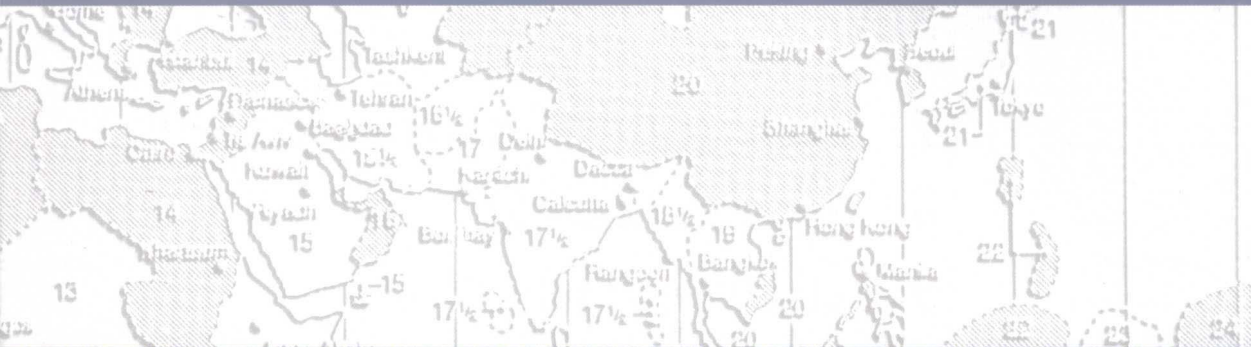




卓越系列·国家示范性高等职业院校特色教材



# 32位微机原理、汇编语言 及接口技术

32BIT MICROCOMPUTER PRINCIPLE,  
ASSEMBLY LANGUAGE  
AND INTERFACE TECHNOLOGY

主编 姜 荣

卓越系列·国家示范性高等职业院校特色教材

# 32 位微机原理、汇编语言 及接口技术

主 编 姜 荣  
副主编 蹇木伟 赵 林  
高 明

 天津大学出版社  
TIANJIN UNIVERSITY PRESS

## 内容简介

本书以 80486 微处理器为背景,讲述 32 位微型计算机原理、汇编语言程序设计和接口技术。全书主要内容有微型计算机系统的组成、微型处理器的指令系统、汇编语言程序设计、微处理器的外部特性与存储器的扩展、基本输入/输出接口、中断控制接口、并行接口、串行接口与通信技术、数模与模数转换电路等。

本书可选作高等职业院校微机原理与接口技术、微机原理及应用和汇编语言程序设计等课程的教材或参考书,也适用于相关专业本科生、计算机应用开发人员、希望了解计算机应用技术的普通读者和培训班学员。

本书配套教学课件,可供教学人员教学使用和学生自学参考。

## 图书在版编目(CIP)数据

32 位微机原理、汇编语言及接口技术/姜荣主编. —天津:天津大学出版社,2009.7

ISBN 978-7-5618-3064-2

I.3… II.姜… III.①微型计算机—理论②微型计算机—接口③汇编语言—程序设计 IV.TP36 TP313

中国版本图书馆 CIP 数据核字(2009)第 106526 号

出版发行 天津大学出版社  
出版人 杨欢  
地 址 天津市卫津路 92 号天津大学内(邮编:300072)  
电 话 发行部:022-27403647 邮购部:022-27402742  
网 址 www.tjup.com  
印 刷 天津泰宇印务有限公司  
经 销 全国各地新华书店  
开 本 169mm × 239mm  
印 张 17.25  
字 数 368 千  
版 次 2009 年 7 月第 1 版  
印 次 2009 年 7 月第 1 次  
印 数 1 - 3 000  
定 价 31.00 元

---

凡购本书,如有缺页、倒页、脱页等质量问题,烦请向我社发行部门联系调换

版权所有 侵权必究

# 前 言

微型计算机技术发展非常迅速,从最初的 8 位微型计算机到今天的 64 位微型计算机,各种新技术、新产品不断出现,在各个领域得到广泛应用,因而掌握微机系统知识及接口技术就显得尤为重要。

根据高等职业教育“实用、通用、够用”的特点,在 2006 年 1 月由姜荣、王芹编写并由西安交通大学出版社出版的《微机原理、汇编语言及接口技术》的基础上,结合近几年的项目教学法及基于工作过程的教学理念,编写了这本教材。本书内容丰富,深入浅出,注重实用,出版的目的是为广大高职院校师生提供一本适用、实用的教材。我们从 2008 年 5 月就开始筹划,收集资料,到了 2009 年 5 月,完成本书的写作。我们的设想和出版理念得到天津大学出版社周小明老师的大力支持,这坚定了我们编写的信心。

本书以 Intel 80486 CPU 和 IBM PC 系列微机为对象,详细、系统地介绍了 32 位微型计算机的基本原理和接口技术。全书共分 9 个项目,内容如下:

- 项目一 微型计算机系统的组成
- 项目二 微处理器的指令系统
- 项目三 汇编语言程序设计
- 项目四 微处理器的外部特性与存储器的扩展
- 项目五 基本输入/输出接口
- 项目六 中断控制接口
- 项目七 并行接口
- 项目八 串行接口与通信技术
- 项目九 数模与模数转换电路

编写时,考虑到高职学生和其他读者的知识水平,采用了浅显、明晰、循序渐进的叙述方法;书中插图简洁,并与文字论述相对照,以方便阅读。本书融原理与应用于一体,有利于教师组织课堂教学和学生课下自学。本书共组织了 9 个项目实战,内容如下:

- 项目实战 1 微型计算机系统的组装与调试
- 项目实战 2 一个简单汇编程序的设计
- 项目实战 3 一个汇编语言程序的设计与调试
- 项目实战 4 一个半导体存储器系统的扩展
- 项目实战 5 设计一个 DMA 控制器接口电路并编程
- 项目实战 6 8259A 中断控制器的应用
- 项目实战 7 并行接口的应用
- 项目实战 8 利用 8251A 设计一串行接口

## 项目实战9 A/D、D/A 转换

本书的理论授课 84 学时,实训 60 学时。

本书由山东省威海职业学院姜荣老师策划并任主编,山东大学蹇木伟老师、山东凯文科技职业学院赵林老师、北洋电气集团的高级工程师高明任副主编。项目一、二、四、五及附录由姜荣编写,项目三由王芹编写,项目六由刘振山编写,项目七由赵林编写,项目八由高明编写,项目九由蹇木伟编写。威海职业学院的张淑红、刘文峰老师也参与了本书部分内容的编写。

感谢威海职业学院领导、教务处领导、信息工程系和机电工程系的领导和同人的支持与帮助,也感谢其他参与编写的兄弟院校的领导和同人。天津大学出版社领导与编辑全力支持本书出版,在此一并致谢。

由于编者学识水平有限,书中难免存在错误及不妥之处,诚心期待同行与读者批评指教。

编者

2009 年 5 月

## 目 录

<b>项目一 微型计算机系统的组成</b> .....	(1)
1.1 项目开篇 微型计算机系统是如何工作的 .....	(1)
1.2 项目备战 微型计算机的系统组成 .....	(2)
任务 1.2.1 认识微型计算机的硬件系统 .....	(3)
任务 1.2.2 了解微型计算机的软件系统 .....	(6)
任务 1.2.3 掌握计算机的信息表示方法 .....	(6)
1.3 项目实战 微型计算机系统的组装与调试 .....	(11)
1.4 项目决战 深入理解微型计算机系统的工作原理 .....	(12)
1.5 项目挑战 微型计算机系统的发展现状及其展望 .....	(13)
<b>项目二 微处理器的指令系统</b> .....	(14)
2.1 项目开篇 一个简单的汇编程序的编写 .....	(14)
2.2 项目备战 汇编语言程序格式与指令系统 .....	(15)
任务 2.2.1 了解汇编语言程序编写格式 .....	(16)
任务 2.2.2 了解 80486 的内部结构 .....	(19)
任务 2.2.3 了解汇编语言的寻址方式 .....	(23)
任务 2.2.4 掌握汇编语言指令系统 .....	(29)
2.3 项目实战 一个简单汇编程序的设计 .....	(62)
2.4 项目决战 深入理解汇编程序格式和微处理器系统 .....	(62)
2.5 项目挑战 了解奔腾系列的指令系统和工作特点 .....	(64)
<b>项目三 汇编语言程序设计</b> .....	(65)
3.1 项目开篇 汇编语言的程序设计与调试 .....	(65)
3.2 项目备战 汇编语言的程序设计 .....	(68)
任务 3.2.1 理解常量、变量和标号的含义及应用 .....	(68)
任务 3.2.2 掌握顺序程序设计的方法与技巧 .....	(74)
任务 3.2.3 掌握分支程序设计的方法与技巧 .....	(76)
任务 3.2.4 掌握循环程序设计的方法与技巧 .....	(81)
任务 3.2.5 理解子程序设计原则和方法 .....	(85)
任务 3.2.6 学会运用调试程序 .....	(95)
3.3 项目实战 一个汇编语言程序的设计与调试 .....	(98)
3.4 项目决战 进一步掌握汇编语言的程序设计技巧和调试方法 .....	(98)
3.5 项目挑战 了解现在常用的编程工具及方法 .....	(100)
<b>项目四 微处理器的外部特性与存储器的扩展</b> .....	(101)

4.1	项目开篇	存储器的扩展与应用	(101)
4.2	项目备战	微处理器的外部特性与存储器的扩展	(102)
	任务 4.2.1	了解 80486 的工作模式	(102)
	任务 4.2.2	了解 80486CPU 的外部引脚	(103)
	任务 4.2.3	了解总线技术	(110)
	任务 4.2.4	了解半导体存储器芯片的结构和技术指标	(112)
	任务 4.2.5	了解常用的几种半导体存储器的工作原理	(115)
	任务 4.2.6	掌握半导体存储器与 CPU 的连接方法	(121)
4.3	项目实战	一个半导体存储器系统的扩展	(127)
4.4	项目决战	进一步掌握 CPU 外部特性和存储器扩展的相关知识	(128)
4.5	项目挑战	了解现在常用的编程工具及方法	(128)
<b>项目五 基本输入/输出接口</b>			(129)
5.1	项目开篇	什么是基本输入/输出接口	(129)
5.2	项目备战	基本接口与数据传送方式	(130)
	任务 5.2.1	了解 I/O 端口的编址与译码	(130)
	任务 5.2.2	了解数据传送方式	(132)
	任务 5.2.3	掌握 DMA 控制器 8237A 的应用	(141)
5.3	项目实战	设计一个 DMA 控制器接口电路并编程	(156)
5.4	项目决战	进一步理解接口电路的传送原理	(156)
5.5	项目挑战	了解奔腾级微型计算机的 DMA 接口技术	(157)
<b>项目六 中断控制接口</b>			(158)
6.1	项目开篇	什么是中断系统	(158)
6.2	项目备战	可编程中断控制器 8259A 的相关知识	(159)
	任务 6.2.1	掌握什么是中断向量表	(159)
	任务 6.2.2	了解可编程中断控制器 8259A 内部结构及引脚功能	(163)
	任务 6.2.3	掌握 8259A 的中断过程	(165)
	任务 6.2.4	了解 8259A 的中断管理方式	(166)
	任务 6.2.5	掌握 8259A 的编程及应用	(169)
6.3	项目实战	8259A 中断控制器的应用	(177)
6.4	项目决战	进一步掌握中断和中断控制器的相关知识	(177)
6.5	项目挑战	进一步了解中断的相关知识	(178)
<b>项目七 并行接口</b>			(179)
7.1	项目开篇	8255A 和 8254 的应用	(179)
7.2	项目备战	可编程并行接口 8255A 和 8254	(180)
	任务 7.2.1	了解 8255A 的内部结构及外部引脚	(180)
	任务 7.2.2	掌握 8255A 的控制字与初始化编程	(182)

任务 7.2.3	掌握 8255A 的工作方式及编程	(184)
任务 7.2.4	掌握 8255A 与 CPU 的接口及应用	(188)
任务 7.2.5	了解可编程定时器 8254 的内部结构及外部引脚	(193)
任务 7.2.6	了解 8254 的工作方式	(195)
任务 7.2.7	掌握 8254 的控制字及编程方法	(200)
任务 7.2.8	掌握 8254 的应用	(203)
7.3	项目实战 并行接口的应用	(206)
7.4	项目决战 进一步掌握并行接口的相关知识	(206)
7.5	项目挑战 进一步了解并行接口的相关知识	(207)
<b>项目八</b>	<b>串行接口与通信技术</b>	<b>(208)</b>
8.1	项目开篇 串行接口与串行通信	(208)
8.2	项目备战 串行口的相关知识	(210)
任务 8.2.1	了解串行接口标准	(210)
任务 8.2.2	了解串行接口的工作原理	(212)
任务 8.2.3	了解可编程串行接口芯片 8251A 内部结构	(214)
任务 8.2.4	认识并了解 8251A 的引脚及其功能	(216)
任务 8.2.5	掌握 8251A 的命令字与初始化编程	(219)
任务 8.2.6	掌握 8251A 的接口技术与应用	(224)
8.3	项目实战 利用 8251A 设计一串行接口	(225)
8.4	项目决战 进一步理解串行通信的含义	(226)
8.5	项目挑战 了解 8251A 的其他应用	(226)
<b>项目九</b>	<b>数模与模数转换电路</b>	<b>(227)</b>
9.1	项目开篇 控制系统中的模拟接口	(227)
9.2	项目备战 模/数和数/模转换器及其接口技术	(229)
任务 9.2.1	掌握 D/A 转换器及其接口技术	(229)
任务 9.2.2	掌握 A/D 转换器及其接口技术	(236)
任务 9.2.3	了解 D/A 和 A/D 器件的选择	(244)
9.3	项目实战 A/D、D/A 转换	(244)
9.4	项目决战 进一步掌握模/数和数/模知识	(245)
9.5	项目挑战 进一步了解数模和模数转换的相关知识	(245)
<b>附录 A</b>	<b>80X86 常用指令表</b>	<b>(246)</b>
<b>附录 B</b>	<b>常用 DOS 功能调用 (INT 21H)</b>	<b>(251)</b>
<b>附录 C</b>	<b>常用 ROM - BIOS 功能调用</b>	<b>(255)</b>
<b>附录 D</b>	<b>汇编语言的开发方法</b>	<b>(259)</b>
<b>附录 E</b>	<b>调试程序 DEBUG 的使用方法</b>	<b>(262)</b>
<b>参考文献</b>		<b>(268)</b>



# 项目一 微型计算机系统的组成

## 本项目导读

学习本项目前,应首先学习前序课程“数字电路”、“计算机文化基础”或“计算机应用”。在本项目中用到“数字电路”中关于数制、码制、寄存器、存储器、译码器等相关知识;用到“计算机文化基础”或“计算机应用”中关于计算机应用、分类及其他相关知识。

本项目从微型计算机基本结构和工作原理出发,重点介绍了微型计算机硬件系统、软件系统的基本组成及微型计算机中信息的表示。通过本项目的学习,可以了解计算机有关知识,为后续内容的学习打下良好的基础。

## 学习建议

在了解微型计算机发展历史和应用的基础上,把重点放在数制转换、补码运算、信息表示和计算机系统结构上。

## 学习进度

本项目教学安排6学时,其中理论4学时,动手实践2学时。



## 1.1 项目开篇 微型计算机系统是如何工作的

计算机技术是20世纪发展最快的技术之一,自1946年第一台电子计算机问世以来,它的发展可谓一日千里。在六十多年的历史里,它经历了由电子管计算机、晶体管计算机、集成电路计算机到大规模、超大规模集成电路计算机四代的更替。而目前,已有“非冯·诺依曼”计算机和“神经”计算机的研制计划,并取得了一定的进展。

微型计算机作为计算机的一种,具有体积小、价格低、工作可靠、使用方便、通用性强等特点。其主要应用有数值计算、数据处理及信息管理和计算机辅助设计、过程控制、人工智能(AI)等几个方向。

图1.1所示为一台个人计算机的外观图,从中可以看出计算机主要由主机、显示器、键盘、鼠标以及能听音乐的耳机或音箱等组成。

那么微型计算机具体由哪些元器件组成,又是如何工作的呢?

图1.2(a)~(i)依次展示了主机箱内部结构、主板、CPU、内存、硬盘、声卡、显



图 1.1 典型个人计算机的外观图

卡、鼠标和键盘。

其工作原理可描述为:用户根据要完成的任务预先编好程序,再通过输入设备(如键盘等)将程序送入存储器中。微型计算机开始工作后,首先将该程序在存储器中的起始地址送入微处理器中的程序计数器(PC)中,微处理器根据PC中的地址值找到对应的存储单元,并取出存放在其中的指令操作码送入微处理器中的指令寄存器(IR)中,由指令译码器(ID)对操作码进行译码,并由微操作控制电路发出相应的微操作控制脉冲序列去取出指令的剩余部分(如果指令不止1个字节的长度),同时执行指令赋予的操作功能。在取指过程中,每取出1个单元的指令,PC自动加1,形成下一个存储单元的地址。以上为一条指令的执行过程,如此不断重复,直至执行完最后一条指令为止。

综上所述,微型计算机的基本工作过程是执行程序的过程,也就是CPU自动从程序存放的第一个存储单元起,逐步取出指令、分析指令,并根据指令规定的操作类型和操作对象,执行指令规定的相关操作。如此重复,周而复始,直至执行完程序的所有指令,从而实现程序的基本功能,这就是微型计算机的基本工作原理。该工作原理可用图1.3描述。

## 1.2 项目备战 微型计算机的系统组成

由前面所述可知,通常所说的计算机,准确地说应该是计算机系统,是由硬件系统和软件系统组成的,按人的要求接收和存储信息,自动进行数据处理和计算,并输出结果信息的机器系统。

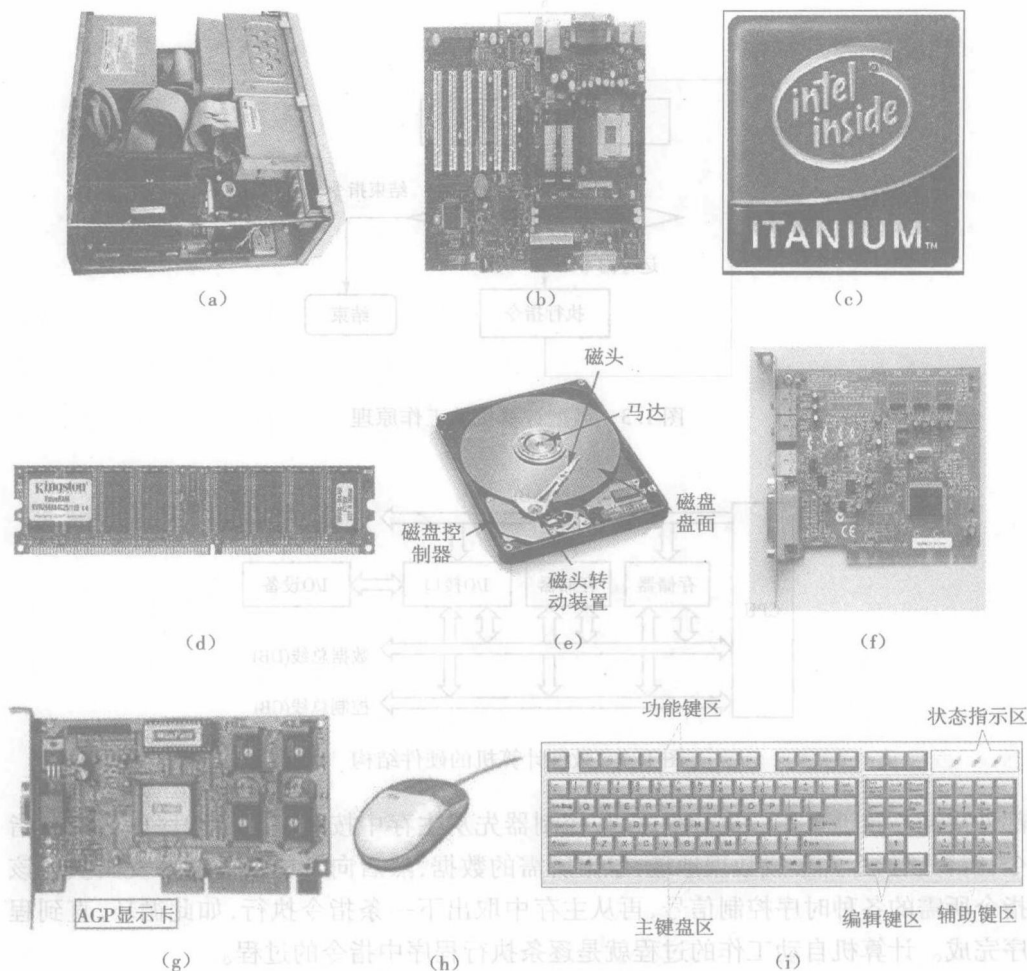


图 1.2 计算机的内部结构

(a) 主机箱内部结构; (b) 主板; (c) CPU; (d) 内存; (e) 硬盘; (f) 声卡; (g) 显卡; (h) 鼠标; (i) 键盘

### 任务 1.2.1 认识微型计算机的硬件系统

图 1.4 为典型微型计算机硬件系统的构成框图,它由微处理器(CPU)、存储器、I/O 接口和 I/O 设备、系统总线等组成。

#### 1. 微处理器

微处理器及其支持电路是微机系统的核心,又称 CPU,它对系统的各个部件进行统一的协调和控制。它是采用大规模或超大规模集成电路技术做成的芯片,芯片内集成了控制器、运算器和若干高速存储单元,即寄存器组。

控制器的主要作用是使整个计算机能够自动地执行程序,并控制计算机各功能

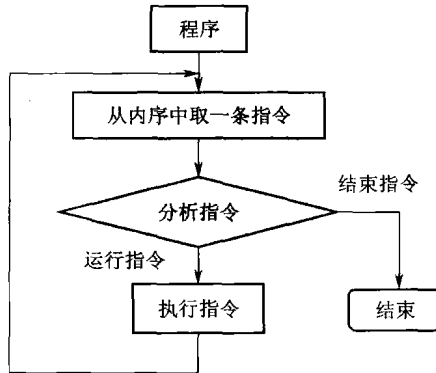


图 1.3 微型计算机的工作原理

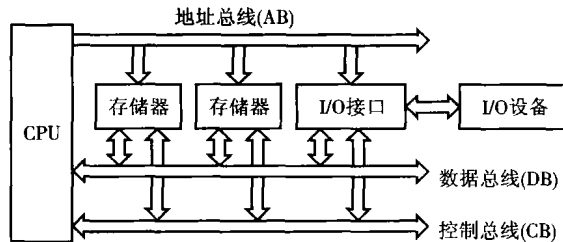


图 1.4 微型计算机的硬件结构

部件协调一致地工作。执行程序时,控制器先从主存中按顺序取出程序中的一条指令,解释该指令并形成数据地址,取出所需的数据,然后向其他功能部件发出执行该指令所需的各种时序控制信号,再从主存中取出下一条指令执行,如此循环,直到程序完成。计算机自动工作的过程就是逐条执行程序中指令的过程。

运算器是计算机中执行各种算术运算和逻辑运算的部件,也叫算术逻辑单元。

寄存器组是 CPU 内部的临时存储单元,其数量根据不同的 CPU 而异。每个寄存器按其功能都有专门的名称和符号,如 32 位微机中存放数据的寄存器名为 EAX、EBX、ECX、EDX 等,存放指令地址的指令寄存器名为 EIP,存放状态信息的标志寄存器名为 EFLAGS 等。

## 2. 存储器

存储器是计算机中具有记忆能力的部件,它能根据地址接收和保存指令或数据,并能根据命令提供有关地址的指令或数据。

微机上的存储器分为“主存”和“辅存”两类。前者主要由半导体存储器组成,造价高,速度快,但容量小,主要用来存放当前正在运行的程序和正待处理的数据;后者主要由磁盘、光盘等存储器构成,造价低,容量大,信息可长期保存,但速度慢,主要用来存放暂不运行的程序和暂不处理的数据。前者被安排在机内的电路板上,CPU 可

以通过总线直接存取,因而也称“内存”;后者被安装在主机箱内或主机箱外,CPU 通过 I/O 接口对其进行存取,所以也称“外存”。

在计算机系统中,存储器是信息更新速度极快的设备。它的外形体积制作得越来越小,容量却越来越大,速度也越来越快,价格越来越低,寿命越来越长,并且一个系统所采用存储器类型也逐渐增多,形成了如图 1.5 所示的存储体系。

### 3. I/O 设备和 I/O 接口

I/O 设备是指微机上配备的输入/输出设备,也称外部设备或外围设备,简称外设,其功能是为微机提供具体的输入/输出手段。

微机上配置的标准输入设备和标准输出设备分别是指键盘和显示器,二者又合称为控制台。此外,常见的输入设备还有鼠标器、数字化仪、扫描仪等;常见的输出设备还有绘图仪、打印机等。作为外部存储器驱动装置的磁盘驱动器,既可看做是一个输出设备又可看做是一个输入设备。随着多媒体技术的发展,发声器、触摸屏、声音图像识别器等输入/输出设备也已经逐步普及。

I/O 接口的功能就是使各种外设的工作速度、驱动方式和 CPU 实现匹配,通过接口电路完成信号变换、数据缓冲以及与 CPU 联络等工作。在微机系统中,较复杂的 I/O 接口电路一般都被做在电路插板上,这种电路插板又被称为“卡”(Card)。

### 4. 系统总线

总线是连接计算机各部分的一组公共传输线,是计算机传送信息的通道。系统总线是指从微处理器引出的若干信号线,CPU 通过它们与存储器和 I/O 设备进行信息交换。

在系统总线中,有传送地址信息的“地址总线”(Address Bus, AB),有传送数据信息的“数据总线”(Data Bus, DB),也有传送控制信息的“控制总线”(Control Bus, CB)。

在一个系统中,有控制和使用总线能力的设备称为“主控器”或“总线请求设备”,如 CPU、DMA 控制器以及协处理器等,而连在总线上的存储器和 I/O 设备则是被访问和控制的对象,被称为“被控器”。

由于系统总线是传送信息的公共通道,因此非常繁忙。其使用特点是:

①在某一时刻,只能有一个主控器控制系统总线;

②在连接系统总线的各个设备中,某时刻只能有一个发送者向总线发送信号,但可以有多个设备从总线上同时获得信号。

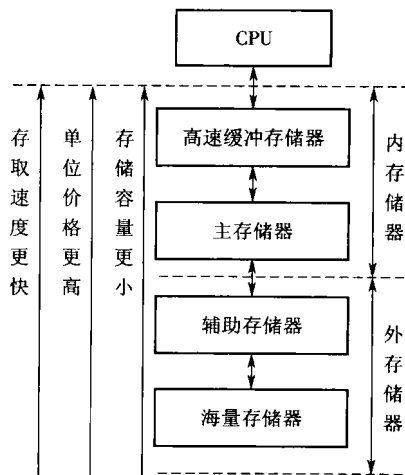


图 1.5 存储器的存储体系

正是由于采用了“总线结构”,才使微机系统具有了组态灵活、扩展方便的特点。

### 任务 1.2.2 了解微型计算机的软件系统

软件是指使计算机运行所需的程序和有关文档。其中,程序是计算任务处理对象和处理规则的描述;文档是与软件研制、维护和使用有关的资料。软件使用户面对的不再是单纯的机器(裸机),而是一台抽象的逻辑机器(虚拟机)。通俗地说,软件是用户与机器的接口。计算机的软件系统由系统软件和应用软件组成。

#### 1. 系统软件

系统软件是用来管理计算机、简化程序设计方法、提高计算机使用效率、发挥和扩大计算机功能的软件。它们主要有操作系统(如 DOS、Windows 等)、语言处理程序(如各种编译程序、解释程序、汇编程序等)、数据库系统(如 Visual Foxpro 等)、网络系统(如基于 TCP/IP 协议的 UNIX、Novell Net Ware、Microsoft Windows NT 以及 IBM OS/2 等)等等。

操作系统(Operating System, OS)是直接运行在裸机上的最基本的系统软件,任何其他软件都必须在 OS 的支持下才能运行。OS 是一个庞大的管理控制程序,它大致包括处理器管理、存储管理、文件管理和作业管理等四个管理功能。

本课程的应用程序是建立在磁盘操作系统 MS-DOS 或 Windows 9x(或 XP)中的 MS-DOS 环境的基础上,利用微软宏汇编程序 MASM 5. X 以上版本和其连接程序 LINK 生成。

#### 2. 应用软件

应用软件是在计算机硬件和系统软件的支持下,为解决各类实际问题而设计的软件。它们主要有:字处理软件(如 Word、WPS 等)、电子表格软件(如 Excel 等)、绘图软件(如 AutoCAD、3DS 等)、课件制作软件(如 PowerPoint、ToolBook、Authorware Professional 等)、网络通信软件(如 IE、Netscape 等)。

本课程的应用程序开发过程中,将利用文本编辑程序编写汇编语言源程序,如 DOS 的全屏幕编辑程序 Edit 或 Windows 的记事本 Notepad。

### 任务 1.2.3 掌握计算机的信息表示方法

既然软件系统是计算机系统的一个不可或缺的部分,是通过控制信息实现对硬件的操作的,那么在计算机中信息又是如何存在,如何处理的呢?

实际上,计算机中信息主要分为数据信息和指令信息两种。数据信息又分为数值型数据和非数值型数据。非数值型数据中主要是字符数据,如英文字母、运算符、汉字等。所有信息必须经过编码后才能输入到计算机中。

#### 1. 数值型数据的表示

一个数值型数据的完整表示包含三个方面的要素:①采用什么进位计数制?②如何表示一个带符号数,即如何使符号数字化?③如何表示带小数点的数?这些问

题在“计算机文化基础”(或“计算机应用基础”)、“数字电路技术”等课程都有详细的介绍,在此不再赘述,仅仅补充带符号数和小数的表示方法。

### (1)带符号数的表示

在计算机中,有符号数有三种表示形式,即原码、反码和补码。

①原码:二进制数的最高位为符号位,符号位为0表示正数,为1表示负数;其余各位等同于真值的绝对值。

②反码:符号位的用法及正数的表示与原码一样;负数的表示是在原码表示的基础上通过将除符号位以外的各位取反(即0变1,1变0)获得。

③补码:符号位的用法及正数的表示与原码一样;负数的表示是在反码表示的基础上通过加1获得,也可理解为将绝对值的二进制(与机器字长有关,不足的用0代替)整个取反,然后加1。

下面通过例1.1~例1.5说明原码、反码和补码的表示及运算。设例1.1~例1.5中的机器字长为8位。

#### 例1.1 原码、反码、补码的运算

$X = + 010\ 1010$ $[X]_{\text{原}} = 0\ \underline{010\ 1010}$ <div style="text-align: center;">↓          ↓</div> <div style="text-align: center;">符号位    尾数</div> $[X]_{\text{反}} = 0\ \underline{010\ 1010}$ <div style="text-align: center;">↓          ↓</div> <div style="text-align: center;">符号位不变    同原码</div> $[X]_{\text{补}} = 0\ \underline{010\ 1010}$ <div style="text-align: center;">↓          ↓</div> <div style="text-align: center;">符号位不变    同原码</div>	$X = - 010\ 1010$ $[X]_{\text{原}} = 1\ \underline{010\ 1010}$ <div style="text-align: center;">↓          ↓</div> <div style="text-align: center;">符号位    尾数</div> $[X]_{\text{反}} = 1\ \underline{101\ 0101}$ <div style="text-align: center;">↓          ↓</div> <div style="text-align: center;">符号位不变    取反</div> $[X]_{\text{补}} = 1\ \underline{101\ 0110}$ <div style="text-align: center;">↓          ↓</div> <div style="text-align: center;">符号位不变    取反加1</div>
--	--

计算机内部采用“补码”表示数,便于实现加减运算。若结果的符号位为0,表示正数;若结果的符号为1,表示负数。最高位产生进位,则自然丢失。补码的符号位与数值部分一起参加运算,并自动获得结果,所得的结果也是补码。另外补码还可以方便实现长度的扩展。

补码运算的一般公式如下:

①补码的加法:  $[X]_{\text{补}} + [Y]_{\text{补}} = [X + Y]_{\text{补}}$ ;

②补码的减法:  $[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$ ,  $[-Y]_{\text{补}}$  即对  $[Y]_{\text{补}}$  求补即可。

**例1.2** 写出十进制25、32、-25、-32的补码。

$$(25)_{10} = (0001\ 1001)_2$$

正数的补码等于它本身,所以十进制数25的补码是0001 1001。同理,十进制数32的补码是0010 0000。

$(-25)_{10}$ 的补码是对其正数的二进制 $(0001\ 1001)_2$ 按位取反,得1110 0110,末

尾加 1, 即  $1110\ 0110 + 1 = 1110\ 0111$ 。同理得  $(-32)_{10}$  的补码为  $1110\ 0000$ 。

**例 1.3** 用补码运算求解十进制  $32 - 25$ 。

解: 十进制	二进制	
$32$	$0010\ 0000$	; 正数的补码是其本身
$- 25$	$\Rightarrow + 1110\ 0111$	; 负数的补码是反码 + 1
$7$	$1\ 0000\ 0111$	

“1”作为进位位, 自动丢失; “0”作为符号位, 表示结果为正数。

**例 1.4** 用补码运算求解十进制  $32 - (-25)$ 。

解: 十进制	二进制	
$32$	$0010\ 0000$	$0010\ 0000$
$- (-25)$	$\Rightarrow - 1110\ 0111$	$\Rightarrow + 0001\ 1001$
$57$	$0011\ 1001$	

**例 1.5** 用补码运算求解  $-25 - 32$ 。

解: 十进制	二进制
$-25$	$1110\ 0111$
$- 32$	$\Rightarrow 1110\ 0000$
$-57$	$1\ 1100\ 0111$

左起第一个“1”作为进位位, 自动丢失; 第二个“1”作为符号位, 表示结果为负数。

在确定了运算字长和数据的表示方法后, 所能表示的数据范围也就确定了。当运算结果超出能表示的数据范围时, 就会产生溢出。如某机字长 8 位, 采用补码表示, 则定点整数的表示范围为  $-128 \sim +127$ 。当运算结果超出这个数, 即产生了溢出, 会使运算结果不正确。因此, 在运算过程中, 必须判断计算机是否产生溢出。若有溢出则停止或转入中断服务程序进行处理。

### (2) 定点数和浮点数

在计算机中表示小数时根据小数点位置是否固定分为定点表示法和浮点表示法。这两种表示方法不仅关系到小数点的问题, 而且关系到数的表示范围、精度及计算机内部电路的复杂程度。

约定小数点隐含地固定在某个位置不变, 这种表示数的方法叫定点表示法。用定点表示法表示的数叫定点数。原则上, 小数点固定在哪一位并无关系, 但为了方便, 总是把小数点规定在数的最前面或最后面, 即总是把所有的数化为纯小数或纯整数来对待。选择哪一种表示法在计算机硬件上并无区别, 可在程序中约定。

定点数在机器中的表示方法如下:

符号位	尾数
-----	----



而将小数点的位置可以改变的表示数的方法称作浮点表示法。浮点表示法类似于科学计数法,任一数均可通过改变指数部分,使小数点位置发生变动。浮点表示法的一般形式是

$$N = 2^P \times S$$

其中, $P$ 、 $S$ 都是带符号的数, $P$ 称为阶码, $S$ 称为尾数,2则是数制的基数,固定不变,所以是隐含的,在计算机内部实际表示时并不出现。

通常,用一位二进制数 $P_f$ 表示阶码的符号位,当 $P_f = 0$ 时,表示阶码为正;当 $P_f = 1$ 时,表示阶码为负。尾数 $S$ 用一位二进制 $S_f$ 表示尾数的符号,当 $S_f = 0$ 时,表示尾数为正;当 $S_f = 1$ 时,表示尾数为负。

浮点数在机器中的表示方法如下:

阶符 $P_f$	阶码 $P$	尾符 $S_f$	尾数 $S$
----------	--------	----------	--------

例 1.6 把二进制  $N = -9 \times 2^5$  表示为浮点数。

阶符 0	阶码 101	尾符 1	尾数 1 001
------	--------	------	----------

## 2. 字符的表示

字符是非数值信息表示的基础,也可以用它来间接表示数值型数据。

### (1) 二—十进制编码

所谓二—十进制编码就是将十进制数的每一位写成二进制数的形式。因十进制有 0~9 这十个数,但 4 位二进制可编出  $2^4 = 16$  种不同组合状态,取其中的 10 种表示 0~9 这十个数,其余作为伪码舍弃。

二—十进制编码的方案很多,最常用的是根据位权所称的 8421 码,又称 BCD 码。它的编码关系如表 1.1 所示。

表 1.1 BCD 编码表

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

例 1.7 利用 BCD 码表示 73.5。

$$(73.5)_{10} = (0111\ 0011.0101)_{\text{BCD}}$$

### (2) ASCII 码

字符数据包括各种运算符号、关系符号、货币符号、控制符号、字母和数字等。国