

高等院校信息技术规划教材

微机接口技术实用教程

任向民 编著

INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY

清华大学出版社



高等院校信息技术规划教材

微机接口技术实用教程

任向民 编著

清华大学出版社
北京

内 容 简 介

本书系统地介绍微型计算机接口技术及应用。全书共分9章,主要内容包括输入/输出接口技术、定时/计数技术、中断技术、DMA技术等,还介绍了当前PC普遍使用的USB接口技术和PCI总线等最新知识。

本书内容简明扼要,面向系统,面向应用,精选了微机接口技术在实际应用中的例子。每章都配有单元实验和标准化试题及参考答案,便于读者进行单元测试和复习。

本书以突出“应用”、注重“技能”为目标,适合作为高等学校计算机及相关专业本科生教材以及工科专业教学参考书,也可作为高等学校成人教育的培训或自学参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

微机接口技术实用教程/任向民编著. —北京:清华大学出版社,2008.3

(高等院校信息技术规划教材)

ISBN 978-7-302-16905-5

I. 微… II. 任… III. 微型计算机—接口—高等学校—教材 IV. TP364.7

中国版本图书馆CIP数据核字(2008)第009206号

责任编辑:袁勤勇 顾 冰

责任校对:梁 毅

责任印制:孟凡玉

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175

邮购热线:010-62786544

投稿咨询:010-62772015

客户服务:010-62776969

印 刷 者:北京市昌平环球印刷厂

装 订 者:三河市李旗庄少明装订厂

经 销:全国新华书店

开 本:185×260 印 张:14.75

字 数:341千字

版 次:2008年3月第1版

印 次:2008年3月第1次印刷

印 数:1~4000

定 价:21.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:026156-01

前言

Foreword

随着计算机科学技术的不断发展,微型计算机得到了广泛的应用,包括自动控制、智能化仪表、家用电器等领域,而微机接口技术是人们利用计算机设计和开发各种应用系统的基础。微机接口技术既包括硬件知识,又包括软件知识,难于掌握,本教材从系统角度出发,在简明扼要的讲述原理的同时,突出应用,通过实例使读者能够较容易地学习和掌握接口技术的基本内容,为后续计算机专业课程的学习打下良好的基础。

本书共分9章。第1章介绍微型计算机接口技术的基础知识,内容包括接口技术概述、接口传输的信息、编址、译码,接口的设计与分析等;第2章介绍总线的基本概念,常用的系统总线(ISA、EISA、VESA、PCI)和外部总线(IEEE 488、RS-232C、SCSI、IEEE 1394、USB);第3~8章介绍并行输入/输出接口(8255A)、串行通信接口(8251A)、定时/计数技术(8253)、中断技术(8259A)、DMA技术(8237A)、D/A与A/D接口(DAC0832、DAC1210、ADC0809、AD574A)的基本原理,以及各个芯片的特点、结构、引脚及功能、控制字、初始化编程和综合程序设计;第9章介绍常用外围设备(包括键盘、显示器、打印机、鼠标、扫描仪等)的基本原理和接口技术。

本书源于教学实践,凝聚了一线任课教师的教学经验,具有以下特点:

- 充分体现知识内容的基础性和系统性,突出“应用”。
- 知识内容具有先进性,特别是技术性、应用性内容。
- 本书配有单元实验,并提供实验参考程序(单元实验均在SAC-86S单板机上调试成功),以提高学生应用接口技术的能力和综合设计能力。
- 本书配有单元测试,有不同难易程度的标准化习题,并附有参考答案,供教师和学生进行测试和练习。
- 本书内容的组织方式深入浅出,循序渐进,选用内容丰富的应用实例,对基本概念、基本技术与方法的阐述准确明晰,通俗易懂。



本书适合作为各类高等学校本科计算机各专业教材以及工科专业教学参考书,也可作为高等学校成人教育的培训或自学参考书。

本书由任向民、宗明魁编写并统稿,在编写过程中得到了清华大学出版社和沈阳沈飞电子科技发展有限公司同志的大力支持和帮助,在此表示衷心的感谢。同时在编写过程中对参考的大量文献资料的作者一并谢意。由于时间仓促和水平所限,书中难免有欠妥之处,敬请专家、读者不吝批评指正。

编 者

2007 年 10 月

目录

Contents

第 1 章 微型计算机接口技术基础	1
1.1 接口技术概述	1
1.1.1 接口的定义	1
1.1.2 接口的功能特点	1
1.1.3 接口分类	3
1.2 接口中传输信息及其组成	3
1.2.1 数据信息	3
1.2.2 状态信息	4
1.2.3 控制信息	4
1.2.4 接口电路的一般结构	4
1.3 接口的编址与译码	5
1.3.1 端口分类	5
1.3.2 I/O 端口的编址方法	6
1.3.3 I/O 端口地址的译码方法	6
1.3.4 CPU 的 I/O 指令	9
1.4 CPU 与外设之间的数据传送方式	9
1.4.1 程序控制方式	9
1.4.2 中断传送方式	10
1.4.3 DMA 传送方式	10
1.5 接口技术的现状及发展	11
1.6 接口的设计与分析	11
1.6.1 接口两侧的分析和设计	12
1.6.2 信号转换	12
1.6.3 驱动程序	12
1.6.4 应注意的问题	12
单元测试 1	13



第 2 章 总线	16
2.1 总线概述	16
2.1.1 总线定义	16
2.1.2 总线的标准	16
2.1.3 总线的分类	17
2.1.4 采用标准总线的优点	18
2.1.5 总线的操作过程	18
2.1.6 总线通信方式	19
2.1.7 总线仲裁	19
2.2 系统总线	20
2.2.1 ISA 总线	21
2.2.2 EISA 总线	23
2.2.3 VESA 总线	24
2.2.4 PCI 局部总线	24
2.2.5 PCI-X 总线	26
2.2.6 PCI Express 总线	26
2.3 外部总线	28
2.3.1 IEEE 488 总线	28
2.3.2 RS-232C 总线	29
2.3.3 SCSI 总线简介	31
2.3.4 IEEE 1394 总线简介	32
2.3.5 USB 总线简介	33
2.3.6 USB 总线与 IEEE 1394 总线的比较	35
单元测试 2	36
第 3 章 并行输入/输出接口	40
3.1 并行接口的基本概念	40
3.2 并行 I/O 接口 8255A	41
3.2.1 8255A 的基本特性	41
3.2.2 8255A 的内部结构	41
3.2.3 8255A 的引脚及功能	42
3.2.4 8255A 的控制字	43
3.2.5 8255A 初始化编程	44
3.2.6 8255A 的工作方式	45
单元实验 8255A 并行接口实验	54
单元测试 3	56

第 4 章 串行接口	58
4.1 串行通信的基本概念	58
4.1.1 传输速率	58
4.1.2 数据传送模式	58
4.1.3 串行通信方式	59
4.1.4 校验方式	60
4.1.5 调制与解调	61
4.2 可编程串行通信接口 8251A	62
4.2.1 8251A 的基本特性	62
4.2.2 8251A 的内部结构	62
4.2.3 8251A 的引脚及功能	64
4.2.4 8251A 的控制字	65
4.2.5 8251A 初始化编程	67
4.2.6 8251A 应用实例	68
单元实验 8251A 接口实验	72
单元测试 4	75
第 5 章 定时/计数技术	78
5.1 定时/计数的基本概念	78
5.1.1 系统定时/计数必要性	78
5.1.2 系统定时的分类	78
5.2 可编程定时器/计数器 8253	79
5.2.1 8253 的主要特性	79
5.2.2 8253 内部结构	79
5.2.3 8253 的引脚及功能	80
5.2.4 8253 方式控制字	82
5.2.5 8253 工作方式	83
5.3 8253 程序设计	87
5.3.1 8253 初始化编程	87
5.3.2 8253 应用实例	88
单元实验 8253 定时计数器实验	90
单元测试 5	92
第 6 章 中断技术	95
6.1 中断的基本概念	95
6.1.1 中断请求与中断屏蔽	95



6.1.2	中断服务程序	96
6.1.3	中断隐操作和中断向量	96
6.1.4	中断响应过程	96
6.2	8086/8088 中断系统	98
6.2.1	中断系统的基本功能	98
6.2.2	8086/8088 的中断系统	98
6.3	可编程中断控制器 8259A	100
6.3.1	8259A 的主要特性	100
6.3.2	8259A 的内部结构	101
6.3.3	8259A 的引脚及功能	102
6.3.4	8259A 控制字	103
6.3.5	8259A 工作方式	108
6.4	8259A 程序设计	111
6.4.1	8259A 初始化编程	112
6.4.2	工作方式编程	112
6.4.3	综合程序设计	113
单元实验	8259A 中断实验	116
单元测试 6		125
第 7 章 DMA 控制器		130
7.1	DMA 的基本概念	130
7.1.1	直接存储器存取 DMA 概念	130
7.1.2	DMA 控制器组成与功能	130
7.1.3	DMA 操作步骤	132
7.2	可编程 DMA 控制器 8237A	133
7.2.1	8237A 主要特性	133
7.2.2	8237A 内部结构	134
7.2.3	8237A 的引脚及功能	135
7.2.4	8237A 方式控制字	137
7.2.5	内部寄存器的寻址	142
7.2.6	软件命令	143
7.3	8237A 程序设计	144
7.3.1	8237A 初始化编程	144
7.3.2	8237A 应用实例	145
单元实验	8237A 控制器实验	147
单元测试 7		150

第 8 章 A/D 和 D/A 接口	153
8.1 A/D 和 D/A 转换的基本概念	153
8.1.1 信号的转换	153
8.1.2 常用传感器	154
8.2 D/A 转换及接口	156
8.2.1 D/A 转换的工作原理	156
8.2.2 D/A 的性能参数和术语	157
8.2.3 典型 D/A 转换器芯片	159
8.3 A/D 转换器接口	165
8.3.1 模拟信号的采样、量化和编码	166
8.3.2 A/D 转换的类型	167
8.3.3 A/D 转换器主要性能参数	167
8.3.4 典型 A/D 转换器芯片	168
单元实验 A/D 与 D/A 转换接口实验	176
单元测试 8	181
第 9 章 常用外围设备及接口	184
9.1 常用外围设备及接口基本知识	184
9.1.1 外围设备的功能	184
9.1.2 外围设备的分类	185
9.2 键盘及其接口	185
9.2.1 键盘	185
9.2.2 键的识别	186
9.2.3 微机键盘及接口	189
9.3 显示器及其接口	190
9.3.1 LED 显示器及其接口	190
9.3.2 CRT 显示器及其接口简介	194
9.3.3 LCD 显示器简介	197
9.4 打印机及其接口	198
9.4.1 打印机的分类	198
9.4.2 打印机的主要性能指标	199
9.4.3 针式打印机	200
9.4.4 喷墨打印机	201
9.4.5 激光打印机	202
9.4.6 打印机接口	203
9.5 交互式人机接口	205



9.5.1 鼠标器	205
9.5.2 扫描仪	207
9.5.3 光笔	209
9.5.4 数字化仪	210
单元实验 8279 键盘显示实验	210
单元测试 9	213
 单元测试参考答案	 216
 参考文献	 224

微型计算机接口技术基础

随着现代计算机技术的不断发展,其外设的种类和形式多种多样,使得 CPU 由过去的对外设直接进行控制和管理发展到利用接口来实现 CPU 与外设之间的信息交换。可见接口技术在微型计算机系统设计与应用过程中有着非常重要的地位。CPU 要与输入/输出设备交换信息,就要借助于接口来实现。

1.1 接口技术概述

输入/输出(input/output,I/O)是计算机与外部世界交换信息所必需的手段。如程序、数据和现场物理量等要通过输入设备进入计算机,计算机运行的结果和各种控制信号要通过输出设备进行显示、打印或实现实时控制等。计算机的外设有机械式、电子式、机电式等,输入/输出信号的形式有数字量、模拟量,信息传送方式有串行、并行等,可见种类繁多,采用的信号和数据传送形式、速度有差异。

因此,当要把这些外部设备与主机连接时,在 CPU 与外设之间要设置一种部件,使 CPU 和外设协调工作,有效地完成 CPU 与外界的信息交换,这种起到界面(interface)作用的部件称为输入/输出接口电路。

1.1.1 接口的定义

接口是微处理器(CPU)与外界的连接部分(电路),是 CPU 与外界进行信息交换的中转站。接口技术是研究 CPU 如何与外部世界进行最佳耦合与匹配,以实现双方高效、可靠地交换信息的一门技术,它是软硬件结合的体现,是微型计算机应用的关键。

1.1.2 接口的功能特点

接口部件作为 CPU 与外设之间的一个界面,使得双方有条不紊地协调动作,从而完成 CPU 与外界的信息交换。按 CPU 与外界交换信息的要求,下面介绍接口部件应具有功能特点。

1. 寻址功能

接口要对选择存储器和输入、输出的信号做出解释,要对送来的片选信号进行识别,

判断当前接口是否被访问,决定接口中的那个寄存器或端口受到访问。

2. 输入/输出功能

接口根据读/写信号决定当前是输入操作还是输出操作,即从总线上接收来自 CPU 的数据和控制信息,或将数据和状态信息送到总线上。

3. 数据缓冲功能

CPU 与外设的工作速度有着明显的差异,CPU 的工作速度较快,外设的工作速度相对于 CPU 速度较慢,为缓解高速 CPU 和低速外设之间的矛盾,接口中设置了数据寄存器或锁存器,用于存放和传输数据,提供状态信息,以避免丢失数据。

4. 设备选择功能

CPU 在同一时间里只能与多台外设中的一台交换信息,这就需要通过接口的地址译码对外设进行寻址,选中与 CPU 进行交换信息的外设。

5. 信号转换功能

外设所能提供和所需要的各种信号常常与微机内部信号形式不同,这就需要对信号进行转换。信号转换包括电平转换、模拟信号—数字信号和数字信号—模拟信号转换、串行—并行和并行—串行转换、数据宽度变换及信号的逻辑关系和时序上的配合所要求的变换等。

6. 接受、解释并执行 CPU 命令的功能

CPU 对外设发出的各种命令都是以代码的形式先到接口电路,再由接口电路解释后,形成一系列控制信号送往外设。

7. 中断管理功能

当外设需要及时得到 CPU 的服务,如在出现故障而要求 CPU 进行处理时,可以通过在接口中设置中断控制逻辑,由它完成向 CPU 提出中断请求,进行中断优先级排队,接收中断响应信号以及向 CPU 提供中断向量等有关中断事务工作。这样,不仅能使 CPU 实时处理紧急情况,还能使快速 CPU 与慢速外设并行工作,大大提高 CPU 的效率。

8. 可编程功能

为使接口具有较强的通用性、灵活性、适应性和可扩充性,现在的接口芯片多数都是可编程的,这样在不改变硬件的条件下,只改变驱动程序就可改变接口的工作方式和功能,以适应不同的用途。

9. 错误检测功能

错误检测在接口的设计中是必须考虑的方面,现在的可编程接口芯片一般能够检测

传输错误和覆盖错误。由于噪声干扰引起的传输错误,接口采用奇偶校验位对传输错误进行检测。此时,在状态寄存器中设置相应的状态位,可以标识由于输入/输出缓冲寄存器中的数据没有及时取走而被新传输的数据所覆盖而产生的覆盖错误。

10. 复位功能

接口可以接收复位信号,使接口本身以及所连的外设具有重新启动功能。

上述功能不是每个接口芯片(电路)都同时具备,不同条件和环境下的系统,其接口芯片的功能及实现方式有所不同,接口电路的复杂程度也各不相同。

1.1.3 接口分类

接口的种类有很多,根据不同的划分方法,可分为以下几类。

(1) 根据通用性可分为专用接口和通用接口。专用接口是为某种用途或某种外设专门设计的接口电路,如 DMA 控制器。通用接口是多种外设可用的接口电路。

(2) 根据可编程性可分为可编程接口和不可编程接口。可编程接口是在不改动硬件的前提下,用户可以通过程序设计方法对接口的工作方式等进行调整,增加了接口的灵活性和可扩展性。

(3) 根据数据的传送方式可分为并行接口和串行接口。并行接口是以字节或字为单位进行传送,串行接口则是按位进行传送。

(4) 根据工作对象可分为面向 CPU 的外围接口和面向外设的 I/O 接口。面向 CPU 的外围接口和 CPU 配套使用,以提高 CPU 的性能,例如中断控制器,可以提高 CPU 的中断控制能力。面向外设的 I/O 接口可将不同类型的外设连接到当前系统中。

1.2 接口中传输信息及其组成

为了完成 CPU 与 I/O 设备之间的信息交换,通常在接口部件中需要传输三种信息。即数据信息、状态信息和控制信息。

1.2.1 数据信息

数据信息是指 CPU 与外设之间交换的基本信息。数据信息形式可分为数字量、模拟量和开关量三种。

1. 数字量

通常以 8 位或 16 位的二进制或 ASCII 码形式传输。例如,由键盘读入的信息,主机送给显示器、打印机的信息。

2. 模拟量

模拟量指电压或电流,以及温度、湿度、位移、压力、流量等非电信号经过传感器转换

成的连续变化的电信号。模拟量经 A/D 转换器变成数字量形式进行传输。例如在控制系统中,多数情况下输入信息是现场连续变化的物理量(模拟量),经过 A/D 转换后才能送入计算机内部。

3. 开关量

开关量即用一位二进制位(“0”或“1”)表示两种状态,如开关的通与断、电机的转与停、阀门的开与关等。

1.2.2 状态信息

用于表示外设工作状态的信号叫状态信息,是由外设通过接口传递到 CPU 的。状态信息反映了外设所处的现行状态,如打印机是否忙(Busy),输入设备是否准备好(Ready)等。

1.2.3 控制信息

控制信息是 CPU 通过接口传送给外设的。在 CPU 与外设的信息交换过程中,需要向外设发布控制命令,这些控制信号由 CPU 发给接口电路,经接口电路解释并做适当变换后(根据实际需要),去控制外设的工作。例如外设的启动信号和停止信号是最基本的控制信息。

1.2.4 接口电路的一般结构

接口电路的一般结构如图 1-1 所示。

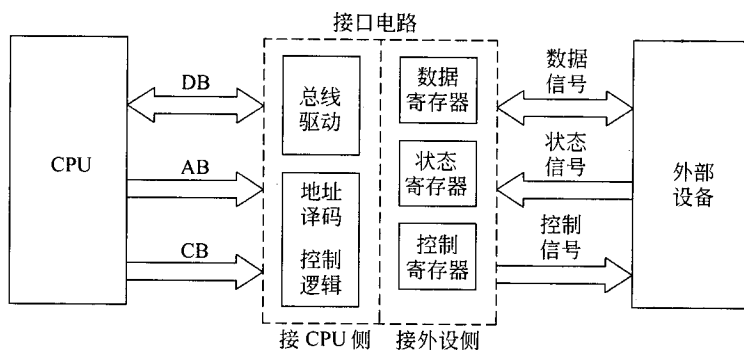


图 1-1 接口电路的结构

1. 接外设侧

在接外设侧的接口电路中,主要包括数据寄存器、状态寄存器和控制寄存器。其中,数据寄存器用于暂时存放输入(来自于外设的)或输出(CPU 写给外设的)的数据;状态寄存器用于暂时存放外设的工作状态信息,供 CPU 进行查询(或向 CPU 申请的中断信息

等),状态寄存器中的内容一般是只读的;控制寄存器用于暂时存放 CPU 发给外设的控制命令,如接口的工作方式命令字,控制寄存器是写寄存器,只能写入,不能读出。

2. 接 CPU 侧

在接 CPU 侧的接口电路中,主要包括总线驱动、地址译码和控制逻辑等。其中,总线驱动器用于实现速度配合和满足驱动能力的需要;地址译码器用于实现对内部寄存器的寻址;控制逻辑电路包括读写控制信号、复位信号等。

对于上述结构,并不是所有的接口都含有,但数据寄存器、地址译码、控制逻辑是必需的。

1.3 接口的编址与译码

由图 1-1 可知,接口电路通常包含若干个寄存器,数据传送时,不同的信息送入不同的寄存器。能够用 IN/OUT 指令对其进行读/写操作的寄存器称为端口寄存器,即能被 CPU 直接访问的寄存器,简称“端口”。

1.3.1 端口分类

在接口电路中,按接口寄存器存放信息的物理意义来分,端口可分为数据端口(数据口)、状态端口(状态口)和控制端口(控制口)三类。

1. 数据端口

数据端口存放数据信息,在输入过程中,数据信息是由外设经过接口电路中的数据端口,此时可使用 IN 指令进行读取,CPU 获得数据。在输出过程中,可执行 OUT 指令将数据信息从 CPU 经过数据总线送入接口电路的数据端口,在通过接口和外设间的数据线送到外设。

2. 状态端口

状态端口存放状态信息,表示外设当前工作状态的信息,CPU 可通过读取这些信息来查询当前的工作情况。在输入接口电路中,状态信息能够表示出输入数据是否准备好的状态,在输出接口电路,状态信息能表示出输出设备忙、闲的状态。

3. 控制端口

控制端口存放控制信息,控制信息是 CPU 通过接口传送给外设的,以控制外设的工作。

数据信息、状态信息和控制信息是不同性质的信息,需要分别进行传送。在大部分微型计算机中,用输入指令(IN)完成输入和用输出指令(OUT)完成输出。这样,为区别输入是数据信息还是状态信息,只要数据端口和状态端口有不同的端口地址就可以了,

为区别输出是数据信息还是控制信息,数据端口和控制端口也必须有不同的端口地址。所以一个接口电路有若干个端口地址。在接口电路中必须有数据口,有一个或多个状态口和控制口。

1.3.2 I/O 端口的编址方法

I/O 端口编址有两种方式,一种是把 I/O 端口与存储器地址统一编址,即采用存储器映像方式;另一种是将 I/O 端口地址与存储器地址分开独立编址,即采用 I/O 映像方式。

1. 采用存储器映像方式(统一编址)

统一编址是将存储空间划出一部分给 I/O 端口。对 I/O 操作与对存储单元操作完全相同。其特点是 I/O 操作能使用对存储器操作的所有指令,不需要专门的 I/O 指令,它使 CPU 访问 I/O 端口的操作比较灵活、方便,有利于提高端口数据处理速度。但 I/O 端口占用了有效的存储器空间。

2. I/O 映像方式(独立编址)

独立编址是指 I/O 端口地址不占用存储空间,所有的 I/O 端口地址单独构成一个 I/O 空间,访问 I/O 端口使用专用 I/O 指令(IN/OUT)。它是微机系统中普遍采用的一种编址方式,其特点是取周期较短,效率较高,寻址范围较大,但需要专门的信号区分是访问存储器还是 I/O 端口。

1.3.3 I/O 端口地址的译码方法

CPU 在对 I/O 端口进行操作时,必须提供所要读/写端口的地址,这时需要端口地址译码来识别和确认端口。

I/O 端口地址译码与存储器空间译码类似,包括片选和字选(端口选择)两部分,片选由外译码电路实现,用来确定接口芯片内部端口的地址范围,字选由芯片内部地址译码电路实现,用来确定接口芯片内部端口的具体地址。也就是说在寻址某个端口时,除了找到该芯片外,还要能区分出不同端口。端口地址译码的方法可分为固定式地址译码和可选式地址译码。

1. 固定式地址译码

固定式端口地址译码是指接口中用到的端口地址不能更改,现在的各种 I/O 接口卡一般都采用该译码方法。当接口芯片有一个或两个端口时,可采用逻辑门电路组成译码电路,当接口芯片中有多个端口时,采用译码器比较方便,如双 2-4 译码器 74LS139、3-8 译码器 74LS138 和 4-16 译码器 74LS154 等。

1) 用逻辑门电路进行译码

用逻辑门电路进行译码是一种最简单最基本的端口地址译码方法,适用于系统中接