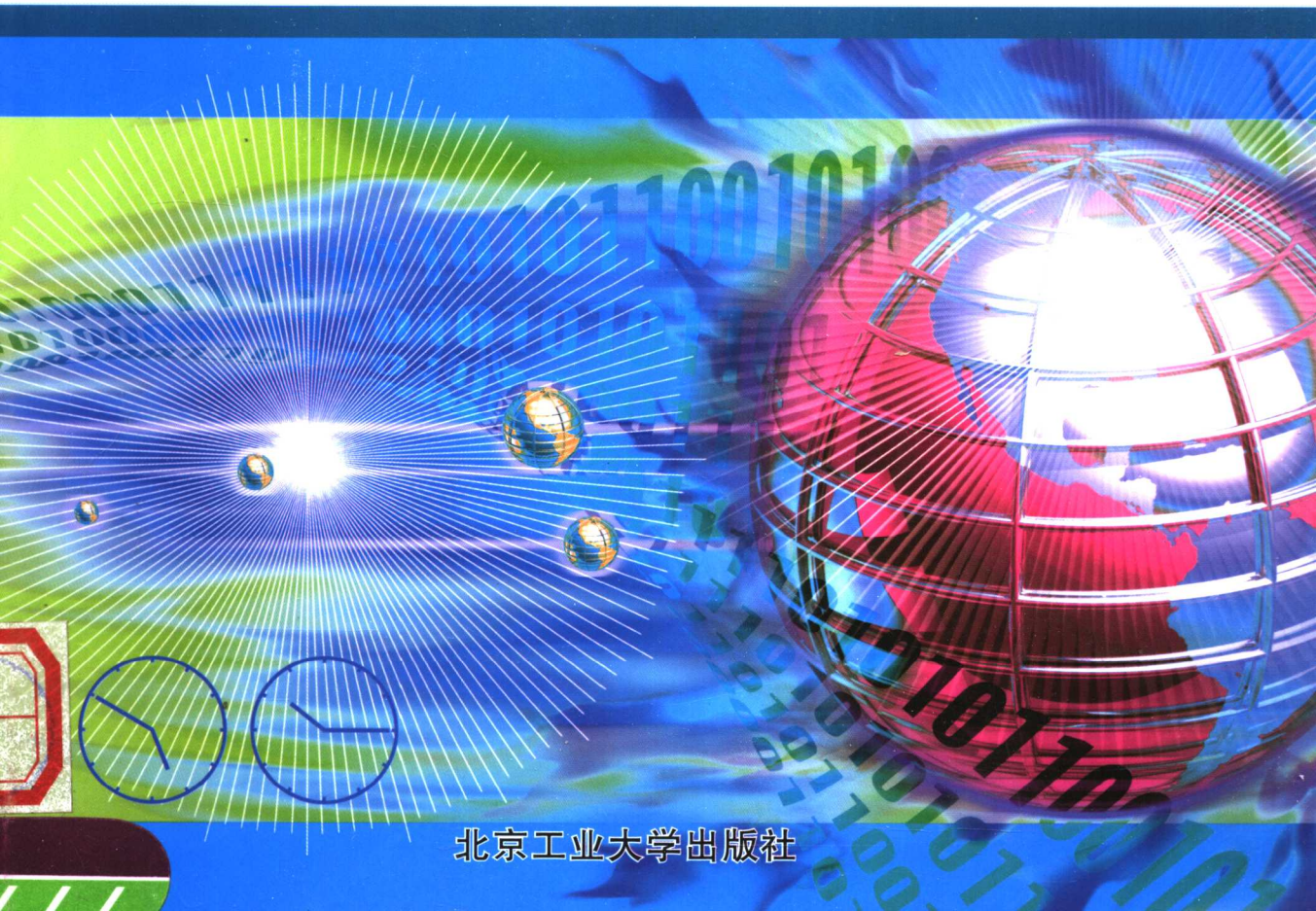
21世纪高职高专计算机系列规划教材
根据教育部最新高职高专教育教学大纲要求编写

汇编语言程序设计

武马群 主编
何福良 钟俊英 黄涛 编著



北京工业大学出版社

高职高专计算机系列规划教材

汇编语言程序设计

武马群　主编

何福良　钟俊英　黄 涛　编著

北京工业大学出版社

内 容 提 要

本书介绍汇编语言程序设计的基础知识、8086 CPU 系统组成、寻址方式及指令系统、程序设计的基本方法。

全书由 10 章组成：第 1 章介绍汇编语言基础知识（包括数制、数和字符编码、逻辑运算）；第 2 章介绍 8086 CPU 及存储器组织（包括 CPU 各部件功能、寄存器、存储器组织、标志寄存器）；第 3 章介绍寻址方式与指令系统。第 4 章介绍汇编语言源程序（包括汇编语言源程序格式、相关伪指令解释、上机调试运行过程）；第 5 章介绍分支结构程序设计（包括无条件转移指令、条件转移指令、分支程序设计举例）；第 6 章介绍循环结构程序设计（包括循环控制指令、循环程序设计举例）；第 7 章介绍子程序设计（包括子程序定义、调用与返回指令、现场保护与恢复、参数传递方法、子程序嵌套和递归）；第 8 章介绍数值运算与代码转换；第 9 章介绍宏汇编（包括宏指令、重复汇编、条件汇编、结构与记录）；第 10 章介绍输入/输出程序设计（包括输入/输出指令、输入/输出方式、中断、DOS 功能调用，BIOS 中断功能调用）。书中，每章均给出了一定数量的习题供练习使用。

本书考虑到高职、高专的教学需要，对于汇编语言中一些艰深内容进行了适当的取舍。本书可作为高职、高专院校计算机、自动控制及其他相关专业的教材和教学参考书。

图书在版编目（CIP）数据

汇编语言程序设计／何福良，钟俊英，黄涛编著.
北京：北京工业大学出版社，2005.7
（高职高专计算机系列规划教材／武马群 主编）
ISBN 7-5639-1523-0

Ⅰ.汇... Ⅱ.①何...②钟...③黄... Ⅲ.汇编语言—程序设计 Ⅳ.TP313

中国版本图书馆 CIP 数据核字（2005）第 075194 号

汇编语言程序设计

武马群 主编

何福良 钟俊英 黄 涛 编著

※

北京工业大学出版社出版发行

邮编：100022 电话：（010）67392308

各地新华书店总经销

徐水宏远印刷厂印刷

※

2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

787 mm×1 092 mm 16 开本 印张 15.25 字数 336 千字

印数：1~5 000 册

ISBN 7-5639-1523-0/T・257

定价：21.00 元

序

进入21世纪以来，随着国民经济发展水平的提高和教育改革的不断深入，我国的职业教育发展迅速，进入了一个新的历史阶段。社会主义现代化建设需要大量高素质的专业人才，而作为我国高等教育重要组成部分的高等职业教育，正肩负着前所未有的使命，为社会主义现代化建设培养大量高素质的劳动者。

区别于传统的本科教育，高等职业教育以培养应用型的人才为主。正是基于发展我国高等职业教育的需要，通过大量调研、反复讨论和修改，我们组织了一批长期工作在教学第一线的教师编写了这套《21世纪高职高专计算机系列规划教材》。

本套教材在编写上具有以下特点：

1. 具有鲜明的高职高专的特点。教材的策划和编写紧密地围绕培养技术应用性专门人才展开，体现了教育部“以应用为目的，以必需、够用为度，以讲清概念、强化应用为教学重点”的教育方针。本套书的作者都是长期从事高职高专教学工作的教师，有着丰富的教学经验，对高职高专学生的认知规律有深入的了解。本套教材适合高等职业学校、高等专科学校、以及本科院校举办的二级职业技术学院和民办职业高校使用。

2. 理论联系实际，强化应用。本套教材章后配有习题和实验题，突出实践技能和动手能力的培养。对于传统的教材，一般按照“提出概念→解释概念→举例说明”这样一种方法，先抽象后具体；本套教材采用“提出问题→解决问题→归纳总结”的方法，先具体后抽象。显而易见，后者更适合高职高专的教学模式，更能培养出具有较强综合职业能力，能够在生产、服务、技术和管理第一线工作的高素质的职业技术专门人才。

3. 适应行业技术发展，体现教学内容的先进性和前瞻性。在教材中注意突出本专业领域的新知识、新技术、新软件，尽可能实现专业教学基础性与先进性的统一。

为了方便教师教学，我们免费为使用本套教材的师生提供电子教学参考资料包：

- PowerPoint 多媒体课件
- 习题参考答案
- 教材中的程序源代码
- 教材中涉及的实例制作的各类素材

有需要的教师可以登录教学支持网站免费下载。在教材使用中有什么意见或建议也可以直接和我们联系，电子邮件地址：scqcwh@163.com。

希望本套教材，在教学实践的过程中，能够得到教师和学生的欢迎，同时期待得到更多的建议和帮助，以便提高本套教材的质量，更好地为培养社会主义现代化建设的高素质人才服务。

武马群

2005年5月

前　言

汇编语言是机器语言的符号化指令代码，是最接近计算机硬件系统、可直接操作和控制计算机系统硬件的实用程序设计语言。因此，汇编语言是一种功能强大的计算机语言。使用汇编语言编写程序，程序设计人员直接面对 CPU 内部组成和内存储器单元，可以实现对于计算机系统硬件的精确控制，在一些对于运行速度、存储空间要求较高的场合（如工业自动化控制、实时控制等），汇编语言是重要的程序设计工具。在一些系统级的程序（如操作系统、硬件接口系统等）设计中，也广泛使用汇编语言。对于使用高级程序设计语言的程序员而言，通过学习和了解汇编语言程序设计，能够深入理解 CPU 工作原理、内存储器分配和操作方式、系统输入/输出方式等内容，对于提高程序设计水平和程序调试能力也很有帮助。

《汇编语言程序设计》是计算机科学、自动控制等专业的必修课程，是《计算机组成原理》、《操作系统》等课程的先修课程。本书以常用的 IBM-PC 8086/8088 CPU 为对象进行编写，是为高职、高专院校计算机类、自动控制类专业编写的教材。本书全面地讲述 8086 宏汇编语言的基本知识和程序设计的基本方法。在编写上，考虑到高职、高专相应专业的教学要求，对部分较为艰深的内容进行适当取舍，使其更加适合高职、高专的教学实际情况。

汇编语言是所有程序设计语言中最为难懂的语言，程序调试、运行结果查看均比一般的计算机语言难。本书作者结合多年的教学实践成果和应用开发经验，在全书编写中贯穿了“在了解 CPU 内部结构、存储器组织、寻址方式、指令系统的基础上，将汇编语言当作高级语言来学习”的思想。编写了通用的数据显示子程序（见本书附录），以帮助读者在编写汇编语言程序时，能够以较为简单的方式观察程序的运行结果，从而降低学习的难度。在示例的选取和程序设计上，按照“具有一定技巧和实用价值，程序代码量不大，与教材内容紧密结合”的原则，选取一些程序简短、能直接说明问题的示例，并对程序中的数据段安排、寄存器使用分配、指令使用等进行详细的说明。各章均精心选择了一定数量的习题供读者练习。

本书第 1、2、3 章由黄涛编写，第 4、5、6 章由钟俊英编写，第 7、8、9、10 章由何福良编写，全书由何福良统稿。

本书可作为高职、高专院校计算机、自动控制及其他相关专业的教材和教学参考书。

本书由西南财经大学匡松教授进行认真的审读并提出宝贵意见，作者深表谢意。由于编者水平所限，书中难免存在错误或不妥之处，敬请读者指正。

编　者

2005 年 7 月

目　录

第 1 章　汇编语言基础知识

【学习目标】

1. 掌握各种常用进制数的表示、转换规则和运算。
2. 掌握带符号数的补码表示方法和补码的运算方法。
3. 了解计算机存取信息的基本数据类型。
4. 了解计算机中字符的表示。

1.1　汇编语言概述

计算机是一种高精度、高速度、高容量的数值计算和信息加工工具，与人类历史上其他计算工具的根本区别在于它具有逻辑判断能力、记忆能力和能够在程序的控制下自动、连续地进行数据处理。因此使用计算机进行数据处理（信息处理）必须要编写程序。编写程序的工具是各种计算机程序设计语言，这些程序设计语言是人与计算机进行交流的工具，包括机器语言、汇编语言和高级语言 3 种类型。

1.1.1　机器语言

计算机只能直接识别的是由二进制数 0 和 1 组成的代码。机器指令（Instruction）就是用二进制编码的指令，每一条机器指令控制计算机完成一个基本操作。每一种处理器（CPU）都有各自的机器指令集，某一种处理器所支持的全部指令的集合就是该处理器的指令系统，指令集（Instruction Set）及使用它们编写程序的规则被称作机器语言（Machine Language）。

机器语言程序是计算机惟一能够直接识别并执行的程序，而使用其他语言编写的程序则必须经过翻译，变换成机器语言程序才能被计算机执行，因此，机器语言程序常被称为目标程序（或目的程序）。

机器指令一般由操作码（Op code）和操作数（Operand）组成。操作码表明处理器要进行的操作，操作数表明参加操作的数据对象。一条机器指令是一组二进制代码，一个机器语言程序就是一段二进制代码序列。由于二进制表达比较繁琐，在实际中常使用对应的十六进制代码表达。例如，完成两个数据 100 和 256 相加的功能，在 8086 CPU 上用二进制表达的机器代码序列如下：

```
10111000 01100100 00000000
00000101 00000000 00000001
10100011 00000000 00100000
```

几乎没有人能够直接读懂上述程序段的功能，因为机器语言就是由看起来毫无意义的一

串 0 和 1 组成的代码。用机器语言编写程序的缺点在于难以理解、可读性差，因此容易出错，发现错误和调试程序都相当困难。因此，只有在计算机发展的早期或者不得已的情况下，才使用机器语言来编写程序。现在，除了有时在程序某处需要直接采用机器指令填充外，几乎没有人采用机器语言编写程序了。

1.1.2 汇编语言

为了克服机器语言的缺点，人们采用便于记忆、并能描述指令功能的符号来表示机器指令，表示指令操作码的符号称为指令助记符，简称助记符。助记符一般是表明指令功能的英语单词或其缩写。指令操作数同样也可以用易于记忆的符号表示。

用助记符表示的指令就是汇编语言指令。汇编语言指令以及使用它们编写程序的规则就形成汇编语言（Assembly Language）。用汇编语言书写的程序就是汇编语言程序，或称汇编语言源程序。实现 100 与 256 相加的汇编语言程序段表达如下：

```
MOV    AX，100          ；取得一个数据 100
ADD    AX，256          ；实现 100+256
MOV [2000H]，AX         ；保存结果
```

汇编语言是一种符号语言，用助记符表示操作码，汇编语言指令与机器指令之间具有一一对应的关系，因此汇编语言比机器语言容易理解和掌握，容易调试和维护，也容易被翻译成机器指令代码。汇编语言源程序已经不能由机器直接执行，而是需要被翻译成机器语言指令才可以由处理器执行，这个翻译的过程称为“汇编”，完成汇编工作的程序就是汇编程序（Assembler）。

汇编语言与机器语言都是与计算机的硬件系统紧密相关的，常被称为“面向机器的语言”，计算机的硬件系统结构不同，汇编语言和机器语言一般也不同，因此汇编语言和机器语言均不具有通用性，通常将这两种语言称为“低级语言”。

1.1.3 高级语言

汇编语言虽然较机器语言直观一些，但仍然繁琐难记。于是在 20 世纪 50 年代，人们研制出了高级程序设计语言（High-level Programming Language）。高级语言是相对于低级语言而言的，高级语言采用接近于人类自然语言的语法形式及数学表达形式，它与具体的计算机硬件无关，具有很好的通用性，更容易被掌握和使用。利用高级语言，即使一般的计算机用户也可以编写软件，而不必懂得计算机的内部结构和工作原理。当然，用高级语言编写的源程序也不会被机器直接执行，也须经过编译或解释程序的翻译才能变为机器语言程序。

目前广泛应用的高级语言有 10 多种，例如简单易用的 BASIC 语言、算法语言 FORTRAN、结构化语言 PASCAL、系统程序语言 C / C++等。

如果用高级语言表达 100 与 256 相加，可以使用标准的数学表达形式，如：X=100+256。

1.1.4 学习汇编语言的必要性

一般，对于使用计算机进行科学计算或事务处理的人员来说，只需要掌握高级语言即可，

而对于开发计算机系统软件或将计算机用于工业过程控制、实时控制的人员来说，则必须掌握汇编语言，通过分析、修改程序中的机器指令代码，扩充计算机的系统软件，增加计算机的功能。过程控制和实时控制程序则要求程序的执行和处理速度高，灵活地驱动各种设备。汇编语言是一种强有力的语言，它能透彻地反映、巧妙而充分地运用计算机硬件的功能及特点，便于编程人员根据自己的需要灵活地编制高级语言难以实现和无法实现的各种程序，随心所欲地控制计算机的运行。汇编语言是计算机能提供的最快而又最有效的语言，也是能够利用计算机所有硬件特性的惟一语言，在许多对运行速度要求很高的场合，汇编语言是必不可少的，如操作系统、编译程序、实时控制等软件多数是用汇编语言编写的。

汇编语言因机型不同（汇编程序不同）而异，但编程的原理、方法与技巧相同，只是指令的助记符号、伪指令的符号不同，数量的多少及语法不同。只要熟练掌握一种汇编语言，其他的汇编语言和高级语言则可举一反三地掌握。通过学习汇编语言，可以深入理解计算机的系统结构和工作原理，一方面可以直接使程序员用汇编语言编写高质量的、直接控制硬件的程序，另一方面也能对使用其他高级语言编写的程序员起到极大的提高作用。

1.2　进位记数制与数制转换

1.2.1　进位记数制

按进位的方法进行计数，称为进位记数制。在日常生活中通常使用十进制记数法，而在计算机内部只能使用二进制计数，在汇编语言程序设计中，为了编程书写及调试程序方便，又引入了八进制和十六进制。

1．十进制记数制

十进制数由 0～9 的 10 个数码组成，基数为 10，按“逢十进一”的原则进行计数。同一个数码在十进制数中的不同数位上，所代表的数值不同，如：十进制数 212.12 中，有 3 个数码为 2，但它们分别处在不同的数位上，它们表示的值分别为 2×10^2，2×10^0，2×10^{-2}，将 10^2，10^0，10^{-2} 称为相应数位的“权”，数位不同，其“权”的大小也不同，表示的数值也不一样。因此，一个进位记数制包含 3 个基本要素：基本符号集、基数和权值。

如果一个十进制数 D 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为：10^{n-1}，10^{n-2}，…，10^1，10^0，$10^{-1}\cdots10^{-m}$，因此，任意一个十进制数都可以表示为

$$D=\pm\sum_{i=-m}^{n-1}(D_i\times10^i)$$

式中，D_i 表示第 i 位的数码，n 为正整数。

2．二进制记数制

二进制数的基本符号集只有 0 和 1 两个数码，基数为 2，按“逢二进一”的原则进行计数。

如果一个二进制数 B 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为：2^{n-1}，2^{n-2}，…，2^1，2^0，$2^{-1}\cdots2^{-m}$，因此，任意一个二进制数都可以表示为：

$$B=\pm\sum_{i=-m}^{n-1}(B_i\times 2^i)$$

其中，B_i表示第 i 位的数码，n 为正整数。

3. 八进制记数制

八进制数由 0、1、2、3、4、5、6、7 的 8 个数码组成，基数为 8，按“逢八进一”的原则进行计数。

如果一个八进制数 O 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为：8^{n-1}，8^{n-2}，…，8^1，8^0，8^{-1}…8^{-m}，因此，任意一个八进制数都可以表示为

$$O=\pm\sum_{i=-m}^{n-1}(O_i\times 8^i)$$

式中，O_i，表示第 i 位的数码，n 为正整数。

4. 十六进制记数制

十六进制数由 0～9 的 10 个数码和 A～F 的 15 个字符组成，基数为 16，按“逢十六进一”的原则进行计数。

如果一个十六进制数 H 的整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为：16^{n-1}，16^{n-2}，…，16^1，16^0，$16^{-1}\cdots 16^{-m}$，因此，任意一个十六进制数都可以表示为

$$H=\pm\sum_{i=-m}^{n-1}(H_i\times 16^i)$$

式中，H_i表示第 i 位的数码，n 为正整数。

由上可以看出，对于任何一个 R 进制数 S，整数部分有 n 位，小数部分有 m 位，那么数位的“权”由左至右依次为：R^{n-1}，R^{n-2}，…，R^1，R^0，$R^{-1}\cdots R^{-m}$，因此，任意一个 R 进制数都可以表示为

$$S=\pm\sum_{i=-m}^{n-1}(S_i\times R^i)$$

式中，S_i表示第 i 位的数码，n 为正整数。

在计算机中，一个数可以有不同的表示，比如：18_{10}就可以表示为 10010_2、12_{16}、22_8 等。不同进制数为了表示区别，通常在数字后面加一个英文字母后缀来表示该数的数制。在汇编语言源程序中，书写十进制数时，数据尾部加一后缀 D（Decimal），但一般省略不写。二进制数用 B（Binary），八进制数用 O（Octal），十六进制数用 H（Hexadecimal）。

各种数制的对照关系如表 1-1 所示。

1.2.2　数制之间的转换

我们习惯于在日常生活中使用十进制数，但在计算机中为便于存储及计算的物理实现，内部只能使用二进制数，在编程时为表示方便，又要使用八进制数和十六进制数，这样就涉及到怎样将一种进制表示的数转换为另一种进制表示的问题。熟练掌握不同进位制数之间的相互转换是学习汇编语言、进行汇编语言程序设计的基础。

1．*R* 进制数与十进制数之间的相互转换

（1）*R* 进制数转换为十进制数

对于基数为 *R* 的数，只要将其各位数字与它的权相乘，其积相加，和数就是十进制数。这种方法称为“按权展开法”。表 1-1 列出了常用进位记数制之间的对照关系。

表 1-1　常用进位记数制对照表

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	00000000	0	0	9	00001001	11	9
1	00000001	1	1	10	00001010	12	A
2	00000010	2	2	11	00001011	13	B
3	00000011	3	3	12	00001100	14	C
4	00000100	4	4	13	00001101	15	D
3	00000101	5	5	14	00001110	16	E
6	00000110	6	6	15	00001111	17	F
7	00000111	7	7	16	00010000	20	10
8	00001000	10	8	17	00010001	21	11

[例 1.1] 1101101.01B

$=1\times2^{0}+0\times2^{1}+1\times2^{2}+1\times2^{3}+0\times2^{4}+1\times2^{5}+1\times2^{6}+0\times2^{-1}+1\times2^{-2}$

=109.3125D

[例 1.2] 3506.2O

$=6\times8^{0}+0\times8^{1}+5\times8^{2}+3\times8^{3}+2\times8^{-12}$

=1862.25D

[例 1.3] 0.2AH

$=2\times16^{-1}+10\times16^{-2}$

=0.1640625D

从上面几个例子可以看到，当从 *R* 进制转换到十进制时，可以将小数点作为起点；分别向左右两边进行转换，即对其整数部分和小数部分分别转换。对于二进制来说，只要把数位是 1 的权值相加，其和就是等效的十进制数。因此，二、十进制转换是最简单的，同时也是最常用的一种转换。

（2）十进制数转换为 *R* 进制数

将十进制数转换为基数为 *R* 的等效表示时，可将此数分成整数与小数部分分别转换，然后再使用小数点连接起来即可。

十进制整数转换成 *R* 进制的整数，可用十进制数连续除以 *R*，直至商等于 0 为止，将其

余数序列倒排序即为对应 R 进制整数。此方法称为“除 R 取余倒读数”法。

[例 1.4] 将 22D 分别转换为二进制数和十六进制数。

22÷2=11 (a_0=0)　　22÷16=1 (a_0=6)

11÷2=5 (a_1=1)　　1÷16=0 (a_1=1)

5÷2=2 (a_2=1)

2÷2=1 (a_3=0)

1÷2=0 (a_4=1)

所以 22D=10110B=16H。

十进制小数转换成 R 进制数时，可连续地乘以 R，每次记下其整数部分，并用余下的小数部乘以 R，直到小数部分为 0 或达到问题所要求的精度为止（对于小数的转换问题，绝大多数的小数都不可能转换至小数部分恰好为 0），得到的整数序列按顺序即可组成 R 进制的小数部分，这种方法称为“乘 R 取整顺读数”法。

[例 1.5] 将 0.3125D 转换成二进制数。

0.3125× 2=0.625 (b_1=0)

0.625 × 2=1.25 (b_2=1)

0.25 × 2=0.5 (b_3=0)

0.5 × 2=1 (b_4=1)

所以 0.3125D=0.0101B。

需要注意的是：十进制小数常常不能准确地换算为等值的二进制小数(或其他 R 进制数)，一般转换到要求的精度即可，因此这种换算有误差存在。

[例 1.6] 将 0.5627 转换成二进制数。

0.5627×2=1.1254 (b_4=1)

0.1254×2=0.2508 (b_2=0)

0.2508×2=0.5016 (b_3=0)

0.5016×2=1.0032 (b_4=1)

0.0032×2=0.0064 (b_5=0)

0.0064×2=0.0128

……

此过程会不断进行下去(小数位永远不会为 0)，因此只能转换到约定的精度为止，如果取 4 位小数则：0.5627D=0.1001B。

如果将十进制数 22.3125 转换成二进制数，可分别进行整数部分和小数部分的转换，然后再使用小数点将整数部分和小数部分连接在一起：

22.3125D=10110.0101B

2. 二、八、十六进制数之间的相互转换

二、八、十六进制数的相互转换在计算机应用中占有重要的地位。由于这 3 种进制的权之间存在内在联系，即 2^3=8、2^4=16，因而它们之间转换比较容易，即每 1 位八进制数相当于 3 位二进制数，每 1 位十六进制数相当于 4 位二进制数。在将二进制数转换为十六进制数或八进制数时，从小数点位置开始，按 4 位一组（十六进制）或 3 位一组（八进制）分别向左右两边延伸分组，在分组过程中中间的 0 不能省略，而在两端位数不够时可以补足 0。

[例 1.7] 将 10110.11B 转换成八进制数和十六进制数。