

新世纪电子信息课程系列规划教材

数字系统设计综合实验教程

LUZI XITONG SHEJI ZONGHE
MIYAN JIAOCHENG

主 编 李桂林



東南大學出版社
SOUTHEAST UNIVERSITY PRESS

新世纪电子信息课程系列规划教材

数字系统设计综合实验教程

主 编 李桂林

副主编 张彩荣 刘丽君
刘天飞

东南大学出版社

• 南京 •

内 容 提 要

本书内容包括绪论、实验内容、实验常见问题及解答、实验软件开发系统(Max+plusII 和 Quartus II)、实验硬件开发系统(Aquila-M250)使用介绍等。在实验内容安排上,按照实验的难易度及不同的训练目标,将其分成基础实验、综合设计实验、课程设计实验三个层次,共 27 个实验,内容由易到难,由浅入深。每部分内容都按实验目的、实验原理、实验内容及步骤、设计示例、实验报告要求、实验思考题的顺序编撰。在前两类实验的设计示例里,提供详细的参考设计原理图、Verilog HDL 程序及仿真波形图;在课程设计实验里提供设计思路和原理图,以培养学生的独立思考能力和充分发挥学生的创造性。实验常见问题及解答是基于软件 Max+plusII 和 QuartusII 以及硬件编程语言 Verilog HDL、实验箱的使用而总结归纳出来的。

书中的实验是编者经过反复的实验及实践积累提炼而成,可作为高等学校电类与非电类学生的实验教材,也可作为广大电子设计爱好者及工程技术人员的参考资料。

图书在版编目(CIP)数据

数字系统设计综合实验教程/李桂林主编. —南京:
东南大学出版社,2011.3
新世纪电子信息课程系列规划教材
ISBN 978-7-5641-2655-1

I. ①数… II. ①李… III. ①数字系统—系统设计—
实验—高等学校—教材 IV. ①TP271-33

中国版本图书馆 CIP 数据核字(2011)第 023214 号

数字系统设计综合实验教程

出版发行 东南大学出版社

出 版 人 江建中

社 址 南京市四牌楼 2 号

邮 编 210096

经 销 全国各地新华书店

印 刷 南京京新印刷厂

开 本 787 mm×1092 mm 1/16

印 张 10.75

字 数 268 千字

书 号 ISBN 978-7-5641-2655-1

印 次 2011 年 3 月第 1 次印刷

版 次 2011 年 3 月第 1 版

定 价 24.00 元

(凡有印装质量问题,请与我社读者服务部联系。电话:025—83792328)

前 言

《数字系统设计》是电类的专业基础课,大规模可编程逻辑器件 FPGA/CPLD 和 EDA 技术的迅猛发展及广泛应用,对《数字系统设计》课程的教学也提出了更高的要求。

《数字系统设计》是一门实践性很强的课程,实验在其教学内容中占据非常重要的地位。我们编写本书的目的是既让学生学习数字系统基本理论又培养其利用软件开发硬件的基本技能,帮助学生在实践中进一步理解书本知识,提高分析问题、解决问题以及实践应用的能力,从而为学习其他专业课程以及参加全国电子设计大赛、全国机器人大赛,为论文设计以及从事电子产品研发等相关工作打下必要的基础。

本书包括绪论、基础实验、综合设计实验、课程设计实验、实验常见问题及解答、实验软件开发系统、实验硬件开发系统 7 个部分。其中,基础实验、综合设计实验、课程设计实验共 27 个,它们由易到难,由浅入深,构成了 3 个不同层次的实验体系,呈现了实验的难易度及不同的训练目标。本教材的每个实验都按实验目的、实验原理、实验内容及步骤、设计示例、实验报告要求、实验思考题的顺序进行编写。在基础实验和综合设计实验的设计示例里,提供详细的参考设计原理图、Verilog HDL 程序及仿真验证波形图;在课程设计实验里提供设计思路和原理图,以培养学生的独立思考能力,充分发挥学生的创造性。实验常见问题及解答是基于软件 Max+plusII、QuartusII 和硬件编程语言 Verilog HDL 以及实验箱的使用而总结归纳出来的。实验软件系统详细地介绍了目前流行的两种开发软件 Max+plusII 和 QuartusII,以具体的例子介绍了入门设计及设计提高。实验硬件系统介绍了目前较为流行的 Aquila-M250 型 FPGA 实验箱和 FPGA10K 开发箱。

本书由李桂林主编。第 1~第 4 章由李桂林、刘丽君编写,第 5 章、第 7.2 节由刘天飞编写,第 6 章、第 7.1 节由张彩荣编写。

由于编者水平有限,错漏之处在所难免,恳请广大读者指正。

编 者

2011 年 1 月

目 录

1	绪论	(1)
1.1	引言	(1)
1.2	EDA 技术简介	(1)
1.3	数字系统设计综述	(2)
1.3.1	数字系统的基本概念	(2)
1.3.2	数字系统的设计方法	(2)
1.3.3	数字系统的实现方式	(3)
1.4	硬件描述语言简介	(4)
1.5	FPGA/CPLD 综述	(5)
1.5.1	FPGA/CPLD 简介	(5)
1.5.2	基于 FPGA/CPLD 的数字系统开发流程	(6)
1.5.3	FPGA/CPLD 主要厂商及产品	(8)
1.5.4	FPGA/CPLD 的 EDA 开发工具	(9)
1.6	数字系统设计实验说明	(9)
1.6.1	实验规则	(9)
1.6.2	实验报告的撰写	(10)
2	基础实验	(11)
2.1	基本组合逻辑电路设计实验	(11)
实验 1	加法器设计	(11)
实验 2	编码器设计	(14)
实验 3	译码器设计	(17)
实验 4	数据选择器设计	(20)
实验 5	数值比较器设计	(22)
实验 6	三态门设计	(24)
2.2	基本时序逻辑电路设计实验	(25)
实验 7	触发器设计	(25)
实验 8	寄存器和锁存器设计	(27)
实验 9	移位寄存器设计	(29)
实验 10	计数器设计	(31)
2.3	基本设计方法训练实验	(32)
实验 11	阻塞与非阻塞语句区别验证	(32)
实验 12	有限状态机设计	(34)
3	综合设计实验	(37)
实验 1	累加器设计	(37)
实验 2	数码管扫描显示电路设计	(39)

实验 3	数字频率计设计	(42)
实验 4	步进电机控制电路设计	(45)
实验 5	交通灯控制器设计	(49)
实验 6	电子数字钟设计	(56)
实验 7	汽车尾灯控制电路设计	(60)
实验 8	按键消抖设计	(64)
实验 9	可编程单次脉冲发生器设计	(67)
实验 10	趣味实验——蛇形电路设计	(74)
4	课程设计实验	(80)
实验 1	数字密码锁设计	(80)
实验 2	简易计算器设计	(81)
实验 3	波形发生器设计	(82)
实验 4	数据采集与监测系统设计	(84)
实验 5	简易 CPU 设计	(85)
5	实验常见问题及解答	(87)
6	实验软件开发系统	(91)
6.1	Max+plusII 开发系统	(91)
6.1.1	Max+plusII 简介	(91)
6.1.2	Max+plusII 工作环境介绍	(91)
6.1.3	Max+plusII 设计入门	(95)
6.1.4	Max+plusII 设计提高	(107)
6.2	QuartusII 开发系统介绍	(115)
6.2.1	QuartusII 简介	(115)
6.2.2	QuartusII 工作环境介绍	(115)
6.2.3	QuartusII 设计入门	(120)
6.2.4	QuartusII 基于宏功能模块的设计	(136)
7	实验硬件开发系统	(142)
7.1	FLEX10K FPGA 实验系统	(142)
7.1.1	实验箱外观	(142)
7.1.2	部件及使用说明	(142)
7.1.3	实验设备的连接及开发过程	(152)
7.1.4	实验箱的功能测试	(153)
7.1.5	FLEX10K 系列器件简介	(153)
7.2	Aquila-M250 型 FPGA 实验系统	(154)
7.2.1	实验箱外观	(154)
7.2.2	部件及使用说明	(154)
参考文献		(163)



绪 论

1.1 引言

计算机技术和微电子工艺的发展使得现代数字系统的设计和应用进入了新的阶段。传统的设计方法已逐步被基于 EDA 技术的芯片设计技术取代。

对大部分学过数字电路设计的人而言,他们的学习过程大都是从基本的组合逻辑开始,再顺序逻辑、简单的模块设计、至复杂完整的系统设计。传统的实验方式,每做一个实验就必须重组一个硬件线路,特别是复杂的线路,相当耗时且不易进行。大规模可编程器件:复杂可编程逻辑器件 CPLD(Complex Programmable Logic Device)和现场可编程门阵列 FPGA(Field Programmable Gate Array)的出现和广泛使用,改进了以往数字电路的学习方式,并且缩短了开发大型数字电路的时间。FPGA/CPLD 具有性能好、可靠性高、容量大、体积小、低功耗、速度快、使用灵活、设计周期短、开发成本低、静态可重复编程、动态在系统重构、硬件功能可以像软件一样通过编程来修改,极大地提高了电子系统设计的灵活性和通用性。利用它,以个人计算机(PC)为平台、借助 EDA 软件,可以达到电路设计输入、仿真、下载验证、修改、编程下载一气呵成,不仅让数字电路的学习效率变得更高,而且让自行设计开发逻辑芯片的梦想得以实现。

在现代数字系统设计中,使用硬件描述语言进行电路设计已成为一个显著的特征,目前使用最广泛的是 Verilog HDL 语言和 VHDL 语言。Verilog HDL 提供了非常精炼和易读的语法,其功能强大,许多大规模的电路设计都是用 Verilog HDL 来完成的。

目前,学会数字系统设计技术,使用大规模、超大规模可编程逻辑器件,掌握现代 EDA、硬件描述语言已成为从事电子设计人员必须具备的基本能力。

1.2 EDA 技术简介

电子设计自动化(Electronic Design Automation, EDA)技术是以大规模可编程逻辑器件为设计载体,以硬件描述语言为系统逻辑描述的主要表达方式,以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具,通过有关的开发软件,自动完成用软件的方法设计从电子系统到硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真,直至对特定目标芯片的适配编译、逻辑映射、编程下载等工作,最终形成集成电子系统或专用集成芯片的一门新技术。

EDA 技术伴随着计算机、集成电路、电子设计的发展经历了计算机辅助设计(Computer Aided Design, CAD)、计算机辅助工程(Computer Aided Engineering, CAE)、EDA 三个发展阶段。现代 EDA 技术涉及面广,内容丰富,从教学和实用的角度看,主要包括:大规模可编程逻辑器件、硬件描述语言、软件开发工具、实验开发系统四个方面。其中,大规模可编程逻辑器件是利用 EDA 技术进行电子系统设计的载体;硬件描述语言是利用 EDA 技术进行电子系统设计的主要表达手段;软件开发工具是利用 EDA 技术进行电子设计的智能化、自动化设计工具;实验开发系统是利用 EDA 技术进行电子系统设计的下载工具及硬件验证工具。

利用 EDA 技术进行数字系统设计,具有以下特点:

(1) 全自动化:用软件方式设计的系统到硬件系统的转换,是由有关的开发软件自动完成的。

(2) 开放性和标准化:现代 EDA 工具普遍采用标准化和开放性框架结构,任何一个 EDA 系统只要建立了一个符合标准的开放式框架结构,就可以接纳其他厂商的 EDA 工具一起进行设计工作。这样就可以实现各种 EDA 工具间的优化组合,并集成在一个易于管理的统一环境之下,实现资源共享。

(3) 操作智能化:可以使设计人员不必深入学习许多的专业知识,还可以免除许多推导运算即可获得优化的设计成果。

(4) 执行并行化:由于多种工具采用了统一的数据库,使得一个软件的执行结果马上可被另一个软件使用,使得原来要串行的设计步骤变成了并行过程,因此也称为“同时工程(Concurrent Engineering)”。

(5) 成果规范化:采用硬件描述语言可以支持从数字系统到门级的多层次的硬件描述。

1.3 数字系统设计综述

1.3.1 数字系统的基本概念

数字系统是指对数字信息进行存储、传输、处理的电子系统,其输入和输出都是数字量。数字系统通常可分为输入/输出接口、数据处理器、控制器三个部分。图 1.3.1 所示为简单的数字系统结构框图。

其中,输入/输出接口是完成将物理量转化为数字量或将数字量转化为物理量的功能部件。

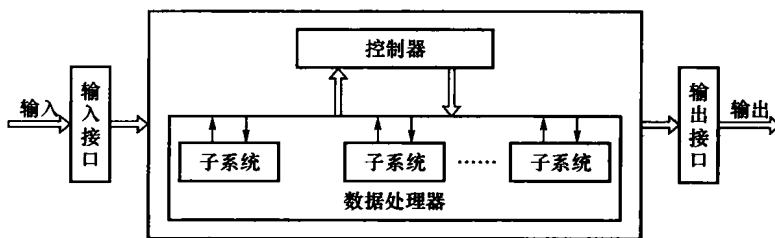


图 1.3.1 数字系统结构框图

数据处理器用于完成数据的采集、存储、运算和传输。数据处理子系统主要由存储器、运算器、数据选择器等功能电路组成。数据处理器与外界进行数据交换,在控制器发出的控制信号的作用下,数据处理器将进行数据的存储和运算等操作。

控制器是执行数字系统算法的核心,具有记忆功能,因此控制器为时序系统。控制器的输入信号是外部控制信号和由数据处理器送来的条件信号,按照数字系统设计方案要求的算法流程,在时钟信号的控制下进行状态的转换,同时产生与状态和条件信号相对应的输出信号。

1.3.2 数字系统的设计方法

EDA 技术的出现使数字系统的设计方法发生了根本的改变。在 EDA 设计中往往采用层次化设计方法,分模块、分层次地进行设计描述。描述系统总功能的设计为顶层设计,描述系统

中较小单元的设计为底层设计。整个设计过程可理解为从硬件的顶层抽象描述向底层结构描述的一系列转换过程,直到最后得到可实现的硬件单元描述为止。层次化设计方法比较灵活,既可采用自顶向下(Top-Down)设计也可采用自底向上(Bottom-Up)设计。

1) 自底向上(Bottom-Up)设计方法

Bottom-Up 设计方法的中心思想是:首先根据对整个系统的测试与分析,由各个功能模块连成一个完整的系统,由逻辑单元组成各个独立的功能模块,由基本门构成各个组合与时序逻辑单元。

Bottom-Up 设计方法有以下几个特点:

- (1) 从底层逻辑库中直接调用逻辑门单元;
- (2) 符合硬件工程师传统的设计习惯;
- (3) 在进行底层设计时缺乏对整个电子系统总体性能的把握;
- (4) 在整个系统完成后,要进行修改较为困难,设计周期较长;
- (5) 随着设计规模与系统复杂度的提高,这种方法的缺点更为突出。

传统的数字系统的设计方法一般都是自底向上的,即首先确定构成系统的最底层的电路模块元件的结构和功能,然后根据主系统的功能要求,将它们组成更大的功能模块,使它们的结构和功能满足高层系统的要求,依此类推,直至完成整个目标系统的 EDA 设计。

2) 自顶向下(Top-Down)设计方法

Top-Down 设计方法的中心思想是:系统层是一个包含输入和输出的顶层模块,并用系统级行为描述加以表达,同时完成整个系统的模块和性能分析;整个系统进一步由各个功能模块组成,每个模块由更细化的行为描述加以表达;由 EDA 综合工具完成到工艺库的映射。

Top-Down 设计方法有以下几个特点:

- (1) 结合模块手段,可以从开始就掌握目标系统的性能状况;
- (2) 随着设计层次向下进行,系统的性能参数将进一步得到细化和确认;
- (3) 可以根据需要及时调整相关的参数,从而保证了设计结果的正确性,缩短了设计周期;
- (4) 当规模越大时,这种方法的优越性越明显。

现代数字系统的设计方法一般都是自顶向下的层次化设计方法,即从整个系统的整体要求出发,自上而下地逐步将系统设计内容细化,把整个系统分割为若干功能模块,最后完成整个系统的设计。系统设计自顶向下大致可以分为三个层次:

(1) 系统层:运用概念、数学知识和框图进行推理和论证,形成总体方案。

(2) 电路层:进行电路分析、设计、仿真和优化,把框图与实际的约束条件以及可测性条件相结合,实行测试和模拟(仿真)相结合的科学实验研究方法,制作直到门级的电路图。

(3) 物理层:真正实现电路的工具。同一电路可以有多种不同的方法来实现。物理层包括 PCB、IC、PLD 或 FPGA 和混合电路集成以及微组装电路的设计等。

在 Top-Down 设计方法中必须经过“设计—验证—修改设计—再验证”的过程,多次反复,直至得到的结果能够完全实现所要求的逻辑功能,并在速度、功耗、价格和可靠性方面实现较为合理的平衡。

1.3.3 数字系统的实现方式

随着集成电路技术和计算机技术的发展,数字系统的实现方法经历了分立元件、SSI(小规模数字集成电路)、MSI(中规模数字集成电路)、LSI(大规模数字集成电路)、VLSI(超大规模数字集成电路)的进程。SSI 时代直接以集成门、触发器为基本器件构成系统。MSI 时代用 MSI

器件如:计数器、译码器、数据选择器等功能电路作为模块构成系统,它们一般是指一些通用集成电路。然而,一个复杂的数字系统往往需要许多 SSI、MSI 器件才能实现,因而支撑的设备体积大、功耗大、成本高,更重要的是用 SSI、MSI 器件做成的设备可靠性差,因而用标准产品实现数字系统的方法现在已很少使用。

在集成技术高度发展的今天,数字系统的实现主要有两类器件:一类是可编程逻辑器件(PLD),另一类是专用集成电路(ASIC),它们都属于大规模或超大规模器件,且有各自的优点。

1) 可编程逻辑器件(PLD)

PLD 主要包括 FPGA 和 CPLD,是一种半定制的器件,器件内已做好各种逻辑资源,用户只需对器件内的资源进行编程连接就可实现所需要的功能,而且可以反复修改,反复编程,直到满足设计要求。用 PLD 实现设计直接面向用户,具有其他方法无可比拟的方便性、灵活性和通用性,且硬件测试和实现快捷、开发效率高、成本低、风险小。现代 FPGA 器件集成度不断提高,等效门数已达到千万门级,在器件中,除集成各种逻辑门和寄存器外,还集成了嵌入式块 RAM、硬件乘法器、锁相环、DSP 块等功能模块,使 FPGA 的使用更为方便。EDA 开发软件对 PLD 器件也提供了强有力的支持,其功能更全面,兼容性更强。

2) 专用集成电路(Application Specific Integrated Circuit, ASIC)

ASIC 是指用全定制方法来实现设计的方式,它在最底层,即物理版图级实现设计,因此也称为掩模(Mask)ASIC。采用 ASIC 能得到最高速度、最低功耗和最省面积的设计。它要求设计者必须使用版图编辑工具从晶体管的版图尺寸、位置及连线开始进行设计,以得到芯片的最优性能。在版图设计时,设计者需手工设计版图并精心布局布线,以获取最佳的性能和最小的面积。版图设计完成后,还要进行一系列检查和验证,包括设计规则检查、电器规则检查、连接性检查、版图与电路图一致性检查等,全部通过后,才可以将得到的标准格式的版图文件(一般为 CIF、GDSII 格式)交付半导体厂家进行流片。

是用 PLD 还是用 ASIC 来实现设计,需根据具体情况进行选择。对于一般的设计开发而言,采用 PLD 器件来实现,可使设计周期短、投入资金少、风险小。对于一些成熟的设计来说,可以考虑把系统中的某些模块,或者整个系统,采用 ASIC 的形式来实现,以获得最优的性价比。

1.4 硬件描述语言简介

硬件描述语言(Hardware Description Language, HDL)是一种用文本形式来描述和设计电路的语言,设计者可利用 HDL 语言来描述所做的设计,然后利用 EDA 工具进行综合和仿真,最后变为某种目标文件,再用 ASIC 或 PLD 具体实现。

HDL 最早是由 Iverson 公司于 1962 年提出的,到 20 世纪 80 年代时,已出现了数十种硬件描述语言,它们对设计自动化起到了推动和促进作用。但是,这些语言一般面向特定的设计领域和层次,而且众多的语言使用户感到无所适从,因此急需一种面向多领域、多层次,并得到普遍认同的 HDL 语言。最终,VHDL 和 Verilog HDL 满足了这种趋势的要求,先后成为 IEEE 标准语言。

1) 超高速集成电路硬件描述语言(Very High Speed Integration Circuit HDL, VHDL)

VHDL 是 1985 年在美国国防部的支持下正式推出的,1987 年成为 IEEE1076 标准,1988 年成为工业标准,1993 年经过修改成为 IEEE1064 标准,1996 年经电路合成标准程序与规格加入的 VHDL,成为 IEEE1976.3 标准。

VHDL 是一种全方位的语言,包括从系统到电路的所有设计层次,在描述风格上支持结

构、数据流和行为三种形式的描述方式。VHDL 具有以下特点:

- (1) 可以将一个系统分成若干层次进行分别设计。
- (2) 每一个设计模块都有定义完善的接口和精确的行为说明。
- (3) 可以使用算法或硬件结构定义一个模块的行为说明。
- (4) 模型化的并发、时序和时钟,支持异步和同步时序电路设计。

2) Verilog HDL

Verilog HDL 是另一种应用较为广泛的硬件描述语言,它是在 C 语言的基础上发展而来的。1983 年 GDA (Gateway Design Automation) 公司的 Phil Moorby 创造了 Verilog HDL, 1984—1985 年开发出了第一个 Verilog 仿真器,1989 年 GDA 公司被 CADENCE 公司收购, Verilog HDL 成了 CADENCE 公司的专利,1995 年成为 IEEE 标准。

Verilog HDL 语言适合算法级 (Algorithm Level)、寄存器传输级 (Register Transfer Level, RTL)、门级 (Gate Level) 和版图级 (Layout Level) 等各个层次的设计和描述,也可用于仿真验证、时序分析等。

3) Verilog HDL 与 VHDL 的比较

两者的共同点在于:都能形式化地、抽象地表示电路的结构和行为;支持逻辑设计中层次与结构的表达可借用高级语言来简化电路的描述;具有电路仿真与验证机制以保证设计的正确性;支持电路描述由顶层到底层的综合和转换;便于文档管理,易于理解和移植重用。

Verilog HDL 和 VHDL 有各自的优势和特点:Verilog HDL 的设计资源比 VHDL 更加丰富;与 VHDL 相比,Verilog HDL 的语法结构更加灵活,比较容易掌握,可使用户集中精力投入设计工作,而不必花费太多的时间在语言和语法的学习上。

目前市场上所有的 EDA 工具都同时支持这两种语言,而在 ASIC 设计领域,Verilog HDL 则占有明显的优势。

1.5 FPGA/CPLD 综述

1.5.1 FPGA/CPLD 简介

PLD 的全称是 Programmable Logic Device (可编程逻辑器件),它是一种数字集成电路的半成品,在其芯片上按一定排列方式集成了大量的门和触发器等基本逻辑元件,用户可利用某种开发工具对其进行加工,即按实际要求将这些片内的元件连接起来(此过程称为编程),使之完成某个逻辑电路或系统的功能,从而成为一个可在实际电子系统中使用的专用集成电路。现在应用最广泛的 PLD 主要是 FPGA 和 CPLD。

1985 年 Xilinx 公司首家推出了现场可编程门阵列器件 (Field Programmable Gate Array, FPGA),它是一种新型的高密度 PLD,采用 CMOS-SRAM 工艺制作。其结构和阵列型 PLD 不同,它的内部由许多独立的可编程逻辑模块组成,逻辑模块之间可以灵活地相互连接,具有密度高、编程速度快、设计灵活和可再配置设计能力等许多优点。FPGA 出现后立即受到世界范围内广大电子工程师的普遍欢迎,并得到迅速发展。

20 世纪 80 年代末,在 Lattice 公司提出在系统可编程 (In System Programmable, ISP) 技术后,相继出现了一系列具备在系统可编程能力的复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD),CPLD 是在 EPLD 的基础上发展起来的,它采用 E²CMOS 工艺制作,增加了内部连线,改进了内部结构体系,因而其性能更好,设计更加灵活,其发展也非常迅速。

FPGA 和 CPLD 都是可编程逻辑器件,它们是在 PAL、GAL 等逻辑器件的基础上发展起来的。同以往的 PAL、GAL 等相比较,FPGA 和 CPLD 的规模比较大,可以代替几十甚至几千块通用 IC 芯片,并且可以反复地编程、擦除,在外围电路不动的情况下用不同软件就可实现不同的功能。

CPLD 和 FPGA 在结构和应用上在以下方面各有特点和长处:

1) 结构

FPGA 器件是由逻辑功能块排列为阵列,并由可编程的内部连线连接这些功能块来实现一定的逻辑功能。CPLD 是由可编程与/或门阵列以及宏单元构成。

2) 集成度

FPGA 可以达到比 CPLD 更高的集成度,同时也具有更复杂的布线结构和逻辑实现。

3) 适合结构

CPLD 组合逻辑的功能很强,一个宏单元就可以分解 10 多个甚至 20~30 个组合逻辑输入,而 FPGA 的 1 个 LUT 只能处理 4 输入的组逻辑,因此,CPLD 适用于设计译码等复杂的组合逻辑。FPGA 的制造工艺确定了 FPGA 芯片中包含的 LUT 和触发器的数量非常多,往往达到数千上万个,而 CPLD 一般只能做到 512 个逻辑单元,因此,如果设计中需要用到大量的触发器,例如设计一个复杂的时序逻辑,就应使用 FPGA。

4) 功率消耗

一般情况下,CPLD 功能消耗比 FPGA 要大,且集成密度越高越明显。

5) 速度

CPLD 的速度优于 FPGA。由于 FPGA 是门级编程,且逻辑块之间是采用分布式互连;而 CPLD 是逻辑块级编程,且逻辑块互连是集总式的。因此,CPLD 比 FPGA 有较高的速度和较大的时间可预测性。

6) 编程方式

目前,CPLD 主要是基于 E^2 PROM 或 FLASH 存储器编程,编程次数达 1 万次,其优点是在系统断电后,编程信息不丢失。CPLD 又可分为在编程器编程和在系统编程两种。在系统编程器件的优点是:不需要编程器可先将器件装焊于印制板,再经过编程电缆进行编程,编程、调试和维护都很方便。

FPGA 大部分是基于 SRAM 编程,其缺点是编程数据信息在系统断电时丢失,每次上电时,需从器件的外部存储器或计算机中将编程数据写到 SRAM 中。其优点是可进行任意次数的编程,并在工作中快速编程,实现板级和系统级的动态配置,因此可称为在线重配置(In Circuit Reconfigurable,ICR)的 PLD 或可重配置硬件(Reconfigurable Hardware Product,RHP)。

7) 使用方便性

在使用方便性上,CPLD 比 FPGA 要好。CPLD 的编程工艺采用 E^2 PROM 或 FLASH 技术,无需外部存储器芯片,使用简单,保密性好。而基于 SRAM 编程的 FPGA,其编程信息需存放在外部存储器上,需外部存储器芯片,且使用方法复杂,保密性差。

1.5.2 基于 FPGA/CPLD 的数字系统开发流程

基于 FPGA/CPLD 器件的数字系统设计流程如图 1.5.1 所示,主要包括设计输入、设计处理、仿真、编程下载和在线测试等步骤。

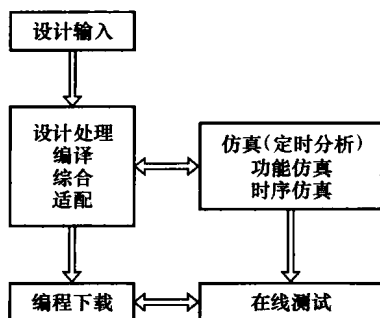


图 1.5.1 基于 FPGA/CPLD 的数字系统开发流程

1) 设计输入

设计输入是由设计者对器件所实现的数字系统的逻辑功能进行描述,主要有原理图输入法、真值表输入法、状态机输入法、波形输入法、硬件描述语言输入法等。常用的有原理图输入法和硬件描述语言输入法。

原理图输入法是基于传统的硬件电路设计思想,把数字逻辑系统用逻辑原理图进行表示的输入方法,使用逻辑器件(即元件符号)和连线等进行描述设计。硬件描述语言输入法是一种用文本形式来描述和设计电路的方法,设计者可利用 HDL 来描述自己的设计,然后利用 EDA 工具进行综合和仿真,最后变为某种目标文件,再用 ASIC 或 FPGA 具体实现。

HDL 和传统的原理图输入法的关系就好比是高级语言和汇编语言的关系。HDL 的可移植性好,使用方便,但效率不如原理图输入法;原理图输入法的可控性好,效率高,比较直观,但设计大规模 PLD 时显得很烦琐,移植性差。

在 Top-Down 设计方法中,描述器件总功能的模块放置在最上层,称为顶层设计;描述器件最基本功能的模块放置在最下层,称为底层设计。可以在任何层次使用原理图或硬件描述语言进行描述。通常做法是:在顶层设计中,使用原理图输入法表达连接关系和芯片内部逻辑到管脚的接口;在底层设计中,使用 HDL 描述各个模块的逻辑功能。

2) 设计处理

设计处理是基于 FPGA/CPLD 的数字系统开发流程中的中心环节,在该阶段,编译软件将对设计输入文件进行逻辑优化、综合,并利用一片或多片 FPGA/CPLD 器件自动进行适配,最后产生可用于编程的数据文件。该环节主要包含设计编译、逻辑综合优化、适配和布局、生成编程文件等。

(1) 设计编译

设计输入完成后,立即进行设计编译,EDA 编译器首先从工程设计文件间的层次结构描述中提取信息,包含每个低层次文件中的错误信息,如原理图中信号线有无漏接、信号有无多重来源、文本输入文件中的关键字有无错误或其他语法错误,并及时标出错误的位置,供设计者排除纠正,然后进行设计规则检查,检查设计有无超出器件资源或规定的限制,并将给出编译报告。

(2) 逻辑综合优化

综合就是将电路的高级语言转换成低级的、可与 FPGA/CPLD 的基本结构相对应的网表文件或程序,由综合器来完成。利用综合器对 HDL 源代码进行综合优化处理,生成门级描述的网表文件,它是将高层次描述转化为硬件电路的关键步骤。

(3) 适配和布局

利用适配器可将综合后的网表文件针对某一确定的目标器件进行逻辑映射,该操作包括底层器件配置、逻辑分割、逻辑优化、布局布线等。

(4) 生成编程文件

适配和布局环节是在设计检验通过后,由 EDA 软件自动完成的,它能以最优的方式对逻辑元件进行逻辑综合和布局,并准确实现元件间的互联,同时 EDA 软件会生成相应的报告文件。适配和布局完成后,可以利用适配所产生的仿真文件做精确的时序仿真,同时产生可用于编程的数据文件。对于 CPLD 而言,是产生熔丝图文件,即 JEDEC 文件;对于 FPGA 而言,则生成数据流文件 BG(Bit-stream Generation)。

3) 仿真与定时分析

仿真和定时分析均属于设计校验,其作用是测试设计的逻辑功能和延时特性。仿真包括功能仿真和时序仿真。功能仿真又称前仿真,这种仿真不考虑信号的延时,主要检验逻辑功能的正确性。时序仿真又称后仿真,是进行布局布线后进行的仿真,它能够模拟器件实际工作时的情况。定时分析器可通过三种不同的分析模式分别对传播延时、时序逻辑性能和建立/保持时间进行分析。

4) 编程与验证

将得到的编程文件通过编程电缆配置 PLD,加入实际激励,进行在线测试。

1.5.3 FPGA/CPLD 主要厂商及产品

目前世界上有十几家生产 FPGA/CPLD 的公司,最大的三家是:Altera 公司、Xilinx 公司、Lattice 公司。

1) Altera 公司 FPGA 和 CPLD 器件系列

- (1) Stratix II 系列 FPGA
- (2) Stratix 系列 FPGA
- (3) ACEX 系列 FPGA
- (4) FLEX 系列 FPGA
- (5) MAX 系列 CPLD
- (6) Cyclone 系列 FPGA 低成本 FPGA
- (7) Cyclone II 系列 FPGA
- (8) MAX II 系列器件
- (9) Altera 宏功能块及 IP 核

2) Xilinx 公司的 FPGA 和 CPLD 器件系列

- (1) Virtex-4 系列 FPGA
- (2) Spartan II & Spartan-3 & Spartan 3E 器件系列
- (3) XC9500 & XC9500XL 系列 CPLD
- (4) Xilinx FPGA 配置器件 SPROM
- (5) Xilinx 的 IP 核
- (6) Cyclone 系列 FPGA 低成本 FPGA

3) Lattice 公司 CPLD 器件系列

- (1) ispMACH4000 系列
- (2) Lattice EC & ECP 系列

需要指出的是,不同厂家对 CPLD 和 FPGA 的称谓不尽相同。Xilinx 公司把基于查找表技术、SRAM 工艺、要外挂配置用的 E²PROM 的 PLD 称为 FPGA,把基于乘积项技术、Flash ROM 工艺的 PLD 称为 CPLD;Altera 公司把自行生产的 PLD 产品 MAX 系列(乘积项技术、E²PROM 工艺)和 FLEX 系列(查表技术、SRAM 工艺)都称为 CPLD。由于 FLEX 系列也是基于查表技术、SRAM 工艺、要外挂配置用的 EPROM,且用法和 Xilinx 公司的 FPGA 一样,所以,很多人把 Altera 公司的 FLEX 系列产品也称为 FPGA。

1.5.4 FPGA/CPLD 的 EDA 开发工具

常用的 FPGA/CPLD 的 EDA 开发工具一般有集成开发工具和专业开发工具两种类型。

1) 集成开发工具

此种类型的开发工具是芯片制造厂商为配合自己的 FPGA/CPLD 芯片而推出的一种集成开发环境,基本上能完成其 FPGA/CPLD 开发的所有工作,包括输入、仿真、综合、布线、下载等。此类开发工具应用在其公司的 FPGA/CPLD 芯片上,能提高设计效率,优化设计结果,充分利用芯片资源。其缺点是综合能力较差,不支持其他器件厂商出品的器件。

由 Altera 公司、Xilinx 公司、Lattice 公司开发的集成开发工具如下:

(1) Altera 公司:MAX+plusII、QuartusII。

(2) Xilinx 公司:Foundition、ISE。

(3) Lattice 公司:ispLEVER。

2) 专业开发工具

此种类型的开发工具能进行更为复杂和更高效的设计。一般有专业设计输入工具、专业逻辑综合器、专业仿真器等。

(1) 常用的专业设计输入工具有:Mentor 公司的 HDL Designer Series,通用编辑器 Ultra-Edit,Innovada 公司的 Visual HDL。

(2) 常用的专业逻辑综合器有:Synplicity 公司的 Synplify 和 Synplify Pro,Synopsys 公司的 FPGA Express、FPGA Compiler 和 FPGA CompilerII,Mentor 公司的 Leonardo Spectrum。

(3) 常用的专业仿真器有:Mentor 的子公司的 ModelSim,Cadance 公司的 NC-Verilog/NC-VHDL/NC-Sim Verilog-XL,Synopsys 公司的 VCS/Scriocco,Aldec 公司的 Active HDL。

1.6 数字系统设计实验说明

本书作为《数字系统设计》课程的实验配套教材,可以根据课程的学时安排选择不同的实验内容。《数字系统设计》课程是一门实践性很强的课程,可安排本书 40%~45%的实验内容。

1.6.1 实验规则

为了维护正常的实验教学次序,提高实验课的教学质量,顺利地地完成各项实验任务,确保人身安全、设备完好,特制定如下实验规则:

(1) 实验前必须充分预习,认真阅读本实验指导书,掌握本次实验的基本原理,并写出预习报告。

(2) 实验时,认真、仔细地写出源程序,进行调试,有问题及时向指导教师请教。

(3) 实验时应仔细观察,如发现有异常现象(电脑故障或实验箱故障)的发生,必须及时报告指导教师,严禁私自乱动。

(4) 使用实验箱时,应注意:

①打开总电源前,不要打开实验箱电源开关。

②不要随意拔插实验箱上的器件。

③实验完毕,及时关闭电源开关。

(5) 实验过程中,应仔细观察实验现象,认真记录实验数据、波形、逻辑关系及其他现象。

(6) 自觉保持实验室的肃静、整洁;实验结束后,必须清理实验桌,将实验设备按规定放好,并填写仪器设备使用记录。

(7) 每个实验结束后,必须按要求及时撰写实验报告,下次实验时交实验指导教师批阅。未交实验报告者,必须在规定时间内交给实验指导教师,否则视为缺做一次实验。

以上实验规则,请同学们自觉遵守,并互相监督。

1.6.2 实验报告的撰写

实验报告是实验工作的全面总结和最终成果的文本呈现,要求能完整而真实地反映实验结果。

实验报告应书写工整、语句通顺、数据真实、图表清晰,并能从实验过程的观察中找出问题进行分析和讨论,发表自己的见解。

1) 预习报告要求

(1) 实验名称;

(2) 实验目的;

(3) 实验内容及实验原理分析;

(4) 设计原理图或实验源程序;

(5) 主要实验步骤。

2) 实验报告要求

(1) 对预习报告中逻辑图和源程序的补充和修改;

(2) 经过整理后的数据、波形等;

(3) 详细的实验步骤;

(4) 分析和结论。



基础实验

本章实验内容主要为数字电路中一些常用的基本数字逻辑单元设计实验和 Verilog HDL 语言基本设计方法训练实验。设置这部分实验有两个目的:一是熟悉实验基本设计流程并初步学会实验箱的使用;二是为后面的综合设计实验及课程设计实验打下基础,同时为复杂电路提供已设计好且经过验证的模块,以备使用。

2.1 基本组合逻辑电路设计实验

实验 1 加法器设计

1) 实验目的

- (1) 复习加法器的分类及工作原理。
- (2) 掌握用图形法设计半加器的方法。
- (3) 掌握用元件例化法设计全加器的方法。
- (4) 掌握用元件例化法设计多位加法器的方法。
- (5) 掌握用 Verilog HDL 语言设计多位加法器的方法。
- (6) 学习运用波形仿真验证程序的正确性。
- (7) 学习定时分析工具的使用方法。

2) 实验原理

加法器是能够实现二进制加法运算的电路,是构成计算机中算术运算电路的基本单元。目前,在数字计算机中,无论加、减、乘、除法运算,都是化为若干步加法运算来完成的。加法器可分为 1 位加法器和多位加法器两大类。1 位加法器又可分为半加器和全加器两种,多位加法器可分为串行进位加法器和超前进位加法器两种。

(1) 半加器

如果不考虑来自低位的进位而将两个 1 位二进制数相加,称半加。实现半加运算的电路则称为半加器。若设 A 和 B 是两个 1 位的加数, S 是两者相加的和, C 是向高位的进位。则由二进制加法运算规则可以得到:

$$\begin{aligned} S &= A'B + AB' = A \oplus B \\ C &= AB \end{aligned} \quad (2.1.1)$$

(2) 全加器

在将两个 1 位二进制数相加时,除了最低位以外,每一位都应该考虑来自低位的进位,即将两个对应位的加数和来自低位的进位三个数相加,这种运算称全加。实现全加运算的电路则称为全加器。