全国高等院校应用型创新规划教材·计算机系列

省重点
精品课程

汇编语言程序设计

王晓虹　编著

配套资源

- 习题与答案
- 教学视频
- PPT
- 二维码教材

清华大学出版社

型创新规划教材·计算机系列

汇编语言程序设计

王晓虹　编著

清华大学出版社
北京

内 容 简 介

本书以经典的 Intel 8086/8088CPU 指令系统与 Microsoft 宏汇编为背景，系统地介绍了汇编语言程序设计的基本理论和方法。

本书共十二章，前九章主要内容包括：宏汇编语言程序设计的基础知识、指令系统、常用伪指令、汇编语言语法规则和程序设计方法、子程序与多模块编程、宏功能程序设计。后三章主要介绍了 8086、8088 汇编语言的应用，包括输入输出程序设计、中断的基本概念及其开发应用技巧、文件操作编程方法等内容。

为方便自学，在重点章节后面增加了理解与练习，通过例题分析，加强对汇编语言的理解与掌握。本书可作为高校计算机本科专业的教材及相关专业本科生的教材，也可作为教师、非计算机专业的研究生及计算机应用技术人员的参考书。

图书在版编目(CIP)数据

汇编语言程序设计/王晓虹编著. —北京：清华大学出版社，2019
(全国高等院校应用型创新规划教材 • 计算机系列)

ISBN 978-7-302-51346-9

Ⅰ. ①汇… Ⅱ. ①王… Ⅲ. ①汇编语言—程序设计—高等学校—教材 Ⅳ. ①TP313

中国版本图书馆 CIP 数据核字(2018)第 229116 号

责任编辑：陈冬梅
装帧设计：杨玉兰
责任校对：周剑云
责任印制：沈　露
出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn, 010-62791865
印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张：**16　　**字　数：**389 千字
版　　次：2019 年 1 月第 1 版　　**印　次：**2019 年 1 月第 1 次印刷
定　　价：48.00 元

产品编号：076149-01

前　言

汇编语言程序设计是计算机及相关专业必修的一门主要专业基础课程。同其他高级语言相比，汇编语言是面向机器的低级语言。由于汇编语言可以直接对硬件资源进行编程，因此具有更高的执行效率，更能充分发挥计算机硬件的功能和特性。汇编语言对掌握相关硬件的程序设计方法、从事相关软件开发和应用起到重要的作用。

本书总共为十二章内容。第一章～第九章以 Intel 8086/8088 系列微机为背景，系统地介绍了汇编语言的基本概念、CPU 中寄存器组织与存储器分段管理的技术、汇编语言指令系统的语法及相关应用、汇编语言程序设计的基本方法。 第十章～第十二章介绍了汇编语言的应用，包括：输入输出及其程序设计方法；中断的基本概念及其开发应用技巧；文件操作编程方法等内容。

本书在涵盖汇编语言基本内容的基础上，突出精讲多练，鼓励学生自主学习，在保证知识的连续性、完整性的同时，对传统的教材内容进行了较大的改动，力求内容精练、突出重点、注重应用。为鼓励学生自主学习和增加对学习内容的选择性，一般在每章后面增加了理解与练习。理解与练习是对重点、难点问题通过举例和例题分析，能在概念上和应用技巧上进一步加深理解和掌握。

全书内容由辽宁石油化工大学的王晓虹、毕于深执笔，其中全书内容由王晓虹、毕于深组织和审核。

本书中的疏漏和不妥之处敬请批评指正。

编　者

2018 年 1 月于辽宁石油化工大学

目　录

第一章　绪　　论

学习要点及目标

- 了解微型计算机系统的组成。
- 了解汇编语言的概念、特点及应用范围。
- 掌握机器语言、汇编语言、高级语言的特点。
- 了解学习汇编语言的意义。

绪论.mp4

核心概念

微型计算机系统　机器语言　汇编语言　高级语言

引导案例

简述机器语言、汇编语言、高级语言的特点。

机器语言、汇编语言是面向机器的语言，即低级语言。高级语言是面向程序设计人员的。若直接对计算机硬件进行操作，通常采用汇编语言编程，以提高运行效率。

案例导学

计算机语言一般包括机器语言、汇编语言、高级语言。汇编语言作为面向机器的语言，用汇编语言设计程序，可以充分利用和发挥计算机硬件的特性和优势。因此，汇编语言在计算机应用中占有重要的地位。本章重点介绍汇编语言的特点及应用。

汇编语言作为面向机器的语言，用汇编语言设计程序，可以充分利用和发挥计算机硬件的特性和优势。因此，汇编语言在计算机应用中占有重要的地位。

第一节　微型计算机系统组成

微型计算机系统包括硬件和软件两部分。

一、微型计算机硬件基本结构

微型计算机的硬件系统主要由微处理器(CPU)、存储器(RAM，ROM)、I/O 接口、I/O 设备、系统总线等构成。总线结构是微机体系结构的特点之一，微处理器、存储器、I/O

接口电路等通过系统总线连接起来，构成了主机部分，I/O 设备通过 I/O 接口实现与主机的信息交换。

1. 微处理器

微处理器是一片集成电路。它是微机系统的核心部件，其主要功能是实现算术逻辑运算以及对全机进行控制。微处理器主要包括运算器和控制器。其中，运算器又称算术逻辑部件，可以完成各种算术运算、逻辑运算以及移位、传输等操作。控制器又称控制部件，它向计算机的各部件发出相应的控制信号，使 CPU 内、外各部件间协调工作，是全机的指挥控制中心。

2. 存储器

存储器是计算机的存储和记忆装置，用来存储程序或数据，由存储单元构成。存储器包括只读存储器 ROM 和随机读写存储器 RAM。ROM 中固化着基本输入输出设备驱动程序和微机启动自检程序等，称为 BIOS 系统程序，它是操作系统软件的组成部分。RAM 又称为内存储器(内存)，它由多片集成电路组成一个个内存条，可方便地在主板上插拔。RAM 用来存放程序和数据，任何要执行的程序和要处理的数据必须先装入 RAM 才能工作。

3. I/O 接口

I/O 接口是计算机与 I/O 设备之间信息交换的桥梁。I/O 接口是由多种集成电路芯片及其他电子器件组成的电路。它是主机与外设、外设与外设之间的硬件接口，不同的外设通过配套的接口电路实现数据缓冲、传送、信号转换等。

4. 总线

总线(BUS)是一组公共数据线、地址线和控制信号线。它把系统中的各个设备及部件连接起来，构成微机的硬件系统。总线在工作时，数据及各种信息传送是分时操作的。计算机采用总线结构，各部件均挂接在系统总线上，使得系统结构简单，易于维护，并为系统功能的扩充或升级提供了很大的灵活性。

5. 外部设备

外部设备一般包括外部存储器(软盘、硬盘)及实现人与计算机交换信息的输入输出装置(如键盘、显示器、打印机等)。外部设备必须通过 I/O 接口才能与系统总线相连。

二、微机软件系统

微机软件系统分为系统软件和用户软件两个层次。

系统软件是由计算机生产厂家提供给用户的一组程序，它又可分为两类：一类是面向机器系统，是操作的系统程序，负责对系统的软、硬件资源进行有效的管理，建立计算机的工作环境；另一类是面向用户的软件，对用户编制的程序进行编辑、编译、连接，加工成计算机能直接执行的目标程序。软件系统的构成如图 1-1 所示。

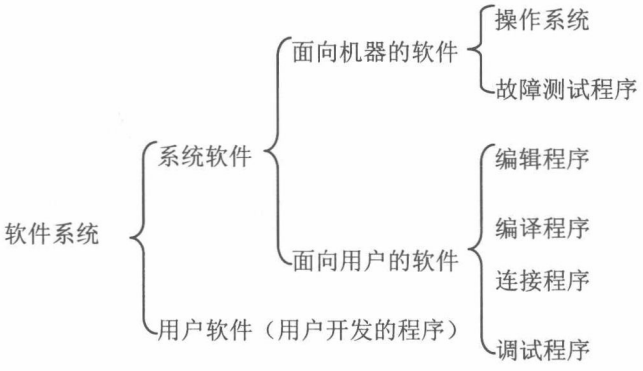

图 1-1 微机软件系统的构成

第二节 计算机语言

当人们使用计算机来完成某些任务时，就必须告诉它怎样具体地处理这些任务。同计算机进行这种交流的工具是什么呢？就是计算机语言。人们利用计算机语言告诉计算机某个问题应如何处理，先做什么，后做什么，即人们用计算机语言安排好处理步骤，每一步都是用计算机语言描述的。这种用计算机语言描述的处理步骤，称为程序。计算机执行程序时，就按照处理步骤完成人所规定的任务。

计算机语言可分为 3 类：机器语言、汇编语言和通用语言。前两类是面向机器的，一般称为低级语言；后一类是面向程序设计人员的，一般称为高级语言。

一、机器语言

虽然可以使用各种语言编写程序，但计算机却只能识别在设计机器时事先规定好的机器指令。机器指令即指挥计算机完成某一基本操作的命令。它们均由 0 和 1 二进制代码串组成。机器指令的一般格式为：

操作码字段	地址码字段

操作码字段指出该指令执行何种操作，地址码字段指出被操作的数据(操作数)和操作结果的存放位置。

例如，将地址为 0000 0100B 的字节存储单元中的内容加 3，若用 Intel 8086/8088 机器指令来完成该操作，则相应的机器指令为：

10000011
00000110
00000100
00000011

这条指令共 4 个字节，其中前 2 个字节的二进制代码是操作码，表示要进行“加”操作，并指明了以何种方式取得两个加数；第三个字节二进制代码指出了第一个加数存放在

偏移地址为00000100B的内存单元，最后一个字节二进制代码指出第二个加数3。

机器指令也常被称为硬指令，它是面向机器的，即不同的计算机规定了自己所特有的、一定数量的基本指令(指令系统)。用机器指令进行描述的语言叫作机器语言，用机器语言编写的程序称为机器语言程序或目标程序。目标程序中的二进制机器指令代码称为目标代码。

使用任何语言编写的程序最终都要转换成机器语言程序，才能被计算机识别、理解并执行。

二、汇编语言

由于机器指令是用二进制表示的，编写、阅读和调试程序都相当困难。于是，人们想出了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，还可以在指令前冠以标号，用来代表该指令的存放地址等。这种用符号书写的、与机器指令一一对应的、并遵循一定语法规则的符号语言就是汇编语言。用汇编语言编写的程序称为汇编语言源程序。例如前面的例子，用汇编语言来书写就成为：

```
MOV  SI, 0004H
ADD  BYTE PTR [SI], 3
```

由于汇编语言是为了方便用户而设计的一种符号语言，因此，用它编写出的源程序并不能直接被计算机识别，必须将它翻译成机器语言程序即目标程序才能被计算机执行。这个翻译工作是由系统软件提供的一个语言加工程序完成的。这个把汇编语言源程序翻译成目标程序的程序称为汇编程序，汇编程序进行翻译的过程叫汇编。这里，汇编程序相当于一个翻译器，它加工的对象是汇编语言源程序，加工的结果是目标程序，如图1-2所示。

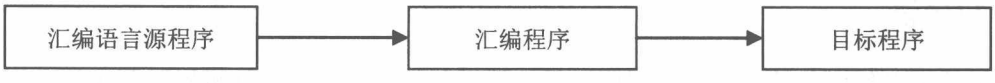

图1-2　汇编语言源程序翻译成目标程序

为了能让汇编程序正确完成翻译工作，必须告诉它源程序从什么位置开始存放，汇编到什么位置结束，数据应放在什么位置，数据类型是什么，等等。这就要求源程序中有一套告诉汇编程序如何进行汇编的命令，这种汇编命令称为伪指令。由此可见，指令助记符、语句标号、数据变量、伪指令及它们的使用规则构成了整个汇编语言的内容。

与机器语言相比，汇编语言易于理解和记忆，所编写的源程序容易阅读和调试。汇编语言的魅力还在于程序占用内存少，执行速度快，并且可直接对硬件编程，能充分发挥计算机的硬件功能。

三、高级语言

高级语言是用接近自然语言的符号对计算机操作步骤进行描述的计算机语言，如Pascal、C 语言等。目前计算机高级语言有数百种之多。高级语言的特点是程序容易编址和调试，科学计算和事件处理能力强，且与机器硬件无关，通用性强；但生成目标代码长度长，占用内存多，执行速度较慢。

上述的高级语言是面向过程的程序设计语言。随着计算机软件技术的发展，出现了面向对象的可视化程序设计语言，如 Java、C++、Delphi 等，这种语言是将数据(属性)及数据的处理过程(方法)封装起来，用对象加以描述。程序设计者通过实现对象，完成软件的开发，但数据处理过程的具体实现采用的仍是面向过程的方法。

第三节 汇编语言的应用范围

汇编语言是计算机所能提供的最快、最有效的语言，也是能够利用计算机所有硬件特性的唯一语言。汇编语言主要应用在实时性要求高、对硬件设备进行控制的场合，如过程控制、媒体接口、通信等用高级语言难以实现的操作，必须使用汇编语言。

目前系统软件的研制虽然已有不少采用高级语言，但给出的目标程序往往还是采用汇编语言的形式，而且还有不少系统软件，必须使用汇编语言编写。汇编语言程序是系统软件的核心成分之一。因此，对于开发、应用计算机的技术人员来说，必须掌握汇编语言，才能分析、修改和扩充计算机系统软件，增加计算机功能。

汇编语言程序设计是从事计算机研究的基础，是计算机研究和应用技术人员必须掌握的一门技术。由于汇编语言与计算机硬件特性有关，因此，要学习汇编语言，就必须首先了解机器硬件资源的结构和使用情况、数据类型及表示方法等。

绪论.ppt

第二章　汇编语言基础知识

学习要点及目标

- 掌握数据类型及表示方法。
- 掌握微处理器的功能结构及其可编程寄存器。
- 理解存储器分段的原理和方法，掌握存储器逻辑地址与物理地址的概念和转换方法。
- 理解堆栈的概念、结构和压栈弹栈的过程。

核心概念

溢出　物理地址　逻辑地址　存储器分段　堆栈

引导案例

进行下列二进制数的运算后，标志 OF、ZF、SF、CF 的值各是多少？

10011010B+11001101B

进行二进制运算后，结果为 01100111B，OF=1，ZF=0，SF=0，CF=1。

案例导学

在计算机内部采用二进制进行运算，引导案例对运算结果影响标志位的状态进行了分析。通常，使用汇编语言编程，直接涉及对 CPU 内部寄存器的操作、存储器的定义和分配，以及对数据类型的转换等问题。因此，本章从程序设计的角度，介绍数据类型及表示方法、微处理器功能结构及其可编程寄存器组，以及存储器等内容。

使用汇编语言编程，直接涉及对 CPU 内部寄存器的操作、存储器的定义和分配，以及对数据类型的转换等问题。因此，本章从程序设计的角度，介绍数据类型及表示方法、微处理器功能结构及其可编程寄存器组，以及存储器等内容。

第一节　数据类型

一、数制及相互转换

8086/8088 宏汇编语言源程序中允许使用二进制数、八进制数、十进制数和十六进制

数。在书写不同数制的数时，常在一个数的尾部用一个字母来表示该数的数制。二进制数用字母 B(Binary)，八进制数用字母 O(Octal)，十进制数用字母 D(Decimal)，十六进制数用字母 H(Hexadecimal)。其中，十进制数尾部字母 D 可缺省。汇编程序在对源程序进行汇编时，能自动将不同数制的数转换成二进制数。

数据类型.mp4

不同进制数之间的对应关系如表 2-1 所示。

表 2-1　不同进制数之间的对应关系

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

在编写或阅读程序时，常需要将一种进制数转换为另一种进制数。熟练掌握不同进制数之间的转换，是进行汇编语言程序设计的基础。

1. N 进制数转换为十进制数

转换方法：按权相加。

【例 2.1】　求 10011.101B 的十进制值。

$$10011.101B=1\times2^4+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-3}$$
$$=16+2+1+0.5+0.125$$
$$=19.625D$$

八进制、十六进制与十进制之间的转换，除基数不同外，方法一样。

2. 十进制数转换为 N 进制数

转换方法：整数部分，除基(N)取余；小数部分，乘基(N)取整。

【例 2.2】　求十进制数 325.8125 的二进制表示。

整数部分：

除基数 2		余数	
2	325		
2	162	1	最低位
2	81	0	
2	40	1	
2	20	0	
2	10	0	
2	5	0	
2	2	1	
2	1	0	
	0	1	最高位

得：325D=101000101B

小数部分：

乘基数 2	整数
0.8125*2=1.625	……..1
0.625*2=1.25	……..1
0.25*2=0.5	……..0
0.5*2=1	……..1

得：0.8125D=0.1101B

于是：325.8125D=101000101.1101B

3. 二进制数转换为八进制数或十六进制数

转换方法：由于八进制数、十六进制数和二进制数的基数成倍数关系，转换较为简单，方法是将二进制数从小数点开始分别向左向右每 3 位分成一组(转换成八进制数时)或每 4 位分成一组(转换成十六进制时)，不足 3 位(或 4 位)的补 0，然后写出对应的八进制数或十六进制数即可。

【例 2.3】 将 10110.11B 转换成十六进制数

$$10110.11B \rightarrow \frac{0001}{1}\ \frac{0110.1100}{6} = \frac{16\cdot CH}{C}$$

4. 八进制数或十六进制数转换为二进制数

转换方法：将每位八进制数写成对应的 3 位二进制数，每位十六进制数写成对应的 4 位二进制数即可。

二、计算机中数和字符的表示

(一)计算机中数的表示方法

计算机处理的数据通常是带符号数，即有正数和负数的区别，如+1101，+0.1101，-1101，-0.1101。在计算机中，正数与负数如何表示呢？为便于计算机识别与处理，通常用数的最高位来表示数的符号，0 表示正数，1 表示负数。日常用“+”或“-”表示符号的数叫真值，而在二进制数的最高位设置符号位，把符号加以数值化，这样的数叫机器数。例如：

真值	机器数
+1101	01101
-1101	11101

带符号数的机器数可以用原码、反码、补码 3 种不同码制来表示，由于补码表示法在加减运算中的优点，现在多数计算机都是采用补码表示法。微机系列机也是采用补码表示法。为此下面将对原码和补码分别进行介绍。

1. 原码表示法

原码是一种比较直观的机器数表示法。用二进制数的最高位表示符号(0 表示正数，1 表示负数)，数的有效值用二进制绝对值表示(与真值相同)。例如，原码表示的整数 01101010 和 11101010，分别对应的真值是+1101010 和-1101010。

在原码表示法中，8 位带符号二进制数能表示的最大数和最小数是 01111111 和 11111111，即-127 和+127。数 0 有两种形式：00000000 和 10000000，它们分别对应于+0 和-0。

原码表示法的机器数作加减法运算时不太方便。例如，要进行(-5)+7 的运算，看起来是作加法，但是两异号数相加实际是进行减法，即作 7-5 的运算。同理，两异号数相减时，实际是进行加法计算。所以对原码表示法的机器数进行加减运算时，不仅需要程序中指令规定的操作种类(加或减)，还要根据两数的符号确定实际的加减操作。加减操作后，要按照一定的规则确定运算结果的符号，例如两异号数相加，运算结果的符号应与绝对值较大的数同号，两异号数相减，运算结果的符号应与被减数同号。

2. 补码表示法

由于原码加减运算时不太方便，因此设想让符号位也作为数的一部分参与运算，使其运算操作简化，无须做过多的判断和处理。补码表示法就具有这一特点。

(1) 补码的定义。

带符号数 x 的补码表示法$[x]_{补}$可定义如下：

$$[x]_{补} = M + x$$

上述定义中，模数 M 根据机器数的位数而定，如 n=8，$M=2^8$。这个 2^8 正好是机器数(无符号数)产生进位而自动舍去的数。

若 X 是正数(即 X≥0)，按照上述定义，模数 M 和一个正数相加，作为溢出量便自动舍去。因此，正数的补码正好同原码相同。例如，真值 X=+00111011B(即+59D)，其补码表示：

$$[+59]_{补}=2^8+00111011=100000000+00111011$$
$$=|1|00111011$$

其中第 1 位自动舍去。

若 X 为负数(即 X＜0)，例如真值 X= -00111011B(即-59D)，其补码表示：

$$[-59]_{补}=2^8+(-00111011)=100000000-00111011$$
$$=11000101$$

从上述两个例子可以看出：用补码表示的机器数，符号位仍然表示数的符号(0 为正数，1 为负数)；对于正数，补码和原码一样，与真值的有效数等同；但对于负数，补码经过变换后，已是另一编码形式，它与真值的有效数已不能等同视之。

(2) 补码表示法中数的范围。

在补码表示法中，当 N=8 时，最大的正数仍是$[127]_{补}$ = 01111111，而数 0 只有一个，即 $[0]_{补}$=00000000，没有 +0 与-0 的区别。$[-127]_{补}$=10000001，而 10000000 却是$[-128]_{补}$，11111111 是$[-1]_{补}$。所以当 n=8 时，用补码表示数的范围是-128～+127，如表 2-2 所示。不难推导出，当 n=16 时，用补码表示数的范围是 -32768～+32767。

表 2-2　补码表示的数(n=8)

十进制数	十六进制数	二进制数
+127	7F	01111111
+126	7E	01111110
⋮	⋮	⋮
+3	03	00000011
+2	02	00000010
+1	01	00000001
0	00	00000000
−1	FF	11111111
−2	FE	11111110
−3	FD	11111101
⋮	⋮	⋮
−127	81	10000001
−128	80	10000000

(3) 由原码变换为补码。

由于正数的原码和补码的机器数一样，所以这里主要是讨论负数的变换。把一个负数的原码变换为补码的方法是：首先保持符号位不变(因为符号位已表示为负数)，然后将有效数各位变反，最低位加 1 即可。例如：

$[-59]_{原}$=10111011

符号位不变，其余各位变反：　11000100

最低位加 1：11000101

所以，　$[-59]_{补}$=11000101

再看一个例子：

设 X=−25 = −19H = −0011001B

则 X 的 8 位补码表示为：$[X]_{补}$=11100111B = E7H

X 的 16 位补码表示为：　$[X]_{补}$=1111111111100111B = FFE7H

从这个例子可以看出 X 的 16 位补码实际上是其 8 位补码的符号扩展。由此得出一个重要结论：一个二进制补码数的符号位(最高位)向左扩展若干位后，仍是该数的补码。

(二)二进制编码

1. 十进制数的二进制编码(BCD 码)

8086/8088 指令支持十进制数的运算，那么十进制数在机器内部也必须用二进制表示，即用十进制数的二进制编码表示。常用的是 BCD 码。BCD 码与十进制数的对应关系如表 2-3 所示。