



普通高等教育“十一五”国家级规划教材



免费提供
电子教案

高等院校规划教材
计算机科学与技术系列

80x86汇编语言程序设计

第2版

马力妮 主编



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十一五”国家级规划教材
高等院校规划教材·计算机科学与技术系列

80x86 汇编语言程序设计

第2版

马力妮 主编
施运梅 关静丽 侯凌燕 参编



机械工业出版社

本书以 Intel 8086/8088 CPU 的指令为主, 以实模式下 80x86 指令为辅, 较系统地介绍了汇编语言的基本理论和程序设计方法, 主要内容包括: 汇编语言程序设计基础知识, 80x86 的寻址方式和指令系统, 汇编语言的程序框架及常用的伪指令和操作符, 汇编语言程序设计的基本技术, 高级汇编技术, 利用汇编语言完成的 I/O 程序设计技术, 汇编语言与 C 语言的混合编程以及汇编语言程序设计实验方法。各章内容重点突出、结构清晰、简洁易懂, 并附有一定数量的以考题形式为主的习题。

本书是高等院校计算机及相关专业本科教学用书, 也可作为自学考试、学历文凭考试的参考教材。

图书在版编目 (CIP) 数据

80x86 汇编语言程序设计/马力妮主编.—2 版.—北京: 机械工业出版社, 2009.6

普通高等教育“十一五”国家级规划教材. 高等院校规划教材. 计算机科学与技术系列

ISBN 978-7-111-27286-1

I. 8… II. 马… 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 (2009) 第 083036 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 刘亚军 版式设计: 霍永明 责任校对: 李秋荣

封面设计: 李 瞳 责任印制: 邓 博

北京机工印刷厂印刷 (兴文装订厂装订)

2009 年 8 月第 2 版第 1 次印刷

184mm × 260mm · 25 印张 · 616 千字

0 001—3 000 册

标准书号: ISBN 978-7-111-27286-1

定价: 42.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379739

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师们能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

根据国家教委的要求，当前的计算机领域的教学改革必须紧跟计算机技术的最新发展动态，用最新的知识、教材、手段结合学生的实际情况进行教学，用科学性并简单易懂、生动活泼的形式进行教学，培养学生的自学能力、独立思考能力、分析和解决问题的能力。

本书编写的主要特点是突出理论性、实践性、先进性、通俗性，力求自学方便，适用于普通高等院校，使学生在尽量短的时间内熟练掌握最基本的汇编语言的功能、用法和编程技巧。

根据 8086/8088 微处理器在 80x86 及 Pentium 微处理器的基础地位，在指令系统中仍以 8086/8088 指令为主，以 80x86 及 Pentium 指令为辅，但在实例中尽量采用 80x86 及 Pentium 指令，使读者在应用中是以 80x86 及 Pentium 的指令为主线，因此在实验指导中用汇编程序的 2 种结构方法和 3 种上机环境来介绍上机过程，读者可根据需求选择实验环境。

本书力求重点突出、语言简洁、举一反三，为避免汇编指令的集中堆砌，将部分指令在相关程序设计中介绍。

在汇编程序设计方法中强调段定义的框架结构，即完整段定义和简化段定义框架，降低了汇编程序设计的复杂度使读者容易掌握汇编程序设计方法。

在实例中采用 CodeView 调试工具来说明空间域的概念，使读者能清楚地了解地址结构、内存结构、寄存器结构和程序的执行过程。

在编程举例中分实例分析、流程图、根据流程图编写程序、程序说明 4 个步骤进行讲解，突出了重点和难点以及汇编程序的编程技巧，使读者能更深入地掌握汇编程序编写的特点，从而使教材便于自学，解决了汇编程序设计课程的教学内容多、学时少、难学的问题。

根据当前汇编程序使用的编译和调试工具的情况，在汇编语言程序设计实验方法中介绍 3 种编译调试工具：其一是 MASM 5.0 编译器、DEBUG 调试程序，主要用于早期的 8086/8088、80286 系统（16 位机）的编译和调试；其二是 Microsoft MASM 6.11 软件包，用于 80386 以上（包括 16 位、32 位机）系统的编译和调试，PWB 集成编译器和 CodeView 调试器；其三也是用于 80386 以上（包括 16 位、32 位机）汇编程序的集成开发环境——RadASM + MASM32。读者可根据需要选择开发环境。

本书提供所有例题的源程序和电子教案需要者可在机械工业出版社网站 www.cmpedu.com 下载。

本书由马力妮主编。第 1 版第 1~5 章由马力妮编写，同时参与其他各章部分内容的编写；第 6、7 章由侯凌燕编写；第 8、9 章由关静丽编写；第 10、11 章由施运梅编写。第 2 版由马力妮对第 1 版进行修订并增加了 RadASM + MASM32 编译器的使用方法，在该环境中将书中所有例题进行调试通过。

由于作者水平有限，不足或错误在所难免，恳请读者批评指正。

作 者

目 录

出版说明

前言

第 1 章 基础知识	1
1.1 汇编语言	1
1.1.1 机器语言与汇编语言	1
1.1.2 高级语言	2
1.1.3 汇编语言的特点	2
1.1.4 汇编语言与高级语言的比较	3
1.2 进位记数制及转换方法	3
1.2.1 常用的进位记数制	3
1.2.2 数制转换	5
1.3 计算机中的数据组织	10
1.3.1 数据组织方式	10
1.3.2 数的表示方法	11
1.3.3 字符的表示方法	16
1.4 二进制数的算术运算和逻辑运算	18
1.4.1 算术运算	18
1.4.2 逻辑运算	19
1.5 习题	20
第 2 章 80x86 计算机组织结构	22
2.1 80x86 计算机的基本结构	22
2.1.1 中央处理器	22
2.1.2 总线结构	22
2.1.3 存储器	23
2.1.4 外围设备	24
2.2 80x86 CPU 的寄存器	24
2.2.1 通用寄存器	24
2.2.2 控制寄存器	27
2.2.3 段寄存器	30
2.3 内存组织结构	30
2.3.1 内存的地址与字节、字的存放	30
2.3.2 堆栈	32
2.4 80x86 CPU 的工作模式	33
2.4.1 实模式	33
2.4.2 保护模式	37
2.5 存储器管理机制	38
2.5.1 分段管理机制	39

2.5.2 分页管理机制	39
2.6 习题	40
第3章 80x86 指令系统和寻址方式	42
3.1 指令格式	42
3.1.1 指令的书写格式	42
3.1.2 提供操作数的方式	43
3.2 寻址方式	45
3.2.1 与数据有关的寻址方式	45
3.2.2 32 位 CPU 的寻址方式	54
3.2.3 与转移地址有关的寻址方式	57
3.3 指令系统	61
3.3.1 数据传送指令	61
3.3.2 算术运算指令	71
3.4 习题	91
第4章 汇编语言程序结构	95
4.1 汇编语言程序的基本结构	95
4.1.1 汇编语言程序的基本框架	95
4.1.2 汇编语言程序的指令格式	97
4.2 汇编语言的数据	98
4.2.1 常量	98
4.2.2 变量	98
4.2.3 标号	101
4.2.4 变量定义时用到的操作符和表达式	102
4.3 运算符与表达式	105
4.3.1 算术运算符	105
4.3.2 逻辑运算符	105
4.3.3 关系运算符	106
4.3.4 数值回送操作符	106
4.3.5 属性操作符	109
4.3.6 移位运算符	111
4.3.7 运算符的优先级	112
4.4 伪指令	112
4.4.1 完整段定义伪指令	112
4.4.2 处理器选择伪指令	117
4.4.3 简化段定义伪指令	118
4.4.4 定位伪指令	120
4.4.5 赋值伪指令	120
4.4.6 LABEL 伪指令	121
4.4.7 对准伪指令	122
4.4.8 标题伪指令	123
4.5 习题	123

第 5 章 顺序、分支和循环程序设计	126
5.1 顺序程序设计	126
5.2 分支程序设计	131
5.2.1 转移指令	131
5.2.2 条件转移指令	132
5.2.3 根据条件设置字节指令	138
5.2.4 分支程序设计的应用举例	140
5.3 循环程序设计	147
5.3.1 循环程序结构	147
5.3.2 循环控制指令	148
5.3.3 单重循环的应用举例	166
5.3.4 多重循环的应用举例	176
5.4 习题	181
第 6 章 子程序设计	185
6.1 子程序的结构	185
6.1.1 子程序的定义	185
6.1.2 子程序的调用与返回指令	186
6.1.3 子程序的编程原则	192
6.1.4 参数的传递	194
6.2 子程序的嵌套与递归	206
6.2.1 子程序的嵌套调用	206
6.2.2 子程序的递归	210
6.3 子程序应用举例	215
6.4 习题	221
第 7 章 高级汇编技术	223
7.1 宏汇编	223
7.1.1 宏定义	223
7.1.2 宏调用和宏展开	223
7.1.3 宏定义和调用时的参数使用	225
7.1.4 宏定义的伪指令	228
7.1.5 宏库的建立和使用	229
7.2 重复汇编	232
7.3 条件汇编	234
7.3.1 条件汇编伪指令	234
7.3.2 使用条件汇编伪指令的程序结构	235
7.4 习题	236
第 8 章 输入/输出与中断	240
8.1 输入/输出设备的数据传送方式	240
8.1.1 I/O 端口	240
8.1.2 I/O 指令	241
8.1.3 I/O 设备的数据传送方式	243

8.2	80x86 的中断系统	246
8.2.1	中断的基本概念	246
8.2.2	中断指令	247
8.2.3	中断分类	248
8.2.4	中断优先级和中断嵌套	250
8.2.5	中断向量表	250
8.2.6	中断过程	252
8.2.7	如何编写中断处理程序	253
8.3	BIOS 与 DOS 中断系统	258
8.3.1	BIOS 与 DOS 中断调用概述	258
8.3.2	键盘 I/O 程序设计	260
8.3.3	显示器 I/O 程序设计	267
8.3.4	BIOS 及 DOS 的时间功能调用	278
8.3.5	图形系统程序设计	282
8.3.6	发声系统程序设计	292
8.4	习题	296
第 9 章	磁盘文件存取方法	299
9.1	以文件句柄方式存取文件	299
9.1.1	DOS 文件功能调用	299
9.1.2	路径和 ASCII 串	300
9.1.3	文件句柄、错误代码和文件属性	301
9.1.4	文件的打开方式	302
9.1.5	读写指针的移动	303
9.2	文件操作举例	304
9.3	习题	314
第 10 章	C/C++ 与汇编语言的混合编程方法	317
10.1	嵌入式汇编	317
10.1.1	嵌入式汇编的格式	317
10.1.2	几点说明	320
10.1.3	编译链接的方法	321
10.2	C/C++ 程序调用汇编源程序	321
10.2.1	一个简单的小例子	322
10.2.2	编译链接的方法	322
10.2.3	C/C++ 程序对汇编源程序的调用规则	324
10.2.4	MASM 6.11 环境下参数传递的变化	332
10.3	习题	335
第 11 章	汇编语言程序设计实验方法	336
11.1	汇编语言上机步骤	336
11.2	用 MASM 5.0 汇编、链接程序	337
11.2.1	运行汇编程序必备的条件	337
11.2.2	执行汇编程序	338

11.2.3 执行链接程序	341
11.3 DEBUG 调试器的使用	342
11.3.1 DEBUG 的主要特点	342
11.3.2 通过 DEBUG 编写、运行汇编程序	342
11.3.3 DEBUG 的进入	344
11.3.4 DEBUG 的主要命令	344
11.3.5 举例	346
11.4 Microsoft MASM 6.11 软件包的使用	348
11.4.1 安装 MASM 6.11	348
11.4.2 安装后的任务	350
11.4.3 PWB 系统的进入和退出	350
11.4.4 PWB 主菜单	350
11.4.5 设置 PWB 的开发环境	351
11.4.6 PWB 应用实例	353
11.5 CodeView 调试器的使用	355
11.5.1 CodeView 调试器的进入和退出	355
11.5.2 CodeView 调试器主窗口及功能键	356
11.5.3 CodeView 功能键	357
11.5.4 CodeView 环境设置	358
11.5.5 用 CodeView 调试汇编程序	358
11.5.6 在 Command 窗口调试程序	360
11.6 RadASM + MASM32 集成开发环境	361
11.6.1 MASM32 的安装	361
11.6.2 链接器和调试器的安装	364
11.6.3 RadASM 的安装	364
11.6.4 配置 RadASM 编译环境	366
11.6.5 RadASM 应用实例	368
11.6.6 修改 ini 配置文件	373
附录	374
附录 A 80x86 指令系统一览表	374
附录 B ASCII 码表	384
附录 C DOS 系统功能调用 (INT 21H)	385
附录 D BIOS 功能调用	386

第1章 基础知识

汇编语言是一种面向机器的语言，是程序设计语言的基础。本章重点介绍用汇编语言编写程序时所需要的基础知识。学习本章后要求了解汇编语言的基本概念、汇编语言的特点、汇编语言与高级语言的关系、掌握不同数制之间数字的转换及计算机中数据的组织形式。

1.1 汇编语言

人与计算机打交道是通过程序设计语言进行的。通常，程序设计语言分为3类：机器语言、汇编语言和高级语言。其中，机器语言和汇编语言是靠近机器的语言。因此，汇编语言是一种初级语言，是一种面向机器的程序设计语言，它的基本内容是机器语言符号化的描述。

尽管程序设计语言的发展很快，但汇编语言是程序设计语言的基础语言，在程序设计语言中占有一定的地位，是唯一可以直接与计算机硬件打交道的语言。

1.1.1 机器语言与汇编语言

1. 机器语言

由于计算机只能接受由“0”或“1”组成的二进制编码指令和数据，因此以二进制形式的指令组成的指令集合称为计算机的机器语言，它是计算机唯一能够直接识别并接受的语言。

例如：计算 $z = x + y$ ，其中 $x = 5$ ， $y = 6$ 。为了读起来方便，用十六进制代替二进制来表示的机器语言指令为：

```
A00000 ; 将变量 x 中的内容 5 送到寄存器 AL 中。  
02060100 ; 将 AL 中的内容 5 加变量 y 中的内容 6，结果送寄存器 AL 中。  
A20200 ; 将最终结果送变量 z 中。
```

如果没有注释，是很难读懂用机器语言编写的机器语言指令的。由此可以看到，用机器语言指令编写程序是不容易的。

2. 汇编语言

早期的计算机是用机器语言来编写程序的，但是，用机器语言编写程序很不方便并容易出错，编写出来的程序也难以修改和调试。为此，出现了代替机器语言的助记符语言，顾名思义，就是用简单且容易记住的符号来代替用“0”或“1”表示的机器语言，这就是汇编语言。如上例：计算 $z = x + y$ ，其中 $x = 5$ ， $y = 6$ 。

机器语言（用十六进制表示）	汇编语言
A00000H	MOV AL, x
02060100H	ADD AL, y

通常，汇编语言的执行语句与机器语言的指令是一一对应的关系，即汇编语言的一个执行语句对应一条机器语言指令。

3. 汇编语言源程序与汇编程序

汇编语言是建立在机器语言之上的，由于计算机不能够直接识别这种符号语言，所以用汇编语言编写的程序必须翻译成机器语言后才能执行。

用汇编语言书写的程序称为汇编语言源程序。汇编语言源程序是在编辑程序中形成的，而把源程序转换为相应的目标程序（即机器语言程序）的翻译程序称为汇编程序（汇编器）。这个翻译过程称为汇编（宏汇编）。

因此，汇编语言源程序也和其他高级语言的源程序一样必须由翻译程序翻译成代码程序（即机器语言程序）才能在机器中执行。翻译成的机器语言代码程序称为相应源程序的目标代码。

目标代码程序是汇编源程序经翻译后的程序，并不是可执行程序，是和高级语言一样须要经过连接程序将本程序所需要的其他目标代码连接定位形成可执行文件，如图 1-1 所示。

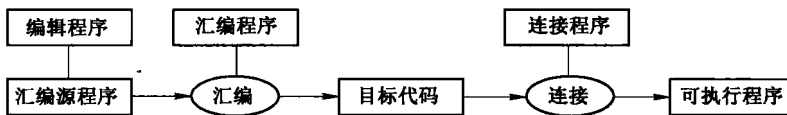


图 1-1 程序的编辑、汇编与连接过程

1.1.2 高级语言

高级语言包括面向过程语言和面向对象语言。面向过程语言，采用的是人们所熟悉的、便于记忆的和容易理解的结构化的语言，面向过程语言的出现为程序设计带来了方便；而面向对象语言是从面向过程语言发展而来的，它改变了编程者的思维方式，使编程者从结构化程序设计到面向对象的程序设计，也就是把数据和处理数据的过程当作一个整体来处理，这样使程序设计更接近于自然语言，设计过程更简单，设计出来的软件质量更高。当前，最常用的高级语言有 Basic、Pascal、C、C++、Java、Visual C++ 等。

1.1.3 汇编语言的特点

尽管高级语言具有一定的优势，具有程序设计简单、代码可移植性强等特点，但汇编语言是程序设计语言的基础语言，在程序设计语言中仍占有一定的地位，是唯一可以直接与计算机硬件打交道的语言。因此，在计算机底层设计中仍采用汇编语言进行设计。例如，使用 Visual C++ 6.0 编译系统时，在调试过程中就有源程序对应的汇编代码窗口，在这个窗口中可设置断点调试 VC 程序。与高级语言相比，汇编语言有如下特点。

1. 汇编语言与机器指令一一对应，可充分理解计算机的操作过程

由于汇编语言指令是机器指令的符号表示，所以汇编指令与机器指令有着一一对应的关系，因此学会汇编语言，在程序设计过程中，就可以充分理解指令在计算机中的执行过程，

为进一步学习程序设计打下良好的基础。

2. 汇编语言是靠近机器的语言

汇编语言面向机器，与机器关系密切，编程时要求熟悉机器硬件系统，因此编写程序繁琐，调试、维护、移植程序困难。正是由于汇编语言与机器关系密切，在汇编程序设计中，可以充分利用机器硬件的全部功能，发挥机器的特点。可最直接和最有效地操纵机器硬件系统，这是高级语言难以完成的。

目前在计算机系统中，某些功能仍靠汇编语言程序来实现。例如，实时过程控制系统、系统的初始化，实际的输入输出设备的操作，就是使用汇编语言来完成的。

3. 汇编语言程序的效率高于高级语言

这里的效率是指用汇编语言编写的源程序在汇编后所得的目标程序效率高。这种目标程序的高效率反映在时间域和空间域两个方面，即运行速度快、目标代码短占用存储空间少。在采用相同算法的前提下，任何高级语言程序在这两个方面的效率都不如汇编语言程序。

汇编语言程序能获得时间和空间高效率的主要原因是：构成汇编语言的汇编指令是机器指令的符号表示，汇编指令和机器指令是一一对应的，另外汇编语言程序能充分利用机器硬件系统的许多特性。

由于汇编语言比高级语言更接近于机器语言，因此能透彻地反映计算机硬件的功能特点，便于使用者灵活地根据自己的要求编制最经济（节省时间，节省内存）的程序。

1.1.4 汇编语言与高级语言的比较

1. 程序

高级语言：面向问题。

汇编语言：面向机器。

2. 软件开发

高级语言：节省软件开发时间，但不允许程序员直接使用微处理器的集成电路芯片的许多特性（寄存器、标志等）。

汇编语言：程序编写比高级语言困难，要首先了解计算机的内部结构，才能充分发挥汇编语言的作用。

3. 代码生成

高级语言：高级语言编译后产生的目标代码程序比较长。

汇编语言：汇编语言编译后产生的目标代码程序短，因而，汇编语言程序执行速度快，所占内存少。

尽管汇编语言的调试时间长，程序设计技巧强，但对于某些应用场合仍采用汇编语言，所以系统软件设计者和实时控制程序设计者，应该能熟练使用汇编语言编写程序和调试程序。

1.2 进位记数制及转换方法

1.2.1 常用的进位记数制

数制包括进位记数制和非进位记数制，在进位记数制中是按进位的方式记数的，如十进

制。在非进位记数制中，用不同的符号表示不同的数据，如罗马数字 III、IV 等。

1. 十进制记数法

特点：

- 1) 它的数值部分是用 10 个不同的数值符号 0~9 来表示的。
- 2) 逢十进一。

$$\begin{aligned} \text{【例 1-1】 } 348.573 &= 3 \times 100 + 4 \times 10 + 8 + 5 \times 1/10 + 7 \times 1/100 + 3 \times 1/1000 \\ &= 3 \times 10^2 + 4 \times 10^1 + 8 \times 10^0 + 5 \times 10^{-1} + 7 \times 10^{-2} + 3 \times 10^{-3} \end{aligned}$$

该表达式是根据十进制数所在的位置来记数的，其中 10^n 为权，即每一位基数的幂为权，十进制的基数就是 10。

一般表达式：

$$\begin{aligned} D &= D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \cdots + D_1 \times 10^1 + D_0 \times 10^0 \\ &\quad + D_{-1} \times 10^{-1} + D_{-2} \times 10^{-2} + \cdots + D_{-m} \times 10^{-m} \\ &= \sum_{i=n-1}^{-m} D_i \times 10^i \end{aligned}$$

式中， n 为整数位数； m 为小数位数； $D_i = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ 。

2. 二进制记数法

特点：

- 1) 它的数值部分用两个不同的数值符号 0 或 1 来表示。
- 2) 逢二进一。

$$\begin{aligned} \text{【例 1-2】 } (11011.101)_2 &= (1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})_{10} \\ &= (16 + 8 + 2 + 1 + 0.5 + 0.125)_{10} = (27.6125)_{10} \end{aligned}$$

通过此式可以看到，完成了二进制转换成十进制的转换过程。这种转换方法为按权展开法。

一般表达式：

$$\begin{aligned} B &= B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \cdots + B_1 \times 2^1 + B_0 \times 2^0 \\ &\quad + B_{-1} \times 2^{-1} + B_{-2} \times 2^{-2} + \cdots + B_{-m} \times 2^{-m} \\ &= \sum_{i=n-1}^{-m} B_i \times 2^i \end{aligned}$$

式中， n 为整数位数； m 为小数位数； $B_i = \{0, 1\}$ 。

3. 八进制记数法

特点：

- 1) 它的数值部分用 8 个不同的数值符号 0, 1, 2, 3, ..., 7 来表示。
- 2) 逢八进一。

$$\begin{aligned} \text{【例 1-3】 } (327)_8 &= (3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0)_{10} \\ &= (3 \times 64 + 2 \times 8 + 7)_{10} \\ &= (215)_{10} \end{aligned}$$

用按权展开法将八进制的 327O 转换成十进制的 215D。

一般表达式：

$$\begin{aligned}
 Q &= Q_{n-1} \times 8^{n-1} + Q_{n-2} \times 8^{n-2} + \cdots + Q_1 \times 8^1 + Q_0 \times 8^0 \\
 &\quad + Q_{-1} \times 8^{-1} + Q_{-2} \times 8^{-2} + \cdots + Q_{-m} \times 8^{-m} \\
 &= \sum_{i=n-1}^{-m} Q_i \times 8^i
 \end{aligned}$$

式中， n 为整数位数； m 为小数位数； $Q_i = \{0, 1, 2, 3, 4, 5, 6, 7\}$ 。

4. 十六进制记数法

特点：

1) 它的数值部分用 16 个不同的数值符号 0, 1, 2, 3, ..., 9, A, B, C, D, E, F 来表示。

2) 逢十六进一。

【例 1.4】将十六进制数 327.A 转换成十进制数。

$$\begin{aligned}
 (327.A)_{16} &= (3 \times 16^2 + 2 \times 16^1 + 7 \times 16^0 + 10 \times 16^{-1})_{10} \\
 &= (768 + 32 + 7 + 0.625)_{10} \\
 &= (807.625)_{10}
 \end{aligned}$$

用按权展开法将十六进制的 327.AH 转换成十进制的 807.625D。

一般表达式：

$$\begin{aligned}
 H &= H_{n-1} \times 16^{n-1} + H_{n-2} \times 16^{n-2} + \cdots + H_1 \times 16^1 + H_0 \times 16^0 \\
 &\quad + H_{-1} \times 16^{-1} + H_{-2} \times 16^{-2} + \cdots + H_{-m} \times 16^{-m} \\
 &= \sum_{i=n-1}^{-m} H_i \times 16^i
 \end{aligned}$$

式中， n 为整数位数； m 为小数位数； $H_i = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ 。

5. 进位记数制小结

一般表达式：

$$\begin{aligned}
 N &= N_{n-1} \times J^{n-1} + N_{n-2} \times J^{n-2} + \cdots + N_1 \times J^1 + N_0 \times J^0 \\
 &\quad + N_{-1} \times J^{-1} + N_{-2} \times J^{-2} + \cdots + N_{-m} \times J^{-m} \\
 &= \sum_{i=n-1}^{-m} N_i \times J^i
 \end{aligned}$$

式中， N 为一个 N 进制数，其每位数 N_i 的取值范围为 0, 1, ..., $N-1$ ； J 为 N 进制数的基数（如二进制数的基数为 2，八进制的基数为 8，十六进制的基数为 16）； J^i 为 N 进制数的权（ $i=n-1, \dots, 2, 1, 0, -1, -2, \dots, -m$ ）； n 为整数位数； m 为小数位数。

特点：

1) 一个任意进制数据的实际值不仅取决于数据本身的数，还取决于该数在数据中的位置。

2) 任意进制数到十进制数的转换采用的是按权展开法。

1.2.2 数制转换

不同数制之间的转换在汇编语言中是常用到的，在编写程序时可以用十进制、十六进

制、八进制和二进制，应熟练掌握它们之间的转换。

1. 二进制数转换成十进制数

二进制数转换成十进制数采用的方法是按权展开法，即按权展开后再相加。

$$\begin{aligned} \text{【例 1-5】 } (111.101)_2 &= (1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3})_{10} \\ &= (4 + 2 + 1 + 0.5 + 0.125)_{10} = (7.625)_{10} \end{aligned}$$

2. 十进制数转换成二进制数

十进制数转换成二进制数包括整数部分的转换和小数部分的转换。其转换方法为：整数部分采用除 2 取余法或降幂法，小数部分采用乘 2 取整或降幂法。

(1) 整数

整数部分采用除 2 取余或降幂法。

1) 除 2 取余法。

【例 1-6】用除 2 取余法计算 $215D = (?)B$ 。

2	215	1	B ₀	<div style="display: flex; align-items: center;"> <div style="border-left: 1px solid black; height: 100px; margin-right: 5px;"></div> <div style="writing-mode: vertical-rl; text-orientation: upright;">低 高</div> </div>
2	107	1	B ₁	
2	53	1	B ₂	
2	26	0	B ₃	
2	13	1	B ₄	
2	6	0	B ₅	
2	3	1	B ₆	
2	1	1	B ₇	
	0				

所以， $215D = (B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0) B = 11010111B$

除 2 取余法也可以采用横向列式的方法，将 215 写在最右侧，逐步向左除 2 取余。

1	←	3	←	6	←	13	←	26	←	53	←	107	←	215
1		1	0	1	0	1	1	1	1					← 余数
B ₇		B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀						← 位置

所以， $215D = (B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0) B = 11010111B$

2) 降幂法。

【例 1-7】用降幂法计算 $116D = (?)B$ 。

小于 116 的二进制权为：

2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1

计算:

$116 - 2^6 = 116 - 64 = 52$	够减	$B_6 = 1$	<div style="display: flex; flex-direction: column; align-items: center;"> 高 <div style="border-left: 1px solid black; border-right: 1px solid black; height: 100px; margin: 0 5px;"></div> 低 </div>
$52 - 2^5 = 52 - 32 = 20$	够减	$B_5 = 1$	
$20 - 2^4 = 20 - 16 = 4$	够减	$B_4 = 1$	
$4 - 2^3 = 4 - 8$	不够减	$B_3 = 0$	
$4 - 2^2 = 4 - 4 = 0$	够减	$B_2 = 1$	
$0 - 2^1 = 0 - 2$	不够减	$B_1 = 0$	
$0 - 2^0 = 0 - 1$	不够减	$B_0 = 0$	

所以, $116D = (B_6 B_5 B_4 B_3 B_2 B_1 B_0) B = 1110100B$

(2) 小数

小数部分采用乘2取整或降幂法。

1) 乘2取整法。

【例 1-8】用乘2取整法计算 $0.6875D = (?) B$ 。

0.6875	取整数	位置	<div style="display: flex; flex-direction: column; align-items: center;"> 高 <div style="border-left: 1px solid black; border-right: 1px solid black; height: 100px; margin: 0 5px;"></div> 低 </div>
$\times \quad \cdot \cdot \cdot \cdot 2$ 1	B_{-1}	
<hr style="width: 100%; border: 0.5px solid black;"/>			
0.3750			
$\times \quad \cdot \cdot 2$ 0	B_{-2}	
<hr style="width: 100%; border: 0.5px solid black;"/>			
0.750			
$\times \quad \cdot 2$ 1	B_{-3}	
<hr style="width: 100%; border: 0.5px solid black;"/>			
0.50			
$\times \quad \cdot 2$ 1	B_{-4}	
<hr style="width: 100%; border: 0.5px solid black;"/>			
0.0			

所以, $0.6875D = (0.B_{-1}B_{-2}B_{-3}B_{-4}) B = 0.1011B$

2) 降幂法。

【例 1-9】用降幂法计算 $0.84375D = (?) B$ 。

计算小于 $0.84375D$ 的二进制权为:

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
↓	↓	↓	↓	↓
0.5	0.25	0.125	0.0625	0.03125

计算:

$0.84375 - 2^{-1} = 0.84375 - 0.5 = 0.34375$	够减	$B_{-1} = 1$
$0.34375 - 2^{-2} = 0.34375 - 0.25 = 0.09375$	够减	$B_{-2} = 1$
$0.09375 - 2^{-3} = 0.09375 - 0.125$	不够减	$B_{-3} = 0$
$0.09375 - 2^{-4} = 0.09375 - 0.0625 = 0.03125$	够减	$B_{-4} = 1$
$0.03125 - 2^{-5} = 0.03125 - 0.03125 = 0$	够减	$B_{-5} = 1$

所以, $0.84375D = (0.B_{-1}B_{-2}B_{-3}B_{-4}B_{-5}) B = 0.11011B$