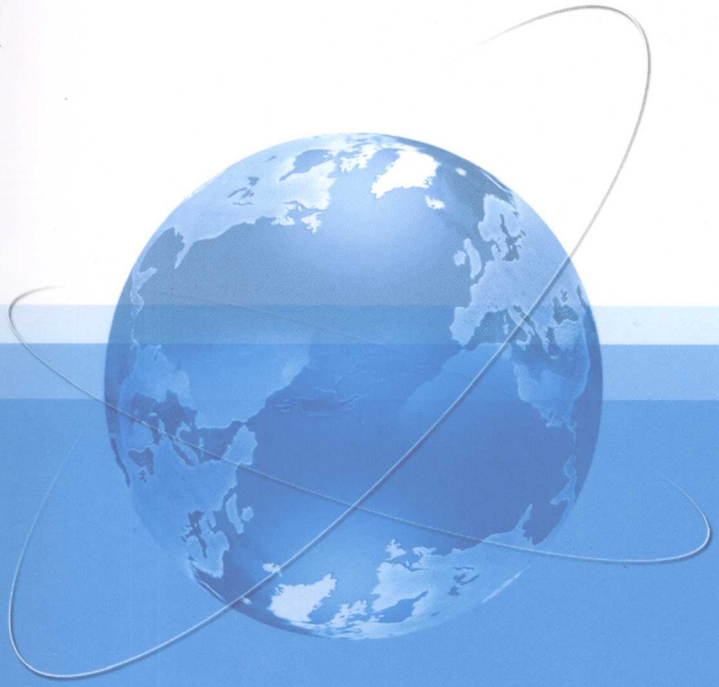
21世纪高职高专规划教材

（计算机类）

汇编语言程序设计



董少明　主编

机械工业出版社
CHINA MACHINE PRESS

汇编语言程序设计

21世纪高职高专规划教材
(计算机类)

汇编语言程序设计

主　编　董少明
副主编　范爱华　陈晓冬
参　编　黄金水　米应恺

机械工业出版社

本书立足于实用性、技能性，以 Intel 8086/8088 指令系统为背景，简明扼要地介绍了汇编语言的基本概念、基本原理和程序设计的基本方法，以大量实例讲述了如何使用汇编语言开发应用程序，并介绍了上机调试运行汇编源程序的方法。全书共分 9 章，分别为概述，8086/8088 的指令系统、汇编语言程序设计基础、上机过程、汇编语言程序设计方法、子程序设计方法、I/O 和中断传送方式、系统功能调用与程序设计、应用系统开发和高级汇编技术。本书提供了大量实例，每章后都附有小结和复习思考题。

本书在内容的选取、概念的引入、文字的叙述、例题和习题的选择方面力求做到循序渐进、结构清晰、明晰易懂。书中列举的一些有代表性的实例，有助于学生提高实际动手能力。

本书可作为高职高专计算机类各专业教材，也可供相关专业学生或电大、函大学生以及自学考试等人员参考使用。

图书在版编目（CIP）数据

汇编语言程序设计/董少明主编. —北京：机械工业出版社，2006.10
21 世纪高职高专规划教材. 计算机类
ISBN 7-111-20172-8

Ⅰ.汇… Ⅱ.董… 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（2006）第 124586 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：余茂祚
责任编辑：余茂祚 版式设计：冉晓华 责任校对：吴美英
封面设计：饶 薇 责任印制：洪汉军
北京京丰印刷厂印刷
2007 年 1 月第 1 版·第 1 次印刷
184mm×260mm·13.75 印张·334 千字
0 001—4 000 册
定价：22.00 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）68326294
编辑热线电话（010）68354423
封面无防伪标均为盗版

21世纪高职高专规划教材
编 委 会 名 单

编委会主任 王文斌

编委会副主任 （按姓氏笔画为序）

王建明　王明耀　王胜利　王寅仓　王锡铭　刘　义
刘晶磷　刘锡奇　杜建根　李向东　李兴旺　李居参
李麟书　杨国祥　余党军　张建华　茆有柏　秦建华
唐汝元　谈向群　符宁平　蒋国良　薛世山　储克森

编委会委员 （按姓氏笔画为序，黑体字为常务编委）

王若明　**田建敏**　成运花　曲昭仲　朱　强　**刘　莹**
刘学应　许　展　**严安云**　李连邺　李学锋　李选芒
李超群　**杨　飒**　**杨群祥**　杨翠明　吴　锐　何志祥
何宝文　余元冠　**沈国良**　张　波　**张　锋**　张福臣
陈月波　**陈向平**　陈江伟　武友德　林　钢　周国良
宗序炎　赵建武　恽达明　**俞庆生**　晏初宏　倪依纯
徐炳亭　**徐铮颖**　韩学军　崔　平　崔景茂　**焦　斌**

总 策 划 余茂祚

前　言

本书是根据教育部有关文件精神，针对高职高专教育特点，由中国机械工业教育协会和机械工业出版社组织全国80多所院校编写的21世纪高职高专规划教材之一。汇编语言程序设计是高等职业技术院校计算机专业的必修课之一。它是操作系统、微型计算机接口技术、单片机等课程的先修课。它是计算机应用与研究的基础，也是计算机专业人员必须接受的专业基础训练之一。因为汇编语言是计算机能够提供给编程者执行速度最快的语言，也是能够利用计算机硬件特性直接控制硬件设备的惟一一种语言，因而在对于软件的空间和时间要求很高的场合，汇编语言是最有效的编程工具。尽管现在已有许多新的、易写的高级语言出现，但是目前功能最强、效率最高的软件很多还都是用汇编语言编写的。本书选择8086/8088作为主讲CPU，主要是考虑到Intel系列CPU的覆盖面很广，现在的计算机很多都是这一系列的产品，就是“奔腾”机也兼容了8086/8088的所有功能，所以学习好8086/8088的汇编语言是为进一步学习和提高打下基础。另外，学习汇编语言程序设计也是为掌握计算机硬件知识和计算机工业控制打下基础。

全书共分9章。第1章介绍了学习汇编语言程序设计所必须掌握的基本知识；第2章介绍了8088的寻址方式及其基本指令集；第3章介绍了汇编语言程序设计基础及汇编语言程序的上机过程；第4章介绍了汇编语言程序设计的方法，包括汇编语言程序设计的基本步骤和各种结构的程序设计方法；第5章介绍了子程序设计方法；第6章介绍了I/O和中断传送方式；第7章介绍了系统常用的BIOS中断功能调用和常用的DOS系统功能调用与程序设计；第8章通过图形显示及动画程序设计、音乐程序设计、磁盘操作程序设计介绍了汇编语言的应用和开发；第9章介绍了高级汇编技术。

本书在内容的选取、概念的引入、文字的叙述、例题和习题的选择方面力求做到循序渐进、结构清晰、明晰易懂。书中列举的一些有代表性的实例，有助于学生提高实际动手能力。

本书由陕西工业职业技术学院董少明主编，并负责全书的总体规划和统稿工作，扬州工业职业技术学院范爱华、长治职业技术学院陈晓冬任副主编。其中，董少明编写了第1、2章，范爱华编写了第3~5章，陈晓冬编写了第6、7章，张家界航空工业职业技术学院黄金水编写了第8章，西安理工大学高等技术学院米应恺编写了第9章。

由于编者水平有限，书中难免有错误和疏漏之处，敬请专家和广大读者批评指正。

编　者

目　录

第1章 概 述

计算机是一种能够按照人们预先存放在内存中的一系列命令连续、高速地进行数据处理的电子机器。能够把人的命令告诉计算机的一套符号系统及其使用规则称为“计算机语言”。到目前为止，计算机语言已经由低级到高级经历了机器语言、汇编语言、高级语言、应用语言的发展过程。其中，汇编语言是一种能够充分利用计算机硬件特性的低级语言，它与计算机的结构有非常紧密的联系。不同的计算机有不同的汇编语言。本书介绍 Intel 8086/8088 的汇编语言。

1.1 计算机语言的分类及特征

1.1.1 机器语言

微型计算机内部所有的信息都是采用二进制 0 和 1 的位串来表示的，机器语言及其指令就是计算机能够直接识别和执行的一组二进制代码，它指明计算机执行时必须完成的一种操作及其操作的对象。一条机器指令通常由操作码和操作数两部分构成，即

操作码	操作数

操作码指出计算机所执行的何种操作，即该指令的功能；而操作数则指出在指令操作过程中所需的操作数据，即操作对象。在指令中可以直接给出操作数本身或者操作数存放的地址，以及操作结果送往何处等信息。

下面的二进制代码序列就是一条 8088 的机器指令：

10111000 00000101 00000000

这条指令的前 8 位是操作码部分，含义是要求计算机完成把指令后面的 16 位数传送到名为 AX 寄存器的操作；后 16 位是操作数部分，指出这是一个 16 位的二进制数。

对于同样的二进制序列，不同型号的 CPU 对它的“理解”是不一样的。比如上面的那一行指令代码在 8088CPU 看来是要求作数据传送，换到另一种 CPU 中完全可能被当作是另一种操作，甚至是错误的指令。所以机器代码与机器本身有着紧密的联系，每一种计算机都有自己的一套指令。

不同的计算机系统具有各自不同的指令，对某种特定的计算机而言，其所有机器指令的集合称为机器指令系统。它既是提供给用户编制程序的基本依据，也是进行计算机逻辑设计的基本依据。指令系统的性能如何，决定了计算机系统的基本功能。机器指令系统及其使用规则构成这种计算机的机器语言。完成特定功能的一系列机器指令的有序集合称为机器语言程序。

依上所述，机器语言具有以下特征：

1）它是惟一的能被计算机识别并执行的语言。

2）它是由 0、1 代码构成的语言，和自然语言相差甚远，不便于阅读和理解。

3）它是面向机器的语言（低级语言）。

1.1.2 汇编语言

机器语言虽然是计算机的“母语”，但对绝大多数使用计算机的人来说，机器语言难以掌握与编程。为了克服机器指令的上述缺点，采用容易记忆的英文符号（称为助记符）来表示指令和数据及其地址，例如用 ADD、SUB 和 JMP 等英文文字或其缩写形式取代原来的二进制操作码，来表示加、减和转移等操作。这种用助记符来表示的机器指令称为汇编指令。每一条机器指令都对应一条汇编指令，所有汇编指令的集合构成计算机的汇编指令系统。

（1）汇编指令。又称为符号指令，它是机器指令符号化的表示。前面的例子用汇编指令书写应为

MOV AX，05H

其中，MOV 为传送指令的助记符；AX 是目的操作数；05H 是源操作数。指令的功能是将 05H 传送到寄存器为 AX 的目的地址。

（2）汇编语言。由汇编指令、汇编伪指令及汇编语言的语法规则组成。

（3）汇编语言源程序。它是按照严格的语法规则用汇编语言编写的程序，也可简称为源程序。

（4）汇编程序。由于计算机不能直接识别和执行汇编语言源程序，因此需要把汇编语言源程序翻译成机器语言程序才可以由计算机执行。这个翻译的过程称为“汇编”，这种把汇编语言源程序翻译成目标程序的语言加工程序称为汇编程序。汇编程序将其翻译为机器语言后，才能交付计算机硬件系统加以识别和执行。汇编程序是为计算机配置的实现把汇编语言源程序翻译成目标程序的一种系统软件。其过程如图 1-1 所示。

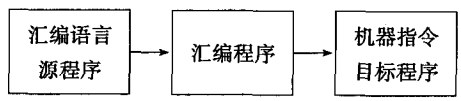


图 1-1　汇编程序过程

一般的微型计算机上配有 2 个汇编程序，即小汇编程序 ASM 和宏汇编程序 MASM。MASM 的功能强大，它支持宏汇编，所以建议使用 MASM。

依上所述，汇编语言具有以下特征：

1）它是机器指令的符号化表示，较接近自然语言，容易编程、阅读和记忆。

2）翻译程序是一对一的转换，生成的目标代码效率高（时空性能好）。

3）适合于在硬件层次上开发程序。

1.1.3 高级语言

汇编语言虽然较机器语言直观，但仍然烦琐难懂。于是人们研制出了高级程序设计语言。高级程序设计语言接近人类自然语言的语法习惯，与计算机硬件无关，用户易于掌握和使用。目前广泛应用的高级语言有多种，如 BASIC、FORTRAN、PASCAL、C 和 C ++ 等。同样道理，用高级语言书写的源程序也必须翻译成机器指令目标程序。完成此翻译任务的程序称为编译程序。这样一来，编译程序和汇编程序好像差别不大，但汇编程序是一对一的转换，而编译程序则是一对多的转换。

依上所述，高级语言具有以下特征：

1）更接近于自然语言，编程、阅读更容易。

2）一台计算机是否支持该高级语言与其硬件系统无关。

3）生成的目标代码效率低（时空性能差），只取决于有无相应的编译程序。

1.1.4 汇编语言的特点

（1）汇编语言与处理器密切相关。由于不同的处理器使用不同的汇编语言，所以汇编语言源程序与高级语言源程序相比，汇编语言程序的通用性、可移植性较差。但与机器语言相比，汇编语言易于理解和记忆，编写的源程序可读性强，源程序翻译成机器语言后的执行文件在存储空间、执行速度方面与机器语言编写的程序大致相当。

（2）汇编语言程序效率高。汇编语言源程序汇编后的目标程序短小精悍，运行效率高。其高效率反映在时间和空间2个方面：①运行速度快。②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时间和空间的效率都不如汇编语言程序。

（3）编写汇编语言源程序比编写高级语言源程序烦琐。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。程序员在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问题，而在使用高级语言编写程序时，程序员不用考虑这些细节问题。

（4）调试汇编语言程序比调试高级语言程序困难。汇编语句指令的有限功能和程序员要注意太多的细节问题往往是造成调试困难的两个主要原因。

汇编语言的主要应用场合如下：

1）程序执行占用较短的时间，或者占用较小存储容量的场合。如操作系统的核心程序段、实时控制系统的软件和智能化仪器仪表的控制程序等。

2）程序与计算机硬件密切相关，程序直接控制硬件的场合。如I/O接口电路的初始化程序段和外部设备的低层驱动程序等。

3）需提高大型软件性能的场合。如计算机系统频繁调用的子程序等。

4）没有合适的高级语言的场合。如开发最新的处理器程序时，暂时没有支持新指令的编译程序。

1.2 计算机的运行基础

1.2.1 进位计数制及其转换

采用数字符号排列，按照由低位向高位进位计数的方法称为进位计数制，简称计数制或进位制。在人们的日常生活中，会碰到各种不同的进位计数制，不仅有最常使用的十进制，还有二进制、八进制、十二进制、十六进制和二十四进制等。

1）二进制。由数字符号0、1构成，逢2进1。

2）八进制。由数字符号0～7构成，逢8进1。

3）十进制。由数字符号0～9构成，逢10进1。

4）十六进制。由数字符号0～9和字母A～F构成，逢16进1。

在计算机中，数的表示仅采用二进位计数制。

计算机仅能识别0、1代码，计算机能够理解的语言只能是由0、1构成的语言。计算机内部处理的数据（数值数据、字符、图形和声音等）必须用0、1代码表示，即用二进制数据表示。而用户在书写时则是可以用任何进制形式表示的。

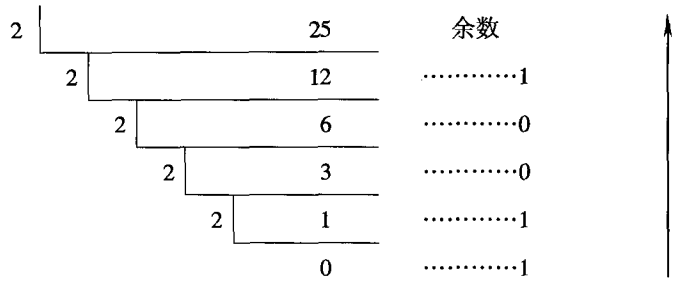
1. 十进制数转换为二进制数　需对其整数和小数部分分别处理，进行转换。

（1）十进制整数转换为二进制整数的方法。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 1-1 将十进制数 25 转换为二进制数。

解

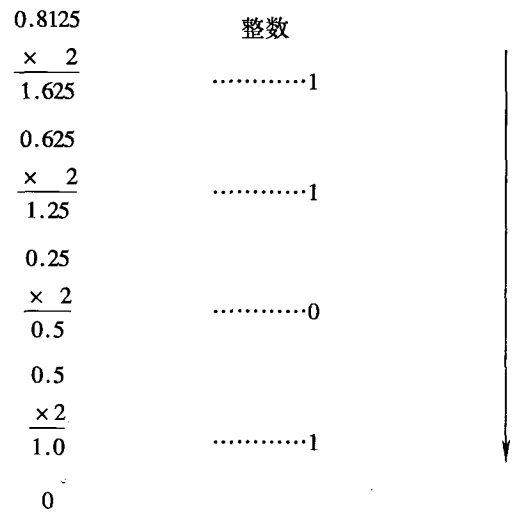


因此 $(25)_{10} = (11001)_2$

（2）十进制小数转换为二进制小数的方法。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 1-2 将十进制数 0.8125 转换为二进制数。

解



因此 $(0.8125)_{10} = (0.1101)_2$

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”的方法，取有限位二进制小数的近似表示。

2. 二进制数转换为十进制数　将二进制数转换为十进制数，只需按位权展开求累加和即可。

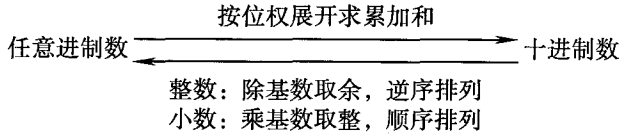
例 1-3 把二进制数 11001.0101 转换为十进制数。

解

$$11001.0101 = 1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}$$
$$= 16+8+0+0+1+0+0.25+0+0.0625$$
$$= (25.3125)_{10}$$

因此 $(11001.0101)_2 = (25.3125)_{10}$

任意进制数与十进制数转换的一般方法如下所示：



说明：基数为相应进位计数制数字符号的个数。

3. 二进制数与十六进制数的相互转换　二进制数转换成十六进制数比较容易，具体方法如下：

1）把二进制数以小数点为界向左、向右每 4 位分成 1 组，不足 4 位的用 0 补齐。

2）把每组 4 位的二进制数转换成 1 位的十六进制数。

3）按从左到右的次序写出转换结果。

例 1-4　把二进制数 10110011.0101111 转换成十六进制数。

解　分组：1011，0011，0101，1110

转换：　B　3　.　5　E

因此 $(10110011.0101111)_2 = (B3.5E)_{16}$

十六进制数转换成二进制数的方法更简单，只需从左到右把每位十六进制数写成相应的 4 位二进制数，不足 4 位则左补 0 凑齐，并把结果写在一起即可。

例 1-5　把十六进制数 3BD.A5 转换成二进制数。

解　$(3BD.A5)_{16}$ = 3　B　D　.　A　5

0011　1011　1101　1010　0101

$(3BD.A5)_{16} = (1110111101.10100101)_2$　（已去掉最左边没有意义的 0）

表 1-1 列出了十进制数 0～15 在二进制、八进制和十六进制下的对应值。为了加快数制转换的速度，应将这张表中的内容熟记于心。

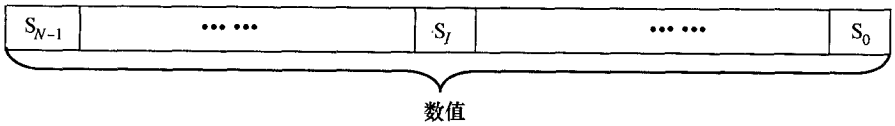
表 1-1　十进制数 0～15 在各种数制下的对应值

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0000	00	0	8	1000	10	8
1	0001	01	1	9	1001	11	9
2	0010	02	2	10	1010	12	A
3	0011	03	3	11	1011	13	B
4	0100	04	4	12	1100	14	C
5	0101	05	5	13	1101	15	D
6	0110	06	6	14	1110	16	E
7	0111	07	7	15	1111	17	F

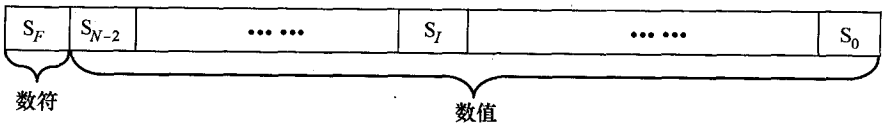
4. 数的书写方法　人们喜欢用十进制，而计算机仅识别二进制。十六进制、八进制则是二进制的缩写，看到了十六进制、八进制，就等于看到了二进制。所以计算机中数的书写方法经常使用的进制有二进制（扩展名 B，BINARY）、八进制（扩展名 O，OCTAL 或 Q，因 O 与 0 容易混淆，所以一般用 Q）、十进制（扩展名 D，DECIMAL，或者不要扩展名）和十六进制（扩展名 H，HEX）。例如，1010B 表示二进制数；5703Q 表示八进制数；3A0BH 表示十六进制数；2048D 表示十进制数。

1.2.2　机器数

计算机中的数据简称为机器数，一个完整的机器数应能描述无符号数和符号数。对于一个字长为 N 位的机器数而言，若表示无符号数时，其 N 位应全部用于表示数值。如



若表示符号数时，其最高位用于表示数的符号（用 0 表示正数，用 1 表示负数，这样的处理称为数字符号的数字化表示），其余的 $N-1$ 位用于表示数值。如



无论是无符号数，还是符号数，都是汇编语言能够直接处理的两种数据，存放在计算机内部的数据是否带有符号，是人为看待的问题，而不是数据本身所具有的属性。

对于符号数，又有不同的编码方式，在计算机中通常采用两种编码表示：原码、补码。

1. 原码　最高位为符号位（正数用 0 表示，负数用 1 表示），其他位为数值位。

例 1-6　X = +45 = +00101101B　　$[X]_{原}$ = 00101101B

X = −45 = −00101101B　　$[X]_{原}$ = 10101101B

2. 补码　正数的补码与原码相同，即符号位用 0 表示，数值位不变；负数的补码则是符号位取 1，数值位逐位取反，末位加 1。

例 1-7　X = +45 = +00101101B　　$[X]_{补}$ = 00101101B

X = −45 = −00101101B　　$[X]_{补}$ = 11010011B

这里是把 +45 和 −45 用 8 位二进制数补码表示的，若用 16 位二进制数补码表示，则为

X = +45 = +00000000 00101101B　　$[X]_{补}$ = 00000000 00101101B

X = −45 = −00000000 00101101B　　$[X]_{补}$ = 11111111 11010011B

由此得知，当一个数从位数较少扩展到位数较多时，对于补码表示的数，正数的最高位用符号位 0 扩展，负数的最高位用符号位 1 扩展。

3. 补码运算　对一个补码表示的数按位求反后再在末尾加 1，可以得到与此数相应的符号相反的数的补码，这种运算称为求补运算。

例 1-8　已知$[117]_{补}$ = 01110101B = 75H、$[-117]_{补}$ = 10001011B = 8BH，对$[117]_{补}$作求补运算。

解　[117]补码为 01110101B，按位求反后得 10001010B，末尾加 1 后得

$$10001011B = 8BH$$

此数正是$[-117]_{补} = 8BH$

两个符号数进行加法或减法运算时，符号位可直接参与运算，不需要判断符号，而计算机结果的最高位仍然表示符号，且

补码的加法规则是　　$[X+Y]_{补} = [X]_{补} + [Y]_{补}$

补码的减法规则是　　$[X-Y]_{补} = [X]_{补} + [-Y]_{补}$

其中，$[-Y]_{补} = -[Y]_{补}$，通过对$-[Y]_{补}$求补得到。如果补码运算的符号位向最高位有进位，进位自动丢失，并不会影响运算结果的正确性。

例 1-9　设$[+25]_{补} = 00011001B$　$[-32]_{补} = 11100000B$　$[-7]_{补} = 11111001B$

$[+32]_{补} = 00100000B$　$[+57]_{补} = 00111001B$

求：(1)(+25)+(−32)　(2)(+25)−(−32)

解　(1) 十进制数运算　　二进制补码数运算

$$\begin{array}{r} (+25) \\ +(-32) \\ \hline -7 \end{array} \qquad \begin{array}{r} 00011001B \\ +\ 11100000B \\ \hline 11111001B \end{array}$$

(2) 十进制数运算　　二进制补码数运算

$$\begin{array}{r} (+25) \\ -(-32) \\ \hline +57 \end{array} \qquad \begin{array}{r} 00011001B \\ +\ 00100000B \\ \hline 00111001B \end{array}$$

例 1-10　设$[-25]_{补} = 11100111B$　$[-32]_{补} = 11100000B$　$[+32]_{补} = 00100000B$

$[-57]_{补} = 11000111B$　$[+7]_{补} = 00000111B$

求：(1)(−25)−(+32)　(2)(−25)−(−32)

解　(1) 十进制数运算　　二进制补码数运算

$$\begin{array}{r} (-25) \\ -(+32) \\ \hline -\ 57 \end{array} \qquad \begin{array}{r} 11100111B \\ +\ 11100000B \\ \hline \boxed{1}\ 11000111B \end{array}$$

进位自动丢失

(2) 十进制数运算　　二进制补码数运算

$$\begin{array}{r} (-25) \\ -(-32) \\ \hline +7 \end{array} \qquad \begin{array}{r} 11100111B \\ +\ 00100000B \\ \hline \boxed{1}\ 00000111B \end{array}$$

进位自动丢失

4. 补码数的表示范围　一个 N 位二进制补码数的表示范围是

$$-2^{N-1} \leqslant N \leqslant 2^{N-1} - 1$$

当 $N = 8$ 时数的表示范围是　　$-128 \leqslant N \leqslant 127$

当 $N = 16$ 时数的表示范围是　　$-32768 \leqslant N \leqslant 32767$

如果两个 8 位二进制补码数的运算结果超过 $-128 \leqslant N \leqslant 127$，或者两个 16 位二制补码数的运算结果超过 $-32768 \leqslant N \leqslant 32767$，则称为运算结果溢出。

1.2.3 常用名词术语及字符的表示

1. 常用的名词术语　位、字节、字及字长是计算机常用的名词术语。

(1) 位(bit)。位是指一个二进制位,它是计算机中信息存储的最小单位。

(2) 字节(Byte)。字节指相邻的 8 个二进制位。通常存储器以字节为单位存储信息。

(3) 字(Word)及字长。字是计算机内部进行数据传递、数据处理的基本单元。通常它与计算机内部的寄存器、运算装置、总线宽度相一致。一个字所包含的二进制位数称为字长。常见的微型计算机的字长有 8 bit、16 bit、32 bit 和 64 bit。在 8086/8088 系统中定义一个字的字长为 16 bit。

2. ASCII 码　计算机在处理信息时,有时需要处理字符或字符串,例如从键盘输入的信息或打印机打印的信息都是以字符方式输入、输出的。因此,计算机必须能用二进制数表示字符。

计算机中最常用的字符编码是美国信息交换标准代码 ASCII(American Standard Code For Information Interchange)。ASCII 码用 7 位二进制数表示字符编码。表 1-2 列出了字符的 ASCII 码值。

例 1-11　写出字符 A、a、0 及 9 的 ASCII 码值。

解　通过查表 1-2,字符 A、a、0 及 9 的 ASCII 码值依次为 41H、61H、30H、39H。

表 1-2　ASCII(美国信息交换标准代码)码值表

位 654 / 3210	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	、	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	、	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	CS	—	=	M	]	m	}
1110	SO	RS	.	>	N		n	~
1111	SI	US	/	?	O	-	o	删除

3. BCD 码　虽然二进制数实现容易，但书写阅读不方便，不符合人们的使用习惯，所以在计算机输入，输出时通常还是采用十进制来表示数，这就需要实现十进制与二进制间的转换。为了转换方便，常采用二进制编码来表示十进制数，这种编码方式称为二-十进制编

码，简称为 BCD（Binary Coded Decimal）码。

BCD 码用 4 位二进制数表示 1 位十进制数。表示的方法有多种，常用的是 8421BCD 码，它的编码规律见表 1-3。

表 1-3　BCD 码值表

十进制数字	BCD 码	十进制数字	BCD 码
0	0000	5	0000
1	0001	6	0001
2	0010	7	0010
3	0011	8	0011
4	0100	9	0100

例 1-12　写出十进制数 314.78 所对应的 BCD 码值。

解　314.78 所对应的 BCD 码值为 0011 0001 0100.0111 1000。

1.2.4　8086/8088 支持的数据类型

8088 机器指令中可以直接处理的数据类型只有 3 种，分别是 8bit 二进制数的字节型、16bit 的字型和 32bit 的双字型。但是根据其具体含义及写法的不同，在汇编语言中又有多种变化。表 1-4 列出了几种常用的数据类型及其表示范围。

表 1-4　各种数据类型的有效范围

数据类型	有效范围	数据类型	有效范围
字节型无符号数	0 ~ 255	字节型带符号数	- 128 ~ 127
字型无符号数	0 ~ 65535	字型带符号数	- 32768 ~ 32767
双字型无符号数	0 ~ 4294967295	双字型带符号数	- 2147483648 ~ 2147483647

在 8088 汇编语言中，数据的书写方法比较丰富，不同的书写形式在计算机内部可能有相同的存储结果，而存储器中的同一个数据也可能因为使用的方法不同而有不同的外部表示形式。所以，内部存储形式相同的数据之间不存在类型转换问题。

例 1-13　把下面几种书写形式的数据转换成字节型内部存储形式。

0E3H，227，-29，-11101B

解　经转换，内部形式完全相同，都是 11100011B。

汇编语言中数据还有一种字符形式的写法，如‘A’、‘0’等。在字符的两边加单引号，其含义是以该字符的 ASCII 值作为数据，所以‘A’相当于 41H，而‘0’相当于 30H。

例 1-14　设存储器中有一个字节型数据 01100010B，试说明该数据在不同的使用环境下几种可能的含义。

解　作为字符的 ASCII 值，该数表示字符‘B’。

作为无符号数，该数表示 98。

作为带符号数，该数表示 + 98。

数据是计算机处理的对象，数据的各种书写方法为编程提供了便利，而同一数据可以具有不同的含义又给学习汇编语言造成一定的困难。

1.3 8086/8088 微型计算机组成结构

1.3.1 计算机系统的组成

计算机系统由硬件系统（Hardware）和软件系统（Software）两部分组成。

1. 硬件系统　计算机硬件系统由运算器、控制器、存储器、I/O 设备 5 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。

微型计算机的硬件系统由主机和外部设备构成。主机由微处理器（Microprocessor）、存储器和 I/O 接口 3 个部分组成，各部分通过系统总线连接在一起。主机的结构如图 1-2 所示。

（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。微处理器即为微型的 CPU，其被微缩在 1 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，即从存储器中取出指令，进行译码，执行相应的运算和操作，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。

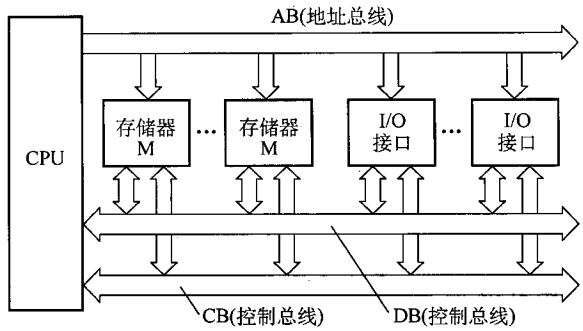


图 1-2　主机的基本结构

2）存储器。存储器是微型计算机的记忆部件。所要处理的数据以及进行这一处理过程的所有命令的集合——程序，都是事先存放在存储器中的，然后让计算机去自动执行的，这就是冯·诺依曼先生提出的著名的“存储程序”思想，这也是现代计算机能自动进行运行的根本保证。主机中的存储器称为内存储器，简称内存。

3）I/O 接口。I/O 接口是主机与外部设备之间通信的桥梁。I/O 接口的任务是处理主机与外部设备之间的数据传送、把外部设备的状态传入主机、接收主机发出的各种控制信号、控制外部设备执行操作。

4）系统总线。系统总线把微处理器、存储器、I/O 接口和外部设备连接起来，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 I/O 设备的控制命令和 I/O 设备对微处理器的请求信号。

（2）外部设备。外部设备一般包括 I/O 设备和大容量存储器两类设备。I/O 设备是指负责计算机进行通信的外部设备，如键盘、显示器和打印机等。大容量存储器则是指可存储大量信息的外部存储器，如磁盘、磁带和光盘等，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。

2. 软件系统　微型计算机的软件系统由系统软件和应用软件 2 大部分组成。