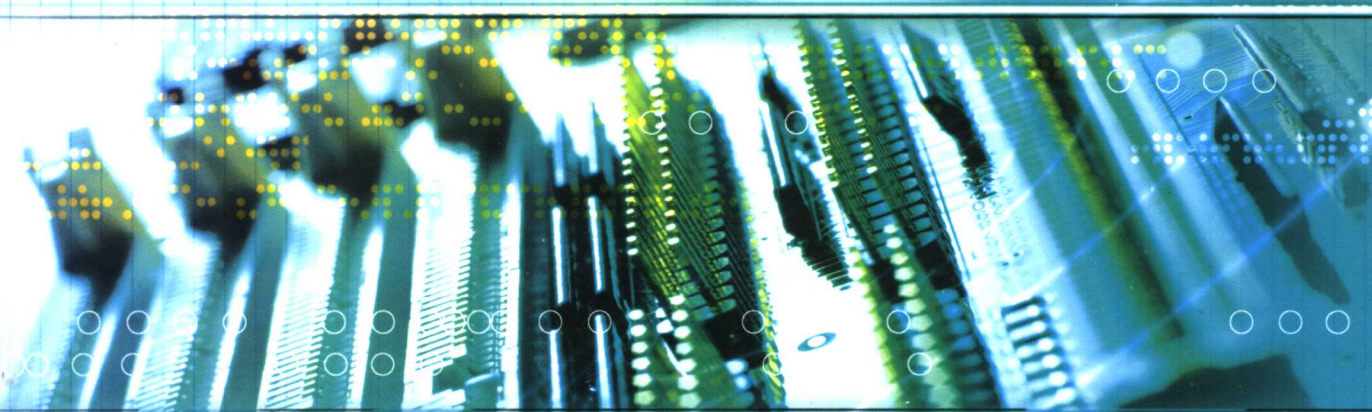


● 高等学校21世纪计算机教材

IBM PC 80X86

汇编语言程序设计 习题解析及实验指导

姜媛媛 白 洁 任卓谊 编著



冶金工业出版社

高等学校 21 世纪计算机教材

IBM PC 80X86 汇编语言程序 设计习题解析及实验指导

姜媛媛 白 洁 任卓谊 编著

北 京

冶金工业出版社

2005

内 容 简 介

本书是冶金工业出版社 2004 年出版的《IBM PC 80X86 汇编语言程序设计》(姜媛媛、任卓谊编著)的配套习题解析及实验指导教材,它是辅助教师教学、学生自学、学生上机实验及复习考试的重要参考书。全书共分三大部分。其中第一部分主要对主教材各章的重要知识要点进行回顾,同时介绍相关的典型例题及各类习题;第二部分为实验指导,供学生上机练习和学习;第三部分为模拟试卷,包括五套模拟试题及其参考答案,可供学生练习或考试之用;书末还有两个附录,包括 BIOS 中断调用和出错信息两部分的内容;最后还附有第一部分各章节习题的参考答案。

本书内容丰富、结构合理、语言通俗易懂,既可作为计算机及相关专业本、专科学生的《IBM PC 80X86 汇编语言程序设计》课程的学习和复习资料,也可作为软件工程技术人员的学习参考书和研究生入学考试的应试辅导资料。

图书在版编目(CIP)数据

IBM PC 80X86 汇编语言程序设计习题解析及实验指导 / 姜媛媛等编著. —北京:冶金工业出版社, 2005.2
ISBN 7-5024-3700-2

I. I... II. 姜... III. 汇编语言-程序设计-高等学校-教学参考资料 IV. TP313

中国版本图书馆 CIP 数据核字(2004)第 143175 号

出版人 曹胜利(北京沙滩嵩祝院北巷 39 号,邮编 100009)

责任编辑 戈兰

湛江蓝星南华印务公司印刷;冶金工业出版社发行;各地新华书店经销

2005 年 2 月第 1 版, 2005 年 2 月第 1 次印刷

787mm × 1092mm 1/16; 19.5 印张; 451 千字; 304 页

30.00 元

冶金工业出版社发行部 电话:(010) 64044283 传真:(010) 64027893

冶金书店 地址:北京东四西大街 46 号(100711) 电话:(010) 65289081

(本社图书如有印装质量问题,本社发行部负责退换)

前 言

一、关于本书

《IBM PC 80X86 汇编语言程序设计》是高等院校计算机硬、软件及应用专业学生都必须学习的核心课程之一，它是计算机组成原理、操作系统及其他核心课程的基础课，也是微机原理、单片机应用等课程的学习基础。同时，该课程对于训练学生掌握程序设计技术，熟悉上机操作和程序调试技术都有重要作用。

另一方面，《IBM PC 80X86 汇编语言程序设计》的内容十分丰富，知识量大，是一门教师普遍感到难教、学生普遍感到难学的课程。为了满足教师教学和学生学习的需要，作者编写了此书，以期能帮助学生快速熟练地掌握相关知识，同时为教师的教学提供指导。

本书与冶金工业出版社 2004 年出版的《IBM PC 80X86 汇编语言程序设计》(姜媛媛、任卓谊编著)配套使用。读者可以在学习主教材的同时结合使用本书，只要读者认真学习这两本书的相关知识，一定能达到较好的学习效果。

二、本书结构

本书共分为三大部分和两个附录，各部分内容如下：

第一部分：章节复习与习题，第 1~13 章。主要对主教材各章的知识要点进行回顾，同时介绍相关的典型例题及各类习题。读者可以在复习各章节内容的基础上做一些相关习题，以及及时巩固所学知识。如对某些习题有疑问，可参考本书末所附的习题参考答案。

第二部分：实验指导，第 14~22 章。介绍了汇编语言上机实验指导，主要用于指导读者上机实验，提高编程能力，同时培养发现问题、解决问题的能力。

第三部分：模拟试卷。给出了五套模拟试卷及其参考答案，可供学生练习或考试之用。

附录 A：BIOS 中断调用。

附录 B：出错信息。包括带编号的错误和不带编号的错误。

三、本书特点

本书语言通俗易懂、内容丰富、例题典型、结构安排合理，且注重理论知识与实际训练相结合，具有较强的适应性，对学生学好汇编语言的相关知识有较大的辅助作用。

四、本书适用对象

本书可作为计算机及相关专业本、专科学生的《IBM PC 80X86 汇编语言程序设计》课程的学习和复习资料，也可作为软件工程技术人员的学习参考书和研究生入学考试的应试辅导资料。

本书由姜媛媛、白洁、任卓谊共同编著，同时蕴含了魏法桂教授的大量心血，整个编写过程还得到了家人和同事的大力支持和鼓励，在此表示衷心感谢。

由于编写时间仓促，水平有限，书中如有疏漏和不足之处，敬请各位读者批评指正。

虽然经过严格的审核、精细的编辑，本书在质量上有了一定的保障，但我们的目标是力求尽善尽美，欢迎广大读者和专家对我们的工作提出宝贵建议，联系方式如下：

电子邮件：**service@cnbook.net**

网址：**www.cnbook.net**

此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者

2004 年 10 月

目 录

第一部分 章节复习与习题

第1章 概述.....2

1.1 学习要点.....2

1.1.1 计算机系统简介.....2

1.1.2 汇编语言简介.....3

1.1.3 计算机中数据表示方法.....4

1.2 典型例题.....8

结束语.....9

习题一.....9

一、选择题.....9

二、填空题.....11

三、简答题.....12

四、应用题.....12

第2章 IBM PC 微处理器的结构及

存储器组成.....14

2.1 学习要点.....14

2.1.1 IBM PC 微处理器基本结构.....14

2.1.2 IBM PC 的存储器组织.....16

2.1.3 堆栈.....18

2.2 典型例题.....18

结束语.....19

习题二.....20

一、选择题.....20

二、填空题.....21

三、简答题.....22

四、应用题.....23

第3章 寻址方式和指令系统.....24

3.1 学习要点.....24

3.1.1 寻址方式.....24

3.1.2 CPU 的指令系统.....25

3.2 典型例题.....31

结束语.....33

习题三.....34

一、选择题.....34

二、填空题.....37

三、简答题.....40

四、应用题.....41

第4章 汇编语言程序结构.....44

4.1 学习要点.....44

4.1.1 汇编语言的语句格式.....44

4.1.2 伪指令.....46

4.1.3 汇编语言的源程序结构.....51

4.1.4 汇编语言的上机过程.....52

4.2 典型例题.....53

结束语.....55

习题四.....56

一、选择题.....56

二、填空题.....58

三、简答题.....62

四、应用题.....64

第5章 基本结构程序设计.....67

5.1 学习要点.....67

5.1.1 顺序结构程序设计.....67

5.1.2 分支结构程序设计.....67

5.1.3 循环结构程序设计.....69

5.2 典型例题.....70

结束语.....72

习题五.....72

一、选择题.....72

二、填空题.....74

三、简答题.....77

四、应用题.....77

第6章 子程序设计及应用.....79

6.1 学习要点.....79

6.1.1 子程序设计	79	三、简答题	112
6.1.2 主程序与子程序的参数传递	82	四、应用题	113
6.1.3 子程序的嵌套和递归	83	第 9 章 输入/输出和中断程序设计	114
6.1.4 基本 DOS 功能子程序调用	83	9.1 学习要点	114
6.1.5 多模块程序设计	84	9.1.1 输入/输出指令	114
6.1.6 汇编语言和高级语言的 混合编程	85	9.1.2 查询传送方式	115
6.2 典型例题	86	9.1.3 中断简介	115
结束语	89	9.1.4 中断处理程序的设计	118
习题六	89	9.1.5 BIOS 功能调用	118
一、选择题	89	9.2 典型例题	119
二、填空题	91	结束语	122
三、简答题	91	习题九	122
四、应用题	92	一、选择题	122
第 7 章 简单程序设计应用	94	二、填空题	123
7.1 学习要点	94	三、简答题	125
7.1.1 串操作程序	94	四、应用题	126
7.1.2 代码与数制之间的转换	98	第 10 章 磁盘文件存取技术	127
7.1.3 冒泡排序	98	10.1 学习要点	127
7.1.4 多精度数运算	98	10.1.1 传统文件管理方式	127
7.2 典型例题	99	10.1.2 扩充文件管理方式	127
结束语	99	10.2 典型例题	128
习题七	100	结束语	129
一、选择题	100	习题十	129
二、填空题	101	一、选择题	129
三、简答题	103	二、填空题	130
四、应用题	103	三、简答题	132
第 8 章 高级汇编语言技术	105	四、应用题	132
8.1 学习要点	105	第 11 章 彩色图形程序设计	133
8.1.1 宏汇编	105	11.1 学习要点	133
8.1.2 重复汇编	106	11.1.1 单色与彩色显示控制	133
8.1.3 条件汇编	107	11.1.2 彩色显示适配器	133
8.2 典型例题	108	11.1.3 EGA/VGA 图形程序设计	134
结束语	109	11.1.4 计算机动画	135
习题八	110	11.2 典型例题	136
一、选择题	110	结束语	137
二、填空题	110	习题十一	137

一、选择题.....	137
二、填空题.....	138
三、简答题.....	138
四、应用题.....	139
第 12 章 发声系统的程序设计.....	140
12.1 学习要点.....	140
12.1.1 可编程内部定时器 8253/54	140
12.1.2 编程结构.....	140
12.1.3 通用发声程序	142
12.1.4 乐曲程序	142
12.2 典型例题.....	143
结束语.....	144
习题十二.....	144
一、选择题.....	144
二、填空题.....	144
三、简答题.....	145
四、应用题.....	145
第 13 章 汇编语言软件开发工具.....	146
13.1 汇编语言程序的开发过程.....	146
13.1.1 源程序的编辑.....	146
13.1.2 源程序的编译	147
13.1.3 目标程序的连接	147
13.1.4 调试可执行程序	147
13.1.5 COM 文件的生成	147
13.2 Turbo Assembler	147
13.2.1 句法分析.....	147
13.2.2 一次汇编多个文件	148
13.2.3 汇编器参数 option.....	148
13.3 Turbo Link.....	149
13.3.1 句法分析.....	149
13.3.2 TLINK 连接参数	149
13.4 Turbo Debugger.....	150
13.4.1 Turbo Debugger 调试界面	150
13.4.2 Turbo Debugger 功能.....	153
13.4.3 Turbo Debugger 应用举例	155
13.4.4 汇编语言调试软件的 文件组织.....	158

第二部分 实验指导

第 14 章 实验指导 1——源程序的建立、

汇编、连接和语法纠错..... 160

14.1 实验目的..... 160

14.2 实验内容..... 160

第 15 章 实验指导 2——分类统计..... 162

15.1 实验内容..... 162

15.1.1 编程思路..... 162

15.1.2 程序框图..... 162

15.1.3 程序清单..... 162

15.1.4 检查程序的运行结果 163 |

15.2 实验题

 166 |

第 16 章 实验指导 3——查找指定的

数据或字符

 168 |

16.1 实验内容..... 168

16.1.1 编程要求..... 168

16.1.2 编程思路..... 168

16.1.3 源程序清单..... 168

16.2 实验题

 169 |

第 17 章 实验指导 4——字符串搜索..... 171

17.1 实验内容..... 171

17.1.1 编程要求..... 171

17.1.2 编程思路..... 171

17.1.3 源程序清单..... 172

17.2 实验题

 173 |

第 18 章 实验指导 5——数制转换..... 175

18.1 实验内容..... 175

18.1.1 编程要求..... 175

18.1.2 编程思路..... 175

18.1.3 源程序清单..... 176

18.2 实验题

 177 |

第 19 章 实验指导 6——多精度数运算..... 179

19.1 实验内容一..... 179

19.1.1 编程要求	179	22.2.2 编程思路	199
19.1.2 编程思路	179	22.2.3 源程序清单	199
19.1.3 程序段清单	179	22.3 实验题	201
19.2 实验内容二	179		
19.2.1 编程要求	179	第三部分 模拟试卷	
19.2.2 编程算法	180	模拟试卷	204
19.2.3 程序框图	180	模拟试卷一	204
19.2.4 源程序清单	181	模拟试卷二	209
19.3 实验题	182	模拟试卷三	213
第 20 章 实验指导 7——宏指令的应用	183	模拟试卷四	218
20.1 实验内容	183	模拟试卷五	224
20.1.1 编程要求	183	模拟试卷参考答案	229
20.1.2 编程思路	183	附录 A BIOS 中断调用	239
20.1.3 程序框图	184	附录 B 出错信息	243
20.1.4 源程序清单	184	B.1 带编号的错误	243
20.2 实验题	185	B.2 不带编号的错误	249
第 21 章 实验指导 8——中断应用编程	187	B.2.1 访问文件错误	249
21.1 实验内容一	188	B.2.2 命令行错误	249
21.1.1 编程要求	188	B.2.3 其他错误	249
21.1.2 编程思路	189	参考答案	251
21.1.3 源程序清单	189	第 1 章	251
21.2 实验内容二	192	第 2 章	253
21.2.1 编程要求	192	第 3 章	255
21.2.2 编程思路	192	第 4 章	258
21.2.3 源程序清单	192	第 5 章	261
21.3 实验题	193	第 6 章	270
第 22 章 实验指导 9——磁盘文件操作	195	第 7 章	281
22.1 实验内容一	195	第 8 章	290
22.1.1 编程要求	195	第 9 章	292
22.1.2 编程思路	195	第 10 章	295
22.1.3 源程序清单	195	第 11 章	300
22.2 实验内容二	198	第 12 章	303
22.2.1 编程要求	198		

第一部分 章节复习与习题

这一部分主要对主教材各章的重要知识要点进行回顾，同时介绍相关的典型例题及各类习题。读者可以在复习各章节内容的基础上练习做一些习题，以巩固和提高所学知识。如果对某些习题有疑问或困难，可以参考本书的习题参考答案。第一部分结构安排如下：

- 第1章 概述
- 第2章 IBM PC 微处理器的结构及存储器组成
- 第3章 寻址方式和指令系统
- 第4章 汇编语言程序结构
- 第5章 基本结构程序设计
- 第6章 子程序设计及应用
- 第7章 简单程序设计应用
- 第8章 高级汇编语言技术
- 第9章 输入/输出和中断程序设计
- 第10章 磁盘文件存取技术
- 第11章 彩色图形程序设计
- 第12章 发声系统的程序设计
- 第13章 汇编语言软件开发工具

第 1 章 概 述

本章介绍的是一些计算机系统及汇编语言方面的基础知识，内容主要包括计算机系统软硬件知识、计算机的硬件环境、计算机中数据的表示方法和数据类型等。这是学习汇编语言必须预先具备的知识。在学习汇编语言之前，应该仔细熟悉这些内容。

1.1 学习要点

1.1.1 计算机系统简介

1. 硬件系统结构

一台完整的计算机系统由硬件系统和软件系统两大部分组成。

从硬件体系结构来看，目前的计算机都是采用冯·诺依曼体系结构。冯·诺依曼结构的主要特点包括 3 个方面：

(1) 硬件系统由运算器、控制器、存储器、输入设备和输出设备 5 大部分组成，这 5 个部分之间既有分工又有合作，共同完成处理任务。

(2) 计算机所处理的数据都以二进制形式进行存储和处理，描述反映数据存储位置的地址也是二进制数。

(3) 计算机以运算器和控制器（合称为中央处理单元）为中心，在控制器的作用下从存储器中依次读出指令和数据，在运算器中执行相应操作，最终结果也在控制器的作用下写回存储器中。这一过程是反复取指令、指令译码和执行指令的过程。

硬件系统涉及的术语有：

(1) 运算器 (ALU)。

运算器又称为算术逻辑单元，用来进行算术或逻辑运算以及位移循环等操作。

(2) 控制器 (CU)。

控制器负责统一指挥计算机各部分协调工作，能根据事先安排好的指令发出各种控制信号，以控制计算机各个部分的工作。控制器包括指令寄存器 IR、指令译码器 ID 和可编程逻辑阵列 PLA 共 3 部分。

(3) 存储器 (Memory)。

存储器是计算机的记忆部件，负责存储处理器要运行的程序和数据。

(4) 输入/输出设备 (I/O)。

输入/输出设备 (I/O) 一般包括一些 I/O 设备和存储器两类。

2. 软件系统结构

计算机软件主要分为应用软件和系统软件两大类。

应用软件是用户为解决科学计算、办公自动化、检测与实时控制等不同领域的任务所编制的应用程序。

系统软件是指不需用户干预的能生成、准备和执行其他程序所需的一组程序。常见的操作系统就属于系统软件。

1.1.2 汇编语言简介

1. 一般概念

1) 机器语言

机器语言是用来直接描述机器指令、使用机器指令的规则及用二进制代码来表示机器指令的一种程序设计语言，是 CPU 能直接识别的惟一语言。机器语言程序的执行效率高，但却难以理解，难读懂。另外，机器语言难于移植，通用性不高。

2) 汇编语言

汇编语言是用符号化的指令助记符来表示二进制机器指令，助记符一般是能够说明指令功能的英语词汇或词汇的缩写。用指令助记符、符号地址等组成的符号指令称为汇编格式指令（或汇编指令）。

汇编语言是汇编指令集、伪指令集和使用它们规则的统称。

汇编语言程序要比用机器指令编写的程序容易理解和维护。

汇编语言的每一条指令都与一条机器指令相对应。

2. 汇编程序

汇编程序是用汇编语言编写的程序，它大大提高了程序的可读性，但失去了 CPU 能直接识别的特性。执行汇编语言程序时，需要用汇编程序将源程序翻译成 CPU 能识别的机器指令序列。

常用的汇编程序开发工具有：Microsoft 公司的 MASM、Borland 公司的 TASM 和 DEBUG 等。

3. 汇编语言的特点

（1）与机器相关性。

汇编语言程序指令是机器指令的一种符号表示，不同类型的 CPU 有不同的机器指令系统，因此有不同的汇编语言。

（2）执行的高效率。

汇编语言的执行速度快、执行效率高。

（3）编写程序的复杂性。

汇编语言是一种面向机器的语言，编程时要安排运算的每一个细节，因此程序的编写过程比较繁琐、复杂。

（4）调试的复杂性。

在通常情况下，调试汇编语言程序要比调试高级语言程序困难。

4. 汇编语言的使用场合

（1）要求执行效率高、反应快的领域。

（2）系统性能的瓶颈，或在大程序中频繁被使用的子程序或程序段。

（3）软件与硬件资源密切，软件要直接和有效控制硬件的场合。

（4）对执行时间和存储容量要求较高的场合。

（5）没有合适的高级语言的场合。

5. 如何学习汇编语言

（1）熟悉 CPU 内部结构和工作原理，熟悉掌握 CPU 的指令系统（包括伪指令），数

据存储组织方式,这是学好汇编语言的前提。

(2) 熟练掌握指令的寻址方式,这是编写汇编语言程序的重要条件。

(3) 通过分析问题得到编程思路及解决方案。

(4) 多读源程序,从研究例子下手,从中获取一些编程的习惯和典型算法。

(5) 培养一种抽象思维能力,汇编语言在某种程度上要编程者想象程序的执行过程,画出数据在存储单元中的存储方式,这有助于程序的编写。

1.1.3 计算机中数据表示方法

1. 数与数制

1) 进位计数制

按进位的原则进行计数,称为进位计数制。常用的进位计数制有十、二、八和十六进制等。进位计数制的基本特点如下:

(1) 逢 R 进一。 R 是计数制中所用到的数码的个数。

(2) 采用位权表示法。在一个进位计数制表示的数中,处在不同数位的数码,代表着不同的数值,某一个数位的数值是由这一位数码的值乘上处在这位的一个固定常数。不同数位上的固定常数称为该位的位权或权。不同数位上有不同的位权值。

一般来说,一个 R 进制数 N 有两种表示方式:

(1) 并列表示方式:

$$(N)_R = (K_{n-1}K_{n-2} \cdots K_1K_0.K_{-1}K_{-2} \cdots K_{-m})_R$$

其中 n 为整数的位数; m 为小数的位数; R 表示基数; K_i 为不同数位上的数值:

$$0 \leq K_i \leq R-1$$

(2) 多项式表示法,也称按权展开式:

$$(N)_R = K_{n-1}R^{n-1} + K_{n-2}R^{n-2} + \cdots + K_1R^1 + K_0R^0 + K_{-1}R^{-1} + \cdots + K_{-m}R^{-m}$$

或:

$$(N)_R = \left(\sum_{i=-m}^{n-1} K_i R^i \right)_R$$

其中 R 代表进位制的基数; m 、 n 为正整数,分别代表 N 的整数部分的位数和小数部分的位数, K 是 R 进制中 R 个数字符号中的任何一个:

$$0 \leq K_i \leq R-1$$

2) 各进制书写规范及对应关系

(1) 二进制: 基数 $R=2$, 基本符号 0、1, 进位规律是“逢二进一”, 尾部标识符 B, 如: 10010B, 10011.101B。

(2) 八进制: 基数 $R=8$, 基本符号 0、1、2、3、4、5、6、7, 进位规律是“逢八进一”, 尾部标识符 O, 如: 65O, 753O。

(3) 十六进制: 基数 $R=16$, 进位规律是“逢十六进一”, 尾部标识符 H, 基本符号 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F(其中 A~F 分别表示十进制数 10、11、12、13、14、15), 如: 3AE7H, 083EH。

(4) 十进制: 基数 $R=10$, 进位规律是“逢十进一”, 尾部标识符 D 或省略, 基本符号 0、1、2、3、4、5、6、7、8、9, 如: 4783D 或 4783。

2. 不同进制数据间的转换

(1) 二(八、十六)进制转换成十进制数据。

将非十进制数 N 转换成十进制数, 只需将 N 按权展开相加即可。

(2) 十进制数转换为二进制数。

十进制数到二进制数的转换, 通常要区分数的整数部分和小数部分, 分别按除 2 取余和乘 2 取整两种不同的方法来完成。

① 对整数部分, 要用除 2 取余法倒序写。

② 对小数部分, 要用乘 2 取整法顺序写。

(3) 十进制数转换为八进制数

十进制数到八进制数的转换过程也可以参照十进制数到二进制数转换的方法实现。

(4) 十进制数转换为十六进制数。

十进制数到十六进制数的转换方法与十进制转换为二进制和八进制的方法类似。

(5) 二、八、十六进制数的关系与转换。

在把二进制数转换成八进制或十六进制表示形式时, 可对每三位或每四位二进制位进行分组, 分组时应保证从小数点所在位置分别向左和向右进行划分, 若小数点左侧(即整数部分)的位数不是 3 或 4 的整数倍, 可以按在数的最左侧补零的方法处理, 对小数点右侧(即小数部分), 应按在数的最右侧补零的方法处理。对不存在小数部分的二进制数(整数), 应从最低位开始向左把其每 3 位划分成一组, 使其对应一个八进制数, 或把其每 4 位划分成一组, 使其对应一个十六进制数。

3. 常用各进制数据的运算

1) 二进制数的运算规则

二进制数之间可以执行算术与逻辑运算, 其规则简单, 容易实现。

(1) 加法规则: $0+0=0$;

$$0+1=1;$$

$$1+0=1;$$

$$1+1=0 \text{ 且产生向高位的一次进位 (即结果为 } 10)$$

(2) 减法规则: $0-0=0$;

$$0-1=1 \text{ 要向高位借位一次}$$

$$1-0=1;$$

$$1-1=0$$

(3) 乘法规则: $0\times 0=0$;

$$0\times 1=0;$$

$$1\times 0=0;$$

$$1\times 1=1$$

(4) 除法规则: 二进制除法的计算方法, 与十进制除法类似, 也由减法、上商等操作逐步完成。

2) 十六进制数的运算规则

(1) 加法规则: “逢十六进一”。

(2) 减法规则：“借一当十六”。

(3) 乘法规则：

① 与十进制乘法一样，逐位相乘，结果错位相加。

② 两位相乘时，化为十进制数相乘。

③ 两位相乘的结果除以 16，商做进位，余数留本位。

(4) 除法规则：除法的计算方法，与十进制除法类似，也由减法、上商等操作逐步完成，但应是“逢十六进一”。

3) 八进制数的运算规则

八进制数的运算规则与十六进制数的运算规则相类似。

4. 计算机中的数据表示与编码

所谓编码，就是用少量简单的基本符号，选用一定的组合规则，以表示出大量复杂多样的信息。基本符号的种类和这些符号的组合规则是一切信息编码的两大要素。

一个数的原值称为真值。在计算机中，数的符号也用二进制数来表示，连符号一起数字化了的数称为机器数。一般最高有效位为符号位，用 0 代表正数，1 表示负数。

1) 原码

原码是一种简单直观表示方法，它是将真值的符号位数字化，将机器数的最高一位作为符号位。数据 0 的原码有两种，即正 0 和负 0：

$$[+0]=000\cdots 000$$

$$[-0]=100\cdots 000$$

2) 反码

反码是原码与补码之间的过渡码，正数的反码与原码相同，负数的反码最高位为 1，其余各位依次取反。数据 0 的反码也有两种，如下所示：

$$[+0]=000\cdots 000$$

$$[-0]=111\cdots 111$$

3) 补码

带符号数 X 的补码定义如下：

$$[X]_{\text{补}} = X, \text{ 若 } X \geq 0;$$

$$[X]_{\text{补}} = 2^n + X \text{ MOD } (2^n), \text{ 若 } X < 0;$$

其中，n 为机器数的位数。

在计算过程中如果产生进位自动舍去，n 位的机器数用补码所能表示的数的范围为： $[-2^{n-1}, 2^{n-1}-1]$ 。

补码中没有正 0 和负 0 之分，即 $[0]_{\text{补}}=000\cdots 000$ 。

4) 补码的运算

二进制整数在计算机内做加减运算时是用补码进行的，其过程为：

运算数据（原码） $\longrightarrow$ 补码 $\longrightarrow$ 运算结果（补码） $\longrightarrow$ 原码

补码的加、减法规则是：

$$[X]_{\text{补}} + [Y]_{\text{补}} = [X+Y]_{\text{补}}$$

$$[X]_{\text{补}} - [Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = [X-Y]_{\text{补}}$$

如果有进位,则自然丢失,从而得到正确结果。

5) 溢出及其判断

当运算结果超出计算机所能表示的范围时,就产生溢出。

运算结果如果超出数据能表示的最大正数或最小负数,就会产生溢出。

6) BCD 码

BCD 码即 8421 码,它用 4 位二进制数来表示十进制数,4 个二进制位的权从高向低分别为 8、4、2 和 1,使用的是基 2 码的 0000、0001、...1001 这 10 种组合,分别表示 0 到 9 这 10 个值。BCD 码包括非压缩的 BCD 码和压缩的 BCD 码两种。

在计算机内实现 BCD 数之间的算术运算时,在某些情况下,需要对运算的结果进行“加 6 修正”。

7) ASCII 码

ASCII 编码即 American Standard Code for Information interchange,是一种常用的字符表示形式。它对 128 个字符进行了编码,其中包括大写字母 A~Z,小写字母 a~z,数字 0~9 及回车、换行和响铃等非控制字符。

有些机器的编码采用 8 位的扩展 ASCII 码编码,可表示 256 个字符。有些时候第 8 位用于奇偶校验。

8) 国标码

GB2312-80 是我国在计算机中表示汉字的标准编码。该标准规定了汉字交换用的基本汉字字符和一些图形字符,它们共计 7445 个,其中汉字有 6763 个。汉字编码表共 94 行(区)和 94 列(位)。行号称为区号,列号称为位号。国标码是直接取第一字节和第二字节编码拼起来得到的,通常用十六进制表示。在一个汉字的区码和位码上分别加十六进制数 20H,即构成该汉字的国标码。

目前,计算机中文平台广泛采用的汉字编码是变形国标码。

5. 基本的数据类型

1) 字节

一个字节由 8 个二进制位组成,最低位是第 0 位,最高位是第 7 位。由于存储器是以字节为单位来组织的,所以存储器的每一单元都是一个字节。

2) 字

一个字由两个字节组成,有 16 个二进制位,最低位是第 0 位,最高位是第 15 位。高 8 位称为高字节,低 8 位称为低字节。它在存储时通常占 2 个连续的字节(存储)单元。

3) 双字

双字由两个字组成,32 个二进制位,占用四个连续的字节(存储)单元。

4) 四字

四字由 4 个字组成,共 64 个二进制位,占用 8 个连续的存储单元。

5) 十字节

由 10 个字节组成,80 个二进制位。

6) 二进制数的符号扩展

符号扩展的具体操作就是把已知信息的最高位扩展到所有更高位。

7) n 位二进制数的表示范围

n 位二进制所能表示的无符号整数的范围: $0 \leq x \leq 2^n - 1$ 。

n 位二进制所能表示的有符号整数(补码表示)的范围: $-2^{n-1} \leq x \leq 2^{n-1} - 1$ 。

在汇编语言中, 常用到 n 为 8 和 16 时的数值范围如下:

$n=8$ 时, 无符号整数的范围: $0 \sim 255$; 有符号整数的范围: $-128 \sim 127$ 。

$n=16$ 时, 无符号整数的范围: $0 \sim 65535$; 有符号整数的范围: $-32768 \sim 32767$ 。

1.2 典型例题

【例 1-1】 计算机中存储信息的最小单位是 ()。

- A. 位
- B. 字节
- C. 字
- D. 字符

【分析与解答】 这是常识, 在计算机中, 每个存储元存储一位二进制信息位 (0 或 1)。

答案: A

【例 1-2】 CPU 惟一能够直接识别的语言是 ()。

- A. 汇编语言
- B. 机器语言
- C. BASIC 语言
- D. 高级语言

【分析与解答】 CPU 能直接识别的语言只有机器语言, 机器语言是一组二进制数据 0 或 1 的组合。

答案: B

【例 1-3】 在计算机内部, 字符通常采用 () 编码。

- A. 国标码
- B. BCD 码
- C. ASCII 码
- D. 原码

【分析与解答】 这是常识, 应该熟记, 在计算机内部, 字符 (不是汉字) 最常用的表示方法是用 ASCII 码, 它已成为国际标准。

答案: C

【例 1-4】 正 0 和负 0 的 8 位反码表示为 ()。

- A. 00000000 和 10000000
- B. 00000000 和 11111111
- C. 00000000 和 11111110
- D. 00000000 和 00000000

【分析与解答】 在计算机中, 数据 0 的反码有两种, 如下所示:

$[+0] = 000 \cdots 000$

$[-0] = 111 \cdots 111$

答案: B

【例 1-5】 计算机通过 () 将汇编源程序转换成 CPU 可直接识别的机器语言。

- A. TASM
- B. 汇编程序
- C. 解释程序
- D. 译码器

【分析与解答】 汇编程序不能直接由机器执行, 而必须先有专门的汇编程序将其翻译成机器语言后, 才能由机器执行。

答案: B