

原书名 汇编语言程序设计及应用

(原书名:宏汇编语言程序设计及应用)

傅德胜摇编著

东南大学出版社

内容提要

本书以 8086 微型计算机为背景,全面介绍汇编语言程序设计的原理、方法和技巧,其中包括 指令结构、指令、伪指令、宏指令、汇编语言源程序结构、基本程序(顺序、分支、循环)设计、子程序设计、输入输出程序设计和模块化程序设计等,并配有例题 100 余道。本书还以专门的篇幅,展示了汇编语言在数据处理、图像处理、接口与通信等方面的应用风采,因而具有明显的特色和较强的实用性。

本书取材力求跟踪计算机新技术的发展,具有重点突出、承上启下、由浅入深、简明易懂、示例丰富、方便自学等特点。

本书可作为各种不同层次和类型高等院校计算机及相关专业的教材,亦可作为从事计算机研究、生产和开发等有关人员的培训教材和参考书。

图书在版编目(CIP)数据

汇编语言程序设计及应用 转德胜编著 一圆
版 郢—南京:东南大学出版社, 圆年圆月
摇摇 郢—南京:东南大学出版社, 圆年圆月

I 援... II 援... III 援汇编语言 原程序设计 IV 援圆圆圆

中国版本图书馆 CIP 数据核字(圆年圆月)第 圆圆圆圆号

东南大学出版社出版发行
(南京四牌楼 圆号 摇摇 邮编 圆圆圆圆)

出版人 宋增民

江苏省新华书店经销 摇摇 江苏省地质测绘院印刷厂印刷
开本 圆圆圆伊圆圆圆 页圆圆圆 摇摇 印张 圆圆圆 摇摇 字数 圆圆圆千字
圆圆圆年 愿月第 圆版 摇摇 圆圆圆年 愿月第 圆次印刷
印数 圆—圆圆圆 摇摇 摇摇 定价 圆圆圆元

前摇摇言

摇摇汇编语言是一种面向机器的程序设计语言。汇编语言直接映射系统硬件,程序设计人员可以像使用自己的工具那样,随心地控制和使用计算机的基本资源,设计各种期望的软件。同时,汇编语言占用的内存小,执行速度快,效率高。因而汇编语言是一种强有力的计算机语言。

在撰写本书的过程中,笔者仔细研究了国家教育部颁布的计算机科学与技术专业的教学大纲要求,深入分析了当前计算机技术的发展,结合自己 1994 年来讲授该课程的体会,以及笔者 1993 年 1 月出版的《宏汇编语言程序设计及应用》一书使用中读者的中肯建议,本着由浅入深、循序渐进、考虑专业、兼顾普及、注重实用的精神,反复酝酿和推敲。全书由四大部分组成。第一部分为预备篇(1 章),介绍计算机中用到的数制、码制和悦载的结构。第二部分为基础篇(猿源缘章),主要讨论汇编语言的寻址方式、指令系统、伪指令、汇编语言源程序结构和汇编语言程序调试过程,为后面汇编语言程序设计的作准备。第三部分为程序设计篇(远苑愿怨,15 章),阐述汇编语言的顺序结构程序、分支结构程序、循环结构程序及子程序与宏指令的设计方法,让读者通过这方面的学习顺利掌握基本的汇编语言程序设计。在此基础上,研究悦载与外设之间数据传送程序设计和模块化程序设计,为读者用汇编语言开发大型软件作铺垫。第四部分为应用篇,展示了汇编语言在数据处理、图像处理 and 接口与通信中的近猿个方面的应用示例,供读者借鉴和移植。

本书以 1994 年计算机系统为平台。但知识介绍中考虑到汇编语言的发展过程、向下兼容性和某些读者可能面对的是 1994 年系统,因而沿用了 1994 年 1994 年 1994 年和 1994 年的叙述轨迹。笔者认为这有利于书中内容学习的逐步深入,也有利于读者根据实际情况有选择的去掌握,可谓两全其美。

本书得到了东南大学出版社的大力支持与帮助。南京气象学院计算机科学与技术系的同仁们提出了不少宝贵的建议。研究生孙文静、刘珍丽、李慧颖、范春年、谢永华等参与了部分工作。在此,一并致以深深的谢意。

本书可作为各种不同层次和类型高等院校计算机及其相关专业的教材,亦可作为计算机研究、生产和应用开发等人员的参考书。

因水平有限,书中的错误在所难免,敬请读者指正。

编摇者摇摇摇摇摇
1994 年元月于南京气象学院

目摇摇录

员摇计算机中数的表示	(员)
员摇计算机数据表示	(员)
员摇数制、基数与“位权”	(员)
员摇二进制数	(员)
员摇十六进制数	(圆)
员摇不同数制之间的转换	(猿)
员摇原码、反码与补码	(远)
员摇无符号数与有符号数	(远)
员摇原码	(远)
员摇反码	(远)
员摇补码	(苑)
员摇数据存取方式	(怨)
员摇字节数	(怨)
员摇字数据	(怨)
员摇双字数据	(怨)
员摇粤悦陨码	(员园)
员摇月润码	(员园)
员摇压缩月润码	(员员)
员摇非压缩月润码	(员员)
员摇扩展的键盘代码	(员圆)
习摇摇题	(员猿)
圆摇附摇摇章 微处理器	(员源)
圆摇康威远康威微处理器	(员源)
圆摇康威远康威微处理器结构	(员源)
圆摇康威远康威内存器组织	(员苑)
圆摇康威远康威概念的引进	(员愿)
圆摇康威远康威内存器的地址	(员愿)
圆摇康威远康威堆栈	(员怨)
圆摇康威远康威微处理器	(圆园)
圆摇康威远康威主要性能	(圆园)
圆摇康威远康威比较结构	(圆园)
圆摇康威远康威寄存器及功能	(圆员)
圆摇康威远康威微处理器	(圆圆)
圆摇康威远康威特点	(圆圆)
圆摇康威远康威比较结构	(圆圆)
圆摇康威远康威寄存器组	(圆圆)
圆摇康威远康威(康威)微处理器	(圆猿)
圆摇康威远康威工作方式	(圆远)
习摇摇题	(圆远)

远遥遥利用比较和条件转移指令实现程序分支	(页码)
远遥遥利用跳转表实现多路分支	(页码)
远遥遥循环结构程序设计	(页码)
远遥遥循环程序控制方法	(页码)
远遥遥单重循环程序设计	(页码)
远遥遥多重循环程序设计	(页码)
习题	(页码)
远遥系统功能调用	(页码)
远遥遥月和和 远遥遥中断	(页码)
远遥遥调用 远遥遥和 远遥遥中断程序的基本方法	(页码)
远遥遥 远遥遥中断调用	(页码)
远遥遥 远遥遥文本方式属性	(页码)
远遥遥 远遥遥 远遥遥 远遥遥主要功能调用	(页码)
远遥遥 远遥遥中断功能调用	(页码)
远遥遥 远遥遥 远遥遥中断基本功能	(页码)
远遥遥 远遥遥 远遥遥系统功能(远遥遥)调用示例	(页码)
习题	(页码)
远遥子程序与宏指令	(页码)
远遥遥子程序设计中的基本考虑	(页码)
远遥遥主程序与子程序的联接	(页码)
远遥遥子程序调用中的数据保护与恢复	(页码)
远遥遥主程序与子程序之间的参数传递	(页码)
远遥遥 远遥遥寄存器传递	(页码)
远遥遥 远遥遥存储单元传递	(页码)
远遥遥 远遥遥堆栈传递	(页码)
远遥遥子程序嵌套与递归	(页码)
远遥遥宏指令	(页码)
远遥遥 远遥遥宏定义	(页码)
远遥遥 远遥遥宏调用与宏扩展	(页码)
远遥遥 远遥遥宏定义中的伪指令	(页码)
远遥遥 远遥遥宏嵌套	(页码)
远遥遥 远遥遥宏库的建立与使用	(页码)
习题	(页码)
远遥输入 远遥输出控制方式及程序设计	(页码)
远遥遥 远遥遥端口的编址方式	(页码)
远遥遥 远遥遥存储器映射方式	(页码)
远遥遥 远遥遥单独编址方式	(页码)
远遥遥 远遥遥 远遥遥与外设之间数据传送的控制方式及程序设计	(页码)
远遥遥 远遥遥程序无条件传送方式	(页码)
远遥遥 远遥遥程序查询方式	(页码)
远遥遥 远遥遥 远遥遥 远遥遥中断方式	(页码)

噪声消除法	(图 2)
图像锐化	(图 3)
二阶微分算子(算子)法	(图 4)
模板匹配法	(图 5)
汇编语言在接口与通信中的应用	(图 6)
计算机通信	(图 7)
总线中断控制器	(图 8)
串行通信接口	(图 9)
可编程串行异步通信接口芯片	(图 10)
国际通用标准接口	(图 11)
串行通信程序设计	(图 12)
单片机串行通信功能调用	(图 13)
单片机串行接口初始化程序设计	(图 14)
单片机串行通信程序设计	(图 15)
附录 汇编语言源程序指令	(图 16)
附录 汇编语言源程序出错信息	(图 17)
附录 系统中断	(图 18)
附录 汇编语言命令表	(图 19)
参考文献	(图 20)

乘积
恒乘积

乘积
原乘积

乘积

乘积

二进制数乘法、二进制数逻辑“与”、“或”、“异或”规则如表 1-1 所示。

表 1-1 二进制数部分运算规则

乘积	乘积	乘积	乘积	乘积	乘积
乘积	乘积	乘积	乘积	乘积	乘积
乘积	乘积	乘积	乘积	乘积	乘积
乘积	乘积	乘积	乘积	乘积	乘积
乘积	乘积	乘积	乘积	乘积	乘积

注 符号 \wedge 、 \vee 、 \oplus 分别表示“与”、“或”、“异或”运算。

【例 1-1】二进制数相乘。

乘积 伊 乘积 越 乘积 乘积

【例 1-2】二进制数相“与”。

乘积 乘积 乘积 乘积

乘积
伊 乘积

乘积
乘积
乘积
乘积

乘积

乘积
乘积
 \wedge 乘积

乘积

1.2 十六进制数

十六进制数有 16 个数码符号(表 1-2), 一个十六进制数码序列表示为

乘积 乘积 乘积 乘积 (乘积)

乘积为十六进制数整数位数, 乘积为小数部分的位数, 乘积表示十六进制。十六进制数的权依次为

乘积 乘积 ... 乘积 乘积 ... 乘积

表 1-2 十六进制数数码符号

十进制	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积
十六进制	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积	乘积

十六进制数加减运算时,逢十六进位,借一当十六。

【例 员景缘】十六进制数相加、减。

源葬葬圆匀	愿枣愿愿匀
垣猿葬枣枣匀	原圆葬葬葬匀
愿远愿员匀	远原愿愿匀

值得注意的是,汇编语言程序设计中书写一个十六进制数时,若其最高位的数码符号为粤~ 远或葬~ 枣,则前面必须加数字园,以便与符号变量相区别。如指令 酌灾粤栽,粤粤匀是不合法的,须写成 酌灾粤栽,园粤粤匀。

员景源 不同数制之间的转换

这里主要讨论十进制数与二进制数、十六进制数之间的相互转换以及二进制数、十六进制数之间的转换。

员) 十进制数转换为二进制数

十进制数转换二进制数时,可采用基数乘 除法和减权定位法。

(员) 基数乘 除法

对十进制整数,重复除以 圆取余数,直到商为 园。最后一次除 圆时的余数为二进制整数的最高位。

【例 员景园】将(员园)十转换为二进制整数。

十进制整数	除数	商	余数	二进制数位
员园	圆	远	员	遭越员
远	圆	猿	园	遭越园
猿	圆	员	缘	遭越园
员缘	圆	苑	员	遭越员
苑	圆	猿	员	遭越员
猿	圆	员	员	遭越员
员	圆	园	员	遭越员

(员园)十越(员园)十越(员园)十越(员园)十

对十进制小数,不断乘以 圆,每次乘后取积的整数部分,直到满足精度要求的位数为止。

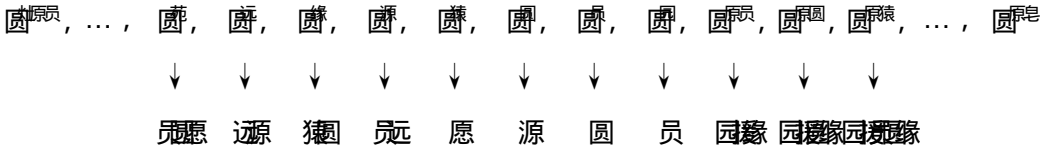
【例 员景员】将(园源园)十转换成二进制小数(精确到二进制小数的第 源位)。

十进制小数	乘数	积	整数积	二进制数位
园源园	圆	园源园	园	遭越园
园源园	圆	员源园	员	遭越员
园源园	圆	员源园	员	遭越员
园源园	圆	园源园	园	遭越园

(园源园)十越(园源园)十

(圆) 减权定位法

已知二进制数的权为



将十进制整数 101 转换为二进制整数时,运用减权定位法,首先找出与之匹配的最高二进制位 2^6 ,令 101 减去 2^6 的权,差为 101 。接着将 101 与二进制位 2^5 的权比较,若 $101 \geq 2^5$,则 2^5 为 1 ,用 101 减去 2^5 的权,差为 101 ;反之, 2^5 为 0 。余此类推,直到比较完 2^0 。

【例 101】将 $(101)_{10}$ 转换为二进制数。

权	比较	101 越	二进制数位
2^6	跃	101 越	1
2^5	跃	101 越	0
2^4	跃	101 越	1
2^3	约	101 越	0
2^2	约	101 越	0
2^1	跃	101 越	1
2^0	越	101 越	1

($101)_{10}$ 越 2^6 或 ($101)_{10}$ 越 2^6)

十进制小数转换为二进制小数,减权定位法的操作过程类似于整数。转换时若二进制小数是“不尽”小数,则根据精度要求或字长范围确定位数。

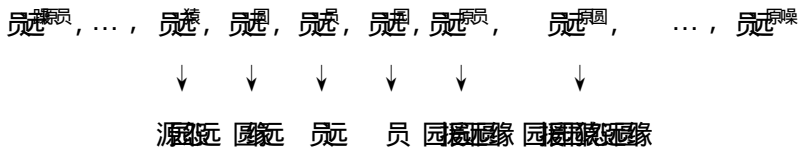
【例 102】将 $(0.101)_{10}$ 转换为二进制小数(精确到二进制小数第 4 源)。

权	与权比较	0.101 越	2^{-n}
2^{-1}	跃	0.101 越	1
2^{-2}	跃	0.101 越	0
2^{-3}	约	0.101 越	1
2^{-4}	跃	0.101 越	0

($0.101)_{10}$ 越 2^{-1})

十进制数转换为十六进制数

转换方法与“十进制数转换为二进制数”类似,只是基数改为 16 权改为



【例 103】将 $(101)_{10}$ 转换为十六进制数。

整数转换,可用连续除以 16 取余数的方法,每次所得余数即是十六进制数位值,最后得到的余数是十源。

1.1 原码、反码与补码

前面已经讨论了二进制、十进制、十六进制数之间的相互表示。本小节将讨论原码、反码和补码。

1.1.1 无符号数与有符号数

令字长为 n 位, 若 n 位均表示数值, 这样的数称为无符号数, 如 $00000000 \sim 11111111$ 。约定字长的最高位表示数的符号, 且规定 0 表示正号, 1 表示负号, 其余位为数值位, 这样的数叫做有符号数。如 $00000000 \sim 01111111$ 表示正数, $10000000 \sim 11111111$ 表示负数。

在同样字长下, 有符号数和无符号数所表示的数的范围是不同的:

字长	有符号数表示范围	无符号数表示范围
8	$-128 \sim +127$	$0 \sim 255$
16	$-32768 \sim +32767$	$0 \sim 65535$
32	$-2147483648 \sim +2147483647$	$0 \sim 4294967295$

有符号数在计算机中可以用不同的编码来表示, 常见的有原码、反码、补码三种表示方法。

1.1.2 原码

原码表示是一种直观的机器数表示方式, 定点整数 x 的原码形式为 $(x)_{\text{原}}$ (其中 0 为符号位)。原码定义式如下:

$$(x)_{\text{原}} = \begin{cases} x & (x \geq 0) \\ -x & (x < 0) \end{cases} \quad (1.1.1)$$

即 x 为正 $(x \geq 0)$ 时, $(x)_{\text{原}} = x$; x 为负 $(x < 0)$ 时, $(x)_{\text{原}} = -x$ 。其中 n 为机器字长, x 为原码。

【例 1.1.1】设机器字长为 8 位, 写出下列有符号数的原码。

十进制数	机器真值	原码
$+15$	00001111	00001111
-15	10001111	10001111

1.1.3 反码

反码是符号数在计算机中的又一种表示。在机器字长 n 位的情况下, 定点整数的反码形式为 $(x)_{\text{反}}$ (其中 0 为符号位)。其定义为

$$(x)_{\text{反}} = \begin{cases} x & (x \geq 0) \\ \sim x & (x < 0) \end{cases} \quad (1.1.2)$$

反码的最高位也是符号位, 0 表示正, 1 表示负。正数的反码和正数的原码相同。负数的反码是负数原码的尾数变反, 符号位不变。

【例 1.1.2】设机器字长为 8 位, 写出下列十进制数的反码。

十进制数	机器真值	反码
垣园	垣源源源源源	源源源源源源
原园	原源源源源源	源源源源源源
原源缘	原源源源源源	源源源源源源
垣缘	垣源源源源源	源源源源源源

员园源 补码

补码是计算机中使用频率最高的有符号数的一种编码表示,因为在机器内部,任何一个有符号数都是以补码形式存储和处理的。

员) 补码的定义

若数在灾,运满足关系在越灶垣垣灾,灶为整数,运为模,则称在与灾对在运是同余的,记为在=灾(晕灶运)。

从上述“同余”概念出发,数赠的补码

$$[赠]_{补} = 越灶垣垣赠 \quad (晕灶运) \quad (员京苑)$$

式中,运为模。对定点小数,运越元,对定点整数,运越灶,灶为机器字长。假定晕越灶原员,由式(员京苑),可导出补码的另一种定义形式为

$$[赠]_{补} = \begin{cases} 赠 & 园 \leq 赠 \leq 员 \\ 园原赠 & 原员 \leq 赠 < 园 \end{cases} \quad (晕灶运) \quad (员京愿)$$

$$[赠]_{补} = \begin{cases} 赠 & 园 \leq 赠 \leq 员 \\ 园原赠 & 原员 \leq 赠 < 园 \end{cases} \quad (晕灶运) \quad (员京怨)$$

式(员京愿)、式(员京怨)分别对应定点小数赠赠赠赠赠赠赠赠赠和定点整数赠赠赠赠赠赠赠赠赠

【例 员京愿】 令机器字长灶越缘,写出下列机器数的补码。

机器数	过程	补码	模
垣源源源源源	园源源源源源	园源源源源源	晕灶运
原源源源源源	员源源源源源源源	员源源源源源	晕灶运
原源源源源源	员源源源源源源源源源源源	员源源源源源	晕灶运

圆) 由原码求补码

除根据定义式(员京苑)~式(员京怨)求取补码外,还可通过数的原码,经适当步骤,获得补码。方法是:

(员) 正整数

【例 员京愿】 设机器字长灶越缘,求垣缘的补码。

$$[赠]_{原} = 越源源源源源$$

$$[赠]_{补} = 越源源源源源$$

正数的补码与原码同。

(圆) 负整数

【例 员京愿】 已知机器字长灶越缘,求原缘的补码。

$$[赠]_{原} = 越源源源源源$$

尾数取反 $\overline{01101010}$

末位加 1 $\overline{01101011}$

可见,负数原码转化为补码表示时,符号位保持不变,其余位取反,然后末位加 1。该过程简称为“取反加 1”。

由补码求真值

定义:正、负号加绝对值形式表示的机器数称为机器数真值。如:

十进制数	机器数	真值	字长
10	00001010	10	原
-10	11110110	-10	原

(1) 正数补码求真值

正数的补码与原码相同,故正数补码的尾数即是真值的尾数,而符号为“0”。

【例】原码 00001010 , 补码 00001010 , 真值 10

(2) 负数补码求真值

负数补码对应的真值可采用下列方法获得:

- ① 符号位不变,其余位取反,末位加 1;
- ② 变符号位为负号,其余位作为真值的数值位。

【例】原码 11110110 , 补码 11110110 , 真值 -10

补码运算

这里讨论补码的加、减运算。其规则是

$[X]_{补} + [Y]_{补} = [X+Y]_{补}$

$[X]_{补} - [Y]_{补} = [X-Y]_{补}$

(1)

【例】原码 00010101 , 补码 00010101 , 原码 11110101 , 补码 11110101

十进制运算

二进制运算

补码运算

101

00010101

00010101 [101]

101

00010101

00010101 [101]

101

00010101

00010101

原 (原)

101

00010101 [101]

101

00010101

00010101