



高等学校“十一五”规划教材

汇编语言程序设计教程

Huibian Yuyan Chengxu Sheji Jiaocheng

花小朋 刘其明 王一飞 编著

中国矿业大学出版社

China University of Mining and Technology Press

汇编语言程序设计教程

花小鹏 刘其明 王一飞 编 著

中国矿业大学出版社

内 容 提 要

本书主要阐述汇编语言程序设计方法和技术。全书共分三部分:第1章和第2章为基础知识部分;第3章和第4章为编程工具部分,主要内容为8086/8088指令系统与寻址方式,以及包括伪指令在内的汇编语言程序格式;第5章至第9章为编程方法部分,内容包括分支、循环、子程序等基本结构,程序设计的基本方法和技术,宏汇编技术和以中断为主的输入输出程序设计方法,以及BIOS和DOS系统功能调用。在内容组织上,将指令系统中控制类指令全部分散到后续章节中结合相关内容介绍。这样,一方面可使学生更好地掌握相关指令的运用;另一方面也使得学生能够尽早上机编程实践,使得理论教学与实践教学同步进行。书中提供了大量程序例题,每章之后均附有习题。

本书可作为高等院校计算机及相关专业的本、专科教材,也可以作为从事软件工程技术编程人员的参考书。

图书在版编目(CIP)数据

汇编语言程序设计教程 / 花小朋,刘其明,王一飞编著. —徐州:
中国矿业大学出版社,2010.11

ISBN 978 - 7 - 5646 - 0876 - 7

I. ①汇… II. ①花… ②刘… ③王… III. ①汇编语言—程序设计—
高等学校—教材 IV. ① TP313

中国版本图书馆 CIP 数据核字(2010)第 219398 号

书 名 汇编语言程序设计教程
编 著 花小朋 刘其明 王一飞
责任编辑 杨 洋 仓小金
责任校对 张海平
出版发行 中国矿业大学出版社有限责任公司
(江苏省徐州市解放南路 邮编 221008)
营销热线 (0516)83885307 83884995
出版服务 (0516)83885767 83884920
网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com
印 刷 徐州中矿大印发科技有限公司
开 本 787×1092 1/16 印张 10.5 字数 262 千字
版次印次 2010 年 11 月第 1 版 2010 年 11 月第 1 次印刷
定 价 22.00 元

(图书出现印装质量问题,本社负责调换)

前 言

汇编语言是计算机能够提供给用户使用的最快而又最有效的语言之一,也是能够利用计算机所有硬件特性并能直接控制硬件的唯一语言。在对程序的空间和时间要求很高的场合,汇编语言的应用是必不可少的;很多需要直接控制硬件的应用场合,更是离不开汇编语言。汇编语言的编程模型是 CPU 内部的寄存器、内存单元、CPU 的芯片端口等。因此,掌握了汇编语言,就了解了计算机的内部结构和 CPU 的工作流程,进而可以更好地提高专业技能。

“汇编语言程序设计”是高等院校计算机科学与技术及相关专业必修的一门专业基础课程,它是计算机组成原理、操作系统及其他核心课程的基础课,也是微机原理、单片机应用等课程的学习基础。通过该课程的学习,一方面能够使学生深入地理解计算机内部完成各种复杂操作和运算的基本原理;另一方面可以很好地培养学生掌握程序设计基本技术和基本方法,对以后学习其他的设计语言大有裨益。

全书以 8086/8088 指令系统为基础,共分 9 章进行讲述。第 1 章为汇编语言基础知识,包括汇编语言概述、数制及数制转换以及计算机中数与字符的表示。第 2 章为微型计算机组织,包括基于微处理器的计算机系统构成、中央处理器和存储器。第 3 章为寻址方式与指令系统,包括 7 种基本寻址方式和 8086/8088 的指令系统。第 4 章为汇编语言程序格式,包括汇编程序功能、伪指令、汇编语言程序格式、汇编语言程序的上机过程。第 5 章为分支程序设计,包括转移指令和分支程序设计方法。第 6 章为循环程序设计,包括循环控制指令和循环程序设计方法。第 7 章为子程序设计,包括子程序的定义、子程序的调用与返回指令、子程序的编写方法、子程序的嵌套、中断指令和 DOS 系统功能调用。第 8 章为高级汇编语言技术,包括宏汇编、重复汇编和条件汇编。第 9 章为输入输出程序设计,包括 I/O 设备的数据传送方式、程序直接控制 I/O 方式、中断传送方式及 BIOS 和 DOS 中断。书中提供了大量程序例题,每章之后均附有若干习题,便于读者复习及检查学习效果。在内容组织上,将指令系统中控制类指令全部分散到后续章节中结合相关内容进行介绍。这样,一方面可使学生更好地

掌握相关指令的运用；另一方面也能够让学生较早进行上机编程实践，使理论教学与实践教学同步进行。

本书可作为高等院校计算机及相关专业的本、专科教材，也可以作为软件工程技术编程人员的参考用书。

本书由花小鹏主编并统稿，第1章至第6章由花小鹏编写，第7章、第8章由刘其明编写，第9章由王一飞编写。由于时间仓促，书中如有错误或不当之处，欢迎读者不吝批评指正。

编 者

2010年9月

目 录

第 1 章 汇编语言基础知识	1
1.1 汇编语言概述	1
1.2 数制及数制转换	2
1.2.1 进位计数制	2
1.2.2 数制转换	2
1.3 计算机中数与字符的表示	4
1.3.1 数的补码表示	4
1.3.2 补码的加法和减法运算	5
1.3.3 无符号数	5
1.3.4 字符表示法	5
习题	6
第 2 章 微型计算机组织	8
2.1 基于微处理器的计算机系统构成	8
2.1.1 硬件	8
2.1.2 软件	9
2.2 中央处理器	9
2.2.1 中央处理器 CPU 的组成	9
2.2.2 寄存器结构	9
2.3 存储器	13
2.3.1 存储单元的地址和内容	13
2.3.2 存储器地址的分段	14
习题	15
第 3 章 寻址方式与指令系统	17
3.1 寻址方式	17
3.1.1 立即寻址方式	17
3.1.2 寄存器寻址方式	18
3.1.3 直接寻址方式	19
3.1.4 寄存器间接寻址方式	19
3.1.5 寄存器相对寻址方式	20
3.1.6 基址变址寻址方式	20

3.1.7 相对基址变址寻址方式	21
3.2 8086/8088 的指令系统	22
3.2.1 数据传送指令	22
3.2.2 算术运算指令	28
3.2.3 逻辑指令	34
3.2.4 串处理指令	37
3.2.5 处理机控制指令	42
习题	43
第 4 章 汇编语言程序格式	47
4.1 汇编程序功能	47
4.2 伪指令	48
4.2.1 表达式赋值伪指令 EQU	48
4.2.2 数据定义伪指令	48
4.2.3 段定义伪指令	50
4.2.4 程序开始和结束伪指令	52
4.2.5 模式选择和简化段定义伪指令	52
4.2.6 地址计数器与对准伪指令	54
4.2.7 基数控制伪指令	55
4.2.8 处理器选择伪指令	55
4.3 汇编语言程序格式	55
4.3.1 名字项	55
4.3.2 操作项	56
4.3.3 操作数项	56
4.3.4 注释项	60
4.4 汇编语言程序的上机操作	62
4.4.1 建立 ASM 文件	62
4.4.2 用汇编程序 TASM 对源文件汇编产生 OBJ 文件	63
4.4.3 用连接程序 TLINK 产生 EXE 文件	66
4.4.4 程序的执行	67
4.4.5 COM 文件	67
习题	68
第 5 章 分支程序设计	70
5.1 转移指令	70
5.1.1 无条件转移指令	70
5.1.2 条件转移指令	72
5.2 分支程序设计方法	75
5.2.1 分支程序的结构形式	75

5.2.2 两路分支程序设计.....	75
5.2.3 多路分支程序设计.....	77
习题	80
第6章 循环程序设计	83
6.1 循环控制指令.....	83
6.2 循环程序设计方法.....	84
6.2.1 循环程序的结构形式.....	84
6.2.2 单重循环程序设计.....	85
6.2.3 多重循环程序设计.....	90
习题	94
第7章 子程序设计	96
7.1 子程序的定义.....	96
7.2 子程序的调用和返回指令.....	97
7.2.1 调用指令(CALL)	97
7.2.2 返回指令(RET)	97
7.3 子程序的编写方法.....	99
7.3.1 子程序的调用和返回.....	99
7.3.2 寄存器的保护和恢复.....	99
7.3.3 主程序和子程序之间的参数传递	100
7.3.4 子程序应用举例	106
7.4 子程序的嵌套	108
7.5 中断指令	111
7.5.1 中断的基本概念	111
7.5.2 引起中断的指令	113
7.5.3 中断和子程序的比较	114
7.6 DOS系统功能调用	114
7.6.1 DOS功能模块的调用方法	114
7.6.2 单个字符的输入输出	115
7.6.3 字符串的输入输出	115
习题.....	117
第8章 高级汇编语言技术.....	119
8.1 宏汇编	119
8.1.1 宏的定义、宏调用和宏展开.....	119
8.1.2 宏参数的特殊运算符	122
8.1.3 与宏有关的伪指令	123
8.1.4 宏库的使用	125

8.2 重复汇编	127
8.2.1 重复伪操作	127
8.2.2 不定重复伪操作	129
8.3 条件汇编	130
8.3.1 条件汇编伪指令的功能	130
8.3.2 条件汇编伪指令的举例	131
习题	131
第9章 输入输出程序设计	133
9.1 I/O 设备的数据传送方式	133
9.1.1 CPU 与外设	133
9.1.2 直接存储器存取(DMA)方式	134
9.2 程序直接控制 I/O 方式	134
9.2.1 I/O 端口	134
9.2.2 I/O 指令	135
9.2.3 I/O 程序举例	135
9.3 中断传送方式	138
9.3.1 8086 的中断分类	138
9.3.2 中断向量表	139
9.3.3 中断过程	140
9.3.4 中断优先级和中断嵌套	140
9.3.5 中断处理程序设计	140
9.4 BIOS 和 DOS 中断	142
9.4.1 键盘 I/O	143
9.4.2 显示器 I/O	145
9.4.3 磁盘文件存取技术	147
习题	152
参考文献	154
附录	155

第1章 汇编语言基础知识

1.1 汇编语言概述

计算机只能直接识别由0和1组成的二进制代码。机器指令就是用二进制编码的,每一条机器指令控制计算机完成一个基本操作。某一种处理器所支持的全部指令的集合就是该处理器的指令系统。指令集及使用它们编写程序的规则被称做机器语言。

机器语言程序是计算机唯一能够直接识别并执行的程序,因此,机器语言程序常被称为目标程序(或目的程序)。用机器语言编写的程序占用内存小,运行速度快,但机器语言不易读、出错率高、难维护,也不能直观地反映用计算机解决问题的基本思路。因此,只有在计算机发展的早期或者不得已的情况下,才使用机器语言编写程序。

汇编语言是一种采用助记符表示的程序设计语言。助记符一般都是英语单词的缩写,因而方便人们书写、阅读和检查,例如用“ADD”表示加法。用汇编语言编写的程序称为汇编语言源程序,它不能由机器直接执行,而是需要被翻译成机器语言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称为汇编程序。

汇编语言的指令与机器语言指令一一对应,汇编语言与机器语言都和计算机的硬件系统紧密相关,计算机的硬件系统结构不同,汇编语言和机器语言一般也不同,因此,汇编语言和机器语言均不具有通用性,通常将这两种语言称为低级语言。

汇编语言比机器语言直观,但仍然繁琐难记。20世纪50年代出现了高级程序设计语言。高级语言是相对于低级语言而言的,高级语言采用接近于人类自然语言的语法形式及数学表达形式,它与具体的计算机硬件无关,具有很好的通用性,更容易被掌握和使用。利用高级语言,即使是一般的计算机用户也可以编写软件,而不必懂得计算机的内部结构和工作原理。当然,用高级语言编写的源程序也不会被机器直接执行,也须经过编译或解释程序的翻译才能变为机器语言程序。目前广泛应用的高级语言有10多种,例如简单易用的BASIC语言、算法语言FORTRAN、结构化语言PASCAL、系统程序语言C/C++等。

与机器语言相比,汇编语言易于理解和记忆,也易于编写、阅读和调试。由于其语句与机器指令语句一一对应,因此,汇编语言可以非常方便有效地控制机器。当编写操作系统或其他系统软件时,汇编语言是不可或缺的工具。汇编语言程序的执行效率比高级语言高,这种高效体现在时间和空间两个方面。汇编程序的执行速度快,在实现同样功能的情况下,汇编语言程序生成的二进制可执行程序短。通过学习汇编语言编程,用户将对计算机体系结构及运行原理产生更深刻的认识,这是从高级语言中无法学到的。再者,汇编语言编程与高级语言编程在很多地方是相通的,掌握汇编语言程序设计,有助于学习高级语言编程。

1.2 数制及数制转换

1.2.1 进位计数制

1.2.1.1 十进制

十进制是习惯上最常用的记数法,它的基本元素是 0~9 十个数码,即其基数为 10,遵循逢十进一的规则。一个任意的十进制数可以表示为:

$$a_n a_{n-1} \cdots a_0 . b_1 b_2 \cdots b_m$$

其值为:

$$a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \cdots + a_i \cdot 10^i + \cdots + a_0 \cdot 10^0 + b_1 \cdot 10^{-1} + b_2 \cdot 10^{-2} + \cdots + b_j \cdot 10^{-j} + \cdots + b_m \cdot 10^{-m}$$

其中, $a_i (i=0, 1, \cdots, n)$, $b_j (j=1, 2, \cdots, m)$ 是 0~9 十个数码中的一个。

上式中相应于每位数字的 10^k 称为该位数字的权,每位数字乘以其权所得到的乘积之和即为所表示的数值。例如:

$$123.45 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

1.2.1.2 二进制

在计算机内,为便于存储及计算的物理实现,采用二进制数表示。二进制的基本元素是 0 和 1,即其基数为 2,遵循逢二进一的规则,各位的权是 2^k 。一个任意的二进制数可以表示为: $a_n a_{n-1} \cdots a_0 . b_1 b_2 \cdots b_m$, 其值为:

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_i \cdot 2^i + \cdots + a_0 \cdot 2^0 + b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \cdots + b_j \cdot 2^{-j} + \cdots + b_m \cdot 2^{-m}$$

其中, $a_i (i=0, 1, \cdots, n)$, $b_j (j=1, 2, \cdots, m)$ 是 0 和 1 两个数码中的一个。例如:

$$101.101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 5.625_{10}$$

1.2.1.3 八进制与十六进制

八进制的基本元素是 0~7,即其基数为 8,遵循逢八进一的原则,各位的权是 8^k 。十六进制的基本元素是 0~9 及 A~F,即其基数为 16,遵循逢十六进一的原则,各位的权是 16^k 。

1.2.1.4 各种记数法书写规范

在计算机里,通常用数字后面跟一个英文字母来表示该数的数制。

十进制数:一般用 D(decimal)表示,如 56D。

二进制数:用 B(binary)表示,如 11010101B。

八进制数:用 O(octal)表示,如 567O。

十六进制数:用 H(hexadecimal)来表示,如 56H。当十六进制数的第一个字符是字母时,在第一个字符之前必须添加一个“0”,如 123H、0FFH、0AB34H 等都是十六进制数。

1.2.2 数制转换

1.2.2.1 非十进制数转换成十进制数

方法:将非十进制数按权展开后相加求和即为相应的十进制数。

【例 1.1】 $101101.101\text{B}=1\times 2^5+1\times 2^3+1\times 2^2+1\times 2^0+1\times 2^{-1}+1\times 2^{-3}=45.625\text{D}$

$$123\text{O}=1\times 8^2+2\times 8^1+3\times 8^0=83\text{D}$$

$$0\text{F}2\text{DH}=15\times 16^2+2\times 16^1+13\times 16^0=3885\text{D}$$

1.2.2.2 十进制数转换成非十进制数

方法:将整数部分与小数部分分别转换。整数部分采用除基数取余数法,直至商为0,先得到的余数为低位,后得到的余数为高位。小数部分采用乘基数取整法,直至乘积为整数或到达指定精度,先得到的整数为高位,后得到的整数为低位。

【例 1.2】 $117\text{D}=1110101\text{B}$

$$117/2=58 \quad (a_0=1)$$

$$58/2=29 \quad (a_1=0)$$

$$29/2=14 \quad (a_2=1)$$

$$14/2=7 \quad (a_3=0)$$

$$7/2=3 \quad (a_4=1)$$

$$3/2=1 \quad (a_5=1)$$

$$1/2=0 \quad (a_6=1)$$

【例 1.3】 $0.8125\text{D}=0.1101\text{B}$

$$0.825\times 2=1.625 \quad (b_1=1)$$

$$0.625\times 2=1.25 \quad (b_2=1)$$

$$0.25\times 2=0.5 \quad (b_3=0)$$

$$0.5\times 2=1.0 \quad (b_4=1)$$

1.2.2.3 二、八、十六进制数之间的转换

方法:将二进制数转换成八进制数可按3位一组进行,转换成十六进制可按4位一组进行,每一组对应八进制或十六进制相应数码;分组时如果位数不够,整数部分在最左边补0,小数部分在最右边补0。将八进制数转换成二进制数,只需将八进制数一位对应换成3位二进制数即可;同样对十六进制数,只需将其中一位对应换成二进制数四位即可。十进制、二进制、八进制和十六进制数之间对应关系如表1.1所示。

【例 1.4】 $1100100.11011\text{B}=\underline{001}\ \underline{100}\ \underline{100}.\underline{110}\ \underline{110}\text{B}=144.66\text{O}$
 $=\underline{0110}\ \underline{0100}.\underline{1101}\ \underline{1000}\text{B}=64.\text{D8H}$

表 1.1 十、二、八和十六进制数之间对应关系表

二进制	十进制	八进制	十六进制
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6

续表 1.1

二进制	十进制	八进制	十六进制
111	7	7	7
1000	8	10	8
1001	9	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

1.3 计算机中数与字符的表示

1.3.1 数的补码表示

在计算机中,数与数的符号都是用二进制来表示的。连数符一起数值化了的数,称为机器数。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。机器数可以用不同的码制来表示,常用的表示法有原码、补码和反码。由于多数机器中整数采用补码表示,这里只介绍补码表示法。

补码表示法中,一个正数的补码就是该数本身,即正号用 0 表示,其余数值部分保持不变;一个负数的补码,负号用 1 表示,其余数值部分按位取反(即 0 变成 1,1 变成 0)并且末位加 1。

【例 1.5】 机器字长为 8 位,求 +42D 的补码。

+42D 可表示为 + 010 1010
 +42D 的补码为 0 010 1010
 十六进制表示为 2 A
 即 $[+42D]_{\text{补}} = 2AH$

【例 1.6】 机器字长为 8 位,求 -42D 的补码。

-42D 可表示为 -010 1010
 -42D 的补码为 1101 0110
 十六进制表示为 D 6
 即 $[-42D]_{\text{补}} = 0D6H$

下面,再来讨论一下 n 位补码表示数的范围。8 位二进制数可以表示 $2^8 = 256$ 个数,当它们是补码表示的带符号数时,表示的数的范围是 $-128 \leq N \leq +127$ 。一般, n 位补码表示的数的范围是:

$$-2^{n-1} \leq N \leq 2^{n-1} - 1$$

所以 $n=16$ 时的表示的数的范围是: $-32768 \leq N \leq +32767$ 。

1.3.2 补码的加法和减法运算

在计算机系统中采用补码表示带符号数的原因:① 用补码表示数据可以将符号位作为有效数位参与运算,根据运算结果的最高位可直接确定结果的符号;② 实现减法运算向加法运算的转化,从而降低运算器硬件电路设计和实现的难度,提高运算速度和可靠性。

1.3.2.1 补码加法运算

运算规则为: $[X]_{\text{补}} + [Y]_{\text{补}} = [X + Y]_{\text{补}}$,即补码之和等于和的补码。

【例 1.7】 机器字长为 8 位, $X = +25D$, $Y = +32D$, 用补码加法规则求 $X + Y$ 。

$$[X]_{\text{补}} = 00011001B, [Y]_{\text{补}} = 00100000B$$

$$[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 00011001B + 00100000B = 00111001B$$

$$X + Y = +011001B = +57D$$

1.3.2.2 补码减法运算

求补运算:对一个二进制数按位求反后在末位加 1 的运算称为求补运算。

减法运算规则为: $[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$,其中 $[-Y]_{\text{补}}$ 只要对 $[Y]_{\text{补}}$ 求补即可得到。

【例 1.8】 机器字长为 8 位, $X = +32D$, $Y = +25D$, 用补码减法规则求 $X - Y$ 。

$$[X]_{\text{补}} = 00100000B, [Y]_{\text{补}} = 00011001B, [-Y]_{\text{补}} = 11100111B$$

$$[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 00011001B + 11100111B = 00000111B$$

$$X - Y = +0000111B = +7D$$

1.3.3 无符号数

在某些情况下,如果要处理的数全是正数,那么再保留符号位就没有意义了。把最高有效位也当做数值处理,这种数称为无符号数。8 位无符号数表示的范围是 $0 \leq N \leq 255$, 16 位无符号数表示范围是 $0 \leq N \leq 65\,535$ 。 n 位无符号数表示范围是 $0 \leq N \leq 2^n - 1$ 。

在计算机中最常用的无符号数是表示地址的数。此外,如双精度数的地位字也是无符号数等。在某些情况下,带符号数(在机器中用补码表示)与无符号数的处理是有差别的,读者在处理数时,应注意它们的区别。

1.3.4 字符表示法

计算机中处理的信息除数值以外,还包括字符,例如从键盘输入的信息或从打印机输出的信息都是以字符方式输入输出的。字符包括:

- (1) 字母: $A \sim Z, a \sim z$;
- (2) 数字: $0 \sim 9$;
- (3) 专用字符: $+, -, *, /, \uparrow, \text{SP}(\text{space 空格}), \dots$
- (4) 非打印字符: $\text{BELL}(\text{bell 响铃}), \text{LF}(\text{line feed 换行}), \text{CR}(\text{carriage return 回车}), \dots$

由于计算机只识别 0 和 1,所以这些字符在计算机中必须以二进制编码的形式表示。一般计算机采用美国信息交换标准代码 ASCII(american standard code for information interchange)来表示。ASCII 码用一个字节(8 位二进制码)来表示一个字符,其中低 7 位为

字符的 ASCII 值,最高位一般用做校验位。表 1.2 列出了用十六进制数表示的部分常用字符 ASCII 值。

表 1.2 常用字符的 7 位 ASCII 值(用十六进制数表示)

字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值
NUL	00	4	34	M	4D	f	66
BEL	07	5	35	N	4E	g	67
LF	0A	6	36	O	4F	h	68
FF	0C	7	37	P	50	i	69
CR	0D	8	38	Q	51	j	6A
SP	20	9	39	R	52	k	6B
!	21	:	3A	S	53	l	6C
"	22	;	3B	T	54	m	6D
#	23	<	3C	U	55	n	6E
\$	24	=	3D	V	56	o	6F
%	25	>	3E	W	57	p	70
&	26	?	3F	X	58	q	71
'	27	@	40	Y	59	r	72
(	28	A	41	Z	5A	s	73
)	29	B	42	[	5B	t	74
*	2A	C	43	\	5C	u	75
+	2B	D	44	]	5D	v	76
,	2C	E	45	↑	5E	w	77
-	2D	F	46	←	5F	x	78
.	2E	G	47	.	60	y	79
/	2F	H	48	a	61	z	7A
0	30	I	49	b	62	{	7B
1	31	J	4A	c	63		7C
2	32	K	4B	d	64	}	7D
3	33	L	4C	e	65	~	7E

习 题

1.1 机器语言、汇编语言和高级语言各自有什么特点? 汇编语言与机器语言和高级语言相比,有哪些优点?

1.2 解释汇编语言源程序、汇编程序及目标程序的概念。

1.3 将下列十进制数转换成二进制数和十六进制数。

A. 127

B. 125.25

C. 32767

D. 4096

- 1.4 将下列二进制数转换成十六进制数和十进制数。
A. 1011101 B. 10000000 C. 11111111 D. 101010101010
- 1.5 将下列十六进制数转换成二进制数和十进制数
A. 5BH B. FBH C. 3456H D. FFCCH
- 1.6 假设机器字长为8位,给出下列十六进制数的补码形式(用二进制表示)。
A. -34 B. A5 C. 9B D. -FE
- 1.7 下列各数均为十六进制数,用8位二进制补码计算下列各题,并用十六进制数表示其运算结果。
A. $(-56)+27$ B. $56+(-27)$
C. $56-(-27)$ D. $-56-(-27)$
- 1.8 写出下列十六进制数在被看做是补码表示带符号数或无符号数时的十进制数值。
A. FE B. 2C C. D8 D. 57
- 1.9 在ASCII表中,字符'0'~'9'与数值0~9之间的编码规律是什么?大写字母与小写字母之间的编码规律是什么?

第2章 微型计算机组织

汇编语言是一种可以直接控制计算机硬件设备的计算机语言。掌握一些计算机的硬件资源知识是学习汇编语言的必要前提,因此在介绍汇编语言之前先介绍计算机硬件系统结构、寄存器和存储器组织。

2.1 基于微处理器的计算机系统构成

计算机系统包括硬件和软件两部分。硬件包括电路、插件板、机柜等;软件则是为了管理和维护计算机而编制的各种程序的总和。

2.1.1 硬件

微型计算机硬件系统主要由中央处理器(CPU)、存储器(Memory)和输入/输出设备(I/O 设备)构成,用系统总线把它们连接在一起,如图 2.1 所示。

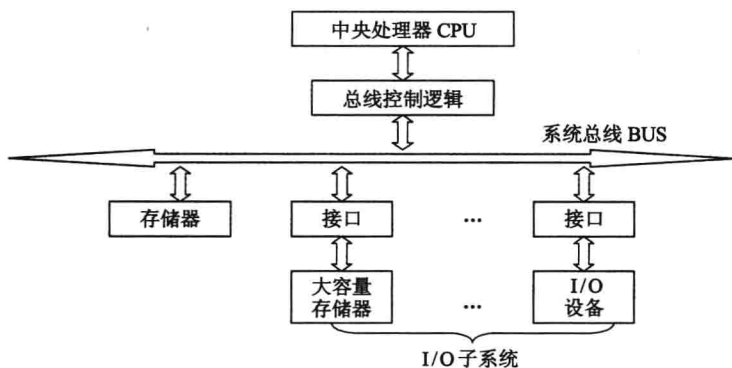


图 2.1 计算机结构

中央处理器包括运算器和控制器两部分,是计算机的核心。运算器执行所有的算术和逻辑运算指令;控制器则负责全机的控制工作,它负责从存储器中逐条取出指令,经译码分析后发出取数、执行、存数等控制操作,用来确保完成程序所要求的功能。

存储器是计算机的记忆部件,用来存放用户编写的程序以及程序中所用的数据、信息和中间结果。

I/O 子系统一般包括 I/O 设备及大容量存储器两类外部设备。I/O 设备包括输入设备和输出设备,负责计算机和外部世界的通信。CPU 与外设通过 I/O 接口电路交换信息。输入设备如键盘、鼠标、扫描仪等,输出设备如显示器、绘图仪、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、光盘等。