21世纪高等学校规划教材 | 计算机科学与技术

汇编语言程序设计

刘辉 王勇 徐建平 编著

清华大学出版社

汇编语言程序设计

刘辉 王勇 徐建平 编著

清华大学出版社
北京

内 容 简 介

本书通过大量例子详细生动地介绍了16位汇编语言的基本知识、程序结构及上机操作和调试步骤。在介绍程序结构时，融合了基本语句语法知识的介绍及相关指令的分析。

本书共7章，分别介绍汇编语言程序的结构形式、常用的各种伪指令和上机操作步骤、与汇编语言程序相关的硬件知识、汇编语言程序中用到的各种指令、程序的结构、子程序结构和参数的传递方法、宏汇编的知识等。

本书可作为高等院校计算机、软件工程专业及信息安全专业本科生、研究生的教材，也可为广大汇编爱好者及广大科技工作者和研究人员提供参考。

图书在版编目(CIP)数据

汇编语言程序设计/刘辉，王勇，徐建平编著. --北京：清华大学出版社，2014
21世纪高等学校规划教材·计算机科学与技术
ISBN 978-7-302-37525-8

Ⅰ. ①汇… Ⅱ. ①刘… ②王… ③徐… Ⅲ. ①汇编语言－程序设计 Ⅳ. ①TP313

中国版本图书馆CIP数据核字(2014)第170941号

责任编辑：刘向威　赵晓宁
封面设计：傅瑞学
责任校对：时翠兰
责任印制：宋　林

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，010-62795954
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：11.75　　**字　　数**：283千字
版　　次：2014年10月第1版　　**印　　次**：2014年10月第1次印刷
印　　数：1～2000
定　　价：29.00元

产品编号：059930-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程（简称‘质量工程’）”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻性。教材由个人申报或各校推荐（通过所在高校的“编委会”成员推荐），经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括：

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人：魏江江

E-mail：weijj@tup.tsinghua.edu.cn

前言

随着现代软件系统越来越庞大复杂，大量经过了封装的高级语言也应运而生。这些高级语言使软件开发人员在开发过程中能够快速、高效地进行编码，从而能够从复杂的编码中解放出来，而专注于程序逻辑的实现。汇编语言是面向机器指令的低级语言，由于其复杂性使得其适用领域逐步减小。那么是不是汇编语言已经无用武之地了呢？是不是我们就不需要学习这门语言了呢？答案是否定的。由于汇编更接近机器语言，能够直接对硬件进行操作，生成的程序与其他语言相比具有更高的运行速度，占用更小的内存空间，因此在一些对于时效性和执行效率要求很高的程序，以及许多大型程序的核心模块，尤其是工业控制中对硬件操作的底层代码，都还要求助于汇编语言。另外，由于高级语言最终都要翻译为机器语言才能被处理器执行，而汇编语言非常接近于机器语言，通过学习汇编语言，软件开发人员可以更清楚地理解能够被处理器执行的机器指令，这对于透彻掌握一门高级语言的底层逻辑也有很大的帮助。所以，各高等院校的计算机科学类专业仍把汇编语言作为学生的必修课，以让学生深入了解计算机的运行原理，为深入理解高级语言的编程打下坚实的基础。

汇编语言作为最接近硬件的计算机编程语言，它既有对硬件直接编程的便利，又有接近于人类自然语言的指令，所以学习汇编语言需要一定的硬件基础知识，严密的思维逻辑和良好的编程习惯。学习汇编语言的难点，在于很多指令的执行需要事先设置默认的寄存器参数。在学习时，要注重各种指令的执行要求，明确默认的参数设置，正确使用各条指令。

本书是编者经过多年的教学总结，把汇编语言的基础教学内容基于学生能快速掌握的原则进行了合理编排整理而成的。王勇编写第 1 章，徐建平编写第 2 和第 7 章，刘辉编写第 3～第 6 章，全书由王勇负责审阅。在讲课安排上，可以把第 3 章的内容分散到第 4～第 6 章中，结合例题讲解；上机实验操作，可以根据讲课内容安排相应的编程操作，刚开始时可以让学生调试书上的例题，让学生掌握上机操作的步骤；基本步骤熟练后让学生自己编程，具体题目可以参照每章的上机实验题目。

本书的配套资源有课程课件、习题答案和例题的源程序。例题源程序的编号，以 ex 开头，如例 4.1 的源程序为 ex401.asm；习题中的编程题目的源程序，以 test 开头，如习题 5.3，源程序为 test53.asm；源程序和上机操作使用的编译软件在 sourceasm 文件夹中。使用中有任何建议和疑问，可与编者联系，Email：hlliuhui@sina.com。

目 录

第1章 汇编语言程序基本知识

1.1 汇编语言程序的结构形式

计算机语言的发展经历了机器语言、汇编语言到高级语言的发展过程。机器语言是使用0和1书写的二进制代码，难于书写和纠错；汇编语言使用接近于人类的语言对计算机的硬件直接发号施令，让内部的各个部件直接进行各种运算；高级语言程序的书写更简单，但是各个函数之间的参数传递比较复杂，逻辑结构性强。

汇编语言程序的结构形式主要有简短模式定义和完整模式定义，程序的书写要用到各种伪操作，这些伪操作类似于高级语言里面的语句。

下面通过一个例子来介绍汇编语言和高级语言的区别。

【例 1-1】 编程实现C＝A＋B，并在屏幕上显示出结果。

算法分析：定义存放加数和被加数值的变量A，B；给A，B以确定的值；实现A＋B的操作并把结果存放在变量C中；输出运算结果。分别用高级语言C++和汇编语言编写的代码如下：

```
/*EX101A.CPP编程实现C=A+B,并在屏幕上显示出结果,用C++实现*/
#INCLUDE "STDAFX.H"
#INCLUDE "STDIO.H"
INT MAIN(INT ARGC,CHAR*ARGV[])                    /*程序从主函数开始*/
    {INT  A,B,C;                                  /*定义变量*/
    A=1;                                          /*直接给变量赋值*/
    B=2;
    C=A+B;                                        /*计算累加结果*/
    PRINTF("C= %D\N",C);                          /*输出结果*/
    RETURN  0;
}
;EX101B.ASM编程实现C=A+B,并在屏幕上显示出结果,用汇编语言实现
DATA  SEGMENT                                     ;定义数据段
    A          DB    ?                          ;定义变量
    B          DB    ?
    C          DB    ?
    STRING     DB    'C= $'
DATA    ENDS
```

```
CODE    SEGMENT                                    ;定义代码段
MAIN    PROC FAR                                   ;主程序从此开始
ASSUME CS:CODE,DS:DATA,ES:DATA
START:
        PUSH    DS
        SUB     AX,AX
        PUSH    AX
        MOV     AX,DATA
        MOV     DS,AX                              ;数据段的地址装入专用寄存器
        MOV     ES,AX
        MOV     A,1                                ;给变量赋值
        MOV     B,2
        MOV     AL,A
        ADD     AL,B                               ; A+B
        MOV     C,AL                               ;运算结果存入 C 变量中
        LEA     DX,STRING
        MOV     AH,09
        INT     21H                                ;输出字符串
        ADD     C,30H                              ;整数转化为字符,因为汇编输出都是字符
        MOV     DL,C
        MOV     AH,2                               ;输出 DL 中字符,这是 21 号中断的功能调用
        INT     21H
        MOV     DL,0AH                             ;输出换行符
        INT     21H
        MOV     DL,0DH                             ;输出回车符
        INT     21H
        RET
   MAIN         ENDP
CODE            ENDS
   END          START
```

汇编语言与高级语言的比较：

- 高级语言书写简单，不需知道硬件的详细操作过程，易于掌握。
- 汇编语言需要程序员定义变量的存放位置，直接对硬件进行编程，需要对硬件进行详细的设计，所以有一定的难度。

【例 1-2】 在屏幕上显示字符串“This is an assembly language program!”

题目分析：

(1) 字符串应存放在一个存储单元中，即放在一个变量中，这要在数据区中定义。

(2) 在代码段中，首先把程序中用到的各个段与相应的寄存器名对应起来，要用到 assume 伪操作。

(3) 调用 DOS 中断显示字符串，中断执行前先做显示的准备操作：数据段的地址存入 DS 寄存器；从数据区的存储单元中取出要显示的字符串的存放地址并存入 DX 寄存器；执行中断操作，显示 DS:DX 中的内容

汇编程序如下：

```
;EX102.ASM 用汇编语言输出一个字符串
DATA SEGMENT                        ;定义数据段
   STR DB 'this is an               ;在 STR 存储单元中的字符串内容
```

```
assembly language
program! ',' $ ',13,10
DATA ENDS                          ;定义代码段
CODE SEGMENT                       ;主程序从此开始
MAIN PROC FAR                      ;指派程序中实际定义的各个段与对应寄存器的联系
ASSUME CS:CODE,DS:DATA

START:                             ;语句标号
  PUSH DS                          ;保护原有的数据段内容到堆栈段
  SUB AX,AX                        ;存储一个 0 值,表示新程序的数据开始存放
  PUSH AX
  MOV AX,DATA                      ;先把数据段的地址临时存入 AX 寄存器中
  MOV DS,AX                        ;再把地址存入数据段寄存器中

  LEA DX,STR                       ;取出要显示的字符串的偏移地址存入 DX 寄存器
  MOV AH,09H                       ;调用 DOS 中断,显示 DS:DX 中的内容
  INT 21H
EXIT:                              ;主要功能执行完毕,返回 DOS 界面
  MOV AX,4C00H
  INT 21H
MAIN ENDP                          ;主程序到此结束
 CODE ENDS                         ;代码段到此结束
    END START                      ;汇编程序到此结束,与前面的 START 相对应
```

以上两个程序都是用汇编的完整模式定义各个段,下面用简短模式定义各个段。

【例 1-3】 求两个数中的最大值。

题目分析:

(1) 先在数据区中定义两个变量并赋值。

(2) 在程序中比较这两个变量的大小,把较大的值存放在存储单元 MAX 中。

程序如下:

```
; EX103.ASM 简短模式书写的程序
.MODEL TINY                        ;定义程序模型
.DATA                              ;定义数据段
    X DB -8                        ;定义变量名称及具体数值
    Y DB 10
    MAX DB ?                       ;只分配存储空间,没有值
.CODE
STARTUP:                           ;语句标号
    MOV AX,@DATA                   ;预定义符@DATA 可以取出数据段的段名
    MOV DS,AX
    MOV AL,X                       ;把 X 的值预存入 AX 寄存器的低 8 位 AL 中
    MOV CL,Y                       ;把 Y 的值预存入 CX 寄存器的低 8 位 CL 中
    CMP AL,CL                      ;比较 AL 和 CL 寄存器的数值大小
    JGE BIG                        ;如果 AL≥CL,则跳转到 BIG 标号处
                                   ;否则,即 AL<CL,顺序执行下述语句
    MOV MAX,CL                     ;把较大值 CL 存入 MAX 单元中
    JMP EXIT
   BIG:
    MOV MAX,AL                     ;把较大值 AL 存入 MAX 单元中
```

```
EXIT:
 MOV AX,4C00H                          ;返回 DOS 界面
 INT 21H
END  STARTUP
```

例 1-2 和例 1-3 分别用完整的段定义和简短模式的段定义，可以看出，具体的执行代码是一样的，区别就在于段定义的开始和结束部分不同。

通过上述例子，大致了解汇编语言程序的构成，了解程序的书写形式及常用的几个语句，后面详细介绍汇编程序的各种指令格式及使用方法。汇编程序由代码段、数据段等构成，这些都需要程序员明确指定。

1.2 汇编语言的各种伪指令

1.2.1 数据定义伪指令

数据定义伪指令的用途是定义一个变量的类型，给变量赋初值，或仅仅给变量分配存储单元，而不赋予特定的值。数据定义伪指令有 DB、DW、DD、DF、DQ、DT 等，而常用的是前三种。

数据定义伪指令的一般格式为：

[变量名]　伪指令定义符　操作数[,操作数…]

其中，方括号中的变量名为任选项，变量名后面不跟冒号。伪指令定义符后面的操作数不止一个。如有多个操作数，相互之间应该用逗号分开。

1. DB (Define Byte)

定义变量的类型为字节(Byte)，给变量分配字节或字节串。DB 伪指令定义符后面的操作数每个占有 1 字节。

2. DW (Define Word)

定义变量的类型为字(Word)。DW 伪指令定义符后面的操作数每个占有 1 个字，即 2 字节。在内存中存放时，低位字在前，高位字在后。

3. DD (Define Double Word)

定义变量的类型为双字(DWORD)。DD 后面的操作数每个占 2 个字，即 4 字节。在内存中存放时，低位字在前，高位字在后。

数据定义伪指令定义符后面的操作数可以是常数、表达式或字符串，但每项操作数的值不能超过由伪指令定义符所定义的数据类型限定的范围。例如，DB 伪指令定义数据的类型为字节，则其范围为无符号数 0～255；带符号数－128～＋127 等。字符串必须放在单引号中。另外，超过两个字符的字符串只能用 DB 伪指令定义。

【例 1-4】 数据定义伪操作的使用。

```
1 DATA   DB   1,2H          ;以 DATA 命名的存储单元中存入 1H、2H,每个数据占 1 字节
2 EXPR   DW   1,2            ;以 EXPR 命名的存储单元中存入 1 和 2,每个数据占一个字,即 2 字节
3 STR    DB   'WELCOM!'     ;以STR 命名的存储单元中存入字符串"WELCOM!",每个字符占 1 字节,字符从
                              左到右依次存储在地址增高的空间中,以字符的 ASCII 码值存储
4 S1     DW   'AB'          ;以 S1 命名的存储单元中存入字符 AB,A 存放在高字节,B 在低字节,如果一
                              个字的空间足够容纳两个字符,则存放在一个字中,所以共占 2 字节
5 S2     DD   'AB'          ;以 S2 命名的存储单元中存入字符 AB,如果一个双字的空间足够容纳两个
                              字符,则存放在一个双字中,所以共占 4 个字节,即存入 42H,41H,00,00
6 OFFAB  DW   S1            ;存入变量 S1 的偏移地址
```

该例子中定义的变量在内存中的存储形式如表 1-1 所示。

表 1-1　数据在内存中的存储

存储单元名称	内存中的数据	内存的地址
⋮	⋮	⋮
DATA	01	0100H
	02	0101H
EXPR	01	0102H
	00	0103H
	02	0104H
	00	0105H
STR	57	0106H
	45	0107H
	4c	0108H
	43	0109H
	4F	010AH
	4D	010BH
	21	010CH
S1	42	010DH
	41	010EH
S2	42	010FH
	41	0110H
	0	0111H
	0	0112H
OFFAB	0D	0113H
	01	0114H
⋮	⋮	⋮

以上第 1 和第 2 句中，分别将常数以字节或字的形式赋予一个变量；第 3 句的操作数是包含 8 个字符的字符串(只有 DB 伪指令才能用)；在第 4 和第 5 句，注意伪指令 DW 和 DD 的区别，虽然操作数均为'AB'两个字符，但存入变量的内容各不相同，第 6 句的操作数是变量 S1，而不是字符串，此句将 S1 的 16 位偏移地址存入变量 OFFAB。

除了常数、表达式和字符串外，问号“?”也可以作为数据定义伪指令的操作数，此时仅给变量保留相应的存储单元，而不赋予变量某个确定的初值。

4. DUP

当同样的操作数重复多次时，可用重复操作符 DUP 表示，其形式为：

```
n DUP(初值[,初值,…])
```

其中，小括号中为重复的内容，n 为重复次数。如果用“n DUP(?)”作为数据定义伪指令定义符的唯一操作数，则汇编程序产生一个相应的数据区，但不赋任何初值。重复操作符 DUP 可以嵌套。表 1-2 所示是用问号和 DUP 表示操作数的几个例子。

表 1-2 DUP 的使用

语　句	说　明
FILLER DB ?	给字节变量 FILLER 分配存储单元，但不赋予特定的值
BUFFER DB 10 DUP(?)	给变量 BUFFER 分配 10 字节的存储空间，但不赋任何初值
ZERO DW 30 DUP(0)	给变量 ZERO 分配一个数据区，共 30 个字(即 60 字节)，每个字的内容均为 0
MASK DB 5 DUP('OK!')	定义一个数据区，存储 5 个重复的字符串'OK!'
ARRAY DB 100 DUP(3 DUP(8),6)	将变量 ARRAY 定义为一个数据区，其中包含重复 100 次的内容：8,8,8,6，共占 400 个字节

通常把用 DUP 作为唯一操作数而定义的变量称为数组。

1.2.2 符号定义伪指令

符号定义伪指令的用途是给一个符号重新命名，或定义新的类型属性等。符号包括汇编语言的变量名、标号名、过程名、寄存器名以及指令助记符等。

常用的符号定义伪指令有 EQU、=(等号)和 LABLE。

1. EQU

格式：

```
名字  EQU  表达式
```

EQU 伪指令将表达式的值赋予一个名字，以后可用这个名字来代替上述表达式。格式中的表达式可以是一个常数、符号、数值表达式或地址表达式等。

例如：

```
CR  EQU   0DH              ; 常数
A   EQU   ASCII_TABLE      ; 变量
STR EQU   64 * 1024        ; 数值表达式
ADR EQU   ES: [BP + DI + 5] ; 地址表达式
CBD EQU   AAM              ; 指令助记符
```

利用 EQU 伪指令，可以用一个名字代表一个数值，或用一个较简短的名字来代替一个较长的名字。

如果源程序中需要多次引用某一表达式，则可以利用 EQU 伪指令定义符给其赋一个

名字，以代替程序中的表达式，从而使程序更加简洁，便于阅读。将来如果改变表达式的值，也只需修改一处，使程序易于维护。需要注意的是，EQU 伪指令不允许对同一符号重复定义。

2. =（等号）

格式：

```
名字 = 表达式
```

=（等号）伪指令的功能与 EQU 伪指令基本相同，主要区别在于它可以对同一个名字重复定义。

例如：

```
COUNT = 100
MOV   CX,COUNT      ; (CX)←100
COUNT = COUNT - 10
MOV   BX,COUNT      ; (BX)←90
```

3. LABLE

格式：

```
名字   LABLE   类型
```

LABLE 伪指令的用途是定义标号或变量的类型。变量的类型可以是 BYTE、WORD、DWORD 等；标号的类型可以是 NEAR 或 FAR。

利用 LABEL 伪指令可以使同一个数据区兼有 BYTE 和 WORD 两种属性，这样在以后的程序中可根据不同的需要分别以字节或字为单位存取其中的数据。

例如：

```
AREAW   LABEL   WORD        ; 变量 AREAW 的类型为 WORD
AREAB   DB 100  DUP(?)      ; 变量 AREAB 的类型为 BYTE
MOV     AREAW,AX            ; AX 送第 1 和第 2 字节中
MOV     AREAB[49],AL        ; AL 送第 50 字节中
```

LABEL 伪指令可以将一个属性已经定义为 NEAR 或后面跟有冒号（隐含属性为 NEAR）的标号再定义为 FAR。

例如：

```
AGAINF LABEL   FAR          ; 定义标号 AGAINF 的属性为 FAR
AGAIN: PUSH    AX           ; 定义标号 AGAIN 的属性为 NEAR
```

上面的过程既可以利用标号 AGAIN 在本段内被调用，也可以利用标号 AGAINF 被其他段调用。

1.2.3　段定义伪指令

段定义伪指令的用途是在汇编语言源程序中定义逻辑段。常用的段定义伪指令有 SEGMENT/ENDS 和 ASSUME 等。