高等学校教材

汇编语言程序设计

钱晓捷 主编

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http://www.phei.com.cn

高等学校教材

汇编语言程序设计

钱晓捷　主编

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书以 Intel 80x86 指令系统和 MASM 6.x 为主体，全面而系统地介绍 16 / 32 位整数、浮点、多媒体指令的汇编语言程序设计方法。全书可分为基础和提高两部分。前 4 章作为基础部分，以当前“汇编语言程序设计”课程的教学为目标，为读者讲解 16 位基本整数指令及其汇编语言程序设计的知识。基础部分的主要内容是：汇编语言程序设计的基础知识、8086 指令详解、MASM 伪指令和操作符、程序格式、程序结构及其设计方法。后 5 章为提高部分，从不同的方面介绍了汇编语言程序设计的深入内容和实际应用知识。提高部分各章的内容相对独立，主要有：32 位 80x86 CPU 的整数指令系统及其编程、汇编语言与 C/C + + 的混合编程、80x87 FPU 的浮点指令系统及其编程、多媒体扩展 MMX 指令系统和 SSE 指令系统。本书各章配有丰富的习题和详细的上机指导，附录提供了 Debug、PWB 和 CodeView 开发工具的使用说明。

本书可选作高等院校《汇编语言程序设计》课程的教材或参考书，主要读者为计算机及相关学科的本、专科学生。本书内容广博、语言浅显、结构清晰、实例丰富，适合于电子、自动控制等专业的高校学生和成教学生，计算机应用开发人员，深入学习微机应用技术的普通读者等。

图书在版编目(CIP)数据

汇编语言程序设计/钱晓捷主编 . - 北京：电子工业出版社，2000.9

高等学校教材

ISBN 7-5053-6059-0

Ⅰ. 汇… Ⅱ. 钱… 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(2000)第 40851 号

丛 书 名：高等学校教材
书　　名：**汇编语言程序设计**
主　　编：钱晓捷
责任编辑：赵家鹏
特约编辑：戴　浩
排版制作：电子工业出版社计算机排版室监制
印 刷 者：天宇星印刷厂
装 订 者：河北涿州桃园装订厂
出版发行：电子工业出版社　URL：http://www.phei.com.cn
　　北京市海淀区万寿路 173 信箱　邮编 100036
经　　销：各地新华书店
开　　本：787 × 1092 1/16　印张：21　字数：531.2 千字
版　　次：2000 年 9 月第 1 版　2001 年 7 月第 3 次印刷
书　　号：ISBN 7-5053-6059-0 / TP·3207
印　　数：8 000 册　　定价：26.00 元

前　言

本书以 Intel 80x86 指令系统和 MASM 6.x 为主体，在 PC 微机 MS-DOS 环境中，以广博的内容、浅显的语言、清晰的结构和丰富的实例，全面而系统地介绍了整数指令、浮点指令、多媒体指令的汇编语言程序设计方法。全书共有 9 章，前 4 章为基础部分，后 5 章为提高部分。

第 1 章——总结性地给出了进行汇编语言程序设计所需要的基本知识，包括微型计算机系统的软硬件组成和数据表示；

第 2 章——以调试程序为实践环境，详尽地讲述了 8086 CPU 每条指令的功能与使用，其中还包括了常用 DOS/ROM-BIOS 功能的调用方法；

第 3 章——针对汇编语言源程序格式，引出程序开发、语句格式、常用伪指令和操作符、段定义等内容；

第 4 章——以程序设计技术为主线，由浅而深地详述了顺序、分支、循环、子程序、宏结构、模块化程序结构和 I/O 程序结构以及各种结构的汇编语言程序设计方法，其中包括丰富的实例；

第 5 章——首先将 16 位基本指令及编程扩展到 32 位环境，然后介绍了从 80186 开始到 Pentium Pro新增的整数指令及其应用；

第 6 章——讨论汇编语言实用的一个方面，即与高级语言 Turbo C / Visual C + + 的混合编程；

第 7 章——论述 80x87 浮点数据、浮点指令和程序设计；

第 8 章——披露 MMX 多媒体扩展指令及其编程方法；

第 9 章——简介 SSE 指令系统。

书中各章还配有相当数量的习题和具体的上机指导，附录介绍了 Debug、PWB 和 CodeView开发工具。

“汇编语言程序设计”是我国高校计算机及相关学科的一门专业基础课程。该课程的教学已有十多年的历史。但是，面对微型机技术的飞速发展，陈旧的内容和一成不变的教学方法愈来愈不能满足 21 世纪的人才需要。在总结教学经验和改革思路的基础上，我们编写了本书。它是一本全新的教材，主要体现了如下三方面的特点：

· 知识全面和教学内容全新

本书的编写参照国内高校和自学考试“汇编语言程序设计”课程的本、专科教学大纲，兼顾相关专业教学要求，既满足当前教学需要，又面向今后改革方向。

本书全面介绍 80x86 指令系统及编程，除完整的 8086/8088 指令外，还包括 32 位新增指令、浮点指令、MMX 和 SSE 指令。本书也介绍相关的汇编语言程序设计方法，例如开发大型程序需要的模块化方法、实际应用中的混合编程技术等。

本书介绍的汇编程序是微软的最新版本 MASM 6.11(升级后成为 6.14)，采用了简化段定义和完整段定义两种源程序格式，介绍了新增的流程控制伪指令等实用内容。

· 教材实用

本书的编写过程是非常严肃认真的。作者阅读了大量的相关教材、书籍和资料，每个细节都力争准确无误。全书示例中的指令、程序段和完整的源程序都经验证，并运行通过。本书原稿作为讲义经过学生试用，反映良好，并修改了已发现的错误。

全书采用浅显、明晰、循序渐进的描述方法，具有前后对照、贯穿始终的风格，加上清晰的结构、丰富的示例，使得本书不仅适合课堂教学，又适合读者自学。

全书的文字和图形都由作者录入和绘制，基本排版格式也经主编确定，使得本书的实际含量很大。加上习题、上机指导和应用软件的介绍，使用本教材无需再购买习题集或上机指导书之类的辅助书籍。

· 突出实践

全书非常强调上机实践，不仅在正文中说明如何查看代码、开发源程序、调试指令；各章还配有丰富的习题和详细的上机指导；附录介绍了调试程序 Debug、集成开发环境 PWB 和源代码级调试工具 CodeView 的使用方法。

本书的结构安排适合尽早上机实践。例如，第 2 章就引入 Debug，用于学习指令功能和调试程序段，可以使学生的上机提前到第 2 或 3 周开始；第 3 章开始就介绍完整的源程序格式，并给出了程序开发方法和程序员工作平台 PWB 集成环境。实际教学中，也可以将第 3 章 3.1 节内容提前到第 2 章的 2.6 或 2.7 节，以便更早使学生能够开发完整的源程序。

本书的内容编排也适合利用多媒体教学等现代化教学手段，讲授课程时可以一边操作一边教学。同时，本书也要求读者按照提示上机实践，这样，必将取得良好的教学效果。

本书内容自成一个完整的知识体系，无需其他先修知识。当然，读者如果具备微机软件的操作能力，可以更好地完成上机实践。读者的高级语言（如 C 或 PASCAL）程序设计知识或经验，将有助于更好地理解第 4 章的程序结构和第 6 章的混合编程。程序设计属于软件方面的内容，但由于汇编语言与硬件相关的特点，读者在计算机或微型机原理方面的知识对于深刻体会指令功能大有益处。“汇编语言程序设计”课程的相关课程是“微机原理与接口技术”，后者是微型机的硬件知识，进一步加强了汇编语言在输入输出和中断等方面的应用。两者共同为读者建立微型计算机的完整知识体系。

钱晓捷同志负责组织本书的编写、提供写作大纲，并编写第 5 章和第 7 章到第 9 章；昝红英同志编写第 4 章，穆玲玲同志编写第 2 章，邱保志同志编写第 6 章，关国利同志编写第 1 章和附录，王延年同志编写第 3 章。全书最后由主编钱晓捷进行了认真统稿，不仅理顺了各章节结构，而且内容和实例等都做到了前后衔接，还验证修改了全部例题程序。全书虽然由多人合作，但统稿后形成了统一风格。

特别值得一提的是，本书初稿完成后，各章目录、简介和部分内容，请清华大学计算机系沈美明教授进行了审阅。沈美明教授认真审阅后提供了宝贵的修改意见。初稿试用当中，编者的许多学生也指出了不少错误。本书的写作过程，还得到了许多教师和学生的支持，他们也提出了建议和意见。编者在这里一并表示感谢。

限于编者的学识水平，本书中难免有疏漏和不当之处，敬请广大同行及读者指正。同时也欢迎读者、尤其是采用本书的教师和学生，共同探讨相关教学内容、教学方法等问题。

主编的电子信箱是：qianxiaojie@371.net

钱晓捷

2000 年 5 月

目　录

第 1 章　汇编语言基础知识

程序设计语言是开发软件的工具,它的发展经历了由低级语言到高级语言的过程。汇编语言是一种面向机器的低级程序设计语言。

汇编语言以助记符形式表示每一条计算机指令,每条指令对应着计算机硬件的一个具体操作。利用汇编语言编写的程序与计算机硬件密切相关,程序员可直接对处理器内的寄存器、主存储器的存储单元以及外设的端口等进行操作,从而能够有效地控制硬件。汇编语言程序运行速度快、占用主存容量少,这些都是高级语言无法替代的。所以,作为一个计算机专业人员,必须掌握汇编语言程序设计方法。

本章介绍用汇编语言进行程序设计所需要了解的基本知识,并引出有关基本概念,希望读者熟悉。第 1 节是计算机系统的一般知识,介绍了计算机系统的硬件、软件及程序设计语言的发展,特别说明了汇编语言的特点和应用场合。第 2 节简述计算机中数据的表示,诸如二进制和十六进制、BCD 码和 ASCII 码、补码和反码、二进制运算等。本书以 Intel 80x86 系列微处理器的指令系统为例讲解汇编语言程序设计方法,因此,必须对 80x86 微处理器和以它们为核心的 PC 机系统有一定的了解,本章的第 3 和 4 节分别介绍了这两方面的内容。

1.1　计算机系统概述

计算机系统分为硬件和软件两大部分。硬件(Hardware)是计算机系统的机器部分,它是计算机工作的物质基础;软件(Software)则是为了运行、管理和维护计算机而编制的各种程序的总和,广义的软件还应该包括与程序有关的文档。利用汇编语言所编写的程序是软件;但是,每条汇编语言指令都使计算机某个具体硬件部件产生相应的动作;所以,利用汇编语言进行程序设计体现了计算机硬件和软件的结合。

1.1.1　计算机的硬件

计算机的硬件分成五大组成部件:运算器、控制器、存储器、输入设备和输出设备。其中,运算器和控制器是计算机的核心,合称中央处理单元 CPU(Central Processing Unit)或处理机、处理器(Processor);CPU 内部还有一些高速存储单元,被称为寄存器(Register)。输入设备和输出设备往往统称为外部设备(Peripheral),简称外设或 I/O 设备。在微型计算机中,CPU 由一个大规模集成电路芯片构成,被称为微处理器(Microprocessor)。

对汇编语言程序员来说,计算机各部件组成的结构如图 1.1 所示。它们主要由中央处理单元、存储器和输入输出设备组成,各部分之间通过系统总线连接。

1. 中央处理单元

中央处理单元包括运算器、控制器和寄存器组。运算器执行所有的算术和逻辑运算;控制器负责把指令逐条从存储器中取出,经译码分析后向机器发出各种控制命令,如取数、执行、运算和存数,从而正确完成程序所要求的功能。寄存器组为处理单元提供各种操作所需要的数据。

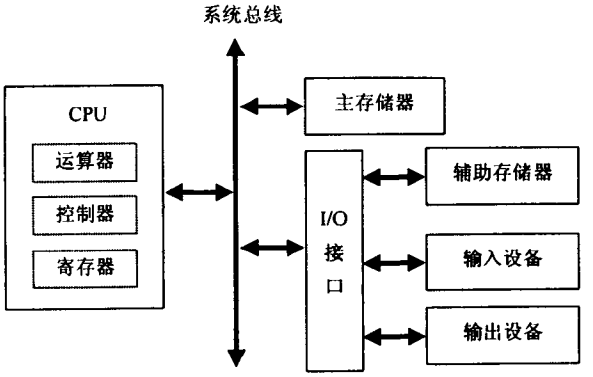

图 1.1 微型计算机的系统组成

2. 存储器

存储器(Memory)是计算机的记忆部件,它用来存放程序以及程序中所涉及的数据。存储器的内容并不因为被读出而消失;它可以重复取出,只有存入新的信息,原来的旧信息才会被更改。

按所在位置,存储器可以分成主存储器和辅助存储器,分别被简称为主存(内存)和辅存(外存)。主存储器存放当前正在执行的程序和使用的数据,CPU 可以直接存取,它由半导体存储器芯片构成,其成本高、容量小、但速度快。辅助存储器可用于长期保存大量程序和数据,CPU 需要通过 I/O 接口访问,它由磁盘或光盘构成,其成本低、容量大,但速度较慢。计算机系统中,主存的容量是有限的,需要辅存来补充;辅存的容量比主存大得多,但存取速度却比主存慢得多。一般来说,程序和数据以文件的形式保存在辅存中,只有在用到它们时,才从辅存读到主存的某个区域中,由中央处理单元控制执行。

按读写能力,存储器可以分成随机存取存储器 RAM 和只读存储器 ROM。CPU 可以从 RAM 读出信息,也可以向 RAM 写入信息,所以 RAM 也被称为读写存储器;而 ROM 中的信息只能被读出,不能被修改。RAM 型半导体存储器可以按地址随机读写,但这类存储器在断电后不能保存信息;ROM 型半导体存储器通常只能被读出,但这类存储器在断电后仍能保存信息。磁盘存储器是可读可写的,而 CD-ROM 光盘则是只读的,这两种存储器都是辅存,都可以长期保存数据,但它们的存取速度都慢于半导体存储器。

存储器由大量存储单元组成。为了区别每个单元,我们将它们编号,于是,每个存储单元就有了一个唯一的存储器地址(Address)。每个存储单元存放 1 个字节量的数据,1 个字节 B(Byte)包含了 8 个二进制位 b(bit)。

存储容量是指存储器所具有的存储单元个数,其基本单位是字节 B。为了表达更大的容量,常用的单位是 KB(千字节)、MB(兆字节)、GB(吉字节)甚至 TB(太字节)。1KB = 2^{10}字节 = 1024 字节、1MB = 2^{20}字节、1GB = 2^{30}字节、1TB = 2^{40}字节。

3. 外部设备

外部设备是实现人机交互和机间通信的一些机电设备。微机系统中,常用的输入设备有键盘、鼠标器等;输出设备有显示器、打印机等;起辅存作用的磁盘和光盘也是以外设的形式连接到系统中的,可以认为它们既是输入设备也是输出设备。

由于外设的种类繁多、工作原理各异,所以每个外设必须通过输入输出接口电路(I/O 接口)与系统连接。程序员所见的 I/O 接口由一组寄存器组成,为了区别它们,各个寄存器进行了编号,形成 I/O 地址,通常被称作 I/O 端口(Port)。系统实际上就是通过这些端口与外设进行通信的。

1.1.2 计算机的软件

软件是计算机系统的重要组成部分,它可以使机器更好地发挥其功能。软件可分为系统软件和应用软件。

1. 系统软件

系统软件是指为了方便使用、维护和管理计算机系统而编制的一类软件及其文档,它包括操作系统,语言翻译程序等。系统软件是面向计算机系统的、通常由计算机厂家提供的程序及其文档,它是用户使用计算机时为产生、准备和执行用户程序所必需的程序。

在系统软件中,最重要的软件是操作系统。操作系统负责管理整个系统的软硬件资源,向用户提供交互的界面,为所有其他程序的运行打下基础。用户借助操作系统使用计算机系统,程序员也要采用操作系统提供的驱动程序编写用户程序。

程序员采用某种程序设计语言编写源程序,利用语言翻译程序将源程序转变成可运行的程序。例如,本书介绍用汇编语言设计源程序的方法,但必须利用"汇编程序"完成源程序的翻译工作。高级语言则采用编译类或解释类程序来完成这个工作。

2. 应用软件

应用软件是解决某一问题的程序及其文档。它覆盖了计算机应用的所有方面,每个应用都有相应的应用程序。

微机系统具有多种多样的应用软件。例如,进行程序设计时要采用文本编辑软件编写源程序,带有丰富格式的字处理软件帮助你书写文章,排版软件则用于书刊出版。

大型的程序设计项目往往要借助软件开发工具(包)。这个开发工具是进行程序设计所用到的各种软件的有机集合,所以也被称为集成开发环境。其中,有文本编辑器,有语言翻译程

序,有用于形成可执行文件的连接程序,还组合有进行程序排错的调试程序等。

1.1.3 计算机的程序设计语言

进行程序设计的语言有很多,可以分成低级语言和高级语言。低级语言有机器语言和汇编语言,高级语言有 C/C++、PASCAL、BASIC 等。

1. 机器语言

计算机能够直接识别的是二进制数 0 和 1 组成的代码。机器指令(Instruction)就是用二进制编码的指令,一条机器指令控制计算机完成一个操作。每种处理器都有各自的机器指令集,某处理器支持的所有指令的集合就是该处理器的指令系统(Instruction Set)。指令集及使用它们编写程序的规则被称作机器语言(Machine Language)。

用机器语言形成的程序是计算机唯一能够直接识别并执行的程序,而用其他语言编写的程序必须经过翻译、变换成机器语言程序;所以,机器语言程序常称为目标程序(或目的程序)。

机器指令一般由操作码(Opcode)和操作数(Operand)构成。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。因为二进制表达比较繁琐,常用对应的十六进制形式表达。例如,完成两个数据 100 和 256 相加的功能,在 8086CPU 上用十六进制表达的代码序列如下:

```
B8 64 00
05 00 01
A3 00 20
```

几乎没有人能够直接读懂该程序段的功能,因为机器语言就是看起来毫无意义的一串代码。用机器语言编写程序的最大缺点是难以理解,因而极易出错,也难以发现错误。所以,只是在计算机发展的早期或不得已的情况下,才用机器语言编写程序。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。

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格式指令。汇编格式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 MASM 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      ;保存和
```

这时候,如果熟悉了有关助记符及对应指令的功能,就可以读懂上述程序段了。

汇编语言是一种符号语言,它用助记符表示操作码,比机器语言容易理解和掌握,也容易调试和维护。但是,汇编语言源程序要翻译成机器语言程序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)。

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)。高级语言比较接近于人类自然语言的语法习惯及数学表达形式,它与具体的计算机硬件无关,更容易被广大计算机工作者掌握和使用。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结构和工作原理。当然,用高级语言编写的源程序不会被机器直接执行,而需经过编译或解释程序的翻译才可变为机器语言程序。

广泛应用的高级语言有十多种,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++等等。用高级语言表达100与256相加,就是通常的数学表达形式:100 + 256。

4. 汇编语言程序设计的意义

高级语言简单、易学,而汇编语言复杂、难懂,是否就没有必要再采用汇编语言了呢?让我们首先比较一下汇编语言和高级语言的特点。

• 汇编语言与处理器密切相关。每种处理器都有自己的指令系统,相应的汇编语言各不相同。所以,汇编语言程序的通用性、可移植性较差。相对来说,高级语言与具体计算机无关,高级语言程序可以在多种计算机上编译后执行。

• 汇编语言功能有限,又涉及寄存器、主存单元等硬件细节,所以编写程序比较繁琐,调试起来也比较困难。高级语言提供了强大的功能,采用类似自然语言的语法,所以容易被掌握和应用,也不必关心诸如标志、堆栈等琐碎问题。

• 汇编语言本质上就是机器语言,它可以直接地、有效地控制计算机硬件,因而容易产生运行速度快、指令序列短小的高效率目标程序。高级语言不易直接控制计算机的各种操作,编译程序产生的目标程序往往比较庞大、程序难以优化,所以运行速度较慢。

可见汇编语言的主要优点就是可以直接控制计算机硬件部件,可以编写在"时间"和"空间"两方面最有效的程序。这些优点使得汇编语言在程序设计中占有重要的位置,是不可被取代的。汇编语言的缺点也是明显的。它与处理器密切有关,要求程序员比较熟悉计算机硬件系统、考虑许多细节问题,导致编写程序繁琐,调试、维护、交流和移植困难。因此,有时可以采用高级语言和汇编语言混合编程的方法,互相取长补短,更好地解决实际问题。

汇编语言的主要应用场合有:

• 程序要具有较快的执行时间,或者只能占用较小的存储容量。例如,操作系统的核心程序段,实时控制系统的软件,智能仪器仪表的控制程序等。

• 程序与计算机硬件密切相关,程序要直接、有效地控制硬件。例如,I/O接口电路的初始化程序段,外部设备的低层驱动程序等。

• 大型软件需要提高性能、优化处理的部分。例如,计算机系统频繁调用的子程序、动态

连接库等。

• 没有合适的高级语言或只能采用汇编语言的时候。例如,开发最新的处理器程序时,暂时没有支持新指令的编译程序。

• 汇编语言还有许多实际应用,例如分析具体系统尤其是该系统的低层软件、加密解密软件、分析和防治计算机病毒等等。

1.2 数据表示

计算机只能识别0/1编码,进入计算机的任何信息都要转换成0/1编码。本节说明计算机如何利用0/1表示现实当中的各种数值、字符等内容。

1.2.1 数制

人们已经习惯了用十进制计数,但计算机则以二进制的形式表达数值,为了便于表示二进制数,人们又常用到十六进制数。

1. 二进制数

计算机中为便于存储及计算的物理实现,采用二进制数。二进制数的特点为:逢二进一,由0、1两个数码组成,基数为2,各个位权以2^k表示。

二进制数$a_n a_{n-1}\cdots a_1 a_0 . b_1 b_2 \cdots b_m$可表示为:

$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots + b_m \times 2^{-m}$

其中a_i、b_j非0即1。

2. 十六进制数

由于二进制数书写较长、难以记忆,因此常用易于与二进制数转换的十六进制数来描述二进制数。十六进制数的基数是16,共有16个数码:0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F(也可以使用小写字母a ~ f),逢十六进位,各个位的位权为16^k。

十六进制数$a_n a_{n-1}\cdots a_1 a_0 . b_1 b_2 \cdots b_m$可表示为:

$a_n \times 16^n + a_{n-1} \times 16^{n-1} + \cdots + a_1 \times 16^1 + a_0 \times 16^0 + b_1 \times 16^{-1} + b_2 \times 16^{-2} + \cdots + b_m \times 16^{-m}$

其中a_i、b_j为0 ~ F中的一个数码。

汇编语言中通常用字母B或b结尾表示一个数据采用二进制(Binary),用字母H或h结尾表示采用十六进制(Hexadecimal)。十进制(Decimal)数据可以用字母D或d结尾,以示强调或区别;也可以不加结尾字母。

3. 数制之间的转换

• 十进制数的整数部分转换为二进制和十六进制数可用除法，把要转换的十进制数的整数部分不断除以二进制和十六进制数的基数 2 或 16，并记下余数，直到商为 0 为止。由最后一个余数起逆向取各个余数，则为该十进制数整数部分转换成的二进制和十六进制数。

例 1.1：十进制整数转换为二进制和十六进制

126D = 01111110B = 7EH

• 十进制数的小数部分转换为二进制和十六进制数则分别乘以各自的基数，记录整数部分，直到小数部分为 0 为止。

例 1.2：十进制小数转换为二进制和十六进制

0.8125D = 0.1101B = 0.CH

• 二进制、十六进制数转换为十进制数可分别套用各自的公式。

例 1.3：二进制和十六进制数转换为十进制数

$0011.1010B = 1\times2^{1}+1\times2^{0}+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}= 3.125D$

$1.2H = 1\times16^{0}+2\times16^{-1}= 1.125D$

• 二进制和十六进制数之间具有对应关系：每四个二进制位对应一个十六进制位，如表 1.1 所示，所以相互转换非常简单。

例 1.4：二进制和十六进制数相互转换

00111010B = 3AH

F2H = 11110010B

表 1.1　不同进制间与 BCD 码的对应关系

十进制	二进制	十六进制	BCD 码
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	8
9	1001	9	9
10	1010	A	
11	1011	B	
12	1100	C	
13	1101	D	
14	1110	E	
15	1111	F	