华东高校计算机基础教育研究会推荐教材

汇编语言程序设计

曹洪其　主编

白洪欢　蔡伯峰　副主编

中国水利水电出版社
上海交通大学出版社
东 南 大 学 出 版 社

内 容 提 要

本书以 8086/8088 为主介绍了汇编语言的基础知识和程序设计方法。全书共分 10 章，分别介绍了汇编语言的基础知识、IBM PC 计算机组织、8086/8088 CPU 的指令系统、汇编语言程序格式与程序调试方法、基本程序设计、子程序设计 、输入输出程序设计和中断技术、系统功能调用和 BIOS 中断调用程序设计、汇编语言与高级语言的连接、TSR 程序设计等内容。

本书深入浅出，循序渐进，系统性强。书中有较多的应用实例，并配有相应的习题。

本书是高职高专计算机专业的教材，同时也可作为非计算机类有关专业的本科教材，或作为从事微型机开发与应用人员的自学参考书。

图书在版编目（CIP）数据

汇编语言程序设计 / 曹洪其主编．—北京：中国水利水电出版社，2001.8
（华东高校计算机基础教育研究会推荐教材）
ISBN 7-5084-0765-2

Ⅰ．**汇…** Ⅱ．**曹…** 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**（2001）**第** 053769 **号**

书　　名	**汇编语言程序设计**
主　　编	**曹洪其**
副 主 编	**白洪欢　蔡伯峰**
出版、发行	**中国水利水电出版社**（**北京市三里河路** 6 **号**　100044） **网址**：www. waterpub. com. cn E-mail：mchannel@public3.bta.net.cn（**万水**） sale@waterpub.com.cn **电话**：（010）68359286（**万水**）63202266（**总机**）68331835（**发行部**）
经　　售	**全国各地新华书店**
排　　版	**北京万水电子信息有限公司**
印　　刷	**北京天竺颖华印刷厂印刷**
规　　格	787×1092 **毫米**　16 **开本**　19.25 **印张**　472 **千字**
版　　次	2001 **年** 8 **月第一版**　2000 **年** 8 **月北京第一次印刷**
印　　数	0001—5000 **册**
定　　价	25.00 **元**

前　　言

汇编语言程序设计是计算机及相关学科的一门专业技术基础课，是微型机系统软件的设计基础。

随着电子技术和计算机技术的发展，新型的微处理器 80X86/Pentium 不断推出，经历了 8086/8088、80286、80386、80486、Pentium、Pentium MMX、Pentium Ⅱ、Pentium Ⅲ等芯片的更新换代。在升级换代的过程中，微处理器的基本结构是向前兼容的，其基本指令集是相同的，只是在 80286 以上的芯片中增加了一些新的指令，以提高计算机的性能。因此，掌握 8086/8088 CPU 是学习 80X86/Pentium 微处理器的基础。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。

本书共有 10 章。第 1 章到第 4 章分别介绍了汇编语言基础知识、IBM PC 计算机组织、8086/8088 CPU 的指令系统和汇编语言程序格式与程序调试，通过这四章的学习，可为学习 8086/8088 CPU 汇编语言的程序设计打下必要的基础。第 5 章和第 6 章以程序设计为主线，介绍了程序设计的基本方法，包括顺序程序设计、分支程序设计、循环程序设计以及子程序设计；第 7 章主要介绍了输入输出程序设计、中断技术，尤其是对 8086/8088 中断系统及编程方法做了重点介绍；第 8 章主要介绍了系统功能调用和 BIOS 中断调用；第 9 章介绍了 TSR 程序设计，这为读者以后开发设备驱动程序、反病毒工具等方面的工作起到一定的作用；第 10 章针对汇编语言和高级语言的优缺点，介绍了汇编语言与高级语言的连接方法，使读者在程序设计中，充分发挥汇编语言和高级语言各自的特长，编制出高质量的程序。

本书是高职高专计算机专业的教材，同时也可作为非计算机类有关专业的本科教材，或作为从事微型机开发与应用人员的自学参考书。

本书第 1 章、第 7 章、第 8 章由曹洪其编写，第 2 章、第 5 章、第 6 章由蔡伯峰编写，第 3 章、第 4 章、第 9 章由白洪欢编写，第 10 章由王秀平编写，全书由曹洪其统稿。

由于编者水平有限，书中难免有疏漏和不妥之处，敬请读者批评指正。

编　者

2001 年 6 月

目　　录

1 汇编语言基础知识

主题词提要

●计算机系统的组成　　●程序设计语言
●汇编语言的应用　　●计算机中信息的表示

本章主要介绍了汇编语言的基础知识。主要内容包括：计算机系统的硬件、软件；程序设计语言，汇编语言的应用；计算机中信息的表示，诸如进位计数制、机器数与真值、8421 BCD码、逻辑数据和字符数据等。通过这一章的学习，为以后各章的学习作好必要的准备。

1.1 计算机系统的组成

计算机系统包括硬件部分和软件部分。硬件（Hardware）是指计算机中的电子线路和物理装置，是计算机工作的物理基础。软件（Software）是相对于硬件而言的，它是指在计算机硬件上运行所需的各种程序和相关的数据及文档。

1.1.1 计算机硬件的组成

在1946年，美籍匈牙利数学家冯·诺伊曼（John Von Neumann）等人在一篇“关于电子计算仪器逻辑设计的初步探讨”的论文中，提出了计算机的组成和工作方式的基本思想。其内容概括如下：

- 计算机由五个组成部分：运算器、控制器、存储器、输入设备和输出设备。
- 在计算机内部采用二进制表示数据和指令。指令（Instruction）是控制计算机工作的，计算机能够识别和执行的命令。
- 计算机采用存储程序控制原理。程序是完成一定任务的指令序列，其中的每条指令规定机器完成一个基本操作。存储程序控制原理就是：首先必须把为解决某特定任务而事先编制好的程序和运行中所需的数据一起输入存储器保存起来，然后由控制器执行程序，使计算机按程序的规定步骤运行。

这些概念奠定了当代计算机的基本结构思想。数十年来，计算机技术虽取得惊人的进步，但到目前为止，大多数计算机仍属于冯·诺伊曼机器结构体系。图1.1是基于冯·诺伊曼理论的计算机的基本组成结构框图。

1. 存储器

存储器的主要功能是存放数据的。在计算机内部，程序和数据都是以二进制的形式来表示。

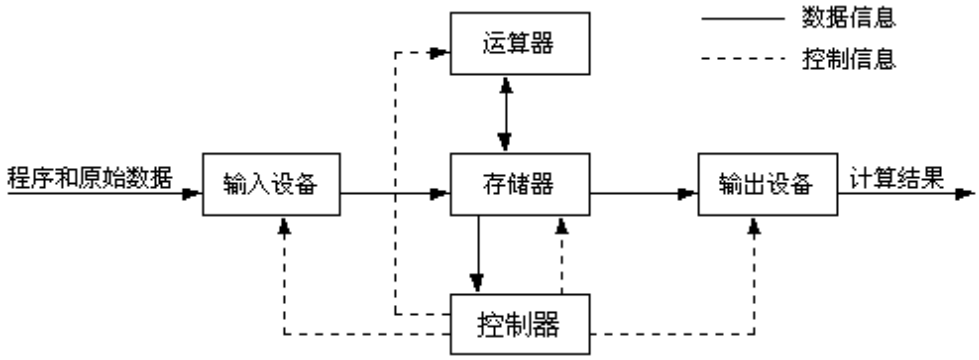

图 1.1 计算机的基本组成结构框图

存储器由大量的存储单元组成，每个存储单元存放着指令代码或数据。为了区别不同的存储单元，每个存储单元都有一个编号，通常将这个编号称为存储单元的地址。用户可以按地址存取信息。每个存储单元的长度随具体的机器而有所不同。有的计算机存储单元长度为一个字节（八位二进制数），而又有一些计算机存储单元的长度可以为一个字或为双字。在 PC 机中，每个存储单元存放一个字节的数据。存储器所能存储的二进制信息量称为存储器容量，表示容量的单位一般用字节或字表示。例如，一个存储芯片有 1024 个单元，每个单元存放 8 位二进制信息，则内存容量为 1024×8 位或说成 1024B（1024 字节）。为了表达更大的单位，常用的单位是 KB（千字节）、MB（兆字节）。

$1KB=2^{10}B=1024B$

$1MB=2^{20}B=1024KB$

$1GB=2^{30}B=1024\ MB$

$1TB=2^{40}B=1024\ GB$

存储器在运算之前，接受从输入设备送来的程序和数据。在运算过程中，提供指令和数据，保存中间结果。运算结束后，保存运算结果。

2. 运算器

运算器的功能是进行信息的加工处理，即执行算术运算和逻辑运算，所以运算器又称为算术逻辑单元（Arithmetic Logic Unit，简称 ALU）。算术运算主要包括加、减、乘、除等运算，逻辑运算主要包括与、或、非、异或等运算。

3. 控制器

控制器是全机的控制中心。对存储器进行信息的存取、运算器进行各种运算、信息的输入输出都是在控制器的控制下进行的。控制器是通过执行程序即指令序列，向机器的各部分发出控制信号来控制整台机器各部件自动协调地工作。

4. 输入设备

输入设备是人与计算机交互的入口，它的任务是输入用户提供的原始信息（如程序和原始数据），并将其变换为计算机能识别的信息形式。常用的输入设备有键盘、鼠标、扫描仪等。

5. 输出设备

输出设备是人与计算机交互的出口，它的任务是将计算机的处理结果以人们能够接受的

形式输出。常用的输出设备有显示器、打印机、绘图仪等。

在组成计算机的上述五大部件中，运算器和控制器合称为中央处理器 CPU（Central Processing Unit）或处理器（Processor）。在微型计算机中，CPU 被称为微处理器（Microprocessor）。中央处理器和存储器组成计算机的主机，这里的存储器也称为内存储器（简称内存）。输入设备和输出设备统称为外部设备，简称外设或 I/O 设备。

由于内存储器容量受到成本的限制而做得比较小，因此为了弥补内存储器容量的不足，就引入了外存储器（简称外存）。外存储器归属外部设备，它与内存储器相比，每位的价格低，存储容量大，但存取速度慢，主要用于存放当前不运行的程序和数据，当需要运行时，必须先调入内存储器。外存储器的种类很多，当前用得最多的是磁盘存储器和光存储器。

1.1.2 计算机的软件系统

软件是计算机系统的重要组成部分，它分为两大类：一类称为系统软件，另一类称为应用软件。

1. 系统软件

系统软件是指用于实现管理和维护计算机系统的软件，其目的是方便用户使用计算机，提高计算机的使用效率，扩充系统的功能。系统软件主要包括操作系统、语言处理程序以及一些常用的实用程序等。

（1）操作系统（Operating system）

操作系统是最重要的系统软件。它的主要作用是控制和管理计算机系统的所有硬件资源和软件资源，合理地组织计算机工作流程，并为用户提供良好的工作环境。用户只要正确地使用操作系统提供的各种命令和系统功能调用，就可以使用户程序在操作系统的统一控制下自动而协调地运行。目前的主要操作系统有：Windows 98、Windows 2000、Windows Me、Windows NT、Unix、Linux 等。

（2）语言处理程序

语言处理程序分为三类：汇编程序、解释程序和编译程序。

汇编程序（Assembler）：将用汇编语言编写的源程序翻译加工成机器语言表示的目标程序的程序。

解释程序（Interpreter）：将用高级语言设计编写的源程序按动态的运行顺序逐句进行翻译并执行的程序。

编译程序（Compiler）：将用高级语言编写的源程序翻译成机器语言表示的目标程序的程序。

从上述可见，汇编方式和编译方式都产生目标程序，而解释方式不产生目标程序。解释程序和编译程序都是高级语言的翻译程序，它们之间的区别是，解释程序的工作方式，便于实现人机对话，但程序执行的速度比较慢。经编译产生的目标程序执行速度快，但对源程序修改后必须重新翻译。

（3）实用程序

常用的实用程序有编辑程序、连接装配程序、测试及诊断程序等。

编辑程序（Editor）为用户提供了方便的编辑环境。用户通过使用编辑程序可建立、修改、存储程序文件、数据文件或其他文件。

连接装配程序（Linking Loader）是用来把若干个目标程序与库文件连接起来，形成能够执行的程序。

测试程序（Checking Program）能够用来检查出程序中的某些错误，诊断程序（Diagnostic Program）能够自动检测计算机硬件的某些故障。

2. 应用软件

应用软件是指为解决某种应用问题而设计的程序及其文档，它是面向具体问题和具体用户的软件。计算机具有多种多样的应用软件，如科学计算程序、数据处理程序、生产过程控制程序、多媒体应用程序等。目前一些应用软件有的已逐步标准化、模块化，形成了解决某类问题的应用程序组合，即软件包（Package）。如财务管理软件包、统计软件包等。

1.2 程序设计语言

程序设计语言就是编制程序的语言，是人与计算机通信的工具。程序设计语言主要包括机器语言、汇编语言和高级语言。

1.2.1 机器语言

机器语言（Machine Language）是能被计算机硬件直接识别并执行的语言，它是一种用二进制代码形式表示的机器指令组成的语言。在机器语言中，每一条指令（Instruction）控制计算机完成一个操作，一台机器的全部指令构成该机的指令系统。

用机器语言编制的程序送入计算机后能直接识别并执行，而用其他语言编制的程序必须经过翻译成机器语言程序，计算机才能执行，所以机器语言程序称为目标程序。由于机器语言程序是用二进制代码表示的，因此机器语言程序不易书写、阅读和理解，而且极易出错，错了也难以修改。所以，只是在计算机发展的早期或在不得已的情况下，才用机器语言编写程序。

1.2.2 汇编语言

为了克服机器语言的缺点，人们提出了汇编语言（Assembly Language）。在汇编语言中，采用一组字母、数字、符号来代表二进制码表示的指令。表示指令操作码的符号称为指令助记符，简称助记符。助记符一般是表明指令操作功能的英语单词或其缩写，同样指令操作数也可以用易于记忆的符号来表示。由于汇编语言是一种符号语言，因此它比机器语言容易理解和掌握。但是，用汇编语言编制的程序（称为汇编语言源程序）计算机不能直接识别和执行，必须翻译成机器语言程序（称为目标程序）才可以由处理器执行。这个翻译的过程称为“汇编”，完成汇编工作的程序就是汇编程序（Assembler）。另外，汇编语言同机器语言一样，随 CPU 的不同而不同，是面向机器的语言，通用性差。机器语言和汇编语言统称为低级语言。

1.2.3 高级语言

高级语言（High-level Programming Language）是一种与具体的计算机硬件无关，独立于机器的通用语言。它比较接近于人类自然语言的语法习惯及数学表达形式，变得更容易掌握

和使用。因此，人们用高级语言编写程序，当不对硬件编程时，可不必了解和熟悉机器的指令系统，而且高级语言程序在各机器上可通用（有时仅需做少量修改）。当然，用高级语言编制的源程序不会被机器直接认识和执行，需经过编译或解释程序翻译成机器语言程序后机器才能执行。

高级语言的缺点是：高级语言源程序当不对硬件编程时，它是在未考虑机器结构特点下编写的，经过翻译后的目标程序往往不够精练、过于冗长，加大了目标程序的长度，导致占有较大的存储空间和较长的执行时间。

1.2.4 汇编语言的应用

高级语言简单、易学。相比之下，汇编语言复杂、难学不直观。但汇编语言是一种面向机器的语言，它可以直接地、有效地控制计算机硬件，由汇编语言产生的目标程序指令序列短、运行速度快。这些优点使得汇编语言在程序设计中占有极重要的地位。汇编语言主要应用于以下几个方面：

- 用于编制能直接有效地控制硬件的程序。例如：I/O 接口电路的初始化程序段、外部设备的低层驱动程序等。
- 用于编制要求运行速度较快的程序，或者只能占用较小内存空间的程序。例如：计算机系统频繁调用的子程序、动态连接库、操作系统的核心程序段、实时控制系统的软件及智能仪器仪表的控制程序等。
- 用于在没有合适的高级语言或只能采用汇编语言的场合。例如：开发最新的处理机程序时，暂时没有支持新指令的编译程序。
- 用于分析计算机系统、编制加密软件、分析和防治计算机病毒等。

1.3 数据信息的表示

计算机中的数据信息包括数值数据和非数值数据。数值数据有确定的值，用来表示数量的多少，对它们能进行各种算术运算和处理。其余数据为非数值数据。在非数值型数据中又有一类最常用的数据，称为字符型数据，它可以方便地表示文字信息。

1.3.1 数制与转换

数制也称为进位计数制。在日常生活中，人们习惯于采用十进制数进行计算，而在计算机中所有的数据都是以二进制代码的形式存储、处理和传送的。但是在输入和输出或书写时，为了用户的方便，也经常用到十六进制数。

1. 十进制数

十进制数的三个主要特点：

（1）由十个不同的数码 0，1，2，3，4，5，6，7，8，9 来表示数。

（2）由低位向高位的进位规则是“逢十进一”。

（3）在数中的任何一个数码所表示的数值等于该数码与该位“位权”值的乘积。第 i 位的“位权”值为 10^i。

十进制数 $a_n a_{n-1}\cdots a_1 a_0.a_{-1}a_{-2}\cdots a_{-m}$ 可表示为：

$a_n\times10^n+a_{n-1}\times10^{n-1}+\cdots+a_1\times10^1+a_0\times10^0+a_{-1}\times10^{-1}+a_{-2}\times10^{-2}+\cdots+a_m\times10^{-m}$

2. **二进制数**

二进制数的三个主要特点：

（1）**由二个不同的数码** 0，1 **来表示数**。

（2）**由低位向高位的进位规则是“逢二进一”**。

（3）**第** i **位的“位权”值为** 2^i。

二进制数 $a_na_{n-1}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m}$ **可表示为：**

$a_n\times2^n+a_{n-1}\times2^{n-1}+\cdots+a_1\times2^1+a_0\times2^0+a_{-1}\times2^{-1}+a_{-2}\times2^{-2}+\cdots+a_m\times2^{-m}$

3. **十六进制数**

十六进制数的三个主要特点：

（1）**由十六个不同的数码** 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F **来表示数**。

（2）**由低位向高位的进位规则是“逢十六进一”**。

（3）**第** i **位的“位权”值为** 16^i。

十六进制数 $a_na_{n-1}\cdots a_1a_0.a_{-1}a_{-2}\cdots a_{-m}$ **可表示为：**

$a_n\times16^n+a_{n-1}\times16^{n-1}+\cdots+a_1\times16^1+a_0\times16^0+a_{-1}\times16^{-1}+a_{-2}\times16^{-2}+\cdots+a_m\times16^{-m}$

表 1.1 **列出了十进制数、十六进制数和二制数之间的关系**。

表 1.1　**十进制数、十六进制数和二进制数的关系**

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

在实际应用中，为了区别一个数是属于何种进位计数制的数，往往在数后加上一个标志符号。十进制数后加字母 D **或不加字母，二进制数后加字母** B，**十六进制后加字母** H。

4. **不同数制之间的转换**

（1）**二进制数与十进制数之间的转换**

① **二进制数转换成十进制数**

二进制数转换成十进制数，只需将二进制数按权展开相加即可。

【例 1.1】$1110.101B=1\times2^3+1\times2^2+1\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=14.625D$

② **十进制数转换成二进制数**

将十进制数转换成二进制数时，采用的方法是：将整数部分和小数部分分别转换成相应的二进制整数部分和小数部分，然后合并在一起即可。

整数部分的转换方法的是“除 2 取余”法。即将十进制数的整数部分连续除以 2，可依次得到余数，直到商为 0 为止，将余数逆序排列便可得到二进制整数。

小数部分的转换方法是“乘 2 取整”法。即将十进制数的小数部分乘以 2，得到一个乘积。再将乘积中的小数部分乘 2，……，如此连续做下去，直到乘积的小数部分为 0 或达到一定的精度为止。将依次得到的乘积的整数部分顺序排列便是转换的结果。

【例 1.2】 将十进制数 18.6875 转换成二进制数。

解：整数部分的转换过程如下：

```
2 | 18
2 |  9    余数=0
2 |  4    余数=1
2 |  2    余数=0
2 |  1    余数=0
   0      余数=1
```

所以　18D=10010B

小数部分的转换过程如下：

```
   0．6875
 ×      2
 ------------
   1．3750    ……  乘积的整数部分为 1
   0．3750
 ×      2
 ------------
   0．7500    ……  乘积的整数部分为 0
 ×      2
 ------------
   1．5000    ……  乘积的整数部分为 1
   0．5000
 ×      2
 ------------
   1．0000    ……  乘积的整数部分为 1
```

所以　0.6875D=0．1011B

整数部分和小数部分并在一起得：

18.6875D=10010．1011B

（2）二进制数与十六进制数之间的转换

二进制数与十六进制数之间的转换很方便，其原因是二进制数与十六进制数之间存在着简单而又直接的关系，即用四位二进制数表示一位十六进制数。

① 二进制数转换成十六进制数

二进制数转换成十六进制数时，只要把每一位十六进制数用相应的四位二进制数代替，就实现了转换。

【例 1.3】 将十六进制数 82C.3E5 转换成二进制数。

解： **十六进制数** 8 2 C . 3 E 5

二进制数 1000 0010 1100 . 0011 1110 0101

所以 82C.3E5H=100000101100.001111100101B

② **十六进制数转换成二进制数**

十六进制数转换成二进制数时，只需从小数点开始，分别向左和向右每四位分为一组，对于整数部分不足四位的前面补 0，**对于小数部分不足四位的后面补** 0，**然后将每四位一组的二进制数用相应的一位十六进制数代替就实现了转换。**

【例 1.4】 **将二进制数** 1101011101.010111001B **转换成十六进制数。**

解： **二进制数** 0110 1001 1101．0101 1100 1000

十六进制数 6 9 D 5 C 8

所以 11010011101.010111001B=69D.5C8H

1.3.2 数值数据的表示

1. **机器数与真值**

在计算机中，由于采用二进制数，因此只能用 0 **和** 1 **两种数码。为了表示正数和负数，专门选择一位符号位来表示数的符号。通常用最高位作为符号位，用“**0**”表示正号，用“**1**”表示负号。数在计算机内的表示形式称为机器数，而把它对应的实际数值称为真值。常用的机器数有三种不同的编码形式，即原码、反码和补码。**

（1）**原码**

最高位表示符号（正数用 0 **表示，负数用** 1 **表示），其他位表示数的绝对值，这种数的表示形式称为原码。若真值为** X，**则其原码记作**$[X]_{原}$。

【例 1.5】**设字长为** 8 **位，分别求**+1001010B **和–**1001010B **的原码。**

X1=＋1001010B $[X1]_{原}$=01001010B

X2=－1001010B $[X2]_{原}$=11001010B

原码表示法的优点是简单易懂，缺点是进行加减运算复杂。当两个数求和时，首先要判断它们的符号。如果符号相同，则绝对值部分相加；否则，绝对值部分相减。若相减，还必须先比较两个数的绝对值的大小，然后做减法，最后给结果确定适当的符号。这使得求和过程变得复杂。为了简化运算过程，把减法运算转换为加法运算，引进了反码和补码表示法。

（2）**反码**

正数的反码和原码相同。负数的反码，可通过将该数原码的各位（符号位除外）按位取反而得到。取反就是 0 **变成** 1，1 **变成** 0。**若真值为** X，**则其反码记作**$[X]_{反}$。

【例 1.6】**设字长为** 8 **位，求分别求**+1001010B **和–**1001010B **的反码。**

X1=+1001010B $[X1]_{反}$=01001010B

X2=－1001010B　　$[X2]_{反}$=10110101B

（3）**补码**

正数的补码和原码相同。负数的补码，就是将该数反码在最低位加 1 **而得到。因此，可从反码的表示法中获得求负数补码的简便方法。若真值为** X，**则其补码记作**$[X]_{补}$。

【例 1.7**】设字长为** 8 **位，分别求**+1001010B **和–**1001010B **的补码。**

X1=+1001010B　　$[X1]_{补}$=01001010B

X2=－1001010B　　$[X2]_{补}$=10110110B

【例 1.8**】已知**$[X]_{补}$=10101101B，**求**$[－X]_{补}$。

解：由$[X]_{补}$**求**$[－X]_{补}$**的方法是，可通过将**$[X]_{补}$**的各位（连同符号位）求反，并在末位加** 1 **得到。**

所以，本例题的$[－X]_{补}$=01010011B。

在计算机中，有符号的数一般采用补码形式表示，这主要的原因是补码的加减运算比原码简单，能将减法运算转换为加法运算。进行补码加减运算的三条规则如下：

① **其符号位要与数值位一起参加运算。**

② **符号运算后如有进位出现，则把这个进位舍去不要。**

③ **补码运算具有的两个运算性质：**

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

【例 1.9**】已知** X=1000011B，Y=－1101101B，**求** X＋Y。

解：　　$[X]_{补}$=01000011B　　$[Y]_{补}$=10010011B

$$\begin{array}{rl} & 01000011=[X]_{补} \\ ＋ & 10010011=[Y]_{补} \\ \hline & 11010110=[X＋Y]_{补} \end{array}$$

所以　X＋Y=－0101010B

【例 1.10**】已知** X=1100010B，Y=0101011B，**求** X－Y。

解：　　$[X]_{补}$=01100010B　　$[Y]_{补}$=00101011B　　$[－Y]_{补}$=11010101

$$\begin{array}{rl} & 01100010=[X]_{补} \\ ＋ & 11010101=[－Y]_{补} \\ \hline & 1\ 00110111=[X－Y]_{补} \end{array}$$

↓

进位自然丢失

所以　X－Y=0110111B

必须注意的是：补码表示的数值范围。例如，字长为 8 **位二进制补码所能表示的数值范围为＋**127 ~ －128**。在进行补码运算时，若运算的结果超出了补码所能表示的数值范围，则会产生溢出错误。**

2. **二—十进制编码**

（1）**二—十进制编码**

用二进制数码来表示十进制数，称为二进制编码的十进制数，简称 BCD（Binary Coded

Decimal）码。

一位十进制数需用 4 位二进制编码来表示。然而，4 位二进制编码可组合成 16 种不同的状态。因此，选择其中的 10 种状态作 BCD 码的方案有许多种，而常采用的有“8421 BCD”码。8421 BCD 码选取 4 位二进制编码的前 10 个代码分别表示 0 ~ 9 这 10 数码，1010 ~ 1111 这 6 个代码未被选用，如表 1.2 所示。

表 1.2 8421 BCD 码

十进制数	8421 BCD 码
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

从表 1.2 可见，8421 BCD 码是有权码，编码中的每一位仍然保留着一般二进制数所具有的位权，而且 4 位代码从左到右的位权依次是 8、4、2、1。8421 BCD 码就是据此而命名的。

【例 1.11】十进制数 98.5 的 8421 BCD 码为 10011000.0101。

要注意 BCD 码与真正的纯二进制是不同的。它貌似二进制，实为十进制，通常用在计算机的输入/输出设备中，作为计算机用的二进制与人们常用的十进制之间的一种过渡性编码，以简化人机关系。

（2）8421 BCD 码的运算

BCD 码是用 4 位二进制数来表示一位十进制数，每 4 位二进制数之间是逢十进一的。但将 BCD 码直接交给计算机运算，由于计算机总是将数作为二进制数来处理，每 4 位二进制数是逢十六进一的，因此运算结果有可能出错。当两个 8421 BCD 码相加时，若相加的和在 0~9 之间，且没有向高位产生进位，则其运算结果正确；若相加的和大于 9，或向高位产生了进位，则应进行加 6 调整。

【例 1.12】求 56＋39。

解：
```
  01010110
＋ 00111001
  10001111
＋     0110    低 4 位加 6 调整
  10010101
```
所以 56＋39=95

BCD 码在计算机中有两种表示法：一种是压缩 BCD 码表示法，即用 4 位二进制编码表示一位十进制数，这种表示法一个字节可表示两位十进制数，如（68）$_{10}$=（01101000）$_{BCD}$；另一种是非压缩 BCD 码表示法，它用一个字节表示 1 位十进制数，其中高 4 位通常为 0000，如（68）$_{10}$=（0000011000001000）$_{BCD}$。

1.3.3 非数值数据的表示

1. 逻辑数据与逻辑运算

逻辑数据由无符号二进制代码组成，每位不表示数值，只表示逻辑值（真或假）。