



普通高等教育“十一五”国家级规划教材

教育部“高等学校教学质量与教学改革工程”立项项目

傅翠娇 曹庆华 编著

计算机接口与通信 实验教程

计算机科学与技术专业实践系列教材

清华大学出版社



教育“十一五”国家级规划教材

计算机科学与技术专业实践系列教材

教育部“高等学校教学质量与教学改革工程”立项项目

计算机接口与通信 实验教程

傅翠娇 曹庆华 编著

清华大学出版社
北京

内 容 简 介

全书共分为两篇和两个附录,第一篇是基本原理实验,共包括6个基本原理设计实验,它们分别是:可编程并行接口芯片8255A的设计、可编程定时器/计数器芯片8253的设计、可编程中断控制器芯片8259A的设计、固定信号格式的串行接口芯片的设计、D/A转换器的接口电路设计及A/D转换器的接口电路设计;第二篇是创新实验,共包括8个实验,它们分别是基于SOPC的打印机并行接口设计实验、双机通信实验、VGA接口动态游戏实验、PS/2键盘接口实验、液晶显示器LCD接口实验、USB画笔实验、语音录放实验及多功能综合性实验;附录A通过一个SOPC系统开发实例来介绍Nios系统的开发流程;附录B介绍了本书的实验环境以及基本的实验步骤。

本书的特点是既有开发实例的介绍也有基础知识的讲解,力求理论讲解和上机实践相结合,以知识的系统性和实用性为一体,使学习基本原理和掌握设计方法相结合。

本书可以作为高等院校计算机、自动化、电子工程及相关专业“微机接口技术”实验的教材,也可作为从事相关工作的工程技术人员的参考书。

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

计算机接口与通信实验教程/傅翠娇,曹庆华编著. —北京:清华大学出版社,2011.4

(计算机科学与技术专业实践系列教材)

ISBN 978-7-302-24322-9

I. ①计… II. ①傅… ②曹… III. ①电子计算机—接口—高等学校—教材 ②计算机通信—高等学校—教材 IV. ①TP334.7 ②TN919

中国版本图书馆CIP数据核字(2010)第253419号

责任编辑:张瑞庆 李 晔

责任校对:时翠兰

责任印制:何 芊

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

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjjc@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185×260

印 张:12

字 数:283千字

版 次:2011年4月第1版

印 次:2011年4月第1次印刷

印 数:1~3000

定 价:20.00元

产品编号:039180-01

前 言

EDA 技术在电子信息、通信、自动控制及计算机应用等领域的重要性日益突出。随着技术市场与人才市场对 EDA 的需求不断提高,也必然会反映到教学和科研领域中来。传统的微机接口实验教学大部分采用传统专用芯片,在内容和形式上基本是固定的。学生只了解接口芯片外部的特性,不清楚接口芯片的内部工作原理,不能掌握微机接口电路的设计方法。

相对于国内外著名高校的基于 EDA 技术的设计型实验,有些高校的计算机接口与通信实验主要采用基于 ISA 总线的实验台开设原理验证型的实验,采用汇编语言设计硬件驱动程序。但随着计算机技术的发展,ISA 总线已经逐渐退出主流的计算机总线。学生走向社会后,采用汇编语言设计驱动程序的硬件设计方式也不能应用于实际的工程开发中。

研究接口芯片的设计是手段,学会应用接口是目的。采用 EDA 设计一个实际的微机接口芯片及其应用,从程序设计、仿真模拟、系统综合优化,到目标电路代码下载到可编程芯片中,最后通过 Nios II 系统连接到微机总线上编写接口驱动程序进行测试,这是一个综合多方面知识,融会贯通的全过程训练,可提高学生在学习硬件的兴趣和信心,充分发挥学生的创造性和综合运用知识的能力,从而达到从验证型学习到研究型学习的转变,课堂教学从以教师为主体到以学生为主体的转变,这也和研究型大学的发展目标一致。

《接口与通信技术实验》是为了配合《接口与通信技术》、《微机原理与接口技术》等计算机硬件基础的理论教学而开设的,它是继《数字逻辑实验》、《计算机组成原理实验》之后开设的第 3 个层次的实验,是计算机专业的一门重要专业基础课的实验课,其目的在于培养学生分析和设计微机应用系统的基本技能。

本书紧扣计算机接口与通信技术理论知识,共包含了 6 个基本原理实验和 8 个创新实验。基本原理实验主要是接口与通信技术中的基本外围芯片的原理性实验,要求每个学生必须完成。学生通过基本原理实验,对接口芯片的使用掌握之后,就有了进行创新实验的基础。创新实验主要是培养学生综合运用所学知识解决实际问题的能力,这些实验由浅入深,每个实验都给学生留有发挥的余地。基本实验是学生必须完成的实验,有兴趣的学生可以选做创新实验。6 个基本原理实验分别是可编程并行接口芯片 8255A 的设计、固定信号格式的串行接口芯片的设计、可编程定时器/计数器芯片 8253 的设计、可编程中断控制器芯片 8259A 的设计、D/A 转换器的接口电路和 A/D 转换器的接口电路。这些基础实验侧重于基础的接口芯片的设计,用 Verilog 硬件描述语言进行实现。8 个创新实验分别是基于 SOPC 的打印机并行接口设计实验、双机通信实验、VGA 接口动态游戏实验、PS/2 键盘实验、液晶显示器 LCD 接口实验、USB 画笔实验、语音录放实验和多功能综合性实验。其中,基于 SOPC 的打印机并行接口设计实验主要是 8255A 接口芯片的应用实验,双机通信实验是串行接口的应用实验,VGA 接口动态游戏实验是通过 VGA 接口在屏幕上显示动态的游戏实验,PS/2 键盘实验主要是 PS/2 接口的应用实验,液晶显示器 LCD 接口实验主要是 PS/2 接口、LCD 接口的应用实验,USB 画笔实验主要是 USB 接口的简单应用实验,语音录

放实验主要是音频的编码解码、A/D、D/A 接口的应用实验,多功能综合性实验是综合应用了数码管、8255A 并行接口、PS/2 键盘接口、LCD 接口及打印机接口的实验。这些实验都是在基础实验的基础上,通过利用 Quartus II 中的 SOPC Builder 构建 Nios II 系统,然后在 Nios II IDE 中通过用 C/C++ 编写应用驱动程序来实现的。

本书的附录 A 部分介绍了 Nios II 系统的开发过程,读者可以按照该过程来构建实验中所需要的 Nios II 系统。附录 B 部分介绍了本书的实验环境和基本实验步骤,读者可以按照该步骤完成基本原理设计实验的内容。

本书既有开发实例的介绍也有基础知识的介绍,力求理论和实践相结合,以知识的系统性和实用性为一体。希望该书可以帮助读者更好地理解和掌握接口与通信知识,把握计算机硬件体系结构的整体框架,以及在对微机系统进行二次开发方面受到训练。

希望读者在使用的过程中遇到什么问题或者有一些好的建议,可以和编者联系,大家共同探讨和分享。

编者在编写本书的过程中得到了很多教师的建议和帮助,也得到了一些朋友的鼓励。曹庆华老师非常支持本书的编写,并在实验内容的选择和实施方式等方面提出了很多建议;艾明晶老师提供了部分基础实验的源代码,在程序设计方面谢鹏悦老师也给予了很大的帮助,清华大学的许忠信教授及黄宝灵老师在语音录放等实验中给予很多的指导,在此我要特别感谢这几位老师。此外,还要感谢程振东、兰博、李伟欣、董譞、骆骏武、闫碧莹、邹璐、温沙沙、陈文涛、焦烨、朱禹皓、徐艺境、何岸泽、王露露等人,他们在第一、二轮教材使用过程中,就实验内容提出了很多宝贵的建议。

本书内容翔实,结构紧凑,条理清楚。由于编者水平有限,书中难免会有疏漏和不足之处,恳请广大读者提出宝贵意见。

编者
2010 年 8 月

目 录

第一篇 基本原理实验

实验一 可编程并行接口芯片 8255A 的设计	3
一、实验目的	3
二、预习要求	3
三、实验要求	3
四、实验原理	3
五、实验步骤与指导	7
六、实验报告要求	17
实验二 可编程定时器/计数器芯片 8253 的设计	18
一、实验目的	18
二、预习要求	18
三、实验要求	18
四、实验原理	18
五、实验步骤与指导	20
六、实验报告要求	26
实验三 可编程中断控制器芯片 8259A 的设计	27
一、实验目的	27
二、预习要求	27
三、实验要求	27
四、实验原理	27
五、实验步骤与指导	29
六、实验报告要求	46
实验四 固定信号格式的串行接口芯片的设计	47
一、实验目的	47
二、预习要求	47
三、实验要求	47
四、实验原理	47
五、实验步骤与指导	48
六、实验报告要求	53
实验五 D/A 转换器的接口电路设计	54
一、实验目的	54
二、预习要求	54

三、实验要求	54
四、实验原理	54
五、实验步骤与指导	55
六、实验报告要求	60
实验六 A/D 转换器的接口电路设计	61
一、实验目的	61
二、预习要求	61
三、实验要求	61
四、实验原理	61
五、实验步骤与指导	63
六、实验报告要求	65

第二篇 创新实验

实验七 基于 SOPC 的打印机并行接口设计	69
一、实验目的	69
二、预习要求	69
三、实验内容	69
四、实验原理	69
五、实验步骤与指导	72
六、实验报告的要求	79
实验八 双机通信实验	80
一、实验目的	80
二、预习要求	80
三、实验内容	80
四、实验原理	80
五、实验步骤与指导	81
六、实验报告要求	82
实验九 VGA 接口动态游戏实验	83
一、实验目的	83
二、预习要求	83
三、实验内容及要求	83
四、实验原理	83
五、实验步骤与指导	86
六、实验报告要求	94
实验十 PS/2 键盘接口实验	95
一、实验目的	95
二、预习要求	95
三、实验要求	95

四、实验原理	95
五、实验步骤与指导	96
六、实验报告要求	102
实验十一 液晶显示器 LCD 接口实验	103
一、实验目的	103
二、预习要求	103
三、实验要求	103
四、实验原理	103
五、实验步骤与指导	105
六、实验报告要求	110
实验十二 USB 画笔实验	111
一、实验目的	111
二、预习要求	111
三、实验要求	111
四、实验原理	111
五、实验步骤与指导	116
六、实验报告要求	125
实验十三 语音录放实验	126
一、实验目的	126
二、预习要求	126
三、实验要求	126
四、实验原理	126
五、实验步骤与指导	129
六、实验报告要求	145
实验十四 多功能综合性实验	146
一、实验目的	146
二、预习要求	146
三、实验要求	146
四、实验原理	146
五、实验步骤与指导	153
六、实验报告要求	162
附录 A Nios 系统的开发流程介绍	163
附录 B 实验环境与实验的基本步骤	177
参考文献	179

第一篇 基本原理实验

这一篇是基础实验部分,共包括 6 个基本原理设计实验,它们分别是可编程并行接口芯片 8255A 的设计、可编程定时器/计数器芯片 8253 的设计、可编程中断控制器芯片 8259A 的设计、固定信号格式的串行接口芯片的设计、D/A 转换器的接口电路设计及 A/D 转换器的接口电路设计,这些实验都是常见接口芯片的设计实验。

实验一 可编程并行接口芯片 8255A 的设计

一、实验目的

- (1) 掌握用 Intel 8255A 扩展 PC 并行接口的方式。
- (2) 掌握 8255A 可编程并行接口芯片的内部结构、工作原理及工作方式。
- (3) 掌握可编程并行接口芯片的设计和应用方法。

二、预习要求

- (1) 仔细阅读实验指导书并查阅相关资料,了解 8255A 的控制原理,掌握 8255A 控制电路的设计思路和设计方法。
- (2) 分别设计 8255A 控制电路、数据选择电路及数据分配电路,并对各部分电路进行时序仿真。
- (3) 设计顶层图形文件,进行时序仿真。
- (4) 对顶层设计文件进行引脚锁定,再编译。

三、实验要求

- (1) 设计一个可编程并行接口芯片 8255A,使所设计电路适用于 8255A 的 3 种工作方式。
 - ① 整个电路由锁存器、三态缓冲器及控制逻辑 3 部分组成。对于控制逻辑模块,可采用文本输入的方法进行设计,使之根据不同的控制字产生不同的控制信号。
 - ② 再采用顶层图形设计的方法,设计出所要求的可编程并行接口芯片,并进行软件仿真,验证 8255A 3 种方式的基本输入输出功能。
 - ③ 把 8255A 的 A 口、B 口、C 口分别与实验台上的开关和 LED 灯相连,开关模拟数据的输入、LED 灯模拟数据的输出。
- (2) 在 DE2(Development and Education Board)教学开发实验板上进行设计的下载和验证。

四、实验原理

1. 8255A 的特点

1) 8255A 的外部特性

8255A 是一个有 40 个引脚的双列直插组件,其具有面向主机系统总线和面向外设两个方向的连接能力。

面向系统总线的信号线如下:

D0~D7: 三态双向数据总线,是 8255A 与 CPU 数据传送的通道,当 CPU 执行输入输出指令时,通过它实现 8 位数据的读/写操作,控制字和状态信息也通过数据总线传送。

$\overline{CS}$: 片选信号线,当这个输入引脚为低电平时,表示芯片被选中,允许 8255A 与 CPU 进行通信。

$\overline{RD}$: 读信号线,当这个输入引脚为低电平时,允许 8255A 通过数据总线向 CPU 发送数据或状态信息,即 CPU 从 8255A 读取信息或数据。

$\overline{WR}$: 写入信号,当这个输入引脚为低电平时,允许 CPU 将数据或控制字写入 8255A。

A1、A0: 片内端口地址信号线,与系统地址总线低位相连,该信号用来寻址 8255A 内部寄存器。

RESET: 复位输入线,当该输入端处于高电平时,所有内部寄存器(包括控制寄存器)均被清除,所有 I/O 口均被置成输入方式。

面向 I/O 设备的信号线如下:

PA0~PA7: 端口 A 输入输出线,一个 8 位的数据输出输入锁存器/缓冲器。

PB0~PB7: 端口 B 输入输出线,一个 8 位的 I/O 锁存器,一个 8 位输入输出缓冲器。

PC0~PC7: 端口 C 输入输出线,一个 8 位的数据输出锁存器/缓冲器,一个 8 位数据输入缓冲器。

2) 8255A 的内部特性

8255A 主要有 4 个部分。

(1) 数据总线缓冲器: 8 位双向三态,它是 8255A 和 CPU 之间的数据接口。I/O 的数据、CPU 输出的控制字及 CPU 输入的状态信息都通过这个缓冲器传送,数据总线缓冲器的 8 根数据线 D7~D0 一般与 8086 CPU 低 8 位数据线相连。并行 I/O 端口(A 口、B 口、C 口),除 C 口输入没有锁存器外,其余 A 口、B 口输入输出都有缓冲器和锁存器,C 口输出也都有缓冲器和锁存器,C 口输入只有缓冲器。通常 A 口和 B 口作为独立工作的 I/O 数据端口,C 口作为控制或状态信息端口。在方式控制字控制下,C 口可以分成两个 4 位端口,每个端口包含一个 4 位锁存器,分别与 A 口和 B 口配合使用,作为与外设之间的联络信号和存放接口电路当前的状态信息。

(2) 读写控制逻辑: 读写控制逻辑由读信号 $\overline{RD}$,写信号 $\overline{WR}$,选片信号 $\overline{CS}$ 以及端口选择信号 A1、A0 等组成。基本操作如表 1 所示。

(3) 输入输出端口 A、B、C: 8255A 包括 3 个输入输出端口,分别是端口 A、端口 B 和端口 C。每个端口都有一个数据输入寄存器和一个数据输出寄存器,并且数据输入寄存器有缓存功能,数据输出寄存器有锁存功能。

(4) A 组控制逻辑和 B 组控制逻辑: A 组控制逻辑控制端口 A 和端口 C 的高 4 位,B 组控制逻辑控制端口 B 和端口 C 的低 4 位。

2. 8255A 的工作方式

1) 8255A 的 0 方式

0 方式的特点: 0 方式是一种基本的输入输出工作方式。基本功能为: 若 A 口、B 口都工作在方式 0,则此时 8255A 的 A 口、B 口两个 8 位数据口,C 口的上半部和 C 口的下半部两个 4 位数据口,任何一个端口都可以作为输入口和输出口。这种方式下,系统没有指定 C 口的某些线作为专门的信号联络线,但是用户可以自定义 C 口的某些线作为信号联络线。CPU 和 8255A 的 A 口、B 口之间传送数据只能用程控方式(即无条件或查询方式),不能用

中断方式。

表 1 8255A 基本操作

$\overline{CS}$	A1	A0	$\overline{RD}$	$\overline{WR}$	读 操 作	内 容
0	0	0	0	1	PA 口 $\Rightarrow$ 数据总线($\Rightarrow$ CPU)	数据
0	0	1	0	1	PB 口 $\Rightarrow$ 数据总线($\Rightarrow$ CPU)	数据
0	1	0	0	1	PC 口 $\Rightarrow$ 数据总线($\Rightarrow$ CPU)	数据或状态
					写 操 作	
0	0	0	1	0	PA 口 $\Leftarrow$ 数据总线($\Leftarrow$ CPU)	数据
0	0	1	1	0	PB 口 $\Leftarrow$ 数据总线($\Leftarrow$ CPU)	数据
0	0	0	0	1	PA 口 $\Rightarrow$ 数据总线($\Rightarrow$ CPU)	数据
0	0	1	0	1	PB 口 $\Rightarrow$ 数据总线($\Rightarrow$ CPU)	数据
0	1	0	1	0	PC 口 $\Leftarrow$ 数据总线($\Leftarrow$ CPU)	数据
0	1	1	1	0	控制寄存器 $\Leftarrow$ 数据总线	控制字
					无操作情况	
1	$\times$	$\times$	$\times$	$\times$	总线悬浮(三态)	
0	$\times$	$\times$	1	1	总线悬浮	
0	1	1	0	1	控制口不能读	

2) 8255A 的 1 方式

1 方式的特点：1 方式是一种选通的输入输出方式。基本功能：A 口和 B 口可以工作在方式 1 作为数据输入口和输出口。这种方式下，规定了 C 口的某些数据线作为信号联络线，C 口没有被指定作为信号联络线的剩余数据线才可传送数据。1 方式下输入时的功能时序如图 1 所示。

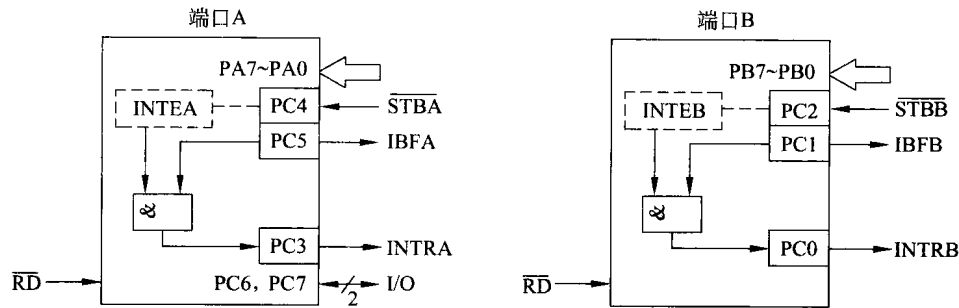


图 1 8255A 1 方式的输入引脚定义

$\overline{STB}$ ：(A 口为 PC4、B 口为 PC2)输入选通信号，低电平有效。当其有效时，将输入设备送来的数据通过 PA7~PA0 或 PB7~PB0 送入 A 口或 B 口。

IBF：(A 口为 PC5、B 口为 PC3)输入缓冲器满，高电平有效，是 8255A 提供给外设的状态信号。当输入设备查询到 IBF 不满即 IBF 为低电平时，输入设备才能送来新的数据，即输入设备发 $\overline{STB}$ 信号有效，当 $\overline{STB}$ 信号有效后，IBF 就被置为高电平，表示输入设备已将数

据输入到 A 口或 B 口,直至 CPU 把数据读走后, $\overline{RD}$ 信号的上升沿使 IBF 变为低电平。

INTR: (A 口为 PC3、B 口为 PC0)中断请求信号,高电平有效,是 8255A 用来向 CPU 提出中断请求的输出信号,只有当 $\overline{STB}$ 、IBF 和 INTE 都为高电平时,INTR 信号才被置为高电平,而 $\overline{RD}$ 信号下降沿使其复位。

INTE: (A 口为 PC4、B 口为 PC2)中断允许,用户通过给 8255A 控制寄存器送 PC4 或 PC2 的置位/复位字来实现允许中断/禁止中断。

1 方式输入的工作时序:以 A 口为例,方式 1 输入过程如下。当输入设备准备好数据,将数据送至 PA7~PA0,同时发送 $\overline{STB}$ 信号,在 $\overline{STB}$ 下降沿控制下,8255A 将 PA7~PA0 上的数据输入到 A 口数据输入寄存器中,同时 8255A 向输入设备发送 IBF 有效,告知输入设备暂缓送数。8255A 可以两种方式通知 CPU 取走数据。第一种方式是用中断方式,在中断允许 INTE=1,IBF 为高电平时, $\overline{STB}$ 的上升沿使 INTR 为高电平,8255A 通过可屏蔽中断控制逻辑 8259A 向 CPU 提出中断申请,在中断服务程序中 CPU 读走数据,一旦读走数据,输入缓冲器满,信号由高电平变为低电平。输入设备仅当检测到 IBF 为低电平后,才开始传送下一个数据,如此循环。第二种方式是用软件查询,CPU 仅当查询到 IBF 为高电平时,才从 8255A 的 A 口取走数据,1 方式输入工作时序如图 2 所示。

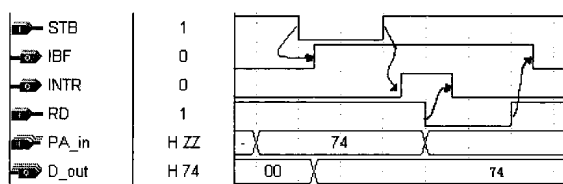


图 2 1 方式输入工作时序

1 方式下输出时的功能时序:以 A 口为例,方式 1 输出过程如下。当输出缓冲器满信号 $\overline{OBF}$ 为高电平时,CPU 执行输出指令,CPU 输出的数据送入 8255A 的 A 口,并使 INTR 复位, $\overline{OBF}$ 置为低电平,通知输出设备 CPU 已把数据输出到了 8255A 的 A 口,输出设备接收到 $\overline{OBF}$ 信号有效后,发送 $\overline{ACK}$ 有效, $\overline{ACK}$ 下降沿将 $\overline{OBF}$ 置为 1, $\overline{ACK}$ 上升沿表示输出设备已从 8255A 指定端口取走数据,此时若 INTE=1,则 INTR 被置为高电平,向 CPU 申请中断,CPU 可采用中断方式输出下一个数据。CPU 也可通过查询 $\overline{OBF}$ 信号,若 $\overline{OBF}$ =1,CPU 输出下一个数据给 8255A,即用查询方式传送数据,1 方式下输出时的功能时序如图 3 所示。

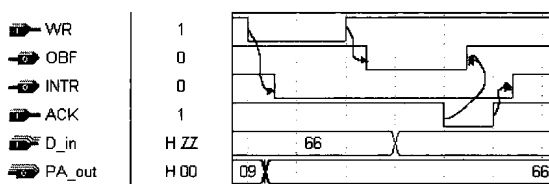


图 3 1 方式输出工作时序

3) 8255A 的 2 方式

2 方式的特点:使用 PA 口作为双向选通的输入输出,一次初始化可以指定 PA 口既作为输入口又作为输出口。设置专用的联络信号线和中断请求线,所以 2 方式下可以采用中断方式和查询方式与 CPU 交换数据。联络线间的时序关系和状态字基本是 1 方式下在输

入和输出两种操作的组合。8255A 2 方式联络信号的定义基本就是 1 方式下的定义,只有中断请求信号共用一组输出信号。

8255A 2 方式的工作时序:2 方式的时序基本上是 1 方式 PA 组输入输出的组合。输入输出的顺序是任意的。输出过程是由 CPU 执行输出指令向 8255A 写数据开始的,而输出是从外设向 8255A 发出选通信号 $\overline{STB}$ 开始的。

五、实验步骤与指导

1. 总体设计思路

对于 8255A 的内部设计将其大致分为 4 个部分:

- (1) 内部逻辑连接部分;
- (2) 0 方式逻辑控制部分;
- (3) 1 方式逻辑控制部分;
- (4) 2 方式逻辑控制部分。

可以把这 4 个部分拿到一个文件中来设计完成。

2. 8255A 控制电路的设计

对于 8255A 的外部设计主要是控制 4 组 8 位双向线的输入输出问题。这个问题将通过外接三态门的方式来解决。在描述内部模块时,将输入端与输出端分别进行描述,并产生输出的使能信号来控制三态门,达到输入输出的目的。不直接使用 BIDIR 类型变量的原因是 BIDIR 类型的信号在顶层仿真时接入到总线的编译中会产生错误。

8255A 控制器 pic8255.v:

输入输出信号定义如下:

```
input CS;                //选片信号
input RD;                //读信号
input WR;                //写信号
input A1,A0;             //片内端口地址信号
input RESET;             //同步清零信号
input[7..0] D_in;        //CPU 向 8255 的数据输入接口
input[7..0] PA_in;       //外设向 8255 A 组数据输入接口
input[7..0] PB_in;       //外设向 8255B 组数据输入接口
input[7..0] PC_in;       //外设向 8255 C 组数据输入接口
output[7..0] D_out;       //8255 向 CPU 的数据输出接口
output[7..0] PA_out;      //8255A 组向外设数据输出接口
output[7..0] PB_out;      //8255B 组向外设数据输出接口
output[7..0] PC_out;      //8255C 组向外设数据输出接口
output d_en;              //D 端口输出使能信号
output pa_en;             //PA 端口输出使能信号
output pb_en;             //PB 端口输出使能信号
output pc0_en, pc1_en, pc2_en, pc3_en, pc4_en, pc5_en, pc6_en, pc7_en;
                           //PC 端口输出使能信号
```

现将其分成以下 6 个部分来实现。

(1) 8255A 可编程并行通信控制接口逻辑设计。8255A 控制的实现主要依靠 6 条控制信号线,以及 1、2 方式下 PC 口的逻辑输出、输入。CS、A1、A0、WR、RD 这 5 条信号线共同组成了一组逻辑功能信号,对应 8255A 的基本操作:

```

wire PA_BUS;           //PA 口输入到 BUS 信号
wire PB_BUS;           //PB 口输入到 BUS 信号
wire PC_BUS;           //PC 口输入到 BUS 信号
wire BUS_PA;           //BUS 输出到 PA 口信号
wire BUS_PB;           //BUS 输出到 PB 口信号
wire BUS_PC;           //BUS 输出到 PC 口信号
wire BUS_CTRL;         //BUS 输出到 CTRL 口信号
wire BUS_Z;            //输出悬浮信号
wire PC_CTRL;          //写状态寄存器信号
wire R;                //0 方式下,读信号

```

根据各读、写控制信号定义赋值如下:

```

//PA 口输入到 BUS 信号
assign PA_BUS= (!CS) & (!A1) & (!A0) & (!RD) & (WR);
//PB 口和 PC 口输入到 BUS 信号
//BUS 输出到 PA 口信号
assign BUS_PA= (!CS) & (!A1) & (!A0) & (RD) & (!WR);
//BUS 输出到 PB 口,PC 口,CTRL 口信号
assign PC_CTRL= (!CS) & (A1) & (RD) & (!WR);           //写控制字信号
assign BUS_Z= ((!CS) & (RD) & (WR)) | CS;             //写悬浮信号
assign R= (!CS) & (!RD) & (WR);                       //读信号

```

(2) 8255A 可编程并行通信数据输入逻辑设计。8255A 有 1 组 CPU 输入数据信号线,3 组外设输入数据信号线。其逻辑如下:当产生送数到外设的信号时,对 D 口输入信号锁存,当 A 口输出时锁存到寄存器 PA_R_OUT;当 B 口输出时锁存到寄存器 PB_R_OUT。

程序框架如下:

```

//0 方式下和 1 方式下送数到 A、B 口
always @ (posedge BUS_PA)           //从 BUS 送数到 A 口
begin
    if (PA_MODE==0 || PA_MODE ==1 )
        if (!PA_IO)    PA_R_OUT<=D_in;
    else if (PA_MODE==2)    PA_R_OUT<=D_in;
end
always @ (posedge BUS_PB)           //从 BUS 送数到 B 口
;

```

采用 assign 赋值语句输出锁存器里的数据:

```

assign PA_out=PA_R_OUT;             //PA 组输出
assign PB_out=PB_R_OUT;             //PB 组输出

```

当外设送数到 8255A 时,0 方式下,当 PA、PB 口处于输入状态时直接锁存;1 方式下,当 PA、PB 口产生“输入选通”信号时对输入数据锁存。

程序框架如下：

```
always @ (posedge stb)
begin
    if(R) //0 方式下读信号有效
    begin
        if(PA_MODE==0 && PB_MODE==0)
        if({A1, A0}==2'B00) //读入 A 口数据
        else if({A1, A0}==2'B01) //读入 B 口数据
        else if({A1, A0}==2'B10) ... //读入 C 口数据
    else
    begin
        if((PA_MODE==1 && PB_MODE==1) || PA_MODE==2)
        if(PA_IO || PA_MODE==2) //读入 A 口数据
        else if(PB_IO && (PA_MODE !=2)) //读入 B 口数据
    wire stb; //D 口输出控制信号
    assign stb=(PA_MODE==1 && PA_IO==1 && (stbacka!=1)) || (PB_MODE==1 && PB_IO==1 &&
    (stbackb!=1)) || (PA_MODE==2 && !stb2) || R;
```

提示：这里设置一中间变量 stb 用来监测读输入数据的请求。

(3) 对 PC 口送数的功能,以及控制命令对 PC 口置位的信号时,都需要对 PC_R_OUT 进行操作,所以写入同一个模块。其逻辑顺序为:当 BUS_CTRL 信号有效时,判断输入高位是否为 1,是,写入工作状态;为 0,则判断最低位是否为 1,为 1,则对所选择的寄存器置数,否则清零。当 BUS_PC 有效时,判断是否为 1 方式,是,则判断 PC 口高,低位的输入输出方式,输出的话则修改寄存器。程序框架如下:

```
always @ (posedge PC_CTRL or posedge RESET)
begin
    if(RESET) //RESET 信号有效,清零
    begin
        PA_MODE<=0;
        PA_IO<=1;
        :
    else //PC_CTRL 信号有效
    begin
        if(A0) //BUS_CTRL 有效, 数据总线写控制寄存器
        if(D_in[7]) //输入最高位为 1,写入方式选择控制字
        :
        else if((D_in[7]==0) && D_in[0]) //对 C 口置位命令
        case(D_in[3:1])
            0: PC_R_OUT[0]<=1;
            :
        else if((D_in[7]==0) && (D_in[0]==0)) //对 C 口复位命令
        case(D_in[3:1])
            0: PC_R_OUT[0]<=0;
            :
    else //BUS_PC 有效,直接送数到 C 口
```