



21st CENTURY  
规划教材

面向21世纪高职高专计算机系列规划教材  
COURSES FOR VOCATIONAL HIGHER EDUCATION, COMPUTER

# 汇编语言程序设计

## ASSEMBLY LANGUAGE PROGRAMMING

龚荣武 主编



科学出版社  
www.sciencep.com



面向21世纪高职高专计算机系列规划教材  
COURSES FOR VOCATIONAL HIGHER EDUCATION: COMPUTER

# 汇编语言程序设计

龚荣武 主编

杨大千 韦 坤 刘 萍 蔡 英 副主编

科学出版社

北京

## 内 容 简 介

本书是高职高专计算机规划教材, 主要内容包括: 微型计算机基础知识、IBM-PC 微型计算机系统概述、8086/8088 指令系统、80X86/Pentium 微处理器指令系统、汇编语言程序设计基本方法、三大结构程序设计(顺序、选择、循环)、子程序设计、汇编语言程序设计示例、模块化程序设计的基本技术、上机实验内容及汇编语言程序调试方法等。全书共分 10 章。内容丰富, 系统性强, 深入浅出。在组织和阐述内容时, 我们尽量将理论知识和具体的实际应用联系起来讲述, 对相应的章节还配有大量实例。在每一章中, 我们将本章的内容、要点等通过“知识点”、“难点”、“要求”等对读者做出提示, 使读者在学习的过程中对每一章的具体知识点、重点难点以及要求掌握和了解的内容有一个清楚的认识, 方便读者对全书内容和组织结构有一个系统化的了解; 书后以附录的方式提供了 ASCII 码表、80X86 指令表、Debug 命令表等相应的学习汇编语言必备资料, 以方便读者自学。

本书可作为高职高专计算机、通信、网络等专业的教材。

### 图书在版编目(CIP)数据

汇编语言程序设计/龚荣武主编. —北京: 科学出版社, 2004

(面向 21 世纪高职高专计算机系列规划教材)

ISBN 7-03-013997-6

I. 汇… II. 龚… 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(2004)第 075285 号

责任编辑: 田悦红/责任校对: 耿 耘

责任印制: 吕春珉/封面设计: 飞天创意

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

世界知识印刷厂 印刷

科学出版社发行 各地新华书店经销

\*

2004 年 8 月 第 一 版 开本: 787×1092 1/16

2004 年 8 月 第一次印刷 印张: 15 1/4

印数: 1—3 000 字数: 341 000

定价: 20.00 元

(如有印装质量问题, 我社负责调换<世知>)

# 面向 21 世纪高职高专规划教材专家委员会

主任 李宗尧

副主任 (按姓氏笔画排序)

丁桂芝 叶小明 张和平 林 鹏  
黄 藤 谢培苏

委员 (略)

## 信息技术系列教材编委会

主任 丁桂芝

副主任 (按姓氏笔画排序)

万金保 方风波 徐 红 鲍 泓

委员 (按姓氏笔画排序)

于晓平	马国光	仁英才	王东红	王正洪
王 玉	王兴宝	王金库	王海春	王爱梅
邓 凯	付百文	史宝会	本柏忠	田 原
申 勇	任益夫	刘成章	刘克敏	刘甫迎
刘经玮	刘海军	刘敏涵	安志远	许殿生
何瑞麟	余少华	吴春英	吴家砮	吴瑞萍
宋士银	宋锦河	张红斌	张环中	张海鹏
张蒲生	张德实	李云程	李文森	李 洛
李德家	杨永生	杨 闯	杨得新	肖石明
肖洪生	陈 愚	周子亮	周云静	胡秀琴
赵从军	赵长旭	赵动庆	郝 梅	唐铸文
徐洪祥	徐晓明	袁德明	郭庚麒	高延武
高爱国	康桂花	戚长政	曹文济	黄小鸥
彭丽英	董振珂	蒋金丹	韩银峰	魏雪英

## 本书编写人员名单

主 编 龚荣武

副主编 刘 萍 蔡 英

撰稿人(按姓氏笔画排序)

尹晓燕	韦 坤	刘 萍
杨大千	龚荣武	蔡 英

## 出版前言

随着世界经济的发展,人们越来越深刻地认识到经济发展需要的人才多元化、多层次的,既需要大批优秀的理论性、研究性的人才,也需要大批应用性人才。然而,我国传统的教育模式主要是培养理论性、研究性的人才。教育界在社会对应用性人才需求的推动下,专门研究了国外应用性人才教育的成功经验,结合国情大力度地改革我国的“高等职业教育”,制定了一系列的方针政策。联合国教科文组织1997年公布的教育分类中将这种教育称之为“高等技术与职业教育”,也就是我们通常所说的“高职高专”教育。

我国经济建设需要大批应用性人才,呼唤高职高专教育的崛起和成熟,寄希望于高职高专教育尽快向国家输送高质量的紧缺人才。近几年,高职高专教育发展迅速。目前,各类高职高专学校已占全国高等院校的近1/2,约有600所之多。教育部针对高职高专教育出台的一系列政策和改革方案主要体现在以下几个方面:

- “就业导向”成为高职高专教育的共识。高职高专院校在办学过程中充分考虑市场需求,用“就业导向”的思想制定招生和培养计划。
- 加快“双师型”教师队伍建设。已建立12个国家高职高专学生和教师的实训基地。
- 对学生实行“双认证”教育。学历文凭和职业资格“双认证”教育是高职高专教育特色之一。
- 高职高专教育以2年学制为主。从学制入手,加快高职高专教学方向的改革,充分办出高职高专教育特色,尽快完成紧缺人才的培养。
- 开展精品专业和精品教材建设。已建立科学的高职高专教育评估体系和评估专家队伍,指导、敦促不同层次、不同类型的学校办出一流的教育。

在教育部关于“高职高专”教育思想和方针指导下,科学出版社积极参与到高职高专教材的建设中来。在组织教材过程中采取了“请进来,走出去”的工作方法。即:由教育界的专家、领导和一线的教师,以及企事业从事人力资源工作的人员组成顾问班子,充分分析我国各地区的经济发展、产业结构以及人才需求现状,研究培养国家紧缺人才的关键要素,寻求切实可行的教学方法、手段和途径。

通过研讨认识到,我国幅员辽阔,各地区的产业结构有明显的差异,经济发展也不平衡,各地区对人才的实际需求也有所不同。相应地,相同专业和相近专业,不同地区的教学单位在培养目标和培养内容上也各有自己的定位。鉴此,适应教育现状的教材建设应该具有多层次的设计。

为了使教材的编写能针对受教育者的培养目标,出版社的编辑分不同地区逐所学校拜访校长、系主任和老师,深入到高职高专学校及相关企事业,广泛、深入地和教学第

一线的老师、用人单位交流，掌握了不同地区、不同类型的高职高专院校的教师、学生和教学设施情况，清楚了各学校所设专业的培养目标和办学特点，明确了用人单位的需求条件。各区域编辑对采集的数据进行统计分析，在相互交流的基础上找出各地区、各学校之间的共性和个性，有的放矢地制定选题项目，并进一步向老师、教育管理者征询意见，在获得明确指导性意见后完成“高职高专规划教材”策划及教材的组织工作：

- 第一批“高职高专规划教材”包括三个学科大系：经济管理、信息技术、建筑。
- 第一批“高职高专规划教材”在注意学科建设完整性的同时，十分关注具有区域人才培养特色的教材出版。
- 第一批“高职高专规划教材”组织过程中，正值高职高专学制从3年制向2年制转轨，教材编写将其作为考虑因素，要求提示不同学制的讲授内容。
- 第一批“高职高专规划教材”编写
  - ◆ 强调以就业岗位对知识和技能需求下的教材体系的系统性、科学性和实用性。
  - ◆ 强调教材以实例为先，应用为目的；围绕应用讲理论，取舍适度，不追求理论的完整性。
  - ◆ 强调提出问题→解决问题→归纳问题的教、学法，培养学生触类旁通的实际工作能力。
  - ◆ 强调课后作业和练习（或实训）真正具有培养学生实践能力的作用。

在“高职高专规划教材”编委的总体指导下，第一批各科教材基本是由系主任，或从教学一线中遴选的骨干教师执笔撰写。在每本书主编的严格审读及监控下，在各位老师的辛勤编撰下，这套凝聚了所有作者及参与研讨的老师们的经验、智慧和资源，涉及三个大的学科近200种的高职高专教材即将面世。我们希望经过近一年的努力，我们奉献给读者的是他们渴望已久的适用教材。同时，我们也清醒地认识到，“高职高专”是正在探索中的教育，加之我们的水平和经验有限，教材的选题和编辑出版会存在许多不尽人意的地方，真诚地希望得到老师和学生的批评建议，以利今后改进，为繁荣我国的高职高专教育不懈努力。

科学出版社

2004年6月1日

# 前 言

微型计算机的使用已非常普及,在国内使用 80X86 处理器的微机占大多数,在这样的微机上进行各种应用软件开发是高职高专计算机专业的学生应该掌握的。

汇编语言是一种低级语言,它与硬件关系最为密切,由于汇编语言具有目标代码短、执行速度快、能够充分利用硬件资源及能够进行精确控制的优点,这些优点是其他高级语言远远不能比拟的。正是这一优点,使得在某些编程中应该、甚至必须使用汇编语言。要了解微型计算机是如何工作,各种硬件在微机系统中起什么作用,仅仅掌握高级语言的程序设计是无法达到要求的,只有掌握了汇编语言程序设计才能深入了解微型计算机的工作原理。为此,高职高专计算机专业学生学习汇编语言,掌握汇编语言基本程序的设计方法,是非常有意义的。

本书以 Intel 8086/8088 系列微机为对象,介绍汇编语言程序设计。在介绍 Intel 8086/8088 微处理器的寻址方式和指令系统的基础上,详细讲解了编写汇编语言程序的基本方法和技巧。掌握这些方法和技巧,不仅有助于利用 Intel 8086/8088 及 Pentium 系列微机的汇编语言程序进行编程设计,并为 Intel 80X86 及 Pentium 系列微机的汇编语言程序设计奠定了基础。

本书内容的安排力求循序渐近、重点突出、难点分散,通过理论课的课堂讲授和上机实验,学生能对以下问题进行汇编的编程:非压缩 BCD 码数的加、减、乘、除;压缩 BCD 数的加、减、乘、除;二进制数(在一个字的长度范围内)的加、减、乘、除;将十进制数转换成二进制数,再变成补码;将二进制数转换成十进制数;各种数、字符的输出等。在内容的组织和阐述中,我们尽量将理论知识和具体的实际应用联系起来讲述,对相应的章节还配有大量实例。在每一章中,我们将本章的内容、要点等通过“知识点”、“难点”、“要求”等对读者做出提示,使读者在学习的过程中对每一章的具体知识点、重点难点以及要求掌握和了解的内容有清楚的认识,方便读者对全书内容和组织结构有系统化的了解;书后我们还以附录的方式提供了 ASCII 码表、80X86 指令表、Debug 命令表等相应的学习汇编语言必备资料,方便读者自学。

本书由龚荣武任主编,刘萍、蔡英任副主编,杨大千编写了第 1、2 章,韦坤编写了第 3、4、5 章,刘萍编写了第 6、7 章,刘萍、杨大千合编第 8 章,蔡英编写了第 9、10 章,杨大千、韦坤、刘萍、尹晓燕整理了附录,全书由龚荣武教授统稿。

在本书的编写过程中,得到科学出版社的热情支持和帮助,在此表示衷心的感谢。

在本书的编写过程中,得到网友 John 的指点和他提供的 Debug 相关资料;得到“编程爱好者”网站(<http://www.programfan.net>)论坛中部分网友的点拨,在此一并表示衷心的感谢。

由于编者水平有限,书中难免存在一些缺点和错误,恳请广大读者和同行专家批评指正。

编 者

2004 年 6 月



# 目 录

<b>第 1 章 微型计算机基础知识</b> .....	1
1.1 微型计算机系统简述 .....	1
1.1.1 微处理器 .....	2
1.1.2 微型计算机系统 .....	2
1.2 微型计算机数和字符的表示 .....	6
1.2.1 计算机中数的表示 .....	6
1.2.2 常用数制及其算术运算 .....	7
1.2.3 数制的转换 .....	8
1.2.4 数的表示 .....	11
1.2.5 加减法运算的溢出条件 .....	13
1.2.6 字符的表示 .....	14
1.3 汇编语言程序设计概述 .....	14
1.3.1 汇编语言 .....	14
1.3.2 汇编语言程序设计 .....	15
习题 .....	15
<b>第 2 章 IBM-PC 微型计算机系统概述</b> .....	17
2.1 8086/8088 的功能结构 .....	17
2.1.1 执行单元 EU .....	18
2.1.2 总线接口单元 BIU .....	18
2.2 8086/8088 的寄存器结构 .....	19
2.2.1 通用寄存器 .....	19
2.2.2 段寄存器 .....	20
2.2.3 状态寄存器 .....	21
2.2.4 指令指针寄存器 .....	22
2.3 存储器的组织 .....	22
2.3.1 存储器的标准结构 .....	22
2.3.2 存储器的分段 .....	23
2.3.3 实际地址和逻辑地址 .....	23
2.3.4 堆栈 .....	24
* 2.4 Intel 80X86 系统高档微处理器简介 .....	24
2.4.1 80286 微处理器 .....	24
2.4.2 80386 微处理器 .....	26
2.4.3 80486 微处理器 .....	26
2.4.4 Pentium 微处理器 .....	26

习题 .....	27
<b>第3章 8086/8088 指令系统</b> .....	<b>29</b>
3.1 指令的基本格式 .....	29
3.1.1 指令的构成 .....	29
3.1.2 8086 指令的通用格式 .....	30
3.2 8086/8088 的寻址方式 .....	31
3.2.1 固定寻址 .....	32
3.2.2 立即寻址 .....	32
3.2.3 寄存器寻址 .....	32
3.2.4 存储器寻址 .....	32
3.2.5 I/O 端口寻址 .....	35
3.3 指令的执行时间 .....	36
3.4 8086/8088 指令系统 .....	38
3.4.1 数据传送类指令及应用 .....	38
3.4.2 算术运算类指令及应用 .....	43
3.4.3 逻辑运算与移位类指令及应用 .....	49
3.4.4 串操作类指令及应用 .....	52
3.4.5 控制转移类指令及应用 .....	52
3.4.6 处理器控制类指令及应用 .....	52
习题 .....	53
<b>第4章 汇编语言程序</b> .....	<b>55</b>
4.1 汇编语言与汇编程序 .....	55
4.1.1 汇编语言 .....	55
4.1.2 汇编程序 .....	56
4.2 汇编语言程序的格式和组成元素 .....	56
4.2.1 语句类型(指令、伪指令、宏指令三类) .....	57
4.2.2 汇编语言语句的四个域 .....	57
4.2.3 汇编语言程序的段结构 .....	62
4.3 伪指令及其应用 .....	63
4.3.1 符号定义伪指令 .....	63
4.3.2 数据定义及存储器分配伪指令 .....	63
4.3.3 标号定义伪指令 .....	66
4.3.4 段定义伪指令 .....	66
4.3.5 过程定义伪指令 .....	68
4.3.6 80X86 指令集选择伪指令 .....	68
4.4 汇编语言程序的上机过程 .....	69
4.4.1 建立 ASM 文件 .....	69
4.4.2 生成 OBJ 文件 .....	69
4.4.3 生成 EXE 文件 .....	70

4.4.4	快速生成可执行文件的方法 .....	71
4.4.5	程序的执行和调试 .....	72
4.4.6	TASM、TLINK 及 Turbo Debug 的使用 .....	76
习题	.....	77
<b>* 第 5 章</b>	<b>80X86/Pentium 微处理器指令系统 .....</b>	<b>79</b>
5.1	80286 增强和扩充指令 .....	79
5.1.1	80286 工作模式 .....	79
5.1.2	有符号整数乘法指令 .....	80
5.1.3	堆栈操作指令 .....	80
5.1.4	移位指令 .....	82
5.1.5	支持高级语言的指令 .....	82
5.1.6	控制保护指令 .....	83
5.2	80386 增强和扩充指令 .....	84
5.2.1	数据传送与扩展指令 .....	84
5.2.2	地址传送指令 .....	85
5.2.3	有符号乘法指令 .....	86
5.2.4	符号扩展指令 .....	86
5.2.5	堆栈操作指令 .....	87
5.2.6	移位指令 .....	87
5.2.7	位操作指令 .....	88
5.2.8	条件设置指令 .....	89
5.3	80486 新增指令 .....	90
5.3.1	字节交换指令 .....	90
5.3.2	互换并相加指令 .....	90
5.3.3	比较并相加指令 .....	91
5.3.4	Cache 管理指令 .....	91
5.4	Pentium 新增指令 .....	92
5.4.1	8 字节比较交换指令 .....	92
5.4.2	处理器特征识别指令 .....	92
5.4.3	读时间标记计数器指令 .....	92
5.4.4	读模型专用寄存器指令 .....	92
5.4.5	写模型专用寄存器指令 .....	93
习题	.....	93
<b>第 6 章</b>	<b>顺序程序设计 .....</b>	<b>94</b>
6.1	汇编语言程序设计的基本步骤 .....	94
6.1.1	问题分析 .....	94
6.1.2	确定算法, 编制流程图 .....	95
6.1.3	编码与调试 .....	96
6.2	顺序程序设计 .....	97

6.2.1	输入/输出的 DOS 功能调用 .....	98
6.2.2	十进制 BCD 码校正指令 .....	100
6.2.3	顺序程序设计举例 .....	104
习题	.....	107
<b>第 7 章</b>	<b>分支程序设计</b> .....	109
7.1	分支程序结构 .....	109
7.1.1	单边选择程序设计 .....	109
7.1.2	多边选择程序设计 .....	111
7.2	转移指令 .....	111
7.2.1	条件转移指令及应用 .....	112
7.2.2	无条件转移指令及应用 .....	114
7.3	分支程序设计 .....	116
7.3.1	用比较转移指令实现分支 .....	116
7.3.2	转移指示表法实现分支 .....	119
习题	.....	124
<b>第 8 章</b>	<b>循环程序设计</b> .....	126
8.1	循环程序结构 .....	126
8.1.1	问题的提出 .....	126
8.1.2	循环程序结构 .....	127
8.2	循环指令 .....	128
8.2.1	重复控制指令(循环控制指令) .....	128
8.2.2	串操作指令及重复前缀 .....	129
8.3	循环程序设计 .....	131
8.3.1	先执行后判断结构的循环程序设计 .....	131
8.3.2	先判断后执行结构的循环程序设计 .....	132
8.3.3	循环程序的控制方法 .....	133
8.4	多重循环 .....	135
习题	.....	138
<b>第 9 章</b>	<b>子程序设计</b> .....	140
9.1	子程序与调用程序 .....	140
9.1.1	子程序设计方法 .....	141
9.1.2	子程序的调用 .....	141
9.2	具有子程序的汇编程序设计 .....	141
9.2.1	含有 NEAR 过程及过程调用的程序设计 .....	141
9.2.2	含有 FAR 过程及过程调用的程序设计 .....	143
9.3	保护现场与恢复现场 .....	144
9.4	过程的参数传递方法 .....	146
9.4.1	约定寄存器法 .....	146
9.4.2	约定存储单元法 .....	147

---

9.4.3 伪指令 EXTRN, PUBLIC 的使用 .....	148
9.4.4 用寄存器传送参数地址表的地址 .....	152
9.4.5 用堆栈传送参数 .....	154
9.5 过程的嵌套和递归调用 .....	157
9.5.1 过程的嵌套 .....	157
9.5.2 过程的递归调用 .....	158
<b>第 10 章 汇编语言程序设计示例 .....</b>	<b>160</b>
10.1 字符处理 .....	160
10.2 代码转换 .....	162
10.3 表的处理 .....	176
10.4 多模块的连接 .....	180
10.4.1 运行汇编源程序的过程 .....	180
10.4.2 多模块的连接 .....	181
<b>附录 .....</b>	<b>183</b>
附录 A ASCII 码表 .....	183
附录 B 80X86 指令表 .....	184
附录 C TASM 5.0 宏汇编程序出错信息表 .....	203
附录 D Debug 命令表 .....	211
附录 E 上机实验指导 .....	218
<b>主要参考文献 .....</b>	<b>229</b>

# 第 1 章 微型计算机基础知识



## 知识点

- 微处理器、微型机和微型计算机系统
- 微型计算机数和字符的表示
- 汇编语言程序设计概述



## 难点

- 微处理器的组成和结构
- 微型计算机数和字符的表示
- 数制的转换
- 汇编语言的概念



## 要求

掌握:

- 微处理器、微机、微机系统的概念
- 微型计算机数和字符的表示及各类进制之间的转换
- 衡量微机的主要性能指标

了解:

- 微型计算机的主要特点
- 汇编语言的基本概念

## 1.1 微型计算机系统简述

在过去的半个多世纪中,计算机技术的产生、发展和应用给整个人类社会带来了翻天覆地的变革,全人类得到了史无前例的解放。自 1946 年第一台电子数字计算机问世以来,计算机技术得到了迅猛的发展,随着微电子技术的不断发展和集成电路加工工艺的进步,计算机的核心部件从电子管、晶体管、中小规模集成电路发展到当今的超大规模集成电路。在 20 世纪 70 年代出现的第一台微型计算机,其主要功能仍然主要集中在“计算工具”上,但是,微型计算机的出现为现代计算机的广泛应用开拓了广阔的前景。今天,计算机特别是微型机技术的发展水平、生产规模和应用程度已经成为衡量一个国家现代化水平的一个重要指标,计算机已经不再是仅仅“用作计算”的工具,它已经在工农业生产、国防、航空航天、海洋工程、新材料、人们的日常生活等各个领域得到了广泛的应用,成为推动科技发展和人类社会进步的源动力。

### 1.1.1 微处理器

微处理器 (Microprocessor), 简单地说, 就是将包括运算器和控制器的中央处理器 CPU 的复杂电路固化在一片大规模集成电路的芯片上, 我们把这种微缩的 CPU 芯片称为微处理器或微处理机, 通常直接用 CPU 来表示。

微处理器的主要功能是进行算术运算、逻辑运算和控制整个计算机自动、协调、高效地完成一系列操作。一般的微处理器芯片的尺寸只有十几个平方毫米, 大的可达十几个平方厘米, CPU 通常固化或直接插在集成电路板上。

### 1.1.2 微型计算机系统

电子数字计算机是一个自动化的计算工具, 要使计算机完成自动计算, 它必须具备一个完备的硬件系统和相应的软件系统。因此, 我们说计算机系统就是指的硬件系统和软件系统的总和。

#### 1. 计算机的硬件系统

计算机的硬件系统是由中央处理器 (Central Processing Unit)、存储器 (Memory)、输入输出接口 (Input/Output Interface) 及 I/O 设备这样三部分并通过系统总线 (Bus) 将它们连接起来的一个整体, 微型计算机的硬件系统同一般的计算机一样, 也是由这样三部分组成的, 其硬件系统的结构如图 1-1 所示。

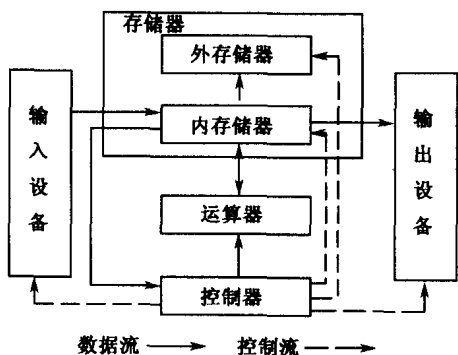


图 1-1 微型计算机硬件系统结构框图

输入输出接口 (Input/Output Interface) 及 I/O 设备这样三部分并通过系统总线 (Bus) 将它们连接起来的一个整体, 微型计算机的硬件系统同一般的计算机一样, 也是由这样三部分组成的, 其硬件系统的结构如图 1-1 所示。

#### (1) 中央处理器

中央处理器简称 CPU, 它是计算机的基本部件。它由三部分组成, 即算术逻辑运算单元、控制单元、通用寄存器。算术逻辑单元承担算术运算和逻辑运算。例如加、减, 有的还有乘除, 以及逻辑与、或、非、异或等运算。控制

单元负责产生计算机工作所需的时钟及控制信号。寄存器是一种快速的存储设备。

现代计算机所使用的 CPU 都是并行工作的。它的主要技术指标对使用者来说是字长和指令的执行速度。从硬件角度谈 CPU 的字长是指 CPU 一次能并行处理的二进制数据的位数。如果 CPU 一次能处理 8 位二进制的的数据, 我们就说它是 8 位 (字长) 的 CPU, 如果 CPU 一次能处理 16 位数据我们就说它是 16 位 (字长) 的 CPU。Intel 8088 CPU 内部具有 16 位的处理能力, 但它的总线却是 8 位的, 为了要处理 16 位的数据, 需要两次访问存储器才能获得 16 位的数据。这种 CPU 称为 16 位的 CPU。指令的执行速度, 在大、中、小型计算机中通常以每秒执行多少万条指令来衡量。这当然也不能准确地反映计算机解决实际问题的指令执行速度。因为指令的执行时间有差异, 而不同的问题所使用的指令也是不同的。用执行指令时间长的指令多, 则每秒执行指令的条数就少, 反之数值则多。可见用每秒执行多少万条指令来衡量 CPU 速度也是近似的。微型计算机 CPU 的字长和指令的长度差异较大, 所以用每秒执行几万条指令来衡量它的速度是很不

准确的。为此,只能根据 CPU 所用的时钟频率来相对地衡量 CPU 的工作速度。

把实现中央处理单元各项功能的电子元件集成在一片芯片上的 CPU 叫微处理器 (Microprocessor)。微处理器并不是微型计算机,它仅仅是微型计算机上的一个主要部件。

### (2) 存储器

存储器是计算机的“记忆”部分。分成内部存储器(简称内存)和外部存储器(简称外存)。这里我们所指的存储器是内存。从硬件的体系结构来看外存是计算机的一种外部设备,内存的存取速度比外存快,但它的价格比外存贵。目前计算机的内存都是由体积小,功耗低的半导体集成电路的存储器组成。这种存储器在断电时,所保存的信号将全部消失。计算机的外存是磁介质存储器,如磁盘、磁带。这种存储器容量大,价格低,断电后保存的信号并不消失,但它的工作速度较内存存储器慢。两种存储器的配合使用,比较理想地解决了计算机的“记忆”功能。

计算机的内存类似于旅馆的房间,对于每一个单元给予一定的编号,以便在存取信息时,根据这种叫做地址的编号来决定存取位置。

存储器的主要技术指标是存储容量,即能保持二进制信息位的多少。通常以字节(1个字节是8个二进制位)为单位来度量存储器的容量。由于一般的微型计算机的存储器的容量总在1千个字节以上,所以又用千字节(KB)为单位。例如我们说某一个微型计算机具有64KB的内存,它实际的存储容量是65536字节,为便于叙述我们通常说成64K字节。另一个技术指标是存取周期,即对存储器进行一次读或写操作所需的时间。存取周期通常用微秒( $\mu\text{s}$ )或毫秒(ns)表示( $1\text{ns} = 10^{-3}\mu\text{s} = 10^{-9}\text{s}$ )。一般微型计算机内存的存取周期约为几百微秒。

### (3) 输入输出接口电路外部设备

计算机的外部设备是复杂的。有机械的、电子的、光电的等等。从外设的电信号来看有模拟的、数字的。外设的工作速度差异也很大。因此,如何把外设同 CPU 很好地连接起来是一个比较复杂的问题。随着电子技术的发展,尤其是大规模集成电路的发展,已经设计和生产了许多功能复杂的标准接口电路。使外设通过接口电路同 CPU 进行连接,从而使硬件设计变得比较容易了。

计算机的外部设备通常有终端、打印设备、外部存储设备等。终端包括显示器和键盘。

### (4) 系统总线

系统总线是 CPU 同存储器或 I/O 设备之间传送信息的连接线。它包括数据总线、地址总线和控制总线。

## 2. 软件系统

计算机对数据的计算和处理通过执行一条一条的机器指令来实现。为了使计算机自动地进行工作,我们要预先把指令以及原始数据放在计算机的存储器里面,在给定了这些指令存放在内存中的起始地址以后,计算机就能够一条一条地依次取出指令,执行指令,完成我们要求计算机所要执行的任务。指令的有序集合叫做程序。

早期的计算机,人们就是用二进制形式的机器码来编写程序。二进制形式的机器码



很难记忆,程序的可读性很差。为了便于编写程序逐步发明了用助记符号来代替机器码的汇编语言程序,或者叫汇编语言,以及比汇编语言更为方便又具有通用性的高级语言。语言是人与计算机交换信息的工具。目前,在多数微型机上都配有 PASCAL、COBOL、BASIC、C 等高级语言。

无论是汇编语言还是高级语言都不能直接为计算机所理解。为使计算机理解,就必须通过各自的“翻译”。我们把程序设计人员按某种语言的语法规则编写的程序叫做源程序。例如我们用汇编语言编写的程序叫汇编源程序,它必须经过叫做“汇编程序”的一种“翻译”程序对源程序进行汇编,即将源程序译成机器语言程序。这种机器语言程序叫做目标程序。同样用 FORTRAN 语言编写的程序叫 FORTRAN 的源程序,起到“翻译”作用的程序叫 FORTRAN 的编译程序。所以我们说 FORTRAN 这种语言是编译性的语言。PASCAL、COBOL、C、ALGO、Ada 等语言都是属于编译性的语言。还有一种语言,它并不产生目标文件,而是对源程序的语句进行逐条翻译,每翻译一条就执行一条,这类语言我们叫做解释性的语言。如 BASIC 语言,现在也有编译性的 BASIC 语言。上述的汇编源程序、汇编程序、高级语言的源程序、编译语言等都是以文件的形式保存在磁盘或磁带上的。

要使用计算机除了要有程序以及对程序的翻译或解释外,还要解决对计算机的管理问题。这个管理包括对硬件资源的管理,也包括对软件资源的管理。硬件资源的管理涉及到内存的分配和 I/O 设备的使用等;软件资源的管理,主要是对上述程序文件的管理。这方面的任务是由操作系统来完成。操作系统实际也是程序。它同硬件、各种编译、用户程序、操作人员之间的关系如图 1-2 所示。

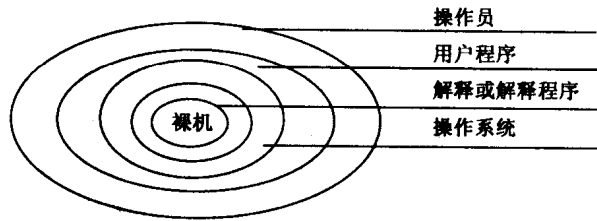


图 1-2 硬软件的关系

可见,只有计算机的硬件系统而没有相应的操作系统和各种高级语言的编译或解释程序以及用户程序的计算机是不能工作的。我们把操作系统和各种语言的编译或解释程序等叫做系统软件。系统软件加上用户程序叫做计算机的软件系统。计算机不同于其他电子设备的地方正是在于如果要使用计算机就必须编写程序。所以它是比较“难使用”的。所谓“难”是把它同电视机、收音机这类设备比较而言的。这类电子设备只要接通电源,会调几个旋钮就行了,而使用计算机则必须编写用户程序。从某种意义上来说计算机是“万能”的。所谓“万能”正是可以编写并执行解决各式各样问题的程序的结果。

下面再讲解一下计算机的工作过程:

无论用什么语言编写的程序,最终总是要转换成机器语言程序的。机器语言程序被调入计算机的内存以后,CPU 从该程序的第一条指令开始,首先取出指令,然后分析并执行指令这样周而复始的进行,至到遇到程序的首条结束指令为止。