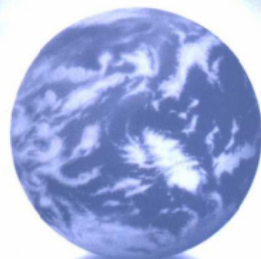


高等院校计算机科学与技术  
“十五”规划教材

# 80x86汇编语言 程序设计



●  
马力妮  
主编



高等院校计算机科学与技术“十五”规划教材

# 80x86 汇编语言程序设计

马力妮 主编  
施运梅 关静丽 侯凌燕 编著



机械工业出版社

本书以 Intel8086/8088 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:汇编语言程序设计的基础知识、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,每章后附有一定数量的以考题形式为主的习题。

实验方法中介绍了两种编译调试工具,为读者提供了方便:一种是  
MASM 5.0 汇编、链接程序,主要用于 8086/8088、80286 系统的编译和调  
试(MASM 和 DEBUG),另一种是 Microsoft MASM6.11 软件包,主要用于  
80386 以上的系统的编译和调试(PWB 集成编译器和 Codeview 调试器)。

本书是高等院校计算机专业及相关专业本科教学用书,也可以作为计  
算机工作者、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. —北京:机械  
工业出版社,2004.7

(高等院校计算机科学与技术“十五”规划教材)

ISBN 7-111-14803-7

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(2004)第 062184 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:蔡 岩

责任印制:李 妍

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2004 年 8 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ ·23.75 印张·587 千字

0 001—5 000 册

定价:33.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68993821、88379646

封面无防伪标均为盗版

## 出版说明

适逢高等院校计算机教育改革的关键时期,为配合各高等院校的教材建设,机械工业出版社会同全国在该领域内享誉盛名,具备雄厚师资和技术力量的高等院校,包括清华大学、南京大学、成都电子科技大学、解放军理工大学通信工程学院、东南大学、北京科技大学等重点名校,组织了多位长期从事教学工作的骨干教师,集思广益,对当前各高等院校的教学现状开展了广泛而深入的研讨,继而紧密结合当前计算机发展需要并针对当前教学改革所提出的问题,精心编写了这套面向普通高等院校计算机专业的系列教材,并陆续出版。

本套教材的选题内容覆盖了普通高等院校计算机专业学生的必修课程,另外还恰如其分地添加了一些选修课程,总体上分为基础、软件、硬件、网络和多媒体五大类。在编写过程中,对教学改革力度比较大、内容新颖以及各院校急需的并且适应社会经济发展的新教材,优先选择出版。

本套教材在写作手法上注重系统性、普及性和实用性,力求达到专业基础课教材概念清晰、深度合理,并且注意与专业课教学的衔接;专业课教材覆盖面广、深浅适中,在体现相关领域最新发展的同时注重理论联系实际。本套教材体现了教育的最新思想,可作为高等院校计算机科学与技术专业的教学用书,同时也是培训班和自学使用的最佳教材。

机械工业出版社

# 前 言

根据国家教委的要求,当前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;用科学性较强并简单易懂、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。

本教材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,使读者在应用中熟悉 80x86 及 Pentium 的指令,在实验指导中则用汇编语言程序的两种结构方法和两种上机环境来介绍上机过程。

本教材力求重点突出、语言简洁,为了避免汇编指令的集中堆砌,书中采取部分指令在相关程序设计中介绍的方法。

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,使读者容易掌握汇编程序设计方法,降低了汇编程序设计的复杂度。

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。

在编程举例中采用实例分析、流程图、根据流程图编写程序、程序说明 4 个步骤使读者深入掌握汇编源程序编写的特点,突出了重点和难点以及汇编程序的编程技巧,使教材便于自学,解决了汇编程序设计课程教学内容多、学时少、难学等问题。

在汇编语言程序设计实验方法中,分别介绍了 2 种上机环境:8086/8088、80286 编译、调试工具(MASM5.0 和 DEBUG)的使用;80386 以上的上机编译、调试工具(MASM6.X 以上版本的 PWB 编译器和 CodeView 调试器)的使用。重点在 PWB 编译器和 CodeView 调试器的使用上,使得 80x86 的所有指令得到很好的应用。

本教材第 1~5 章由马力妮老师编写,同时也参与编写其他章节的部分内容,第 6、7 章由侯凌燕老师编写,第 8、9 章由关静丽老师编写,第 10、11 章由施运梅老师编写,在此向参加本教材编写的老师表示衷心的感谢。

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。

本教材中的所有例题的源程序和电子教案可以在机工网 [www.cmpbook.com](http://www.cmpbook.com) 上下载。

作 者

# 目 录

出版说明

前言

<b>第 1 章 基础知识</b> .....	1
1.1 什么是汇编语言 .....	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 进位计数制及转换方法 .....	3
1.2.1 常用的进位计数制 .....	3
1.2.2 数制之间的转换 .....	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 二进制的算术运算和逻辑运算 .....	17
1.4.1 二进制算术运算 .....	17
1.4.2 逻辑操作 .....	18
1.5 习题 .....	19
<b>第 2 章 80x86 计算机组织结构</b> .....	21
2.1 80x86 计算机的基本结构 .....	21
2.1.1 中央处理器 .....	21
2.1.2 总线结构 .....	21
2.1.3 存储器 .....	22
2.1.4 外围设备 .....	23
2.2 80x86 CPU 的寄存器 .....	23
2.2.1 通用寄存器 .....	23
2.2.2 控制寄存器 .....	25
2.2.3 段寄存器 .....	28
2.3 内存组织结构 .....	29
2.3.1 内存的地址与字节、字的存放 .....	29
2.3.2 堆栈 .....	30
2.4 80x86 CPU 的工作模式 .....	31
2.4.1 实模式 .....	31
2.4.2 保护模式 .....	36
2.5 存储器管理机制 .....	37

2.5.1 分段管理机制 .....	37
2.5.2 分页管理机制 .....	37
2.6 习题 .....	38
<b>第3章 80x86 指令系统和寻址方式 .....</b>	<b>40</b>
3.1 指令格式 .....	40
3.1.1 指令的书写格式 .....	40
3.1.2 提供操作数的方式 .....	41
3.2 寻址方式 .....	42
3.2.1 与数据有关的寻址方式 .....	43
3.2.2 32位CPU的寻址方式 .....	51
3.2.3 与转移地址有关的寻址方式 .....	55
3.3 指令系统 .....	58
3.3.1 数据传送指令 .....	58
3.3.2 算术运算指令 .....	68
3.4 习题 .....	88
<b>第4章 汇编语言程序结构 .....</b>	<b>92</b>
4.1 汇编语言程序的基本结构 .....	92
4.1.1 汇编程序的基本框架 .....	92
4.1.2 汇编语言的指令格式 .....	94
4.2 汇编语言的数据 .....	95
4.2.1 常量 .....	95
4.2.2 变量 .....	95
4.2.3 标号 .....	98
4.2.4 变量定义时用到的操作符和表达式 .....	99
4.3 运算符与表达式 .....	102
4.3.1 算术运算符 .....	102
4.3.2 逻辑运算符 .....	102
4.3.3 关系运算符 .....	103
4.3.4 数值回送操作符 .....	103
4.3.5 属性操作符(PTR、:、THIS、SHORT、HIGH、LOW) .....	105
4.3.6 移位运算符 .....	108
4.3.7 运算符的优先级 .....	108
4.4 伪指令 .....	109
4.4.1 完整段定义伪指令(SEGMENT、ENDS、ASSUME) .....	109
4.4.2 处理器选择伪指令 .....	114
4.4.3 简化段定义伪指令 .....	114
4.4.4 定位伪指令 ORG .....	117
4.4.5 赋值伪指令(EQU、=) .....	117
4.4.6 LABEL 伪指令 .....	118
4.4.7 对准伪指令(EVEN) .....	119
4.4.8 标题伪指令 .....	120
4.5 习题 .....	120

<b>第 5 章 顺序、分支和循环程序设计</b> .....	123
5.1 顺序程序设计 .....	123
5.2 分支程序设计 .....	128
5.2.1 转移指令 .....	128
5.2.2 条件转移指令 .....	129
5.2.3 根据条件设置字节指令 .....	135
5.2.4 分支程序设计的应用举例 .....	137
5.3 循环程序设计 .....	144
5.3.1 循环程序结构 .....	144
5.3.2 循环控制指令 .....	145
5.3.3 单重循环的应用举例 .....	163
5.3.4 多重循环的应用举例 .....	173
5.4 习题 .....	177
<b>第 6 章 子程序设计</b> .....	181
6.1 子程序的结构 .....	181
6.1.1 子程序的定义 .....	181
6.1.2 子程序的调用与返回指令 .....	182
6.1.3 子程序的编程原则 .....	187
6.1.4 参数的传递 .....	190
6.2 子程序的嵌套与递归 .....	202
6.2.1 子程序的嵌套调用 .....	202
6.2.2 子程序的递归 .....	205
6.3 子程序应用举例 .....	211
6.4 习题 .....	216
<b>第 7 章 高级汇编技术</b> .....	218
7.1 宏汇编 .....	218
7.1.1 宏定义 .....	218
7.1.2 宏调用和宏展开 .....	219
7.1.3 宏定义和调用时的参数使用 .....	220
7.1.4 宏定义的伪指令 .....	223
7.1.5 宏库的建立和使用 .....	224
7.2 重复汇编 .....	227
7.2.1 重复伪指令 .....	227
7.3 条件汇编 .....	229
7.3.1 条件汇编伪指令 .....	229
7.3.2 使用条件汇编伪指令的程序结构 .....	230
7.4 习题 .....	232
<b>第 8 章 输入/输出与中断</b> .....	235
8.1 输入/输出设备的数据传送方式 .....	235
8.1.1 I/O 端口 .....	235
8.1.2 I/O 指令 .....	236



8.1.3 I/O 设备的数据传送方式 .....	238
8.2 80x86 的中断系统 .....	241
8.2.1 中断的基本概念 .....	241
8.2.2 中断指令 .....	242
8.2.3 中断分类 .....	243
8.2.4 中断优先级和中断嵌套 .....	245
8.2.5 中断向量表 .....	245
8.2.6 中断过程 .....	247
8.2.7 如何编写中断处理程序 .....	248
8.3 BIOS 与 DOS 中断系统 .....	253
8.3.1 BIOS 与 DOS 中断调用概述 .....	253
8.3.2 键盘 I/O 程序设计 .....	255
8.3.3 显示器 I/O 程序设计 .....	262
8.3.4 BIOS 及 DOS 的时间功能调用 .....	273
8.3.5 图形系统程序设计 .....	277
8.3.6 发声系统程序设计 .....	287
8.4 习题 .....	291
<b>第 9 章 磁盘文件存取方法</b> .....	<b>294</b>
9.1 以文件句柄方式存取文件 .....	294
9.1.1 DOS 文件功能调用 .....	294
9.1.2 路径和 ASCII 串 .....	295
9.1.3 文件句柄、错误代码和文件属性 .....	296
9.1.4 文件的打开方式 .....	298
9.1.5 读写指针的移动 .....	298
9.2 文件操作举例 .....	299
9.3 习题 .....	309
<b>第 10 章 C/C++ 与汇编语言的接口设计</b> .....	<b>312</b>
10.1 嵌入式汇编 .....	312
10.1.1 嵌入式汇编的格式 .....	312
10.1.2 几点说明 .....	314
10.1.3 编译链接的方法 .....	316
10.2 C/C++ 程序调用汇编源程序 .....	316
10.2.1 一个简单的小例子 .....	316
10.2.2 编译链接的方法 .....	317
10.2.3 C/C++ 程序对汇编源程序的调用规则 .....	319
10.2.4 MASM6.11 环境下参数传递的变化 .....	326
10.3 习题 .....	328
<b>第 11 章 汇编语言程序设计实验方法</b> .....	<b>329</b>
11.1 汇编语言上机步骤 .....	329
11.2 用 MASM 5.0 汇编、链接程序 .....	330
11.2.1 运行汇编程序必备的条件 .....	330

11.2.2	执行汇编程序 .....	330
11.2.3	执行链接程序 .....	334
11.3	DEBUG 调试器的使用 .....	335
11.3.1	DEBUG 的主要特点 .....	335
11.3.2	通过 DEBUG 编写、运行汇编程序 .....	335
11.3.3	DEBUG 的进入 .....	337
11.3.4	DEBUG 的主要命令 .....	337
11.3.5	举例 .....	339
11.4	Microsoft MASM6.11 软件包的使用 .....	341
11.4.1	安装 MASM6.11 .....	341
11.4.2	安装后的任务 .....	343
11.4.3	PWB 系统的进入和退出 .....	343
11.4.4	PWB 主菜单 .....	344
11.4.5	设置 PWB 的开发环境 .....	344
11.4.6	PWB 应用实例 .....	346
11.5	CodeView 调试器的使用 .....	348
11.5.1	CodeView 调试器的进入和退出 .....	348
11.5.2	CodeView 调试器主窗口及功能键 .....	348
11.5.3	CodeView 功能键 .....	350
11.5.4	CodeView 环境设置 .....	350
11.5.5	用 CodeView 调试汇编程序 .....	351
11.5.6	在 Command 窗口调试程序 .....	352
<b>附录</b>	.....	354
附录 A	80x86 指令系统一览表 .....	354
附录 B	ASCII 码表 .....	365
附录 C	DOS 系统功能调用 (INT 21H) .....	366
附录 D	BIOS 功能调用 .....	367

# 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、汇编语言和高级语言的关系、掌握数制之间的转换及计算机中数据的组织形式。

## 1.1 什么是汇编语言

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、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,是惟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,它是计算机惟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。例如上例: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
A20200H	MOV z,AL

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,是和高级语言一样需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 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的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。而面向对象的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,是惟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——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。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. 汇言程序效率高于高级语言

在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:也就是运行速度快、目标代码短。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,要首先了解微机的内部结构,才能充分地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,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 进位计数制及转换方法

### 1.2.1 常用的进位计数制

数制包括进位计数制和非进位计数制,在进位计数制中是按进位的方式计数的,如:十进制。在非进位计数制中,用不同的符号表示不同的数据,如:罗马数字 III→3, IV→4 等。

#### 1. 十进制计数法

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计数的,其中  $10^N$  为权,每一位基数的幂为权,十进制的基数就是 10。

一般式:

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

式中,  $n$  为整数位数,  $m$  为小数位数。

$$D=0,1,2,3,4,5,6,7,8,9。$$

## 2. 二进制记数法

特点:

1) 它的数值部分用 2 个不同的数值符号来表示 0 或 1。

2) 逢二进一。

【例 1-2】

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

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,这个转换方法为按权展开法。

一般式:

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

式中,  $n$  为整数位数,  $m$  为小数位数。

$$B=0,1。$$

## 3. 八进制

特点:

1) 它的数值部分用 8 个不同的数值符号来表示 0,1,2,3,……7。

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。

一般式:

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

$$= \sum_{i=n-1}^{-m} Q_i \times 8^i$$

式中,  $n$  为整数位数,  $m$  为小数位数。

$$Q = 0, 1, 2, 3, 4, 5, 6。$$

#### 4. 十六进制

特点:

- 1) 它的数值部分用 16 个不同的数值符号来表示 0, 1, 2, 3, …… 9, A, B, C, D, E, F。
- 2) 逢十六进一。

【例 1-4】将十六进制 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 327.AH 转换成十进制的 807.625D。

一般式:

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

式中,  $n$  为整数位数,  $m$  为小数位数。

$$H = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。$$

#### 5. 进位记数制小结

一般表达式:

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

式中,  $N$  为数码, 取决于  $J$  进制的取值范围, 如二进制数的取值范围为 0 或 1。

$n$  为整数位数。

$m$  为小数位数。

$J$  为基数。

$J^i$  为权。(  $i = 0, 1, 2, \dots, n$  )

特点:

- 1) 数码是所表示的实际值, 这个实际值不仅取决于数码本身, 还取决于数码在数中的位置。
- 2) 任意进制到十进制的转换采用的是按权展开法。

### 1.2.2 数制之间的转换

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, 它们之间的转换应熟练掌握。

### 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。

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

### 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

【例 1-6】用除 2 取余法计算  $215D = (?)_B$ 。

			余数	位置	
2	215	.....	1	B <sub>0</sub>	↑ 低       高
2	107	.....	1	B <sub>1</sub>	
2	53	.....	1	B <sub>2</sub>	
2	26	.....	0	B <sub>3</sub>	
2	13	.....	1	B <sub>4</sub>	
2	6	.....	0	B <sub>5</sub>	
2	3	.....	1	B <sub>6</sub>	
	1	.....	1	B <sub>7</sub>	

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。

$$\begin{array}{cccccccc} 1 & \leftarrow & 3 & \leftarrow & 6 & \leftarrow & 13 & \leftarrow & 26 & \leftarrow & 53 & \leftarrow & 107 & \leftarrow & 215 \\ 1 & & 1 & & 0 & & 1 & & 0 & & 1 & & 1 & & 1 & \leftarrow \text{余数} \\ B_7 & & B_6 & & B_5 & & B_4 & & B_3 & & B_2 & & B_1 & & B_0 & \leftarrow \text{位置} \end{array}$$

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

【例 1-7】用降幂法计算  $116D = (?)_B$ 。

小于 116 的二进制权为:

$$\begin{array}{ccccccc} 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array}$$

计算:

$116 - 2^6 = 116 - 64 = 52$	够减	B <sub>6</sub> = 1	↑ 高       低
$52 - 2^5 = 52 - 32 = 20$	够减	B <sub>5</sub> = 1	
$20 - 2^4 = 20 - 16 = 4$	够减	B <sub>4</sub> = 1	
$4 - 2^3 = 4 - 8$	不够减	B <sub>3</sub> = 0	
$4 - 2^2 = 4 - 4 = 0$	够减	B <sub>2</sub> = 1	
$0 - 2^1 = 0 - 2$	不够减	B <sub>1</sub> = 0	
$0 - 2^0 = 0 - 1$	不够减	B <sub>0</sub> = 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。

【例 1-8】用乘 2 取整法计算  $0.6875D = (?)B$ 。

0.6875	取整数		位置	
× 2	.....	1	B <sub>-1</sub>	高 ↓ 低
0.3750				
× 2	.....	0	B <sub>-2</sub>	
0.750				
× 2	.....	1	B <sub>-3</sub>	
0.50				
× 2	.....	1	B <sub>-4</sub>	
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 <sub>-1</sub> = 1
$0.34375 - 2^{-2} = 0.34375 - 0.25 = 0.09375$	够减	B <sub>-2</sub> = 1
$0.09375 - 2^{-3} = 0.09375 - 0.125$	不够减	B <sub>-3</sub> = 0
$0.09375 - 2^{-4} = 0.09375 - 0.0625 = 0.03125$	够减	B <sub>-4</sub> = 1
$0.03125 - 2^{-5} = 0.03125 - 0.03125 = 0$	够减	B <sub>-5</sub> = 1

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

(3) 既有整数又有小数的情况

整数部分除 2 取余, 小数部分乘 2 取整, 也可用降幂法。

【例 1-10】 $30.75D = (?)B$ 。

1) 整数部分采用除 2 取余法:

1 ← 3 ← 7 ← 15 ← 30	
1 1 1 1 0	余数
B <sub>4</sub> B <sub>3</sub> B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	位置

$30D = (B_4 B_3 B_2 B_1 B_0)B = 11110B$

2) 小数部分用乘 2 取整法计算  $0.75D = (?)B$ 。