



21世纪高职高专机电类立体化精品教材·电子信息系列
工学结合教学改革与创新成果

汇编语言程序设计

HUIBIAN YUYAN
CHENGXU SHEJI

常国权 赵凯 张捐净 主编

教学资源包

电子课件: 包含8章PPT课件
教学参考: 包含课时规划和课程说明
课后习题: 课后习题答案
案例库: 补充教学案例
资源拓展: 包含学习网站和拓展阅读等
教学检测: 2份期末考试卷

教学服务: www.jxzy.com.cn

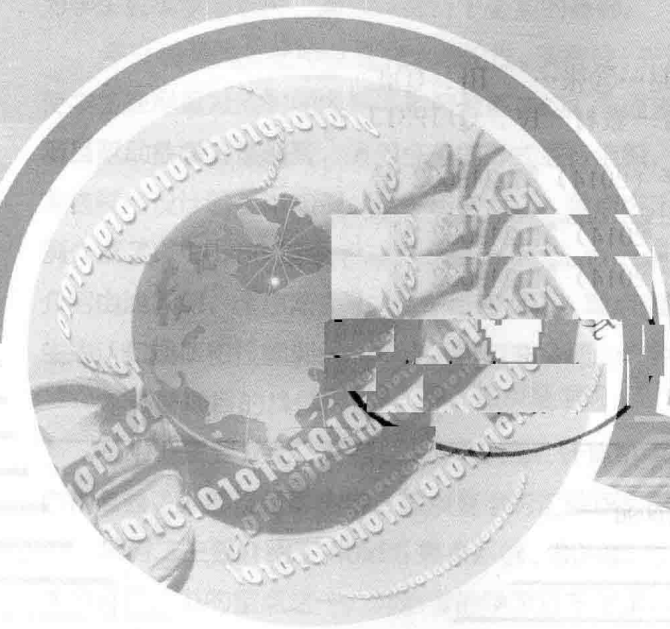


21世纪高职高专机电类立体化精品教材·电子信息系列
工学结合教学改革与创新成果

汇编语言程序设计

HUIBIAN YUYAN
CHENGXU SHEJI

常国权 赵凯 张捐净 主编
郭丽敏 孔娟 副主编



东南大学出版社
·南京·

内容提要

本书以 80x86 CPU 的指令系统为主,以项目驱动和任务的模式系统地介绍了汇编语言的基础理论知识和程序设计基本方法。全书共分八个项目,分别介绍了汇编语言基础知识、80x86 的指令系统、汇编语言程序结构、汇编语言程序设计基本方法、子程序和宏、输入/输出设计、文件存取技术、汇编语言扩展等内容。

本书在编排上注重理论与实践相结合,采用任务式教学模式,突出实践环节,充分体现“工学结合一体化”教学思想。本书将项目分解为若干任务,每个任务由任务描述、任务分析、知识准备、任务实施四部分组成,全书共设置任务 36 个。正文中设置了小提示、拓展提高以及知识链接等特色模块,意在提高学生的学习兴趣,促进学生的全面发展。全书共设置小提示 32 个,知识链接 11 个,拓展提高 10 个。每个项目最后设置了项目小结和项目考核内容。

本书可作为高职、高专计算机应用专业及相关专业的教材,也可作为自学考试及相关行业岗位培训用书或相关工程技术人员的参考用书,同时也可供从事计算机应用与开发的各类人员学习和使用。

图书在版编目(CIP)数据

汇编语言程序设计 / 常国权, 赵凯, 张捐净主编.
—南京: 东南大学出版社, 2014. 11
(21 世纪高职高专机电类立体化精品教材·电子信息系列)

ISBN 978-7-5641-5286-4

I. ①汇… II. ①常…②赵…③张… III. ①汇编语言—程序设计—高等职业教育—教材 IV. ①TP313

中国版本图书馆 CIP 数据核字 (2014) 第 249082 号

汇编语言程序设计

出版发行: 东南大学出版社

社 址: 南京市四牌楼 2 号, 邮编 210096

出版人: 江建中

印 刷: 北京旺银永泰印刷有限公司

开 本: 787mm×1092mm 1/16

印 张: 17

字 数: 354 千

版 次: 2014 年 11 月第 1 版

2014 年 11 月第 1 次印刷

书 号: ISBN 978-7-5641-5286-4

定 价: 39.00 元

(凡因印装质量问题, 请直接与营销中心调换, 电话: 025-83791830)

汇编语言程序设计是工院校和高等职业院校计算机专业及相关专业一门重要的专业技术基础课程。它讲解的是面向机器硬件的低级语言。汇编语言为程序员提供最直接操作机器硬件系统的途径,利用它可以编写出在“时间”和“空间”两个方面最具效率的程序。学习这门课程的主要任务是使学生掌握计算机硬件组成知识及汇编语言程序设计的方法,培养学生良好的程序设计风格与严密的逻辑思维能力,提高学生分析问题、解决问题的能力,培养学生的实践动手能力、独立思考能力、分析和解决问题的能力,为学生后续学习软件、硬件课程打下坚实的基础。

本书编写的主要特点是突出理论性、实践性、先进性、通俗性,力求教学、自学方便,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。教材以项目驱动教学方法编写,采用任务的形式进行组织,以通俗易懂的语言向读者讲述汇编语言程序设计的基本知识。全书结构清晰、重点突出,用简洁的语言、丰富的插图和实例介绍了以 80x86 指令系统为主的汇编语言基础理论知识和程序设计基本方法。项目介绍由浅入深、循序渐进,将汇编语言程序设计的基础知识融于项目实例之中,符合学生的认知规律和技能训练的特点,可以充分调动学生的学习积极性与主动性。

本书各项目的具体内容如下:

项目一主要介绍了汇编语言基础知识、计算机中数和字符的表示、80x86 微处理器、内存存储器、外部设备等内容,重点是学习汇编语言基础知识。

项目二主要介绍了 80x86 指令系统、80x86 寻址方式、80x86 基本指令等内容。该项目是本书的重点之一,为后续的学习打下了坚实的基础。

项目三主要讲述汇编语言的表达式、伪指令、DOS 功能调用以及上机调试过程等基本知识。

项目四主要介绍了汇编语言程序设计的概述、顺序程序设计、分支程序、循环和串操作等内容。该项目也是本书的重点之一,为后续的学习打下了坚实的基础。

项目五主要介绍了子程序设计技术和宏操作的相关知识及编程技巧。

项目六主要介绍了输入/输出概念、中断及中断程序设计、BOIS 功能调用、鼠标、键盘、显示器等基本的知识点。

项目七主要介绍了磁盘文件、传统磁盘文件管理方式、扩展磁盘文件管理方式、磁盘文件管理及应用等文件存取技术的相关知识点。

项目八主要介绍了汇编语言扩展技术，包括：条件汇编技术、重复汇编技术、多模块技术以及 C 与汇编混合编程技术。

本书在编排上注重理论与实践相结合，采用任务式教学模式，突出实践环节，充分体现“工学结合一体化”教学思想。本书将项目分解为若干任务，每个任务由任务描述、任务分析、知识准备、任务实施四部分组成，全书共设置任务 36 个。正文中设置了小提示、拓展提高以及知识链接等特色模块，意在提高学生的学习兴趣，促进学生的全面发展。全书共设置小提示 32 个，知识链接 11 个，拓展提高 10 个。每个项目最后设置了项目小结和项目考核内容。

本书由安阳工学院计算机科学与信息工程学院的常国权、赵凯、张捐净、郭丽敏、孔娟编写。具体章节编写为：项目四、项目八由常国权编写，项目一、项目五、项目六、项目七由赵凯和张捐净编写，项目二、项目三由郭丽敏和孔娟编写。

本书在编写过程中参考了大量的参考文献和书籍。虽然编者在编写本书的过程中付出了很大的努力，但由于学识水平和时间的限制，书中难免有不当之处，敬请读者不吝指正。

编者

CONTENTS

目 录

项目一 汇编语言基础知识

任务一：认识汇编语言	2
任务描述	2
任务分析	2
知识准备	2
一、汇编语言	2
二、汇编语言的特点	3
三、恰当地使用汇编语言	4
任务实施	5
任务二：数值的表示	6
任务描述	6
任务分析	6
知识准备	6
一、原码，反码，补码	6
二、字符编码	8
任务实施	9
任务三：80x86 处理器的存储器与堆栈	9
任务描述	9
任务分析	9
知识准备	9
一、从 8086 到 80286	9
二、80386 和 80486	11
三、Pentium 和 Pentium Pro	13
四、80x86 微处理器的编程结构	14
五、计算机存储概述	17
六、实模式下存储器的组织	18
七、保护模式下存储器的组织	21
八、堆栈的结构与数据存取	23

任务实施	24
------------	----

项目小结	24
项目考核	24

项目二 80x86的指令系统

任务一：80x86 指令系统概述	27
任务描述	27
任务分析	27
知识准备	27
任务实施	29
一、了解 80x86 的指令种类	29
二、了解 80x86 的指令格式	30
任务二：80x86 寻址方式	30
任务描述	30
任务分析	30
知识准备	30
任务实施	41
任务三：了解 8086 基本指令	43
任务描述	43
任务分析	43
知识准备	43
一、数据传送指令	43
二、算术运算指令	52
三、逻辑运算指令	60
四、程序控制类指令	68
五、输入 / 输出类指令	69
六、处理器控制类指令	69
任务实施	69

项目小结	71	知识准备	102
项目考核	71	任务实施	104
项目三 汇编语言程序结构			
任务一：了解汇编语言程序设计要素	75	任务二：设计分支程序——转移指令	106
任务描述	75	任务描述	106
任务分析	75	任务分析	106
知识准备	75	知识准备	106
一、汇编语句	75	任务实施	113
二、常量	76	任务三：设计分支程序——双分支结构	113
三、数值表达式	78	任务描述	113
四、地址表达式	79	任务分析	113
五、变量和标号的定义	79	知识准备	114
六、变量和标号的属性	81	任务实施	114
七、变量定义伪指令	83	任务四：设计分支程序——多分支结构	116
八、段定义伪指令	83	任务描述	116
九、段使用设定伪指令 ASSUME	84	任务分析	116
十、ORG 语句	85	知识准备	116
十一、源程序开始和结束伪指令	86	任务实施	117
任务实施	87	任务五：设计循环程序——单循环结构	119
任务二：调用 DOS 功能	87	任务描述	119
任务描述	87	任务分析	119
任务分析	88	知识准备	119
知识准备	88	任务实施	126
任务实施	90	任务六：设计循环程序	
任务三：创建汇编语言开发环境	91	——多重循环结构	128
任务描述	91	任务描述	128
任务分析	91	任务分析	128
知识准备	92	知识准备	128
任务实施	93	任务实施	133
项目小结	98	任务七：设计程序——串操作	134
项目考核	99	任务描述	134
		任务分析	134
		知识准备	135
		任务实施	139
项目四 汇编语言程序设计基本方法			
任务一：设计顺序结构程序	102	任务八：设计数据运算程序	
任务描述	102	——BCD 数加法	140
任务分析	102	任务描述	140

任务分析	140
知识准备	140
任务实施	142

任务九：设计数据运算程序

——BCD 数减法	143
任务描述	143
任务分析	143
知识准备	143
任务实施	144

任务十：设计数据运算程序

——BCD 数乘法	145
任务描述	145
任务分析	145
知识准备	146
任务实施	146

任务十一：设计数据运算程序

——BCD 数除法	147
任务描述	147
任务分析	147
知识准备	147
任务实施	148

项目小结	149
------------	-----

项目考核	152
------------	-----

项目五 子程序和宏

任务一：设计子程序	156
任务描述	156
任务分析	156
知识准备	156
一、子程序的定义	156
二、子程序的调用和返回	157
三、现场的保护与恢复	160
四、主程序与子程序之间的参数传递	161
任务实施	164

任务二：使用宏	165
任务描述	165
任务分析	165

知识准备	165
任务实施	175

项目小结	178
------------	-----

项目考核	178
------------	-----

项目六 输入/输出设计

任务一：输入 / 输出基本概念	181
任务描述	181
任务分析	181
知识准备	181
一、输入 / 输出端口地址	181
二、基本输入 / 输出指令	182
三、串输入 / 输出指令	182
四、数据传送方式	184
任务实施	186
一、无条件传送方式举例	186
二、查询方式举例	187

任务二：中断及中断程序设计	189
----------------------------	-----

任务描述	189
任务分析	189
知识准备	189
一、中断和中断传送方式	189
二、中断向量表	190
三、中断响应过程	193
四、外部中断	194
五、内部中断	197
六、中断优先级和中断嵌套	198
任务实施	200

任务三：BIOS 功能调用——键盘	204
--------------------------------	-----

任务描述	204
任务分析	204
知识准备	205
一、键盘数据区	205
二、键盘输入的 INT 16H 操作	207
任务实施	208

任务四：BIOS 功能调用——显示器	210
---------------------------------	-----

任务描述	210
任务分析	210

知识准备	211
一、显示模式简介	211
二、文本显示模式	212
任务实施	214
一、直接写屏方法	214
二、彩色文本模式的 BIOS 中断调用	215
项目小结	217
项目考核	217

项目七 文件存取技术

任务一：编写顺序存取方式程序	220
任务描述	220
任务分析	220
知识准备	220
一、磁盘文件概念	220
二、顺序存取方式	222
任务实施	224
任务二：编写随机存取方式程序	226
任务描述	226
任务分析	226
知识准备	226
任务实施	228
任务三：编写随机分块存取方式程序	230
任务描述	230
任务分析	230
知识准备	231
任务实施	231
任务四：编写文件代号存取方式程序	233
任务描述	233
任务分析	233
知识准备	234
任务实施	237
任务五：编写移动读写指针程序	239
任务描述	239
任务分析	239
知识准备	239

任务实施	241
任务六：操作外部文件	243
任务描述	243
任务分析	243
知识准备	243
任务实施	244
项目小结	246
项目考核	246

项目八 汇编语言扩展

任务一：重复汇编	249
任务描述	249
任务分析	249
知识准备	249
任务实施	251
任务二：条件汇编	252
任务描述	252
任务分析	252
知识准备	252
任务实施	254
任务三：多模块程序设计	254
任务描述	254
任务分析	254
知识准备	255
一、多模块技术	255
二、多模块的参数设置	255
任务实施	256
任务四：C 与汇编混合编程	259
任务描述	259
任务分析	259
知识准备	259
任务实施	261
项目小结	263
项目考核	263
参考文献	264

项目一

汇编语言基础知识

项目要点


- 认识汇编语言
- 数值的表示
- 80x86 处理器
- 存储器与堆栈
- 外部设备

引言


本项目讲解的是汇编语言的基础知识。通过介绍机器和汇编语言、计算机中数值的表示、Intel 80x86 处理器家族、存储器与堆栈、外部设备的访问，由软件向硬件、由内向外分层次介绍了汇编语言的基本知识。

1

任务一：认识汇编语言

 任务描述

小张是某学校计算机专业的学生，目前正在学习汇编语言程序的相关知识。但什么是机器语言，机器语言与汇编语言有什么关系呢？在高级语言流行的今天，什么场合适合使用汇编语言编程？

 任务分析

首先了解汇编语言的来历，接着明确汇编语言的优缺点，最后就可以找到哪些场合下使用汇编语言了。

 知识准备

一、汇编语言

1. 机器语言

机器语言是用二进制代码表示的计算机能直接识别和执行的一种机器指令系统的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有灵活、直接执行和速度快等特点。一条指令就是机器语言的一个语句，它是一组有意义的二进制代码，指令的基本格式包括操作码字段和地址码字段。其中，操作码指明了指令的操作性质及功能，地址码则给出了操作数或操作数的地址。用机器语言编写程序，编程人员首先要熟记所用计算机的全部指令代码和代码的含义。编写程序时，程序员不仅需要自己处理每条指令和每一数据的存储分配和输入/输出，还需要记住编程过程中每步所使用的工作单元处在何种状态。这是一件十分烦琐的工作。编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且，编出的程序全是0和1的指令代码。

例如，用 Intel 8086 指令写一个两数相加的程序片段。具体要求是把偏移 2200H 存储单元中的数与偏移 2201H 存储单元中的数相加，将它们的和送入偏移 2202H 存储单元。完成这一工作的程序片段包含三条机器指令，用十六进制形式表示如下：

```
A0 00 20  
    02 06 01 20  
A2 02 20
```

几乎没有人能直接看出该程序片段的功能，原因是程序员难以掌握机器语言。正因为程序员难以用机器语言写程序，更难以写出健壮的程序；加之用机器语言编制出

的程序也不易为人们理解、记忆和交流。所以，只是在早期或不得已时才用机器语言写程序，现在几乎没有人用机器语言编写程序了。

2. 汇编语言

为了克服机器语言的上述缺点，人们采用便于记忆、并能描述指令功能的符号来表示指令的操作码。这些符号被称为指令助记符。助记符一般是说明指令功能的英语词汇或者词汇的缩写。同时，也用符号表示操作数，如 CPU 的寄存器、存储单元地址等。用指令助记符、地址符号等符号表示的指令称为汇编格式指令。

汇编语言是汇编格式指令、伪指令的集合及其表示、使用这些指令的一组规则。伪指令的概念留待以后介绍。用汇编语言书写的程序称为汇编语言程序，或称为汇编语言源程序，或简称为源程序。

3. 汇编程序

由于 CPU 能直接识别的唯一语言是机器语言，所以用汇编语言编写的源程序必须被翻译成用机器语言表示的目标程序后才能由 CPU 执行。把汇编语言源程序翻译成目标程序的过程称为汇编。完成汇编任务的程序叫作汇编程序。汇编过程如图 1-1 所示。

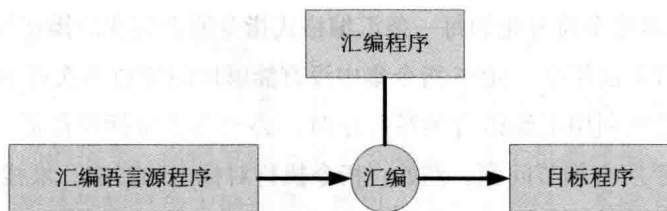


图 1-1 汇编过程示意图

二、汇编语言的特点

由于汇编语言使用指令助记符和符号地址，所以它要比机器语言容易掌握得多。与高级语言相比较，汇编语言有如下特点。

1. 汇编语言与机器关系密切

因为汇编格式指令是机器指令的符号表示，所以汇编格式指令与机器有着密切的关系，汇编语言也与机器有着密切的关系，确切地说汇编语言与机器的 CPU 有着十分密切的关系。对于各种不同类型的 CPU，要使用各种不同的汇编语言。于是，对于各种不同类型的 CPU，也就有各种不同的汇编程序。

由于汇编语言与机器关系十分密切，汇编语言源程序与高级语言源程序相比，通用性和可移植性要差得多。但通过汇编语言可最直接和最有效地控制机器，这常常是大多数高级语言难以做到的。

chapter
01chapter
02chapter
03chapter
04chapter
05chapter
06chapter
07chapter
08

2. 汇编语言程序效率高

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和空间两个方面：其一是运行速度快；其二是目标程序短。在采用相同算法的前提下，任何高级语言程序在这两方面的效率都不如汇编语言程序，许多情况下更是远远不及。



知识链接

汇编语言程序能获得“时空”高效率的主要原因有两个：

(1) 构成汇编语言主体的汇编格式指令是机器指令的符号表示，每一条汇编格式指令都是所对应的某条机器指令的“化身”。

(2) 汇编语言程序能直接并充分利用机器硬件系统的许多特性。高级语言程序在上述两点上要逊色得多。

3. 编写汇编语言源程序烦琐

编写汇编语言源程序要比编写高级语言源程序烦琐得多。汇编语言是面向机器的语言，高级语言是面向过程或面向目标、对象的语言。如下两点突出表现了汇编语言的这一特性：

(1) 作为机器指令符号化的每一条汇编格式指令所能完成的操作极为有限。例如，Z80 指令集中没有乘法指令，8086 指令集中没有能够同时完成两次算术运算的指令。

(2) 程序员在利用汇编语言编写程序时，必须考虑包括寄存器、存储单元和寻址方式在内的几乎所有细节问题。例如：指令执行对标志的影响，堆栈设置的位置等。在使用高级语言编写程序时，程序员不会遇到这些琐碎却重要的问题。

4. 汇编语言程序调试困难

调试汇编语言程序往往要比调试高级语言程序困难。汇编格式指令的功能有限，且程序员要注意太多的细节问题是造成这种困难的两个客观原因；汇编语言提供给了程序员最大的“舞台”，而程序员往往为了追求“时空”上的高效率而不顾程序的结构，这是造成调试困难的主观原因。

三、恰当地使用汇编语言

1. 汇编语言的优缺点

为了恰当地使用汇编语言，先来明确一下它的优缺点。

汇编语言的主要优点是利用它可以编写出在“时空”两个方面最有效率的程序。另外，通过它可最直接和最有效地操纵机器硬件系统。

汇编语言的主要缺点是它面向机器，与机器关系密切，它要求程序员比较熟悉机器硬件系统，要考虑许多细节问题，最终导致程序员编写程序烦琐；调试程序困难；

维护、交流和移植程序更困难。

正是由于汇编语言与机器关系密切，才使汇编语言具有其他高级语言所不具备的上述优点和缺点。为了利用汇编语言的优点，必须付出相应的代价。但汇编语言的每一个优点常常闪耀出诱人的光芒，使人们勇敢地面对它的缺点。

2. 使用汇编语言的场合

根据汇编语言的优缺点，要恰当地使用汇编语言，即尽可能地“扬长避短”。是否利用汇编语言编写程序，要看具体的应用场合，要充分考虑软件的开发时间和软件的质量等多方面的因素。下列应用场合可考虑使用汇编语言编写程序。

(1) 对软件的执行时间或存储容量有较高要求的场合。例如：系统程序的关键核心，智能化仪器仪表的控制系统，实时控制系统等。

(2) 需要提高大型软件性能的场合。通常把大型软件中执行频率高的子程序（过程）用汇编语言编写，然后把它们与其他程序连接。

(3) 软件与硬件关系密切，软件要有直接和有效控制硬件的场合。如设备驱动程序等。

(4) 没有合适的高级语言的场合。

3. 适度地追求“时空”效率

在用汇编语言编写程序时，追求“时空”效率要适度。在编写汇编语言程序时，要尽量利用最恰当的指令，以便节约一个字节或节省几个机器周期。但时至今日，计算机硬件系统的整体性能已极大地提高。所以，除非不得已，不要为节约少量字节或机器周期而影响程序的结构性、健壮性和可读性等，要在确保汇编语言程序上述性能良好的前提下追求“时空”性能。

任务实施

用汇编语言程序实现 x^2+y^2 运算，其中 x 、 y 已分别存放在寄存器 A、B 中，计算结果存入寄存器 C 中。

MUL A, A; 将寄存器 A 的内容与寄存器 A 的内容相乘，结果送寄存器 A（计算 x^2 ）

MUL B, B; 将寄存器 B 的内容与寄存器 B 的内容相乘，结果送寄存器 B（计算 y^2 ）

ADD A, B; 将寄存器 A 的内容与寄存器 B 的内容相加，结果送寄存器 A（计算 x^2+y^2 ）

MOV C, A; 将寄存器 A 的内容传送到寄存器 C

4 条汇编指令 MUL A, A/MUL B, B/ADD A, B/MOV C 中，A 分别代替了 4 条机器指令 00110000/00110101/00010001/00001000；MUL、ADD 和 MOV 分别表示“乘”（multiply）、“加”（adding）和“传送”（move）。显然，这种使用助记符表示的汇编指令远比单调的二进制序列好记，且可读性好。

2

任务二：数值的表示



任务描述

小李是某计算机公司的员工，目前接到公司通知，需要为新员工讲解计算机中数值的表示方式。



任务分析

计算机中存储信息的最小单位称为位，在绝大多数系统中它只能表示两种状态。这两种状态可分别代表 0 和 1。计算机系统内部采用二进制表示数值数据，也采用二进制编码表示非数值数据和指令，其主要原因就在于此。



知识准备

一、原码，反码，补码

数值数据是一种带符号数，即有正负之分。在计算机中，数的符号（+ 或 -）和数的值一样都要采用二进制 0、1 码。在计算机中通常采用补码编码来表示数值数据，而数值数据又分为定点数和浮点数两种形式。这里仅对定点数作出讨论。

1. 原码表示法

原码表示法是机器数的一种简单的表示法。用 0 表示正号，用 1 表示负号，数值一般用二进制形式表示。

2. 反码表示法

机器数的反码可由原码得到。如果机器数是正数，则该机器数的反码与原码一样；如果机器数是负数，则该机器数的反码是对它的原码（符号位除外）各位取反而得到的。反码通常作为求补过程的中间形式，即在一个负数的反码的末位上加 1，就得到了该负数的补码。

3. 补码表示法

机器数的补码可由原码得到。如果机器数是正数，则该机器数的补码与原码一样；如果机器数是负数，则该机器数的补码是对它的原码（除符号位外）各位取反，并在末位加 1 而得到的。

补码有两条重要的性质：

(1) 补码的零是唯一的，即

$$[+0]_{\text{补}} = [-0]_{\text{补}} = 0.000 \cdots 000$$

等于 2 的 4 次幂，因此二进制数与十六进制数之间能方便地转换，即 4 位二进制数对应 1 位十六进制数，或者 1 位十六进制数对应 4 位二进制数。因此，常把二进制数改写成十六进制数，在汇编语言程序设计过程中尤其如此。

在书写时，为了区别十进制数和二进制数，通常在十六进制数后加一个字母 H。

二、字符编码

计算机不仅要求能处理数值数据，还要求能处理非数值数据。非数值数据最常见的就是字符信息，如英文字母、标点符号、十进制数位以及诸如 \$、%、+ 等符号和一些控制信息。在计算机中对于这一类数据的表示，同样采用的是二进制编码的方法。国际上出现了一些不同的编码方法，并在不同的领域或场合得以应用，下面列举几种常见的编码方法。

1. EBCDIC 码

IBM 公司在开发 IBM System 360 机器之前，使用了一种 6 位的扩展 BCD 码来表示字符和数字。但是，这种编码方式所表示的字符很有限，例如，小写字符就无法表示。因此，IBM System 360 系统的设计者进一步将 BCD 码扩展为 8 位，使用了一种扩展的二 - 十进制编码的交换码（Extended Binary Coded Decimal Interchange Code，EBCDIC 码）。

2. ASCII 码

目前，国际上普遍采用的字符系统是 7 位的美国国家信息交换标准字符码（American Standard Code for Information Interchange，ASCII 码）。ASCII 码是从使用了几十年的电传打字设备的编码方案中直接衍生出来的，这些电传设备使用 19 世纪 80 年代发明的 5 位摩尔编码。到了 20 世纪 60 年代，这种 5 位编码方式的局限性已经变得非常明显，国际标准化组织 ISO 就设计了一种 7 位的编码方案，并于 1967 正式公布，就是现在的 ASCII 码。

3. 统一字符编码

无论是 EBCDIC 码还是 ASCII 码，都是建立在拉丁语系字母的基础上的。因此，这些编码方式在对非拉丁字母提供数据表示的能力方面受到了很大限制，而全球大多数人都使用非拉丁语系。随着全世界所有的国家都开始使用计算机，每个国家都在设计一套能够有效地表示自己本国语言的编码。然而，这些编码都不能与其他国家的编码相互兼容，这样也就在融入全球经济的道路上设置了另一种障碍。

为了防止这种情况继续恶化，1991 年成立了一个由工业和社会领导人组成的协会，并建立了一种新的国际信息交换代码，称为统一字符编码（Unicode）。