语言程序设计的主要知识和接口技术的应用。本书可作为计算机及相关专业的本科、专科生教材,也可作为工程技术人员的参考书。

本书配有授课电子教案,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 2399929378, 电话: 010-88379750)。

图书在版编目(CIP)数据

汇编语言与接口技术/叶继华主编.—2 版.—北京:机械工业出版社,2012.12

普通高等教育计算机规划教材

ISBN 978-7-111-40461-3

I. ①汇… II. ①叶… III. ①汇编语言—程序设计—高等学校—教材
②微型计算机—接口技术—高等学校—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2012)第 274932 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:和庆娣 王 荣

责任印制:张 楠

北京双青印刷厂印刷

2013 年 1 月第 2 版·第 1 次印刷

184mm×260mm·20.25 印张·502 千字

0001-3000 册

标准书号:ISBN 978-7-111-40461-3

定价:39.90 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

教材网:<http://www.cmpedu.com>

销售一部:(010)68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010)88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010)88379203

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- 1) 反映计算机技术领域的新发展和新应用。
- 2) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、多媒体光盘、课程设计和毕业设计指导等内容。
- 3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精练、图文并茂，使学生可以快速掌握，学以致用。
- 4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- 5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

汇编语言相对于高级语言来说,要求更深入地了解硬件结构,编程与调试过程烦琐,但它是计算机能提供给用户的最快而又最有效的语言,也是能够利用计算机所有硬件特性并能直接控制硬件的一种语言,它可以充分发挥机器系统的优越特性,达到最佳的时间和空间运行效率。汇编语言尤其适用于软件与硬件关系密切、软件需要直接有效地控制硬件的场合,如设备控制驱动程序、接口技术等,是高校计算机专业必修的核心课程之一。计算机接口技术是计算机专业的专业课程,也是自动控制、通信等专业的必修课程,作为计算机应用的重要方面,接口技术大量采用可编程接口芯片,其所用的编程语言一般是汇编语言。

目前,整个中国高等教育正处于教学改革的年代,新的教学思路、新的课程体系和教学内容正在形成。在教学过程中,如果能够将“汇编语言”和“接口技术”这两门课程有效地结合起来,不仅可使学生加深对汇编语言的理解和掌握,而且能更好地掌握接口电路的应用编程,强化学生的实践能力。实践证明,通过“汇编语言和接口技术”的集成教学,能够达到这个目的。

随着信息技术的发展,嵌入式系统的应用在不断扩大并呈现高端发展的趋势,高等学校开设嵌入式系统课程已是必然。考虑到国内高校计算机专业的课时限制,为了在有限的课时内既能学习广泛使用的 Intel 的 80x86/Pentium 系列微处理器或者兼容的微处理器,又兼顾嵌入式系统的学习,本书在第一版的基础上,保留了 80x86 家族中最基本的 8086/8088 指令集、8086 微型机主要接口内容的介绍,增加了对嵌入式系统知识的介绍。其中新增的嵌入式系统内容主要为 ARM 系列微处理器的一般概念、基于 ARM 的汇编语言程序设计和典型的 ARM 微处理器接口电路。

汇编语言课程集硬件、软件两方面为一体,随着硬件和软件技术的发展,以及计算机应用领域广度和深度的拓展,相关的理论与技术与硬件都处于更新、被替换状况,现在已经有了集成的编程调试环境等,而目前教程仍以传统的内容为重点,因此必须重视汇编语言内容的更新,本书结合 MASM 5.1、MASM 6.11 和 ADS 三种形式对汇编语言的编程进行了介绍。

全书共分 11 章。第 1 章介绍了微型计算机的有关特点和发展情况,以及计算机所使用的语言,着重阐述了计算机内的数据表示形式,并提供了一个简单的实例。第 2 章介绍了 Intel 8086 微处理器的内部结构及其存储器地址的形成,为后面章节的学习进行必要的准备。第 3 章着重介绍 Intel 8086/8088 的寻址方式和指令系统。第 4 章介绍了 MASM 宏汇编程序支持的各种汇编语言知识,主要叙述了 MASM 汇编程序支持的多种伪指令,以及组成指令语句、伪指令语句等的格式和规则要求,结合 MASM 5.1 编程环境介绍了汇编语言程序的编写和调试过程,最后还介绍了在程序的设计中有时会用到的 DOS 和 BIOS 功能调用。第 5 章主要是综合运用前面的知识,结合程序实例,介绍汇编语言程序设计的方法和技术。第 6 章前面部分介绍了结构和联合、记录等数据类型,宏和重复块等操作,以及汇编语言与高级语言的连接,后面部分着重介绍了 80x86 的编程技术。第 7 章介绍了嵌入式系统的定义、特点、组成以及 ARM 体系结构和处理器的基本知识,最后介绍了处理器的工作模式和具体开发环

境，并举例展现了嵌入式 ARM 的具体开发流程等。第 8 章介绍 ARM 指令集，ARM 指令集包括标准 ARM 指令集和 Thumb 指令集，并介绍 ARM 的伪指令以及 ARM 汇编语言编程知识。第 9 章主要介绍了微机接口的功能、组成、分类，输入/输出设备的数据传送方式，以及中断的基本概念、管理、实现、应用以及 DMA 的原理、DMA 控制器以及它的编程。第 10 章主要介绍可编程接口芯片中的 8255A 并行接口芯片、8251A 串行接口芯片、8253 定时/计数器的概念、功能、内部结构、工作方式以及编程方法。第 11 章主要介绍了 PXA27x 微处理器的典型接口。

本书各章均附有习题供读者练习，以帮助读者掌握和理解有关内容。

本书由叶继华主编，参与编写的还有甘登文、罗贤海、周琪云、杨志文、王仕民、胡全连、高明华、王懿华、郑焱、刘长红、江爱文、刘晓东、张光河，参与本书工作的还有余彤仑、邱晓红、罗文兵、邓方旗等，在此一并致谢。

由于水平有限，书中难免有不妥和错误之处，敬请读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 基础知识	1
1.1 微型计算机概述	1
1.1.1 微型计算机的特点	1
1.1.2 微型计算机系统的层次	1
1.2 计算机语言	3
1.2.1 机器语言	3
1.2.2 汇编语言	3
1.2.3 高级语言	3
1.2.4 汇编语言示例和特点	3
1.3 数据表示	4
1.3.1 数值数据	4
1.3.2 字符数据	7
1.4 习题	9
第 2 章 8086 微处理器	10
2.1 Intel 8086 微处理器	10
2.1.1 Intel 8086 CPU 的内部结构	10
2.1.2 Intel 8086 内部寄存器	12
2.2 存储器物理地址的形成	14
2.2.1 存储器结构	14
2.2.2 物理地址的形成	15
2.2.3 存储器单元的地址和内容	17
2.3 习题	18
第 3 章 8086 指令系统	19
3.1 8086 的寻址方式	19
3.1.1 数据的寻址方式	19
3.1.2 程序转移地址的寻址方式	24
3.1.3 对端口的寻址方式	26
3.2 8086 的指令系统	26
3.2.1 数据传送类指令	26
3.2.2 算术运算类指令	31
3.2.3 位操作类指令	38
3.2.4 串操作类指令	42

3.2.5	控制转移类指令	45
3.2.6	处理机控制类指令	54
3.3	习题	55
第4章	MASM 汇编语言知识	58
4.1	MASM 汇编语言格式	58
4.1.1	指令语句格式	58
4.1.2	伪指令语句格式	58
4.2	汇编语句表达式	59
4.2.1	常量、变量和表达式	59
4.2.2	标号	62
4.2.3	表达式中的运算符	62
4.3	伪指令	67
4.3.1	数据定义伪指令	68
4.3.2	符号定义伪指令	69
4.3.3	段定义相关伪指令	69
4.3.4	子程序(过程)定义伪指令	73
4.3.5	其他伪指令	73
4.3.6	程序正常结束方式	75
4.3.7	MASM 汇编语言源程序结构	76
4.4	MASM 汇编语言程序的上机过程	78
4.4.1	MASM 汇编程序的有关概念	78
4.4.2	MASM 汇编语言程序的上机过程	79
4.5	DOS 功能调用和 BIOS 功能调用	84
4.5.1	常用的 DOS 功能调用	85
4.5.2	DOS 功能调用	87
4.5.3	BIOS 功能调用	94
4.6	习题	101
第5章	汇编语言程序设计	104
5.1	程序设计概述	104
5.1.1	汇编语言程序设计的一般步骤	105
5.1.2	流程图	106
5.2	顺序程序设计	106
5.3	分支程序设计	111
5.3.1	用条件转移指令实现程序分支	112
5.3.2	用跳转表实现多路分支	115
5.4	循环程序设计	119
5.4.1	循环程序的结构	119
5.4.2	循环控制的方法	120
5.4.3	单重循环程序设计	120

5.4.4	多重循环程序设计	126
5.5	子程序设计	130
5.5.1	子程序的概念	131
5.5.2	子程序的定义	131
5.5.3	子程序的设计方法	131
5.5.4	子程序应用举例	133
5.5.5	子程序的嵌套与递归调用	138
5.6	模块化程序设计	140
5.7	习题	142
第6章	高级汇编语言程序设计	148
6.1	高级汇编技术	148
6.1.1	结构和联合	148
6.1.2	记录	150
6.1.3	宏	152
6.1.4	重复汇编	164
6.1.5	条件汇编	166
6.2	MASM 汇编语言与高级语言的连接	168
6.2.1	PASCAL 语言程序与汇编语言程序的连接	168
6.2.2	C 语言程序与汇编语言程序的连接	169
6.3	80x86 的寻址方式和扩充的指令	171
6.3.1	80x86 数据的寻址方式	171
6.3.2	80x86 程序转移地址的寻址方式	172
6.3.3	80x86 扩充的指令	172
6.3.4	80x86 扩充的伪指令	177
6.3.5	保护方式专用指令	179
6.4	80x86 汇编语言编程	181
6.4.1	实地址方式汇编语言程序设计	181
6.4.2	保护方式的进入和退出	186
6.4.3	保护方式汇编语言程序设计	187
6.5	习题	193
第7章	嵌入式系统概述	195
7.1	嵌入式系统的概念	195
7.1.1	嵌入式系统的定义	195
7.1.2	嵌入式系统的特点	195
7.1.3	嵌入式系统的组成	196
7.2	ARM 体系结构简介	196
7.2.1	ARM 处理器系列	197
7.2.2	ARM 处理器的工作基础	198
7.3	ARM 开发环境	202

7.3.1	ARM 的开发流程	202
7.3.2	开发环境	202
7.3.3	简单实例	204
7.4	习题	210
第 8 章	ARM 汇编语言程序设计	211
8.1	ARM 处理器的寻址方式	211
8.2	ARM 处理器汇编语言指令	212
8.2.1	ARM 指令概述	213
8.2.2	Thumb 指令概述	224
8.3	ARM 汇编语言伪指令	224
8.3.1	符号定义伪指令	224
8.3.2	数据定义伪指令	225
8.3.3	汇编控制伪指令	227
8.3.4	其他常用的伪指令	228
8.4	ARM 汇编语言语句格式	228
8.4.1	ARM 汇编语言中的符号	228
8.4.2	ARM 汇编语句中的表达式	230
8.5	ARM 汇编语言程序设计	230
8.5.1	汇编语言的程序格式	230
8.5.2	ARM 指令的汇编程序设计	231
8.6	习题	238
第 9 章	微型机接口	239
9.1	微型机接口的基本知识	239
9.2	I/O 设备数据传送方式	240
9.2.1	微型机 I/O 端口的布局	240
9.2.2	I/O 端口地址的译码	241
9.2.3	CPU 与外设之间的信息传送方式	241
9.3	中断系统	242
9.3.1	中断系统概述	242
9.3.2	8086/8088 的中断系统	244
9.3.3	可编程中断控制器 8259A	247
9.4	DMA	255
9.4.1	DMA 传送方式	255
9.4.2	DMA 控制器 8237A	256
9.5	习题	263
第 10 章	可编程接口芯片	264
10.1	8255 并行接口	264
10.1.1	并行接口的概念	264
10.1.2	Intel 8255A 可编程并行接口	264

10.1.3 Intel 8255A 的控制字	266
10.1.4 Intel 8255A 的工作方式	267
10.1.5 Intel 8255A 编程	269
10.2 8251 串行接口	272
10.2.1 串行接口的概念	272
10.2.2 Intel 8251A 可编程串行接口	272
10.2.3 Intel 8251A 编程	275
10.3 可编程定时/计数器芯片	278
10.3.1 定时/计数器的概念	279
10.3.2 Intel 8253 可编程定时/计数器	279
10.3.3 Intel 8253 控制字和工作方式	282
10.3.4 Intel 8253 编程	286
10.4 习题	287
第 11 章 嵌入式系统接口电路	288
11.1 PXA27x 处理器	288
11.1.1 PXA270 处理器组成	288
11.1.2 XScale 内核	290
11.2 PXA270 处理器接口电路	290
11.2.1 GPIO	291
11.2.2 存储器接口	292
11.2.3 UART	294
11.2.4 中断控制器	295
11.2.5 LCD 控制器	297
11.2.6 以太网接口	299
11.2.7 JTAG 接口	302
11.2.8 其他接口	303
11.3 习题	305
附录	306
附录 A DEBUG 的使用	306
附录 B 汇编程序出错信息	311
参考文献	314

第1章 基础知识

随着计算机技术的发展，计算机尤其是微型计算机已经成为社会生产活动中不可缺少的工具。本章介绍了微型计算机的有关知识以及计算机所使用的语言，着重阐述了汇编语言的有关知识和计算机内部的数据表示形式。

1.1 微型计算机概述

本节主要介绍了微型计算机的特点和微型计算机系统的组成。

1.1.1 微型计算机的特点

电子计算机从性能、体积上通常可分为巨型机、大型机、中型机、小型机、微型机和单片机等六类。微型计算机（简称为微型机、微机）是伴随着大规模集成电路的发展而诞生的，它有以下几个特点：

- (1) 采用大规模和超大规模集成电路，主流的微处理器采用单片 VLSI 形式。
- (2) 标准的工业化装配结构，体积小、重量轻，稳定性好，便于系统升级。
- (3) 开放的标准体系结构，多元化大规模工业化生产，性能价格比高。
- (4) 微型计算机的应用范围十分广泛，信息化社会中它无所不在。

1.1.2 微型计算机系统的层次

当今微电子技术水平迅速提高，平均每 18 个月电子电路的集成度提高一倍，使得微处理器结构设计者可在片内实现各种先进体系结构，微处理器及外围支援器件的性价比达到前所未有的水平，微计算机系统的性价比迅速提高，达到传统意义的工作站与小型机的水平。

1. 微处理器（Microprocessor, μP ）

微型计算机的产生与发展主要表现在其核心部件——微处理器的发展上，每当一种新的微处理器出现后，都能带动微型计算机其他部件的发展。

微处理器不包含微型计算机硬件的全部功能，但它是微型计算机控制、处理的核心。目前主流的微处理器采用单片 VLSI 电路，体系结构技术、工作频率已达空前高的水平。

主流微处理器具有通用性，不仅用于微型机也用于工作站及超级计算机。

微处理器一般由算术逻辑部件、寄存器、控制部件及内部总线组成，如图 1-1 所示。

2. 微型计算机（Microcomputer, μC ）

微型计算机是指以微处理器为核心，配以存储器、输入/输出接口和相应的辅助电路所构成的裸机。把微型计算机集成在一个芯片上就构成了单片微型计算机（单片机）。

微处理器是执行指令的核心，它的性能决定了整个微型计算机的性能。

存储器用于指令代码、操作数和运行结果的存储。

输入/输出接口电路用于微处理器与外部设备的连接，主要包括：并口、串口、外存接口、显示器接口、网络接口、声音接口等。

系统总线将上述模块连接起来，作为各种信息的通路，按信息类别分为数据、地址、控制三类总线。如图 1-2 所示为微型计算机的基本结构。

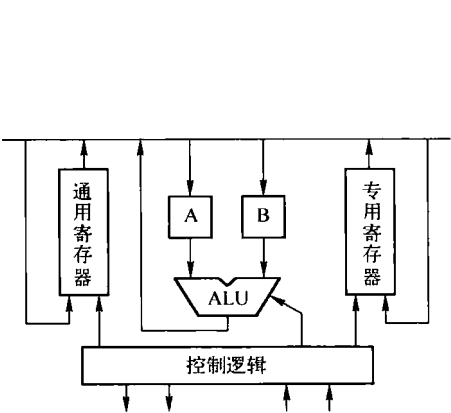


图 1-1 微处理器结构框图

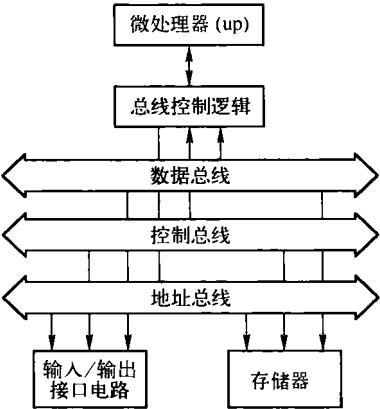


图 1-2 微型计算机基本结构

3. 微型计算机系统（Microcomputer System）

微型计算机系统是指以微型计算机为主体，配以相应的外围设备及其他专用电路、电源、面板、机架以及软件系统所构成的系统。如图 1-3 所示为微处理器、微型计算机、微型计算机系统三者的关系。

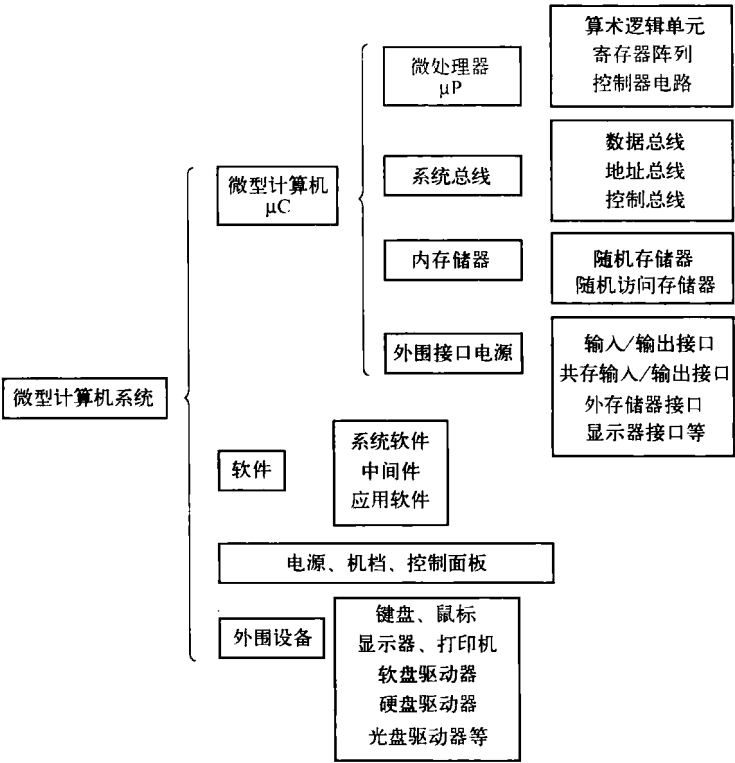


图 1-3 微处理器、微型计算机、微型计算机系统三者关系

软件系统主要包括系统软件、中间软件和应用软件。

外围设备主要有软驱、硬驱、光驱、键盘、鼠标和显示器等。

1.2 计算机语言

计算机语言（Computer Language）是用于人与计算机之间通信的语言，是计算机能接受的语言，是人与计算机之间传递信息的媒介。计算机语言的种类非常多，总体可分为机器语言、汇编语言、高级语言三大类。

1.2.1 机器语言

计算机能够直接识别的数据是由二进制数 0 和 1 组成的代码。机器指令就是用二进制代码组成的指令，一条机器指令控制计算机完成一个基本操作。

用机器语言编写的程序是计算机唯一能够直接识别并执行的程序，而用其他语言编写的程序必须经过翻译才能转换成机器语言程序，所以，机器语言程序被称为目标程序。

1.2.2 汇编语言

为了克服机器语言的缺点，人们采用助记符表示机器指令的操作码，用变量代替操作数的存放地址等，这样就形成了汇编语言。所以汇编语言是一种用符号书写的、基本操作与机器指令相对应（一一对应）的、并遵循一定语法规则的计算机语言。

用汇编语言编写的程序称为汇编源程序，扩展名为 ASM。

汇编语言是一种符号语言，比机器语言容易理解和掌握，也容易调试和维护。但是，汇编语言源程序要翻译成机器语言程序才可以由计算机执行，这个翻译的过程称为“汇编”。这种把汇编源程序翻译成目标程序的语言加工程序称为汇编程序。汇编程序的主要功能有：

- (1) 检查源程序。
- (2) 测出源程序中的语法错误，并给出出错信息。
- (3) 产生源程序的目标程序，并可给出列表文件（LST 文件）。
- (4) 展开宏指令。

1.2.3 高级语言

汇编语言虽然较机器语言直观，但仍然烦琐难懂。于是人们研制出了高级程序设计语言。高级程序设计语言接近于人类自然语言的语法习惯，与计算机硬件无关，易被用户掌握和使用。

目前广泛应用的高级语言有多种，如 BASIC、Fortran、Pascal、C、C++ 等。

1.2.4 汇编语言示例和特点

1. 汇编语言程序示例

图 1-4 是一个汇编语言源程序的示例。其中定义了一个数据段（1~3 行）和一个代码段（5~16 行），在数据段中定义了一个字符串，代码段中的代码完成该字符串的输出显示。

```

01 DATAS SEGMENT
02     STRING DB 'Hello World!',13,10,'$'
03 DATAS ENDS
04
05 CODES SEGMENT
06     ASSUME CS:CODES,DS:DATAS
07 START:
08     MOV AX,DATAS
09     MOV DS,AX
10     LEA DX,STRING
11     MOV AH,9
12     INT 21H
13
14     MOV AH,4CH
15     INT 21H
16 CODES ENDS
17     END START
18

```

图 1-4 一个汇编语言源程序示例

2. 汇编语言的特点

- (1) 汇编语言与处理器密切相关。
- (2) 汇编语言程序效率高。
- (3) 编写汇编语言源程序比编写高级语言源程序烦琐。
- (4) 调试汇编语言程序比调试高级语言程序困难。

3. 汇编语言的主要应用场合

- (1) 程序执行占用较短的时间，或者占用较小存储容量的场合。
- (2) 程序与计算机硬件密切相关，程序直接控制硬件的场合。
- (3) 需提高大型软件性能的情况。
- (4) 没有合适的高级语言的场合。

1.3 数据表示

计算机是用于处理数据的，在计算机内部数据有数值数据和字符数据两种主要的表示形式。

1.3.1 数值数据

人们在日常生活中习惯使用十进制数，而二进制数由于简单、容易实现，是数字系统中、特别是计算机中广泛采用的一种数制。但使用二进制数不方便，比如表示一个十进制数时，需用 4 位二进制数才能表示一位十进制数，所用的位数太多，读/写很不方便，所以在实际工作中又常采用八进制或十六进制，因此计算机中数据表示常用二进制、八进制、十进制和十六进制。

二进制：由一串 0、1 组成，其后跟字母 B。

八进制：由数字 0~7 组成，其后跟字母 O 或 Q。

十进制：由数字 0~9 组成，其后跟字母 D，可缺省。

十六进制：由 0~9 及 A~F 组成，其后跟字母 H，如果数的第一个字符是 A~F，则应在其前加数字 0。由于二进制数的基数太小，书写和阅读都不方便，而十六进制的基数 16 =

2^4 ，这样二进制数与十六进制之间能方便地转换。因此，习惯把二进制数改写成十六进制数，在汇编语言程序设计时尤其如此。

数值数据分为有符号数和无符号数。无符号数最高位表示数值，而有符号数最高位表示符号。有符号数由两部分组成：一部分是表示数的符号，另一部分是表示数的数值。由于数的符号是一个具有正、负两种值的离散信息，所以它可以用一位二进制数来表示。通常是以 0 表示正数，以 1 表示负数。对于一个 n 位二进制数，如果数的第一位为符号位，那么余下的 $n-1$ 位就表示数的数值部分。我们把直接用正号“+”和负号“-”来表示符号的二进制数称为符号数的真值。数的真值形式是一种原始形式，无法直接用在数字计算机中。但是，当将符号数值化之后，便可以在计算机中使用它了。因此在计算机中使用的符号数便称为机器数。一般机器数有三种表示形式，即原码、反码和补码，常用的是补码。

1. 原码

原码又称为“符号-数值表示”。用当原码形式表示正数和负数时，最高位是符号位。对于正数，符号位表示的 0；对于负数，符号位表示的 1，其余各位表示数值位，这样就得到了有符号数的原码表示。

【例 1-1】 假如两个带符号的二进制数分别为 S_1 和 S_2 ，其真值形式为

$$S_1 = +11001 \quad S_2 = -01011$$

则 S_1 和 S_2 的原码表示形式为

$$[S_1]_{\text{原}} = 011001 \quad [S_2]_{\text{原}} = 101011$$

根据上述原码形成规则，一个 n 位的整数 S （包括一位符号位）的原码一般表达式为

$$[S]_{\text{原}} = \begin{cases} S & 0 \leq S < 2^{n-1} \\ 2^{n-1} - S & -2^{n-1} < S \leq 0 \end{cases}$$

对于定点小数而言，一般将小数点定在最高位的左边，此时，数值小于 1。定点小数原码一般表达式为

$$[S]_{\text{原}} = \begin{cases} S & 0 \leq S < 1 \\ 1 - S & -1 < S \leq 0 \end{cases}$$

由原码的一般表达式中可以得出：

(1) 当 S 为正数时， $[S]_{\text{原}}$ 和 S 的区别只是增加一位用 0 表示的符号位。由于在数的左边增加一位 0 对该数的数值并无影响，所以 $[S]_{\text{原}}$ 就是 S 本身。

(2) 当 S 为负数时， $[S]_{\text{原}}$ 和 S 的区别是增加了一位用 1 表示的符号位。

(3) 在原码表示中，有两种不同形式的 0，即

$$[+0]_{\text{原}} = 0.00\cdots 0$$

$$[-0]_{\text{原}} = 1.00\cdots 0$$

2. 反码

反码又称为“对 1 的补数”。当用反码表示时，最高位为符号位，符号位为 0 代表正数，符号位为 1 代表负数。对于正数，反码和原码相同，即符号位用 0 表示，数值位不变。而对于负数，反码的数值位按位取反形成，即 0 变 1、1 变 0。所以，反码数值的形成与它的符号位有关。

【例 1-2】 假如两个带符号的二进制数分别为 S_1 和 S_2 ，其真值形式为

$$S_1 = +11001 \quad S_2 = -01011$$

则 S_1 和 S_2 的反码表示形式为

$$[S_1]_{\text{反}} = 011001 \quad [S_2]_{\text{反}} = 110100$$

根据上述的反码形成规则，一个 n 位的整数 S （包括一位符号位）的反码一般表达式为

$$[S_1]_{\text{反}} = \begin{cases} S & 0 \leq S < 2^{n-1} \\ (2^n - 1) + S & -2^{n-1} < S \leq 0 \end{cases}$$

同样，对于定点小数，如果小数部分的位数为 m ，则它的反码一般表达式为

$$[S_1]_{\text{反}} = \begin{cases} S & 0 \leq S < 1 \\ (2 - 2^m) + S & -1 < S \leq 0 \end{cases}$$

从反码的一般表达式可以看出：

- (1) 正数 S 的反码 $[S]_{\text{反}}$ 与原码 $[S]_{\text{原}}$ 相同。
- (2) 对于负数 S ，其反码 $[S]_{\text{反}}$ 的符号位为 1，数值部分是将原码数值按位求反。
- (3) 在反码表达式中，0 的表示有两种不同的形式，即

$$[+0]_{\text{反}} = 0.00 \cdots 0$$

$$[-0]_{\text{反}} = 1.11 \cdots 1$$

3. 补码

补码又称为“对 2 的补数”。在补码表示方法中，正数的表示同原码和反码的表示是一样的，即符号位用 0 表示，数值位值不变；而对于负数，将原码转变成补码的规则是：符号位不变，仍为 1，数值部分逐位变反，在最低位加 1，也就是反码加 1 形成。

对一个二进制数按位求反后在末位加 1 的运算称为求补运算。

对一个正数的补码将其符号位和数值位都按位求反后再在末位加 1，可以得到与此正数相对的负数的补码；对负数的补码同样操作可得到与此负数相对的正数的补码，即

$$[X]_{\text{补}} \xrightarrow{\text{求补}} [-X]_{\text{补}} \xrightarrow{\text{求补}} [X]_{\text{补}}$$

【例 1-3】 如两个带符号的二进制数分别为 S_1 和 S_2 ，其真值表达式为

$$S_1 = +11001 \quad S_2 = -01011$$

则 S_1 和 S_2 的补码表示形式为

$$[S_1]_{\text{补}} = 011001 \quad [S_2]_{\text{补}} = 110101$$

根据上述补码形成规则，一个 n 位的整数 S （包括一位符号位）的补码一般表达式为

$$[S_1]_{\text{补}} = \begin{cases} S & 0 \leq S < 2^{n-1} \\ 2^n + S & -2^{n-1} \leq S < 0 \end{cases}$$

同样，对于定点小数，补码一般表达式可写成

$$[S_1]_{\text{补}} = \begin{cases} S & 0 \leq S < 1 \\ 2 + S & -1 \leq S < 0 \end{cases}$$

由补码的一般表达式可以看出：

- (1) 正数 S 的补码 $[S]_{\text{补}}$ 、反码 $[S]_{\text{反}}$ 和原码 $[S]_{\text{原}}$ 是相同的。
- (2) 对于负数，补码 $[S]_{\text{补}}$ 的符号位为 1，其数值部分为反码的数值末位加 1。
- (3) 在补码表示法中，0 的表示形式是唯一的。即

$$[+0]_{\text{补}} = 0.00 \cdots 0$$

$$[-0]_{\text{补}} = 0.00 \cdots 0$$