

<<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：9787121116025

10位ISBN编号：7121116022

出版时间：2010

出版时间：电子工业出版社

作者：周锋

页数：306

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

本书是计算机通信国家级教学团队成果和北京市精品教材立项项目。

全书以8086处理器和16位微型计算机系统为实例进行讲解，共分10章，主要包括微机基础、微处理器的功能结构、指令系统与寻址方式、汇编语言程序设计、处理器外部特性与输入/输出、存储器、中断与定时技术、常用接口技术、外部设备接口、总线等内容。

本书提供配套PPT课件和相关教学资源，读者可以通过华信教育资源网([www.hxedu.com.cn](http://www.hxedu.com.cn))免费注册下载。

本书适合作为高等学校理工科各专业微机原理与接口技术教材，也是相关技术人员或爱好者的参考书。

## &lt;&lt;微机原理与接口技术&gt;&gt;

## 书籍目录

第1章 微机基础 1.1 计算机中的数和字符 1.1.1 进位制 1.1.2 数的机器表示 1.1.3 字符编码 1.2 微机的基本结构 1.2.1 冯诺依曼结构 1.2.2 微机的硬件组成 1.3 微机的软件系统 1.3.1 指令和指令系统 1.3.2 程序和语言 1.3.3 微机软件系统 1.4 微机系统的发展 1.4.1 微机的诞生 1.4.2 微处理器的发展 1.4.3 微机系统的发展 习题1第2章 微处理器的功能结构 2.1 8086的内部结构 2.2 8086的寄存器结构 2.2.1 通用寄存器 2.2.2 段寄存器 2.2.3 指令指针寄存器 2.2.4 标志寄存器 2.3 8086的存储组织 2.3.1 物理空间与字节序 2.3.2 逻辑空间 2.3.3 存储管理 2.4 8086的I/O组织 2.5 32位微处理器 2.5.1 内部结构 2.5.2 寄存器结构 2.5.3 工作模式 2.5.4 保护模式下的存储组织 2.5.5 任务管理 习题2第3章 指令系统与寻址方式 3.1 8086寻址方式 3.1.1 指令 3.1.2 操作数 3.1.3 寻址方式 3.2 8086指令格式 3.3 8086指令系统 3.3.1 数据传送指令 3.3.2 算术运算指令 3.3.3 逻辑指令 3.3.4 串指令 3.3.5 控制转移指令 3.3.6 处理器控制指令 3.4 80x86寻址方式及指令的扩充 3.4.1 寻址方式的扩充 3.4.2 指令系统的扩充 习题3第4章 汇编语言程序设计 4.1 汇编语言概述 4.1.1 汇编语言的特点 4.1.2 汇编程序 4.1.3 汇编语言的组成 4.2 伪指令 4.2.1 程序格式 4.2.2 段定义 4.2.3 变量定义 4.2.4 常数、变量、标号、表达式 4.2.5 符号定义 4.2.6 地址对齐 4.2.7 结构定义 4.3 程序的基本结构 4.3.1 基本的程序框架 4.3.2 程序结构 4.3.3 顺序结构 4.3.4 分支结构 4.3.5 循环结构 4.4 子程序结构 4.4.1 子程序定义 4.4.2 子程序调用和返回 4.4.3 环境的保存和恢复 4.4.4 参数的传递 4.4.5 高级语言参数传递 4.4.6 递归 4.5 宏和条件汇编 4.5.1 宏 4.5.2 条件汇编 习题4第5章 处理器外部特性与输入/输出 5.1 处理器外部特性 5.1.1 8086的引脚功能 5.1.2 8086工作时序 5.1.3 两种模式下系统的典型连接 5.2 I/O接口概述 5.2.1 I/O接口的功能 5.2.2 I/O接口的基本结构 5.2.3 I/O端口的编址 5.2.4 I/O端口地址译码 5.2.5 基本输入/输出接口 5.3 处理器与外设的数据传送方式 5.3.1 程序控制方式 5.3.2 中断方式 5.3.3 直接存储器访问方式 习题5第6章 存储器 6.1 多级存储体系 6.2 半导体存储器 6.2.1 半导体存储器的分类 6.2.2 RAM的结构 6.2.3 现代RAM 6.3 内存储器的组织 6.3.1 内存储器的结构 6.3.2 存储体的构造 6.3.3 存储器与CPU的连接 6.4 Cache 6.4.1 Cache的基本原理 6.4.2 Cache的结构 6.4.3 Cache的策略 6.4.4 Pentium的Cache 习题6第7章 中断与定时技术 7.1 中断的基本概念 7.1.1 中断 7.1.2 中断源及中断源识别 7.1.3 中断优先级 7.1.4 中断嵌套 7.1.5 中断处理过程 7.2 8086的中断系统 7.2.1 8086中断类型 7.2.2 中断向量表 7.2.3 8086中断响应过程 7.3 可编程中断控制器8259A 7.3.1 8259A的内部结构及引脚 7.3.2 8259A的工作方式 7.3.3 8259A的编程 7.4 8259A应用举例 7.5 硬件中断服务程序的编写 7.6 定时与计数技术 7.6.1 可编程定时/计数器8253/8254 7.6.2 8253/8254工作方式 7.6.3 8253/8254编程 7.6.4 8253/8254的应用 习题7第8章 常用接口技术 8.1 可编程并行接口8255A 8.1.1 8255A的内部结构及引脚 8.1.2 8255A的控制字 8.1.3 8255A的工作方式 8.1.4 8255A的应用 8.2 串行接口 8.2.1 串行通信基础 8.2.2 串行接口标准 8.2.3 可编程串行通信接口Intel 8251 8.3 DMA传输 8.3.1 DMA传输过程 8.3.2 DMA控制器8237A 8.3.3 8237A应用实例 8.4 数/模、模/数转换 8.4.1 模拟输入/输出系统 8.4.2 数/模转换器接口 8.4.3 模/数转换器接口 习题8第9章 外部设备接口 9.1 概述 9.2 键盘 9.2.1 键盘接口 9.2.2 PC键盘及接口 9.3 打印机接口 9.4 显示器 9.4.1 LED显示器 9.4.2 CRT显示器及接口 9.4.3 液晶显示器 习题9第10章 总线 10.1 概述 10.1.1 总线的发展过程及分类 10.1.2 总线的传送过程 10.1.3 总线控制方式 10.1.4 总线的性能参数 10.2 ISA总线 10.3 PCI总线 10.3.1 PCI体系的基本概念 10.3.2 PCI总线的特点 10.3.3 PCI总线信号定义 10.3.4 Pci总线事务 10.4 USB总线 10.4.1 概述 10.4.2 USB体系结构 10.4.3 USB总线协议 习题10参考文献

## &lt;&lt;微机原理与接口技术&gt;&gt;

## 章节摘录

版权页：插图：8.2.3 可编程串行通信接口Intel 8251 Intel 8251A是可编程的串行通信接口芯片，它的主要特点如下：有两种工作方式。

可实现串行同步通信，也可实现串行异步通信。

在同步方式下，波特率为0~64kBaud/s；在异步方式下，波特率为0~19.2kBaud/s。

同步方式下，每个字符可以用5、6、7或8位来表示。

可采用单同步符或双同步符，同步符可由用户自行设定。

并且内部能自动检测同步字符，或由外部电路检测同步符，从而实现同步。

除此之外，8251A也允许同步方式下增加奇/偶校验位进行校验。

异步方式下，每个字符也可以用5、6、7或8位来表示，用1位作为奇偶校验，可有1位、1.5位或2位停止位。

时钟频率为传输波特率的1、16或64倍。

并能自动检测和处理终止字符。

全双工的工作方式，其内部提供具有双缓冲器的发送器和接收器。

提供出错检测，具有奇偶、溢出和帧错误三种校验电路。

1.8251A的外部引脚 8251A共有28个引脚，采用双列直插式封装，各引脚的功能如下。

(1) 面向CPU的连接信号 D7~D0：三态双向数据线。

(RD)：读信号线，输入，低电平有效。

(WR)：写信号线，输入，低电平有效。

(CS)：片选信号线，输入，低电平有效。

C/(D)：控N/数据端口选择线，输入，C/(D)为0时选择数据端口；C/(D)为1时选择控制端口。

可以看出，8251A内部占用两个端口地址。

RESET：复位信号线，输入，高电平有效，且高电平的宽度必须大于6个时钟信号宽度。

TxRDY：发送器准备好信号，输出，高电平有效。

当发送保持寄存器为空，且允许发送(CTS)为低电平以及发送器允许发送时，TxRDY为高电平。

CPU写入一个字符后，TxRDY变为低电平。

TxRDY可用作发送中断请求信号。

TxEMPTY：发送器空，输出，高电平有效。

发送器空指的是发送移位寄存器空，此时，发送器处于空闲状态。

RxRDY：接收器准备好信号，输出，高电平有效。

接收器收到一个字符后RxRDY变为高电平，字符被CPU读走后，又恢复为低电平。

RxRDY可用作接收中断请求信号。

## <<微机原理与接口技术>>

### 编辑推荐

《高等学校工程创新型"十二五"规划计算机教材:微机原理与接口技术》适合作为高等学校理工科各专业微机原理与接口技术教材，也是相关技术人员或爱好者的参考书。

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>