

■ 高等学校计算机教材 ■

# 汇编语言

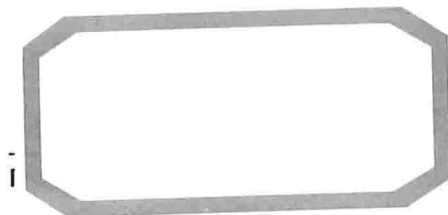
## 实用教程

■ 郑阿奇 主编 ■ 吕升旭 编著 ■



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



# 汇编语言实用教程

郑阿奇 主 编  
吕升旭 编 著

电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

本书以 80x86 指令系统为主体, 详细介绍汇编语言程序设计, 共 10 章。首先复习汇编语言数据运算基础和 80x86 微处理器体系结构, 通过一个实例介绍汇编语言程序上机操作过程; 然后介绍指令系统和寻址方式、汇编语言源程序的组成和 8086/8088 常用指令, 在此基础上, 介绍基本程序设计、子程序设计、输入/输出与中断程序设计和简单应用程序设计。本书的主要特点是循序渐进, 结构清晰, 重点突出, 教和学比较方便。

本书可作为高等学校有关专业汇编语言程序设计课程教材或参考书, 也可作为自学和微机应用技术人员参考用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

### 图书在版编目 ( CIP ) 数据

汇编语言实用教程 / 郑阿奇主编, 吕升旭编著. —北京: 电子工业出版社, 2011.5

高等学校计算机教材

ISBN 978-7-121-13397-8

I. ①汇… II. ①郑… ②吕… III. ①汇编语言—程序设计—高等学校—教材 IV. ①TP313

中国版本图书馆 CIP 数据核字 (2011) 第 078565 号

责任编辑: 郝黎明 特约编辑: 田领红

印 刷: 北京市顺义兴华印刷厂

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 20.75 字数: 531.2 千字

印 次: 2011 年 5 月第 1 次印刷

印 数: 4 000 册 定价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 前 言

本书以 80x86 指令系统为主体，详细介绍汇编语言程序设计。主要特点是循序渐进，结构清晰，重点突出，教和学比较方便。各章内容安排基本思路如下。

第 1 章汇编语言运算基础：系统复习汇编语言程序设计使用的数据及其算术和逻辑运算，这些内容在此前的课程已经介绍过，但学生对这部分内容的掌握个体上是有差别的，学习本课程之前需要复习这些内容。读者可根据情况选择学习。

第 2 章微型计算机的体系结构：主要介绍 80x86 微处理器及其操作方式，为 80x86 汇编语言程序设计提供操作基础。

第 3 章汇编语言程序及其操作过程：在介绍机器语言程序的基础上，介绍汇编语言程序。通过一个典型实例介绍机器语言程序上机操作方法，重点是机器语言程序的调试。在此基础上，介绍程序的编辑、编译、连接和运行。

第 4 章汇编语言指令系统和寻址方式：在简单介绍指令系统分类的基础上，介绍指令操作数和为了得到操作数而采用的寻址方式。

第 5 章汇编语言源程序组成：介绍汇编语言源程序书写格式、数据表达式、常用伪指令和正常结束方式。

第 6 章 8086/8088 常用指令：主要介绍数据传送类指令、算术运算类指令、位操作类指令和处理器控制类指令，控制转移类指令在第 7 章节中介绍，字符串处理类指令在第 10 章中介绍。

第 7 章基本程序设计：在介绍控制转移类指令的同时，系统介绍用汇编语言编写程序的步骤和方法，其中包括解决基本问题的算法。

第 8 章子程序设计：介绍子程序的定义、子程序的调用和返回、参数传递，另外，还有子程序的嵌套与递归调用。

第 9 章输入/输出与中断程序设计：在介绍输入和输出的基本概念和四种数据传送方式的基础上，介绍中断系统程序设计。然后介绍 BIOS 中断和 DOS 中断调用。

第 10 章简单应用程序设计：介绍字符串处理类指令及其程序设计，介绍十进制数算术运算调整指令及其程序设计，最后提供应用程序设计举例。

通过教程学习，结合实例上机练习，一般能够基本掌握汇编语言程序设计。

本书由南京师范大学郑阿奇主编，吕升旭编著，参加本书编写的还有梁敬东、顾韵华、王洪元、刘启芬、丁有和、曹弋、徐文胜、殷红先、张为民、姜乃松、彭作民、王一莉、徐斌、王志瑞、周怡明、刘博宇、周怡君、郑进、刘毅等。

本书配有教学课件和所有实例源代码文件，有需要者可以在出版社网站免费下载。网站地址为：<http://www.hxedu.com.cn>。

由于我们的水平有限，错误在所难免，敬请广大师生、广大读者批评指正。

意见建议邮箱：[easybooks@163.com](mailto:easybooks@163.com)。

编 者  
2010 年 11 月

# 目 录

第 1 章 汇编语言运算基础 .....	1
1.1 二进制数 .....	1
1.1.1 二进制 .....	1
1.1.2 二进制数运算 .....	3
1.1.3 十六进制数值运算规则 .....	5
1.1.4 二、八、十六、十进制数相互转换 .....	6
1.2 计算机中二进制数的表示及运算 .....	8
1.2.1 不带符号的整数表示 .....	9
1.2.2 带符号的整数表示 .....	9
1.2.3 浮点数表示 .....	12
1.3 计算机中字符表示 .....	13
1.3.1 ASCII 码 .....	14
1.3.2 汉字编码 .....	15
1.4 计算机中逻辑值表示及运算 .....	17
1.4.1 “与”运算 .....	17
1.4.2 “或”运算 .....	17
1.4.3 “非”运算 .....	18
1.4.4 “异或”运算 .....	18
1.4.5 逻辑代数基本公式 .....	19
习题 .....	20
第 2 章 微型计算机的体系结构 .....	21
2.1 微型计算机基本组成 .....	21
2.2 微处理器 .....	22
2.2.1 Intel 公司微处理器简介 .....	22
2.2.2 Intel 8086/8088 微处理器功能结构 .....	23
2.3 Intel 8086/8088 CPU 寄存器组 .....	26
2.3.1 通用寄存器 .....	27
2.3.2 段寄存器 .....	28
2.3.3 控制寄存器 .....	29
2.4 存储器 .....	32
2.4.1 存储器的段结构 .....	32
2.4.2 物理地址与逻辑地址 .....	33
2.4.3 内存地址空间 .....	35
2.5 堆栈 .....	36

2.5.1	堆栈的概念	36
2.5.2	堆栈的组织	37
2.5.3	堆栈的操作	38
2.6	输入/输出设备	40
	习题	40
<b>第3章</b>	<b>汇编语言程序及其操作过程</b>	<b>42</b>
3.1	机器语言程序	42
3.1.1	机器语言程序	42
3.1.2	机器语言程序的输入和运行	43
3.1.3	Debug 调试工具	44
3.2	汇编语言程序	50
3.2.1	汇编语言	50
3.2.2	汇编语言程序	50
3.3.3	Emu8086 的使用	52
3.3	汇编语言程序的编辑、编译、连接和运行	55
3.3.1	汇编语言程序的编辑	55
3.3.2	MASM 汇编开发环境	57
3.3.3	汇编语言程序的编译、连接和运行	61
	习题	65
<b>第4章</b>	<b>汇编语言指令系统和寻址方式</b>	<b>67</b>
4.1	汇编语言指令的格式	67
4.2	指令系统分类	68
4.2.1	数据传送类指令	68
4.2.2	算术运算类指令	69
4.2.3	位操作类指令	70
4.2.4	控制转移类指令	70
4.2.5	字符串处理类指令	71
4.2.6	处理器控制类指令	71
4.3	操作数的类型	72
4.3.1	立即数操作数	72
4.3.2	寄存器操作数	73
4.3.3	存储器操作数	74
4.4	立即寻址方式	74
4.5	寄存器寻址方式	75
4.6	存储器寻址方式	76
4.6.1	直接寻址方式	77
4.6.2	寄存器间接寻址方式	79
4.6.3	寄存器相对寻址方式	81

4.6.4	基址变址寻址方式	83
4.6.5	相对基址变址寻址方式	85
4.6.6	字符串操作寻址方式	87
	习题	87
<b>第 5 章</b>	<b>汇编语言源程序组成</b>	<b>89</b>
5.1	汇编语言源程序的格式	89
5.1.1	段的定义	89
5.1.2	简化段定义格式	93
5.1.3	完整段定义格式	94
5.2	汇编语言数据	96
5.2.1	常量	96
5.2.2	变量	98
5.2.3	标号	103
5.2.4	变量和标号的属性	103
5.3	表达式与运算符	107
5.3.1	算术运算符	107
5.3.2	移位运算符	109
5.3.3	逻辑运算符	109
5.3.4	关系运算符	110
5.3.5	其他运算符	110
5.3.6	运算符的优先级	111
5.4	常用伪指令	112
5.4.1	属性定义伪指令 LABEL	112
5.4.2	段寻址伪指令 ASSUME	113
5.4.3	过程定义伪指令 PROC/ENDP	114
5.4.4	调整偏移量伪指令	115
5.4.5	标题伪指令 TITLE	117
5.4.6	程序结束伪指令 END	118
5.4.7	模块连接伪指令 PUBLIC 和 EXTRN	119
5.5	汇编语言源程序正常结束方式	121
5.5.1	使用 4CH 系统功能调用正常结束	121
5.5.2	使用程序段前缀正常结束	121
	习题	123
<b>第 6 章</b>	<b>8086/8088 常用指令</b>	<b>125</b>
6.1	数据传送类指令	125
6.1.1	通用数据传送指令	126
6.1.2	堆栈操作指令	128
6.1.3	数据交换指令	130

6.1.4	查表转换指令	131
6.1.5	地址传送指令	131
6.1.6	标志位传送指令	133
6.1.7	端口输入/输出指令	134
6.2	算术运算类指令	136
6.2.1	加法指令	136
6.2.2	减法指令	139
6.2.3	乘法指令	142
6.2.4	除法指令	144
6.2.5	符号扩展指令	145
6.2.6	BCD 码十进制调整指令	146
6.3	位操作类指令	147
6.3.1	逻辑运算指令	147
6.3.2	移位指令	150
6.3.3	循环移位指令	153
6.4	处理器控制类指令	156
6.4.1	标志位操作指令	156
6.4.2	CPU 控制指令	157
	习题	158
<b>第 7 章</b>	<b>基本程序设计</b>	<b>160</b>
7.1	程序设计方法	160
7.1.1	程序设计步骤	160
7.1.2	程序流程图	161
7.2	顺序结构程序设计	162
7.2.1	顺序程序设计	162
7.2.2	简单查表法代码转换	165
7.3	分支程序设计	167
7.3.1	无条件转移指令	168
7.3.2	条件转移指令	172
7.3.3	单分支程序设计	176
7.3.4	双分支程序设计	177
7.3.5	多分支程序设计	179
7.4	循环结构程序设计	186
7.4.1	循环控制指令	186
7.4.2	循环程序的结构	188
7.4.3	单重循环程序设计	189
7.4.4	多重循环程序设计	195
	习题	197



第 8 章 子程序设计 .....	199
8.1 子程序的定义 .....	199
8.1.1 子程序的格式 .....	199
8.1.2 子程序的说明信息 .....	200
8.2 子程序的调用和返回 .....	201
8.2.1 调用指令 .....	202
8.2.2 返回指令 .....	205
8.2.3 现场保护和恢复 .....	207
8.3 子程序与主程序间的参数传递 .....	208
8.3.1 利用寄存器传递参数 .....	208
8.3.2 利用堆栈传递参数 .....	210
8.3.3 利用约定存储单元传递参数 .....	213
8.4 子程序的嵌套与递归调用 .....	215
8.4.1 子程序的嵌套调用 .....	215
8.4.2 子程序的递归调用 .....	216
习题 .....	218
第 9 章 输入/输出与中断程序设计 .....	220
9.1 输入和输出的基本概念 .....	220
9.1.1 I/O 端口地址 .....	220
9.1.2 I/O 端口编址 .....	221
9.1.3 I/O 指令 .....	223
9.2 数据传送方式 .....	224
9.2.1 程序控制方式 .....	225
9.2.2 中断控制方式 .....	229
9.2.3 直接存储器存取方式 .....	230
9.3 中断 .....	230
9.3.1 中断的分类 .....	231
9.3.2 中断向量表 .....	232
9.3.3 中断优先级和中断嵌套 .....	235
9.3.4 中断指令 .....	237
9.3.5 中断过程 .....	238
9.3.6 中断处理程序的设计 .....	241
9.4 BIOS 与 DOS 中断调用 .....	243
9.4.1 BIOS 与 DOS 中断概述 .....	243
9.4.2 键盘 BIOS 功能调用 .....	246
9.4.3 键盘 DOS 功能调用 .....	249
9.4.4 显示器 BIOS 功能调用 .....	252
9.4.5 显示器 DOS 功能调用 .....	260

9.4.6	打印机 BIOS 功能调用 .....	261
9.4.7	时钟 BIOS 和 DOS 功能调用 .....	261
	习题 .....	262
<b>第 10 章</b>	<b>简单应用程序设计</b> .....	<b>263</b>
10.1	汇编语言的字符串 .....	263
10.1.1	字符串及其操作命令 .....	263
10.1.2	重复前缀 .....	270
10.2	十进制数算术运算调整指令及其应用 .....	274
10.2.1	压缩 BCD 码的算术运算调整指令 .....	276
10.2.2	非压缩 BCD 码的算术运算调整指令 .....	279
10.3	代码转换 .....	283
10.3.1	其他进制数到二进制数的转换 .....	283
10.3.2	二进制数到其他进制数的转换 .....	291
	习题 .....	296
附录 A	ASCII 码字符表 .....	298
附录 B	DOS 系统功能调用 (INT 21H) .....	300
附录 C	BIOS 中断调用 (INT n) .....	307
附录 D	指令表 .....	312
附录 E	伪指令表 .....	318

## 汇编语言运算基础

人们最常用的数制是十进制，而计算机中采用二进制。所以，用计算机解决问题时，必须把十进制转换为二进制才能进行运算，运算的结果又要转换为十进制人们才容易阅读。本章介绍二进制的运算及其相关内容。

### 1.1 二进制数

#### 1.1.1 二进制

为了理解二进制，先从人们熟悉的十进制开始。

##### 1. 十进制

十进制 (Decimal) 是最常用的数制，其基本特征如下：

- (1) 基数为 10，采用 0、1、2、3、4、5、6、7、8、9 共十个数码；
- (2) 逢十进一；
- (3) 处于不同位置上的数码位权不同。个位上的“1”位权是 1 (即  $10^0$ )，十位上的“1”位权是 10 (即  $10^1$ )，百位上的“1”位权是 100 (即  $10^2$ )，……十进制数整数部分第  $n$  位的数码位权是  $10^{n-1}$ ，从小数点往后数，第  $m$  位数码位权是  $10^{-m}$ 。

例如，十进制数 108.625 用位权表示如下：

$$108.625=1\times 10^2+0\times 10^1+8\times 10^0+6\times 10^{-1}+2\times 10^{-2}+5\times 10^{-3}$$

十进制数 108.625，也可表示为  $(108.625)_{10}$  或者 108.625D。

##### 2. 二进制

计算机中使用二进制 (Binary)，其基本特征如下：

- (1) 基数为 2，采用 0、1 共两个数码；
- (2) 逢二进一；
- (3) 二进制数中处于不同位置上的数码代表不同的位权。二进制数整数部分第  $n$  位的数码位权是  $2^{n-1}$ ，从小数点往后数，第  $m$  位数码位权是  $2^{-m}$ 。

例如，二进制数 1101100.101 用位权表示如下：

$$1101100.101=1\times 2^6+1\times 2^5+0\times 2^4+1\times 2^3+1\times 2^2+0\times 2^1+0\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}$$

对于二进制小数，小数点向右移一位，数就扩大 2 倍，相当于乘 2；反之，小数点向左移一位，数就缩小 2 倍，相当于除 2。每一位二进制位权对应的十进制如表 1.1 所示。

表 1.1 二进制位权对应的十进制

二进制	十进制	二进制	十进制	二进制	十进制
1	1	10000	16	0.1	0.5
10	2	100000	32	0.01	0.25
100	4	1000000	64	0.001	0.125
1000	8	10000000	128	0.0001	0.0625

二进制数 1101100.101，可以表示为 $(1101100.101)_2$ ，也可表示为 1101100.101B 或 1101100.101b。

### 3. 八进制

由于 $2^3=8$ ，一位八进制 (Octonary) 可以直接对应 3 位二进制。有时用八进制表达相应的二进制。八进制基本特征如下：

- (1) 基数为 8，采用 0、1、3、4、5、6、7 共八个数码；
- (2) 逢八进一，借一当八；
- (3) 八进制数整数部分第  $n$  位的数码位权是  $8^{n-1}$ ，从小数点往后数，第  $m$  位数码位权是  $8^{-m}$ 。

例如，八进制数 154.5 用位权表示如下：

$$(154.5)_8 = 1 \times 8^2 + 5 \times 8^1 + 4 \times 8^0 + 5 \times 8^{-1}$$

八进制数 154.5，可以表示为 $(154.5)_8$ ，也可表示为 154.5Q 或 154.5q。有些书八进制后缀采用字母“O”表示，但字母“O”与数字“0”很像，容易混淆。

### 4. 十六进制

由于 $2^4=16$ ，一位十六进制 (Hex) 可以直接对应 4 位二进制，所以经常用十六进制表达相应的二进制。十六进制基本特征如下：

- (1) 基数为 16，采用 0、1、3、4、5、6、7、8、9、A、B、C、D、E、F 共十六个数码。其中，A~F(或者 a~f) 分别代表十六进制的 10、11、12、13、14、15；
- (2) 逢十六进一，借一当十六；
- (3) 十六进制数整数部分第  $n$  位的数码位权是  $16^{n-1}$ ，从小数点往后数，第  $m$  位数码位权是  $16^{-m}$ 。

例如，十六进制数 6C.A 用位权表示如下：

$$(6C.A)_{16} = 6 \times 16^1 + 12 \times 16^0 + 10 \times 16^{-1}$$

十六进制数 6C.A，可以表示为 $(6C.A)_{16}$ ，也可表示为 6C.AH 或 6C.Ah。当十六进制数的第 1 位为 A~F(或者 a~f) 时，在其前面加上数字 0，例如 0A6H。

表 1.2 列出了 0~15 四种数制之间的对应关系。

也就是说：

$$(14)_{10} = (1110)_2 = (16)_8 = (E)_{16}$$

或者

$$14D = 1110B = 16Q = EH$$

表 1.2 常见数制的对应关系

十进制(D)	二进制(B)	八进制(Q)	十六进制(H)
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

### 1.1.2 二进制数运算

二进制数运算主要包括加 (+)、减 (-)、乘 (×) 和除 (÷)。下面介绍二进制数运算规则，在此基础上说明二进制的优点。

#### 1. 二进制数的运算规则

二进制数比较简单，所以它的运算规则也很简单。

##### (1) 二进制数的加法

根据“逢二进一”规则，二进制数加法的法则如下：

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

【例 1.1】 求 $(1101)_2+(1001.01)_2$

$$\begin{array}{r} 1101 \\ + 1001.01 \\ \hline 10110.01 \end{array}$$

所以， $(1101)_2+(1001.01)_2=(10110.01)_2$

##### (2) 二进制数的减法

根据“借一当二”的规则，二进制数减法的法则如下：

$$0-0=0 \quad 0-1=1(\text{借 } 1) \quad 1-0=1 \quad 1-1=0$$

【例 1.2】 求 $(1101)_2-(1001.01)_2$

$$\begin{array}{r} 1101 \\ - 1101.01 \\ \hline 0011.11 \end{array}$$

所以,  $(1101)_2 - (1001.01)_2 = (11.11)_2$

### (3) 二进制数的乘法

二进制数乘法的法则如下:

$$0 \times 0 = 0 \quad 0 \times 1 = 0 \quad 1 \times 0 = 0 \quad 1 \times 1 = 1$$

二进制数的乘法由低位到高位, 用乘数的每一位去乘被乘数, 若乘数的某一位为 1, 则该次部分积为被乘数; 若乘数的某一位为 0, 则该次部分积为 0。某次部分积的最低位必须和本位乘数对齐, 所有部分积相加的结果则为相乘得到的乘积。

**【例 1.3】** 求  $(1001)_2 \times (1010)_2$

$$\begin{array}{r} 1001 \\ \times 1010 \\ \hline 0000 \\ 1001 \\ 0000 \\ 1001 \\ \hline 1011010 \end{array}$$

所以,  $(1001)_2 \times (1010)_2 = (1011010)_2$

### (4) 二进制数的除法

二进制数除法的法则如下:

$$0 \div 1 = 0 \quad 1 \div 1 = 1$$

二进制数除法可先从被除数的最高位开始, 将被除数(或中间余数)与除数相比较, 若被除数(或中间余数)大于除数, 则用被除数(或中间余数)减去除数, 商为 1, 并得相减之后的中间余数; 否则商为 0。再将被除数的下一位移下补充到中间余数的末位, 重复以上过程, 就可得到所要求的各位商数和最终的余数。

注意, 除数不能为 0。

**【例 1.4】** 求  $(1110101)_2 \div (1001)_2$

$$\begin{array}{r} 1101 \\ 1001 \overline{) 1110101} \\ \underline{1001} \phantom{000} \\ 1011 \phantom{00} \\ \underline{1001} \phantom{00} \\ 1001 \phantom{0} \\ \underline{1001} \\ 0 \end{array}$$

所以,  $(1110101)_2 \div (1001)_2 = (1101)_2$

## 2. 二进制的优点

通过对上面二进制的介绍, 可以看出二进制具有下列优点:

- (1) 十进制有十个状态, 在自然界中, 用某种器件表示十种状态比较难。二进制只有 0 和 1 两个状态, 易于实现;
- (2) 二进制运算规则简单, 运算功能比较容易实现。

### 1.1.3 十六进制数值运算规则

十六进制数的运算可以采用先把该十六进制数转换为十进制数，经过计算后再把结果转换为十六进制数的方法，但这样做比较烦琐。其实，只要按照“逢十六进一”（借一当十六）的规则，直接用十六进制数计算也是很方便的。

#### 1. 十六进制数的加法

十六进制数的加法就是让十六进制数的每一位对应相加，如果和大于或等于 16 时，则进位。

**【例 1.5】** 求 $(05C3)_{16}+(3D25)_{16}$

$$\begin{array}{r} 05C3 \\ + 3D25 \\ \hline 42E8 \end{array}$$

所以， $(05C3)_{16}+(3D25)_{16}=(42E8)_{16}$

#### 2. 十六进制数的减法

十六进制数的减法也与十进制数类似，够减时可直接相减，不够减时从向高位借一当十六。

**【例 1.6】** 求 $(3D25)_{16}-(05C3)_{16}$

$$\begin{array}{r} 3D25 \\ - 05C3 \\ \hline 3762 \end{array}$$

所以， $(3D25)_{16}-(05C3)_{16}=(3762)_{16}$

#### 3. 十六进制数的乘法

十六进制数的乘法用乘法规则计算，但结果必须用十六进制数表示。

十六进制数的乘法也是由低位到高位，用乘数的每一位去乘被乘数，某次部分积的最低位必须和本位乘数对齐，所有部分积相加的结果则为相乘得到的乘积。

**【例 1.7】** 求 $(3F)_{16}\times(1B)_{16}$

$$\begin{array}{r} 3F \\ \times 1B \\ \hline 2B5 \\ 3F \\ \hline 6A5 \end{array}$$

所以， $(3F)_{16}\times(1B)_{16}=(6A5)_{16}$

#### 4. 十六进制数的除法

十六进制数的除法可以仿照十进制数除法规则处理，仅仅需要注意它是十六进制数。在不习惯的情况下，每一步可以先把它变成十进制，计算后的商再变成十六进制商的一个位。

**【例 1.8】** 求 $(6A5)_{16}\div(1B)_{16}$

$$\begin{array}{r} \phantom{1B} \overline{) 6A5} \\ \underline{51} \phantom{0} \\ 195 \\ \underline{195} \\ 0 \end{array}$$

所以,  $(65A)_{16} \div (1B)_{16} = (3F)_{16}$

### 1.1.4 二、八、十六、十进制数相互转换

前面介绍了十进制、二进制、八进制和十六进制,就它们之间数的互相转换方法来说,主要就是十进制数与非十进制数(二、八、十六进制数)互相转换,十进制数与非十进制数之间转换方法类似。二、八和十六进制数之间的转换比较容易。

#### 1. 十进制数与二、八、十六进制数相互转换

十进制数与非十进制数(二、八、十六进制数)转换,包括非十进制数转换为十进制数和十进制数转换为非十进制数。

##### (1) 非十进制数转换为十进制数

根据前面的介绍,非十进制转换为十进制方法为:将非十进制数按权展开,即用位权表示法展开,而后进行相加。

**【例 1.9】** 将二进制数 101011 转换为十进制数。

$$(101011)_2 = 2^5 + 2^3 + 2^1 + 2^0 = 32 + 8 + 2 + 1 = 43$$

所以,  $(101011)_2 = 43$

##### (2) 十进制数转换为非十进制数

十进制数转换为非十进制数需要对整数部分和小数部分分别进行转换。

整数部分转换方法为“除基取余”法,直到商为 0 为止,将余数倒排即可。

小数部分转换方法为“乘基取整”法,直到乘积的小数部分为 0,或小数点后的位数达到了所需的精度为止,将积的整数部分顺排即可。

**【例 1.10】** 将十进制数 108.625 转换为二进制数。

整数部分:

	十进制数	余数	
2	108	0	↑ 低位         ↓ 高位
2	54	0	
2	27	1	
2	13	1	
2	6	0	
2	3	1	
2	1	1	
	0		



小数部分:

十进制数	积的整数部分	
0.625		
× 2		
1.250	1	↓ 高位
0.25		
× 2		
0.5	0	
0.5		
× 2		
1.0	1	↓ 低位

所以,  $(108.625)_{10} = (1101100.101)_2$

**【例 1.11】** 将十进制数 108.625 转换为八进制数。

整数部分:

十进制数	余数	
8   108	4	↑ 低位
8   13	5	
8   1	1	↑ 高位
0		

小数部分:

十进制数	积的整数部分
0.625	
× 8	
5.00	5

所以,  $(108.625)_{10} = (154.5)_8$

**【例 1.12】** 将十进制数 108.625 转换为十六进制数。

整数部分:

十进制数	余数	
16   108	12	↑ 低位
16   6	6	↑ 高位
0		

小数部分:

十进制数	积的整数部分
0.625	
× 16	
10.0	10

所以,  $(108.625)_{10} = (6C.A)_{16}$

通过例 1.10、例 1.11 和例 1.12 可知:  $(108.625)_{10} = (1101100.101)_2 = (154.5)_8 = (6C.A)_{16}$

或者,  $108.625 = 1101100B = 154.5Q = 6C.AH$