



普通高等教育“十一五”国家级规划教材

*Microcomputer Interface
Techniques & Application*



微型计算机 接口技术及应用

主编 刘乐善
编者 刘乐善 李 畅 刘学清



华中科技大学出版社

<http://www.hustp.com>



普通高等教育“十一五”国家级规划教材

*Microcomputer Interface
Techniques & Application*

微型计算机 接口技术及应用

主编 刘乐善

编者 刘乐善 李 畅 刘学清



华中科技大学出版社

<http://www.hustp.com>

中国·武汉

版 权 声 明

本书内容已申请技术专利,未经授权,不得擅自摘录,特此声明!

图书在版编目(CIP)数据

微型计算机接口技术及应用/刘乐善 主编. —武汉:华中科技大学出版社,2011.7
ISBN 978-7-5609-7006-6

I. 微… II. 刘… III. 微型计算机-接口技术 IV. TP364.7

中国版本图书馆 CIP 数据核字(2011)第 050998 号

微型计算机接口技术及应用

刘乐善 主编

策划编辑:周芬娜

责任编辑:王汉江

封面设计:潘 群

责任校对:张 琳

责任监印:张正林

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:武汉佳年华科技有限公司

印 刷:华中科技大学印刷厂

开 本:787mm×1092mm 1/16

印 张:25

字 数:651 千字

版 次:2011 年 7 月第 1 版第 1 次印刷

定 价:39.80 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

内 容 简 介

本书以微机接口为对象,深入地阐述了现代微机接口技术的原理及应用。全书共 13 章,前 10 章为接口技术的基本内容,集中介绍与分析了用户设备接口的共性技术,分别详细讨论了各种传统常用接口的电路设计及控制程序编写。后 3 章是接口技术新发展的内容,对 PCI 总线接口(桥)和 WDM 设备驱动程序的设计进行了深入具体讨论。

本书内容全面,具有较好的可操作性、可用性及可读性。在内容的组织与安排等方面具有特色。本书适用面宽,既可作为高等院校工科所有专业研究生、本科、专科教材或专业技术培训教材,也是广大从事微型计算机应用与开发人员值得一读的自学参考书。

前 言

在微机系统中,微处理器的强大功能必须通过外部设备(简称外设)才能实现,而外设与微处理器之间的信息交换及通信又是靠接口来实现的,所以,微机应用系统的研究和微机化产品的开发,从硬件角度来讲,就是接口的研究和开发,接口技术已成为直接影响微机系统的功能和微机推广应用的关键之一,也是新发展起来的嵌入式微机应用的基础技术。根据接口在微机系统中的作用和接口技术的基本任务,可以说接口技术是一门遍及微机应用各个领域的通用技术。因此,微机接口技术已成为当代理工科大学生必须学习的一种基本知识和科技与工程技术人员必须掌握的基本技能。

现代微机接口技术增加了许多新功能,采用了许多新技术、新方法,引入了许多新概念、新名词,需要逐步了解与学习。接口技术所涉及的知识面广泛,技术层次越来越高,而且实践性很强,这给学习和掌握微机接口技术带来一定的困难,因此需要相关的教材提供帮助,本书就是为此而编写的。

本教材与其他版本相比,在接口技术的基本概念、所采用的技术及处理问题的方法上都有很大的变化与发展。概括起来,本教材具有如下创新特点,并以此作为编写的指导思想。

● 新概念

教材提出接口技术分层次的概念,把接口整体内容分为上层设备接口与底层总线接口,即接口的基本内容与接口新发展的内容两个层次,理顺了接口技术的系统,形成了接口技术自身的体系结构,为全面学习与掌握微机接口技术指明了方向与目标。

● 新技术

教材采用 PCI 总线技术与设备驱动程序设计技术,为培养学生研究与开发现代微机接口应用系统打下高起点的基础,为实际从事 PCI 总线接口设计与设备驱动程序设计提供有力的参考。

● 新方法

教材引入软件模型法,在分析与设计接口技术的硬件时,采用软件模型法(程序设计模型法)进行处理,这样既可简化对硬件内部细节的了解,又不影响用户对计算机硬件资源的开发与应用。

以上的新概念与新方法是在微机接口技术类教材中首次提出与应用,是作者对微机接口技术课程多年教学研究与教材建设所获得的成果,并已申请技术专利,属于作者的权益加以保护,愿与广大读者共享。

在上述创新思维的指导下,本教材形成了自己的特色,主要有如下几点。

1. 教材反映接口技术发展的主流,技术先进

本书是讨论在以 PCI 为中心的多总线结构和 Windows 系统下的微机接口技术及应用,内容涉及现代微机 PCI 总线接口设计技术和 Windows 操作系统内核设备驱动程序设计技术,反映了当前微机接口技术发展的主流。

2. 教材层次分明,结构清晰,可操作性强

按照接口分层次的概念,全书分为基本内容和新发展的内容两个层次,第 2~10 章是基本内容,第 11~13 章是新发展的内容,层次分明,便于根据不同教学要求具体组织与安排教学内容,操作方便。

书中既有传统的常用外设接口,也有新型接口;既考虑了接口技术的共性,也考虑了各类接口的特点;既有上层设备接口,又有底层总线接口;既有用户态的接口应用程序,又有核心态的设备驱动程序,内容丰富、结构清晰。

3. 教材实例丰富,方法具体,可用性好

教材从实际应用出发,在讲清楚基本原理和概念的基础上,配合大量的应用实例,展示接口的设计原理、设计方法和设计步骤。这些实例来自科研与教学实践的成果,实用性强,具有较好的参考意义与可用性。通过把基本理论应用于解决实例中的一些问题可以训练与培养自己的实际工作能力。

4. 教材由浅入深,概念清晰,可读性强

对新概念、新技术、新方法的切入,从初学者的立场出发,由浅入深,在从已知到未知的对比分析中,逐步提升。同时通过各种实例帮助化解一些难以理解的问题。再加上教材概念清晰,语言通俗,图文并茂,使教材有较强的可读性,便于自学。

微机接口技术是一门实践性很强的课程,除了课堂理论学习之外,还需要强有力的实践性环节与之配合。为此,我们编写了实验教材,研制并推出了“32 位微机接口与原理实验平台”和“32 位微机应用与开发实训平台”。前者适于配合课堂教学实验,后者用于课程设计、毕业设计、实习和实际动手能力的实训等多种实践环节。实验系统和本教材的内容紧密配合,相互补充,教材中举出的接口实例,可以通过实验平台进行实际操作和实验,真正做到课堂原理讲授和实践环节一脉相承。

本书由刘乐善主编,李畅编写了第 13 章和第 11 章及第 12 章的程序,刘学清编写第 3 章和第 10 章,其余章节由刘乐善编写。全书由刘乐善统稿。本书的出版得到了全国高校计算机专业教学指导委员会和华中科技大学计算机科学与技术学院的大力支持,华中科技大学出版社为本书的出版付出了辛勤的劳动,在此一并表示衷心的感谢。同时要特别感谢参考文献的作者。

由于编者水平有限,书中错误和不足之处,望读者及专家赐正。

作 者

2010 年 12 月

目 录

第 1 章 概述	(1)
1.1 微机接口技术的作用与基本任务	(1)
1.2 微机接口技术的层次	(2)
1.3 微机接口技术内容的划分	(2)
1.3.1 接口技术的基本内容	(3)
1.3.2 接口技术的新内容	(3)
1.4 微机接口技术的基本概念	(3)
1.4.1 设备接口与总线桥	(3)
1.4.2 设备接口	(4)
1.4.3 总线桥	(6)
1.5 微机接口技术的发展概况	(7)
1.6 分析硬件的软件模型方法	(8)
1.7 本书的重点与内容安排	(9)
1.7.1 本书内容的重点	(9)
1.7.2 本书内容的安排	(10)
习题 1	(10)
第 2 章 总线技术	(11)
2.1 总线的作用	(11)
2.2 总线的组成	(11)
2.3 总线标准及总线的性能参数	(12)
2.4 总线传输操作过程	(13)
2.5 总线与接口的关系	(13)
2.6 ISA 总线	(14)
2.6.1 ISA 总线的特点	(14)
2.6.2 ISA 总线的信号线定义	(14)
2.6.3 ISA 总线与 I/O 设备接口的连接	(16)
2.7 现代微机总线技术的新特点	(16)
2.7.1 多总线技术	(17)
2.7.2 总线的层次化结构	(17)
2.7.3 总线桥	(18)
2.7.4 多级总线下接口与总线的连接	(18)
2.7.5 层次化总线结构对接口技术的影响	(19)
习题 2	(19)
第 3 章 I/O 端口地址译码技术	(21)
3.1 I/O 地址空间	(21)

3.2 I/O 端口	(21)
3.2.1 I/O 端口	(21)
3.2.2 I/O 端口共用技术	(22)
3.2.3 I/O 端口地址编址方式	(22)
3.2.4 独立编址方式的 I/O 端口访问	(23)
3.3 I/O 端口地址分配及选用的原则	(24)
3.3.1 PC 微机 I/O 地址的分配	(25)
3.3.2 现代微机 I/O 地址的分配	(25)
3.3.3 I/O 端口地址选用的原则	(26)
3.4 I/O 端口地址译码	(26)
3.4.1 I/O 地址译码的方法	(26)
3.4.2 I/O 地址译码电路的输入与输出信号线	(27)
3.5 I/O 端口地址译码电路设计	(27)
3.5.1 I/O 端口地址译码电路设计的几个问题	(27)
3.5.2 I/O 地址译码电路设计举例	(28)
习题 3	(33)
第 4 章 定时/计数技术	(34)
4.1 定时与计数	(34)
4.2 微机系统中的定时类型	(34)
4.3 外部定时方法及硬件定时器	(35)
4.4 可编程定时/计数器 82C54A	(36)
4.4.1 82C54A 的外部连接特性与内部结构	(36)
4.4.2 82C54A 的命令字	(37)
4.4.3 82C54A 的工作方式	(39)
4.4.4 82C54A 的计数初值计算及装入	(44)
4.4.5 82C54A 的初始化	(45)
4.5 定时/计数器的应用	(45)
4.5.1 82C54A 在微机系统中的应用设置	(46)
4.5.2 微机系统的 82C54A 初始化程序段	(47)
4.5.3 定时/计数器 82C54A 的应用举例	(47)
习题 4	(58)
第 5 章 中断技术	(60)
5.1 中断	(60)
5.2 中断类型	(60)
5.2.1 硬中断	(60)
5.2.2 软中断	(61)
5.3 中断号	(62)
5.3.1 中断号与中断号的获取	(62)
5.3.2 中断响应周期	(62)
5.3.3 中断号的分配	(63)

5.4	中断触发方式与中断排队方式	(64)
5.5	中断向量与中断向量表	(64)
5.5.1	中断向量与中断向量表	(65)
5.5.2	中断向量表的填写	(66)
5.6	中断处理过程	(66)
5.6.1	可屏蔽中断的处理过程	(67)
5.6.2	不可屏蔽中断和软件中断的处理过程	(67)
5.7	中断控制器	(67)
5.7.1	82C59A 外部特性和内部寄存器	(67)
5.7.2	82C59A 端口地址	(69)
5.7.3	82C59A 的工作方式	(69)
5.7.4	82C59A 的编程命令	(70)
5.7.5	82C59A 对中断管理的作用	(74)
5.8	可屏蔽中断系统	(75)
5.8.1	可屏蔽中断系统的组成	(75)
5.8.2	中断系统的初始化	(75)
5.9	用户对系统中断资源的使用	(77)
5.9.1	修改中断向量	(78)
5.9.2	发中断屏蔽/开放和中断结束命令	(79)
5.9.3	编写中断服务程序	(80)
5.10	中断服务程序设计	(80)
习题 5		(87)
第 6 章	DMA 技术	(89)
6.1	DMA 传输	(89)
6.1.1	DMA 传输的特点	(89)
6.1.2	DMA 传输的过程	(89)
6.2	DMA 操作	(90)
6.2.1	DMA 操作类型	(90)
6.2.2	DMA 操作方式	(91)
6.3	DMA 控制器与 CPU 之间的总线控制权转移	(91)
6.3.1	DMA 控制器的两种工作状态	(91)
6.3.2	DMA 控制器与 CPU 之间的总线控制权转移	(92)
6.4	DMA 控制器 82C37A	(93)
6.4.1	DMA 控制器的外部特性	(93)
6.4.2	内部寄存器及编程命令	(94)
6.4.3	DMA 控制器的工作时序	(100)
6.5	微机中 DMA 系统的组成与初始化	(101)
6.5.1	DMA 系统组成	(101)
6.5.2	DMA 系统初始化	(102)
6.6	用户对系统 DMA 资源的使用	(103)

6.6.1	DMA 传输参数设置	(103)
6.6.2	传输参数设置举例	(103)
习题 6		(105)
第 7 章	并行接口	(106)
7.1	并行接口的特点	(106)
7.2	并行接口电路结构形式	(106)
7.3	可编程并行接口芯片 82C55A	(107)
7.3.1	82C55A 外部特性	(107)
7.3.2	82C55A 内部结构	(108)
7.3.3	82C55A 的端口地址	(108)
7.3.4	82C55A 工作方式	(108)
7.3.5	82C55A 编程命令	(109)
7.4	82C55A 的 0 方式及其应用	(113)
7.5	82C55A 的 1 方式及其应用	(122)
7.5.1	1 方式下联络信号线的设置	(123)
7.5.2	1 方式的工作时序	(124)
7.5.3	1 方式的状态字	(125)
7.5.4	1 方式并行接口设计	(126)
7.6	82C55A 的 2 方式及其应用	(130)
7.6.1	2 方式下联络信号线的设置及时序	(130)
7.6.2	2 方式的状态字	(131)
7.6.3	2 方式并行接口设计	(131)
习题 7		(134)
第 8 章	串行通信接口	(136)
8.1	串行通信的基本概念	(136)
8.1.1	串行通信的基本特点	(136)
8.1.2	串行通信传输的工作方式(制式)	(136)
8.1.3	串行通信中的差错检测	(137)
8.1.4	串行通信的同步方式	(138)
8.1.5	串行通信中的调制与解调	(139)
8.2	串行通信中的传输速率控制	(140)
8.2.1	数据传输速率控制的实现方法	(140)
8.2.2	波特率与发送/接收时钟	(140)
8.2.3	波特率时钟发生器设计	(142)
8.3	串行通信中的数据格式	(146)
8.3.1	起止式异步通信数据格式	(146)
8.3.2	面向字符的同步通信数据格式	(147)
8.4	串行通信接口标准	(147)
8.4.1	RS-232C 接口标准	(147)
8.4.2	RS-485 接口标准	(151)

8.4.3 RS-232C 与 RS-485 的转换	(154)
8.5 串行通信接口电路	(154)
8.5.1 串行通信接口的基本任务	(155)
8.5.2 串行通信接口电路的组成	(155)
8.6 基于 8251A(USART)用户扩展串行通信接口	(155)
8.6.1 8251A 的外部特性	(156)
8.6.2 8251A 内部寄存器及编程命令	(156)
8.6.3 8251A 的初始化内容与顺序	(160)
8.6.4 基于 8251A 的串行通信接口设计	(160)
8.7 基于 16550(UART)的微机系统串行通信接口	(168)
8.7.1 16550 的外部引脚特性	(169)
8.7.2 16550 的内部寄存器及端口地址	(170)
8.7.3 16550 的编程	(174)
8.7.4 基于 16550 的串行通信接口设计	(178)
习题 8	(184)
第 9 章 A/D 与 D/A 转换器接口	(186)
9.1 模拟量接口	(186)
9.2 A/D 转换器	(186)
9.2.1 A/D 转换器的主要技术指标	(186)
9.2.2 A/D 转换器的外部特性	(187)
9.3 A/D 转换器与 CPU 接口的原理和方法	(188)
9.3.1 A/D 转换器与 CPU 的连接	(188)
9.3.2 A/D 转换器的数据传输	(188)
9.3.3 A/D 转换器接口控制程序中的在线数据处理	(189)
9.4 A/D 转换器接口设计	(189)
9.4.1 A/D 转换器接口设计方案的分析	(189)
9.4.2 A/D 转换器接口设计	(190)
9.5 D/A 转换器	(204)
9.5.1 D/A 转换器的主要技术指标	(204)
9.5.2 D/A 转换器的外部特性	(204)
9.6 D/A 转换器与 CPU 接口的原理和方法	(205)
9.6.1 D/A 转换器与 CPU 的连接	(205)
9.6.2 D/A 转换器接口的主要任务	(205)
9.6.3 D/A 转换器接口设计方案的分析	(205)
9.7 D/A 转换器接口电路设计	(206)
习题 9	(210)
第 10 章 基本人机交互设备的接口	(211)
10.1 键盘接口	(211)
10.1.1 键盘的类型	(211)
10.1.2 键盘的结构与工作原理	(211)

10.1.3	非编码键盘接口设计	(214)
10.2	LED 数码显示器接口	(218)
10.2.1	LED 显示器的结构与原理	(218)
10.2.2	LED 显示器的字形码	(219)
10.2.3	LED 显示器的显示方式	(219)
10.3	可编程键盘/LED 接口芯片 82C79A	(220)
10.3.1	外部特性与内部模块	(220)
10.3.2	编程命令与状态字	(223)
10.3.3	数码显示器接口设计	(226)
10.4	打印机接口	(231)
10.4.1	并行打印机接口标准	(232)
10.4.2	并行打印机接口设计	(234)
习题 10		(236)
第 11 章	PCI 总线接口	(237)
11.1	PCI 总线的特点	(237)
11.2	PCI 总线的信号定义	(238)
11.3	PCI 总线命令	(241)
11.4	PCI 总线数据传输	(242)
11.4.1	PCI 总线数据传输协议	(242)
11.4.2	PCI 总线数据传输过程	(243)
11.5	PCI 总线的三种地址空间	(245)
11.6	PCI 设备	(246)
11.7	PCI 总线配置空间	(247)
11.7.1	PCI 配置空间的作用	(247)
11.7.2	PCI 配置空间的格式	(247)
11.7.3	PCI 配置空间的功能	(247)
11.7.4	PCI 配置空间的映射关系	(252)
11.7.5	PCI 配置空间的访问	(254)
11.8	PCI 接口卡的设计	(263)
11.8.1	PCI 接口卡设计要求	(264)
11.8.2	PCI 总线接口卡的设计方案	(264)
11.8.3	PCI 总线接口芯片 PLX9054	(264)
11.8.4	PCI 总线接口卡电路设计	(265)
11.8.5	PCI 接口卡配置空间初始化	(267)
11.8.6	基于 PCI 总线的接口程序设计	(268)
11.9	PCI 中断	(279)
11.9.1	PCI 中断的特点	(279)
11.9.2	PCI 中断共享	(280)
11.9.3	PCI 中断响应(回答)周期	(281)
11.9.4	PCI 设备的中断申请	(281)

11.9.5	PCI 中断程序举例	(283)
11.10	PCI 总线 DMA 传输	(288)
11.10.1	PCI DMA 传输的特点	(288)
11.10.2	PCI DMA 传输过程	(289)
11.10.3	PCI DMA 控制器	(289)
11.10.4	PCI DMA 的初始化流程	(289)
11.10.5	PCI DMA 程序举例	(289)
习题 11		(295)
第 12 章	微机接口技术中 PCI 设备驱动程序设计	(296)
12.1	为什么要使用设备驱动程序	(296)
12.2	Windows 体系结构下程序的分类	(296)
12.3	Windows 的驱动程序类型	(298)
12.4	驱动程序的主要例程	(298)
12.5	设备驱动程序与用户应用程序有哪些不同	(300)
12.6	编写设备驱动程序需要了解的几个技术问题	(300)
12.7	PCI 接口卡驱动程序设计要求与程序结构	(302)
12.7.1	设计要求	(302)
12.7.2	PCI 接口卡的程序结构	(302)
12.8	访问 PCI 配置空间的驱动程序	(303)
12.8.1	驱动程序对配置空间的访问方法	(303)
12.8.2	访问配置空间的驱动程序设计要点	(303)
12.8.3	从 PCI 卡配置空间读取的信息举例	(304)
12.9	访问 I/O 设备端口的驱动程序	(304)
12.9.1	驱动程序对 I/O 端口的访问方法	(304)
12.9.2	访问 I/O 端口的驱动程序设计要点	(305)
12.9.3	驱动程序中创建一个 I/O 映射实例并进行 I/O 操作过程的程序段	(305)
12.9.4	访问 I/O 设备端口的驱动程序设计举例	(306)
12.10	访问存储器的驱动程序	(314)
12.10.1	Windows 下存储器读写的方法	(314)
12.10.2	访问存储器的驱动程序设计要点	(314)
12.10.3	驱动程序中创建一个存储器映射对象并进行存储器读写操作过程的程序段	(315)
12.10.4	访问存储器的驱动程序设计举例	(316)
12.11	处理中断的驱动程序	(323)
12.11.1	处理 Win32 程序硬件中断的方法	(323)
12.11.2	处理 Win32 程序硬件中断的驱动程序设计要点	(324)
12.11.3	处理 Win32 程序硬件中断的过程	(324)
12.11.4	处理 Win32 程序硬件中断的驱动程序设计	(324)
12.12	处理 MS-DOS 程序硬件中断的驱动程序	(330)

12.12.1	处理 MS-DOS 程序硬件中断的方法	(330)
12.12.2	处理 MS-DOS 程序硬件中断的驱动程序设计要点	(331)
12.12.3	处理 MS-DOS 程序硬件中断的过程	(331)
12.12.4	处理 MS-DOS 程序硬件中断的驱动程序设计	(331)
12.13	处理 DMA 传输的驱动程序	(336)
12.14	Win32 应用程序对驱动程序的调用	(338)
12.14.1	Win32 程序对驱动程序的调用过程	(339)
12.14.2	调用过程中所使用的相关 API 函数与例程	(339)
12.14.3	Win32 应用程序对驱动程序的调用举例	(341)
12.15	MS-DOS 程序对驱动程序的调用	(341)
12.15.1	MS-DOS 程序对驱动程序的调用过程	(341)
12.15.2	虚拟设备驱动程序 VDD 的内容	(342)
12.15.3	MS-DOS 程序调用驱动程序举例	(343)
12.16	采用 DDK 编写驱动程序	(345)
12.16.1	开发工具 DDK	(345)
12.16.2	采用 DDK 编写驱动程序举例	(345)
12.17	驱动程序的开发	(354)
12.17.1	开发驱动程序的工具软件	(354)
12.17.2	驱动程序的安装和调试	(356)
习题 12		(358)
第 13 章	USB 通用串行总线	(359)
13.1	通用串行总线概述	(359)
13.1.1	USB 的发展过程	(359)
13.1.2	USB 的设计目标及特点	(360)
13.1.3	USB 物理接口与电气特性	(361)
13.1.4	USB 信号定义	(362)
13.1.5	USB 数据编码与解码	(363)
13.2	USB 系统组成和拓扑结构	(364)
13.2.1	USB 系统组成	(364)
13.2.2	USB 系统拓扑结构	(367)
13.3	通用串行总线的通信模型与数据流模型	(368)
13.3.1	通信模型	(368)
13.3.2	数据流模型	(369)
13.4	USB 传输类型	(370)
13.5	USB 交换包格式	(371)
13.5.1	标志包	(372)
13.5.2	数据包	(373)
13.5.3	握手包	(374)
13.5.4	预告包	(374)
13.6	USB 设备状态和总线枚举	(374)

13.7	USB 设备设计	(376)
13.8	USB 总线接口芯片 PDIUSB12	(376)
13.8.1	PDIUSB12 外部特性及内部结构	(376)
13.8.2	PDIUSB12 命令字	(379)
13.8.3	PDIUSB12 的典型连接方式	(380)
习题 13		(381)
参考文献		(382)

第 1 章 概 述

在微机系统中,微处理器的强大功能必须通过外部设备(简称外设)才能实现,而外设与微处理器之间的信息交换及通信又是靠接口来实现的,所以,微机应用系统的研究和微机化产品的开发,从硬件角度来讲,就是接口电路的研究和开发,接口技术已成为直接影响微机系统的功能和微机推广应用的关键之一,特别是嵌入式微机应用的基础技术。微机的应用是随着外设不断更新和接口技术的发展而深入到各个领域的。因此,微机接口技术已成为当代理工科大学生必须学习的一种基本知识和科技与工程技术人员必须掌握的基本技能。

现代微机接口技术增加了许多新功能,采用了许多新技术,引入了许多新概念、新名词,需要逐步了解与学习。本书是讨论在以 PCI 为中心的多总线结构和 Windows 系统下的微机接口技术及应用。

1.1 微机接口技术的作用与基本任务

1. 接口在微机系统中的作用

在微机系统中,接口处于微机总线与设备之间,进行 CPU 与设备之间的信息交换。接口在微机系统所处的位置决定了它在 CPU 与设备之间的桥梁与转换作用,接口与其两侧的关系极为密切。因此,接口技术是随 CPU 技术及总线技术的变化而发展的(当然,也与被连接的设备密切相关),尤其与总线的关系密不可分。也就是说,微机系统的总线结构不同,与其相连的接口层次就不同。

2. 接口技术的基本任务

设置接口的目的有两条:通过接口实现设备与总线的连接;连接起来以后,CPU 通过接口对设备进行访问,即操作或控制设备。

因此,接口技术的内容就是围绕设备与总线如何进行连接及 CPU 如何通过接口对设备进行操作展开的。这涉及接口两端的连接对象及通过什么途径去访问设备等一系列的问题。

例如,对设备的连接问题,涉及微机的总线结构是单总线还是多总线;对设备的访问问题,涉及微机的运行环境操作系统是 DOS 还是 Windows 或 LINUX。这些都是接口技术需要进行分析和讨论的内容。

3. 变化中的不变

接口连接设备的任务不会改变。不管是单级总线,还是多级总线;不管是 ISA 总线,还是 PCI 总线;不管是并行方式,还是串行方式;不管是中小规模的接口芯片,还是 VLS 接口芯片;不管是高速设备还是低速设备;不管是标准设备,还是非标准设备。即不分总线结构、数据的宽度、接口电路结构形式、设备的类型,最后都要落实到把外部设备连接到微机系统中去,不会因为上述种种的不同而有所改变,这一点是十分明确的,只是连接的层次、方法、步骤或使用的接口芯片有所不同而已。

同样,通过接口操作设备的任务不会改变。不管是上层应用程序,还是底层驱动程序;不管是 MS-DOS 程序,还是 Win32 程序;不管是实模式程序,还是保护模式程序;不管是 ISA 总

线程序,还是 PCI 总线程序;不管是直接访问程序,还是间接访问程序;不管是汇编语言程序,还是 C 语言程序;不管是中小规模的接口芯片程序,还是 VLS 接口芯片程序。即不分操作系统、处理器的工作方式、总线结构层次、设备资源分配方式、不同的编程语言、接口电路结构形式,最后都要落实到对外部设备的访问,不会因为上述种种的不同而改变,这一点是十分明确的,只是访问的方法、途径、步骤或使用的工具有所不同而已。

1.2 微机接口技术的层次

从早期 PC 微机发展到现代微机,影响接口变化的主要有两大因素。一是总线结构不同,属于硬件上的变化。PC 微机是单总线,只有一级总线,即 ISA 总线;现代微机是多总线,有三级总线,即 Host 总线、PCI 总线、ISA 总线。二是操作系统不同,属于软件上的变化。PC 微机上运行的是 DOS 系统,现代微机上运行的是 Windows 系统,嵌入式微机上一般是运行 LINUX 系统。这种变化使现代接口在完成连接设备和访问设备的任务时产生了根本不同的处理方法,形成了接口分层次的概念,大大促进了接口技术的发展,丰富了接口技术的内容。

接口分层次是微机接口技术在观念上的改变,是接口技术随总线技术的发展而提升的新概念。在考虑设备与 CPU 连接时,不能停留在过去单级总线的接口形态和接口不分层次的传统观念上。

1. 总线结构的改变,使得接口在连接设备时在硬件上要分层次

PC 微机采用单级总线——ISA 总线,设备与 ISA 总线之间只有一层接口。现代微机采用多级总线,总线与总线之间用总线桥连接,例如,PCI 总线与 ISA 总线之间的接口称为 PCI-ISA 桥。因此,除了设备与 ISA 总线之间的那一层接口之外,还有总线与总线的接口——总线桥。在这种情况下,作为连接总线与设备之间的接口就不再是单一层次的,就要分层次了。把设备与 ISA 总线之间的接口,称为设备接口;把 PCI 总线与 ISA 总线之间的接口,称为总线接口。与 PC 微机相比,现代微机的外部设备进入系统需要通过两级接口才行,即通过设备接口和总线接口把设备连接到微机系统。

2. 操作系统的改变,使得通过接口访问设备时在软件上也要分层次

早期微机采用 DOS 操作系统,应用程序享有与 DOS 操作系统同等的特权级,因此,应用程序可以直接访问和使用系统的硬件资源,毫无阻碍。现代微机在使用 Windows 操作系统时,由于保护机制,不允许应用程序直接访问硬件,在应用程序与底层硬件之间增加设备驱动程序,应用程序通过调用驱动程序去访问底层硬件,把设备驱动程序作为应用程序与底层硬件之间的桥梁。因此,访问设备时除了写应用程序之外,还要写设备驱动程序。在 Windows 操作系统下,作为操作与控制设备的接口程序就不再是只有单一的应用程序了,程序也要分层次。把访问设备的 MS-DOS 程序和 Win32 程序称为上层应用程序,把直接操作与控制底层硬件的程序称为核心层驱动程序。与 16 位机相比,现代微机对外部设备的操作与控制需要通过两层程序才行,即通过应用程序和设备驱动程序才能访问设备。

1.3 微机接口技术内容的划分

按照接口分层次的概念,不难把接口技术的内容分为两部分:一部分是接口的上层,包括设备接口及应用程序,构成接口的基本内容;另一部分是接口的下层,包括总线接口及设备驱