

 高等教育规划教材

基于Proteus 的 计算机系统实验教程

——逻辑、组成原理、体系结构、微机接口

赖晓铮 编著



免费提供电子教案、高清电路图

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS





——逻辑、组成原理、体系结构、微机接口

赖晓铮 编著

机械工业出版社

本书是“数字逻辑”“计算机组成原理”“计算机体系结构”“微机接口”等课程的配套实验教材,系统介绍了数字逻辑基础,计算机的组成原理、体系结构及接口技术。全书分4章,共22个实验。第1章是基础的数字逻辑实验,包括触发器与寄存器、逻辑门与算术电路、组合逻辑电路、比较器和仲裁电路、时序逻辑电路;第2章是计算机组成原理实验,从状态机开始,依次论述了CPU主要组成部件(运算器、存储器、微程序控制器和硬布线控制器)的设计与使用;第3章是计算机体系结构实验,介绍了3种主流的CPU架构(微程序、硬布线、流水线),并且给出了堆栈及CPU嵌套中断的硬件实现方法;最后,第4章介绍了8种常用的微型计算机接口,并且分别与第3章中的微程序CPU一起,搭建一个完整的微型计算机最小系统。

本书内容全面,方法新颖:本书所有实验只涉及基本的数字逻辑器件,不需要学习FPGA及EDA设计的知识;本书所有实验可以在虚拟仿真工具——Proteus上仿真进行,也适用于实验箱教学模式;本书所有实验都是开放式设计,鼓励学生自己动手设计CPU和计算机系统。

本书可作为高等院校计算机、软件及电子信息等专业本科生的实验教材,也可供计算机硬件爱好者、创客及工程技术人员参考使用。

图书在版编目(CIP)数据

基于Proteus的计算机系统实验教程:逻辑、组成原理、体系结构、微机接口/赖晓铮编著. —北京:机械工业出版社,2017.6

高等教育规划教材

ISBN 978-7-111-57043-1

I. ①基… II. ①赖… III. ①计算机系统—高等学校—教材
IV. ①TP3

中国版本图书馆CIP数据核字(2017)第129409号

机械工业出版社(北京市百万庄大街22号 邮政编码 100037)

策划编辑:王海霞 责任编辑:王海霞

责任校对:张艳霞 责任印制:常天培

涿州市京南印刷厂印刷

2017年7月第1版·第1次印刷

184 mm × 260 mm · 18.5印张 · 449千字

0001-3000册

标准书号:ISBN 978-7-111-57043-1

定价:49.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:(010)88379833

机工官网:www.cmpbook.com

读者购书热线:(010)88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面无防伪标均为盗版

金书网:www.golden-book.com

出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才,是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。

2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。

3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。

4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

前言

本书主要为“数字逻辑”“计算机组成原理”“计算机体系结构”“微机接口”等计算机系统类课程的实验和课程设计提供一条不同以往“面板插线”和“可编程逻辑”的实验教学路线——基于 Proteus 的“虚拟仿真”实验教学。

传统的“面板插线”实验教学方式已经推行了几十年，积累了丰富的实验资源。但是受限于动手能力和实验条件，学生仅能通过实验箱面板上插线连接去验证实验箱的功能，不可能去改动实验箱硬件，所以学生缺乏动手实践的机会。而且实验过程的插拔线操作烦琐，实验箱损坏率较高。学生耗费大量时间在线缆连接上，枯燥无味，实验效果较差。

新晋的“可编程逻辑”实验教学方式允许学生运用硬件描述语言在可编程逻辑芯片 FPGA 上进行数字逻辑实验和 CPU 设计，灵活度很高，可以实现达到工业级应用标准的复杂 CPU 架构。但是，学生要有较好的 EDA 技术基础和硬件描述语言编程实践，才能很好地在 FPGA 上进行实验，否则无法理解 FPGA 设计的相关细节。由于专业课程体系的安排，计算机或软件专业的学生很难在“计算机组成原理”“计算机体系结构”基础课之前有足够的时间和精力去深入掌握 FPGA 设计知识。而且，计算机或软件专业亦缺乏对 FPGA 领域熟悉的老师，需要重新培训。

本书力图用简单、直观的方法，使枯燥的计算机系统硬件的基础知识变得直观、形象，让学生方便、快捷地进行数字逻辑、计算机组成原理、计算机体系结构实验，而不是追求“高、深、难、炫”的设计技术，以减轻学生和老师的负担。学生应在掌握了基础的计算机硬件知识之后，在实际需要的时候再学习 FPGA 技术去指导研发工作，而不是本末倒置，为了做课程实验而去学习 FPGA 技术，耗费大量时间和精力在实验工具上。

本书论述的虚拟仿真实验教学方法，首先继承了传统面板插线实验教学的“低门槛”特点，即只要掌握基本的数字逻辑概念，不需要深入学习 FPGA 设计等专业知识，也能在虚拟环境中运用常见的中小逻辑器件“积木式”设计和搭建 CPU；其次，虚拟仿真实验教学具有跟可编程逻辑实验教学一样的“高灵活性”特点，即在 Proteus 虚拟仿真环境中，学生可以从基本数字逻辑器件开始学习，进而验证计算机的组成部件功能，甚至参照主流的 CPU 体系结构，设计相同指令集而不同硬件架构的 CPU。上述“低门槛”和“高灵活性”特点的结合，有利于提降低基础课的实验门槛，提高学生对计算机基础课的学习兴趣，增强学生的创新意识，培养动手实践能力。

本书设计和搭建的 CPU 架构都是“透明”的：硬件可以看见每根导线和每个端口上的电平高低，编程直接采用最底层的机器语言，有利于同学们直观了解 CPU 内部运作机制，牢固掌握所学的知识。而且，本书还挑选了常用的计算机 I/O 外设接口，与微程序 CPU 一起搭建一个完整且最小的微型计算机系统。学生还可以在本书实验的基础上，自己动手移植和搭建硬布线 CPU 或者流水线 CPU 的最小微机系统。

本书的编写得到了 Proteus 软件中国区代理——广州风标信息技术有限公司的大力支持，李垚圣、黄永荣、邓毓峰等同学为本书的编写付出了辛勤的工作，华南理工大学计算机科学与技术专业 2012 级、2013 级、2014 级的同学们对本书提出了大量宝贵的意见，在此对他们表示最诚挚的感谢！

目 录

出版说明

前言

第1章 数字逻辑实验	1
1.1 触发器与寄存器实验	1
1.1.1 实验概述	1
1.1.2 总线通路	1
1.1.3 触发器	3
1.1.4 寄存器	4
1.1.5 实验步骤	6
1.1.6 思考题	7
1.2 逻辑门与算术电路实验	7
1.2.1 实验概述	7
1.2.2 逻辑门	8
1.2.3 算术电路	9
1.2.4 串行进位加法器	9
1.2.5 并行进位加法器	12
1.2.6 实验步骤	13
1.2.7 思考题	14
1.3 组合逻辑电路实验	14
1.3.1 实验概述	14
1.3.2 译码器	15
1.3.3 编码器	16
1.3.4 数据选择器	17
1.3.5 奇偶校验电路	18
1.3.6 实验步骤	18
1.3.7 思考题	19
1.4 数据比较器和仲裁器电路实验	20
1.4.1 实验概述	20
1.4.2 数据比较器	20
1.4.3 仲裁器	22
1.4.4 实验步骤	25
1.4.5 思考题	26
1.5 时序逻辑电路实验	26

录

1.5.1 实验概述	26
1.5.2 计数器原理	26
1.5.3 异/同步计数器	28
1.5.4 加法/减法计数器	30
1.5.5 任意进制计数器	31
1.5.6 电子钟	32
1.5.7 实验步骤	34
1.5.8 思考题	34
第2章 计算机组成原理实验	35
2.1 状态机实验	35
2.1.1 实验概述	35
2.1.2 状态机原理	35
2.1.3 环形计数器	35
2.1.4 扭环计数器	36
2.1.5 状态机示例：交通灯	37
2.1.6 实验步骤	41
2.1.7 思考题	42
2.2 运算器实验	42
2.2.1 实验概述	42
2.2.2 算术逻辑运算器 74LS181	42
2.2.3 串行乘法运算	45
2.2.4 实验步骤	47
2.2.5 思考题	48
2.3 存储器实验	48
2.3.1 实验概述	48
2.3.2 存储器电路	48
2.3.3 ROM 批量导入数据的技巧	52
2.3.4 实验步骤	55
2.3.5 思考题	56
2.4 微程序控制器实验	56
2.4.1 实验概述	56
2.4.2 数据通路	58

2.4.3	微程序原理	59	3.3.5	译码 (D) 阶段及“暂停” 机制	126
2.4.4	微程序控制器	61	3.3.6	执行 (E) 阶段及“气泡” 机制	131
2.4.5	时序发生器	64	3.3.7	写回 (W) 阶段及“旁路” 机制	139
2.4.6	实验步骤	64	3.3.8	中断处理过程及“中断延迟” 机制	142
2.4.7	思考题	67	3.3.9	流水线相关问题	148
2.5	硬布线控制器实验	68	3.3.10	实验步骤	149
2.5.1	实验概述	68	3.3.11	思考题	152
2.5.2	单周期硬布线控制器	68	3.4	嵌套中断 CPU 实验	153
2.5.3	多周期硬布线控制器	72	3.4.1	实验概述	153
2.5.4	实验步骤	75	3.4.2	硬布线堆栈电路	153
2.5.5	思考题	76	3.4.3	基于硬布线堆栈的嵌套中断 CPU	158
第3章	计算机体系结构实验	77	3.4.4	实验步骤	164
3.1	微程序 CPU 实验	77	3.4.5	思考题	165
3.1.1	实验概述	77	第4章	微机接口实验	166
3.1.2	CPU 指令集	77	4.1	I/O 接口扩展实验	166
3.1.3	微程序 CPU 架构	81	4.1.1	实验概述	166
3.1.4	时序电路 (CLOCK)	83	4.1.2	8255A 芯片的结构	166
3.1.5	微程序控制器 (CONTROLLER)	84	4.1.3	8255A 芯片的工作方式	168
3.1.6	取指及中断处理过程	89	4.1.4	“CPU + 8255A”微机系统	170
3.1.7	寄存器及 I/O 操作指令	92	4.1.5	实验步骤	173
3.1.8	存储器及堆栈操作指令	94	4.1.6	思考题	176
3.1.9	跳转系列指令	96	4.2	定时器/计数器实验	176
3.1.10	算术逻辑运算系列指令	97	4.2.1	实验概述	176
3.1.11	实验步骤	101	4.2.2	8253A 芯片的结构	176
3.1.12	思考题	104	4.2.3	8253A 芯片的工作方式	178
3.2	硬布线 CPU 实验	105	4.2.4	“CPU + 8253A”微机系统	182
3.2.1	实验概述	105	4.2.5	实验步骤	184
3.2.2	硬布线 CPU 架构	105	4.2.6	思考题	185
3.2.3	硬布线 CPU 的控制器	107	4.3	串口通信实验	185
3.2.4	硬布线 CPU 的状态机流程图	110	4.3.1	实验概述	185
3.2.5	实验步骤	115	4.3.2	8251A 芯片的结构及功能	186
3.2.6	思考题	117	4.3.3	8251A 芯片的工作方式	190
3.3	流水线 CPU 实验	118	4.3.4	“CPU + 8253A + 8251A”微机 系统	192
3.3.1	实验概述	118			
3.3.2	流水线 CPU 架构	118			
3.3.3	指令流水线及取指 (F) 阶段	122			
3.3.4	数据通路概述	123			

4.3.5 实验步骤	195	4.7.7 “嵌套中断 CPU + 8259A” 微机 系统	226
4.3.6 思考题	197	4.7.8 实验步骤	230
4.4 模 - 数转换实验	197	4.7.9 思考题	233
4.4.1 实验概述	197	4.8 DMA 实验	233
4.4.2 ADC0809 芯片的结构及工作 方式	197	4.8.1 实验概述	233
4.4.3 “CPU + ADC0809” 微机 系统	199	4.8.2 DMA 原理	233
4.4.4 实验步骤	201	4.8.3 8237A 芯片的结构	234
4.4.5 思考题	202	4.8.4 8237A 芯片的内部寄存器	237
4.5 数 - 模转换实验	202	4.8.5 8237A 芯片的命令字和 状态字	240
4.5.1 实验概述	202	4.8.6 8237A 芯片的初始化过程和工作 时序	245
4.5.2 DAC0832 芯片的结构及工作 方式	202	4.8.7 “CPU + 外部存储器 + 8237A” 微机系统	246
4.5.3 实验步骤	206	4.8.8 实验步骤	250
4.5.4 思考题	207	4.8.9 思考题	253
4.6 液晶屏显示实验	207	附录	254
4.6.1 实验概述	207	附录 A Proteus 虚拟仿真软件 简介	254
4.6.2 LCD1602 液晶芯片的结构	207	A.1 Proteus 软件概述	254
4.6.3 8255 芯片的工作方式	210	A.2 电路绘制与仿真技巧	254
4.6.4 “CPU + LCD1602” 微机 系统	212	附录 B 计算机硬件课程综合实验 平台系统	271
4.6.5 实验步骤	214	B.1 实验平台系统简介	271
4.6.6 思考题	216	B.2 实验平台系统操作说明 (交通灯)	273
4.7 中断控制器实验	216	B.3 实验平台系统操作说明 (CPU + 8255A)	277
4.7.1 实验概述	216	B.4 仿真器驱动安装步骤详解	280
4.7.2 8259 芯片的结构	216	B.5 实验箱硬件电路原理图	281
4.7.3 8259A 芯片的工作方式	218		
4.7.4 8259A 芯片的命令字	221		
4.7.5 8259A 芯片的初始化编程	224		
4.7.6 8259A 芯片的中断响应过程	225		

低电平有效的开关 $\overline{NO_BUS}$ (寄存器 74LS374)、 $\overline{DR_BUS}$ (寄存器 74LS273) 和 $\overline{SPT_BUS}$ (移位寄存器 74LS194) 负责控制各个器件输出所保存的数据到系统 BUS。

低电平有效的开关 SET 是 74LS74 的置 1 开关、低电平有效的开关 CLR (触发器 74LS74 和寄存器 74LS175 共用) 和 MR (移位寄存器 74LS194) 是各个器件相应的清零开关。

注意: 寄存器 74LS374 是没有清零功能的, 而寄存器 74LS273 虽有置 0 端 MR, 可以设置清零功能, 但是在本实验中 MR 端接高电平, 实际上是取消了 74LS273 的清零功能。

第 1 章 数字逻辑实验

1.1 触发器与寄存器实验

1.1.1 实验概述

本实验的主要内容是构建一条 8 位总线通路，将拨码开关、数码管及触发器、寄存器等逻辑器件通过单条总线连接起来；通过拨码开关手动输入数据到某个触发器或寄存器；从一个寄存器向另一个寄存器赋值；利用移位寄存器实现数据的置数、左移、右移等功能。通过本实验加深理解总线和触发器的概念，同时熟悉由触发器组成的寄存器和移位寄存器的功能。

1.1.2 总线通路

总线是指为多个器件服务的一组公用信息线，其主要用途是作为多个器件之间进行数据传送的公共通路。如图 1-1 所示，触发器、寄存器堆、8 位总线（BUS）、输入单元（拨码开关）和输出单元（数码管）等不同的器件都挂在同一条总线上。其中，触发器包括 JK 触发器和 D 触发器；寄存器堆则由 4 位寄存器 74LS175、8 位寄存器 74LS374（R0）和 74LS273（DR），以及 8 位移位寄存器 74LS194 构成。除了寄存器 74LS374 自带三态门结构外，其他寄存器的输出都经过三态门 74LS244 和总线 BUS_ [0..7] 相连，以保证在任何时刻总线上都只有唯一的数据存在，避免数据冲突。

在图 1-1 中，左边是上述器件的控制开关，其中，除 CLK、R0_CLK、DR_CLK 和 SFT_CLK 为上升沿控制信号，其余开关均为电平控制信号。

上升沿有效的开关 CLK（触发器 74LS73、74LS74 和寄存器 74LS175 共用）、R0_CLK（寄存器 74LS374）、DR_CLK（寄存器 74LS273）和 SFT_CLK（移位寄存器 74LS194）负责把总线 BUS 上的数据打入各自的器件。

低电平有效的开关 $\overline{R0_BUS}$ （寄存器 74LS374）、 $\overline{DR_BUS}$ （寄存器 74LS273）和 $\overline{SFT_BUS}$ （移位寄存器 74LS194）负责控制各个器件输出所保存的数据到总线 BUS。

低电平有效的开关 $\overline{SET}$ 是 74LS74 的置 1 开关，低电平有效的开关 $\overline{CLR}$ （触发器 74LS73、74LS74 和寄存器 74LS175 共用）和 $\overline{MR}$ （移位寄存器 74LS194）是各个器件相应的清零开关。

注意：寄存器 74LS374 是没有清零功能的，而寄存器 74LS273 虽有置 0 端 $\overline{MR}$ ，可以设置清零功能，但是在本实验中 $\overline{MR}$ 端接高电平，实际上是取消了 74LS273 的清零功能。

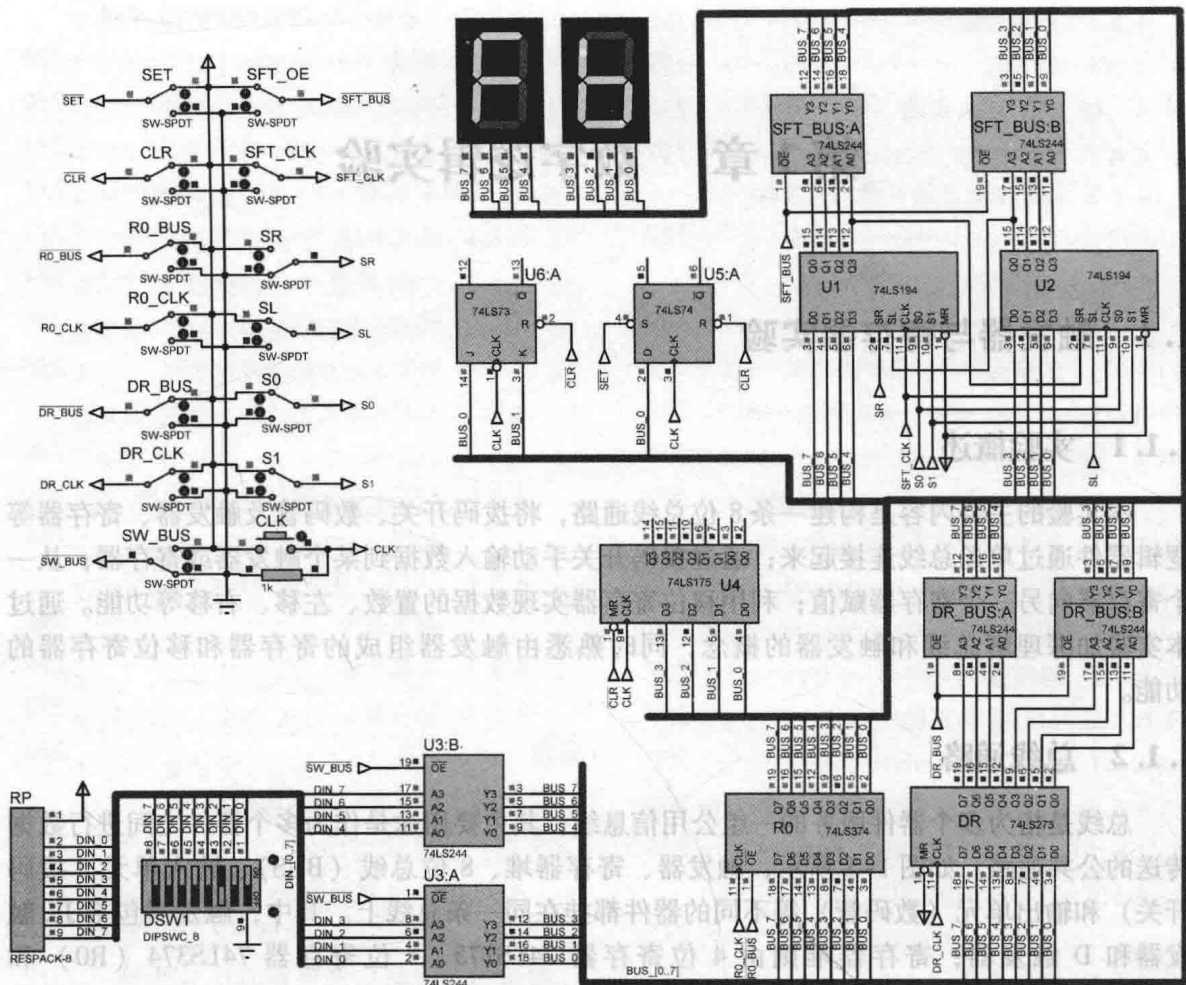


图 1-1 总线通路图

高电平有效的开关 S_L 、 S_R 、 S_0 、 S_1 是移位寄存器 74LS194 的专用开关，负责其置数、移位等功能的设置，详见下文移位寄存器 74LS194 的真值表 1-1。

表 1-1 三态门 74LS244 的真值表

输 入		输 出	
A	$\overline{OE}$	Y	
0	0	0	
1	0	1	
x	1	Z	

注：表中的 x 表示任意选取 0/1 值。后文不一一说明。

总线输入单元如图 1-2 所示。其中，拨码开关 DSW1 和上拉电阻 RP 连接一个 8 位的输入总线 DIN，用来设置输入 DIN 总线的数据；低电平有效的开关 SW_BUS 控制三态门 74LS244（逻辑功能见表 1-1），实现总线 DIN 与总线 BUS 的一对一连通（BUS 总线数据格式见表 1-2）。总线输出单元则是一对数码管，用来显示总线 BUS 上的当前数据。

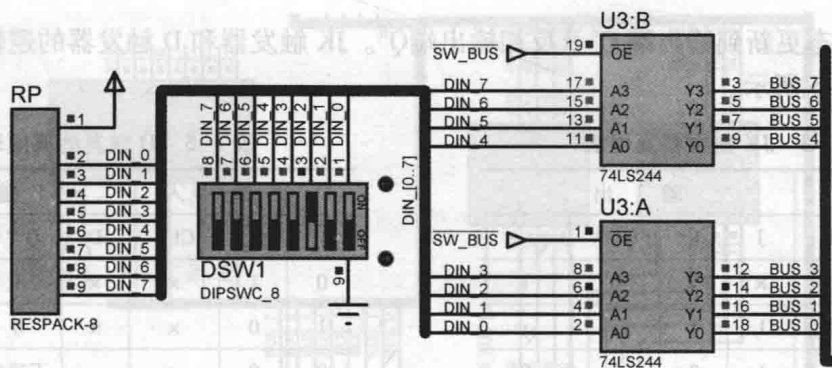


图 1-2 总线的输入单元

表 1-2 BUS 总线数据格式

BUS_7 (最高位)	BUS_6	BUS_5	BUS_4	BUS_3	BUS_2	BUS_1	BUS_0 (最低位)	总线数据 (十六进制)
1	1	1	1	1	0	1	1	0xFB

1.1.3 触发器

触发器是一种具有记忆功能的逻辑器件，它具有两个稳定的状态：“0”和“1”。在适当的触发信号作用下，触发器状态会发生翻转，即触发器可由一个稳态转换到另一个稳态。当输入的触发信号消失后，触发器翻转后的状态保持不变。触发器是构成时序逻辑电路的基础部件，可用作数据的寄存、移位、计数、分频、波形发生等用途。根据电路结构和功能不同，触发器有 RS 触发器、JK 触发器、D 触发器、T 触发器等不同类型，RS 触发器和 T 触发器如表 1-3 所示。本实验使用的是 JK 触发器 74LS73 和 D 触发器 74LS74，如图 1-3 所示。

表 1-3 RS 触发器和 T 触发器真值表

元 器 件	符 号	逻辑功能			
RS 触发器		S	0	0	1
		R	0	1	0
		Q^{n+1}	Q^n	0	1
T 触发器		T	0	0	1
		Q^{n+1}	Q^n	0	1
		Q^{n+1}	Q^n	0	1

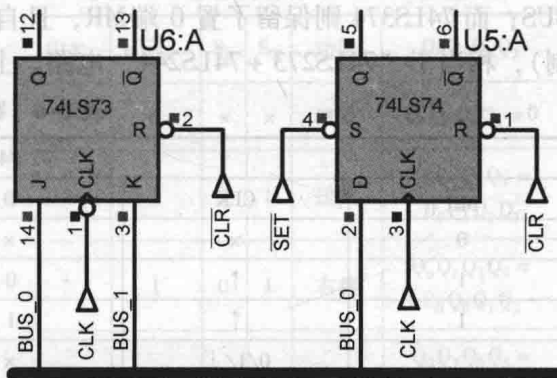


图 1-3 JK 触发器和 D 触发器

JK 触发器是双端输入的边沿触发器，其 CLK 端为触发信号的时钟端， $\overline{R_d}$ 为异步置 0 端。在 CLK 信号下降沿时刻，若输入端 J、K 和 $\overline{R_d}$ 皆为 1，则触发 JK 触发器输出端 Q^n 和 $\overline{Q}^n$ 翻转。D 触发器则是单端输入的边沿触发器，其 CLK 端亦为触发信号的时钟端， $\overline{S_d}$ 和 $\overline{R_d}$ 分别为异步置 1 端和置 0 端。与 JK 触发器不同的是，D 触发器是在 CLK 信号上升沿时刻，将

输入端 D 的状态更新到输出端 Q^n 及反相输出端 $\overline{Q^n}$ 。JK 触发器和 D 触发器的逻辑功能分别如表 1-4 和表 1-5 所示。

表 1-4 JK 触发器真值表

输 入		输 出			
$\overline{R_d}$	CLK	J	K	Q^{n+1}	$\overline{Q^{n+1}}$
0	×	×	×	0	1
1	↓	1	1	Q^n	$\overline{Q^n}$
1	↓	1	0	1	0
1	↓	0	1	0	1
1	↓	0	0	Q^n	$\overline{Q^n}$

表 1-5 D 触发器真值表

输 入				输 出	
$\overline{S_d}$	$\overline{R_d}$	CLK	D	Q^{n+1}	$\overline{Q^{n+1}}$
0	1	×	×	1	0
1	0	×	×	0	1
0	0	×	×	不确定	
1	1	↑	1	1	0
1	1	↑	0	0	1
1	1	↓	×	Q^n	$\overline{Q^n}$

1.1.4 寄存器

作为具有记忆功能的逻辑器件，触发器最重要的功能就是组成寄存器。如图 1-4 所示，本实验使用了三种通用寄存器：74LS175、74LS273 和 74LS374。4 位寄存器 74LS175 相当于 4 个并行的 D 触发器 74LS74 组合，省略了置 1 端，保留了置 0 端 $\overline{MR}$ 用以清零。而 8 位寄存器 74LS273 的逻辑功能则相当于 8 个并联的 D 触发器 74LS74 组合， $D_0 \sim D_7$ 为并行输入端， $Q_0 \sim Q_7$ 为并行输出端（省略了反相输出端 $\overline{Q_x}$ ），CLK 端为时钟脉冲（上升沿触发）；输出端 Q_x 的状态只取决于 CLK 端时钟脉冲到来时刻输入端 D_x 的状态，输出状态的更新发生在 CLK 端脉冲的上升沿。另一方面，8 位寄存器 74LS374 与 74LS273 的逻辑功能基本相同，两者的区别是：74LS273 省略了置 0 端 $\overline{MR}$ ，其输出端必须经过三态门 74LS244 连接到数据总线 BUS；而 74LS374 则保留了置 0 端 $\overline{MR}$ ，且自带三态门输出控制结构（由输出使能端 OE 控制），相当于“74LS273 + 74LS244”电路。上述寄存器的逻辑功能详见表 1-6。

表 1-6 寄存器真值表

74LS175				
$\overline{MR}$	CLK	D	Q^{n+1}	$\overline{Q^{n+1}}$
0	×	×	0	1
1	↑	0	0	1
1	↑	1	1	0
	0/1/↓	×	Q^n	$\overline{Q^n}$
74LS273&74LS374				
$\overline{MR}$	OE	CLK	D	Q^{n+1}
0	0	×	×	0
1	0	↑	0	0
1	0	↑	1	1
1	0	0/1/↓	×	Q^n
×	1	×	×	Z

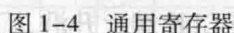
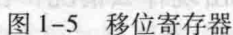


表 1-7 移位寄存器 74LS194 真值表



注：“右移”和“左移”的定义跟 74LS194 的摆放位置有关（图 1-5 中 74LS194 的输出端从左到右是 $Q_0 \rightarrow Q_3$ ）。



1.1.5 实验步骤

实验 1: 总线实验

1) 令 $\overline{R0_BUS} = \overline{DR_BUS} = \overline{SFT_BUS} = \overline{SW_BUS} = 1$, 启动仿真, 手动拨码开关在总线 DIN 上置位数据 0x55, 比较拨码开关所在的总线 DIN 与总线 BUS 上的数据。

2) 令 $\overline{SW_BUS} = 0$, 三态门 74LS244 导通, 记录 BUS 总线的的数据, 与 DIN 总线相比较。

实验 2: 触发器实验

1) 令 $\overline{R0_BUS} = \overline{DR_BUS} = \overline{SFT_BUS} = 1$, $\overline{SW_BUS} = 0$, 启动仿真, 手动拨码开关输入数据到 BUS 总线, 改变 JK 触发器 74LS73 的 J 和 K 端 (即 BUS 总线的 BUS_0 和 BUS_1) 状态, 或者置位, 74LS73 的 $\overline{R_d}$ 端, 观察并记录 CLK 端上升沿、下降沿跳变时刻 74LS73 输出 Q 端和 $\overline{Q}$ 端的状态。

2) 令 $\overline{R0_BUS} = \overline{DR_BUS} = \overline{SFT_BUS} = 1$, $\overline{SW_BUS} = 0$, 启动仿真, 手动拨码开关输入数据到 BUS 总线, 改变 D 触发器 74LS74 的输入 D 端 (即 BUS 总线的 BUS_0) 状态, 或者置位 74LS74 的 $\overline{S_d}$ 端、 $\overline{R_d}$ 端, 观察并记录 CLK 端上升沿、下降沿跳变时刻 74LS74 输出 Q 端和 $\overline{Q}$ 端的状态。

实验 3: 寄存器实验

1) 手动拨码开关输入数据到 BUS 总线, 观察此时寄存器 74LS175 的输入端 D_x 状态, 观察并记录 CLK 上升沿、下降沿跳变时刻, 寄存器 74LS175 输出端 Q_x 和反相输出端 $\overline{Q_x}$ 的状态。观察当 74LS175 的 $\overline{MR}$ 端置 0 后, 寄存器 74LS175 输出端 Q_x 和反相输出端 $\overline{Q_x}$ 的状态。

2) 令 $\overline{R0_BUS} = \overline{DR_BUS} = \overline{SFT_BUS} = 1$, $\overline{SW_BUS} = 0$, 启动仿真, 三态门 74LS244 导通, 手动拨码开关输入数据 0xAA 到总线, 观察此时寄存器 74LS374 和 74LS273 输出端的状态。

3) 令寄存器 R_0 (74LS374) 的 $R0_CLK$ 端上升沿跳变, 把总线上的数据 0xAA 存入 R_0 。

4) 令 $\overline{SW_BUS} = 1$, 三态门 74LS244 阻断, 观察此时总线 BUS 的状态。

5) 令 $\overline{R0_BUS} = 0$, 74LS374 输出选通, 观察总线 BUS 的状态。

6) 令寄存器 D_R (74LS273) 的 DR_CLK 端上升沿跳变, 把总线上的 0xAA 数据存入 D_R , 观察寄存器 74LS273 的输出端。

7) 再令 $\overline{R0_BUS} = 1$, 观察寄存器 74LS374 的输出端, 比较寄存器 74LS175、74LS273 和 74LS374 的异同。

8) 手动拨码开关输入新的数据 0x55 到总线 BUS ($\overline{SW_BUS} = 0$)。此时, 新的数据会冲掉 R_0 寄存器保存的原有数据 0xAA 么? 若再令 $\overline{R0_BUS} = 0$, 会出现什么情况?

9) 假设手动拨码开关分别打入数据 0xAA 和 0x55 到 R_0 寄存器 (74LS374) 和 D_R 寄存器 (74LS273), 并且同时令 $\overline{R0_BUS} = 0$ 和 $\overline{DR_BUS} = 0$, 会出现什么情况? 在总线上可以同时选择多个寄存器输出数据 (导通输出端的三态门) 么?

实验 4：移位寄存器实验

令 $\overline{R0_BUS} = \overline{DR_BUS} = \overline{SFT_BUS} = 1$ ， $\overline{SW_BUS} = 0$ ，启动仿真，通过拨码开关输入总线 BUS 任意 8 位二进制数，赋值 74LS194 的输入端 $D_0D_1D_2D_3$ 。按照表 1-8 置位 74LS194 的 MR 、 S_L 、 S_0 、 S_L 和 S_R 端，观察并记录 CLK 端上升沿、下降沿跳变时刻输出端 $Q_0Q_1Q_2Q_3$ 的状态，在表 1-8 中填写观测结果和功能总结。

表 1-8 移位寄存器实验记录表

清除	模式		时钟	输入	串 行		输 出	串 行		输 出
MR	S_L	S_0	CLK	$D_0D_1D_2D_3$	S_L	S_R	$Q_0Q_1Q_2Q_3$	S_L	S_R	$Q_0Q_1Q_2Q_3$
1	0	0	↑		×	×		×	×	
1	1	1	↑		×	×		×	×	
1	0	1	↑		0	0		1	0	
1	0	1	↑		0	1		1	1	
1	1	0	↑		0	0		1	0	
1	1	0	↑		0	1		1	1	

1.1.6 思考题

- 1. 为什么常见的 CPU 都是 8 位、16 位或 32 位总线？可以使用 7 位或 10 位的总线么？计算机总线的位数是由什么决定的？32 位 CPU 是否一定比 8 位 CPU 的处理能力强？
- 2. 表 1-3 中所列的触发器可以互相转化，请以 D 触发器 74LS74 为基础构造电路实现 JK 触发器 74LS73 的功能，并分别以 JK 触发器 74LS73 和 D 触发器 74LS74 为基础构造电路实现 T 触发器的功能。
- 3. 在总线通路中，寄存器 74LS273 输出端能否直连总线 BUS？若不能，请说明原因。
- 4. 移位寄存器 74LS194 的 S_L 端和 S_R 端是提供 $D_0D_1D_2D_3$ 端移入数据还是保存 $D_0D_1D_2D_3$ 端移出数据？假设需要保存 $D_0D_1D_2D_3$ 端移出的数据，怎么修改 74LS194 电路？
- 5. 请使用移位寄存器 74LS194 实现对 4 位二进制数进行“ $\times 2$ ”乘法操作和“ $\div 2$ ”除法操作，并且给出适用上述操作的二进制数的表示范围（注意：4 位二进制数可能是无符号数，也可能是有符号数，两者的表示范围是不同的）。

1.2 逻辑门与算术电路实验

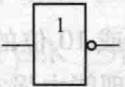
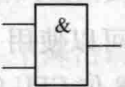
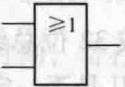
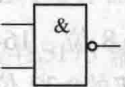
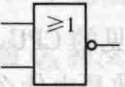
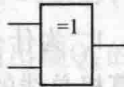
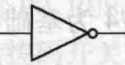
1.2.1 实验概述

本实验的主要内容是熟悉常用逻辑门的概念和功能；了解半加器和全加器电路的结构；理解补码的原理；通过逻辑门电路搭建 4 位串行进位加法器和并行进位加法器的算术电路，并且比较两种进位加法器结构的异同。

1.2.2 逻辑门

逻辑门是构成组合逻辑电路的基础部件，具有一个或多个输入端和唯一的输出。与触发器不同，逻辑门是一种无记忆的静态开关电路：任何时刻输出端的信号仅取决于该时刻输入端的信号组合，而与输入/输出端原有的状态无关。此处所指“信号”都是布尔代数的二值逻辑信号（只有两种离散状态“0”和“1”）。常用逻辑门见表1-9。

表 1-9 常用逻辑门

输 入		输 出					
A	B	逻辑非	逻辑与	逻辑或	逻辑与非	逻辑或非	逻辑异或
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0
表达式		$\overline{A}$	$A \cdot B$	$A + B$	$\overline{A \cdot B}$	$\overline{A + B}$	$A \oplus B$
国标符号							
仿真符号							

上述逻辑门电路如图1-6所示，最左边一列是最基础的逻辑门：与门、或门和非门，图中所有其他的逻辑门都可以由这三种逻辑门组合形成。例如，第2列的与非门等效于与门和非门串联，第3列的或非门等效于或门和非门串联。较为复杂的是最右边的异或门（输入相异则输出1，输入相同则输出0）74LS86及其等效逻辑电路（如图1-6中最右边方框内）。

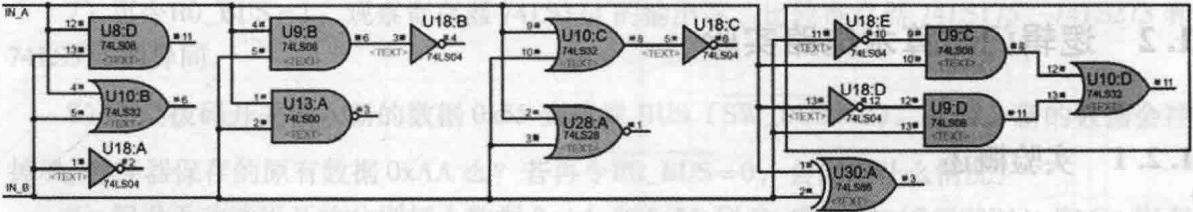


图 1-6 逻辑运算电路图

1.2.3 算术电路

逻辑门除了可以实现逻辑运算以外，还可以组合电路形式实现算术运算，例如实现两个二进制数的加法：图 1-7 所示为最基本的一位二进制数加法电路——半加器和全加器（FA）。两者之间的区别在于半加器不考虑低位的进位，只考虑两个二进制数相加的结果 S 和向高位的进位 C；而全加器则是在半加器的基础上考虑了来自低位的进位信号 C_{i-1} 。

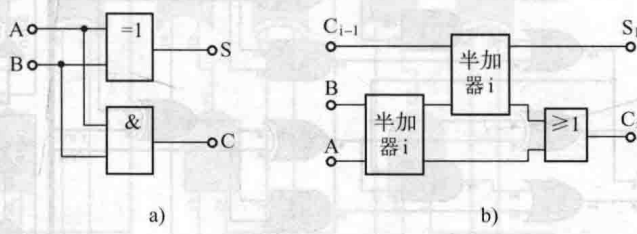


图 1-7 一位半加器和全加器电路示意图

半加器的逻辑表达式： $S_i = A_i \oplus B_i$ 且 $C_{i+1} = A_i \cdot B_i$

全加器的逻辑表达式： $S_i = A_i \oplus B_i \oplus C_i$ 且 $C_{i+1} = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_i$

本实验的算术电路如图 1-8 所示：在 BUS 总线下方是图 1-6 所示的逻辑门电路。而 BUS 总线左边是输入单元，由 SW_BUS 控制拨码开关向 BUS 总线输入数据。BUS 总线上方则是由逻辑门组成的 4 位加法电路：串行进位加法器和并行进位加法器。总线 BUS 同时为两个加法器提供相同的输入端 $BUS_ [0..3]$ 和 $BUS_ [4..7]$ ，运算结果则显示在输出单元的数码管上（左侧数码管显示并行进位加法器结果，而右侧数码管显示串行进位加法器结果）。

1.2.4 串行进位加法器

串行进位加法器（又称为行波进位加法器）是由若干位全加器串行级联组成的多位二进制数加法电路，如图 1-9 所示。图 1-9b 所示是每一位加法器电路，即全加器（FA）电路。由 4 个相同的 FA 电路串联（低位 FA 的进位输出 C 直接与相邻高位 FA 的进位输入 C_{i-1} 相连），构成了图 1-9a 所示的 4 位串行进位加法器电路。其中最高位是符号位，加法器的位间进位从低位往高位逐位串行传送，最高数值位向符号位的进位和符号位本身的进位通过异或得到溢出判断位 OF（LED 显示）。

如图 1-10a 所示，计算机所能处理的数据是无限数轴的一部分，其范围与位数有关。图中，4 位二进制数的最高位是符号位，其余 3 位是数值位；其数值表示范围是 $-8 \sim 7$ ，并且构成一个闭合圆环（圆环内圈是数值的真值，圆环外圈是真值对应的补码）。当计算机的数值超过最大值 +7（顺时针越过边界）时，即正溢出，变为最小值；反之，当数值小于最小值 -8 （逆时针越过边界）时，即负溢出，变为最大值。