

面向 **21** 世纪

高等学校计算机类专业系列教材

微型计算机原理

Principles of Microcomputers

王忠民 主编

王忠民 王钰 王晓婕 编著



西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校计算机类专业系列教材

微型计算机原理

Principles of Microcomputers

王忠民 主编

王 钰 王晓婕 编著

西安电子科技大学出版社

2003

内 容 简 介

本书结合大量实例,全面、系统、深入地介绍了微型计算机的工作原理、汇编语言程序设计以及常用可编程接口芯片的工作原理与应用技术。

全书共分为8章,内容包括:微型计算机系统导论,计算机中的数制和编码,80x86微处理器,80x86指令系统,汇编语言程序设计,半导体存储器,输入/输出与中断以及可编程接口芯片及应用等。每章开始给出本章的主要内容、重点难点以及本章内容在整个课程中所处的地位,每章后给出本章小结和练习题。为了便于组织教学和自学,本书配有多媒体CAI教学光盘和实验指导书。

本书结构合理,实例丰富,深入浅出,文笔流畅,既可作为高等院校计算机类各专业“微机原理与接口技术”课程的教材及成人高等教育的教材,也可供广大从事计算机软、硬件开发的工程技术人员参考。

★ 本书配有电子教案,有需要的老师可与我社联系,免费索取。

图书在版编目(CIP)数据

微型计算机原理 / 王忠民主编. —西安:西安电子科技大学出版社, 2003.7

(面向21世纪高等学校计算机类专业系列教材)

ISBN 7-5606-1245-8

I. 微… II. 王… III. 微型计算机—理论—高等学校—教材 IV. TP36

中国版本图书馆CIP数据核字(2003)第037161号

策 划 马武装

责任编辑 雷鸿俊

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)8242885 8201467 邮 编 710071

http://www.xduph.com

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西画报社印刷厂

版 次 2003年7月第1版 2003年7月第1次印刷

开 本 787毫米×1092毫米 1/16 印张21

字 数 496千字

印 数 1~4000册

定 价 22.00元

ISBN 7-5606-1245-8/TP·0654(课)

XDUP 1516001-1

*** 如有印装问题可调换 ***

序

第三次全国教育工作会议以来,我国高等教育得到空前规模的发展。经过高校布局和结构的调整,各个学校的新专业均有所增加,招生规模也迅速扩大。为了适应社会对“大专业、宽口径”人才的需求,各学校对专业进行了调整和合并,拓宽专业面,相应地教学计划、大纲也都有了较大的变化。特别是进入 21 世纪以来,信息产业发展迅速,技术更新加快。面对这样的发展形势,原有的计算机、信息工程两个专业的传统教材已很难适应高等教育的需要,作为教学改革的重要组成部分,教材的更新和建设迫在眉睫。为此,西安电子科技大学出版社聘请南京邮电学院、西安邮电学院、重庆邮电学院、吉林大学、杭州电子工业学院、桂林电子工业学院、北京信息工程学院、深圳大学、解放军电子工程学院等 10 余所国内电子信息类专业知名院校长期在教学科研第一线工作的专家教授,组成了高等学校计算机、信息工程类专业系列教材编审专家委员会,并且面向全国进行系列教材编写招标。该委员会依据教育部有关文件及规定对这两大类的教学计划和课程大纲,目前本科教育的发展变化和相应系列教材应具有的特色和定位以及如何适应各类院校的教学需求等进行了反复研究、充分讨论,并对投标教材进行了认真评审,筛选并确定了高等学校计算机、信息工程类专业系列教材的作者及审稿人,这套教材预计在 2004 年全部出齐。

审定并组织出版这套教材的基本指导思想是力求精品、力求创新、优中选优、以质取胜。教材内容要反映 21 世纪信息科学技术的发展,体现专业课内容更新快的要求;编写上要具有一定的弹性和可调性,以适合多数学校使用。体系上要有所创新,突出工程技术型人才培养的特点,面向国民经济对工程技术人才的需求,强调培养学生较系统地掌握本学科专业必需的基础知识和基本理论,有较强的本专业的基本技能、方法和相关知识,培养学生具有从事实际工程的研发能力。在作者的遴选上,强调作者应在教学、科研第一线长期工作,有较高的学术水平和丰富的教材编写经验;教材在体系和篇幅上符合各学校的教学计划要求。

相信这套精心策划、精心编审、精心出版的系列教材会成为精品教材,得到各院校的认可,对于新世纪高等学校教学改革和教材建设起到积极的推动作用。

系列教材编委会

2002 年 8 月

高等学校计算机、信息工程类专业

系列教材编审专家委员会

主任：杨震（南京邮电学院副院长、教授）
副主任：张德民（重庆邮电学院通信与信息工程学院院长、教授）
韩俊刚（西安邮电学院计算机系主任、教授）
李荣才（西安电子科技大学出版社总编辑、教授）

计算机组

组长：韩俊刚（兼）
成员：（按姓氏笔画排列）
王小民（深圳大学信息工程学院计算机系主任、副教授）
王小华（杭州电子工业学院计算机分院副院长、副教授）
孙力娟（南京邮电学院计算机系副主任、副教授）
李秉智（重庆邮电学院计算机学院院长、教授）
孟庆昌（北京信息工程学院教授）
周娅（桂林电子工业学院计算机系副主任、副教授）
张长海（吉林大学计算机科学与技术学院副院长、教授）

信息工程组

组长：张德民（兼）
成员：（按姓氏笔画排列）
方强（西安邮电学院电信系主任、教授）
王晖（深圳大学信息工程学院电子工程系主任、副教授）
胡建萍（杭州电子工业学院电子信息分院副院长、副教授）
徐祎（解放军电子工程学院电子技术教研室主任、副教授）
唐宁（桂林电子工业学院通信与信息工程系副主任、副教授）
章坚武（杭州电子工业学院通信工程分院副院长、教授）
康健（吉林大学通信工程学院副院长、教授）
蒋国平（南京邮电学院电子工程系副主任、副教授）

总策划：梁家新
策划：马乐惠 云立实 马武装 马晓娟
电子教案：马武装

前 言

本书是根据作者多年从事计算机软、硬件开发和教学实践经验,为满足高等院校电类各专业本科生“微机原理与接口技术”课程教学而编写的。

全书共分为 8 章。第 1 章微型计算机系统导论,通过一个模型机介绍了计算机的五大组成部件、三总线结构及计算机工作过程,以期使读者建立计算机整机概念,为后续章节学习打下基础;第 2 章计算机中的数制和编码,介绍计算机中的数制及其相互转换,带符号数的表示方法,十进制数的二进制编码(BCD 码)以及字符的 ASCII 编码等;第 3 章 80x86 微处理器,重点介绍 8086/8088 CPU 的内部结构、寄存器结构、引脚功能以及存储器管理等,并对具有代表性的 Intel 主流 CPU 系列的最新技术做了适当介绍;第 4 章 80x86 指令系统,在简要介绍 80x86 指令格式后重点介绍操作数和转移地址的寻址方式以及 80x86 指令系统;第 5 章汇编语言程序设计,以 Microsoft 公司的宏汇编程序 MASM 为背景,介绍面向 80x86 的汇编语言程序设计方法;第 6 章半导体存储器,在简要介绍半导体存储器的分类和基本存储单元电路的基础上,重点介绍常用的几种典型存储器芯片及其与 CPU 之间的连接与扩展问题,并对目前广泛应用的几种新型存储器芯片做了简要介绍;第 7 章输入/输出与中断,介绍输入/输出接口的基本概念, CPU 与外设间数据传送的方式,重点介绍中断传送方式及其相关技术;第 8 章可编程接口芯片及应用,介绍与 80x86 系列微处理器配套使用的通用可编程接口芯片的原理及应用技术。

为了便于组织教学和自学,本书在结构上特别注意既便于教师的课堂教学,又便于学生自学。全书结构合理,注重应用,实例丰富,叙述上力求深入浅出;每章开始给出本章的主要内容、重点难点以及本章内容在整个课程中所处的地位;对易犯错误或重点内容书中都给予了特别强调;每章后给出本章小结和大量经过精心准备的练习题;本书还配有多媒体 CAI 教学光盘和实验指导书。

由于本课程的实践性很强,在教学中应特别注意加强实践教学环节,通过大量的上机实践,使学生真正掌握微型计算机的工作原理、汇编语言程序设计方法以及接口电路设计技术等内容,培养学生初步具备软、硬件方面的实际开发能力。

本书第 1 章、第 2 章、第 3 章以及第 8 章由王忠民编写,第 4 章和第 5 章由王钰编写,第 6 章和第 7 章由王晓婕编写。王忠民担任主编并负责本书大纲拟订与统稿工作。

本书的编写得到了西安邮电学院韩俊刚教授、赵亚婉教授的关心,正是由于他们的指导、帮助和大力支持,才使本书得以顺利完成并交付出版。

由于作者水平所限,书中难免存在一些不足与疏漏之处,恳请读者不吝指正。

编 者

2003 年 3 月

目 录

| | | | |
|-------------------------------|----|------------------------------------|-----|
| 第1章 微型计算机系统导论..... | 1 | 习题..... | 31 |
| 1.1 引言..... | 1 | 第3章 80x86 微处理器..... | 33 |
| 1.2 计算机的发展概况..... | 2 | 3.1 80x86 微处理器简介..... | 33 |
| 1.3 微型计算机系统的组成..... | 3 | 3.2 8086/8088 微处理器..... | 35 |
| 1.3.1 硬件..... | 4 | 3.2.1 8086/8088 内部结构..... | 35 |
| 1.3.2 软件..... | 5 | 3.2.2 8086/8088 寄存器结构..... | 38 |
| 1.4 微型计算机硬件系统..... | 5 | 3.2.3 总线周期的概念..... | 42 |
| 1.4.1 微型计算机系统的组成..... | 5 | 3.2.4 8086/8088 引脚及其功能..... | 43 |
| 1.4.2 微处理器..... | 6 | 3.3 8086/8088 存储器和 I/O 组织..... | 49 |
| 1.4.3 存储器..... | 8 | 3.3.1 8086/8088 存储器组织..... | 49 |
| 1.4.4 I/O 接口与输入/输出设备..... | 10 | 3.3.2 8086/8088 的 I/O 组织..... | 54 |
| 1.5 微型计算机的工作过程..... | 10 | 3.4 从 80286 到 Pentium 系列的技术发展..... | 55 |
| 本章小结..... | 13 | 3.4.1 80x86 寄存器组..... | 55 |
| 习题..... | 14 | 3.4.2 80x86 存储器管理..... | 57 |
| 第2章 计算机中的数制和编码..... | 15 | 3.4.3 80286 微处理器..... | 67 |
| 2.1 无符号数的表示及运算..... | 15 | 3.4.4 80386 微处理器..... | 70 |
| 2.1.1 无符号数的表示方法..... | 15 | 3.4.5 80486 微处理器..... | 76 |
| 2.1.2 各种数制的相互转换..... | 17 | 3.4.6 Pentium 系列微处理器..... | 78 |
| 2.1.3 二进制数的运算..... | 18 | 本章小结..... | 83 |
| 2.2 带符号数的表示及运算..... | 19 | 习题..... | 84 |
| 2.2.1 机器数与真值..... | 19 | 第4章 80x86 指令系统..... | 86 |
| 2.2.2 机器数的表示方法..... | 19 | 4.1 8086/8088 指令系统..... | 86 |
| 2.2.3 真值与机器数之间的转换..... | 21 | 4.1.1 8086/8088 指令格式..... | 86 |
| 2.2.4 补码的加减运算..... | 22 | 4.1.2 8086/8088 指令系统的寻址方式..... | 88 |
| 2.2.5 溢出及其判断方法..... | 25 | 4.1.3 8086/8088 指令系统..... | 96 |
| 2.3 信息的编码..... | 26 | 4.2 80x86/Pentium 指令系统..... | 137 |
| 2.3.1 二进制编码的十进制数(BCD 编码)..... | 26 | 4.2.1 80x86 寻址方式..... | 137 |
| 2.3.2 ASCII 字符编码..... | 28 | 4.2.2 80286 增强与增加的指令..... | 138 |
| 2.4 数的定点与浮点表示法..... | 29 | 4.2.3 80386/80486 增强与增加的指令..... | 142 |
| 2.4.1 定点表示..... | 29 | 4.2.4 Pentium 系列处理器增加的指令..... | 146 |
| 2.4.2 浮点表示..... | 30 | 本章小结..... | 148 |
| 本章小结..... | 31 | | |

| | |
|--|------------|
| 习题 | 149 |
| 第 5 章 汇编语言程序设计 | 153 |
| 5.1 汇编语言的基本概念 | 153 |
| 5.2 汇编语言源程序的格式 | 154 |
| 5.2.1 分段结构 | 155 |
| 5.2.2 汇编语言语句的类型和格式 | 156 |
| 5.3 伪指令语句 | 160 |
| 5.3.1 数据定义伪指令 | 161 |
| 5.3.2 符号定义伪指令 | 163 |
| 5.3.3 段定义伪指令 | 164 |
| 5.3.4 过程定义伪指令 | 168 |
| 5.3.5 模块定义与连接伪指令 | 169 |
| 5.3.6 处理器选择伪指令 | 170 |
| 5.4 宏指令语句 | 170 |
| 5.4.1 常用的宏处理伪指令 | 170 |
| 5.4.2 宏指令与子程序的区别 | 172 |
| 5.5 汇编语言程序的上机过程 | 173 |
| 5.5.1 用编辑程序建立汇编语言源程序文件 (ASM 文件) | 173 |
| 5.5.2 用汇编程序 MASM 将 ASM 文件 汇编成目标程序文件(OBJ 文件) | 175 |
| 5.5.3 用连接程序 LINK 生成可执行 程序文件(EXE 文件) | 176 |
| 5.5.4 程序的执行 | 177 |
| 5.5.5 汇编语言和 DOS 操作系统的接口 | 177 |
| 5.5.6 常用系统功能调用和 BIOS 中断 调用 | 178 |
| 5.6 汇编语言程序设计的基本方法 | 186 |
| 5.6.1 顺序程序设计 | 186 |
| 5.6.2 分支程序设计 | 187 |
| 5.6.3 循环程序设计 | 190 |
| 5.6.4 子程序设计 | 193 |
| 5.7 发挥 80386 及其后继机型的优势 | 202 |
| 5.7.1 充分利用高档机的 32 位字长特性 | 202 |
| 5.7.2 通用寄存器可作为指针寄存器 | 205 |
| 5.7.3 与比例因子有关的寻址方式 | 205 |
| 本章小结 | 206 |
| 习题 | 207 |

| | |
|--------------------------------|------------|
| 第 6 章 半导体存储器 | 210 |
| 6.1 概述 | 210 |
| 6.1.1 存储器的分类 | 210 |
| 6.1.2 半导体存储器的分类 | 210 |
| 6.1.3 半导体存储器的主要技术指标 | 211 |
| 6.1.4 半导体存储器芯片的基本结构 | 212 |
| 6.2 随机读写存储器(RAM) | 214 |
| 6.2.1 静态 RAM | 214 |
| 6.2.2 动态 RAM | 217 |
| 6.3 只读存储器(ROM) | 220 |
| 6.3.1 掩膜式只读存储器(MROM) | 220 |
| 6.3.2 可编程只读存储器(PROM) | 221 |
| 6.3.3 可擦除、可再编程的只读存储器 | 221 |
| 6.4 存储器的扩展 | 226 |
| 6.4.1 存储芯片的扩展 | 226 |
| 6.4.2 存储器与 CPU 的连接 | 228 |
| 6.5 几种新型存储器简介 | 231 |
| 本章小结 | 233 |
| 习题 | 233 |
| 第 7 章 输入/输出与中断 | 235 |
| 7.1 I/O 接口概述 | 235 |
| 7.1.1 I/O 接口的作用 | 235 |
| 7.1.2 CPU 与外设交换的信息 | 236 |
| 7.1.3 I/O 接口的基本结构 | 237 |
| 7.1.4 I/O 端口的编址 | 237 |
| 7.2 CPU 与外设之间数据传送的方式 | 238 |
| 7.2.1 程序传送方式 | 239 |
| 7.2.2 中断传送方式 | 241 |
| 7.2.3 直接存储器存取(DMA)传送方式 | 242 |
| 7.3 中断技术 | 244 |
| 7.3.1 中断的基本概念 | 244 |
| 7.3.2 中断优先级和中断的嵌套 | 246 |
| 7.4 8086/8088 中断系统 | 249 |
| 7.4.1 8086/8088 的中断源类型 | 249 |
| 7.4.2 中断向量表 | 251 |
| 7.4.3 8086/8088 的中断处理过程 | 252 |
| 7.4.4 中断服务程序的设计 | 252 |
| 7.5 可编程中断控制器 Intel 8259A | 253 |

| | | | | | |
|-------------------------|------------------------------|-----|---------------|-----------------------|-----|
| 7.5.1 | 8259A 的功能 | 253 | 8.2.3 | 各种工作方式的功能 | 284 |
| 7.5.2 | 8259A 的内部结构及外部引脚 | 254 | 8.2.4 | 8255A 的应用举例 | 288 |
| 7.5.3 | 8259A 的工作方式 | 256 | 8.3 | 串行通信及可编程串行接口芯片 | |
| 7.5.4 | 8259A 的编程 | 258 | | 8251A | 293 |
| 本章小节 | | 265 | 8.3.1 | 串行通信的基本概念 | 294 |
| 习题 | | 266 | 8.3.2 | 串行通信接口及其标准 | 298 |
| 第 8 章 可编程接口芯片及应用 | | 267 | 8.3.3 | 可编程串行接口芯片 8251A | 301 |
| 8.1 | 可编程定时器/计数器芯片 8253/8254 | 267 | 8.3.4 | 8251A 初始化编程 | 306 |
| 8.1.1 | 8253 的结构与功能 | 267 | 8.3.5 | 8251A 应用举例 | 307 |
| 8.1.2 | 8253 的编程 | 270 | 8.4 | 模/数(A/D)与数/模(D/A)转换技术 | |
| 8.1.3 | 8253 的工作方式 | 273 | | 及其接口 | 309 |
| 8.1.4 | 8254 与 8253 的区别 | 277 | 8.4.1 | D/A 转换接口 | 309 |
| 8.1.5 | 8253 应用举例 | 277 | 8.4.2 | A/D 转换接口 | 316 |
| 8.2 | 可编程并行接口芯片 8255A | 280 | 本章小结 | | 323 |
| 8.2.1 | 8255A 的引脚与结构 | 280 | 习题 | | 324 |
| 8.2.2 | 8255A 的工作方式与控制字 | 282 | 主要参考文献 | | 326 |

1

第...章

微型计算机系统导论

本章简要介绍微型计算机的发展历史；根据冯·诺依曼计算机设计思想，主要介绍微型计算机硬件系统的组成，三总线结构(地址总线 AB、数据总线 DB、控制总线 CB)以及组成计算机的五大部件(运算器、控制器、存储器、输入及输出设备)；介绍软件在计算机系统中的作用；通过在模型机上运行一个简单的程序说明计算机的工作过程。

1.1 引言

电子计算机是由各种电子器件组成的，能够自动、高速、精确地进行算术运算、逻辑控制和信息处理的现代化设备。自从其诞生以来，已被广泛应用于科学计算、数据(信息)处理和过程控制等领域。

有关统计资料表明，计算机早期的主要应用领域是科学计算。在科学研究，特别是理论研究中，经过严密的论证和推导，得出非常复杂的数学方程，要求得方程的解。如果手工计算，可能要经过数月、数年的时间，有时甚至是无法完成的。面对这样的难题，计算机可以发挥其强大的威力。计算机在这样的科学计算中，一般用高级语言编程。高级语言是面向用户的，也就是说，用高级语言编写程序比较容易和方便，经过短时间的学习和训练，一般人都能编出功能很复杂的程序。计算机在科学计算中的应用，除了用高级语言编写程序外，与其在信息处理和过程控制领域的应用相比较，还有以下两个特点：第一，它没有很强的实时性要求，虽然使用者在运行程序时也希望能尽快得到运算结果，但对结果产生的时间没有严格的要求，结果产生的迟早不影响结果的有效性。第二，在科学计算中，需要输入计算机的数据，一般不是从某种物理现场实时采集的，不需要有专用的完成数据采集任务的输入设备；同样，计算的结果一般也不完成对外界的控制功能，不需要有专门的输出设备与其他系统相连。

在数据(信息)处理和过程控制应用领域，情况则要复杂得多。除了对系统的实时性有很高的要求外，还要用专门的输入设备将有关信息输入计算机，用专门的输出设备输出处理结果或对被控对象实施控制。因此，仅仅具备高级语言编程方面的知识是远远不够的。实时数据(信息)处理和过程控制要求实时性，希望编写的程序更精练，运行起来更快(一般情况下，对于完成相同的任务，用机器语言或汇编语言编写的程序运行起来比用高级语言编出的程序快得多)；专用的输入/输出设备与计算机的连接和编程控制(称为接口)，更不是只具有高级语言编程知识所能胜任的。为此，必须对计算机的工作原理有更深入的了解，对

计算机的逻辑组成、工作原理、与外界的接口技术以及直接依赖于计算机逻辑结构的机器语言、汇编语言编程方法等需要进一步的学习。本课程就是基于这一目的而设置的。

1.2 计算机的发展概况

计算机的发展，从一开始就是和电子技术，特别是微电子技术密切相关的。人们通常按照构成计算机所采用的电子器件及其电路的变革，把计算机划分为若干“代”来标志计算机的发展。自 1946 年世界上第一台电子计算机 ENIAC 问世以来，计算机技术得到了突飞猛进的发展，在短短的几十年里，计算机的发展已经历了四代：电子管计算机、晶体管计算机、集成电路计算机和大规模、超大规模集成电路计算机。目前，各国正加紧研制和开发第五代“非冯·诺依曼”计算机和第六代“神经”计算机。

微型计算机属于第四代计算机，是 20 世纪 70 年代初期研制成功的。一方面是由于军事、空间及自动化技术的发展，需要体积小、功耗低、可靠性高的计算机；另一方面，大规模集成电路技术的不断发展也为微型计算机的产生打下了坚实的物质基础。

微处理器(MicroProcessor)是微型计算机的核心芯片，它是将计算机中的运算器和控制器集成在一片硅片上制成的集成电路。这样的芯片也被称为中央处理单元(Central Processing Unit)，简称为 CPU。

微型计算机(MicroComputer)是由微处理器(CPU)、存储器和 I/O 接口电路组成的计算机。

30 多年来，微处理器和微型计算机获得了极快的发展，几乎每两年微处理器的集成度就要翻一番，每 2~4 年更新换代一次，现已进入第五代。

1. 第一代——4 位或低档 8 位微处理器

第一代微处理器的典型产品是 Intel 公司 1971 年研制成功的 4004(4 位 CPU)及 1972 年推出的低档 8 位 CPU 8008。它们均采用 PMOS 工艺，集成度约为 2000 只晶体管/片；指令系统比较简单，运算能力差，速度慢(指令的平均执行时间约为 10~20 μs)；软件主要使用机器语言及简单的汇编语言编写。

2. 第二代——中高档 8 位微处理器

微处理器问世以后，众多公司纷纷研制微处理器，逐渐形成以 Intel 公司、Motorola 公司、Zilog 公司产品为代表的三大系列微处理器。第二代微处理器的典型产品有 1974 年 Intel 公司生产的 8080 CPU、Zilog 公司生产的 Z80 CPU、Motorola 公司生产的 MC6800 CPU 以及 Intel 公司 1976 年推出的 8085 CPU。它们均为 8 位微处理器，具有 16 位地址总线。

第二代微处理器采用 NMOS 工艺，集成度约为 9000 只晶体管/片，指令的平均执行时间为 1~2 μs 。指令系统相对比较完善，已具有典型的计算机体系结构以及中断、存储器直接存取(DMA)功能。由第二代微处理器构成的微机系统(如 Apple - II 等)已经配有单用户操作系统(如 CP/M)，并可使用汇编语言及 BASIC、FORTRAN 等高级语言编写程序。

3. 第三代——16 位微处理器

第三代微处理器的典型产品是 1978 年 Intel 公司生产的 8086 CPU、Zilog 公司生产的

Z8000 CPU 和 Motorola 公司生产的 MC6800 CPU。它们均为 16 位微处理器，具有 20 位地址总线。

用这些芯片组成的微型计算机有丰富的指令系统、多级中断系统、多处理机系统、段式存储器管理以及硬件乘/除法等。为方便原 8 位机用户，Intel 公司在 8086 推出后不久便很快推出准 16 位的 8088 CPU，其指令系统与 8086 完全兼容，CPU 内部结构仍为 16 位，但外部数据总线是 8 位的。同时，IBM 公司以 8088 为 CPU 组成了 IBM PC、PC/XT 等准 16 位微型计算机，由于其性价比高，很快就占领了市场。

1982 年，Intel 公司在 8086 基础上研制出性能更优越的 16 位微处理器芯片 80286。它具有 24 位地址总线，并具有多任务系统所必需的任务切换、存储器管理功能以及各种保护功能。同时，IBM 公司以 80286 为 CPU 组成了 IBM PC/AT 高档 16 位微型计算机。

4. 第四代——32 位高档微处理器

1985 年，Intel 公司推出了 32 位微处理器芯片 80386，其地址总线也为 32 位。80386 有两种结构：80386SX 和 80386DX。这两者的关系类似于 8088 和 8086 的关系。80386SX 的内部结构为 32 位，外部数据总线为 16 位，采用 80287 作为协处理器，指令系统与 80286 兼容。80386DX 的内部结构、外部数据总线皆为 32 位，采用 80387 作为协处理器。

1990 年，Intel 公司在 80386 基础上研制出新一代 32 位微处理器芯片 80486，其地址总线仍然为 32 位。它相当于把 80386、80387 及 8 KB 高速缓冲存储器(Cache)集成在一块芯片上，性能比 80386 有较大提高。

5. 第五代——64 位高档微处理器

第五代微处理器的典型产品是 1993 年 Intel 公司推出的 Pentium(奔腾，Intel 586)以及 IBM、Apple 和 Motorola 三家公司联合生产的 Power PC。Pentium 微处理器的数据总线为 64 位，地址总线为 36 位，有两条超标量流水线，两个并行执行单元及双高速缓冲存储器，工作频率有 50 MHz、66 MHz、133 MHz 和 166 MHz 等。Power PC 是一种精简指令集计算机(RISC, Reduced Instruction Set Computer)，也是一种性能优异的 64 位微处理器，其中也采用了先进的超标量流水线技术及双高速缓冲存储器。精简指令集计算机的特点是指令规整，从而使指令译码电路也变得简单，译码速度更快；指令系统中只设置了使用频率较高的指令，因而指令条数少，指挥指令执行的控制逻辑电路也简单，指令执行速度快。与精简指令集计算机对应的是复杂指令集计算机(CISC, Complex Instruction Set Computer)，Intel 公司的 Pentium 微处理器及其以前的微处理器产品都属于 CISC。

1.3 微型计算机系统的组成

微型计算机(MicroComputer)是指以微处理器为核心，配上存储器、输入/输出接口电路等所组成的计算机(又称为主机)。微型计算机系统(Micro Computer System)是指以微型计算机为中心，配以相应的外围设备、电源和辅助电路(统称硬件)以及指挥计算机工作的系统软件所构成的系统。与一般的计算机系统一样，微型计算机系统也是由硬件和软件两部分组成的，如图 1.1 所示。

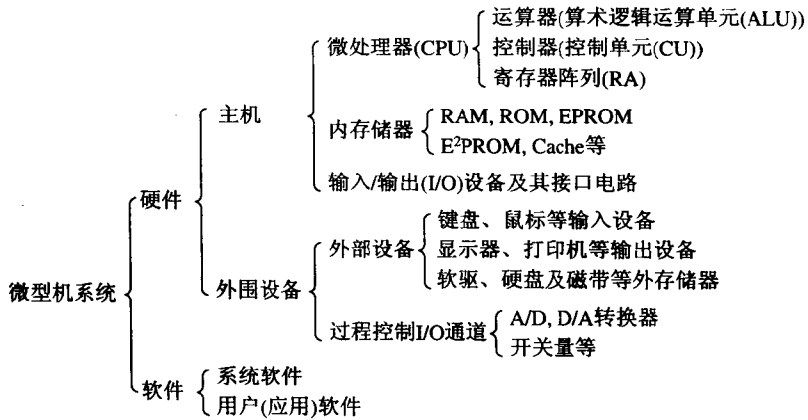


图 1.1 微型计算机系统的组成

1.3.1 硬件

微型计算机硬件系统的组成如图 1.2 所示。其中主要包括微处理器 CPU、存储器和输入/输出(I/O)设备及其接口电路。

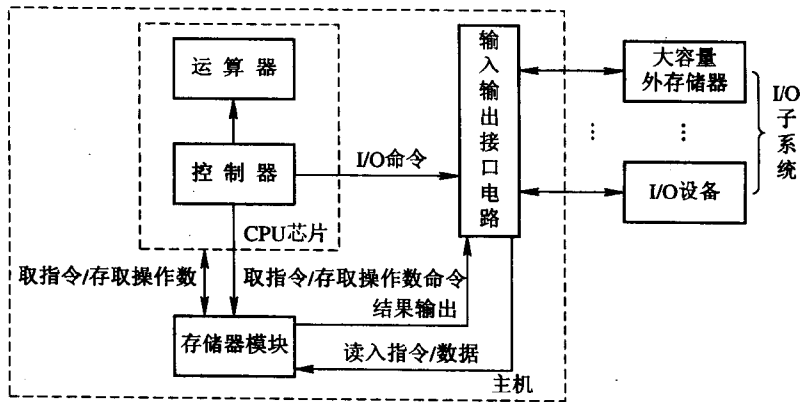


图 1.2 微型计算机硬件系统的组成

运算器和控制器被集成在一片被称之为 CPU 的芯片中，它是微型计算机的运算、控制中心，用来实现算术、逻辑运算，并对全机进行控制。

存储器(又称为主存或内存)用来存储程序或数据，计算机要执行的程序以及要处理的数据都要事先被装入到内存中才能被 CPU 执行或访问。

输入/输出(I/O)接口是微机与输入/输出设备之间的桥梁，这种接口电路又称做“I/O 适配器”(I/O Adapter)。这里有必要强调一下大容量外存储器(外存)与内存储器(主存或内存)之间的关系。由于微型计算机内存容量有限，因而需要使用大容量的外存储器作为内存的后援设备，它的容量可以比内存大很多，但存取速度却比内存慢得多。因此，除必要的系统程序外，一般程序(包括数据)都存放在外存中，只有在运行时，才把它从外存传送到内存的某个区域，再由 CPU 控制执行。

1.3.2 软件

所谓软件,就是为了管理、维护计算机以及为完成用户的某种特定任务而编写的各种程序的总和。计算机的工作就是运行程序,通过逐条地从存储器(内存)中取出程序中的指令并执行指令规定的操作而实现某种特定的功能,因此,软件是微型计算机系统不可缺少的组成部分。微型计算机软件包括系统软件和用户(应用)软件。

用户软件就是用户为解决各种实际问题而编写的程序。

系统软件是指不需要用户干预的,为其他程序的开发、调试以及运行等建立一个良好环境的程序。它主要包括操作系统 OS(Operating System)和系统应用程序。操作系统是一套复杂的系统程序,用于提供人机接口和管理、调度计算机的所有硬件与软件资源。其中最为重要的核心部分是常驻监控程序,计算机启动后,常驻监控程序始终存放在内存中,它接收用户命令,并执行相应的操作;操作系统还包括用于执行 I/O 操作的 I/O 驱动程序,每当用户程序或其他系统程序需要使用 I/O 设备时,通常并不是该程序执行 I/O 操作,而是由操作系统利用 I/O 驱动程序来执行任务;此外,操作系统还包括用于管理存放在外存中大量数据的文件管理程序,文件管理程序和 I/O 驱动程序配合使用,用于文件的存取、复制和其他处理。系统应用程序很多,如各种高级语言的编译程序、汇编程序、诊断和调试程序、文字处理程序、服务性工具程序及数据库管理程序等。

1.4 微型计算机硬件系统

1.4.1 微型计算机系统的组成

到目前为止,计算机仍沿用 1940 年由冯·诺依曼首先提出的体系结构。其基本设计思想为:

① 以二进制形式表示指令和数据。

② 程序和数据事先存放在存储器中,计算机在工作时能够高速地从存储器中取出指令并加以执行。

③ 由运算器、控制器、存储器、输入设备和输出设备等五大部件组成计算机系统。

微型计算机体系结构的特点之一是采用总线结构,通过总线将微处理器(CPU)、存储器(RAM, ROM)、I/O 接口电路等连接起来,而输入/输出设备则通过 I/O 接口实现与微机的信息交换,如图 1.3 所示。

所谓总线,是计算机中各功能部件间传送信息的公共通道,是微型计算机的重要组成部分。它们可以是带状的扁平电缆线,也可以是印刷电路板上的一层极薄的金属连线。所有的信息都通过总线传送。根据所传送信息的内容与作用不同,总线可分为以下三类。

(1) 地址总线 AB(Address Bus): 在对存储器或 I/O 端口进行访问时,通过地址总线传送由 CPU 提供的要访问存储单元或 I/O 端口的地址信息,以便选中要访问的存储单元或 I/O 端口。AB 是单向总线。

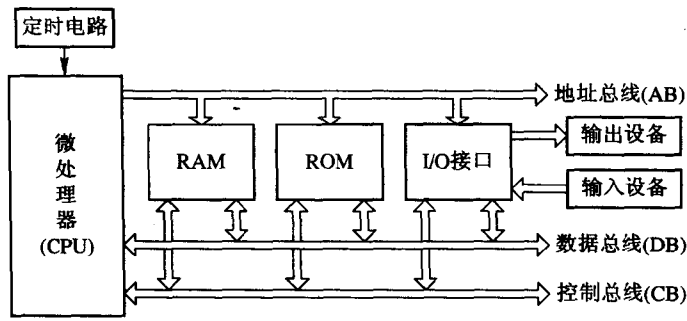


图 1.3 微型计算机硬件系统结构

(2) 数据总线 DB(Data Bus): 从存储器取指令或读写操作数, 对 I/O 端口进行读写操作时, 指令码或数据信息通过数据总线送往 CPU 或由 CPU 送出。DB 是双向总线。

(3) 控制总线 CB(Control Bus): 各种控制或状态信息通过控制总线由 CPU 送往有关部件, 或者从有关部件送往 CPU。CB 中每根线的传送方向是一定的, 图 1.3 中 CB 作为一个整体, 用双向表示。

采用总线结构时, 系统中各部件均挂在总线上, 可使微机系统的结构变得简单, 易于维护, 并具有更好的可扩展性。一个部件(插件), 只要符合总线标准就可以直接插入系统, 为用户对系统功能的扩充或升级提供了很大的灵活性。

1.4.2 微处理器

微处理器由运算器(ALU)、控制器(CU)和内部寄存器阵列(RA)三部分组成, 如图 1.4 所示。现将各部件的功能简述如下。

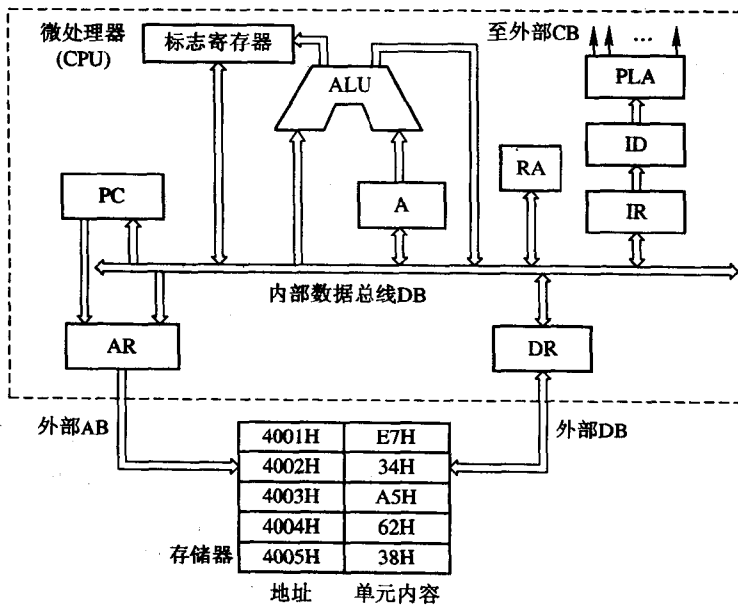


图 1.4 微处理器结构

1. 运算器

运算器又称算术逻辑单元(ALU, Arithmetic Logic Unit), 用来进行算术或逻辑运算以及移位循环等操作。参加运算的两个操作数一个来自累加器 A(Accumulator), 另一个来自内部数据总线 DB, 可以是数据缓冲寄存器(DR, Data Register)中的内容, 也可以是寄存器阵列(RA, Register Array)中某个寄存器的内容。计算结果送回累加器 A 暂存。

2. 控制器

控制器又称控制单元(CU, Control Unit), 是全机的指挥控制中心。它负责把指令逐条从存储器中取出, 经译码分析后向全机发出取数、执行、存数等控制命令, 以保证正确完成程序所要求的功能。

(1) 指令寄存器 IR(Instruction Register): 用来存放从存储器取出的将要执行的指令码。当执行一条指令时, 先把它从内存储器取出并送到数据缓冲寄存器 DR 中, 然后再传送到指令寄存器 IR 中。

(2) 指令译码器 ID(Instruction Decoder): 用来对指令寄存器 IR 中的指令操作码字段(指令中用来说明指令功能的字段)进行译码, 以确定该指令应执行什么操作。

(3) 可编程逻辑阵列 PLA(Programmable Logic Array): 用来产生取指令和执行指令所需要的各种微操作控制信号, 并经过控制总线 CB 送往有关部件, 从而使计算机完成相应的操作。

3. 内部寄存器阵列

寄存器阵列 RA(Register Array)实际上相当于微处理器内部的 RAM。微处理器内部有了这些寄存器后, 就可避免频繁访问存储器, 缩短指令长度和指令执行时间, 提高机器的运行速度, 方便程序设计。虽然不同计算机的 CPU 中其 RA 会有所不同, 但一般至少要有以下几个寄存器。

1) 程序计数器 PC(Program Counter)

程序计数器有时也被称为指令指针 IP(Instruction Pointer)。它被用来存放下一条要执行指令所在存储单元的地址。在程序开始执行前, 必须将它的起始地址, 即程序的第一条指令所在的存储单元地址送入 PC。当执行指令时, CPU 将自动修改 PC 内容, 以便使其保持的总是将要执行的下一条指令的地址。由于大多数指令是按顺序执行的, 因而修改的办法通常只是简单地对 PC 加 1。但遇到跳转等改变程序执行顺序的指令时, 后继指令的地址(即 PC 的内容)将从指令寄存器 IR 中的地址字段得到。

2) 地址寄存器 AR(Address Register)

地址寄存器用来存放正要取出的指令的地址或操作数的地址。由于在内存单元和 CPU 之间存在着操作速度上的差异, 因而必须使用地址寄存器来保持地址信息, 直到内存的读/写操作完成为止。

在取指令时, PC 中存放的指令地址送到 AR, 根据此地址从存储器中取出指令。

在取操作数时, 将操作数地址通过内部数据总线送到 AR, 再根据此地址从存储器中取出操作数; 在向存储器存入数据时, 也要先将待写入数据的地址送到 AR, 再根据此地址向存储器写入数据。

3) 数据缓冲寄存器 DR(Data Register)

数据缓冲寄存器用来暂时存放指令或数据。从存储器读出时，若读出的是指令，经 DR 暂存的指令经过内部数据总线送到指令寄存器 IR；若读出的是数据，则通过内部数据总线送到运算器或有关的寄存器。同样，当向存储器写入数据时，也首先将其存放在数据缓冲寄存器 DR 中，然后再经数据总线送入存储器。

可以看出，数据缓冲寄存器 DR 是 CPU 和内存、外部设备之间信息传送的中转站，用来补偿 CPU 和内存、外围设备之间在操作速度上存在的差异。

4) 指令寄存器 IR(Instruction Register)

指令寄存器用来保存从存储器取出的将要执行的指令码，以便指令译码器对其操作码字段进行译码，产生执行该指令所需的微操作命令。

5) 累加器 A(Accumulator)

累加器是使用最频繁的一种寄存器。在执行算术逻辑运算时，它用来存放一个操作数，而运算结果通常又放回累加器，其中原有信息随即被破坏。因此，顾名思义，累加器是用来暂时存放 ALU 运算结果的。显然，CPU 中至少应有一个累加器。目前 CPU 中通常有多个累加器。当使用多个累加器时，就变成了通用寄存器堆结构，其中任何一个既可存放目的操作数，也可存放源操作数。

6) 标志寄存器 FLAGS(Flag Register)

标志寄存器有时也称为程序状态字 PSW(Program Status Word)。它用来存放执行算术运算指令、逻辑运算指令或测试指令后建立的各种状态码内容以及对 CPU 操作进行控制的控制信息。标志位的具体设置及功能随微处理器型号的不同而不同。编写程序时，可以通过测试有关标志位的状态(0 或 1)来决定程序的流向。

1.4.3 存储器

这里介绍的存储器是指内存储器(又称为主存或内存)。它是微型计算机的存储和记忆装置，用来存放指令、原始数据、中间结果和最终结果。

在计算机内部，程序和数据都以二进制形式表示，8 位二进制代码作为一个字节。为了便于对存储器进行访问，存储器通常被划分为许多单元，每个存储单元存放一个字节的信息，每个存储单元分别赋予一个编号，称为地址。如图 1.5 所示，地址为 4005H 的存储单元中存放了一个 8 位二进制信息 00111000B。

计算机在执行程序时，CPU 会自动而连续地从内存储器中取出要执行的指令，并执行指令规定的操作。也就是说，计算机每完成一条指令，至少有一次为取指令而访问内存的操作。内存储器是计算机主机的一部分，一般把具有一定容量且速度较高的存储器作为内存储器，CPU 可直接用指令对内存储器进行读/写。在微型计算机中，通常用半导体存储器作为内存储器。

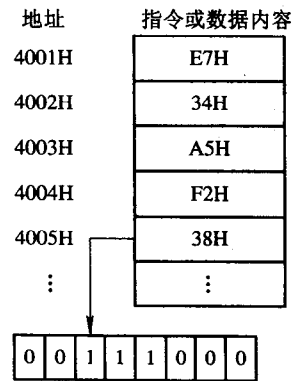


图 1.5 内存单元的地址和内容