

高等院校信息技术实验教程丛

HUIBIAN YUYAN CHENGXU SHEJI SHIJIAN JIAOCHENG

汇编语言程序设计实践教程

林聪仁 编著



厦门大学出版社
XIAMEN UNIVERSITY PRESS

国家一级出版社
全国百佳图书出版单位

汇编语言程序设计实践教学

林聪仁 编著

厦门大学出版社

图书在版编目(CIP)数据

汇编语言程序设计实践教程/林聪仁编著. —厦门:厦门大学出版社, 2011. 9
ISBN 978-7-5615-4016-9

I. ①汇… II. ①林… 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(2011)第 164667 号

厦门大学出版社出版发行

(地址:厦门市软件园二期望海路 39 号 邮编:361008)

<http://www.xmupress.com>

xmup@public.xm.fj.cn

三明华光印务有限公司印刷

2011 年 9 月第 1 版 2011 年 9 月第 1 次印刷

开本:787×1092 1/16 印张:11.5

字数:290 千字 印数:1~2000 册

定价:20.00 元

本书如有印装质量问题请直接寄承印厂调换

高等院校信息技术实验教程丛书编委会

主任委员:王 琳

副主任委员:隋榕生 李名世

委 员:程曙艳 胡晓毅 黄联芬 李名世 林聪仁 刘舜奎
彭侠夫 隋榕生 王 琳 王晓雪 姚 铭 游佰强

序

21 世纪,科学技术的发展日新月异,信息化时代的来临使信息科学与技术深入社会生活的各个领域。其发展水平已成为衡量一个国家科技实力的重要标志之一。各国都把培养大量高水平的信息科学人才作为科技发展的重要战略目标。

培养高水平的信息科学人才,应重视学生的工程素质和实践能力的培养,提高学生分析问题解决实际问题的能力,这也是当前社会对毕业生专业技能的要求。各高校通过实验课程、课程设计、毕业设计、毕业实习以及组织各种竞赛来提高学生的实践能力、设计与制作能力。

实验是自然科学的基础,是一切科学创造的源泉。学生在本科阶段存在课程多,学时少,实验、实践锻炼的机会更少的问题。一方面由于扩招引起的指导教师、实验资源不足;另一方面也缺少一批实用、高效的实验教材。在厦门大学出版社的大力支持下,我们组织完成了这套“高等院校信息技术实验教程丛书”的编写工作。参与编写该丛书的作者都是担任相关课程的老师或实验指导老师,该丛书是在相关课程经过多年使用的实验讲义的基础上编写而成,收集了较多不同难度的实验项目,供实验课选择。

“高等院校信息技术实验教程丛书”包括《电子技术实验教程》、《电机与电力拖动实验教程》、《可编程控制器(PLC)实验教程》、《自控原理及计算机控制实验教程》、《过程控制实验教程》、《单片机原理与接口技术实验教程》、《电磁场与微波技术实验教程》、《数据库技术实验教程》、《汇编语言程序设计实践教程》、《数字信号处理(DSP)实验教程》十本实验指导书。

在此,我们向所有支持和参与该丛书出版的单位和同志表示感谢,特别要向李茂青教授、许茹教授在该丛书的编写、出版中做出的指导性工作表示感谢。同时,感谢该丛书中使用的实验设备的生产厂家提供的支持。

由于作者的水平与能力有限,丛书中的不足与问题难免,恳请广大师生批评指正。

高等院校信息技术实验教程丛书编委会
2008 年 1 月于厦门大学海韵园

前 言

汇编语言程序设计是计算机专业的一门重要的专业基础课,也是电类各专业的专业基础课“微机原理与接口技术”的重要组成部分。本书的内容涵盖理论教学和实验教学,建议课时为 2+2,即每周理论课和实验课各两节。与现有众多教材相比,本书有以下三大特色。

其一,本书将枯燥的理论教学与繁杂的实践教学紧密结合,使学生只要一书在手就能完全掌握汇编语言程序设计的基本知识和技能。长期以来,理论教学与实验教学严重脱节,实验教学处于附属和可有可无的地位。近年来,许多教师认识到了这一弊端,编写了一些实验教材,但理论教学与实验教学还是处于割裂的状态。本书进行了新的尝试,将理论内容与实验内容融为一体,互相紧密关联,大大提高学生的学习效率和效果。

其二,本书将所有教学内容重新进行了精心的编排,打破了以往的先指令系统,再语法规则,最后编程举例的固有结构,而是按知识点的逻辑关系安排讲授顺序。将指令系统与编程应用交叉讲解,编程举例按功能应用分类而不是按流程结构分类。每章都安排实验项目,每个实验项目都有必做的验证型实验内容和设计型实验内容,还有选做的综合应用型内容,还尽量提出一些引导学生思考的问题,提高学生的分析问题和解决问题的能力。习题的设置也有极强的针对性,让学生充分消化理论知识。与以往的教材相比,本教材从结构、内容到论述方法都有彻底的改变。

本书的第三个特色是特别注重学生的程序调试能力的培养。目前,不管是高级语言还是汇编语言的教材都极少在这方面进行论述,也没有一本专门讲授调试方法的教材,学生无法接受程序调试能力的训练。而良好的程序调试习惯和能力可以说比编程能力更重要。所以本书除了特别强调程序调试的概念外,在各个实验项目中都要求学生进行程序修改、程序设计和程序查错,并要求学生在实验报告中充分体现调试过程和调试数据。通过这些措施使学生接受程序调试的思想,锻炼提高程序调试能力。

本书共分七章,原计划还有高级汇编技术、高档微处理器新增指令、汇编语言与 C 语言混合编程及 WIN32 汇编语言程序设计等四章。因课时有限,一般课内无法讲到这些内容,所以本书暂不把这些章节列入。有兴趣进一步学习的同学可自行查阅其他教材。

本书汇集了作者近 20 年的教学经验,但由于作者水平有限,加上编写时间仓促,错误和不妥之处在所难免,敬请批评指正。

作者
2011 年 7 月

目 录

第一章 汇编语言程序设计基础	(1)
1.1 二进制数和十六进制数	(1)
1.1.1 二进制数	(1)
1.1.2 十六进制	(2)
1.1.3 十进制数与二进制、十六进制数的转换	(3)
1.2 无符号数和带符号数	(4)
1.2.1 无符号数	(5)
1.2.2 带符号数	(5)
1.2.3 真值与机器数的转换	(6)
1.2.4 真值与机器数的关系	(7)
1.2.5 计算机中带符号数为什么要用补码	(8)
1.3 微机基本原理	(9)
1.3.1 微机基本结构框图	(9)
1.3.2 指令的基本执行过程	(10)
1.4 8086CPU	(12)
1.4.1 微处理器概述	(12)
1.4.2 8086CPU 内部结构	(13)
1.4.3 8086 寄存器	(13)
1.4.4 标志寄存器	(14)
1.4.5 判断 OF 标志的三种方法	(18)
1.4.6 标志位小结	(19)
1.5 DEUBG 操作(一)	(20)
1.5.1 DEBUG 的启动	(20)
1.5.2 退出 DEBUG	(20)
1.5.3 DEBUG 的命令	(20)
1.5.4 寄存器命令	(22)
1.5.5 汇编命令	(23)
1.5.6 单步运行命令	(24)
1.6 判断标志位实验	(24)
1.6.1 实验目的	(24)
1.6.2 实验准备	(24)
1.6.3 必做实验	(25)

1.6.4 选做实验	(26)
1.6.5 思考题	(26)
1.7 计算机中的十进制数及其运算原理	(26)
1.7.1 BCD 码	(26)
1.7.2 计算机中 BCD 码的运算	(26)
1.8 BCD 码运算实验	(28)
1.8.1 实验目的	(28)
1.8.2 实验准备	(28)
1.8.3 必做实验	(28)
1.8.4 选做实验	(30)
1.8.5 思考题	(30)
习题	(30)
第二章 汇编语言程序设计入门	(31)
2.1 寻址方式	(31)
2.1.1 隐含寻址	(32)
2.1.2 立即寻址	(32)
2.1.3 寄存器寻址	(32)
2.1.4 段寄存器寻址	(33)
2.1.5 I/O 端口寻址	(33)
2.1.6 转移地址的寻址	(33)
2.1.7 存储器操作数的寻址	(33)
2.2 数据传送指令	(37)
2.2.1 通用数据传送指令	(38)
2.2.2 交换指令	(41)
2.2.3 堆栈操作指令	(41)
2.2.4 查表指令	(43)
2.2.5 输入/输出指令	(44)
2.2.6 地址传送指令	(45)
2.2.7 标志位传送指令	(46)
2.2.8 数据传送指令对标志位的影响	(46)
2.3 汇编语言源程序结构	(47)
2.3.1 分段结构	(48)
2.3.2 语句格式	(49)
2.3.3 变量和标号	(49)
2.3.4 操作数、表达式和操作符	(49)
2.3.5 常用伪指令	(50)
2.4 汇编语言程序上机过程	(52)
2.4.1 编辑	(52)
2.4.2 汇编	(53)
2.4.3 连接	(54)

2.4.4 调试(DEBUG 操作(二))	(55)
2.5 指令错误分析及寻址方式实验	(57)
2.5.1 实验目的	(57)
2.5.2 实验准备	(57)
2.5.3 必做实验	(57)
2.5.4 选做实验	(60)
2.5.5 思考题	(60)
习题	(60)
第三章 算术运算程序	(62)
3.1 跳转指令	(62)
3.1.1 无条件跳转指令	(62)
3.1.2 条件跳转指令	(65)
3.1.3 循环控制指令	(66)
3.2 算术运算指令	(67)
3.2.1 二进制运算指令	(67)
3.2.2 BCD 码调整指令	(70)
3.3 多字节加/减运算程序	(72)
3.3.1 多字节加/减运算程序的基本结构	(72)
3.3.2 多字节二进制加法程序	(72)
3.3.3 程序调试(DEBUG 操作(三))	(74)
3.4 多字节加/减程序实验	(77)
3.4.1 实验目的	(77)
3.4.2 实验准备	(77)
3.4.3 必做实验	(77)
3.4.4 选做实验	(78)
3.4.5 思考题	(78)
3.5 多字节除法运算程序	(78)
3.5.1 多字节除法程序基本编程方法	(78)
3.5.2 除法指令法的除法程序	(79)
3.5.3 被除数左移法的除法程序	(81)
3.6 多字节除法程序实验	(83)
3.6.1 实验目的	(83)
3.6.2 实验准备	(83)
3.6.3 必做实验	(83)
3.6.4 选做实验	(84)
3.6.5 思考题	(84)
3.7 多字节乘法运算程序	(84)
3.7.1 多字节乘法程序的基本编程方法	(84)
3.7.2 乘法指令法的多字节乘法程序	(85)
3.7.3 阶乘程序	(88)

3.8 多字节乘法程序实验	(90)
3.8.1 实验目的	(90)
3.8.2 实验准备	(90)
3.8.3 必做实验	(90)
3.8.4 选做实验	(91)
3.8.5 思考题	(91)
习题	(92)
第四章 代码转换程序	(93)
4.1 逻辑运算指令	(93)
4.1.1 双操作数逻辑运算指令	(93)
4.1.2 单操作数逻辑运算指令	(94)
4.2 移位指令	(94)
4.2.1 逻辑移位指令	(94)
4.2.2 算术移位指令	(94)
4.2.3 循环移位指令	(95)
4.2.4 带 CF 循环移位指令	(95)
4.3 十六进制数转换为 BCD 码	(96)
4.3.1 十六进制数转换为十进制数的人工计算方法	(96)
4.3.2 十六进制数转换为 BCD 码的编程方法	(96)
4.3.3 除 0AH 取余法的十六进制转 BCD 码程序	(98)
4.3.4 连乘 2 法的十六进制转 BCD 码程序	(100)
4.4 十六进制数转 BCD 码程序实验	(101)
4.4.1 实验目的	(101)
4.4.2 实验准备	(101)
4.4.3 必做实验	(101)
4.4.4 选做实验	(102)
4.4.5 思考题	(102)
4.5 BCD 码转换为十六进制数	(102)
4.5.1 十进制数转换为十六进制数的人工计算方法	(103)
4.5.2 BCD 码转换为十六进制数的编程方法	(103)
4.5.3 连乘 0AH 法的 BCD 码转十六进制数程序	(104)
4.6 BCD 码转十六进制数程序实验	(105)
4.6.1 实验目的	(105)
4.6.2 实验准备	(105)
4.6.3 必做实验	(105)
4.6.4 选做实验	(106)
4.6.5 思考题	(106)
4.7 十六进制数与 BCD 码转换方法总结	(106)
4.7.1 整数转换编程方法的数学原理分析	(106)
4.7.2 整数转换编程方法与人工转换方法比较	(107)

4.7.3 小数转换编程方法分析	(108)
习题	(108)
第五章 系统调用程序	(110)
5.1 功能调用和中断调用	(110)
5.1.1 中断指令和中断返回指令	(110)
5.1.2 功能调用	(110)
5.1.3 中断调用	(111)
5.2 键盘输入和屏幕显示功能调用	(111)
5.2.1 键盘输入功能调用	(111)
5.2.2 屏幕显示功能调用	(112)
5.2.3 常用 ASCII	(113)
5.2.4 ASCII 与十进制数和十六进制数的转换	(114)
5.3 键盘输入和屏幕显示程序	(114)
5.3.1 编程思路	(114)
5.3.2 程序流程	(116)
5.3.3 程序清单	(116)
5.4 键盘输入屏幕显示程序实验	(119)
5.4.1 实验目的	(119)
5.4.2 实验准备	(119)
5.4.3 必做实验	(120)
5.4.4 选做实验	(120)
5.4.5 思考题	(120)
习题	(121)
第六章 表处理程序	(122)
6.1 串操作指令及重复前缀	(122)
6.1.1 串操作指令	(122)
6.1.2 串重复前缀	(123)
6.1.3 简单应用举例	(124)
6.2 无符号数和带符号数条件跳转指令	(126)
6.2.1 无符号数条件跳转指令	(126)
6.2.2 带符号数条件跳转指令	(127)
6.3 查表程序和顺序搜索程序	(128)
6.3.1 求正弦函数程序	(128)
6.3.2 顺序搜索程序	(134)
6.4 表处理程序实验(1)	(135)
6.4.1 实验目的	(135)
6.4.2 实验准备	(135)
6.4.3 必做实验	(136)
6.4.4 选做实验	(136)
6.4.5 思考题	(137)

6.5 排序程序和对分搜索程序	(137)
6.5.1 排序程序	(137)
6.5.2 对分搜索程序	(139)
6.6 表处理程序实验(2).....	(143)
6.6.1 实验目的	(143)
6.6.2 实验准备	(144)
6.6.3 必做实验	(144)
6.6.4 选做实验	(145)
6.6.5 思考题	(145)
习题.....	(145)
第七章 子程序及其参数传递.....	(146)
7.1 子程序调用和返回指令	(146)
7.1.1 调用指令	(146)
7.1.2 返回指令	(147)
7.1.3 过程定义伪指令	(148)
7.1.4 处理机控制指令	(148)
7.2 子程序编程基本原理	(149)
7.2.1 通用子程序	(149)
7.2.2 主程序与子程序间参数传递	(150)
7.2.3 保护现场、恢复现场.....	(150)
7.2.4 子程序与宏指令的区别	(154)
7.3 子程序与主程序参数传递	(154)
7.3.1 寄存器传递	(155)
7.3.2 内存变量直接传递	(157)
7.3.3 参数表传递和地址表传递	(159)
7.3.4 堆栈传递	(162)
7.4 子程序参数传递实验	(167)
7.4.1 实验目的	(167)
7.4.2 实验准备	(167)
7.4.3 必做实验	(168)
7.4.4 选做实验	(168)
7.4.5 思考题	(169)
习题.....	(169)
参考文献.....	(170)

第一章 汇编语言程序设计基础

科学家发明计算机的最初目的是代替人工进行复杂的数值计算,目前计算机应用已拓展到社会生活的几乎所有领域,但数值计算还是计算机的最基本功能和应用,也是其他所有应用的基础,从最简单的自动控制应用到最复杂的人工智能应用,都离不开数值计算。所以要充分利用计算机的强大功能,首先必须掌握计算机中有关数的概念,它们的表示方法、运算规则,及它们与日常生活中使用的实际数值之间的关系和转换。

高级语言中所用到的数据类型及运算符与人们的日常生活习惯基本一致,所以用高级语言编程可以很容易地进行数值计算。但高级语言中的一个简单的运算符编译成机器语言(与汇编语言一一对应)时往往需要用一段复杂的子程序来实现。所以要真正掌握计算机中数的计算原理,必须采用汇编语言编程。

汇编语言是面向硬件的语言,与计算机的硬件工作原理密切相关。所以要掌握好汇编语言编程必须了解计算机的基本结构和工作原理,特别是 CPU 的内部结构和原理。

本章讨论计算机中数的表示和计算及微机基本结构原理两个方面的内容,为汇编语言程序设计的学习打下坚实基础。

1.1 二进制数和十六进制数

日常生活中常用的数制有十进制、十二进制、六十进制等。因为计算机的硬件是数字电路,只能用高、低两种电平来表示 0 和 1 两种数字,所以计算机只能用二进制数进行计算,为了书写方便,程序中一般写成十六进制数。八进制数较少使用,本书不作讨论。

从数学上意义看,任意进制数的 n 位整数 m 位小数“ $X_{n-1}X_{n-2}X_{n-3}\cdots X_2X_1X_0X_{-1}X_{-2}\cdots X_{-m+1}X_{-m}$ ”的数值 X 可以用以下公式表示:

$$X = X_{n-1} \times E^{n-1} + X_{n-2} \times E^{n-2} + X_{n-3} \times E^{n-3} + \cdots + X_2 \times E^2 + X_1 \times E^1 + X_0 + X_{-1} \times E^{-1} + X_{-2} \times E^{-2} + \cdots + X_{-m+1} \times E^{-m+1} + X_{-m} \times E^{-m} \quad (0 \leq X_i \leq E-1) \quad (1.1)$$

不同进制的数只是 E 不同,如十进制 $E=10$,十六进制 $E=16$,二进制 $E=2$,其他的数学原理是完全一样的。本书主要讨论整数部分的计算,小数部分的计算方法类似。

1.1.1 二进制数

二进制数(Binary)每位只有 0、1 两种数字,书写习惯上一般加后缀 B,如:1011 0100B, 0101 1100 0011 1001B 等。计算机中的二进制数一般为 8 位、16 位、32 位或 64 位等。

根据十进制数的运算规则类推,可以得到二进制数的运算规则,具体如下:

- (1)加: $0+0=0, 0+1=1, 1+1=1$ 进位 1,高位加低位的进位;
- (2)减: $0-0=0, 1-1=0, 1-0=1, 0-1=1$ 借位 1,高位减低位的借位;
- (3)乘: $0 \times 0=0, 0 \times 1=0, 1 \times 1=1$,乘数每位与被乘数每位相乘,结果再相加;

- (4)除:(a)被除数从最高位开始,每次取 1 位试减除数;
 (b)够减则此位商为 1,余数与被除数的下一位组合再试减除;
 (c)不够减则此位商为 0,被除数再取 1 位再试减;
 (d)如此重复,直至被除数的最后 1 位,最后不够减除数的部分为余数。

二进制数加、减、乘、除的人工计算过程如表 1.1 的例子所示,加、减过程相对简单。乘法的计算过程可称为“被乘数左移法”,除法的计算过程可称为“除数右移试减法”。以后我们将知道,为了简化硬件和编程,乘、除运算过程要进行改进。

本例是整数的计算,若含有小数,只要注意小数点的位置,计算方法是完全一样的。

对于加法和减法计算,被加(减)数、加(减)数的小数点要对齐,和(差)的小数点位置与被加(减)数或加(减)数的小数点位置相同。

对于乘法计算,积的小数位数等于被乘数小数位数和乘数小数位数之和。

对于除法计算,要先将除数的小数点右移,使之转化为整数,被除数的小数点也相应地右移相同的位数,商的小数点位置与被除数的小数点位置相同。

为了清晰表示二进制数,本书中一般在每 4 位二进制间加一个空格,要注意在源程序中数字之间是不能有空格的。

表 1.1 二进制加减乘除

加	1011 1001
	+ 0111 1001
	<u>10011 0010</u>

减	1011 1001
	- 0111 0110
	<u>0100 0011</u>

乘	10111001
	× 01111001
	<u>10111001</u>
	10111001
	10111001
	10111001
	+ 10111001
	<u>0101011101110001</u>

除	0000000100110100
	10011010√101110010101010
	<u>10011010</u>
	11111010
	<u>10011010</u>
	11000001
	<u>10011010</u>
	10011110
	<u>10011010</u>
	010010

1.1.2 十六进制

计算机中只能用二进制表示数,但在源程序中若都用二进制表示数,则书写的数字位太多,且容易写错,所以在源程序及书面表示中,一般用十六进制数代替二进制数。

二进制数转换为十六进制数的规则是:从最低位开始,每 4 位二进制数对应 1 位十六进制数。

十六进制数(Hexadecimal)有 0—9 和 A—F 共 16 个数字。

在汇编语言源程序中,十六进制数一般加后缀 H,当十六进制数的最高位是字母时,在前面必须加一个数字 0 以示与符号名、变量名等的区别。例如:0A9H=1010 1001B。高级语言中十六进制数一般加前缀 0x,有些汇编语言也接受这种表示方法。

十六进制、二进制数与十进制数的对照如表 1.2 所示,表中的 BCD 码在后续章节中讨论。

表 1.2 数制对照表

十进制	二进制	十六进制	压缩型 BCD 码		非压缩型 BCD 码	
			二进制表示	十六进制表示	二进制表示	十六进制表示
0	0000 0000B	00H	0000 0000B	00H	0000 0000 0000 0000B	0000H
1	0000 0001B	01H	0000 0001B	01H	0000 0000 0000 0001B	0001H
2	0000 0010B	02H	0000 0010B	02H	0000 0000 0000 0010B	0002H
3	0000 0011B	03H	0000 0011B	03H	0000 0000 0000 0011B	0003H
4	0000 0100B	04H	0000 0100B	04H	0000 0000 0000 0100B	0004H
5	0000 0101B	05H	0000 0101B	05H	0000 0000 0000 0101B	0005H
6	0000 0110B	06H	0000 0110B	06H	0000 0000 0000 0110B	0006H
7	0000 0111B	07H	0000 0111B	07H	0000 0000 0000 0111B	0007H
8	0000 1000B	08H	0000 1000B	08H	0000 0000 0000 1000B	0008H
9	0000 1001B	09H	0000 1001B	09H	0000 0000 0000 1001B	0009H
10	0000 1010B	0AH	0001 0000B	10H	0000 0001 0000 0000B	0100H
11	0000 1011B	0BH	0001 0001B	11H	0000 0001 0000 0001B	0101H
12	0000 1100B	0CH	0001 0010B	12H	0000 0001 0000 0010B	0102H
13	0000 1101B	0DH	0001 0011B	13H	0000 0001 0000 0011B	0103H
14	0000 1110B	0EH	0001 0100B	14H	0000 0001 0000 0100B	0104H
15	0000 1111B	0FH	0001 0101B	15H	0000 0001 0000 0101B	0105H

十六进制数的加、减计算可以参照二进制数和十进制数的加、减运算规则和过程直接进行计算,也可以先将十六进制数转换成二进制数,计算后再转换成十六进制数。

十六进制数的乘、除计算一般要先将被乘数、乘数、被除数、除数转换为二进制数,然后用二进制数进行乘法和除法运算,最后将计算结果转换为十六进制数。

对于十六进制数的加、减、乘、除运算,这里不再举例了,同学们可自行练习。

1.1.3 十进制数与二进制、十六进制数的转换

尽管计算机只能用二进制数,但在源程序中也可直接使用十进制数,加后缀 D 或省略后缀,汇编程序对源程序进行汇编时会自动将十进制数转换为二进制数。为了让数字的意义直观明了,在程序中经常要将数字写成二进制或十六进制的形式,所以下面讨论十进制数与二进制数、十六进制数转换的人工计算方法。

十进制数转换为二进制数时,整数部分和小数部分必须分别进行转换,且所用的方法不同,整数部分的方法是“除 2 取余法”,小数部分的方法是“乘 2 取整法”,如表 1.3 的例子所示。

十进制数转换为十六进制数时,可以按以上方法先将十进制数转换为二进制数,再将二进制数转换为十六进制数。二进制数转换为十六进制数时,要从小数点开始,向左、向右每次截取 4 位二进制数对应 1 位十六进制数。

当然也可以直接将十进制数转换为十六进制数,整数部分采用“除 16 取余法”,小数部分

采用“乘 16 取整法”。

要注意的是:用“乘 2 取整法”或“乘 16 取整法”将十进制小数转换为二进制小数或十六进制小数时,有可能无法完全精确转换,因为一个十进制小数多次乘 2 或乘 16 后的小数部分有可能永远不会等于 0。所以,十进制小数转换为二进制小数或十六进制小数时有可能会有误差,要根据精度的要求截取适当的小数位数。

表 1.3 十进制数转换为二进制数举例

整数:除 2 取余	小数:乘 2 取整
100	0 .568
50 0 = B0	B ₋₁ = 1 .136
25 0 = B1	B ₋₂ = 0 .272
12 1 = B2	B ₋₃ = 0 .544
6 0 = B3	B ₋₄ = 1 .088
3 0 = B4	B ₋₅ = 0 .176
1 1 = B5	B ₋₆ = 0 .352
0 1 = B6	B ₋₇ = 0 .704
∴ 100 = 0110 0100B	B ₋₈ = 1 .408
	∴ 0.568 = 0.1001 0001B

二进制数转换为十进制数时,整数和小数部分都用“按幂展开”的方法,十六进制数转换为十进制数的方法也是“按幂展开”,即按本节开篇中的公式(1.1)进行计算。请同学们按习题进行练习,这里就不再举例了。

二进制数与十六进制数的转换相对简单,4 位二进制数对应 1 位十六进制数。

1.2 无符号数和带符号数

在日常生活中,我们一般使用十进制数,有正数和负数;而在计算机中,我们只能存储和计算二进制数。那么,如何用二进制形式的数来表示日常生活中的十进制数,并且能区分正数和负数呢?解决这个问题才能使计算机有实用价值。为了解决这个问题,我们首先必须理解清楚机器数和真值这两个概念,掌握它们之间的相互关系。

机器数:计算机中存储的二进制数称为机器数,书写时一般用十六进制表示。

真值:机器数所表示的实际数值称为真值。为了不与机器数相混淆,真值一般用十进制表示。

机器数有无符号数和带符号数两大类,无符号数就是没有正负的数,带符号数就是有正有负的数。本书讨论有关机器数与真值关系的问题时,以 8 位二进制整数为例。其原理可推广到 16 位、32 位、64 位二进制数,若其中含有一部分位数的小数,只要是用定点数表示,原理也是一样的。

8 位二进制机器数与十进制真值之间的关系如图 1.1 所示,图中的机器数用十六进制表示。从图中可以看出,计算机中一个二进制数可以用来表示不同的实际数值(或符号),反过来一个实际的数值在计算机中可以有不同的二进制表示形式。

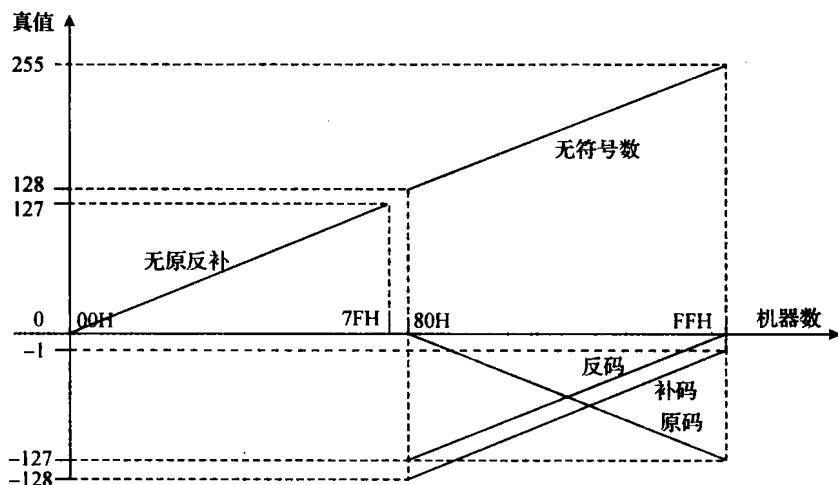


图 1.1 8 位机器数与真值关系

1.2.1 无符号数

计算机中的数一般为 8 位、16 位、32 位或 64 位二进制，若所有二进制位都用来表示数值，没有符号位，此二进制机器数就称为无符号机器数，简称无符号数。

按前一节的二进制数转换为十进制数的计算方法“按幂展开”可知：

最小的 8 位无符号数是 0000 0000B(00H)，对应的真值是 0；

最大的 8 位无符号数是 1111 1111B(0FFH)，对应的真值 255。

所以，8 位无符号数能表示的十进制数的范围是 0~255；同理，16 位无符号数能表示的十进制数的范围是 0~65535。

1.2.2 带符号数

在计算机中为了区分和表示正数、负数，将 8 位、16 位、32 位或 64 位二进制数的最高位作为符号位，0 表示正数、1 表示负数；剩余的 7 位、15 位、31 位或 63 位作为数值位，表示此数的绝对值大小，这就是带符号机器数，简称带符号数。

带符号数有原码机器数、反码机器数和补码机器数三种表示方法，分别简称为原码、反码和补码。下面以 8 位带符号数为例，说明这三种表示方法的原理。

1. 原码

不管正数还是负数，最高位是符号位，0 是正数、1 是负数，低 7 位是该数的绝对值。

最大的 8 位原码是 0111 1111B(7FH)，对应的真值是 127；

最小的 8 位原码是 1111 1111B(FFH)，对应的真值是 -127；

原码 0000 0000B(00H) 和 1000 0000B(80H) 对应的真值都是 0，分别称为正零和负零。

2. 反码

正数的反码与原码相同；负数的反码是此负数的原码的数值位每位取反，符号位 1 不变。

最大的 8 位反码是 0111 1111B(7FH)，对应的真值是 127；