

高等院校计算机系列教材

汇编语言程序设计

主 编 李 浪 熊 江 睦仁武
副主编 焦 铭 黄 顺 欧阳陈华 魏祖雪

E m a i



WUHAN UNIVERSITY PRESS

武汉大学出版社

TP313

99

2007

高等院校计算机系列教材

汇编语言程序设计

主 编 李 浪 熊 江 睦仁武

副主编 焦 铭 黄 顺 欧阳陈华 魏祖雪

第 1 版 (2007)



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

汇编语言程序设计/李浪,熊江,睦仁武主编. —武汉:武汉大学出版社,2007.1

高等院校计算机系列教材

ISBN 978-7-307-05395-3

I. 汇… II. ①李… ②熊… ③睦… 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(2006)第 153830 号

责任编辑:黄金文 夏炽元

责任校对:刘 欣

版式设计:支 笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.com.cn)

印刷:湖北省新华印务有限公司

开本:787×1092 1/16 印张:16.5 字数:392 千字

版次:2007 年 1 月第 1 版 2007 年 1 月第 1 次印刷

ISBN 978-7-307-05395-3/TP·233 定价:25.00 元

版权所有,不得翻印;凡购买我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。

高等院校计算机系列教材

编 委 会

主 任：刘 宏，湖南师范大学数学与计算机科学学院计算机系主任，教授
编 委：（以姓氏笔画为序）

王 毅，湘潭大学信息工程学院副院长，副教授

乐晓波，长沙理工大学计算机与通信工程学院计算机科学与技术系
主任，教授

许又泉，邵阳学院信息电气工程系

羊四清，湖南人文科技学院计算机系主任，副教授

刘先锋，湖南师范大学数学与计算机科学学院，教授

刘连浩，中南大学信息工程学院计算机系教授

全惠云，湖南师范大学数学与计算机科学学院信息与计算科学系主
任，教授

沈 岳，湖南农业大学信息科学技术学院院长，副教授

张小梅，凯里学院数学与计算机科学系副主任，副教授

杨克昌，湖南理工学院计算机与信息工程系教授

何迎生，吉首大学数学与计算机科学学院信息与计算科学系副主任

邱建雄，长沙学院计算机科学与技术系副教授

李勇帆，湖南第一师范学院信息技术系主任，教授

周 昱，吉首大学师范学院数学与计算机科学系副主任

罗新密，湖南商学院计算机与电子工程系副教授

徐雨明，衡阳师范学院计算机科学系副主任，副教授

郭国强，湖南文理学院计算机科学与技术系主任，教授

晏峻峰，湖南中医药大学计算机系副教授

龚德良，湘南学院计算机科学系副主任，副教授

蒋伟进，湖南工业大学计算机科学与技术系副主任，副教授

熊江，重庆三峡学院数学与计算机科学学院副教授

谭敏生，南华大学计算机学院院长，副教授

戴祖雄，湖南科技大学计算机科学与工程学院

执行编委：黄金文，武汉大学出版社计算机图书事业部主任，副编审



内容提要

本书以 8086/8088 系列微机为背景机,以 8086/8088 CPU 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、debug 常用命令及用法举例、80x86 的寻址方式和指令系统、汇编语言程序格式、汇编语言程序设计、高级汇编技术、DOS 中断和 BIOS 中断、输入输出程序设计等。本书内容充实、重点突出,所选例题均具有较强的代表性,适合举一反三,所有章节都附有相应的习题与上机实验练习,对部分重点习题附有参考答案,不同专业可根据需要选用。

本书适合作为计算机、自动化、机电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。



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、计算机组成原理和单片机技术的前导课程。

本书共分为八章,第一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、计算机中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。第二章介绍了 debug 常用命令及用法,debug 是调试汇编语言的一种有效工具。第三章介绍了 80x86 寻址方式与指令系统。指令系统是一台计算机所有指令的集合,它是本书重要的一章,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。第四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。第五章介绍了汇编语言程序设计中常见的几种程序设计方法: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。第六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 Turbo C 中使用嵌入式汇编和汇编语言与 Turbo C 语言的混合编程。第七章主要讨论了汇编语言程序员能控制的 DOS 中断和 BIOS 中断的功能及其使用方法。第八章主要讨论了汇编语言的输入/输出设计,并给出了输入/输出应用的几个实例。

本书每章后有相应的实验和习题,部分重点、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教学实践的经验,如把 debug 的使用单独成章,并且放在第三章寻址方式与指令系统的前面,目的就是便于对后面的指令进行学习。

本书第一章和第八章第一节、第三节由重庆三峡学院的熊江编写,第二章和第五章由衡阳师范学院李浪编写,第三章由衡阳师范学院欧阳陈华编写,第四章由衡阳师范学院焦铭编写,第六章由衡阳师范学院黄顺编写,第七章由衡阳师范学院眭仁武编写,第八章第二节由重庆三峡学院魏祖雪编写,最后由李浪老师总编纂。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,感谢学生们认真帮助我们进行校对和程序调试。

我们还要特别感谢武汉大学出版社的所有同志们,黄金文老师对本书出版作了许多建设性的指导,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。本书是他们多年教学工作的积累和总结。但肯定还存在错误和不足,恳请读者指正和谅解,您的指正是我们的期待,我们的联系方式是 lilang911@126.com。

最后,我们再次感谢所有帮助和关心我们的朋友,谢谢你们使用本书,并祝你们学习成功。

作者

2006 年 11 月于衡阳

目 录

第一章 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	1
1.1.2 不同进制数之间的转换	3
1.1.3 二进制数及十六制数的算术运算和二进制数的逻辑运算	6
1.1.4 数据表示	9
1.2 计算机语言基本概念	14
1.2.1 机器语言	15
1.2.2 汇编语言	15
1.2.3 高级语言	16
1.3 80x86 CPU 的编程结构	18
1.3.1 80x86 微处理器的发展历程	18
1.3.2 8086/8088 的内部结构	20
1.3.3 80x86CPU 的寄存器结构	22
1.4 80x86 存储器组织结构	26
1.4.1 存储器简介	26
1.4.2 存储单元的地址和内容	28
1.4.3 32 位微机的存储单元的地址和内容	30
习题 1	31
第二章 DEBUG 常用命令的功能及使用举例	33
2.1 DEBUG 的主要用途及命令集	33
2.2 DEBUG 的主要命令功能与格式	34
2.3 DEBUG 使用举例	38
2.4 使用 DEBUG 调试和运行可执行文件	40
习题 2	45
第三章 80x86 寻址方式与指令系统	46
3.1 操作数类型	46
3.2 寻址方式	46
3.2.1 立即寻址方式	47
3.2.2 寄存器寻址方式	47
3.2.3 直接寻址方式	48



3.2.4 寄存器间接寻址方式	49
3.2.5 寄存器相对寻址方式	49
3.2.6 基址变址寻址方式	50
3.2.7 相对基址变址寻址方式	51
3.3 80x86 的指令系统	53
3.3.1 数据传送指令	53
3.3.2 算术运算指令	59
3.3.3 逻辑指令	63
3.3.4 串处理指令	69
3.3.5 控制转移指令	72
3.3.6 处理机控制	79
实验 掌握操作数的各种寻址方式	80
习题 3	82

第四章 汇编语言程序格式

4.1 汇编程序功能	84
4.2 汇编语言语句格式	85
4.2.1 名字项	85
4.2.2 操作码项	86
4.2.3 操作数项	86
4.2.4 注释项	88
4.3 伪操作	88
4.3.1 处理器选择伪操作	89
4.3.2 段定义伪操作	89
4.3.3 程序开始和结束伪操作	92
4.3.4 数据定义及存储器分配伪操作	92
4.3.5 表达式赋值伪操作 EQU	94
4.3.6 地址计数器与对准伪操作	95
4.3.7 基数控制伪操作	95
4.3.8 过程定义伪指令	96
4.4 汇编语言程序的上机过程	96
4.4.1 建立汇编语言的工作环境	96
4.4.2 建立汇编源程序	97
4.4.3 用 MASM 程序产生目标文件	97
4.4.4 编译不通过,重新修改	97
4.4.5 用 LINK 程序产生 EXE 文件	97
4.4.6 程序的执行	97
4.4.7 Windows 环境下的汇编语言集成编程环境	98
实验 汇编语言程序格式及上机过程	100
习题 4	101



第五章 汇编语言程序设计	103
5.1 顺序结构	103
5.2 分支结构	106
5.2.1 显示转移指令实现的分支结构	106
5.2.2 用伪指令实现的分支结构	109
5.3 循环结构	111
5.3.1 循环程序设计方法	112
5.3.2 多重循环程序设计	116
5.4 子程序	117
5.4.1 过程定义伪操作	117
5.4.2 子程序的调用和返回指令	118
5.4.3 保存与恢复寄存器	119
5.4.4 主程序与子程序参数传递方式	120
5.4.5 子程序的嵌套	128
实验一 顺序程序编写与调试	130
实验二 分支程序	133
实验三 循环程序与子程序	134
习题 5	136
第六章 高级汇编语言技术	139
6.1 宏汇编	139
6.1.1 宏定义、宏调用和宏展开	139
6.1.2 LOCAL 伪操作在宏中的作用	140
6.1.3 算术、逻辑、关系操作符在宏中的使用	143
6.2 重复汇编	144
6.2.1 重复伪操作	144
6.2.2 不定重复伪操作	145
6.3 条件汇编	146
6.3.1 条件伪操作的一般格式	147
6.3.2 条件伪操作的举例	147
6.4 汇编语言与其他高级语言的嵌入	149
6.4.1 在 Turbo C 中使用嵌入式汇编	149
6.4.2 汇编语言与 Turbo C 语言的混合编程	149
实验 高级汇编技术的使用	151
习题 6	154
第七章 DOS 中断和 BIOS 中断	156
7.1 键盘 I/O	157
7.1.1 字符码和扫描码	158



7.1.2 BIOS 键盘中断	159
7.1.3 DOS 键盘功能调用	160
7.2 显示器 I/O	165
7.2.1 字符属性	165
7.2.2 BIOS 显示中断	168
7.2.3 DOS 显示功能调用	175
7.3 打印机 I/O	177
7.3.1 DOS 打印功能	178
7.3.2 打印机的控制字符	178
7.3.3 BIOS 打印功能	182
7.4 串行通信口 I/O	185
7.4.1 串行通信接口	185
7.4.2 串行口功能调用	188
习题 7	192
第八章 输入/输出程序设计	194
8.1 工作原理	194
8.1.1 I/O 端口地址	194
8.1.2 I/O 系统的工作原理	195
8.2 数据的输入/输出方式	197
8.2.1 直接 I/O 方式	197
8.2.2 查询 I/O 方式	200
8.2.3 中断 I/O 方式	201
8.3 输入/输出应用实例	213
习题 8	217
实验 有关输入/输出的功能调用	218
附录 1 中断向量地址一览	220
附录 2 DOS 系统功能调用 (INT 21H)	222
附录 3 BIOS 系统功能调用	229
附录 4 部分习题参考答案	236
参考文献	250



第一章 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

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。例如人们常用的十进制,生活中也常常遇到其他进制,如六十进制(每分钟 60 秒、每小时 60 分钟,即逢 60 进 1),十二进制(商业中不少包装计量单位“一打”),十六进制(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,因为解放前的 1 斤等于 16 两)等,计算机中采用二进制等。

“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: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叫做基数。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 235,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 十个字符的组合表示,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 两个数字表示,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 十六个字符的组合表示,其中 A~F 对应十进制的 10~15,它的基数是 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而不采用十进制是因为二进制具有如下的几个特点:

其一: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“0”和“1”。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。

其二:运算规则简单

二进制的计算规则非常简单,二进制求积规则仅有 4 条:

$0 \times 0 = 0$; $0 \times 1 = 0$; $1 \times 0 = 0$; $1 \times 1 = 1$,而十进制求积规则有 81 条。



其三:适合逻辑运算

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

1. 十进制数的特点

(1) 由十个数码 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是: 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是: 10^{-1} 、 10^{-2} 、 10^{-3} 等。

例如:十进制数 945.68 可以表示为: $945.68 = 9 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$

2. 二进制数的特点

(1) 由两个数码 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是: 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是: 2^{-1} 、 2^{-2} 、 2^{-3} 等。

例如:二进制数 101.01 可以表示为: $101.01 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$ 。

3. 八进制数的特点

(1) 由八个数码 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是: 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是: 8^{-1} 、 8^{-2} 、 8^{-3} 等。

例如:八进制数 235.74 可以表示为: $235.74 = 2 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 + 7 \times 8^{-1} + 4 \times 8^{-2}$

4. 十六进制数的特点

(1) 由十六个数码 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是: 16^0 、 16^1 、 16^2 等,小数点右边从左至右其各位的位权依次是: 16^{-1} 、 16^{-2} 等。

例如:十六进制数 3A.C1 可以表示为: $3A.C1 = 3 \times 16^1 + 10 \times 16^0 + 12 \times 16^{-1} + 1 \times 16^{-2}$

一般而言,对于任意的 R 进制数

$a_{n-1}a_{n-2}\cdots a_1a_0 \cdot a_{-1}\cdots a_{-m}$ (其中 n 为整数位数, m 为小数位数)

可以表示为以下和式:

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

为基数)

在计算机里,通常用数字后面跟一个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);如记数符号 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 ABCDEF 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。



1.1.2 不同进制数之间的转换

1. 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。

对整数部分:除以 R 取余法,即整数部分不断除以 R 取余数,直到商是 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)。

对小数部分: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(最靠近小数点),最后得到的整数为最低位)。

(1) 十进制数转换为二进制数

十进制整数部分转换成二进制数整数部分的方法是:“除 2 取余,余数倒排”;对十进制小数部分转换成二进制数小数部分的方法是:“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 109.8125D 转换成二进制数。

整数部分的转换:

$$109/2 = 54 \cdots 1 \quad (a_0 = 1)$$

$$54/2 = 27 \cdots 0 \quad (a_1 = 0)$$

$$27/2 = 13 \cdots 1 \quad (a_2 = 1)$$

$$13/2 = 6 \cdots 1 \quad (a_3 = 1)$$

$$6/2 = 3 \cdots 0 \quad (a_4 = 0)$$

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

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

$$109D = 1101101B$$

小数部分的转换:

$$0.8125 \times 2 = 1.625 \quad (b_1 = 1)$$

$$0.625 \times 2 = 1.25 \quad (b_2 = 1)$$

$$0.25 \times 2 = 0.5 \quad (b_3 = 0)$$

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

$$0.8125D = 0.1101B$$

最后结果为:109.8125D = 1101101.1101B

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,然后组合起来得 109.8125D = 1101101.1101B。

(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 123.32D 转换成八进制数(转换结果取 3 位小数)。

整数部分的转换:

$$123/8 = 15 \cdots 3 \quad (a_0 = 3)$$



$$15/8 = 1 \cdots 7 \quad (a_1 = 7)$$

$$1/8 = 0 \cdots 1 \quad (a_2 = 1)$$

$$123D = 173Q$$

小数部分的转换:

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

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

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

$$0.32D \approx 0.243Q$$

$$\text{最后结果为: } 123.32D = 173.243Q$$

(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: “除 16 取余”, 小数部分: “乘 16 取整”。

例 1.3 将 58412.45D 转换成十六进制数(转换结果取 3 位小数)。

整数部分的转换:

$$58412/16 = 3650 \cdots 12 \quad (a_0 = 12D = 0CH)$$

$$3650/16 = 228 \cdots 2 \quad (a_1 = 2)$$

$$228/16 = 14 \cdots 4 \quad (a_2 = 4)$$

$$14/16 = 0 \cdots 14 \quad (a_3 = 14D = 0EH)$$

$$58412D = E42CH$$

小数部分的转换:

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

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

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

$$0.45D = 0.733H$$

$$\text{最后结果为: } 58412.45D = E42C.733H$$

2. 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 11.01B 转换成等值的十进制数。

$$11.01B = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 2 + 1 + 0.25 = 3.25D$$

例 1.5 将八进制数(2576.4)₈ 转换成等值的十进制数。

$$(2576.4)_8 = 2 \times 8^3 + 5 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 + 4 \times 8^{-1} = (1406.5)_{10}$$

例 1.6 将十六进制数(A3D.B)₁₆ 转换成等值的十进制数。

$$(A3D.B)_{16} = 10 \times 16^2 + 3 \times 16^1 + 13 \times 16^0 + 11 \times 16^{-1} = (2621.6875)_{10}$$

3. 二进制与八、十六进制之间的转换

(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, 数的表示长度也就越短, 因此常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 位二进制数码分成一组, 整数部分向左分组, 不足位数左补 0。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。它们的对应关系如表 1.1 所示。

例 1.7 将二进制数 $(10101110.00101011)_2$ 转换成八进制数。

$$\left(\begin{array}{ccc} \underline{010} & \underline{101} & \underline{110} \\ 2 & 5 & 6 \end{array} . \begin{array}{ccc} \underline{001} & \underline{010} & \underline{110} \\ 1 & 2 & 6 \end{array} \right)_2 = (256.126)_8$$

(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,因此常需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 位二进制数码分成一组,整数部分向左分组,不足位数左补 0。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。

表 1.1

常用进制数对照表

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 1.8 将二进制数 $(11101110.0010101111)_2$ 转换成十六进制数。

$$(\begin{array}{cc} \underline{1110} & \underline{1110} \\ \text{E} & \text{E} \end{array} . \begin{array}{ccc} \underline{0010} & \underline{1011} & \underline{1100} \\ 3 & \text{B} & \text{C} \end{array})_2 = (\text{EE.3BC})_{16}$$

(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:将每一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 $(2376.16)_8$ 转换为二进制数。

八进制数	2	3	7	6	.	1	6
转换成二进制数	010	011	111	110	.	001	110

最后结果为: $(2376.16)_8 = (10011111110.00111)_2$

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:是将 1 位十六进制数用 4 位二进制数码来表示,再按顺序排列好。

例 1.10 把二进制数 $(111101011101001.011)_2$ 转换为十六进制数。

二进制按 4 位分组:	<u>0111</u>	<u>1010</u>	<u>1110</u>	<u>1001</u>	.	<u>0110</u>
转换成十六进制数:	7	A	E	9	.	6

最后结果为: $(111101011101001.011)_2 = (7AE9.6)_{16}$

注意:从以上例题我们看到二进制和八进制、十六进制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 所示为常用进制数对照表。例如将十进制数 678 转换为二进制数,可以先转换成八进制数(除以 8 求余法)得 1246,再按每位八进制数转为 3 位二进制数,求得 1010100110B,如还要转换成十六进制数用 4 位一组很快就能得到 2A6H。

1.1.3 二进制数及十六制数的算术运算和二进制数的逻辑运算

1. 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 $1001\text{B} + 101\text{B} = ?$