



普通高等教育“十二五”规划教材
高等院校计算机系列教材

Huibiananyuyan Ji Yingyongjiaocheng

汇编语言及应用教程

李浪 熊江 齐忠 ◎ 主编



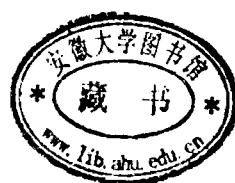
华中科技大学出版社

<http://www.hustp.com>

普通高等教育“十二五”规划教材
高等院校计算机系列教材

汇编语言及应用教程

主 编	李 浪	熊 江	齐 忠
副主编	谢新华	邹 祎	李哲涛
	任文进	方 刚	



华中科技大学出版社
中国·武汉

内 容 提 要

本书以 8086/8088 系列计算机为背景机,以 8086/8088 为基础,由浅入深地介绍了汇编语言程序设计相关知识。全书共 8 章,主要包括汇编语言基础知识、80X86 的寻址方式与指令系统、汇编语言程序格式、汇编语言程序设计、高级汇编语言技术、DOS 中断与 BIOS 中断、I/O 程序设计、汇编语言典型应用编程实例等。本书内容充实、重点突出,特别强调汇编语言的实用编程技巧,侧重于初学者学习本书后,具有一定有实际动手编程能力。所有章节都附有相应的习题与上机实验,对部分重点习题附有参考答案,不同专业可根据需要选用。

本书融合作者多年来的实际教学经验,深知作为初学者学习汇编语言的特点,对于学习中重、难点都有相应的例题与练习。

本书适合作为计算机、自动化、电子信息与通信类等专业的本科与专科教材,也可作为工程技术人员参考用书。

图书在版编目(CIP)数据

汇编语言及应用教程/李浪 熊江 齐忠 主编. —武汉:华中科技大学出版社,2010.11
ISBN 978-7-5609-6458-4

I. 汇… II. ①李… ②熊… ③齐… III. 汇编语言-程序设计-高等学校-教材 IV. TP313

中国版本图书馆 CIP 数据核字(2010)第 144890 号

汇编语言及应用教程

李浪 熊江 齐忠 主编

策划编辑:黄金文

责任编辑:朱建丽

封面设计:范翠璇

责任校对:刘 竣

责任监印:熊庆玉

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编 430074 电话:(027)87557437

录 排:武汉众欣图文照排

印 刷:华中科技大学印刷厂

开 本:787mm×1092mm 1/16

印 张:18

字 数:438 千字

版 次:2010 年 11 月第 1 版第 1 次印刷

定 价:28.80 元



华中出版

本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

高等院校计算机系列教材

编 委 会

主 任:刘 宏

副 主 任:全惠云

编 委:(以姓氏笔画为序)

王志刚	王 毅	乐小波	刘先锋	刘连浩
刘 琳	羊四清	阳西述	许又全	陈书开
陈倩诒	邱建雄	杨凤年	李勇帆	李 浪
张 文	张小梅	何昭青	何迎生	周 昱
罗新密	胡玉平	郭广军	徐雨明	徐长梅
高金华	黄同成	符开耀	龚德良	谭敏生
谭 阳	熊 江	戴经国	瞿绍军	

执行编委:黄金文

前 言

汇编语言程序设计是计算机及电子信息与通信类专业一门重要的基础课程,是培养学生直接使用计算机硬件资源能力的一门课程。同时,它也是学习操作系统、接口与通信技术、计算机组成原理和嵌入式系统开发的前导课程。

本书共分为 8 章,第 1 章介绍了汇编语言基础概念,给出了汇编语言编程所需要的基本知识,包括数据表示与转换、编码及逻辑运算和信息在计算机中的表示方法,80X86 的系统结构、寄存器等,最后还介绍了计算机语言的基本概念。第 2 章介绍了 80X86 寻址方式与指令系统,是本书重要的一章,也是本书的重点和难点,正因为如此,本章一些指令都列举了相应的例题以便于学习和理解。第 3 章介绍了汇编语言程序格式,主要讲述编写汇编语言程序必备的一些格式,包括如何上机进行调试。第 4 章介绍了汇编语言程序设计中常见的几种程序设计方法,即顺序程序设计、分支程序设计、循环程序设计和子程序设计,这几种程序设计方法是汇编语言程序设计基础,复杂的程序都是由它们构成的。第 5 章介绍了高级汇编语言技术,主要阐述了宏汇编、重复汇编、条件汇编,同时为了便于学习者掌握汇编语言与高级语言的接口技术,还介绍了在 VC 中使用嵌入式汇编和汇编语言与 VC 语言的混合编程。第 6 章主要讨论了 DOS 中断和 BIOS 中断的功能及其使用方法。第 7 章主要讨论了汇编语言的 I/O 程序设计,并给出了 I/O 应用的几个实例。第 8 章重点强调对前面知识的应用,是综合性编程的一章,给出了一些汇编语言的典型应用技巧和实例,可以强化学习效果和对汇编语言的应用。

本书每章后有相应的实验和习题,部分重、难点习题在附录中给出了参考答案。

本书在章节的编排上结合多年实践教学的经验,如把 DEBUG 的使用作为附录 A,可供大家在汇编语言的学习过程中方便使用 DEBUG 指令进行调试。

本书第 1 章和第 7 章为熊江编写,第 2 章由谢新华编写,第 3 章和第 5 章由齐忠编写,第 4 章及附录 A、B 由李浪、李哲涛编写,第 6 章由邹祎编写,第 7 章由熊江和方刚共同编写,第 8 章由邹祎、李浪编写,最后由李浪、任文进总编纂。我们衷心感谢兄弟院校的领导和学者和同仁们对本书的支持和肯定,感谢我们的学生认真帮助我们校对和调试程序。

还要特别感谢华中科技大学出版社的所有同志们,是你们的艰辛工作,才使得本书早日与读者见面。

本书的所有作者都是多年从事汇编语言教学的老师,是作者多年教学工作的积累和总结。尽管我们再三校对,但书中难免还存在错误和不足,恳请读者指正和谅解,您的指正是我们的期待,我们的联系方式:lilang911@126.com。

最后,感谢所有帮助和关心本书的朋友。

作 者

2010 年 5 月

目 录

第 1 章 汇编语言基础知识	(1)
1.1 数据表示与运算	(1)
1.1.1 进位记数制与不同基数之间的转换	(1)
1.1.2 数据表示	(8)
1.2 8086/8088 系统结构	(16)
1.2.1 80X86 的编程结构	(16)
1.2.2 80X86 存储器组织结构	(26)
1.3 计算机语言基本概念	(31)
1.3.1 机器语言	(31)
1.3.2 汇编语言	(32)
1.3.3 高级语言	(33)
习题 1	(35)
第 2 章 80X86 寻址方式与指令系统	(37)
2.1 操作数类型	(37)
2.2 寻址方式	(38)
2.2.1 操作数寻址	(38)
2.2.2 指令寻址	(42)
2.3 80X86 的指令系统	(43)
2.3.1 数据传送指令	(43)
2.3.2 算术运算指令	(51)
2.3.3 逻辑指令	(56)
2.3.4 字符串操作指令	(62)
2.3.5 控制转移指令	(67)
2.3.6 处理机控制指令	(73)
实验一 掌握操作数的各种寻址方式	(74)
习题 2	(76)
第 3 章 汇编语言程序格式	(79)
3.1 汇编语言概述	(79)
3.2 汇编语言程序格式	(79)
3.2.1 汇编程序分段结构	(80)
3.2.2 汇编语言语句的类型及组成	(80)
3.2.3 名字	(80)
3.2.4 操作码	(81)

3.2.5 操作数	(81)
3.2.6 注释	(83)
3.3 伪指令	(84)
3.3.1 处理器方式伪指令	(84)
3.3.2 段定义伪指令	(85)
3.3.3 程序开始和结束	(86)
3.3.4 数据定义伪指令	(87)
3.3.5 符号定义伪指令	(88)
3.3.6 地址计数器和定位伪指令	(89)
3.3.7 基数选择伪指令	(90)
3.3.8 过程定义	(90)
3.4 汇编语言程序的实现	(90)
3.4.1 DOS 环境下汇编语言的实现	(90)
3.4.2 Windows 环境下汇编语言的实现	(91)
实验一 汇编语言程序格式及上机过程	(92)
习题 3	(93)
第 4 章 汇编语言程序设计	(96)
4.1 顺序结构	(96)
4.2 分支结构	(99)
4.2.1 显示转移指令实现的分支结构	(99)
4.2.2 用伪指令实现的分支结构	(101)
4.3 循环结构	(104)
4.3.1 循环程序设计方法	(104)
4.3.2 多重循环程序设计	(109)
4.4 子程序	(110)
4.4.1 过程定义伪操作	(110)
4.4.2 子程序的调用和返回指令	(111)
4.4.3 保存与恢复寄存器	(112)
4.4.4 主程序与子程序参数传递方式	(112)
4.4.5 子程序的嵌套	(121)
实验一 顺序程序	(122)
实验二 分支程序	(125)
实验三 循环程序与子程序	(127)
习题 4	(128)
第 5 章 高级汇编语言技术	(132)
5.1 宏汇编	(132)
5.1.1 宏定义	(132)
5.1.2 宏调用和宏展开	(133)

5.1.3 宏调用中的参数使用	(133)
5.1.4 LOCAL 在宏汇编中的作用	(134)
5.1.5 PURGE	(135)
5.1.6 宏指令与子程序的区别	(135)
5.2 重复汇编	(135)
5.2.1 重复伪操作	(136)
5.2.2 不定重复伪操作	(136)
5.3 条件汇编	(137)
5.4 汇编语言与高级语言的接口	(137)
实验一 高级汇编语言技术的使用	(138)
习题 5	(141)
第 6 章 DOS 中断和 BIOS 中断	(143)
6.1 DOS 功能调用及 BIOS 功能调用的概念	(143)
6.1.1 DOS 功能调用	(143)
6.1.2 BIOS 功能调用	(144)
6.1.3 两者间的关系	(144)
6.2 DOS 功能调用及应用	(145)
6.2.1 键盘功能调用	(145)
6.2.2 键盘功能调用应用举例	(146)
6.2.3 显示功能调用	(148)
6.2.4 显示功能调用应用举例	(149)
6.2.5 打印功能	(149)
6.2.6 串行口功能调用	(150)
6.3 BIOS 功能调用及其应用	(152)
6.3.1 键盘输入	(152)
6.3.2 显示输出	(155)
6.3.3 打印输出	(166)
6.3.4 串行通信口 I/O	(168)
实验一 中断调用	(172)
习题 6	(174)
第 7 章 I/O 程序设计	(176)
7.1 工作原理	(176)
7.1.1 I/O 端口地址	(176)
7.1.2 I/O 系统的工作原理	(177)
7.2 数据的 I/O 方式	(179)
7.2.1 直接 I/O 方式	(179)
7.2.2 查询 I/O 方式	(182)
7.2.3 中断 I/O 方式	(183)

7.3 I/O 应用实例	(195)
实验一 有关 I/O 的功能调用	(203)
习题 7	(204)
第 8 章 汇编语言典型应用编程实例	(206)
8.1 打字计时程序	(206)
8.2 视频显示系统程序设计	(214)
8.2.1 显示器	(214)
8.2.2 显示适配器	(214)
8.2.3 显示方式	(215)
8.2.4 电子钟程序的实现	(216)
8.3 音频发声系统程序设计	(220)
8.3.1 计算机的发声原理	(220)
8.3.2 发音程序	(220)
8.3.3 乐曲演奏程序	(221)
8.4 C 语言与汇编语言的混合编程	(224)
8.4.1 汇编语言嵌入 C 语言程序	(224)
8.4.2 在汇编语言中调用 C 语言程序	(226)
8.5 设备驱动程序设计举例	(227)
习题 8	(236)
附录 A DEBUG 常用命令的功能及使用举例	(237)
附录 B 中断向量地址一览	(250)
附录 C DOS 系统功能调用 (INT 21H)	(252)
附录 D BIOS 系统功能调用	(259)
附录 E 部分习题参考答案	(265)
参考文献	(280)

第 1 章 汇编语言基础知识

1.1 数据表示与运算

1.1.1 进位记数制与不同基数之间的转换

人们在生产实践和日常生活中创造了多种表示数的方法,这些数的表示规则称为数制。例如,人们常用的是十进制数,生活中也常常遇到其他进制数,如六十进制数(每分钟 60 秒、每小时 60 分钟,即逢 60 进 1)、十二进制数(铅笔“一打”12 支)、十六进制数(在某些场合如中药、金器的计量单位还在沿用这种计数方法)、二进制数(计算机中采用的计数方法)等。

“进位制”是指用一组固定的数字符号和统一的规则表示数的方法。进位制涉及两个基本问题是基数和权。

基数:在计数制中,每个数位(数字位置)所用到的不同数字的个数称为基数。如十进制数的基数为 10,二进制数的基数为 2,十六进制数的基数为 16。

权:一个数字处于不同位置时,它所代表的数值是不同的,其数值等于该数字乘以一个与数码所在位有关的常数,这个数称为该位上的权。如十进制数 123,其百位上的权为 10^2 ,十位上的权为 10^1 ,个位上的权为 10^0 。

十进制计数法的加法规则是“逢十进一”,任意一个十进制可用 0、1、2、3、4、5、6、7、8、9 这 10 个字符的组合表示,它的基数是 10。二进制计数法的加法规则是“逢二进一”,任意一个二进制数可用 0、1 两个字符表示,其基数为 2。十六进制计数法的加法规则是“逢十六进一”,任意一个十六进制可用 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 共 16 个字符的组合表示,其中 A~F 对应十进制的 10~15,十六进制的基数为 16。

二进制是计算机中采用的数制,计算机中之所以采用二进制数而不采用十进制数,是因为二进制数具有如下的几个特点。

1. 简单易行,容易实现

因为二进制数只有“0”和“1”两个基本符号,易于用两种对立的物理状态表示。例如,可用电灯开关的“闭合”状态表示“1”,用“断开”状态表示“0”;晶体管的导通表示“1”,截止表示“0”;电容器的充电和放电、电脉冲的有和无、脉冲极性的正与负、电位的高与低等一切有两种对立稳定状态的器件都可以表示二进制的“1”和“0”。而十进制数有 10 个基本符号,要用 10 种状态才能表示,所用电子器件实现起来是很困难的。

2. 运算规则简单

二进制数的计算规则非常简单,二进制数求积规则仅有 4 条,即

$$0 \times 0 = 0; 0 \times 1 = 0; 1 \times 0 = 0; 1 \times 1 = 1$$

而十进制数求积规则有 81 条。

3. 适合逻辑运算

二进制数中的“0”和“1”分别表示逻辑代数中的假值(False)和真值(True)。二进制数代表逻辑值容易实现逻辑运算。

1) 十进制数的特点

(1) 由 10 个数码(0~9)组成;

(2) 基数是 10,逢十进一;

(3) 小数点左边从右至左其各位的位权依次为 10^0 、 10^1 、 10^2 、 10^3 等,小数点右边从左至右其各位的位权依次为 10^{-1} 、 10^{-2} 、 10^{-3} 等。

例如,十进制数 521.78 可以表示为

$$521.78 = 5 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

2) 二进制数的特点

(1) 由 2 个数码(0、1)组成;

(2) 基数是 2,逢二进一;

(3) 小数点左边从右至左其各位的位权依次为 2^0 、 2^1 、 2^2 、 2^3 等,小数点右边从左至右其各位的位权依次为 2^{-1} 、 2^{-2} 、 2^{-3} 等。

例如,二进制数 110.11 可以表示为

$$110.11 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

3) 八进制数的特点

(1) 由 8 个数码(0~7)组成;

(2) 基数是 8,逢八进一;

(3) 小数点左边从右至左其各位的位权依次为 8^0 、 8^1 、 8^2 、 8^3 等,小数点右边从左至右其各位的位权依次为 8^{-1} 、 8^{-2} 、 8^{-3} 等。

例如,八进制数 357.65 可以表示为

$$357.65 = 3 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 + 6 \times 8^{-1} + 5 \times 8^{-2}$$

4) 十六进制数的特点

(1) 由 16 个数码(0~9 和 A~F)组成;

(2) 基数是 16,逢十六进一;

(3) 小数点左边从右至左其各位的位权依次为 16^0 、 16^1 、 16^2 等,小数点右边从左至右其各位的位权依次为 16^{-1} 、 16^{-2} 等。

例如,十六进制数 5B.E3 可以表示为

$$5B.E3 = 5 \times 16^1 + 11 \times 16^0 + 14 \times 16^{-1} + 3 \times 16^{-2}$$

一般而言,对于任意的 R 进制数 $a_{n-1}a_{n-2}\cdots a_1a_0.a_{-1}\cdots a_{-m}$ (其中 n 为整数位数, m 为小数位数),可以表示为以下和式:

$$a_{n-1} \times R^{n-1} + a_{n-2} \times R^{n-2} + \cdots + a_1 \times R^1 + a_0 \times R^0 + a_{-1} \times R^{-1} + \cdots + a_{-m} \times R^{-m}$$

式中, R 为基数。

在计算机里,通常用数字后面跟一个大写英文字母来表示该数的数制的书写方法:

(1) 二进制数尾部加 B(Binary)。

(2) 十六进制数尾部加 H(Hexadecimal),如果记数符号由 A、B、C、D、E、F 打头,头部

应加 0, 如 0A8F5H, 记数符号 a、b、c、d、e、f 不区分大小写, 与 A、B、C、D、E、F 等效;

(3) 十进制数尾部加 D(Decimal), 但通常可以省略;

(4) 八进制数尾部加 O(Octal) 或者 Q。

5) 十进制数转换为非十进制数

将十进制数转换为 R 进制数: 整数部分和小数部分须分别遵守不同的转换规则。

整数部分: 除以 R 取余法, 即整数部分不断除以 R 取余数, 直到商为 0 为止, 余数倒排 (最先得到的余数为最低位, 最后得到的余数为最高位)。

小数部分: 乘 R 取整法, 即小数部分不断乘以 R 取整数, 直到小数为 0 或达到有效精度为止, 顺序排列得到的整数 (最先得到的整数为最高位, 最后得到的整数为最低位)。

(1) 十进制数转换为二进制数。

十进制数整数部分转换成二进制数整数部分的方法为“除 2 取余, 余数倒排”; 十进制数小数部分转换成二进制数小数部分的方法为“乘 2 取整, 顺序排列”。为了将一个既有整数部分又有小数部分的十进制数转换成二进制数, 可以将其整数部分和小数部分分别转换, 然后再组合。

例 1.1 将 115.6875D 和 (93/128)D 转换成二进制数。

解 整数部分的转换如下。

$$115/2=57 \quad (a_0=1B)$$

$$57/2=28 \quad (a_1=1B)$$

$$28/2=14 \quad (a_2=0B)$$

$$14/2=7 \quad (a_3=0B)$$

$$7/2=3 \quad (a_4=1B)$$

$$3/2=1 \quad (a_5=1B)$$

$$1/2=0 \quad (a_6=1B)$$

$$115D=1110011B$$

小数部分的转换如下。

$$0.6875 \times 2 = 1.3750 \quad (b_1=1B)$$

$$0.375 \times 2 = 0.750 \quad (b_2=0B)$$

$$0.750 \times 2 = 1.5 \quad (b_3=1B)$$

$$0.5 \times 2 = 1.0 \quad (b_4=1B)$$

$$0.8125D=0.1011B$$

最后结果为 $115.6875D=1110011.1011B$ 。

$$(93/128)D=1 \times (64/128) + 0 \times (32/128) + 1 \times (16/128) + 1 \times (8/128) + 1 \times (4/128) + 0 \times (2/128) + 1 \times (1/128) = 0.101101B$$

注意: 一个十进制整数一定能完全准确地转换成二进制整数, 一个十进制小数不一定能完全准确地转换成二进制小数, 可以根据精度要求转换到小数点后某一位为止即可。将其整数部分和小数部分分别转换。

(2) 十进制数转换为八进制数。

十进制数转换成八进制数, 方法同十进制数转换成二进制数, 只不过“除 2 取余”变为

“除 8 取余”，“乘 2 取整”变为“乘 8 取整”。

例 1.2 将 253.32D 转换成八进制数(转换结果取 3 位小数)。

解 整数部分的转换如下。

$$253/8=31 \quad (a_0=5Q)$$

$$31/8=3 \quad (a_1=7Q)$$

$$3/8=0 \quad (a_2=3Q)$$

$$253D=375Q$$

小数部分的转换如下。

$$0.32 \times 8 = 2.56 \quad (b_1=2Q)$$

$$0.56 \times 8 = 4.48 \quad (b_2=4Q)$$

$$0.48 \times 8 = 3.84 \quad (b_3=3Q)$$

$$0.32D \approx 0.243Q$$

最后结果为 $253.32D=375.243Q$ 。

(3) 十进制数转换为十六进制数。

十进制数转换成十六进制数,整数部分为“除 16 取余”,小数部分为“乘 16 取整”。

例 1.3 将 68413.45D 转换成十六进制数(转换结果取 3 位小数)。

解 整数部分的转换如下。

$$68413/16=4275 \quad (a_0=13D=0DH)$$

$$4275/16=267 \quad (a_1=3H)$$

$$267/16=16 \quad (a_2=11D=0BH)$$

$$16/16=1 \quad (a_3=0H)$$

$$1/16=0 \quad (a_4=1H)$$

$$68413D=10B3DH$$

小数部分的转换如下。

$$0.45 \times 16 = 7.2 \quad (b_1=7H)$$

$$0.2 \times 16 = 3.2 \quad (b_2=3H)$$

$$0.2 \times 16 = 3.2 \quad (b_3=3H)$$

$$0.45D=0.733H$$

最后结果为 $68413.45D=10B3D.733H$ 。

6) 非十进制数转换为十进制数

把各非十进制数(二进制数、八进制数、十六进制数)按权展开求和得到的结果即为十进制数。

例 1.4 将二进制数 101.01B 转换成等值的十进制数。

$$\begin{aligned} \text{解} \quad 101.01B &= (1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2})D \\ &= (4 + 0 + 1 + 0.25)D = 5.25D \end{aligned}$$

例 1.5 将八进制数 $(3577.4)_8$ 转换成等值的十进制数。

$$\text{解} \quad 3577.4Q = (3 \times 8^3 + 5 \times 8^2 + 7 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1})D = 1919.5D$$

例 1.6 将十六进制数 A3E.BH 转换成等值的十进制数。

解 A3E.BH=(10×16²+3×16¹+14×16⁰+11×16⁻¹)D=2622.6875D

7) 二进制数与八进制数、十六进制数之间的转换

(1) 二进制数转换成八进制数。

8 是 2 的整数次幂,即 $8=2^3$,因此 3 位二进制数相当于 1 位八进制数,它们之间的转换关系也相当简单。由于二进制数表示数值的位数较长,进制越大,数的表达长度也就越短,因此常需要用八进制数来表示二进制数。

二进制数转换成八进制数的方法是:将二进制数从小数点开始分别向左(整数部分)和向右(小数部分)每 3 位二进制数分成一组,整数部分向左分组,不足位数左边加 0 补足;小数部分向右分组,不足部分右边加 0 补足,然后将每组二进制数转化成八进制数即可。它们的对应关系见表 1.1。

表 1.1 常用数制对照表

十进制数	二进制数	八进制数	十六进制数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
⋮	⋮	⋮	⋮

例 1.7 将二进制数 11101110.00101001B 转换成八进制数。

解 $\begin{array}{ccc} \underline{011} & \underline{101} & \underline{110} \\ 3 & 5 & 6 \end{array} . \begin{array}{ccc} \underline{001} & \underline{010} & \underline{010} \\ 1 & 2 & 2 \end{array} \text{B} = 356.122\text{Q}$

(2) 二进制数转换成十六进制数。

16 是 2 的整数次幂,即 $16=2^4$,因此 4 位二进制数相当于 1 位十六进制数,它们之间的

转换关系也相当简单。由于二进制数表示数值的位数较长,因此常用十六进制数来表示二进制数。

二进制数转换成十六进制数的方法是:将二进制数从小数点开始分别向左(整数部分)和向右(小数部分)每 4 位二进制数分成一组,整数部分向左分组,不足位数左边加 0 补足;小数部分向右分组,不足部分右边加 0 补足,然后将每组二进制数转化成十六进制数即可。

例 1.8 将二进制数 11101100.0011101111B 转换成十六进制数。

解 $\begin{array}{ccc} \underline{1110} & \underline{1100} & . & \underline{0011} & \underline{1011} & \underline{1100} \\ \text{E} & \text{C} & . & 3 & \text{B} & \text{C} \end{array}$ $\text{11101100.0011101111B} = \text{EC.3BCH}$

(3) 八进制数转换成二进制数。

八进制数转换成二进制数的方法是:将每位八进制数写成相应的 3 位二进制数,再按顺序排列好。

例 1.9 把八进制数 3576.15Q 转换为二进制数。

解

八进制数	3	5	7	6	.	1	5
二进制数	011	101	111	110	.	001	101

最后结果为 $3576.15\text{Q} = 11101111110.001101\text{B}$ 。

注意:整数前的高位零和小数后的低位零可取消。

(4) 十六进制数转换成二进制数。

十六进制数转换成二进制数的方法是:将每位十六进制数用 4 位二进制数来表示,再按顺序排列好。

例 1.10 把十六进制数 8AE9.7H 转换为二进制数。

解

十六进制数	8	A	E	9	.	7
二进制数	1000	1010	1110	1001	.	0111

最后结果为 $8\text{AE}9.7\text{H} = 1000101011101001.0111\text{B}$ 。

注意:从以上例题可以看到,二进制数和八进制数、十六进制数之间的转换非常直观,要把一个十进制数转换成二进制数可以先转换为八进制数或十六进制数,然后再快速地转换成二进制数。同样,在转换中若要将十进制数转换为八进制数和十六进制数时,也可以先把十进制数转换成二进制数,然后再转换为八进制数或十六进制数,如表 1.1 所示为常用数制对照表。

例如,将十进制数 789D 转换为二进制数,可以先转换成八进制数(除以 8 求余法)得 1425Q ,再按每位八进制数转为 3 位二进制数,求得 1100010101B ,如果还要转换成十六进制数用 4 位一组很快就能得到 315H 。

8) 二进制数的算术运算

在计算机中,二进制数可作算术运算规则如下。

加法: $0+0=0$; $1+0=0+1=1$; $1+1=10$ (有进位 1)

减法: $0-0=0$; $10-1=1$ (借一当二); $1-0=1$; $1-1=0$

乘法: $0\times 0=0$; $0\times 1=1\times 0=0$; $1\times 1=1$

除法: $0/1=0$; $1/1=1$

例 1.11 $1001\text{B}+101\text{B}=?$

$$\begin{array}{r} \text{解} \quad 1001 \\ + \quad 101 \\ \hline = 1110 \end{array}$$

结果为 $1001\text{B}+101\text{B}=1110\text{B}$ 。

例 1.12 $1101\text{B}\times 11\text{B}=?$

$$\begin{array}{r} \text{解} \quad 1101 \\ \times \quad 11 \\ \hline 1101 \\ + 1101 \\ \hline = 100111 \end{array}$$

结果为 $1101\text{B}\times 11\text{B}=100111\text{B}$ 。

9) 十六进制数的算术运算

十六进制数的运算可以采用先把该十六进制数转换为十进制数,经过计算后再把结果转换为十六进制数的方法,但这样做比较烦琐。其实,只要按照逢十六进一的规则,直接用十六进制数来计算也是很方便的。

例 1.13 $05\text{C}5\text{H}+6\text{D}25\text{H}=?$

$$\begin{array}{r} \text{解} \quad 05\text{C}5 \\ + \quad 6\text{D}25 \\ \hline = 72\text{E}A \end{array}$$

结果为 $05\text{C}5\text{H}+6\text{D}25\text{H}=72\text{E}A\text{H}$ 。

注意:十六进制加法运算中,当两个一位数之和 S 小于 16 时,与十进制数同样处理;如果两个一位数之和不小于 16 时,则应该用 $S-16$ 及进位 1 来取代 S 。

例 1.14 $7\text{D}26\text{H}-05\text{C}2\text{H}=?$

$$\begin{array}{r} \text{解} \quad 7\text{D}26 \\ - \quad 05\text{C}2 \\ \hline = 7764 \end{array}$$

结果为 $7\text{D}26\text{H}-05\text{C}2\text{H}=7764\text{H}$ 。

注意:十六进制数的减法也与十进制数类似,够减时可直接相减,不够减时则向高位借 1 为 16 的规则。

例 1.15 $07\text{D}5\text{H}\times 00\text{BCH}=?$

$$\begin{array}{r} \text{解} \quad 07\text{D}5 \\ \times 00\text{BC} \\ \hline 5\text{DFC} \\ + 5627 \\ \hline = 5\text{C}06\text{C} \end{array}$$

结果为 $07\text{D}5\text{H}\times 00\text{BCH}=5\text{C}06\text{CH}$

注意:十六进制数的乘法可以用十进制数的乘法规则来计算,但结果必须转化为十六进制数表示。

1.1.2 数据表示

数据(Data)是表征客观事物的、可以被记录的、能够被识别的各种符号,包括字符、符号、表格、声音、图形和图像等。简而言之,一切可以被计算机加工、处理的对象都可以称为数据。数据可在物理介质上记录或传输,并通过外部设备被计算机接收,经过处理而得到结果。

数据能被送入计算机加以处理,包括存储、传送、排序、归并、计算、转换、检索、制表和模拟等操作,以得到人们需要的结果。数据经过解释并赋予一定的意义后,便成为信息。计算机中数据的常用单位有位、字节和字。

计算机采用二进制,运算器运算的是二进制数,控制器发出的各种指令也表示成二进制数,存储器中存放的数据和程序也是二进制数,在网络上进行数据通信时发送和接收的还是二进制数。显然,在计算机内部到处都是由0和1组成的数据流。

计算机中最小的数据单位是二进制的—个数位,简称为位(Bit)。计算机中最直接、最基本的操作就是对二进制位的操作。

字节(Byte):字节简称为B,8位为1个字节,1个字节由8个二进制数位组成。字节是计算机中用来表示存储空间大小的基本容量单位。例如,计算机内存的存储容量、磁盘的存储容量等都是以字节为单位表示的。除用字节为单位表示存储容量外,还可以用千字节(KB)、兆字节(MB)及十亿字节(GB)等表示存储容量。它们之间存在下列换算关系:

$$1\text{B}=8\text{bit}$$

$$1\text{KB}=1024\text{B}=2^{10}\text{B}$$

$$1\text{MB}=1024\text{KB}=2^{10}\text{KB}=2^{20}\text{B}=1024\times 1024\text{B}$$

$$1\text{GB}=1024\text{MB}=2^{10}\text{MB}=2^{30}\text{B}=1024\times 1024\text{KB}$$

$$1\text{TB}=1024\text{GB}=2^{10}\text{GB}=2^{40}\text{B}=1024\times 1024\text{MB}$$

要注意位与字节的区别:位是计算机中最小数据单位,字节是计算机中基本信息单位。

字(Word):在计算机中作为一个整体被存取、传送、处理的二进制数字字符串称为一个字或单元,每个字中二进制位数的长度,称为字长。一个字由若干个字节组成,不同的计算机系统的字长是不同的,常见的有8位、16位、32位、64位等,字长越长,计算机—次处理的信息位就越多,精度就越高,字长是计算机性能的一个—个重要指标。

计算机的基本功能是对数据进行运算和加工处理。数据有两种,一种是数值数据,如3.1416、-5.61828……;另一种是非数值数据(信息),如A、b、+、=……无论哪一种数据在计算机中都是用二进制数表示的。计算机只能直接识别二进制数,所有的符号都是用二进制数表示的,数的正、负号也是用二进制数表示。数值的最高位用“0”和“1”分别表示数的正、负号。数值处理采用二进制运算,非数值处理采用二进制编码,它们具有运算简单、电路实现方便、成本低廉等优点。

在计算机内部,数字和符号都用二进制数表示,两者合在一起构成数的机内表示形式,