

<<微机原理与接口技术>>

图书基本信息

书名：<<微机原理与接口技术>>

13位ISBN编号：9787121144318

10位ISBN编号：712114431X

出版时间：2011-8

出版时间：顾晖、梁惺彦、等 电子工业出版社 (2011-08出版)

作者：顾晖，梁惺彦 编

页数：296

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<微机原理与接口技术>>

内容概要

《微机原理与接口技术：基于8086和Proteus仿真》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。

《微机原理与接口技术：基于8086和Proteus仿真》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。

《微机原理与接口技术：基于8086和Proteus仿真》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。

书籍目录

第1章 微型计算机系统概述 1.1 绪论 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 1.2.2 微型计算机系统的工作原理 1.3 微型计算机系统的启动过程 1.4 微型计算机系统的EDA设计方法 习题 第2章 计算机中的数和数制 2.1 数制 2.1.1 数制的表示 2.1.2 数制之间的转换 2.1.3 二进制数的运算 2.2 带符号二进制数的表示与运算 2.2.1 带符号二进制数的表示 2.2.2 带符号二进制数的运算 2.3 二进制编码的十进制数 2.3.1 BCD码的编码方法 2.3.2 8421BCD码的加、减运算 2.4 无符号数 2.5 字符的表示 习题 第3章 8086微型计算机系统 3.1 8086微处理器的结构 3.1.1 8086的内部结构 3.1.2 8086的寄存器结构 3.2 8086微处理器的工作模式及引脚特性 3.2.1 8086的工作模式 3.2.2 8086的引脚特性 3.3 8086微型计算机系统 3.3.1 8086微型计算机系统的硬件组成 3.3.2 8086微型计算机系统的存储器组织 3.3.3 8086微型计算机系统的I/O组织 3.4 8086微型计算机系统的总线时序 3.4.1 基本概念 3.4.2 最小模式下的总线周期时序 3.4.3 最大模式下的总线周期时序 习题 第4章 8086寻址方式与指令系统 4.1 概述 4.2 8086寻址方式 4.2.1 立即寻址 4.2.2 寄存器寻址 4.2.3 直接寻址 4.2.4 寄存器间接寻址 4.2.5 寄存器相对寻址 4.2.6 基址变址寻址 4.2.7 相对基址变址寻址 4.3 8086指令系统 4.3.1 数据传送指令 4.3.2 算术运算指令 4.3.3 位运算指令 4.3.4 串操作指令 4.3.5 控制转移指令 4.3.6 处理器控制指令 习题 第5章 8086汇编语言程序设计 5.1 汇编语言基础知识 5.1.1 概述 5.1.2 汇编源程序的结构 5.1.3 汇编语言的语句 5.1.4 汇编语言的数据 5.1.5 汇编语言的操作符与表达式 5.2 汇编语言的伪指令 5.2.1 变量定义伪指令 5.2.2 符号定义伪指令 5.2.3 段定义伪指令 5.2.4 过程定义伪指令 5.2.5 模块定义和结束伪指令 5.2.6 其他伪指令 5.3 系统功能调用 5.3.1 DOS功能调用 5.3.2 BIOS功能调用 5.4 汇编语言程序设计 5.4.1 程序的质量标准 5.4.2 汇编语言程序设计的基本步骤 5.4.3 顺序结构程序设计 5.4.4 分支结构程序设计 5.4.5 循环结构程序设计 5.4.6 子程序设计 5.4.7 汇编语言程序设计举例 5.5 汇编语言程序的上机过程 5.5.1 上机环境 5.5.2 上机过程 5.5.3 运行调试 习题 第6章 Proteus仿真平台的使用 6.1 Proteus简介 6.1.1 Proteus ISIS编辑环境 6.1.2 Proteus ARES编辑环境 6.2 Proteus ISIS基本使用 6.2.1 可视化界面及工具 6.2.2 基本操作 6.2.3 元件的查找与选取 6.2.4 元件的使用 6.2.5 连线 6.2.6 元件标签 6.2.7 器件标注 6.2.8 属性分配工具 (PAT) 6.2.9 全局标注器 6.3 Proteus ISIS下8086的仿真 6.3.1 编辑电路原理图 6.3.2 设置外部代码编译器 6.3.3 添加源代码并选择编译器 6.3.4 仿真调试 习题 第7章 存储器 7.1 半导体存储器的分类 7.1.1 RAM的分类 7.1.2 ROM的分类 7.2 半导体存储器的主要技术指标 7.3 典型存储器芯片介绍 7.4 存储器与系统的连接 7.4.1 存储器扩展 7.4.2 存储器地址译码方法 7.4.3 8086CPU与存储器的连接 习题 第8章 输入/输出接口 8.1 I/O接口概述 8.1.1 CPU与I/O设备之间交换的信息 8.1.2 I/O接口的主要功能 8.1.3 I/O接口的结构 8.1.4 输入/输出的寻址方式 8.2 常用I/O接口芯片 8.3 CPU与外设之间的数据传送方式 8.3.1 程序控制方式 8.3.2 中断方式 8.3.3 直接存储器存取方式 8.3.4 通道控制方式和I/O处理器 习题 第9章 可编程接口芯片 9.1 可编程接口芯片概述 9.2 可编程并行接口芯片8255A 9.2.1 8255A的内部结构及引脚功能 9.2.2 8255A的工作方式 9.2.3 8255A的编程 9.2.4 8255A的应用举例 9.3 可编程定时/计数器8253/8254 9.3.1 8253的内部结构及引脚功能 9.3.2 8253的工作方式 9.3.3 8253的初始化 9.3.4 8253应用举例 9.4 可编程串行通信接口芯片8251A 9.4.1 串行数据传送方式 9.4.2 传输速率和传送距离 9.4.3 同步串行通信与异步串行通信 9.4.4 通用可编程串行通信接口芯片8251A 习题 第10章 中断与中断管理 10.1 中断概念 10.1.1 中断与中断源 10.1.2 中断系统的功能 10.1.3 简单的中断处理过程 10.1.4 中断源识别及优先权判断 10.2 8086的中断系统 10.2.1 8086的中断类型 10.2.2 中断向量和中断向量表 10.2.3 8086中的中断响应和处理过程 10.3 可编程中断控制器8259A 10.3.1 8259A的结构 10.3.2 8259A的引脚 10.3.3 8259A的中断处理过程 10.3.4 8259A的工作方式 10.3.5 8259A的编程与应用 10.4 中断程序设计 10.4.1 中断设计方法 10.4.2 中断程序设计举例 习题 第11章 数模与模数转换及应用 11.1 物理信号到电信号的转换 11.1.1 概述 11.1.2 几种常见的传感器 11.2 数模转换及应用 11.2.1 数模转换器的基本原理 11.2.2 数模转换器的性能参数 11.2.3 8位D/A转换器DAC0832 11.3 模数转换及应用 11.3.1 模数转换器的基本原理 11.3.2 模数转换器的性能参数 11.3.3 8位A/D转换器ADC0808/0809 习题 第12章 总线 12.1 总线的概念 12.2 系统总线 12.2.1 ISA总线 12.2.2 EISA总线 12.2.3 PCI总线 12.3 外部总线 12.3.1 RS-232C串行总线 12.3.2 USB总线 习题 第13章 Proteus ISIS仿真基础实例 13.1 基本I/O应用——I/O译码 13.1.1 功能说明 13.1.2 Proteus电路设计 13.1.3 代码设计 13.1.4 仿真分析与思考 13.2 定时/计数器8253的应用——波形发生器 13.2.1 功能说明 13.2.2 Proteus电路设计 13.2.3 代码设计 13.2.4 仿真分析与思考 13.3 并行接口芯片8255A的

应用——键盘与数码管 13.3.1 功能说明 13.3.2 Proteus电路设计 13.3.3 代码设计 13.3.4 仿真分析与思考
13.4 中断应用——8259A芯片的使用 13.4.1 功能说明 13.4.2 Proteus电路设计 13.4.3 代码设计 13.4.4 仿真分
析与思考 13.5 模数转换——ADC0808的使用 13.5.1 功能说明 13.5.2 Proteus电路设计 13.5.3 代码设计
13.5.4 仿真分析与思考 13.6 数模转换——DAC0832的使用 13.6.1 功能说明 13.6.2 Proteus电路设计 13.6.3
代码设计 13.6.4 仿真分析与思考 13.7 串行通信——8251A的使用 13.7.1 功能说明 13.7.2 Proteus电路设计
13.7.3 代码设计 13.7.4 仿真分析与思考 附录A VSM仿真的元件库 参考文献

章节摘录

版权页：插图：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：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。

系统BIOS的启动代码首先要做的事情就是进行POST。

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。

<<微机原理与接口技术>>

编辑推荐

《微机原理与接口技术:基于8086和Proteus仿真》是普通高等教育“十二五”规划教材,电子电气基础课程规划教材之一。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>