普通中等专业教育机电类规划教材

汇编语言及程序设计

重庆机器制造学校　甘　玲　主编

机械工业出版社

普通中等专业教育机电类规划教材

汇编语言及程序设计

主编 甘 玲

参编 (以姓氏笔画为序)

张 运

李革新

崔玉玲

主审 黄清虎

机械工业出版社

本书是普通中等专业学校计算机专业汇编语言课程教材，主要介绍IBM－PC及其兼容机汇编语言的基础知识和程序设计方法。全书共分7章，分别介绍了微型计算机系统的组成、8086/8088微处理器的结构和指令系统、汇编语言程序设计方法和技巧、汇编语言的扩展应用。书中提供了大量的例题，每章后都附有习题，附录中还给出了实验指导书。

本书语言精练、通俗易懂，叙述由浅入深、循序渐进，思路清晰、结构严谨。本书除作为中等专业学校计算机专业教材外，也可供大专院校、高等职业技术学院计算机专业的学生使用，同时也可作为工程技术人员自学的参考书。

图书在版编目(CIP)数据

汇编语言及程序设计/甘玲主编. —北京：机械工业出版社，1999.10（2000.4重印）
普通中等专业教育机电类规划教材
ISBN 7－111－07137－9

Ⅰ.汇… Ⅱ.甘… Ⅲ.汇编语言-程序设计-专业学校-教材 Ⅳ.TP313

中国版本图书馆CIP数据核字(2000)第04046号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
责任编辑：王小东 版式设计：霍永明 责任校对：魏俊云
封面设计：姚 毅 责任印制：路 琳
中国建筑工业出版社密云印刷厂印刷·新华书店北京发行所发行
2000年3月第1版第2次印刷
787mm×1092mm 1/16·15印张·365千字
5001－9000册
定价：20.00元
凡购本书，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、68326677-2527

前　言

本书是根据原机械工业部教育司于1996年修订的《机械工业部中等专业学校教学计划与教学大纲》的精神，在机械工业部中专计算机专业教学指导委员会的具体指导下，组织长期从事汇编语言教学的部分教师编写的。

汇编语言是计算机专业的技术基础课，是操作系统等其它课程的必要先修课。本书以8086/8088 CPU为对象介绍汇编语言的基础知识和程序设计方法，这些内容在学习其它类型CPU的汇编语言时也是完全适用的。

为了适应计算机的发展和应用，让更多的人了解计算机的工作原理，本书从应用的角度介绍微型计算机的结构、8086/8088指令系统和汇编语言程序设计方法。借助汇编语言可以充分调动微型计算机的所有硬件特性并能直接控制硬件，从而让读者进一步弄清程序在计算机中的运行机制，能够在更高层次上对计算机及计算机语言有新的认识。

本书共分7章，第一章主要介绍微型计算机系统的组成、微处理器的发展、汇编语言的特点和计算机的运算基础；第二章主要介绍8086/8088微处理器系统结构；第三章主要介绍8086/8088寻址方式和指令系统；第四章主要介绍汇编语言的语句格式、程序结构、伪指令和汇编语言程序的上机过程；第五章主要介绍程序设计的基本步骤，介绍顺序、分支、循环、子程序的程序设计方法和技巧；第六章主要介绍宏汇编技术、多模块程序设计；第七章主要介绍汇编语言在中断调用、DOS功能调用和I/O程序设计等方面的应用，同时还介绍汇编语言与C语言的接口等问题。

本书除作为中等专业学校计算机专业汇编语言课程教材外，也可供大专院校、高等职业技术学院计算机专业的学生使用，同时可作为工程技术人员自学的参考书。

本书由重庆机器制造学校甘玲编写第一章第三节、第二章、第三章前两节、第四章前三节、第六章第三节部分内容、第七章第一节，上海机电工业学校张运编写第一章前两节、第六章、第七章后三节，中原机械工业学校崔玉玲编写第三章第二节、第四章第三节，咸阳机器制造学校李革新编写第五章，由甘玲担任主编，厦门市工业学校黄清虎担任主审。参加审稿讨论的还有机械工业计算机专业教学指导委员会委员：江西省机械工业学校刘学军、广东省机械工业学校韩穗、福建省高级工业学校余力，湖南省机械工业学校刘翌南等，他们提出了许多宝贵的修改意见。在编写过程中还得到了中国人民解放军后勤工程学院李生林副教授、重庆机器制造学校唐健高级讲师的热情指导，在此一并表示衷心的感谢。

由于编者水平有限，编写时间仓促，书中错漏在所难免，恳请读者批评指正。

编者于重庆

1999年3月8日

目　录

第一章　微型计算机基础知识

计算机的产生、发展和应用，带动着人类在科学道路上迅猛前进。自 1946 年第一台电子计算机问世后，其发展速度相当惊人，随着主要电子元件从电子管、晶体管、中小规模集成电路发展到大规模集成电路，70 年代初研制出了第一台微型计算机。众所周知，最初的电子计算机只是作为一种现代化的计算工具，而微型计算机的出现为计算机的广泛应用开拓了极其广阔的前景，展示了它在信息社会中日益显要的地位，特别是微型计算机的科技水平、生产规模和应用程度已成为衡量一个国家现代化水平的重要标志。微型计算机的发展和应用已引起全社会的关注，并已转化成巨大的推动社会前进的生产力。

微处理器是微型计算机的“心”，指挥这颗“心”跳动的是它的指令系统，是人们赋予它的程序，汇编语言是与微处理器最密切的程序设计语言。本章将介绍微型计算机系统的组成、汇编语言的特点和计算机的运算基础。

第一节　微型计算机系统概述

计算机系统由硬件系统（Hardware）和软件系统（Software）两部分组成。微型计算机系统也是由硬件系统和软件系统组成，只不过它的硬件大部分集中放在主机箱中。

一、微型计算机系统的组成

（一）硬件系统

计算机硬件系统由运算器、控制器、存储器、输入/输出设备五大部分组成，其中运算器和控制器构成中央处理器 CPU（Central Processor Unit），中央处理器和存储器（Memory）构成主机。

微型计算机硬件系统由主机和外围设备构成，主机由微处理器（Microprocessor）、存储器和输入/输出（I/O）接口三个部分组成，各部分由系统总线把它们连接在一起，主机的结构如图 1－1 所示。

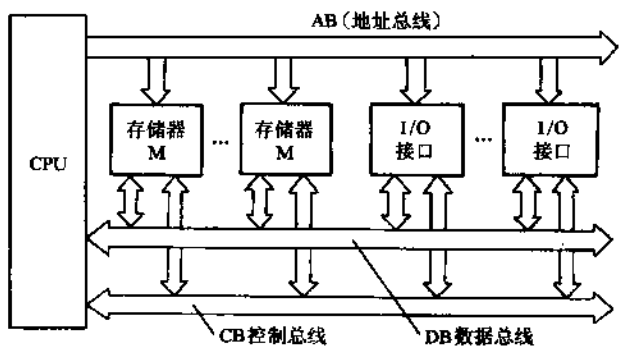

图 1－1　主机的基本组成

1. 主机

微型计算机硬件系统的核心是主机，主机的核心部件是微处理器。

（1）微处理器　微处理器微缩在一片或几片大规模集成电路芯片上，它的任务是处理存放在存储器中的程序指令：从存储器中取出指令，进行译码，执行算术运算和逻辑运算，再存放数据，并控制整个微型计算机自动、协调地完成程序功能。

（2）存储器　存储器是微型计算机的记忆部件，类似于人的大脑。主机中的存储器称为内存储器，简称内存。内存由一个个存储单元组成，每个存储单元都有一个地址，人们编写

的程序及程序中所用到的数据、编码信息及中间结果在程序运行过程中一般放在内存中。内存分为只读存储器 ROM（Read－Only Memory）和随机读写存储器 RAM（Random Access Memory）两类。ROM 用于存放起动、自检、设备驱动程序等永久性信息；RAM 中存放的信息读出时不变，写入时新内容将取代原来的内容，断电后其中所有的内容都将丢失。

（3）输入/输出接口　输入/输出接口是主机与外围设备之间通信的桥梁，由寄存器和译码器组成。微处理器有自己的内部特性，用户不能更改；输入/输出设备也有自己的特性，其工作速度比微处理器低得多，因此，要使两者结合起来协调工作，就需要适当的接口。输入/输出接口的任务是处理主机与外围设备之间的数据传送，把外围设备的状态传入主机，接收主机发出的各种控制信号，控制外围设备执行操作。

2. 外围设备

外围设备一般包括输入/输出设备和大容量存储器两类设备。输入/输出设备是指负责计算机进行通信的外围设备，如键盘、显示器、打印机等。大容量存储器则是指可存储大量信息的外存储器，如磁盘、磁带、光盘等，简称外存。由于内存的容量有限，所以计算机使用外存作为内存的后援设备，外存的容量一般比内存大得多，但从外存中存取信息的速度比从内存中存取信息的速度慢得多。所以除了必要的系统程序外，一般的程序是存放在外存中，要运行该程序时，才把它从外存传送到内存的某个区域，然后由微处理器控制执行。

3. 系统总线

系统总线把微处理器、存储器、输入/输出接口和外围设备连接起来，用来传送它们之间的信息。系统总线包括数据总线、地址总线和控制总线。数据总线负责传送数据，数据包括指令代码、原始数据、中间数据和结果数据；地址总线用来指出数据的来源地和目的地；控制总线则负责控制总线的动作，传送微处理器对存储器或输入/输出设备的控制命令和输入/输出设备对微处理器 CPU 的请求信号。系统总线的工作由总线控制逻辑负责指挥。

由于构成微型计算机硬件系统的各个部件，如：机箱、主机板、电源、软盘驱动器、硬盘驱动器以及显示卡、软盘/硬盘卡和串行、并行输入/输出卡等都被设计成了模块结构，它们的机械安装尺寸约定为统一规格，并采用不易接错的插、接口方式连接。因此，完全可以用世界各地的不同厂家生产的微型计算机部件组装成一台合格的兼容微型计算机。微型计算机发展过程中的这一开放性举措是相当重要的，它打破了个别生产厂家的垄断，推动了大规模生产技术的发展，使硬件系统的价格逐步降低，为微型计算机步入家庭成为真正意义上的“个人计算机”奠定了坚实的基础。

（二）软件系统

微型计算机的软件系统是由系统软件和应用软件两大部分组成。系统软件是厂家提供给用户、帮助用户编写和调试应用程序的程序集合，它包括如下内容：操作系统、监控程序、输入/输出驱动程序、文本编辑程序、编译和解释程序、调试程序、连接定位程序、数据库管理系统等；应用软件则是为解决某一个实际问题而编写的程序集合，如：科学计算程序、数据处理程序、企业管理程序、电算化软件等。

操作系统（Operating System）是系统软件的指挥中枢，它的主要作用是统一管理微型计算机的所有资源，包括微处理器、存储器、输入/输出设备以及其它的系统软件和应用软件。用户在使用微型计算机时，无需过问系统中各个资源的分配和使用情况，也不必为各种输入/输出设备编制设备驱动程序，用户只需要正确使用操作系统提供的各种命令和系统调用功

能，就可以使应用程序在操作系统的控制下自动而协调地运行。目前微型计算机常用的操作系统有：MS－DOS、Windows 3.X、Windows 95/98、Windows NT、Netware、Macintosh、OS/2 Wrap、Unix、Linux 等。

操作系统的主要部分是常驻内存的监控程序。只要一开机，监控程序就驻留内存，通过键盘接受用户的命令，从而控制操作系统执行相应的操作。

输入/输出驱动程序是用来对输入/输出设备进行管理和控制。当系统程序或应用程序需要使用输入/输出设备时，通过调用输入/输出驱动程序对相应的设备发出命令，从而完成微处理器和输入/输出设备之间的信息传送。

文本编辑程序是用来输入和编辑文本并将其存入存储器中。文本是指由数字、字母、符号等信息所组成的文件，它可以是一个用汇编语言或高级语言编写的程序，也可以是一组数据或一份报告。例如 MS－DOS 中的 EDIT 就是一个能方便地输入和编辑文本的全屏幕文本编辑程序。

编译和解释程序是把人们编写的程序翻译成微型计算机能识别的二进制代码的一种系统程序。编译程序（Compiler）是先把高级语言程序翻译成机器语言程序，然后再执行；而解释程序（Interpreter）则是一边翻译一边执行。

调试程序是系统提供给用户的能监控用户程序的一种工具，例如 MS－DOS 中的 DEBUG，它可以装入、修改、显示和逐条执行一个程序。通常，简单的汇编语言程序可以通过 DEBUG 来建立、修改和执行。

连接定位程序（LINK）是用来把要执行的程序与库文件或其它已经翻译的子程序连接起来，形成微型计算机能够运行的执行文件。

概括起来，微型计算机的硬件系统就像是人的躯壳，而软件就像是人的灵魂，两者相互依存、缺一不可。微型计算机系统的组成如图 1－2 所示。

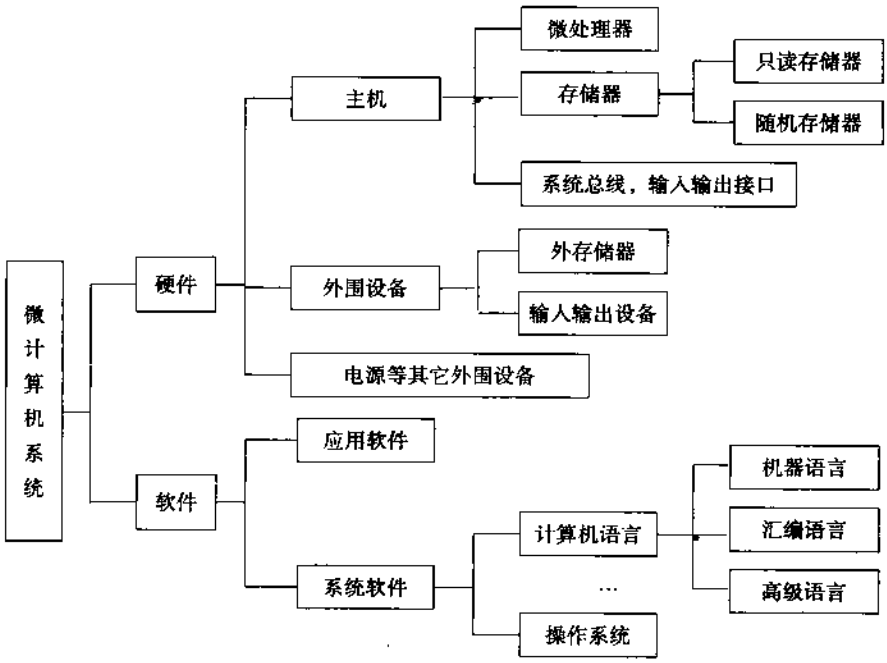

图 1－2　微型计算机系统的组成

二、微处理器的发展进程

微处理器的历史可追溯到1971年，在这短短的二十几年里，微处理器的发展日新月异，产品像潮水般涌向市场，推动着全球经济的大发展，下面以Intel公司的80x86系统为例介绍微处理器的发展历史。

1971年，美国旧金山南部的Intel集成电子产品公司推出了世界上第一代微处理器4004，它是4位微处理器，每个芯片上含有2300个晶体管，支持45条指令。从此以后，Intel便与微处理器结下了不解之缘。

1973年，Intel公司推出了第二代微处理器8080，它是8位微处理器，每个芯片上含有4900个晶体管，运算速度提高了一个数量级，指令系统比较完善，寻址能力有所增强。

以后微处理器的发展进程如表1－1所示。

表1－1　微处理器发展历史简表

CPU	推出年份	主频/MHz	CPU插槽类型	内部cache/KB	外部cache	集成晶体管数	总线宽度
8086	1978年6月	4.77	焊接在主板上	无	无	2.9万	16位
8088	1979年6月	4.77	焊接在主板上	无	无	2.9万	8位
80286	1982年2月	6～20	焊接在主板上	无	无	13.4万	16位
80386	1985年10月	12.5～33	焊接在主板上	8	有	27.5万	32位
80486	1989年4月	25～100	Socket3（238插孔）	8	有	120万	32位
Pentium（P54C）	1993年3月	60～66	Socket4（320插孔）	16	有	310万	32位
	1993年6月	75～200	Socket5（320插孔）	16	有	330万	32位
Pentium Pro	1995年11月	133～200	Socket8（387插孔）	16	CPU内集成256～512KB	550万	32位
Pentium MMX（P55C）	1997年1月	166～233	Socket7（321插孔）	32	有	450万	32位
Pentium Ⅱ	1997年5月	233～450	Slot1插槽	32	与CPU、Tag RAM一起集成在板上	750万	64位

第二节　汇编语言简介

一、计算机语言的分类

计算机语言种类很多，按语言层次可分为：机器语言、汇编语言和高级语言。下面举例说明分别用三种语言编制的程序。

例1－1　已知a＝2，b＝6，c＝3，求Y＝a＋b－c的值并显示结果。

（一）机器语言

机器语言是计算机能直接识别的程序设计语言，由二进制代码组成，因此这种语言的可读性很差，使用这种语言编写程序很不方便。

实现例 1－1 的机器语言程序如下：

```
B8 02 00
05 06 00
2D 03 00
8B D0
80 CA 30
B4 02
CD 21
B4 4C
CD 21
```

（二）汇编语言

汇编语言是一种面向机器的程序设计语言，是机器语言符号化的描述。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。相对机器语言来说，汇编语言一般用与操作含义相同的英语单词或单词缩写来作为指令操作符，因而可读性较好，但是编写比较复杂的程序还是很困难。

实现例 1－1 的汇编语言程序如下：

```
SSEG    SEGMENT STACK "STACK"
        DB 64 DUP(?)
SSEG    ENDS
CSEG    SEGMENT "CODE"
ASSUME  CS: CSEG, SS: SSEG
START:  MOV   AX, 2
        ADD   AX, 6
        SUB   AX, 3
        MOV   DX, AX
        OR    DL, 30H
        MOV   AH, 2
        INT   21H
        MOV   AH, 4CH
        INT   21H
CSEG    ENDS
        END START
```

（三）高级语言

高级语言是最接近于人类自然语言的程序设计语言。由于高级语言脱离了机器指令，因而可读性好，但同时也造成高级语言对硬件的控制能力很弱，此外生成执行代码效率较低。当前广泛使用的高级语言有：C、FoxPro、Pascal、Fortran、Cobol、Basic 等，下面以 C 语言为例说明。

实现例 1－1 的 C 语言程序如下：

```
#include <stdio.h>
int main ()
 {
    int a, b, c, y;
    a=2;
    b=6;
    c=3;
    y=a+b-c;
    printf ("%d \n", y);
    return 0;
}
```

二、汇编语言的特点

微型计算机的迅速发展，特别是它在控制、网络与通信等方面的广泛应用，使汇编语言程序设计技术不仅为系统程序员使用，而且为广大微型计算机用户所普遍使用。

汇编语言是计算机提供给用户的最快而又最有效的语言，它能透彻地反映、巧妙充分地运用计算机的硬件功能及特点，便于编程人员根据自己的需要灵活地编制各种程序，控制计算机的运行。在对运行速度要求很高的场合，汇编语言是必不可少的，如操作系统、编译程序等，多数是用汇编语言编写的。

汇编语言是机器语言的符号化表示，因此，汇编语言能产生最快而最有效的代码。通常情况下，用汇编语言编写的程序生成的执行代码长度只有相应用 C 语言编写的程序生成的执行代码长度的 1/10。

通常把用汇编语言编写的程序称为汇编语言源程序，在本书中简称源程序。汇编语言源程序与高级语言程序一样，必须由翻译程序翻译成代码程序，才能被计算机认识，此代码程序称为源程序的目标程序。将源程序翻译成目标程序的翻译程序通常称为汇编程序，将源程序翻译成目标程序的过程通常称为汇编。IBM－PC 机有两个汇编程序：小汇编程序 ASM 和宏汇编程序 MASM，MASM 的功能比 ASM 强，它支持宏汇编，所以建议使用 MASM。

第三节　计算机的运算基础

一、进位计数制及其相互转换

（一）各种进位计数制的特点

按进位的原则进行计数的方法称为进位计数制，简称进位制。日常生活中多用十进制，而在计算机中则采用二进制。由于二进制不易书写和阅读，所以又引入了八进制和十六进制。

1．十进制

十进制的特点是：用 0～9 十个不同的数字符号，且按逢 10 进位的原则进行计数。例如：827．31，小数点左边的第 1 位代表个位，它的值是 $7\times10^0=7$；左边第 2 位代表十位，它的值是 $2\times10^1=20$；左边第 3 位代表百位，它的值是 $8\times10^2=800$；小数点右边第 1 位是

十分位，它的值是 $3\times10^{-1}=0.3$；右边第 2 位是百分位，它的值是 $1\times10^{-2}=0.01$。十进制数 827.31 的展开式为：

$$827.31=8\times10^{2}+2\times10^{1}+7\times10^{0}+3\times10^{-1}+1\times10^{-2}$$

其中“10”称为十进制的“基数”，“10^{2}、10^{1}、10^{0}、10^{-1}、10^{-2}”称为十进制数各相应位的“权”。每一位的值等于该位数字与该位权的乘积，各位值的累加和表示整个数的大小。

2. 二进制

二进制的特点是：用 0 和 1 两个不同的数字符号，且按逢 2 进位的原则进行计数。例如 1101.01，与十进制类似有展开式：

$$(1101.01)_2=1\times2^{3}+1\times2^{2}+0\times2^{1}+1\times2^{0}+0\times2^{-1}+1\times2^{-2}=13.25$$

其中“2”称为二进制的“基数”，“2^{3}、2^{2}、2^{1}、2^{0}、2^{-1}、2^{-2}”称为二进制数各相应位的“权”。二进制各位的值也是各位数字与该位权的乘积，各位值的累加和表示整个数的十进制大小。

3. 八进制

八进制的特点是：用 0、1、…、7 八个不同的数字符号，且按逢 8 进位的原则进行计数。“8”是八进制的“基数”，8 的各次幂是八进制各位的“权”。八进制的求值方法类似于二、十进制，例如：

$$(314)_8=3\times8^{2}+1\times8^{1}+4\times8^{0}=204$$

4. 十六进制

十六进制的特点是：用 0、1、…、9、A、B、C、D、E、F 十六个不同的数字符号，且按逢 16 进位的原则进行计数。“16”是十六进制的“基数”，16 的各次幂是十六进制各位的“权”。十六进制的求值方法类似于二、八、十进制。例如：

$$(6AE)_{16}=6\times16^{2}+A\times16^{1}+E\times16^{0}=1536+160+14=1710$$

二、八、十、十六进制数可分别在数字后面加后缀 B、Q（或 O）、D、H 表示。例如：1011B、27Q、16D、2AH 分别表示二进制数 1011、八进制数 27、十进制数 16、十六进制数 2A。十进制数经常省略后缀 D。

（二）不同进位制之间的转换

1. 二进制与十进制之间的互换

（1）二进制转换为十进制　由前述可知，只需将二进制数每一位的数字与该位的权相乘，再将各位的数值加在一起就得到了相应的十进制数。

例 1－2　$1101011.01B=1\times2^{6}+1\times2^{5}+1\times2^{3}+1\times2^{1}+1\times2^{0}+1\times2^{-2}$

$=64+32+8+2+1+0.25$

$=107.25$

（2）十进制转换为二进制

1）十进制整数转换为二进制整数的方法是：用 2 不断地去除要转换的十进制数，直至商为 0。每次的余数即为二进制数位，最初得到的余数是二进制整数的最低位。这就是所谓的“除 2 取余”法。

例 1－3　将十进制数 357 化成二进制数。

$$
\begin{array}{r|l}
2 & 357 \\
2 & 178 \cdots\cdots 1 \\
2 & 89 \cdots\cdots 0 \\
2 & 44 \cdots\cdots 1 \\
2 & 22 \cdots\cdots 0 \\
2 & 11 \cdots\cdots 0 \\
2 & 5 \cdots\cdots 1 \\
2 & 2 \cdots\cdots 1 \\
2 & 1 \cdots\cdots 0 \\
 & 0 \cdots\cdots 1
\end{array}
$$

因此：357 = 101100101B

2）十进制小数转换为二进制小数的方法是：用 2 不断地去乘要转换的十进制小数，直至乘积的小数部分为 0。每次所得的整数部分即为二进制数位，最初得到的整数即是二进制小数的最高位。这就是所谓的“乘 2 取整”法。

例 1－4 将十进制数 0．8125 化成二进制数。

$$
\begin{array}{r}
0.8125 \\
\times \quad 2 \\
\hline
①.6250 \\
\times \quad 2 \\
\hline
①.2500 \\
\times \quad 2 \\
\hline
⓪.5000 \\
\times \quad 2 \\
\hline
①.0
\end{array}
$$

因此：0.8125 = 0.1101B

需要注意的是：当十进制小数不能用有限位二进制小数精确表示时，可根据精度要求，采用“零舍一入”法，取有限位二进制小数近似表示。

例 1－5 将十进制数 0.158 化成二进制数（精确到第七位小数）。

$$
\begin{array}{r}
0.158 \\
\times \quad 2 \\
\hline
⓪.316 \\
\times \quad 2 \\
\hline
⓪.632 \\
\times \quad 2 \\
\hline
①.264 \\
\times \quad 2 \\
\hline
⓪.528 \\
\times \quad 2 \\
\hline
①.056 \\
\times \quad 2 \\
\hline
⓪.112 \\
\times \quad 2 \\
\hline
⓪.224 \\
\times \quad 2 \\
\hline
⓪.448
\end{array}
$$

因此：0.158≈0.00100100B

十进制混合小数转换为二进制数，只要将十进制混合小数的整数部分和纯小数部分按上述方法分别进行转换，然后合起来即可。例：357.8125 = 101100101.1101B。

2. 二进制与八进制之间的互换

(1) 二进制转换为八进制　二进制数转换为八进制数时，以小数点为界，每 3 位二进制数对应 1 位八进制数：整数部分从低位向高位，每 3 位一组，不足 3 位左边补 0；小数部分从高位向低位，每 3 位一组，不足 3 位右边补 0，便可得到相应的八进制数。

例 1－6　将二进制数 1011011．01011 转换成八进制数。

$$\underset{\downarrow}{001}\ \underset{\downarrow}{011}\ \underset{\downarrow}{011}\ \underset{\downarrow}{010}\ \underset{\downarrow}{110}$$
$$1\quad 3\quad 3\quad 2\quad 6$$

因此：1011011．01011B = 133．26Q

(2) 八进制转换为二进制　八进制数转换成二进制数时，只需将每位八进制数码用 3 位二进制数表示即可。

例 1－7　将八进制数 251．76 转换成二进制数。

$$2\quad 5\quad 1\quad .\quad 7\quad 6$$
$$010\ 101\ 001\quad .\quad 111\ 110$$

因此：251．76Q = 10101001．11111B

3. 二进制与十六进制之间的互换

(1) 二进制转换为十六进制　二进制数转换成十六进制数时，以小数点为界，每 4 位二进制数对应 1 位十六进制数：整数部分从低位向高位，每 4 位一组，不足 4 位时左边补 0；小数部分从高位向低位，每 4 位一组，不足 4 位时右边补 0，便可得到相应的十六进制数。

例 1－8　将二进制数 100111100001．001110111 转换成十六进制数。

$$\underset{\downarrow}{1001}\ \underset{\downarrow}{1110}\ \underset{\downarrow}{0001}\ \cdot\ \underset{\downarrow}{0011}\ \underset{\downarrow}{1011}\ \underset{\downarrow}{1000}$$
$$9\quad E\quad 1\quad \cdot\quad 3\quad B\quad 8$$

因此：100111100001．001110111B = 9E1．3B8H

(2) 十六进制转换为二进制　十六进制数转换成二进制数时，只需将每位十六进制数用 4 位二进制数表示即可。

例 1－9　将 2F7．ACH 转换成二进制数。

$$2\quad F\quad 7\quad \cdot\quad A\quad C$$
$$0010\ 1111\ 0111\quad \cdot\quad 1010\ 1100$$

因此：2F7．ACH = 1011110111．101011B

二、机器数

设有两个二进制数在数学上表示为：

$N_1 = +1011011$

$N_2 = -1011011$

N_1 和 N_2 在机器中的表示为：

N_1：01011011

N_2：11011011

即数的符号在机器中数字化了，符号“+”用0表示，符号“-”用1表示。

一个数在机器中的表示形式称为机器数；而把数本身如 +1011011、-1011011 称为真值，真值也可以用八、十、十六进制表示。

以上用8位数说明的概念，其定义同样适用于 n 位数。

机器数分为有符号数和无符号数；还可以分为定点数和浮点数。数在机器中究竟采用哪种表示方法都是事先约定的。

（一）有符号数的表示方法

上面所说的机器数的表示方法，即最高位是符号位，0表示正数的符号，1表示负数的符号，这种表示方法称为有符号数的表示方法。有符号数在机器中的表示通常有三种：原码、反码和补码。下面以定点整数为例说明这三种代码，在说明之前，先介绍模的概念。

把一个计数器的容量称为模或模数，记为 M 或记为 MOD M。例如：一个 n 位二进制计数器，它的容量为 2^n，所以它的模 $M=2^n$。

假设 $n=3$，则 $M=2^3=8$，计数范围为 000～111。当已经计数到 111 时，再加 1，机器中又变成 000，进位 1 自然丢失。也就是说，当模 $M=2^3$ 时，2^3 和 0 在机器中的表示是相同的。

1. 原码

前面介绍的有符号数的表示方法，实际上就是原码的表示法。X 的原码定义为：

$$[X]_{原}=\begin{cases}2^n+X & 当\ 0\leqslant X<2^{n-1}\\ 2^{n-1}-X & 当\ -2^{n-1}<X\leqslant 0\end{cases}\qquad(\text{MOD } 2^n)$$

由原码的定义可知：

1）当 $X>0$ 时，$[X]_{原}$ 与 X 的区别只是符号位用0表示；

2）当 $X<0$ 时，$[X]_{原}$ 与 X 的区别只是符号位用1表示；

3）当 $X=0$ 时，有 $[+0]_{原}$ 和 $[-0]_{原}$ 两种情况：

$$[+0]_{原}=\underbrace{000\cdots0}_{n\text{ 个 }0}\qquad(\text{MOD } 2^n)$$

$$[-0]_{原}=\underbrace{100\cdots0}_{n-1\text{ 个 }0}\qquad(\text{MOD } 2^n)$$

例 1-10 如果 $n=8$，$X=+125$，则有 $[X]_{原}=01111101B$

如果 $n=8$，$X=-125$，则有 $[X]_{原}=11111101B$

2. 反码

负数的反码是它的正数按位取反（包括符号位）形成的。X 的反码定义为：

$$[X]_{反}=\begin{cases}2^n+X & 当\ 0\leqslant X<2^{n-1}\\ (2^{n-1}-1)+X & 当\ -2^{n-1}<X\leqslant 0\end{cases}\qquad(\text{MOD } 2^n)$$

由定义可知：

1）当 $X>0$ 时，$[X]_{反}=[X]_{原}$，即 $[X]_{原}$ 与 X 的区别只是符号位用0表示；

2）当 $X<0$ 时，$[X]_{反}$ 的符号位用 1 表示，其余位为它的原码各位取反；

3）当 $X=0$ 时，$[X]_{反}$ 有两种情况：

$$[+0]_{反}=\underbrace{000\cdots0}_{n\text{ 个 }0} \quad (\text{MOD } 2^n)$$

$$[-0]_{反}=\underbrace{111\cdots1}_{n\text{ 个 }1} \quad (\text{MOD } 2^n)$$

例 1－11 如果 $n=8$，$X=+125$，则有 $[X]_{反}=[X]_{原}=01111101B$

如果 $n=8$，$X=-125$，则有 $[X]_{反}=10000010B$

3. 补码

如果有两个整数 A 和 B，当用一正数 M 去除，所得余数相等时，则称 A 和 B 对模 M 是同余的。

当 A、B 对模 M 同余时，就称 A、B 在以 M 为模时是相等的，记为：

$$A=B \quad (\text{MOD } M)$$

例 1－12 $16=26\ (\text{MOD } 10)$

$8=20\ (\text{MOD } 12)$

于是可得到：$A=A+M\ (\text{MOD } M)$

$A=A+2M\ (\text{MOD } M)$

如果 $A=-3$，$M=10$，则有

$-3=-3+10=7\ (\text{MOD } 10)$

这就是说，在模 M 为 10 时，-3 等于 7。我们称之为：以 10 为模时，-3 的补码是 7。

由上述可知，补码表示法可以把负数转化为正数，使减法转换为加法，从而使正负数的加减运算转换为加法运算。这不仅提高了运算速度，而且节省了机器设备，所以补码是应用最广泛的一种机器数表示法。

X 的补码定义为：

$$[X]_{补}=2^n+X \quad \text{当 } -2^{n-1}\leqslant X<2^{n-1} \quad (\text{MOD } 2^n)$$

由定义可知：

1）当 $X>0$ 时，$[X]_{补}=[X]_{反}=[X]_{原}$，即 $[X]_{补}$ 与 X 的区别只是符号位用 0 表示；

2）当 $X<0$ 时，$[X]_{补}$ 的符号位为 1，其余位为它的原码各位取反，再在最低位加 1；

3）当 $X=0$ 时，$[+0]_{补}=[-0]_{补}=000\cdots0$。

例 1－13 如果 $n=8$，$X=+125$，则有 $[X]_{补}=01111101B=[X]_{原}=[X]_{反}$

如果 $n=8$，$X=-125$，则有 $[X]_{补}=10000011B$

（二）无符号数的表示方法

无符号数与有符号数的区别仅在于，无符号数没有符号位，全部有效位均用来表示数的大小。

在计算机中，无符号数常用来表示地址。此外，双精度数的低位字也用无符号数表示。

（三）定点数的表示方法

定点数有两种特殊的形式，一种是定点整数：小数点在数的最右方，即为纯整数；另一种是定点小数：小数点在符号位之后，即为纯小数。小数点都是隐含的，不作单独的信息存