



普通高等教育“十二五”规划教材
电子电气基础课程规划教材

微机原理与接口技术

——基于8086和Proteus仿真

顾 晖 梁惺彦 等编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

电子电气基础课程规划教材

微机原理与接口技术

——基于 8086 和 Proteus 仿真

顾 晖 梁惺彦 陈 越 编著
张 洁 华 琇 胡 慧

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从微型计算机系统应用的角度出发,以 Intel 8086 微处理器和 IBM PC 系列微机为主要对象,系统介绍了微型计算机的基本组成、工作原理、接口技术及应用。全书共 13 章,包括:微型计算机系统概述、计算机中的数和数制、8086 微型计算机系统、8086 寻址方式与指令系统、8086 汇编语言程序设计、Proteus 仿真平台的使用、存储器、输入/输出接口、可编程接口芯片、中断与中断管理、数模与模数转换及应用、总线和 Proteus ISIS 仿真实例。

本书内容全面、实用性强,原理、技术与应用并重,并特别介绍了利用 EDA 工具——Proteus ISIS 的实验方法,讲述有特点和新意。书中提供了较多的实例,这些实例全部在 Proteus 中调试通过,设计方案同时适用于实验箱实验的教学方式。

本书可作为高等院校工科电气与电子信息类专业本科生的教材,也可作为研究生教材或供有关工程技术人员参考使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

微机原理与接口技术:基于 8086+Proteus 仿真/顾晖等编著. —北京:电子工业出版社,2011.8
电子电气基础课程规划教材
ISBN 978-7-121-14431-8

I. ①微… II. ①顾… III. ①微型计算机—理论—高等学校—教材②微型计算机—接口—高等学校—教材
IV: ①TP36

中国版本图书馆 CIP 数据核字(2011)第 171293 号

责任编辑:凌毅

印刷:北京丰源印刷厂

装订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开本:787×1092 1/16 印张:19 字数:510 千字

印次:2011 年 8 月第 1 次印刷

册数:4 000 册 定价:35.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

“微机原理与接口技术”是高等学校电子信息工程、通信工程、自动化、电气工程及其自动化等工科电气与电子信息类各专业的核心课程。本课程的任务是使学生从系统的角度出发，掌握微机系统的基本组成、工作原理、接口电路及应用方法，使学生掌握微机系统的开发能力。为了适应教学的需要，编者在总结了多年的教学科研实践经验、对有关微型计算机技术资料进行综合提炼的基础上，编写了本书。

本书特别考虑了内容的选取与组织，注意从微机应用的需求出发，以 Intel 8086 微处理器和 IBM PC 系列微机为主要对象，系统、深入地介绍了微型计算机的基本组成、工作原理、接口技术及应用，把微机系统开发过程中用到的硬件技术和软件技术有机地结合起来。全书共 13 章，包括微型计算机系统概述、计算机中的数和数制、8086 微型计算机系统、8086 寻址方式与指令系统、8086 汇编语言程序设计、Proteus 仿真平台的使用、存储器、输入/输出接口、可编程接口芯片、中断与中断管理、数模与模数转换及应用、总线和 Proteus ISIS 仿真实例。

本书有如下特色：

(1) 内容精练。本书以 Intel 8086 微处理器和 IBM PC 系列微机为主要对象，重点突出，内容全面。

(2) 实用性强。本书从应用需求出发，在讲清基本原理的基础上，强调软硬件结合的思维方法和实践动手能力的培养，侧重微机系统的设计。

(3) 实验手段先进。本书介绍了可适用于该课程教学实践的先进的 EDA 工具——Proteus 的用法，并引入大量实例。书中实例全部按照课程内容进行规划，对于同一个问题提供了多种不同的实现方案来解决，可以使学生更好地体会到技术的发展，较好体现了从整体→局部→整体的知识体系。而且，书中所介绍的实例方案同样适用于在实验箱上进行实验。

(4) 可读性强。书中内容力求文字精练、语言流畅。在内容安排上还注意由浅入深、分散难点。特别是在接口部分，注意形成芯片结构、编程和应用一体化的讲解体系，以便学生理解和应用。

本书的编写采用集体讨论、分工编写、交叉修改的方式进行。本书的第 1、3、11 章由张洁编写；第 4、5、12 章由顾晖编写；第 7、8 章由陈越编写；第 6、13 章由梁愷彦编写；第 9 章由华琇编写，第 10 章由胡慧编写。全书由顾晖、梁愷彦统稿并最后定稿。

本书配有电子课件、源程序包等教学资源，读者可以登录电子工业出版社华信教育资源网 (www.hxedu.com.cn) 下载。

本书的编写工作得到了广州风标电子技术公司的支持。在本书的编写过程中，广州风标电子技术公司的梁树先和杨炼指导了 Proteus 仿真实例的设计，在此向他们表示真诚的感谢！

由于编者水平有限，加之时间仓促，书中错误和不当之处在所难免，敬请读者批评指正。

编者

2011 年 6 月

目 录

第 1 章 微型计算机系统概述1	
1.1 绪论.....1	
1.2 微型计算机系统的硬件组成与工作原理.....1	
1.2.1 微型计算机系统的硬件组成.....1	
1.2.2 微型计算机系统的工作原理.....3	
1.3 微型计算机系统的启动过程.....4	
1.4 微型计算机系统的 EDA 设计方法.....6	
习题.....7	
第 2 章 计算机中的数和数制8	
2.1 数制.....8	
2.1.1 数制的表示.....8	
2.1.2 数制之间的转换.....9	
2.1.3 二进制数的运算.....12	
2.2 带符号二进制数的表示与运算.....12	
2.2.1 带符号二进制数的表示.....12	
2.2.2 带符号二进制数的运算.....14	
2.3 二进制编码的十进制数.....15	
2.3.1 BCD 码的编码方法.....15	
2.3.2 8421BCD 码的加、减运算.....16	
2.4 无符号数.....18	
2.5 字符的表示.....18	
习题.....19	
第 3 章 8086 微型计算机系统21	
3.1 8086 微处理器的结构.....21	
3.1.1 8086 的内部结构.....21	
3.1.2 8086 的寄存器结构.....23	
3.2 8086 微处理器的工作模式及引脚特性.....25	
3.2.1 8086 的工作模式.....25	
3.2.2 8086 的引脚特性.....26	
3.3 8086 微型计算机系统.....30	
3.3.1 8086 微型计算机系统的硬件组成.....30	
3.3.2 8086 微型计算机系统的存储器组织.....31	
3.3.3 8086 微型计算机系统的 I/O 组织.....33	
3.4 8086 微型计算机系统的总线时序.....33	
3.4.1 基本概念.....33	
3.4.2 最小模式下的总线周期时序.....35	
3.4.3 最大模式下的总线周期时序.....39	
习题.....41	
第 4 章 8086 寻址方式与指令系统42	
4.1 概述.....42	
4.2 8086 寻址方式.....42	
4.2.1 立即寻址.....42	
4.2.2 寄存器寻址.....43	
4.2.3 直接寻址.....43	
4.2.4 寄存器间接寻址.....44	
4.2.5 寄存器相对寻址.....45	
4.2.6 基址变址寻址.....46	
4.2.7 相对基址变址寻址.....46	
4.3 8086 指令系统.....48	
4.3.1 数据传送指令.....48	
4.3.2 算术运算指令.....55	
4.3.3 位运算指令.....61	
4.3.4 串操作指令.....65	
4.3.5 控制转移指令.....68	
4.3.6 处理器控制指令.....74	
习题.....74	
第 5 章 8086 汇编语言程序设计78	
5.1 汇编语言基础知识.....78	
5.1.1 概述.....78	
5.1.2 汇编源程序的结构.....78	
5.1.3 汇编语言的语句.....79	

5.1.4	汇编语言的数据	81	6.3	Proteus ISIS 下 8086 的仿真	131
5.1.5	汇编语言的操作符与表达式	82	6.3.1	编辑电路原理图	131
5.2	汇编语言的伪指令	85	6.3.2	设置外部代码编译器	133
5.2.1	变量定义伪指令	85	6.3.3	添加源代码并选择编译器	134
5.2.2	符号定义伪指令	86	6.3.4	仿真调试	136
5.2.3	段定义伪指令	87	习题		137
5.2.4	过程定义伪指令	88	第 7 章 存储器		138
5.2.5	模块定义和结束伪指令	88	7.1	半导体存储器的分类	138
5.2.6	其他伪指令	89	7.1.1	RAM 的分类	139
5.3	系统功能调用	90	7.1.2	ROM 的分类	139
5.3.1	DOS 功能调用	90	7.2	半导体存储器的主要技术指标	140
5.3.2	BIOS 功能调用	91	7.3	典型存储器芯片介绍	140
5.4	汇编语言程序设计	92	7.4	存储器与系统的连接	145
5.4.1	程序的质量标准	92	7.4.1	存储器扩展	145
5.4.2	汇编语言程序设计的基本步骤	92	7.4.2	存储器地址译码方法	146
5.4.3	顺序结构程序设计	92	7.4.3	8086CPU 与存储器的连接	147
5.4.4	分支结构程序设计	93	习题		151
5.4.5	循环结构程序设计	95	第 8 章 输入/输出接口		153
5.4.6	子程序设计	97	8.1	I/O 接口概述	153
5.4.7	汇编语言程序设计举例	101	8.1.1	CPU 与 I/O 设备之间交换的信息	153
5.5	汇编语言程序的上机过程	107	8.1.2	I/O 接口的主要功能	154
5.5.1	上机环境	107	8.1.3	I/O 接口的结构	154
5.5.2	上机过程	107	8.1.4	输入/输出的寻址方式	155
5.5.3	运行调试	108	8.2	常用 I/O 接口芯片	156
习题		110	8.3	CPU 与外设之间的数据传送方式	157
第 6 章 Proteus 仿真平台的使用		113	8.3.1	程序控制方式	157
6.1	Proteus 简介	113	8.3.2	中断方式	159
6.1.1	Proteus ISIS 编辑环境	113	8.3.3	直接存储器存取方式	159
6.1.2	Proteus ARES 编辑环境	113	8.3.4	通道控制方式和 I/O 处理器	160
6.2	Proteus ISIS 基本使用	114	习题		160
6.2.1	可视化界面及工具	114	第 9 章 可编程接口芯片		162
6.2.2	基本操作	117	9.1	可编程接口芯片概述	162
6.2.3	元件的查找与选取	118	9.2	可编程并行接口芯片 8255A	162
6.2.4	元件的使用	123	9.2.1	8255A 的内部结构及引脚功能	162
6.2.5	连线	127	9.2.2	8255A 的工作方式	164
6.2.6	元件标签	129	9.2.3	8255A 的编程	166
6.2.7	器件标注	129	9.2.4	8255A 的应用举例	167
6.2.8	属性分配工具 (PAT)	130	9.3	可编程定时/计数器 8253/8254	175
6.2.9	全局标注器	130			

9.3.1	8253 的内部结构及引脚功能	175	11.3	模数转换及应用	241
9.3.2	8253 的工作方式	177	11.3.1	模数转换器的基本原理	241
9.3.3	8253 的初始化	180	11.3.2	模数转换器的性能参数	242
9.3.4	8253 应用举例	182	11.3.3	8 位 A/D 转换器	
9.4	可编程串行通信接口芯片 8251A	185		ADC0808/0809	243
9.4.1	串行数据传送方式	185		习题	249
9.4.2	传输速率和传送距离	186	第 12 章 总线		250
9.4.3	同步串行通信与异步串行通信	187	12.1	总线的概念	250
9.4.4	通用可编程串行通信接口		12.2	系统总线	251
	芯片 8251A	188	12.2.1	ISA 总线	251
	习题	196	12.2.2	EISA 总线	253
第 10 章 中断与中断管理		199	12.2.3	PCI 总线	253
10.1	中断概念	199	12.3	外部总线	255
10.1.1	中断与中断源	199	12.3.1	RS-232C 串行总线	255
10.1.2	中断系统的功能	199	12.3.2	USB 总线	258
10.1.3	简单的中断处理过程	200		习题	261
10.1.4	中断源识别及优先级判断	203	第 13 章 Proteus ISIS 仿真基础实例		262
10.2	8086 的中断系统	205	13.1	基本 I/O 应用——I/O 译码	262
10.2.1	8086 的中断类型	206	13.1.1	功能说明	262
10.2.2	中断向量和中断向量表	207	13.1.2	Proteus 电路设计	262
10.2.3	8086 中的中断响应和处理		13.1.3	代码设计	266
	过程	210	13.1.4	仿真分析与思考	267
10.3	可编程中断控制器 8259A	212	13.2	定时/计数器 8253 的应用——波形	
10.3.1	8259A 的结构	213		发生器	267
10.3.2	8259A 的引脚	215	13.2.1	功能说明	267
10.3.3	8259A 的中断处理过程	215	13.2.2	Proteus 电路设计	267
10.3.4	8259A 的工作方式	216	13.2.3	代码设计	269
10.3.5	8259A 的编程与应用	219	13.2.4	仿真分析与思考	270
10.4	中断程序设计	226	13.3	并行接口芯片 8255A 的应用——	
10.4.1	中断设计方法	226		键盘与数码管	272
10.4.2	中断程序设计举例	227	13.3.1	功能说明	272
	习题	232	13.3.2	Proteus 电路设计	272
第 11 章 数模与模数转换及应用		233	13.3.3	代码设计	272
11.1	物理信号到电信号的转换	233	13.3.4	仿真分析与思考	277
11.1.1	概述	233	13.4	中断应用——8259A 芯片的使用	277
11.1.2	几种常见的传感器	233	13.4.1	功能说明	277
11.2	数模转换及应用	234	13.4.2	Proteus 电路设计	277
11.2.1	数模转换器的基本原理	234	13.4.3	代码设计	279
11.2.2	数模转换器的性能参数	237	13.4.4	仿真分析与思考	280
11.2.3	8 位 D/A 转换器 DAC0832	238			

13.5 模数转换——ADC0808 的使用	281	13.6.4 仿真分析与思考	289
13.5.1 功能说明	281	13.7 串行通信——8251A 的使用	289
13.5.2 Proteus 电路设计	281	13.7.1 功能说明	289
13.5.3 代码设计	283	13.7.2 Proteus 电路设计	289
13.5.4 仿真分析与思考	286	13.7.3 代码设计	291
13.6 数模转换——DAC0832 的使用	286	13.7.4 仿真分析与思考	292
13.6.1 功能说明	286	附录 A VSM 仿真的元件库	294
13.6.2 Proteus 电路设计	286	参考文献	296
13.6.3 代码设计	288		

第1章 微型计算机系统概述

1.1 绪 论

计算机是一种高速、精确的信息处理和传递的工具，目前得到了广泛的应用。自 1946 年第一台电子计算机 ENIAC 问世以来，计算机的发展主要经历了电子管、晶体管、中小规模集成电路、大规模和超大规模集成电路等几个阶段。微型计算机是建立在大规模和超大规模集成电路技术基础上的第四代计算机的总称，它具有体积小、重量轻、更新发展迅速和应用面广等特点。

将计算机的核心器件中央处理器（运算器和控制器）集成在一块半导体芯片上，配以存储器、I/O 接口电路及系统总线等设备的计算机，称为微型计算机，简称微机。人们可以从不同的角度对微型机进行分类。通常以微处理器型号为标志划分微型计算机，如 286 计算机、386 计算机、486 计算机、Pentium 计算机、Pentium II 计算机、Pentium III 计算机、Pentium 4 计算机等；也可以按计算机运算部件处理的数据位数来划分，如 8 位计算机、16 位计算机、32 位计算机、64 位计算机等，位数越多计算机运算速度越快。

以微型计算机为主体，配上系统软件和外设之后，就构成了微型计算机系统。一个完整的微型计算机系统由微型计算机硬件系统和软件系统两大部分构成。计算机硬件是计算机系统中由电子、机械和光电元件组成的各种计算机部件和设备的总称，是计算机完成各项工作的物质基础，是构成计算机的看得见、摸得着的物理部件。它是计算机的“躯壳”，包括：CPU、主板、内存、硬盘、键盘和鼠标等。计算机软件是指计算机运行时需要的各种程序及有关资料，是计算机的“灵魂”。计算机硬件和软件是计算机系统缺一不可的组成部分。

1.2 微型计算机系统的硬件组成与工作原理

1.2.1 微型计算机系统的硬件组成

关于微型计算机的硬件组成，传统的描述是沿用美籍匈牙利科学家冯·诺依曼提出的“存储程序和程序控制”的设计思想。按照这一设计思想，微型计算机系统的硬件由运算器、控制器、存储器、输入设备和输出设备五大基本部件组成，如图 1-1 所示。计算机内部采用二进制数据格式表示数据和指令。程序将被事先存入主存储器中，计算机工作时能在不需要操作人员干预的情况下，自动逐条取出指令并执行。

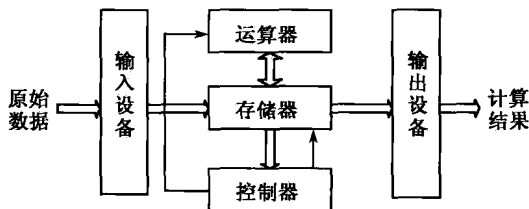


图 1-1 微型计算机系统硬件组成

对于微型计算机的硬件组成，还可以细分一下，如图 1-2 所示。

1. 微处理器

微处理器包含运算器和控制器，微处理器的性能基本决定了计算机的性能，是微型计算机的核心。随着微电子技术的发展，微型计算机的发展基本遵循摩尔定律。1971年，Intel公司研制成功了第一台微处理器 Intel 4004，并以此为核心组成了微型计算机 MCS-4。1973年该公司又研制成功了8位微处理器 Intel 8080。随后，其他许多公司竞相推出各类微处理器和微型计算机产品。1977年，美国 APPLE 公司推出了著名的 APPLE II 计算机，它采用8位的微处理器，是第一种被广泛应用的微型计算机，开创了微型计算机的新时代。1981年，IBM公司基于 Intel 8088 芯片推出的 IBM-PC 计算机以其优良的性能、低廉的价格及技术上的优势迅速占领市场，使微型计算机进入了一个迅速发展使用时期。在短短的十几年内，微型计算机经历了从8位到16位、32位再到64位的发展过程。

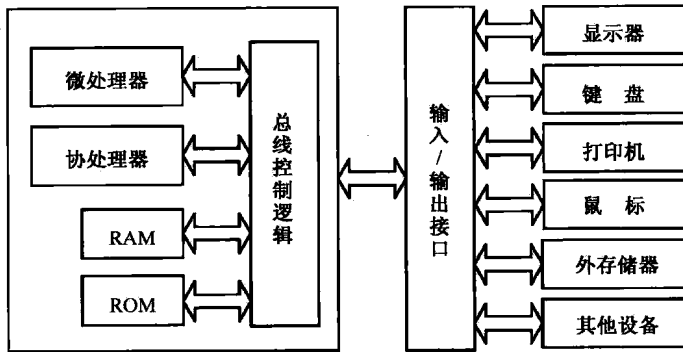


图 1-2 微型计算机系统硬件组成（细分）

2. 协处理器

协处理器用于特定任务的处理，以减轻系统微处理器的负担，是微型计算机系统的选配硬件。例如，数学协处理器可以控制数字处理；图形协处理器可以处理视频绘制。常见的协处理器有 Intel 8087。

3. 内存储器（也称主存或内存）

内存储器用于存放计算机正在运行的程序和用到的数据等，分为随机存取存储器 RAM（Random Access Memory）和只读存储器 ROM（Read-Only Memory）两大类。

随机存取存储器（RAM）：RAM 接受程序的控制，可由用户写入数据或读出数据，但是断电后数据会消失。RAM 可以用来临时存放程序、输入数据和中间结果等。

只读存储器（ROM）：ROM 中的信息由厂家预先写入，一般用来存放自检程序、配置信息等。通常只能读出而不能写入，断电后信息不会丢失。

4. 总线控制逻辑

微型计算机系统采用总线结构，总线是连接计算机各组成部件的公共数据通路。在微型计算机系统中，总线分片内总线、片级总线和系统总线。其中片内总线用以连接 CPU 内部的各个部件，例如 ALU、通用寄存器、内部 Cache 等。片级总线用以连接 CPU、存储器及 I/O 接口等电路，构成所谓的主机板。系统总线主要用来连接外部设备，系统总线的直观形式就是主板上的扩充插槽。

主板与外部设备之间的数据传输必须通过系统总线，所以系统总线包含的信号线必须满足下列各种输入/输出操作的需要：

- ① 访问分布于主板之外的存储器;
- ② 访问 I/O 接口;
- ③ 适应外部中断方式;
- ④ 适应存储器直接与外部设备交换信息。

总线控制逻辑的任务就是产生和接受这些操作所需要的信号。

5. 外存储器（也称辅存或外存）

外存储器用来存储大量暂时不参加运算或处理的数据和程序，是主存的后备和补充。常见的外存储器主要有：

硬盘：安装在主机箱内，常见容量有：40GB、80GB、120GB 等。

光盘：信息读取要借助于光驱，其容量为 650MB。

DVD 光盘：存储密度高，存储容量大，容量一般为 4.7GB。

U 盘：是利用闪存在断电后还能保持存储的数据不丢失的特点而制成的，特点是重量轻、体积小。

移动硬盘：可以通过 USB 接口即插即用，特点是体积小、重量轻、容量大、存取速度快。

1.2.2 微型计算机系统的工作原理

1. 指令、指令系统和程序

指令是指示计算机执行某种操作的命令，它由一串二进制代码组成。一条指令通常由操作码和操作数两个部分组成。操作码指明该指令要完成的操作的类型或性质，如取数、加法、减法或逻辑乘等。操作数指明操作对象本身或操作对象所存储的位置——寄存器名或存储单元地址。

一台计算机所能识别和执行的全部指令的集合，称为该计算机的指令集合或指令系统，它描述了计算机内全部的控制信息和“逻辑判断”能力。不同计算机的指令系统包含的指令种类和数目也不同。一般均包含算术运算型、逻辑运算型、数据传送型、条件控制型、输入/输出等指令。指令系统是表征一台计算机性能的重要因素，它的格式与功能不仅直接影响到机器的硬件结构，而且也直接影响到系统软件，影响到机器的适用范围。

计算机根据人们预定的安排（即存储在内存中的程序），自动地进行数据的快速计算和加工处理。人们预定的安排是通过一个指令序列来表达的，这个指令序列就称为程序。一条指令规定计算机执行的一个基本操作，一个程序规定计算机完成的一个完整任务。

2. 机器语言、汇编语言和高级语言

机器语言是直接由二进制代码指令表达的计算机语言。二进制代码指令是用 0 和 1 组成的一串代码，它们有一定的位数，并分成若干段，各段的编码表示不同的含义。用机器语言编写的指令是计算机唯一能直接识别和执行的指令。

为了便于记忆和书写，每条二进制代码指令都可以用字母或符号来表示，称为汇编语言。汇编语言能够直接访问与硬件相关的存储器或 I/O 端口，能对关键代码进行更准确的控制，能避免因线程共同访问或者硬件设备共享引起的死锁，能最大限度地发挥硬件的功能。

使用汇编语言编写的指令和程序，计算机不能直接识别和执行，需要通过一种程序将汇编语言翻译成机器语言后才能被计算机识别和执行。这种起翻译作用的程序叫汇编程序。利用汇编程序将汇编语言翻译成机器语言的过程称为汇编。

由于汇编语言依赖于计算机硬件体系结构，而且助记符量大难记，于是人们又设计了更加易用的高级语言。例如，C++、Java 等。高级语言程序“看不见”机器的硬件结构，不能用于编写直接访问机器硬件资源的系统软件或设备控制软件。为此，一些高级语言提供了与汇编语

言之间的调用接口。用汇编语言编写的程序，可作为高级语言的一个外部过程或函数，利用堆栈来传递参数或参数的地址。

3. 微型计算机系统的工作原理

微型计算机（简称微机）的工作就是运行程序。正常情况下，一台微机通电后就开始运行某种程序。运行程序就是依次逐条从存储器中取出指令并完成指令规定的动作。如前所述，微机系统的基本工作原理是存储程序和程序控制。按照这一原理，预先要把指挥计算机如何进行操作的指令序列（程序）和原始数据通过输入设备输送到计算机的内存储器中。每条指令明确规定了计算机从哪个地址取数，进行什么操作，然后送到什么地址等。在执行程序时，CPU 根据当前程序指针寄存器的内容取出指令并执行指令，然后再取出下一条指令并执行，如此循环下去，直到程序结束指令时才停止。微机的整个工作过程就是不断地取指令和执行指令的过程，期间所涉及的计算机硬件部件有：内存储器、指令寄存器、指令译码器、控制器、运算器和输入/输出设备等。

1.3 微型计算机系统的启动过程

微型计算机系统的启动过程中有一个非常完善的硬件自检机制。微型计算机上电启动的过程如下：

① 当电源开关被按下时，电源就开始向主板和其他设备供电，此时电压还不稳定，主板控制芯片组会向 CPU 发出并保持一个 RESET（复位）信号，让 CPU 初始化。当电源开始稳定供电后（从不稳定到稳定的过程只是一个短暂的瞬间），芯片组便撤去 RESET 信号（如果是手动按下计算机面板上的 Reset 按钮来重启机器，那么松开该按钮时芯片组就会撤去 RESET 信号），CPU 马上就从存储器地址 FFFF0H 处开始执行指令，这个地址在系统 BIOS 的地址范围内，放在这里的只是一条跳转指令，以使 CPU 跳到系统 BIOS 的启动代码处执行指令。

② 系统 BIOS 的启动代码首先要做的事情就是进行 POST (Power On Self Test, 加电自检)。POST 的主要任务是检测系统中的一些关键设备是否存在和能否正常工作，如内存和显卡等。由于 POST 的检测过程在显卡初始化之前，因此如果在 POST 自检的过程中发现了一些致命错误，如没有找到内存或者内存有问题时（POST 过程只检查 640KB 常规内存），是无法在屏幕上显示出来的，这时系统 BIOS 可通过扬声器发声来报告错误情况，声音的长短和次数代表了错误的类型。在正常情况下，POST 过程进行得非常快，几乎无法感觉到。

③ 接下来，系统 BIOS 将查找显卡的 BIOS，存放显卡 BIOS 的 ROM 芯片的起始地址通常在 C0000H 处，系统 BIOS 找到显卡 BIOS 之后调用它的初始化代码，由显卡 BIOS 来完成显卡的初始化。大多数显卡在这个过程通常会在屏幕上显示出一些显卡的信息，如生产厂商、图形芯片类型、显存容量等内容，这就是我们开机看到的第一个画面，不过这个画面几乎是一闪而过的，也有的显卡 BIOS 使用了延时功能，以使用户可以看清显示的信息。接着系统 BIOS 会查找其他设备的 BIOS 程序，找到之后同样要调用这些 BIOS 内部的初始化代码来初始化这些设备。

④ 查找完所有其他设备的 BIOS 之后，系统 BIOS 将显示它自己的启动画面，其中包括系统 BIOS 的类型、序列号和版本号等内容。同时屏幕底端左下角会出现主板信息代码，包括 BIOS 的日期、主板芯片组型号、主板的识别编码和厂商代码等。

⑤ 接着系统 BIOS 将检测 CPU 的类型和工作频率，并将检测结果显示在屏幕上，这就是我们开机看到的 CPU 类型和主频。接下来系统 BIOS 开始测试主机所有的内存容量，并同时在屏幕上显示内存测试的数值，即屏幕上滚动显示的内存计数器。

⑥ 内存测试通过之后，系统 BIOS 将开始检测系统中安装的一些标准硬件设备，这些设备包括：硬盘、CD-ROM、串行接口和并行接口等。另外，绝大多数新版本的系统 BIOS 在这一阶段还要自动检测和设置内存的定时参数、硬盘参数和访问模式等。

⑦ 标准设备检测完毕后，系统 BIOS 内部的支持即插即用的代码将开始检测和配置系统中安装的即插即用设备，每找到一个设备之后，系统 BIOS 都会在屏幕上显示出设备的名称和型号等信息，同时为该设备分配中断、DMA 通道和 I/O 端口等资源。

⑧ 到这一步为止，所有硬件都已经检测配置完毕了，系统 BIOS 会重新清屏并在屏幕上方显示出一个系统配置列表，其中概略地列出了系统中安装的各种标准硬件设备，以及它们使用的资源和一些相关工作参数。

⑨ 接下来系统 BIOS 将更新 ESCD (Extended System Configuration Data, 扩展系统配置数据)。ESCD 是系统 BIOS 用来与操作系统交换硬件配置信息的数据，这些数据被存放在 CMOS (一小块特殊的 RAM, 由主板上的电池来供电) 之中。通常 ESCD 数据只在系统硬件配置发生改变后才会进行更新，所以不是每次启动机器时我们都能够看到“Update ESCD... Success”这样的信息。

⑩ ESCD 数据更新完毕后，系统 BIOS 的启动代码将进行它的最后一项工作，即根据用户指定的启动顺序从硬盘或光驱启动。以从 C 盘启动为例，系统 BIOS 将读取并执行硬盘上的主引导记录，主引导记录接着从分区表中找到第一个活动分区，然后读取并执行这个活动分区的分区引导记录，而分区引导记录将负责加载并引导操作系统。

上面介绍的就是计算机在打开电源开关（或按 Reset 键）进行冷启动时所要完成的各种初始化工作，如果按组合键 Ctrl+Alt+Del（或从 Windows 中选择重启计算机）来进行热启动，那么 POST 过程将被跳过去，直接从第③步开始，而且第⑤步的检测 CPU 和内存测试也不会再进行。无论是冷启动还是热启动，系统 BIOS 都会重复上面的硬件检测和引导过程。

目前，微型计算机上常见的操作系统有 DOS、OS/2、UNIX、Linux 和 Windows 等。DOS (Disk Operating System) 是磁盘操作系统的简称，是学习其他操作系统的基础。DOS 家族包括 MS-DOS、PC-DOS、DR-DOS 等。MS-DOS 是 Microsoft Disk Operating System 的简称，是由美国微软公司提供的磁盘操作系统。在 Windows 98 以前，DOS 是个人 PC 兼容计算机的最佳配置，而 MS-DOS 则是最普遍使用的 PC 兼容 DOS。

MS-DOS 采用模块结构，它由 4 部分组成：引导程序、IO.SYS 模块、MSDOS.SYS 模块和 COMMAND.COM 模块。

1. 引导程序

引导程序又叫“引导记录”，其作用是检查当前盘上是否有 DOS 系统文件。若有，则把 DOS 系统从磁盘装入内存。

2. IO.SYS 模块

IO.SYS 是 MS-DOS 和 ROM BIOS 之间的接口程序。它和 ROM BIOS 一起完成系统标准设备的管理，例如，标准的输入设备（键盘）和标准的输出设备（显示器）的驱动等。

3. MSDOS.SYS 模块

MSDOS.SYS 是 MS-DOS 的核心，包括 DOS 功能调用和实现 DOS 文件系统管理程序，用于实现文件管理，包括文件管理、目录管理、内存管理等功能。例如，在磁盘上建立文件、向文件中写入内容、从文件中读出内容等。它以功能调用的形式实现用户和 MS-DOS 之间的程序级接口。

4. COMMAND.COM 模块

COMMAND.COM 是命令处理程序，主要功能是负责接收、识别、解释和执行用户从键盘输入的 MS-DOS 命令。

DOS 的命令处理程序可以处理两类命令：一类是内部命令，这类命令处理程序就包含在 COMMAND.COM 之内。启动 DOS 时，内部命令被装入系统内存，用户可以随时使用。常用的内部命令有：DIR、DEL、COPY 等。另一类是外部命令，每个外部命令处理程序都是一个单独的文件，即以磁盘文件的形式存储在 DOS 磁盘中，它的文件名和命令名相同。启动 DOS 时，外部命令没有被装入系统内存。外部命令的执行过程：从键盘输入外部命令名时，COMMAND.COM 把这个外部命令处理程序文件从磁盘上读入内存，然后把控制转移到该文件中的程序，执行这个程序，之后返回到 DOS。常用的外部命令有 SYS、FORMAT、DISKCOPY 等。

用户在使用命令时，内部命令和外部命令没有什么区别。使用外部命令的话，会有一个把外部命令处理程序从磁盘读入内存的过程，而内部命令没有这个过程，因为内部命令处理程序在启动计算机时已经读入内存。

1.4 微型计算机系统的 EDA 设计方法

随着计算机与微电子技术的发展，微型计算机系统的设计正朝着速度快、容量大、体积小、重量轻的方向发展。推动这一发展的就是日趋进步和完善的 EDA 设计技术。EDA，即电子设计自动化，是近几年迅速发展起来的将计算机软件、硬件、微电子技术交叉运用的现代电子设计学科，其功能涵盖了电路描述、电路合成、电路仿真等三大电路设计工作。目前进入我国并具有广泛影响的 EDA 软件有 Protel、PSPICE、Multisim10、Modelsim 和 Proteus 等。这些工具都有较强的功能，一般均可以进行电路设计与仿真，有的还具有 PCB 自动布局布线、输出多种网表文件等功能。下面需要特别介绍的是 Proteus 这个软件。

Proteus 软件是英国 Labcenter Electronics 公司开发的 EDA 工具软件，由广州风标电子技术有限公司作为中国的销售代理。Proteus 软件除了可以实现原理图布图、PCB 自动或人工布线和 SPICE 电路仿真等其他 EDA 工具软件（如 Multisim）具有的功能外，还可以仿真处理器及其外围电路并实时使用 RAM、ROM、键盘、马达、LED、LCD、A/D、D/A、部分 SPI 器件和部分 I²C 器件。从原理图布图、代码调试到微处理器与外围电路协同仿真，一键切换到 PCB 设计，Proteus 真正实现了从概念到产品的完整设计。Proteus 是目前世界上唯一将电路仿真软件、PCB 设计软件和虚拟模型仿真软件三合一的设计平台，其处理器模型支持 8086、8051、HC11、PIC10/12/16/18/24/30/DsPIC33、AVR、ARM 和 MSP430 等，并还在持续增加其他系列处理器模型。在编译方面，它也支持 MASM IAR、Keil 和 MPLAB 等多种编译器。

Proteus 可以仿真数字和模拟、交流和直流等数千种元器件；可以提供示波器、逻辑分析仪、交直流电压表、交直流电流表等仿真仪表资源。Proteus 还提供了比较丰富的测试信号（包括模拟信号和数字信号）用于电路的测试。在 Proteus 绘制好原理图后，调入已编译好的目标代码文件（.HEX 文件或 .EXE 文件），可以在 Proteus 的原理图中看到模拟的实物运行状态和过程。

Proteus 提供的独特的微处理器协同仿真功能（VSM）使得 Proteus 成为了“微机原理”课程教学的先进助手。Proteus 不仅可将许多微处理器实例功能形象化，也可将许多微处理器实例运行过程形象化。前者可在相当程度上得到实物演示实验的效果，后者则是实物演示实验难以达到的效果。它的元器件、连接线路等和传统的微机原理实验硬件高度对应，这可以在相当程

度上替代了传统的微机原理实验教学的功能，如元器件选择、电路连接、电路检测、电路修改、软件调试、运行结果等。

使用 Proteus 软件进行微型计算机系统的仿真设计，是虚拟仿真技术和计算机多媒体技术相结合的综合运用，在不需要硬件投入的条件下，为微机原理课程的学习提供了一个便捷、灵活的实验环境，有利于培养学生的电路设计能力及仿真软件的操作能力。

综上所述，本书推荐使用 Proteus 软件，采用软件仿真与硬件实验箱实验相结合的方式实验。这种方式依托 Proteus 软件的强大软件仿真功能，在电路设计这个环节上给学习者以充分的灵活性和多样性，非常有利于培养学习者的学习积极性和实践设计动手能力。

本书将在第 6 章和第 13 章详细介绍如何使用 Proteus 软件进行微机原理实验仿真。

习 题

1. 计算机硬件的基本组成部分有哪些？简述各部分的功能。
2. 简述微型计算机系统的组成。
3. 简述总线控制逻辑的任务。
4. 简述冯·诺依曼的“程序存储和程序控制”原理。
5. 简述微型计算机系统的启动过程。
6. BIOS 是什么？简述系统 BIOS 代码完成的工作。
7. 常见的 EDA 软件有哪些？请列举 2~3 个，并简述它们的特点。

第2章 计算机中的数和数制

2.1 数 制

2.1.1 数制的表示

1. 计数制

数制也称为计数制，是指用一组固定的数字符号和统一的规则表示数的方法。对于任意 r 进制数 N ，可以用下式表示：

$$\sum_{i=-m}^n a_i r^i = a_{-m} r^{-m} + \dots + a_{-2} r^{-2} + a_{-1} r^{-1} + a_0 r^0 + a_1 r^1 + \dots + a_n r^n$$

其中：

① a_i 为数码，每一种进制数都有固定的数字符号，这个符号就是数码。

二进制有 2 个数码：0、1；

八进制有 8 个数码：0、1、2、3、4、5、6、7；

十进制有 10 个数码：0、1、2、3、4、5、6、7、8、9；

十六进制有 16 个数码：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。

② i 为数位，数位是指数码在一个数中所处的位置。

例如，十六进制数 56.78 从左到右的数位分别为：1、0、-1、-2。

③ r 为基数，基数是指在某计数制中，每个数位上能使用的数码的个数。

二进制基数为 2；

八进制基数为 8；

十进制基数为 10；

十六进制基数为 16。

④ r^i 为权，权是基数的幂，这个幂次由数位决定。

二进制第 i 位上的权为 2^i ；

八进制第 i 位上的权为 8^i ；

十进制第 i 位上的权为 10^i ；

十六进制第 i 位上的权为 16^i 。

例如，十六进制数 56.78 从左到右每一位的权分别为： 16^1 、 16^0 、 16^{-1} 、 16^{-2} 。

2. 计算机中常用的计数制

在日常生活中，人们最常用的是十进制计数制。而在计算机中，为了便于数的存储和表示，使用的是二进制计数制。但是由于二进制数据书写和记忆不方便，在计算机系统中还常使用八进制和十六进制等计数制。计算机中常用计数制的属性见表 2-1。

表 2-1 计算机中常用计数制

数 制	基 数	数 码	运 算 规 则	书 写 后 缀
二进制	2	0, 1	逢二进一，借一当二	B
八进制	8	0, 1, 2, 3, 4, 5, 6, 7	逢八进一，借一当八	O 或 Q

续表

数制	基数	数码	运算规则	书写后缀
十进制	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	逢十进一, 借一当十	D
十六进制	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	逢十六进一, 借一当十六	H

*** 说明:**

• 为了区别所使用的数制, 一般用以下两种书写格式表示:

① 用括号将数字括起, 后面加数制区分, 数制用下标的形式给出;

② 用后缀区分, 二进制数、十进制数、八进制数、十六进制数的后缀分别为字母 B (或 b)、D (或 d)、O (或 o) 或 Q (或 q)、H (或 h)。

例如: 十六进制数 56.78 可以表示成 $(56.78)_{16}$ 或 56.78H;

十进制数 56.78 可以表示成 $(56.78)_{10}$ 或 56.78D。

• 汇编程序规定, 使用首字符是字母的十六进制数时, 前面需加 0 来表示。

例如: 0B56.A8H、0FFH 等。

• 在没有混淆的情况下, 十进制数可以省略后缀 D (或 d)。

2.1.2 数制之间的转换

1. 其他数制数转换为十进制数

二进制数、八进制数和十六进制数转换为十进制数的方法是: “按权展开”。

【例 2-1】 将 1010.101B、23.4Q 和 56.78H 转换成十进制数。

解

$$1010.101B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 10.625D$$

$$23.4Q = 2 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1} = 19.5D$$

$$FA3.4H = 15 \times 16^2 + 10 \times 16^1 + 3 \times 16^0 + 4 \times 16^{-1} = 4003.25D$$

2. 十进制数转换为其他数制数

把十进制数转换为其他数制数的方法很多, 通常采用的方法有降幂法和乘法。

(1) 降幂法

假设要转换的十进制数为 N 。

步骤 1: 找出最接近 N 并小于等于 N 的 r 进制位权值 r^i ;

步骤 2: 找到满足 $0 \leq C < r$ 的最大数 C , 使得 $N - C \times r^i < r^i$, C 即为转换结果 (r 进制数) 第 i 位的位码 a_i ;

步骤 3: 计算 $N - C \times r^i$, 并用此值作为新的 N 值, 即 $N \leftarrow N - C \times r^i$;

步骤 4: i 自减 1, 即 $i \leftarrow i - 1$, 得到下一个位权值 r^i 。

重复步骤 2~步骤 4, 直至 N 为 0 或转换结果达到所需精度。

(2) 乘法

采用乘法把十进制数转换为二、八或十六进制数时, 整数部分除以基数取余, 直至商为 0; 小数部分乘以基数取整, 直至积为整数或小数位数由精度确定。

【例 2-2】 把十进制数 117.8125 转换成二进制数。

解 方法一: 降幂法

小于该数 117.8125 的位权是 64, 32, 16, 8, 4, 2, 1, 0.5, 0.25, 0.125, 0.0625, 按下列过程求出每位的位码。