

MODEL SMALL  
.586  
.STACK 64  
.DATA

Intel 80X86/Pentium

# 汇编语言程序设计

张雪兰 编著



北京理工大学出版社

# Intel 80X86/Pentium 汇编语言程序设计

张雪兰 编著

北京理工大学出版社

## 内 容 简 介

本书以当今广为流行的 Intel 80X86 及 Pentium 为 CPU 的 IBM PC 机为背景,深入讨论了汇编语言程序设计技术。全书共分十章,包括:计算机基础知识、微型计算机硬件系统简介、IBM PC 机的寻址方式与指令系统、汇编语言程序组织、程序基本结构及其程序设计、子程序设计、高级汇编语言技术、I/O 程序设计、系统功能及中断调用的进一步讨论、汇编语言高级编程技巧等。内容由浅入深、循序渐进、实例丰富,尤其后三章的程序都是有一定难度的实际应用,很有参考价值。而且每章后均附有习题,以便读者检查及巩固所学知识。

本书既可作为计算机及其相关专业的本科生教材,也可供从事计算机开发及其研究的工程技术人员参考。

### 图书在版编目(CIP)数据

Intel 80X86/Pentium 汇编语言程序设计/张雪兰编著. —北京:北京理工大学出版社, 2001.6

ISBN 7-81045-770-5

I. J… II. 张… III. 汇编语言-程序设计 IV. TP313

中国版本图书馆 CIP 数据核字(2001)第 022903 号

责任印制:母长新 责任校对:陈玉梅

北京理工大学出版社出版发行  
(北京市海淀区中关村南大街 5 号)  
邮政编码 100081 电话(010)68912824  
各地新华书店经售  
北京市房山先锋印刷厂印刷

\*

787 毫米 × 1092 毫米 1/16 开本 20.25 印张 489 千字  
2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷  
印数:1—4000 册 定价:28.00 元

---

(本书如有印装错误,我社负责调换)

# 前 言

汇编语言是当今能够最大限度地发挥计算机硬件特性，并且满足苛刻的实时处理要求的最有效的语言。使用汇编语言编写高性能的系统或应用程序对计算机专业人员极具挑战性。因为无论 CPU 的速度如何加快，内存容量如何增长，但空间、速度及性能对于有竞争力的软件来讲永远都是十分重要的。汇编语言程序设计是高校计算机硬件、软件及应用专业本科生必修的核心课程之一，它不仅是计算机操作系统和计算机组成原理等其它核心课程的必要先修课，而且对于学生了解计算机内部运行机制、训练学生掌握程序设计技术等都有很重要的作用。

由于汇编语言本身的特点，学习汇编语言离不开实际的计算机系统，因此选择一个典型的系统，不仅对于组织教材是重要的，而且对于理论联系实际地进行教学也是十分必要的。本书选择了当今广为流行的以 Intel 80X86 及 Pentium 为 CPU 的微型计算机系统作为硬件平台，深入讨论了计算机基础知识、微型计算机硬件系统简介、IBM PC 机的寻址方式与指令系统、汇编语言程序组织、程序基本结构及其程序设计、子程序设计、高级汇编语言技术、I/O 程序设计、系统功能及中断调用的进一步讨论、汇编语言高级编程技巧等内容，并在附录中给出了 DOS 系统功能调用、BIOS 中断调用、DEBUG 调试工具的参考资料，以方便读者查阅。书中提供了大量程序实例，尤其后三章的程序都是有一定难度的实际应用，例如中断处理程序设计、磁盘文件存取技术、鼠标及键盘事件处理、图形显示、使用 DOS EXEC 功能执行程序、TSR 程序设计、保护模式程序设计等，这些例子均能在 PII 机上正确运行，其汇编环境是 MASM5.1 或 MASM6.11。另外，每章后均附有习题，便于读者检查及巩固所学知识。

在本书编写过程中，不仅总结了作者多年的教学经验，还得到了许多同志的热情帮助，特别是承蒙清华大学计算机系温冬婵副教授在百忙之中审阅了全稿，在此一并表示衷心地感谢。

虽竭尽全力，但难免有不妥之处，敬请读者批评指正。

作 者  
于北京理工大学

# 目 录

<b>第一章 基础知识</b> .....	1
1.1 计数制与不同数制数之间的转换.....	1
1.2 二进制数算术和逻辑运算.....	3
1.3 数和字符在计算机中的表示方法.....	4
习题一.....	7
<b>第二章 微型计算机硬件系统简介</b> .....	9
2.1 微型计算机系统简介.....	9
2.2 汇编语言概述.....	10
2.2.1 程序设计语言概述.....	10
2.2.2 汇编语言概述.....	11
2.3 Intel 公司微处理器简介.....	14
2.4 程序可见寄存器组.....	15
2.5 存储器.....	19
2.5.1 基本概念.....	19
2.5.2 IBM PC 机存储器映像.....	20
2.5.3 存储器分段管理.....	20
2.5.4 实模式存储器寻址.....	20
2.5.5 保护模式存储器寻址.....	22
2.6 外部设备.....	22
习题二.....	22
<b>第三章 IBM PC 机的寻址方式与指令系统</b> .....	24
3.1 与数据有关的寻址方式.....	24
3.2 与转移地址有关的寻址方式.....	29
3.3 数据传送指令.....	32
3.4 算术运算指令.....	41
3.4.1 二进制算术运算指令.....	41
3.4.2 十进制算术运算指令.....	50
3.5 逻辑指令.....	54
3.6 程序控制指令.....	58
3.7 处理机控制指令.....	68
3.8 串操作指令.....	69
3.9 条件字节设置指令.....	73
习题三.....	74
<b>第四章 汇编语言程序组织</b> .....	78
4.1 汇编语言语句格式.....	78

4.2 汇编语言源程序结构 .....	78
4.3 常用伪指令 .....	84
4.4 结构伪指令 .....	88
4.5 汇编语言操作符 .....	92
4.6 汇编语言程序上机过程 .....	95
4.7 数据的输入输出 .....	103
4.8 微处理器伪指令 .....	115
4.9 简化段定义 .....	116
习题四 .....	119
<b>第五章 程序基本结构及其程序设计 .....</b>	<b>122</b>
5.1 顺序结构及简单程序设计 .....	122
5.2 分支结构及程序实现 .....	123
5.3 循环结构及程序实现 .....	130
习题五 .....	137
<b>第六章 子程序设计 .....</b>	<b>139</b>
6.1 子程序结构及设计方法 .....	139
6.2 子程序参数传递 .....	141
6.3 嵌套与递归子程序 .....	154
习题六 .....	159
<b>第七章 高级汇编语言技术 .....</b>	<b>160</b>
7.1 宏指令 .....	160
7.2 宏指令库 .....	174
7.3 重复伪指令 .....	178
7.4 条件伪指令 .....	183
习题七 .....	187
<b>第八章 I/O 程序设计 .....</b>	<b>189</b>
8.1 概述 .....	189
8.2 程序查询方式 .....	189
8.3 中断传送方式 .....	195
8.4 DMA 方式简介 .....	208
习题八 .....	209
<b>第九章 系统功能及中断调用的进一步讨论 .....</b>	<b>211</b>
9.1 磁盘文件存取技术 .....	211
9.1.1 DOS 文件管理功能简介 .....	211
9.1.2 句柄式文件管理功能 .....	211
9.1.3 字符设备句柄式输入 / 输出 .....	222
9.2 鼠标及键盘事件处理 .....	223
9.3 图形显示 .....	230
习题九 .....	235

<b>第十章 汇编语言高级编程技巧</b> .....	236
10.1 模块化程序设计.....	236
10.1.1 模块通信.....	236
10.1.2 模块连接.....	242
10.1.3 模块组织建议.....	246
10.2 汇编语言程序与高级语言程序的连接.....	248
10.2.1 C 语言程序中直接嵌入汇编代码.....	248
10.2.2 在 C 程序中直接调用汇编子程序.....	250
10.2.3 编写汇编子程序注意事项.....	253
10.2.4 使用 C 和汇编语言混合编程的编译及连接.....	255
10.2.5 混合编程实例.....	256
10.2.6 汇编语言程序调用 C 函数.....	257
10.3 使用 DOS EXEC 功能执行程序.....	257
10.3.1 DOS 的内存分配与释放功能简介.....	257
10.3.2 使用 DOS EXEC 功能加载并执行程序.....	258
10.4 TSR 程序设计.....	264
10.5 保护模式程序设计.....	270
10.5.1 保护模式基础知识.....	270
10.5.2 存储器分页机制.....	274
10.5.3 保护模式程序示例.....	276
习题十.....	286
<b>附录 A DOS 系统功能调用</b> .....	288
<b>附录 B BIOS 中断调用</b> .....	300
INT 10H (显示器 I/O).....	300
INT 11H (测试设备类型).....	301
INT 12H (测试内存大小).....	302
INT 13H (磁盘 I/O).....	302
INT 14H (串行通讯中断).....	303
INT 15H (系统服务).....	303
INT 16H (键盘 I/O).....	304
INT 17H (打印机中断).....	304
INT 33H (鼠标中断).....	304
<b>附录 C DEBUG 调试工具</b> .....	311
<b>主要参考文献</b> .....	314

# 第一章 基础知识

## 1.1 计数制与不同数制数之间的转换

### 一、什么是计数制

计数制是指用一组固定的数字符号和统一的规则表示数的方法。讨论计数制要涉及到两个基本问题：基数和权。在计数制中，每个数位（数字位置）所用到的不同数字的个数叫做基数。例如人们习惯使用的十进制，是采用 0~9 这十个数字表示的，它的基数是 10。在一个数中，数字在不同的数位所代表的数值是不同的，每个数字所表示的数值等于它本身乘以与所在数位有关的常数，把这个常数叫做位权，简称权。例如十进制数个位的位权是 1，十位的位权是 10，百位的位权是 100，千位的位权是 1000……。相邻两位权的比值就等于基数。一个数的数值大小就等于它的各位数码乘以相应位权的总和。例如十进制  $698 = 6 \times 100 + 9 \times 10 + 8 \times 1$ 。

在日常生活中，人们经常使用十进制、十二进制、六十进制等。

### 二、计算机中常用的计数制

在计算机中，常使用的有二进制、八进制、十进制、十六进制，见表 1-1。

#### 1. 十进制

由十个数字 0~9 组成，10 为基数，小数点左边从右至左其各位的位权依次是： $10^0$ 、 $10^1$ 、 $10^2$ 、 $10^3$ ……小数点右边从左至右其各位的位权依次是： $10^{-1}$ 、 $10^{-2}$ 、 $10^{-3}$ ……。例如十进制数 678.5 可以表示为：

$$678.5 = 6 \times 10^2 + 7 \times 10^1 + 8 \times 10^0 + 5 \times 10^{-1}$$

#### 2. 二进制

由数字 0 和 1 组成，2 为基数，小数点左边从右至左其各位的位权依次是： $2^0$ 、 $2^1$ 、 $2^2$ 、 $2^3$ ……小数点右边从左至右其各位的位权依次是： $2^{-1}$ 、 $2^{-2}$ 、 $2^{-3}$ ……。

#### 3. 八进制

由于用二进制数书写长而且不易阅读，因此在计算机中还经常使用八进制和十六进制。八进制数由八个数字 0~7 组成，8 为基数，小数点左边从右至左其各位的位权依次是： $8^0$ 、 $8^1$ 、 $8^2$ 、 $8^3$ ……小数点右边从左至右其各位的位权依次是： $8^{-1}$ 、 $8^{-2}$ 、 $8^{-3}$ ……。

#### 4. 十六进制

由十六个数字符号（数字 0~9、符号 A、B、C、D、E、F）组成，16 为基数，小数点左边从右至左其各位的位权依次是： $16^0$ 、 $16^1$ 、 $16^2$ 、 $16^3$ ……小数点右边从左至右其各位的位权依次是： $16^{-1}$ 、 $16^{-2}$ 、 $16^{-3}$ ……。



表 1-1 计算机中常用计数制的基数和数码表

进位制	基数	数码
二进制数	2	0,1
八进制数	8	0,1,2,3,4,5,6,7
十进制数	10	0,1,2,3,4,5,6,7,8,9
十六进制数	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

### 三、不同数制间的转换

#### 1. 把其它数制的数转换成十进制数

把其它数制的数转换成十进制数的方法是把其它数制数按各数位的权值展开求和。

##### (1) 二进制数转换成十进制数

$$(1110.1)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} = (14.5)_{10}$$

即二进制数 1110.1 等于十进制数 14.5。

##### (2) 八进制数转换成十进制数

$$(4567)_8 = 4 \times 8^3 + 5 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 = (2423)_{10}$$

即八进制数 4567 等于十进制数 2423。

##### (3) 十六进制数转换成十进制数

$$(56AF)_{16} = 5 \times 16^3 + 6 \times 16^2 + A \times 16^1 + F \times 16^0 = (22191)_{10}$$

即十六进制数 56AF 等于十进制数 22191。

#### 2. 把十进制数转换成其它数制的数

把十进制数转换成其它数制数的方法较多，通常采用的方法是：整数部分转换采用除基数取余法，所得的第一个余数为最低位，最后一个余数为最高位。小数部分转换采用乘基数取整法，所得的第一个整数为最高位，最后一个整数为最低位。

##### (1) 十进制数转换成二进制数

整数转换采用除二取余法，小数转换采用乘二取整法。

例如，把十进制数 14.125 转换成二进制数，需要对整数和小数部分分别转换，然后把它们拼接到一起，见图 1-1。

对整数部分的转换	对小数部分的转换
$  \begin{array}{r}  2 \overline{) 14} \quad \text{余 } 0 \\  \underline{2} \phantom{0} \\  2 \overline{) 7} \quad \text{余 } 1 \\  \underline{2} \phantom{0} \\  2 \overline{) 3} \quad \text{余 } 1 \\  \underline{2} \phantom{0} \\  2 \overline{) 1} \quad \text{余 } 1 \\  \underline{2} \phantom{0} \\  2 \overline{) 0} \quad \text{余 } 0 \\  \phantom{2} \phantom{0} \\  0  \end{array}  $	$  \begin{array}{r}  0.125 \\  \times \phantom{00} 2 \\  \hline  0.250 \quad \text{整数为 } 0 \\  \times \phantom{00} 2 \\  \hline  0.500 \quad \text{整数为 } 0 \\  \times \phantom{00} 2 \\  \hline  1.000 \quad \text{整数为 } 1  \end{array}  $

图 1-1 十进制数 14.125 转换成二进制数

把整数和小数部分加起来得到转换结果，即  $(14.125)_{10} = (1110.001)_2$

注意，小数部分有时通过有限次乘法得不到 1.0 的结果，则应按照精度要求截取适当的位数。

同理，把十进制数转换成八进制数，其整数转换采用除八取余法，小数转换采用乘八取整法。把十进制数转换成十六进制数，其整数转换采用除十六取余法，小数转换采用乘十六取整法。

### 3. 其它数制数之间的转换

#### (1) 二进制数与八进制数之间的转换

把二进制数转换成八进制数，方法是以小数点为界，整数部分向左、小数部分向右每三位二进制数组成一位八进制数，不足三位者以 0 补齐（整数部分左补 0，小数部分右补 0）。

要把八进制数转换成二进制数，只需把每位八进制数用三位二进制数表示即可。例如，把  $(10110.1)_2$  转换成八进制数：

$$(10110.1)_2 = (010110.100)_2 = (26.4)_8$$

又如，把  $(32.5)_8$  转换成二进制数

$$(32.5)_8 = (011010.101)_2$$

#### (2) 二进制数与十六进制数之间的转换

把二进制数转换成十六进制数，方法是以小数点为界，整数部分向左、小数部分向右每四位二进制数组成一位十六进制数，不足四位者以 0 补齐（整数部分左补 0，小数部分右补 0）。

要把十六进制数转换成二进制数，只需把每位十六进制数用四位二进制数表示即可。例如，把  $(10110.1)_2$  转换成十六进制数：

$$(10110.1)_2 = (00010110.1000)_2 = (16.8)_{16}$$

又如，把  $(3A.5)_{16}$  转换成二进制数：

$$(3A.5)_{16} = (00111010.0101)_2$$

## 1.2 二进制数算术和逻辑运算

### 一、二进制数算术运算

二进制数算术运算非常简单，其运算规则如下：

#### 1. 加法运算规则

二进制加法运算规则是逢二进一。即：

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

例.  $1101+1011=11000$

#### 2. 减法运算规则

二进制减法运算规则是借 1 当 2。即：

$$0-0=0 \quad 1-0=1 \quad 1-1=0 \quad 0-1=1 \quad (\text{向高位借 } 1)$$

例.  $1101-1011=0010$

#### 3. 乘法运算规则

二进制乘法运算规则是 0 乘以任何数得 0，1 乘以任何数得该数。即：

$$0 \times 0=0 \quad 0 \times 1=0 \quad 1 \times 0=0 \quad 1 \times 1=1$$

例.  $1101 \times 101=1000001$

#### 4. 除法运算规则

二进制除法运算规则是 0 除以 1 得 0, 1 除以 1 得 1, 0 做除数无意义。即:

$$0 \div 1 = 0$$

$$1 \div 1 = 1$$

例.  $1000001 \div 1101 = 101$

## 二、二进制数逻辑运算

计算机所能实现的另一基本运算是逻辑运算, 逻辑运算的结果只能是“真”或“假”, 在计算机中用“1”或“0”表示。逻辑运算是按位进行的, 基本的逻辑运算包括或、与、非运算。

### 1. 逻辑或

逻辑或运算规则是 1 和任何数相或得 1, 只有 0 或 0 得 0。其运算符号为  $\vee$  或 +。

$$0 \vee 0 = 0$$

$$0 \vee 1 = 1$$

$$1 \vee 0 = 1$$

$$1 \vee 1 = 1$$

或者表示为:  $0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 1$

例.  $1100 \vee 1001 = 1101$

### 2. 逻辑与

逻辑与运算规则是 0 和任何数相与得 0, 1 和任何数相与该数值不变。其运算符号为  $\wedge$  或  $\times$ 。

$$0 \wedge 0 = 0$$

$$0 \wedge 1 = 0$$

$$1 \wedge 0 = 0$$

$$1 \wedge 1 = 1$$

或者表示为:  $0 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 0 = 0$

$1 \times 1 = 1$

例.  $1100 \wedge 1001 = 1000$

### 3. 逻辑非

逻辑非运算是单目运算, 其运算规则是 1 非为 0, 0 非为 1。其运算符号为  $\neg$ , 或者在逻辑变量上方加一横线。

$$\neg 0 = 1$$

$$\neg 1 = 0$$

或者表示为:  $\bar{0} = 1$

$\bar{1} = 0$

例.  $\neg 1011 = 0100$

### 4. 逻辑异或

逻辑异或运算规则是 0 和任何数相异或该数值不变, 1 和任何数相异或该数值变反。其运算符号为:  $\nabla$

$$0 \nabla 0 = 0$$

$$0 \nabla 1 = 1$$

$$1 \nabla 0 = 1$$

$$1 \nabla 1 = 0$$

例.  $1101 \nabla 001 = 0100$

## 1.3 数和字符在计算机中的表示方法

计算机只能识别二进制数, 因此数字、字母、符号等信息在计算机中都是用二进制位串形式表示的。

## 一、数字在计算机中的表示

计算机中的数字用二进制数形式表示，数字可以分为带符号数和无符号数。

### 1. 带符号数

在计算机中数和符号都用二进制形式表示，一般用最高有效位表示数的符号，正数用 0 表示，负数用 1 表示，把一个数及其符号在机器中的这种表示称为机器数。机器数可以用不同的编码来表示，常用的有原码、反码和补码。由于 IBM PC 机和多数机器中的带符号数都采用的是补码表示形式，因此这里只介绍补码表示法。

#### (1) 补码的定义

设有  $n$  位二进制数表示带符号数的补码，则对于数  $X$  的补码记作  $[X]_{\text{补}}$ ， $[X]_{\text{补}}$  的定义为：

$$[X]_{\text{补}} = \begin{cases} X & 0 \leq X < 2^{n-1} \\ 2^n - |X| = 2^n + X & -2^{n-1} \leq X < 0 \end{cases}$$

从以上定义可以看出，0 的补码表示只有一种。设有 8 位二进制数表示带符号数的补码，则  $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000$ 。对于符号位为 1 后续其它位全为 0 的情况，表示为最小负数。例如对于 8 位二进制数来讲，10000000 这个数在补码表示中被定义为 -128，于是 8 位二进制数所能表示的带符号数的范围是 -128 ~ +127，而 16 位二进制数所能表示的带符号数的范围是 -32768 ~ +32767。

例. 设  $A = +65$ ， $B = -65$ ，则按补码的定义，有：

$$[A]_{\text{补}} = 01000001$$

$$[B]_{\text{补}} = 2^n + B = 256 + (-65) = 10000000 - 01000001 = 10111111$$

#### (2) 快速求负数补码的方法

按照以上定义求负数的补码比较麻烦，我们可以用一种较简单的方法对负数求补，方法是先写出该负数的相反数（正数）的二进制形式，再对其按位求反（1 变为 0，0 变为 1），最后在末位加 1。实际上就是把其正数连同符号位自左至右按位求反，直至遇到最右边的一个 1 及其后的所有 0 保持不变即可。

例.  $+65 = 01000001$ ，按照上述方法，则：

$$[-65]_{\text{补}} = 10111111$$

而  $+46 = 00101110$ ，则：

$$[-46]_{\text{补}} = 11010010$$

### 2. 无符号整数

从以上叙述可以看出，机器数的表示方法就是把数的符号数码化了，它既能表示数的绝对值又能表示数的符号。如果把所有有效位都用于表示数的绝对值而不保留符号位（即所有数位都是有效数字），则用这种方法表示的数叫做无符号数。计算机中有时需要无符号数形式，例如地址用无符号整数表示。一个 8 位、16 位、32 位的二进制数可表示的最大无符号整数为十进制数的 255、65535（64K）、4294967295（4G）。

## 二、字符编码

计算机只能识别二进制数，因此计算机中的数字、字母、符号、控制字符等也必须用二

进制进行编码，编码方法有多种，微型机中普遍采用的是 ASCII 码（美国标准信息交换码），ASCII 码现已被国际标准化组织（ISO）接收为国际标准，称为 ISO-646。ASCII 码有 7 位版本和 8 位版本两种，国际上通用的 ASCII 码是 7 位版本。它包含 10 个阿拉伯数字、52 个英文大小写字母、32 个标点符号和运算符、34 个控制码，共 128 个字符，所以可用 7 位二进制数表示。7 位 ASCII 码见表 1-2。

表 1-2 ASCII 码表

十进制值	十六进制值	字 符							
000~007	00~07	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
008~015	08~0F	BS	HT	LF	VT	FF	CR	SO	SI
016~023	10~17	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
024~031	18~1F	CAN	EM	SUB	ESC	FS	GS	RS	US
032~039	20~27	SP	!	"	#	\$	%	&	'
040~047	28~2F	(	)	*	+	,	-	.	/
048~055	30~37	0	1	2	3	4	5	6	7
056~063	38~3F	8	9	:	;	<	=	>	?
064~071	40~47	@	A	B	C	D	E	F	G
072~079	48~4F	H	I	J	K	L	M	N	O
080~087	50~57	P	Q	R	S	T	U	V	W
088~095	58~5F	X	Y	Z	[	\	]	^	_
096~103	60~67	`	a	b	c	d	e	f	g
104~111	68~6F	h	i	j	k	l	m	n	o
112~119	70~77	p	q	r	s	t	u	v	w
120~127	78~7F	x	y	z	{		}	~	DEL

表中 NUL、BEL、LF、FF、CR、DEL 等是控制字符，NUL 表示空，BEL 是告警符，BS 是退格符，LF 是换行符，FF 是换页符，CR 是回车符，SP 是空格符，DEL 是删除符。

要确定一个数字、字母、符号或控制字符的 ASCII 码，可在表中先找出它的位置，然后找出它所对应的十进制值或十六进制值。

例如小写字母“a”的 ASCII 码其十进制值是 97，十六进制值是 61H（H 表示十六进制数），转换成二进制值是 01100001B（B 表示二进制数）。大写字母 A~Z 的 ASCII 码是 41H~5AH，小写字母 a~z 的 ASCII 码是 61H~7AH。数字 0~9 的 ASCII 码值用十进制数表示是 48~57，用十六进制数表示是 30H~39H。在 IBM PC 机中要输出的十进制数必须用 ASCII 码表示，例如要想在显示器上显示 9 应该给出 39H。

### 三、BCD 码

虽然二进制运算规则简单、计算机处理容易实现，但二进制数不直观、书写容易出错，加之人们习惯用十进制数形式，因此计算机中的数字有时也用十进制形式表示。这里介绍常用的表示十进制数的 BCD 编码。

BCD(Binary Code Decimal)码叫做二进制编码的十进制数，也称二-十进制数。一位十进制数可以用四位二进制数表示，其表示方法有多种，通常用的是 8421 BCD 码，即 4 位二进制数自左至右其权分别为 8、4、2、1。8421 BCD 码的十个十进制数字 0~9 分别用四位二进制数的前十种表示，即用二进制数的 0000~1001 分别表示 8421 BCD 码的 0~9，详见表 1-3。

BCD 码有压缩 BCD 码和非压缩 BCD 码两种形式。

表 1-3 BCD 码表

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

## (1) 压缩 BCD 码

每位压缩 BCD 码占用四位 bit (二进制位), 一个字节 (8 位 bit) 可以存放两位压缩 BCD 码。

例. 把十进制数 3456 转换成压缩 BCD 码

$$(3456)_{10} = (00110100\ 01010110)_{\text{压缩 BCD}}$$

## (2) 非压缩 BCD 码

每位非压缩 BCD 码占用一个字节 (8 位 bit), 其中高四位为 0, 低四位为 BCD 码。

例. 把十进制数 3456 转换成非压缩 BCD 码

$$(3456)_{10} = (00000011\ 00000100\ 00000101\ 00000110)_{\text{非压缩 BCD}}$$

## 习 题 一

1. 把下列十进制数分别转换成二进制数和十六进制数。

- (1) 127                      (2) 238                      (3) 4096                      (4) 30                      (5) 12

2. 把下列二进制数分别转换成十进制数和十六进制数。

- (1) 00111110              (2) 00010101              (3) 10101001  
(4) 11010110              (5) 11111111

3. 把下列十六进制数分别转换成二进制数和十进制数。

- (1) 456A              (2) FB              (3) F13              (4) BA90              (5) FFFF

4. 完成下列二进制数的算术运算。

- (1) 00111110 + 10101001  
(2) 10101010 - 00010101  
(3) 01010011 + 10011101  
(4) 00011101 - 00000111  
(5) 00111101 + 10000111  
(6) 11111000 - 11100000  
(7) 101 × 110  
(8) 110000 ÷ 110  
(9) 1001 × 101  
(10) 10100 ÷ 1001

5. 完成下列二进制数的逻辑运算。

- (1) 00111110 ∧ 10101001  
(2) 10101010 ∨ 00010101  
(3) 10111010 ∧ 11111011  
(4) 11000011 ∨ 11011011  
(5) ¬ 01010011

(6) 11111111 $\nabla$ 11111111

(7) 10101010 $\nabla$ 00010101

(8) 10101010 $\nabla$ 01010101

6. 以下均为十六进制数，试说明当把它们分别看做无符号数或用补码形式表示的带符号数时，其所表示的十进制数是多少？

(1) FFFF            (2) FB            (3) EF13

(4) BA90            (5) 0

7. 以下均为两位十六进制数，试说明当把它们分别看做无符号数或字符的 ASCII 值时，它们所表示的十进制数和字符是什么？

(1) 30            (2) 31            (3) 36            (4) 39            (5) 41

(6) 42            (7) 61            (8) 62            (9) 24            (10) 2B

8. 对于第 4 题的(1)~(6)小题的计算结果，若把它当做是以补码形式表示的二进制数，其十进制值是多少？

9. 把以下十进制数分别以非压缩 BCD 码和压缩 BCD 码两种形式表示。

(1) 99            (2) 35            (3) 86            (4) 39            (5) 1

10. 以十六进制数形式写出以下字符的 ASCII 值。

(1) a            (2) A            (3) d            (4) D            (5) y

(6) Y            (7) 8            (8) 5            (9) z            (10) Z

## 第二章 微型计算机硬件系统简介

由于汇编语言是一种符号化了的面向机器的语言，因此介绍汇编语言编程需要基于一种机器模型，本书以目前广为流行的 IBM PC 及其兼容机为模型，本章介绍与汇编语言编程相关的该类机器的一些硬件知识，这些知识对于编程人员很重要，若不了解它们，学习后续章节会有困难，因此应该认真学习本章。

### 2.1 微型计算机系统简介

一个微型计算机（简称微型机或微机）系统由计算机硬件和计算机软件两大部分组成。

#### 一、计算机硬件

硬件是组成计算机的物理设备的总称，硬件包括 CPU、内存储器、I/O 设备和系统总线。微型机硬件的基本结构如图 2-1 所示。

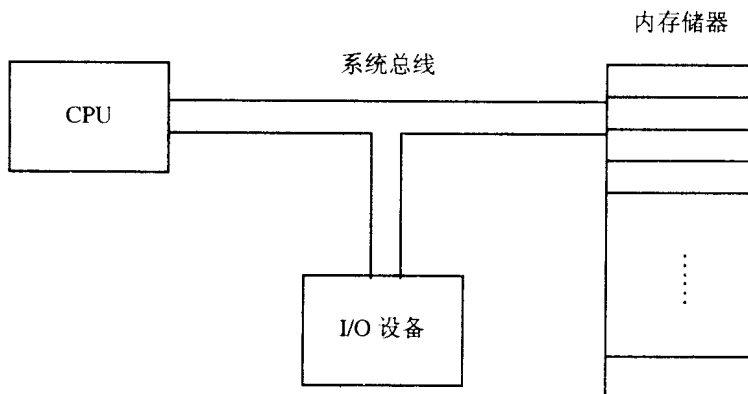


图 2-1 微机硬件基本结构

微型机各部分功能为：

#### (1) CPU

CPU (Central Processing Unit) 是中央处理器的英文缩写，也被称做微处理器。它是采用 LSI 大规模集成电路工艺把运算器和控制器集成在一块芯片上的，具有运算能力和控制功能，IBM PC 机中的 CPU 是 Intel 公司的 80X86 系列。

#### (2) 内存储器

存储器是计算机存储信息的部件，分为内存储器（简称内存或主存）和外存储器（简称外存或辅存）。内存是采用 LSI 工艺制成的半导体存储器，是 CPU 直接访问的存储器，用来存放当前正在使用的程序和数据。内存又分为 RAM（随机存取存储器）和 ROM（只读存储器），微机中系统软件的核心部分常被固化在 ROM 中。



### (3) 外部设备

外部设备包括输入设备、输出设备和外存储器。

① 输入设备：把程序、数据等信息输入到计算机的设备叫做输入设备。微机的输入设备有键盘、鼠标等。

② 输出设备：输出处理结果的设备叫做输出设备。微机的输出设备有显示器、打印机等。

③ 外存储器：相对于内存而言，外存的主要特点是容量大、速度较慢，微机中常用的外存包括软盘、硬盘、光盘等。软盘和硬盘采用磁记录技术存储信息，光盘采用激光技术存储信息。目前常用的软盘有 3 寸双面高密 1.44MB 等，硬盘容量可达几十 GB，一片光盘大约可存储 640MB 信息。

### (4) 系统总线

系统总线是计算机中互连各部件的一组信号线，包括数据总线、地址总线、控制总线，它们负责在 CPU 与内存、外设之间传送数据、地址和控制信息。采用总线结构使得各部件间互连变得十分简单。

## 二、计算机软件

软件包括系统软件和应用软件。

### (1) 系统软件

计算机生产厂家提供给用户的一组程序，这些程序是用户使用计算机所必需的。系统软件包括操作系统、翻译程序、辅助程序等。

① 操作系统：系统软件中的核心软件，是管理计算机软硬件资源、协调各部分有效工作的一组程序集合。

② 翻译程序：把高级语言源程序或汇编语言源程序翻译成机器语言程序的程序。翻译程序包括各种高级语言的编译程序（如 C 编译程序）、解释程序、汇编程序。

③ 辅助程序：为用户提供一些常用的服务性功能的程序。例如编辑程序（具有建立、修改、删除文本等功能的程序）、调试程序（监督和控制程序执行的程序）、连接程序（把经翻译程序翻译成的目标模块和库文件连接到一起形成一个可执行程序的程序）、装入程序（把外存的程序装入内存以便运行的程序）等。

### (2) 应用软件

也称应用程序，为解决各类实际问题而设计的程序。可以是为解决某类问题而设计的通用软件或为解决特定问题而编制的专用程序。

## 2.2 汇编语言概述

### 2.2.1 程序设计语言概述

如上节所述，计算机软件就是一些程序。程序需要用程序设计语言编写，程序设计语言包括：

机器语言：直接用二进制代码的机器指令表示的语言。用机器语言书写的程序叫做机器