教育部高职高专规划教材

Jiaoyubu Gaozhi Gaozhuan Guihua Jiaocai

汇编语言程序设计

——方法·技术·应用

周学毛 主编

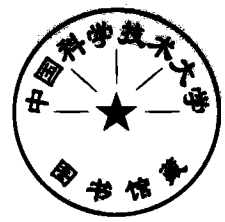
高等教育出版社

教育部高职高专规划教材

汇编语言程序设计

——方法·技术·应用

周学毛　主编



高等教育出版社

内容提要

本书是教育部高职高专规划教材。

“汇编语言程序设计”是计算机专业学生必修的一门核心专业课程，对培养程序设计能力，理解计算机工作原理，从事软件开发和硬件应用均具有非常重要作用。

本书以80X86与奔腾微机为背景，以8086汇编语言程序设计为核心，以MASM5.0与MASM6.0为实践环境，以实模式为主体，在详细介绍汇编语言基本概念和基本语法的基础上，全面地讲述汇编语言程序设计的一般方法、实用技术和应用范例。

本书严格按照教育部高职高专规划教材的要求编写，内容全面，语言简明，难点分散，例题丰富，习题多样。全书共10章，第1、2、3章讲述汇编语言程序与汇编语言程序设计的基本常识，第4、5、6章讲述汇编语言中程序三种基本结构的实现，第7、8章讲述子程序设计、中断和系统功能调用，第9、10章讲述汇编语言程序设计方法与汇编语言程序设计应用。

本书是一本有特色的汇编语言程序设计教材，适合于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院和民办高校计算机及相近专业使用，亦是一本难得的自学教材。

图书在版编目(CIP)数据

汇编语言程序设计——方法·技术·应用/周学毛主编.
—北京：高等教育出版社，2002.7(2003重印)
ISBN 7-04-010835-6

Ⅰ.汇… Ⅱ.①周… ②潘… Ⅲ.汇编语言—程序设计—高等学校：技术学校—教材 Ⅳ.TP313

中国版本图书馆CIP数据核字(2002)第030962号

汇编语言程序设计——方法·技术·应用
周学毛 主编

出版发行	高等教育出版社	购书热线	010-64054588
社　　址	北京市西城区德外大街4号	免费咨询	800-810-0598
邮政编码	100011	网　　址	http://www.hep.edu.cn
总　　机	010-82028899		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	北京民族印刷厂		
开　　本	787×1092　1/16	版　　次	2002年7月第1版
印　　张	18	印　　次	2003年9月第4次印刷
字　　数	420 000	定　　价	22.70元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

责任编辑　孙淑华
封面设计　王凌波
版式设计　陆瑞红
责任校对　殷　然
责任印制　陈伟光

郑重声明

出 版 说 明

教材建设工作是整个高职高专教育教学工作中的重要组成部分。改革开放以来，在各级教育行政部门、学校和有关出版社的共同努力下，各地已出版了一批高职高专教育教材。但从整体上看，具有高职高专教育特色的教材极其匮乏，不少院校尚在借用本科或中专教材，教材建设仍落后于高职高专教育的发展需要。为此，1999 年教育部组织制定了《高职高专教育基础课程教学基本要求》（以下简称《基本要求》）和《高职高专教育专业人才培养目标及规格》（以下简称《培养规格》），通过推荐、招标及遴选，组织了一批学术水平高、教学经验丰富、实践能力强的教师，成立了“教育部高职高专规划教材”编写队伍，并在有关出版社的积极配合下，推出一批“教育部高职高专规划教材”。

“教育部高职高专规划教材”计划出版 500 种，用 5 年左右时间完成。出版后的教材将覆盖高职高专教育的基础课程和主干专业课程。计划先用 2～3 年的时间，在继承原有高职、高专和成人高等学校教材建设成果的基础上，充分汲取近几年来各类学校在探索培养技术应用性专门人才方面取得的成功经验，解决好新形势下高职高专教育教材的有无问题；然后再用 2～3 年的时间，在《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》立项研究的基础上，通过研究、改革和建设，推出一大批教育部高职高专教育教材，从而形成优化配套的高职高专教育教材体系。

“教育部高职高专规划教材”是按照《基本要求》和《培养规格》的要求，充分汲取高职、高专和成人高等学校在探索培养技术应用性专门人才方面取得的成功经验和教学成果编写而成的，适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院和民办高校使用。

教育部高等教育司

2000 年 4 月 3 日

前　言

汇编语言是能够直接利用计算机所有硬件特性并直接控制计算机硬件的惟一语言，是计算机的基础语言。“汇编语言程序设计”是计算机专业学生必修的一门核心专业课程，对培养学生程序设计能力，理解计算机工作原理，从事软件开发和硬件应用均具有非常重要的作用。

本书以 80X86 与奔腾微机为背景，以 8086 汇编语言程序设计为核心，以 MASM 5.0 与 MASM 6.0 为实践环境，以实模式为主体，在详细介绍汇编语言基本概念和基本语法的基础上，全面地讲述汇编语言程序设计的一般方法、实用技术和应用范例。全书共 10 章，第 1、2、3 章讲述汇编语言程序与汇编语言程序设计的基本常识，第 4、5、6 章讲述汇编语言中程序三种基本结构的实现，第 7、8 章讲述子程序设计、中断和系统功能调用，第 9、10 章讲述汇编语言程序设计方法与汇编语言程序设计应用。

本书的编写强调突出高职高专培养高等技术应用性人才的特色，严格按照教育部高职高专规划教材的要求编写，遵循以应用为目的，以必需、够用为度的教学原则。本书的编写立足在以程序设计为中心、突出程序设计能力培养的同时，还特别强调学生学习能力的培养。本书内容全面，语言简明，难点分散，例题丰富，习题多样，针对性强，是一本有特色的汇编语言程序设计教材，对使用本书的读者只要求有一门高级语言程序设计的知识基础即可。

本书没有涉及 80X86 汇编语言及保护模式编程与协处理器编程。编者在 1997 年曾尝试编写了基于 80X86 的汇编语言程序设计教材，经过这些年的教改实践，编者认为学习汇编语言的关键是先学好 8086 汇编语言，无论是从教、学的难度，还是从教、学的效果来看，以 8086 汇编语言程序设计为核心是一种最佳的选择。80X86 汇编语言向上兼容，有了 8086 汇编语言基础，相信可以较自然地向 80X86 及奔腾汇编语言程序设计延伸。

使用本书，建议安排 64~80 学时，其中理论教学 48 学时，实践教学 16~32 学时，并进行为期 1 周的课程设计。讲授的重点是第 2~7 章，第 8~10 章以学生自学为主。

本书由周学毛副教授主持编写，潘建军老师参加编写第 7 章、第 8 章、第 9 章与第 10 章，由张如健副教授主审。

本书终能与读者见面，除了要衷心感谢所有参考文献的作者，还要特别感谢高等教育出版社前后三位责任编辑李琰、李慧、孙淑华的大力支持与辛勤劳动。

最后编者有一个忠诚的愿望和一个殷切的期待，希望与读者交流，希望能得到读者使用本书的宝贵意见与建议。编者的 Email：hnzhou@sina.com。

周学毛

2002 年春节于岳麓山

目　录

第 1 章　汇编语言程序设计基础

本章介绍学习汇编语言程序设计所必须的计算机软硬件基础知识，包括汇编语言的特点、微计算机的基本结构、计算机中的数据表示和汇编语言程序设计的一般步骤及方法。

1.1　机器语言与汇编语言

程序设计语言是程序设计人员和计算机进行会话的语言，它遵循一定的规则和形式，构成程序的实现工具。

设计程序设计语言时有一基本前提：既要让计算机能识别处理，又要让程序设计人员感到方便。机器与人之间相距甚远，兼顾二者设计了多种程序设计语言。接近机器的语言称为低级语言，接近人的语言称为高级语言。机器语言与汇编语言属于低级语言。

1.1.1　指令与程序

任何计算机都在程序的控制下进行有效的工作，程序工作由一系列指令完成。

指令用于指出计算机要进行的特定操作和操作对象，实际上是计算机能识别的一组二进制代码。一台计算机的指令系统是该计算机全部指令的集合。指令系统是计算机基本功能的体现，计算机的性能与它有很大关系，不同类型的计算机的指令系统差异较大。

程序是为使计算机实现预期目的而用程序设计语言设计的一系列操作，其最终形式是指令序列。

1.1.2　机器语言

机器语言是以计算机硬件能直接执行和理解的指令系统为基础而形成的语言，它与计算机硬件密切相关，为特定的计算机而设计。

相应的用机器语言编写的程序称为机器语言程序。机器语言直接对硬件编程，编写的程序占用资源少，运行速度快，效率高；但编写难、调试难，不易阅读、不易理解，通用性也差。

计算机可以直接执行机器语言程序。

机器语言除用于编写计算机最底层的核心系统程序外，实际应用中很少直接使用。

1.1.3　汇编语言

汇编语言是一种符号化的机器语言，即用助记符号代替机器语言的二进制代码。助记符号一般是英语单词缩写，方便了人们的书写、阅读和检查。汇编语言的引入在一定程度上克服了机器语言的不足，同时保留了机器语言的长处。

汇编语言的指令基本上对应机器语言的指令，通常专属于某种机型或某一系列机型。用汇编语言编写的程序就是机器语言程序的符号表示，能够直接利用计算机硬件系统的特性，能

够直接利用 CPU 的指令系统和各种寻址方式，能够将计算机的功能全面提供给程序设计者。用汇编语言能够编写高质量的程序，但也要求程序设计者在计算机寄存器一级进行操作，对计算机性能有更深的了解。汇编语言一般用在系统软件的开发扩充，或修改系统设备，或对速度、空间有特别要求的程序设计。

实际使用的往往是宏汇编语言，它包含一般汇编语言的功能，采用了高级语言的一些特性，是一种接近高级语言的汇编语言。

用汇编语言编写的程序称为汇编语言源程序，汇编语言源程序在计算机中不能直接运行，必须通过一个称为汇编程序的机器语言程序将源程序翻译成相应的目标代码程序才能运行。汇编程序是系统软件的核心成分，汇编语言源程序通过汇编程序翻译成目标代码程序的过程称为汇编。汇编形成的目标代码程序还不能直接运行，必须经过连接程序连接形成可执行的代码程序才能运行。

汇编语言是计算机的基础语言，涉及计算机底层硬件，学习内容枯燥，记忆东西繁多，起步困难，因此，学习时要有思想准备。

1.2　数据表示方法

计算机作为信息处理工具，数据表示问题是计算机工作的出发点。计算机内部基于硬件处理的简单的原因，采用二进制（Binary）作为数据表示的基础，日常生活中习惯使用十进制（Decimal）数据，汇编语言程序中喜欢使用十六进制（Hexadecimal）数据。

1.2.1　数制及其转换

1. 二进制数

二进制数由 0、1 两个数码构成，基数为 2，第 i 位的权为 2^i，运算逢 2 进 1。

书写时在尾部加注字母 B 或下标 2 描述。其他进制数的书写加注相应的字母或下标描述，缺省一般指十进制数。

二进制数 $a_n a_{n-1}\cdots a_0.b_{-1}b_{-2}\cdots b_{-m}$ 的值是：

$a_n\times 2^n+a_{n-1}\times 2^{n-1}+\cdots+a_0+b_{-1}\times 2^{-1}+b_{-2}\times 2^{-2}+\cdots+b_{-m}\times 2^{-m}$，

其中 a_i 和 b_i 为 0、1 两个数码中一个。

【例 1-1】　101011B=$1\times 2^5+0\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0$= 43D

100000000B=1×2^8=256D

n 位二进制数可以表示 2^n 种组合。4 位能表示 16 种组合（0 至 15 的整数），8 位能表示 256 种组合（0 至 255 的整数），16 位能表示 64×1024（64K）种组合。

2. 十六进制数

使用二进制数在计算机中容易实现，便于存储，抗干扰性强。但位数较多时，阅读、书写、记忆、输入等不太方便。

若干年前，程序员发现操作一般是对位的“组”，而不是个别的一些位操作。 最早的微处理器 4 位，自然想到用 4 位一组缩写二进制数，即十六进制数。

十六进制的数由 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个数码构成，

其中数码 A、B、C、D、E、F 对应于十进制数 10、11、12、13、14、15，基数为 16，第 i 位的权为 16^i，运算逢 16 进 1。

书写时在尾部加注字母 H 或下标 16 描述。

描述以字母开头的十六进制数前面应加一数码 0，以与标识符相区别。

十六进制数 $a_n a_{n-1}\cdots a_0.b_{-1}b_{-2}\cdots b_{-m}$ 的值是：

$a_n\times16^n+a_{n-1}\times16^{n-1}+\cdots+a_0+b_{-1}\times16^{-1}+b_{-2}\times16^{-2}+\cdots+b_{-m}\times16^{-m}$，其中 a_i 和 b_i 为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个数码中一个。

【例 1-2】 $1011H=1\times16^3+0\times16^2+1\times16^1+1\times16^0=4113D$

$14AFH=1\times16^3+4\times16^2+10\times16^1+15\times16^0=5295D$

汇编语言调试系统中的数全部使用十六进制数，汇编列表文件中代码指令及内存地址也使用十六进制数表示。可以说十六进制数是汇编语言的书写工具，应尽快习惯使用。

3. 八进制数

计算机中一个字节是八位，也常用八进制（Octal）数来缩写二进制数。

八进制数由 0、1、2、3、4、5、6、7 八个数码构成，基数为 8，第 i 位的权为 8^i，运算逢 8 进 1。

书写时在尾部加注字母 O 或下标 8 描述。

由于字母 O 与数字 0 容易混淆，八进制亦常用尾标 Q 标识。

八进制数 $a_n a_{n-1}\cdots a_0.b_{-1}b_{-2}\cdots b_{-m}$ 的值是：

$a_n\times8^n+a_{n-1}\times8^{n-1}+\cdots+a_0+b_{-1}\times8^{-1}+b_{-2}\times8^{-2}+\cdots+b_{-m}\times8^{-m}$，其中 a_i 和 b_i 为 0、1、2、3、4、5、6、7 八个数码中一个。

【例 1-3】 $4056Q=4\times8^3+0\times8^2+5\times8^1+6\times8^0=2094D$

$1001Q=1\times8^3+0\times8^2+0\times8^1+1\times8^0=513D$

4. 非十进制数与十进制数转换

非十进制数转换成十进制数，只需采用按权相加法，将各位数码与其对应权值的乘积相加即可。

【例 1-4】 $101101.01B=1\times2^5+1\times2^3+1\times2^2+1+1\times2^{-2}=45.25D$

$0F2DH=15\times16^2+2\times16^1+13\times16^0=3885D$

$345Q=3\times8^2+4\times8^1+5\times8^0=229D$

十进制数转换成非十进制数，转换时将整数部分与小数部分分别转换。整数部分采用除基数取余法，直至商为 0，先得到的余数为低位，后得到的余数为高位。小数部分采用乘基数取整法，直至乘积为整数或达到控制精度。

【例 1-5】 17D=10001B；17/2=8…1，8/2=4…0，4/2=2…0，2/2=1…0，1/2=0…1

57D =71Q；57/8=7…1，7/8=0…7

57D =39H；57/16=3…9，3/16=0…3

【例 1-6】 0.625D=0.101B；0.625×2=1.25…1，0.25×2= 0.5…0，0.5×2=1…1

0.625D =0.5Q ；0.625×8=5…5

0.625D =0.AH ；0.625×16=10…A

5. 二、八、十六进制数之间的转换

二、八、十六进制数之间的转换可以先将其转换成十进制数，再将对应十进制数转换成需要转换的进制数。

由于二、八、十六进制数之间的特殊关系，还可以采用分组法快速转换。将二进制数转换成八进制数可按三位一组进行分组，转换成十六进制数可按四位一组进行分组，每一组对应八进制或十六进制的相应数码，参见表 1-1，组合即得转换结果。分组时如位不够，整数部分在最左边补 0，小数部分在最右边补 0。

将八进制数转换成二进制数，只需将八进制数每一位对应转换成二进制数三位即可。同样将十六进制数转换成二进制数，只需将其每一位对应转换成二进制数四位即可。

表 1-1　二、八、十六进制对应关系表

二进制	001	010	011	100	101	110	111								
八进制	1	2	3	4	5	6	7								
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

【例 1-7】　1100100.11010B=001 100 100. 110 100=144.64Q

1100100.11010B= 0110 0100 .1101 0000=64.D0H

反过来 1100100.11010B 即是八进制数 144.64Q、十六进制 64.D0H 转换所得的二进制数。

1.2.2　数值数据编码

数的原值称为真值，是计算机中表示的数的实际数值。

数在计算机中的二进制表示形式称为机器数。

机器数可以用不同的码制表示，常用的有原码、补码与反码三种，广泛使用的是原码和补码两种，有符号数采用补码表示。常将数用中括号括起来，在尾部加注下标原、反、补明确码制。

机器数将数的符号也数字化，一般用最高有效位表示数的符号，“0”代表正数，“1”代表负数。

机器数的表示形式还与存储位数有关。

【例 1-8】　数+1010B，相应机器数为 00001010；假定是字节表示

数–1010B，相应机器数为 10001010

机器数的形式值不一定等于真值，但与真值肯定存在某种对应关系。

1. 原码

原码表示数是一种比较直观的机器数的表示方法，原码与真值的区别仅仅是数的符号数字化。

【例 1-9】　X= +1011B，$[X]_{原}$=00001011；字节表示

Y= –1011B，$[Y]_{原}$=10001011

原码表示中，最高位为符号位，其他位表示数的绝对值。

n 位原码表示的数的范围为[$-2^{n-1}+1$，$2^{n-1}-1$]。8 位原码表示的数的范围是[-127，127]，16 位原码表示的数的范围是[-32767，32767]。

原码表示数简单直观，做乘除运算比较方便，可取绝对值直接运算，而符号单独处理。但对于应用得最多的加减运算，原码表示不太方便，像（-2）+3 表面上做加法操作，而实际需做（3-2）减法操作。

2. *反码*

正数的反码与原码相同，负数的反码等于其原码（除符号位）按位取反，也等于其对应正数的补码（含符号位）按位取反。

【例 1-10】 X= +1011B，$[X]_{反}$=00001011；字节表示

Y= -1011B，$[Y]_{反}$=11110100

反码表示数的范围与原码相同。

3. *补码*

补码表示能让符号位一同参与数的运算，能将减法转化成加法进行运算。

计算机的补码来源于数学中的“模”运算，是以 2^n 为模，基于 2 的补码。

正数的补码等于其本身，与其原码相同。负数的补码等于其原码（除符号位）按位取反，末位加 1。

【例 1-11】 X=+1011B，$[X]_{补}$=00001011；字节表示

Y= -1011B，$[Y]_{补}$=11110101

负数的补码也等于其对应正数的补码（含符号位）按位取反，末位加 1，也就是反码末位加 1。

$[X]_{补}+[Y]_{补}=[X+Y]_{补}$，符号位同时参加运算，只要不溢出，补的和等于和的补。

$[X-Y]_{补}=[X]_{补}+[-Y]_{补}$，减法运算转化成加法运算，符号位同时参加运算。

如何从补码求原码或真值，可对补码再一次求补得到，$[[X]_{补}]_{补}=[X]_{原}$。其实对于正数，补码的真值就等于其自身。

n 位补码表示的数的范围为[-2^{n-1}，$2^{n-1}-1$]。为扩大数的表示范围，可用多个字来表示一个数。8 位补码表示的数的范围是[-128，127]，16 位补码表示的数的范围是[-32768，32767]。

在计算机中，负数用补码表示，符号数用补码表示。

表 1-2 给出了 8 位二进制数的原码、补码及反码表示。

表 1-2 8 位二进制数的编码表示

二 进 制 数	原 码	补 码	反 码
00000000	+0	+0	+0
00000001	+1	+1	+1
00000010	+2	+2	+2
……	……	……	……
01111110	+126	+126	+126
01111111	+127	+127	+127
10000000	-0	-128	-127
10000001	-1	-127	-126

续表

二进制数	原　码	补　码	反　码
……	……	……	……
11111101	–125	–3	–2
11111110	–126	–2	–1
11111111	–127	–1	–0

4. 无符号数

在某些情况下，要处理的数据全是正数，此时保留符号位毫无意义。如将符号位也作为数值位处理，可形成无符号数。

【例 1-12】　10011001B，表示无符号整数是 $1\times2^7+1\times2^4+1\times2^3+1$，即 153

表示有符号整数是$-(1\times2^4+1\times2^3+1)$，即 –25

n 位无符号整数的范围为[0，2^n-1]，8 位无符号整数范围为[0，255]，16 位无符号整数范围为[0，65535]。

在计算机中，用无符号整数表示存储空间的地址。

5. BCD 码

BCD 码（Binary Coded Decimal）使用 4 位二进制数来表示 1 位十进制数，常称为二进制编码的十进制数。

4 位二进制数能表示 16 种状态，可用其中任意 10 种状态来表示十进制数字 0~9，由此形成 8421 码、2421 码、余 3 码等多种 BCD 码，最常用的是 8421 码。

8421 码编码方法见表 1-3，8421 是指用于编码的 4 位二进制各位的权，由此也不难推出 2421 码的编码方法。

表 1-3　8421 码编码表

十进制	0	1	2	3	4	5	6	7	8	9
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

【例 1-13】　1329 用 BCD 码表示为：0001 0011 0010 1001

67536 用 BCD 码表示为：0110 0111 0101 0011 0110

BCD 码的使用为十进制数在计算机内的表示提供了一种简单的手段，计算机提供了直接处理 BCD 码的一组指令，但 BCD 码参与运算有许多麻烦，这些指令在实际编程时很少使用。

十进制数的 BCD 码表示并不是十进制数通过数制转换形成的二进制数。

BCD 码有压缩 BCD 码和非压缩 BCD 码两种不同存储方式。压缩 BCD 码用 1 个字节存放两个十进制数字，每个十进制数字占 4 位，一个十进制数字占低 4 位，另一个十进制数字占高 4 位；而非压缩 BCD 码用 1 个字节存放 1 个十进制数字，每个十进制数字占每个字节的低 4 位，高 4 位一般为 0。

【例 1-14】　9329 用 BCD 码表示为：1001 0011 0010 1001

非压缩 BCD 码存放需 4 个字节，压缩 BCD 码存放只需 2 个字节。

1.2.3 字符数据编码

字符数据包括字母、数字、专用字符及一些控制字符，字符数据的使用方便了人和计算机交换信息，美国信息交换标准代码 ASCII 码（American Standard Code for Information Interchange）是最主要的字符编码方式。

ASCII 码采用 1 个字节的低 7 位进行编码，能表示 128 种字符。最高位用作奇偶校验位。

表 1-4 用十六进制形式给出了主要字符的 ASCII 码。数字 0~9 的 ASCII 码为 30H~39H，大写字母 A~Z 的 ASCII 码为 41H~5AH，小写字母 a~z 的 ASCII 码为 61H~7AH，控制字符的 ASCII 码为 00H~1FH 及 7FH，专用字符的 ASCII 码散列其中。

表 1-4 主要字符十六进制 ASCII 码表

字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII
NUL	00	+	2B	;	3B	K	4B	[	5B	k	6B
BEL	07	,	2C	<	3C	L	4C	\	5C	l	6C
LF	0A	–	2D	=	3D	M	4D	]	5D	m	6D
FF	0C	/	2E	>	3E	N	4E	↑	5E	n	6E
CR	0D	.	2F	?	3F	O	4F	←	5F	o	6F
SP	20	0	30	@	40	P	50	,	60	p	70
!	21	1	31	A	41	Q	51	a	61	q	71
“	22	2	32	B	42	R	52	b	62	r	72
#	23	3	33	C	43	S	53	c	63	s	73
$	24	4	34	D	44	T	54	d	64	t	74
%	25	5	34	E	45	U	55	e	65	u	75
&	26	6	36	F	46	V	56	f	66	v	76
‘	27	7	37	G	47	W	57	g	67	w	77
(	28	8	38	H	48	X	58	h	68	x	78
)	29	9	39	I	49	Y	59	i	69	y	79
*	2A	:	3A	J	4A	Z	5A	j	6A	z	7A

1.2.4 内存中的数据

数据在计算机内部采用何种方式，依赖于程序的执行情况，可用二进制、 BCD 码与字符方式，最终总是采用二进制存储。计算机在输入、输出及存储时采用编码，在运算过程的前后进行“码值”与“值码”的转换是必不可少的。

【例 1-15】 无符号整数 2002 以二进制方式、压缩 BCD 码方式、未压缩 BCD 码方式、字符方式表示，分别需 13 位、16 位、32 位、32 位存储。

数据的存储是以数据长度为单位，按字节顺序存放，数据的低位放低字节地址，数据的高位放高字节地址。简言之，低位低地址，高位高地址。

不加特别说明，汇编语言程序设计中提到的数都是整数，无符号整数，常用字节（8 位）、字（16 位）、双字（32 位）表示。如实际数据的位数少，对数据位数要进行扩展，一般是对