



● 高等学校计算机基础教材系列

计算机技术基础

主 编 郝兴伟

Power

学图书馆



高等教育出版社
HIGHER EDUCATION PRESS

高等学校计算机基础教材系列

计算机技术基础

主 编	郝兴伟	副主编	巩裕伟
编 者	刘 捷	张鸿烈	杨占敏
	张立群	刘 毅	焦文江

高等教育出版社

内容提要

本书是为高等院校非计算机专业高年级学生编写的计算机技术基础课程的教材,详细地讲解了计算机的基本硬件技术、软件技术及有关的理论和开发工具。全书分为九大模块,包括:计算机组成及工作原理、计算机操作系统、算法与数据结构、C++语言与面向对象技术、可视化编程、数据库技术、MS SQL Server 关系数据库系统、多媒体技术及应用以及网络应用开发。模块基本上涵盖了计算机学科的主要内容。本书在讲解上注重理论和应用,不但可以提高学生的计算机理论水平,而且具有很强的应用性。

在内容安排上,各个模块没有太强的先后顺序和依赖性,各学校可以根据自己的实际情况选择不同的模块进行教学。另外,该书内容密切结合目前全国计算机等级考试(三级)的要求,学生通过该书的学习,可以为参加全国计算机等级考试(三级)打好基础。

本书可作为高等院校理工科相关专业的计算机技术基础教材。对于那些希望全面了解计算机基本技术,包括硬件技术和软件技术的一般读者,本书也是一本内容比较全面、深度适中的参考书。

图书在版编目(CIP)数据

计算机技术基础/郝兴伟主编. —北京:高等教育出版社, 2003.6 (2004 重印)
ISBN 7-04-012298-7

I. 计... II. 郝... III. 电子计算机—高等学校—教材 IV. TP3

中国版本图书馆 CIP 数据核字 (2003) 第 045279 号

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-82028899

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
印 刷 北京星月印刷厂

开 本 787×1092 1/16
印 张 24.5
字 数 590 000

版 次 2003 年 6 月第 1 版
印 次 2004 年 1 月第 2 次印刷
定 价 28.10 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

本世纪,我国已经把实现国民经济和社会信息化放到了国家发展战略的高度上。因此,在高等学校的人才培养中,对计算机和信息技术的教育已经不能停留在简单的计算机操作和应用这样一个层面,必须在此基础上进行更深入一步的教学研究,使计算机基础教育再上一个台阶。

当前,多数高等院校的计算机公共教学主要以计算机操作为主,另外还根据学科的不同分别开设了一些计算机编程语言课程,其目的在于提高学生的计算机应用水平。在长期的教学实践中,我们感觉到这样的教学模式不利于学生计算机基础理论的培养,限制了学生知识面的拓展,不利于学科融合。只有加强学生对计算机基础理论和技术理解,才能够使学生对计算机硬件的整体构架和工作原理有一个清晰的认识,对计算机软件系统有一个全面的掌握,为学生未来的计算机应用和学科融合打好基础。为此,我们组织编写了本书,其内容涵盖了计算机学科的主要基础课程和专业课程,目的是使学生对计算机学科有一个较为全面的认识。

在本书的组织过程中,我们力求做到:

(1) **内容广泛** 从计算机硬件的组成和工作原理到算法和数据结构,都进行了广泛而较深入的介绍。

(2) **讲解深度适中** 在注意内容广泛的前提下,还特别注意到内容的讲解深度。对于每一部分内容,我们并没有只提出一些概念,而是有着较为全面的讲解,增加了内容的实用性。

(3) **硬件和软件结合** 考虑到大学教学的实际,我们将计算机硬件和软件基础进行了结合,并且以软件讲解为主。

(4) **模块化结构** 由于课程内容较多,因此内容按照模块组织,不同的学科可以按照学科特点选择讲授。

本书全部由计算机学院有着丰富教学经验的专业教师编写。本书所编写的内容进行了多次研讨和试讲。第1章由刘捷老师编写,第2章由张鸿烈老师编写,第3章、第4章和第5章由郝兴伟老师编写,第6章、第7章由杨占敏、张立群老师编写,第8章由刘毅和焦文江老师编写,第9章由巩裕伟老师编写。在本书的编写过程中,孟祥旭教授、王海洋教授、龙世立教授对本书的结构及内容提出了许多非常好的建设性意见,对此深表谢意。

在教学过程中,我们收集了大量的教学素材,制作了完整的课件,课件采用浏览器方式制作,内容丰富,便于网络教学和自学,需要教学课件光盘的教师可与作者联系,作者的E-mail为:hxw@sdu.edu.cn。同时我们建立了计算机基础教学网络平台,教师和学生可以针对学习中遇到的相关问题进行讨论,网站的地址为:<http://jcjy.sdu.edu.cn>。

编 者

2003.5

参考文献

- 1 刘捷. 微型计算机工作原理. 济南: 山东大学出版社, 2002
- 2 汤子瀛等. 计算机操作系统. 西安: 西安电子科技大学出版社, 2000
- 3 陈向群等译. 现代操作系统. 北京: 机械工业出版社, 1999
- 4 王素华. 操作系统教程. 北京: 人民邮电出版社, 1995
- 5 张尧学, 史美林. 计算机操作系统教程. 北京: 清华大学出版社, 2000
- 6 周长林, 左万历. 计算机操作系统教程. 北京: 高等教育出版社, 1994
- 7 屠立德, 屠祁. 操作系统基础. 北京: 清华大学出版社, 2000
- 8 Tanenbaum A S. Modern Operating System, 2nd Edition. Prentice - Hall, 2002
- 9 Silberschatz A, Galvin P. Operating System Concept. John Wiley & Sons Inc., 2002
- 10 张景准. 数据结构. 济南: 山东大学出版社, 1994
- 11 严蔚敏. 数据结构(第二版). 北京: 清华大学出版社, 1998
- 12 Wirth N 著. 算法 + 数据结构 = 程序. 曹德和, 刘椿年译. 北京: 科学出版社, 1984
- 13 Christian K 著. Microsoft C++ 程序设计指南. 王国印, 车达志译. 北京: 清华大学出版社, 1993
- 14 Young M J 著. Visual C++ 4 从入门到精通. 邱仲番等译. 北京: 电子工业出版社, 1997
- 15 数据库系统概论(第二版). 萨师煊, 王珊. 高等教育出版社, 1995
- 16 Silberschatz A. Korth H F etc. Database System Concepts (Third Edition). McGraw - Hill Inc, 2002
- 17 王珊, 陈红. 数据库系统原理教程. 北京: 清华大学出版社, 1998
- 18 张健沛, 戴宗荫, 李玉忱, 杨静. 数据库系统教程. 北京: 中国铁道出版社, 1995
- 19 王能斌. 数据库系统原理. 北京: 电子工业出版社, 2001
- 20 袁鹏飞. SQL Server 数据库应用开发技术. 北京: 人民邮电出版社, 1998
- 21 周绪, 管丽娜等. SQL Server 2000 中文版入门与提高. 北京: 清华大学出版社, 2001
- 22 钟玉琢, 沈洪, 黄荣怀. 多媒体技术. 北京: 清华大学出版社, 1999
- 23 赵子江. 多媒体技术应用教程. 北京: 机械工业出版社, 2001
- 24 王岚, 刘光然, 曲建民. 计算机多媒体技术. 天津: 南开大学出版社, 2001
- 25 郝兴伟. 计算机技术及应用. 北京: 高等教育出版社, 2002
- 26 靳慧峰. 新概念 JSP 网络应用教程. 北京: 科海集团公司, 2001
- 27 范国平. 新概念 XML 教程. 北京: 科海集团公司, 2001
- 28 沈兆阳. Java 与 XML 数据库整合应用. 北京: 清华大学出版社, 2002
- 29 黄理等. JSP 深入编程. 北京: 希望电子出版社, 2001
- 30 吴艾. XML/JSP 网页编程教程. 北京: 希望电子出版社, 2001
- 31 殷兆麟. Java 语言程序设计. 北京: 高等教育出版社, 2002

目 录

第1章 微型计算机的组成及工作原理	(1)	2.2.4 系统调用及类型	(28)
1.1 微型计算机系统的三个层次	(1)	2.3 进程及处理机管理	(29)
1.2 微型计算机系统的硬件结构	(1)	2.3.1 进程的定义及特征	(29)
1.2.1 三总线结构及框图	(2)	2.3.2 进程调度	(32)
1.2.2 微型计算机主要组成部分 的结构及功能	(3)	2.3.3 进程同步及信号量机制	(33)
1.2.3 指令与程序的执行	(7)	2.3.4 进程通信	(35)
1.2.4 程序执行过程举例	(8)	2.3.5 死锁	(36)
1.3 微型计算机系统的主要性能指标	(9)	2.4 存储管理	(37)
1.3.1 字长	(9)	2.4.1 存储管理的任务和功能	(38)
1.3.2 存储器容量	(10)	2.4.2 分区分配存储管理	(38)
1.3.3 运算速度	(10)	2.4.3 页式存储管理	(39)
1.3.4 外设扩展能力	(10)	2.4.4 请求分页式存储管理	(40)
1.3.5 软件配置情况	(10)	2.4.5 分段存储管理	(41)
1.4 计算机系统结构的分类	(11)	2.4.6 段页式存储管理	(42)
1.4.1 单指令流单数据流(SISD)	(11)	2.5 文件管理	(43)
1.4.2 多指令流单数据流(MISD)	(11)	2.5.1 文件管理的任务和功能	(43)
1.4.3 单指令流多数据流(SIMD)	(11)	2.5.2 文件分类	(43)
1.4.4 多指令流多数据流(MIMD)	(11)	2.5.3 文件的逻辑结构	(44)
1.5 WINTEL 平台	(12)	2.5.4 文件目录	(44)
1.5.1 Intel x86 处理器系列	(12)	2.5.5 文件存取控制	(44)
1.5.2 Intel x86 16/32 位微处理器	(13)	2.5.6 文件物理结构的组织	(44)
1.6 RISC 与 CISC 结构	(16)	2.6 设备管理	(46)
1.6.1 RISC 的提出	(17)	2.6.1 设备的类型	(47)
1.6.2 RISC 的特征	(18)	2.6.2 设备与控制器之间的接口	(47)
1.6.3 RISC 与 CISC 的竞争	(21)	2.6.3 设备分配	(48)
习题一	(22)	2.6.4 设备分配算法	(48)
第2章 计算机操作系统	(24)	2.6.5 设备独立性	(49)
2.1 计算机操作系统概述	(24)	2.6.6 逻辑设备名到物理设备名的 映射	(49)
2.1.1 什么是操作系统	(24)	2.6.7 SPOOLing 技术	(49)
2.1.2 操作系统的特征	(24)	习题二	(50)
2.2 操作系统接口	(25)	第3章 算法与数据结构	(52)
2.2.1 用户界面的任务和功能	(25)	3.1 计算机的问题求解模型	(52)
2.2.2 用户界面的设计特点	(25)	3.2 C 语言基础	(52)
2.2.3 操作系统常用命令分类简介	(25)	3.2.1 C 语言程序结构	(52)

3.2.2 C语言基本符号	(53)	3.9 排 序	(114)
3.2.3 数据、数据类型、类型定义 和标准函数	(53)	3.9.1 基本概念	(114)
3.2.4 运算符、表达式、赋值运算和赋值 表达式	(61)	3.9.2 插入排序	(115)
3.2.5 赋值语句和输入/输出	(63)	3.9.3 交换排序	(116)
3.2.6 分支语句	(63)	3.9.4 选择排序	(119)
3.2.7 循环控制语句	(65)	3.9.5 归并排序	(120)
3.2.8 函数	(67)	3.9.6 基数排序	(121)
3.2.9 预处理命令	(69)	3.10 文 件	(123)
3.2.10 文件操作	(69)	3.10.1 文件的类别	(123)
3.3 算法与数据结构的基本概念	(71)	3.10.2 文件的物理结构	(123)
3.3.1 概念与术语	(71)	3.10.3 文件的操作	(124)
3.3.2 数据的逻辑结构	(71)	习题三	(125)
3.3.3 数据的存储结构	(72)	第4章 C++语言与面向对象技术	(126)
3.3.4 算法分析与设计	(72)	4.1 从C到C++语言	(126)
3.4 线性结构	(74)	4.1.1 新的C语言的习惯	(126)
3.4.1 线性表	(75)	4.1.2 const类型说明	(128)
3.4.2 堆栈	(76)	4.1.3 volatile类型说明	(129)
3.4.3 队列	(77)	4.1.4 void类型	(129)
3.4.4 串	(77)	4.1.5 引用类型	(130)
3.5 多维数组与广义表	(78)	4.1.6 初始化	(131)
3.5.1 多维数组及其存储	(78)	4.1.7 内连(inline)函数	(131)
3.5.2 矩阵的压缩存储	(79)	4.1.8 缺省函数参数	(132)
3.5.3 广义表	(82)	4.1.9 new和delete函数	(132)
3.6 树	(85)	4.1.10 标准模板库 STL	(133)
3.6.1 树的基本概念及存储结构	(85)	4.2 面向对象的思维方式	(133)
3.6.2 二叉树	(88)	4.3 类与对象	(133)
3.6.3 常用操作及算法	(91)	4.3.1 类的定义	(134)
3.7 图	(93)	4.3.2 封装和抽象	(136)
3.7.1 图的概念	(93)	4.3.3 静态成员	(136)
3.7.2 图的存储	(96)	4.3.4 友元	(137)
3.7.3 图的遍历	(101)	4.3.5 构造函数和析构函数	(138)
3.7.4 图的应用	(103)	4.4 类的继承性与派生类	(138)
3.8 查找	(106)	4.4.1 公有派生类和私有派生类	(139)
3.8.1 基本概念	(106)	4.4.2 多重基类	(139)
3.8.2 顺序查找	(107)	4.4.3 抽象基类	(140)
3.8.3 折半查找	(108)	4.5 虚函数和多态性	(141)
3.8.4 二叉排序树和平衡二叉树	(110)	4.5.1 虚函数的特征	(144)
3.8.5 B ₊ 树和B ⁺ 树	(111)	4.5.2 虚函数调用	(145)
3.8.6 哈希表(散列表)	(113)	4.5.3 多态性	(145)
		4.6 构造函数和析构函数——规则 和激活次序	(146)

4.6.1 非派生类的构造函数	(146)	5.7.5 支持滚动	(194)
4.6.2 派生类的构造函数	(147)	5.7.6 实验文本的滚动	(194)
4.6.3 构造函数规则和特殊情况	(148)	5.8 绘制图形	(197)
4.6.4 析构函数规则和特殊情况	(148)	5.8.1 设备情景对象	(198)
习题四	(148)	5.8.2 选择绘图工具	(198)
第 5 章 可视化编程	(151)	5.8.3 设置绘图属性	(200)
5.1 使用 Visual C++ 6.0	(151)	5.8.4 鼠标操作与鼠标消息	(200)
5.1.1 Microsoft Visual Studio 6.0 开发平台	(151)	5.8.5 绘图	(201)
5.1.2 使用 Visual Studio 编写和调试 C++ 程序	(154)	5.8.6 应用举例	(201)
5.2 用 Visual C++ 开发 Windows 应用 系统	(156)	5.9 文件	(208)
5.2.1 Windows 程序的特点	(156)	5.9.1 在文档类中实现数据的临时 保存	(208)
5.2.2 Windows 的编程特点和编程 方法	(156)	5.9.2 将数据保存到磁盘文件中	(210)
5.2.3 一个典型的 Windows 应用程序	(158)	习题五	(214)
5.3 MFC 基础类库	(160)	第 6 章 数据库技术	(215)
5.3.1 基类 CObject	(161)	6.1 数据库基础知识	(215)
5.3.2 应用程序框架类	(161)	6.1.1 数据库技术的产生与发展	(215)
5.4 MFC Windows 程序的创建及消息 机制	(162)	6.1.2 数据库系统的组成	(217)
5.4.1 使用 MFC AppWizard 创建 Windows 程序框架	(162)	6.1.3 数据库的体系结构与数据独 立性	(217)
5.4.2 FMC 的消息机制	(168)	6.1.4 数据库管理系统	(218)
5.5 菜单、工具栏和状态栏	(169)	6.1.5 数据模型	(219)
5.5.1 菜单	(169)	6.2 关系数据库	(220)
5.5.2 工具栏	(173)	6.2.1 关系数据模型的基本 概念、定义	(220)
5.5.3 状态栏	(175)	6.2.2 关系代数	(222)
5.6 对话框和控件	(176)	6.2.3 关系代数实例	(226)
5.6.1 创建对话框资源	(176)	6.3 关系数据库规范化理论	(227)
5.6.2 创建对话框类	(178)	6.3.1 关系模型的设计问题	(227)
5.6.3 定义对话框的消息处理函数	(180)	6.3.2 函数依赖	(228)
5.6.4 编写控件处理函数	(182)	6.3.3 范式	(230)
5.6.5 使用对话框	(184)	6.3.4 Armstrong 公理系统	(233)
5.6.6 处理对话框的输入	(185)	6.4 关系数据库标准语言 SQL	(234)
5.7 文本的输入/输出	(188)	6.4.1 SQL 概述	(234)
5.7.1 设备情景对象	(188)	6.4.2 SQL 的数据定义	(235)
5.7.2 输出文本的基本步骤	(189)	6.4.3 SQL 的数据查询	(237)
5.7.3 键盘输入	(190)	6.4.4 SQL 的数据更新	(241)
5.7.4 虚拟键码	(193)	6.4.5 嵌入式 SQL	(243)
		6.5 数据库设计与管理	(245)
		6.5.1 数据库设计概述	(245)
		6.5.2 实体联系模型	(247)

6.5.3 E-R 模型转换成关系模型	(251)	8.3.2 视频信号格式	(294)
6.5.4 数据库的完整性	(251)	8.3.3 多媒体数据数字的压缩 及主要算法	(295)
6.5.5 并发处理与并发控制	(252)	8.4 常用媒体播放软件	(297)
习题六	(254)	8.4.1 Winamp	(297)
第 7 章 MS SQL Server 关系数据库系统	(257)	8.4.2 RealPlayer	(299)
7.1 SQL Server 概述	(257)	8.4.3 Windows Media Player	(299)
7.1.1 SQL Server 的发展简史	(257)	8.5 图像处理技术概述	(299)
7.1.2 SQL Server 的客户机/服务器 系统	(257)	8.5.1 Photoshop 简介	(300)
7.1.3 SQL Server 的安装与设置	(259)	8.5.2 图像的类型、大小和分辨率	(300)
7.2 TRANSACT-SQL 的使用	(262)	8.5.3 色彩模式和模型	(301)
7.2.1 TRANSACT-SQL 编程语言	(263)	8.5.4 色调、色相、饱和度和对比度	(302)
7.2.2 TRANSACT-SQL 的数据类型	(263)	8.6 Photoshop 的基本操作	(302)
7.2.3 执行 TRANSACT-SQL 语句的 方法	(267)	8.6.1 Photoshop 的窗口界面	(302)
7.2.4 查询	(269)	8.6.2 图像文件操作技术	(302)
7.3 存储过程	(271)	8.7 绘图操作	(303)
7.3.1 存储过程简介	(271)	8.7.1 图层的基本功能	(304)
7.3.2 创建和执行存储过程	(272)	8.7.2 画笔属性及混合模式	(304)
7.3.3 维护存储过程	(272)	8.7.3 绘图工具的使用	(304)
7.3.4 存储过程应用举例	(273)	8.7.4 选择工具的使用	(305)
7.4 触发器	(274)	8.7.5 填充工具的使用	(305)
7.4.1 触发器概述	(274)	8.7.6 图像的变换	(306)
7.4.2 触发器的创建和维护	(275)	8.7.7 图像色彩的调整	(306)
7.4.3 触发器的应用	(276)	8.8 图层、通道、路径	(306)
7.5 开放式数据库互连——ODBC	(278)	8.8.1 图层	(306)
7.5.1 ODBC 结构	(279)	8.8.2 通道	(307)
7.5.2 ODBC 接口	(281)	8.8.3 路径	(308)
习题七	(282)	8.9 文字输入及特效字制作实例	(309)
第 8 章 多媒体技术及应用	(283)	8.9.1 图像中文字的输入操作	(309)
8.1 多媒体技术概述	(283)	8.9.2 文字属性的设定和编辑	(309)
8.1.1 多媒体的定义	(283)	8.9.3 特殊效果字体制作实例	(309)
8.1.2 多媒体硬件设备	(283)	习题八	(310)
8.1.3 多媒体软件	(290)	第 9 章 网络应用开发	(312)
8.2 音频信号的获取与处理	(290)	9.1 超文本标记语言 HTML	(312)
8.2.1 声音的概念及基本参数	(290)	9.1.1 HTML 标记语法和文档结构	(312)
8.2.2 语音数据的数字化	(291)	9.1.2 HTML 文档标记	(315)
8.2.3 常见的声音文件和音乐 文件格式	(292)	9.1.3 列表	(316)
8.3 视频信号的获取与处理	(293)	9.1.4 表格	(316)
8.3.1 常用图像格式	(293)	9.1.5 表单	(318)
		9.2 扩展标记语言 XML	(321)
		9.2.1 XML 概述	(321)

9.2.2 XML 文档的创建	(322)	9.4 JSP 技术及应用	(359)
9.2.3 文档类型定义 DTD	(325)	9.4.1 JSP 概述	(359)
9.2.4 使用 CSS 格式化数据	(327)	9.4.2 JSP 的基本语法	(361)
9.2.5 可扩展样式语言 XSL	(329)	9.4.3 内置对象	(361)
9.2.6 XML 的文档对象模型	(335)	9.4.4 指令元素	(363)
9.3 Java 语言及应用	(339)	9.4.5 动作元素	(365)
9.3.1 Java 程序开发环境	(340)	9.4.6 JSP 的数据库访问	(367)
9.3.2 基础知识	(340)	9.4.7 JSP 对 XML 文档的访问	(372)
9.3.3 类和对象	(343)	习题九	(376)
9.3.4 Java 小程序 Applet	(350)	参考文献	(378)
9.3.5 Java 的多线程机制	(354)		

第1章 微型计算机的组成及工作原理

计算机硬件是计算机系统的重要组成部分,它是计算机系统功能实现的物理载体,所有的计算机程序功能最终都必须由计算机硬件来实现。本章将以微型计算机为讲解对象,介绍计算机硬件系统的基本组成和工作原理。

1.1 微型计算机系统的三个层次

“微电脑”,“微机”是日常俗语,准确的称谓应是——微型计算机系统。微型计算机系统从全局到局部有三个层次:微型计算机系统——微型计算机——微处理器,为了以后学习时不致混淆,首先有必要了解这三个层次的确切含义。

1) 微处理器

微处理器(Microprocessor)也常称为微处理机,它并不是微型计算机,但它是微型计算机的核心部件。微处理器包括算术逻辑部件(Arithmetic Logic Unit, ALU)、控制部件(Control Unit, CU)和寄存器组(Registers, R)三个基本部分以及内部总线。相当于一般计算机系统结构中的运算器和控制器的组合。通常由几片(早期)或一片LSI、VLSI器件组成。其实,它就是一般计算机系统概念的中央处理器(CPU)。

2) 微型计算机

微型计算机(Micro Computer)是以微处理器为核心,加上由大规模集成电路制作的存储器M(ROM和RAM)、I/O(输入/输出)接口和系统总线组成的。该层次即微型计算机系统的主板。

3) 微型计算机系统

微型计算机系统(Micro Computer System)是以微型计算机为核心,再配以相应的外围设备、电源、辅助电路和控制微型计算机工作的软件而构成的完整的计算系统。软件分为系统软件和应用软件两大类。系统软件是用来支持应用软件的开发与运行的,它包括操作系统、标准实用程序和各種语言处理程序等。应用软件是用来为用户解决具体应用问题的程序及有关的文档和资料。

要注意,在上述的三个层次中,单纯的微处理器不是计算机,单纯的微型计算机也不是完整的微型计算机系统,它们都不能独立工作,只有微型计算机系统才是完整的信息处理系统,才有实用意义,才可以正常工作。

1.2 微型计算机系统的硬件结构

1.2.1 三总线结构及框图

目前的各种微型计算机系统产品,无论是简单的单片机、单板机系统,还是较复杂的个人计

算机(PC机)系统,从硬件体系结构来看,采用的基本上是计算机的经典结构——冯·诺依曼结构。这种结构的特点是:

- (1) 由运算器、控制器、存储器、输入设备和输出设备五大部分组成;
- (2) 数据和程序以二进制代码形式不加区别地存放在存储器中,存放位置由地址指定,地址码也为二进制形式;
- (3) 控制器是根据存放在存储器中的指令序列,即程序来工作的,并由一个程序计数器(即指令地址计数器)控制指令的执行。控制器具有判断能力,能根据计算结果选择不同的动作流程。

由此可见,任何一个微型机系统都是由硬件和软件两大部分组成的。图 1.1 所示为具有这种结构特点的微型计算机典型硬件组成框图。微处理器 MPU 中包含了上述的运算器和控制器;RAM 和 ROM 为存储器;I/O 外设及接口是输入、输出设备的总称。各组成部分之间通过系统总线,包括地址总线 AB、数据总线 DB、控制总线 CB 联系在一起。

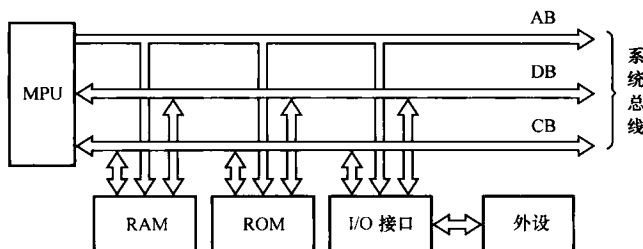


图 1.1 微型计算机的结构框图

有时也将微型计算机的这种系统结构称为单(套)总线结构,简称总线结构。采用总线结构可使微型计算机的系统构造比较方便,并且具有更大的灵活性和更好的可扩展性、可维护性。根据系统总线组织方法的不同,可把总线结构分为单(套)总线、双(套)总线、双重总线三类,如图 1.2 所示。

图 1.2(a)所示的是单(套)总线结构。图 1.1 所示的实际上就是这种结构,在单总线结构中,系统存储器 M 和 I/O 使用惟一的一套信息通路,因而微处理器对存储器和 I/O 的读写只能分时进行。大部分中低档微机都是采用这种结构,因为它的逻辑结构简单、成本低廉、实现容易。

图 1.2(b)所示的是双(套)总线结构的示意图。I/O 和 M 各自具有到 MPU 的总线通路,这种结构的 MPU 可以分别在两套总线上同时与 M 和 I/O 口交换信息,相当于展宽了总线带宽,提高了总线的数据传输速率。目前有的单片机和高档微机就是采用这种结构。不过,在这种结构中,MPU 要同时管理其与 M 和 I/O 的通信,这势必加重 MPU 在管理方面的负担,为此,通常采用专门的 I/O 处理芯片,即所谓的智能 I/O 接口来履行 I/O 管理任务,以减轻 MPU 的负担。

图 1.2(c)所示的是双重总线结构。在这种结构中,主 MPU 通常通过局部总线访问局部 M 和局部 I/O,这时的工作方式与单总线情况是一样的。通常,MPU 也作为主设备访问全局 M 和 I/O。当其他并列微处理器需要对全局 M 和全局 I/O 访问时,必须由总线控制逻辑统一安排才

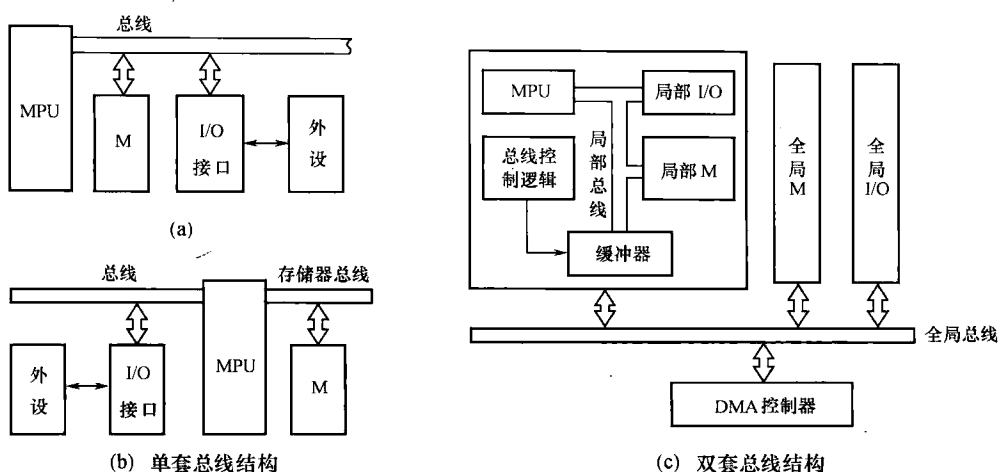


图 1.2 系统三类总线组织方法

能进行,这时该微处理器就是系统的主控设备。如图 1.2(c)中的 DMA 控制器也可成为系统的主控设备,全局 I/O 外设和全局 M 之间便可利用系统总线进行 DMA 操作。在其他处理器进行全局 I/O 的同时,MPU 还可以通过局部总线对局部 M 或局部 I/O 进行访问。显然,这种结构可以实现双重总线上并行工作,并且对等效总线带宽的增加、系统数据处理和数据传输效率的提高,效果更明显。目前各种高档微型机和工作站基本上都是采用这种双重总线结构。

1.2.2 微型计算机主要组成部分的结构及功能

微机的主要组成部分有微处理器(MPU)、存储器、I/O 设备接口和总线。

1. 微处理器(MPU)

微处理器是微型计算机的运算和指挥控制中心。不同型号的微型计算机,其性能的差别首先在于其微处理器性能的不同,而微处理器性能又与它的内部结构、硬件配置有关。每种微处理器有其特有的指令系统。但无论哪种微处理器,其内部基本结构总是相同的,都有控制器、运算器和内部总线及缓冲器三大部分,每部分又各由一些基本部件组成,如图 1.3 所示。这种结构是以第二代 MPU 为基础的。下面介绍各基本部件的功能。

1) 算术逻辑单元 ALU(Arithmetic Logic Unit)

ALU 是运算器的核心。它是以全加器为基础,辅之以移位寄存器及相应控制逻辑组合而成的电路,在控制信号的作用下可完成加、减、乘、除四则运算和各种逻辑运算。

2) 累加器 ACC、累加锁存器和暂存器

累加器 ACC(ACCumulator)通常简称为累加器 A,它实际上是通用寄存器中使用最频繁、功能最强的一个。由于它总是提供送入 ALU 的两个运算操作数之一,且运算后的结果又总是送回它之中,这就决定了它与 ALU 的联系特别紧密,因而把它和 ALU 一起归入运算器中,而不归在通用寄存器组中。

累加锁存器的作用是防止 ALU 的输出通过累加器 A 直接反馈到 ALU 的输入端。

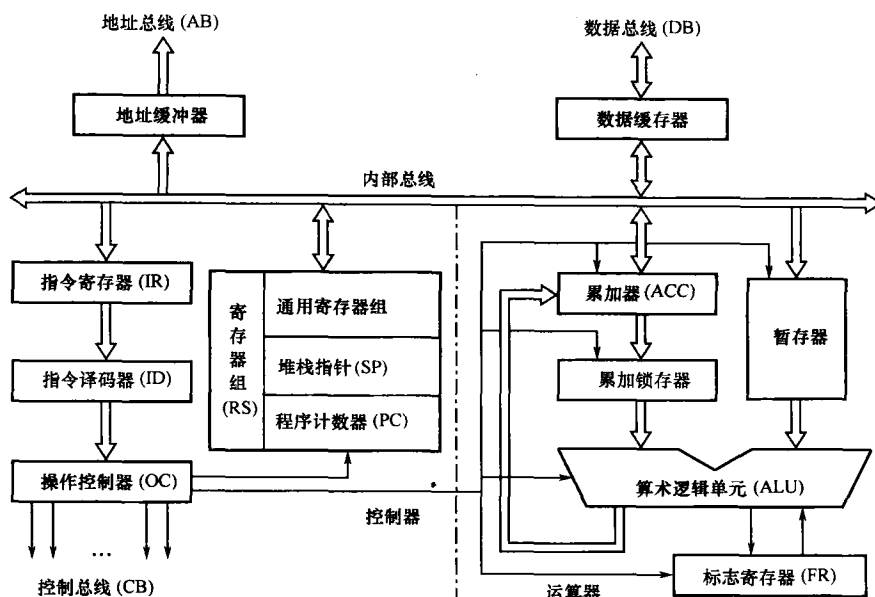


图 1.3 微处理器典型结构示意图

暂存器的作用与累加器 A 有点相似,都是用于保存操作数,只是操作结果只保存于累加器 A,而不保存到暂存器。

3) 标志寄存器 FR (Flags Register)

FR 用于寄存 ALU 操作结果的某些重要状态或特征,如是否溢出、是否为零、是否为负、是否有进位、是否有偶数个 1 等。每种状态或特征用一位来标志。由于 ALU 的操作结果存放在累加器 A 中,因而 FR 也反映了累加器 A 中所存放数据的特征。FR 中的状态标志常为 CPU 执行后续指令时所用,例如,根据某种状态标志来决定程序是顺序执行还是跳转执行。

在 80386 以后的处理器中,FR 除存放状态标志外,还存放控制处理器工作方式的控制标志和系统标志。

4) 寄存器组 RS (Register Set 或 Registers)

RS 实质上是微处理器的内部 RAM,因受芯片面积和集成度所限,其容量(即寄存器数目)不可能很多。寄存器组可分为专用寄存器和通用寄存器。专用寄存器的作用是固定的,图 1.3 中的堆栈指针 SP、程序计数器 PC、标志寄存器 FR 即为专用寄存器。通用寄存器用途广泛并可由程序员规定其用途。通用寄存器的数目因微处理器而异,如 8086 有 AX、BX、CX、DX、BP、SP、SI、DI 共 8 个 16 位通用寄存器,80386/80486 有 EAX、EBX、ECX、EDX、ESI、EDI、EBP、ESP 共 8 个 32 位通用寄存器,等等。

由于有了这些寄存器,在需要重复使用某些操作数或中间结果时,就可将它们暂时存放在寄存器中,避免对存储器的频繁访问。通用寄存器除了可高效地存储数据外,还可作为间址、基址、变址寻址时的地址指针,从而缩短指令长度和指令执行时间,加快 CPU 的运算处理速度,同时也给编程带来方便。因此,在高档微机 CPU 的设计中无不对通用寄存器进行精心设计,RISC

结构 CPU 的设计更将其发挥得淋漓尽致。

除了上述两类程序员可用的寄存器外,微处理器中还有一些不能直接为程序员所用的寄存器,如前述累加锁存器、暂存器和指令寄存器等,它们仅受内部定时与控制逻辑的控制。

5) 堆栈和堆栈指针 SP

在计算机中广泛使用堆栈作为数据的一种暂存结构。堆栈由堆栈区和堆栈指针构成。栈区是一组按先进后出(FILO)或后进先出(LIFO)方式工作的寄存器或存储单元,用于存放数据。当它是由微处理器内部的寄存器组构成时,称为硬件堆栈;当它是由软件在内存中开辟的一个特定 RAM 区构成时,称为软件堆栈。目前绝大多数微处理器都支持软件堆栈。

堆栈指针(Stack Pointer, SP)是用来指示栈顶地址的寄存器,用于自动管理栈区,指示当前数据存入或取出的位置。在堆栈操作中,将数据存入栈区称为“压入”(PUSH);从栈区中取出数据称为“弹出”(POP)。无论是压入还是弹出,只能在栈顶进行。每压入或是弹出一个堆栈元素,栈指针均会自动修改,以便自动跟踪栈顶位置。

SP 的初值由程序员设定。一旦设定初值后,便意味着栈底在内存储器中的位置已经确定,此后 SP 的内容即栈顶位置便由 CPU 自动管理。随着堆栈操作的进行,SP 值会自动变化,其变化方向因栈区的编址方式而异。栈区的编址方式有向下增长型和向上增长型两种。对于向下增长型堆栈,将新数据压入其中时,SP 自动减量,向上移动而指向新的栈顶;当数据从栈中弹出时,SP 自动增量,向下移动而指向新的栈顶。对于向上增长型堆栈则相反。

堆栈主要用于中断处理与过程(子程序)调用。堆栈的“先进后出”操作方式给中断处理和子程序调用/返回(特别是多重中断与多重调用)带来很大方便。在中高档的微机系统中,堆栈分为系统堆栈(由操作系统设置)和用户堆栈(由用户程序员设置)。

6) 程序计数器 PC(Program Counter)

PC 用于存放下一条要执行的指令的地址码。程序中的各条指令一般是按执行的顺序存放在存储器中的。开始时,PC 中的地址码为该程序第一条指令所在的地址编号。在顺序执行指令的情况下,每取出指令的一个字节(通常微处理器的指令长度是不等的,有的只有一个字节,有的是两个或更多个字节),PC 的内容自动加 1,于是,当从存储器取完一条指令的所有字节时,PC 中存放的是下一条指令的首地址。若要改变程序的正常执行顺序,就必须把新的目标地址装入 PC,这称为程序发生了转移。指令系统中有一些指令用来控制程序的转移,称为转移指令。

可见,PC 是维持微处理器有序地执行程序的关键性寄存器,是任何微处理器不可缺少的。

也有一些微处理器(如 80X86 系列 MPU),不是用一个 PC 来直接指示下一条待执行指令的地址,而是用代码段寄存器(CS)和指令指针寄存器(IP)通过内部的转换来间接给出待取出指令的地址。

7) 指令寄存器 IR(Instruction Register)、指令译码器 ID(Instruction Decoder)和操作控制器 OC(Operation Controller)

这三个部件是整个微处理器的指挥控制中心,对协调整个微型计算机有序工作极为重要。它根据用户预先编好的程序,依次从存储器中取出各条指令,放在指令寄存器 IR 中,通过指令译码(分析)确定应该进行什么操作,然后通过操作控制器 OC,按确定的时序向相应的部件发出微操作控制信号。操作控制器 OC 中主要包括节拍脉冲发生器、控制矩阵、时钟脉冲发生器、复位电路和启停电路等控制逻辑。这几个部件对微处理器设计人员来说是关键,微处理器用户可

以不必过多关心。

2. 存储器

存储器可分为内存(主存)和外存(辅存)。内存是微型计算机的主要存储和记忆部件,用以存放即将使用或正在使用的数据(包括原始数据、中间结果和最终结果)和程序。微型机的内存都是采用半导体存储器。外存或称辅存则用于存放暂时不用的程序和数据。

1) 内存单元的地址和内容

内存中存放的数据和程序,从形式上看都是二进制数。内存是由一个个内存单元组成的,每一个内存单元中一般存放一个字节(8位)的二进制信息。内存单元的总数目称为内存容量。

微型机通过给各个内存单元规定不同地址来管理内存。这样,CPU 便能识别不同的内存单元,并正确地对它们进行操作。注意,内存单元的地址和内存单元的内容是两个完全不同的概念,与旅馆中的房间号码和对应的单元房间相类似,不应混淆。

2) 内存操作

CPU 对内存的操作有读、写两种。读操作是 CPU 将内存单元的内容读入 CPU 内部,而写操作是 CPU 将其内部信息传送到内存单元保存起来。显然,写操作的结果改变了被写单元的内容,而读操作则不改变被读单元中原有内容。

3) 内存分类

按工作方式不同,内存可分为两大类:随机存取存储器 RAM(Random Access Memory)和只读存储器 ROM(Read Only Memory)。

RAM 可以被 CPU 随机地读写,又称随机读写存储器。这种存储器用于存放用户装入的程序、数据及部分系统信息。当机器断电后所存信息消失,因此 RAM 归于易失性存储器。

ROM 中的信息只能被 CPU 随机读取,而不能由 CPU 任意随机写入。机器断电后,信息并不丢失,显然 ROM 应属于非易失性存储器。所以,这种存储器主要用来存放各种程序,如汇编程序、各种高级语言解释或编译程序、监控程序、基本 I/O 程序等标准子程序,也用来存放各种常用数据和表格等。ROM 中的内容一般是由生产厂家或用户使用专用设备写入固化的。

3. 输入/输出设备接口

输入/输出(I/O)设备是微型计算机系统的重要组成部分,微型机通过它与外部交换信息,完成实际工作任务。常用输入设备有键盘、鼠标、扫描仪等。常用输出设备有显示器、打印机、绘图仪等。磁带、磁盘、光盘的驱动器既是输入设备,又是输出设备。通常把它们统称为外围设备,简称外设。实际应用中,凡在 CPU 执行指令之前或之后需对信息进行加工的设备均可称为外设。大至一台数控加工中心,小至一只发光二极管等等。

外围设备的种类繁多,结构、原理各异,有机械式、电子式、电磁式等。与 CPU 相比,外围设备的工作速度相差悬殊,处理的信息从数据格式到逻辑时序一般不可能直接兼容,因此,微型机与外围设备间的连接与信息交换不能直接进行,而必须设计一个接口电路作为两者之间的桥梁。其中用于系统本身的接口电路已做在称为主板芯片组的集成电路中,其余的接口电路又称适配器(Adapter),可供用户选择,连接于系统总线的插槽中,控制和驱动外设。

4. 总线

总线实际上是一组导线,是各种公共信号线的集合,用于作为微型计算机中所有各组成部

分传输信息共同使用的“公路”。

1) 数据总线 DB(Data Bus)

数据总线用来传输数据信息,是双向三态(1,0,高阻)总线,CPU既可通过DB从内存或输入设备读入数据,又可通过DB将内部数据送至内存或输出设备。当其他主设备使用DB时,CPU将其设为高阻。

2) 地址总线 AB(Address Bus)

地址总线用于传送CPU发出的地址信息,是单向三态(1,0和高阻)总线。传送地址信息的目的是指明与CPU交换信息的内存单元或I/O设备。当其他主设备使用DB时,CPU将其设为高阻。

3) 控制总线 CB(Control Bus)

控制总线用来传送控制信号、时序信号和状态信息等。其中有的是CPU向内存和外设发出的信息,有的则是内存或外设向CPU发出的信息。可见,CB中每一根线的方向是一定的、单向的,但作为一个整体则是双向的,所以在各种结构框图中,凡涉及到控制总线CB,均以双向线表示。

1.2.3 指令与程序的执行

微型计算机每执行一条指令都是分成三个阶段进行:取指令(Fetch)、分析指令(Decode)和执行指令(Execute)。

(1) 取指令阶段的任务 根据程序计数器PC中的值从存储器读出现行指令,送到指令寄存器IR,然后PC自动加1,指向下一条指令地址。

(2) 分析指令阶段的任务 将IR中的指令操作码译码,分析其指令性质。如指令要求操作数,则寻找操作数地址。

(3) 执行指令阶段的任务 取出操作数,执行指令规定的操作。根据指令不同还可能写入操作结果。

微型计算机程序的执行过程,实际上就是周而复始地完成这三个阶段操作的过程,直至遇到停机指令时才结束整个机器的运行,如图1.4所示。

当然,这三个阶段操作并非在各种微处理器中都是串行完成的,除了早期的8位微处理器外,各种16位机、32位机都可将这三个阶段操作分配给两个或两个以上的独立部件并行完成。例如,8088CPU内有总线接口BIU和执行部件EU,因而在EU中执行一条指令的同时,BIU就可以取下一条指令,它们在时间上是重叠的。至于80386和80486,其并行处理能力则更强,它们采用了六级流水线结构。

由于有了流水线结构,不同指令的取指、分析、执行三个阶段可并行处理,因而处理器在执行程序过程中基本上不需要等待指令执行。

由于程序的指令和数据都存放于内存中,所以在

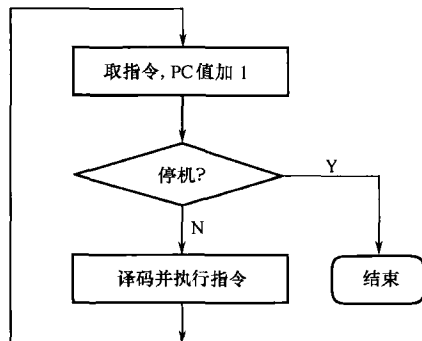


图 1.4 程序执行过程