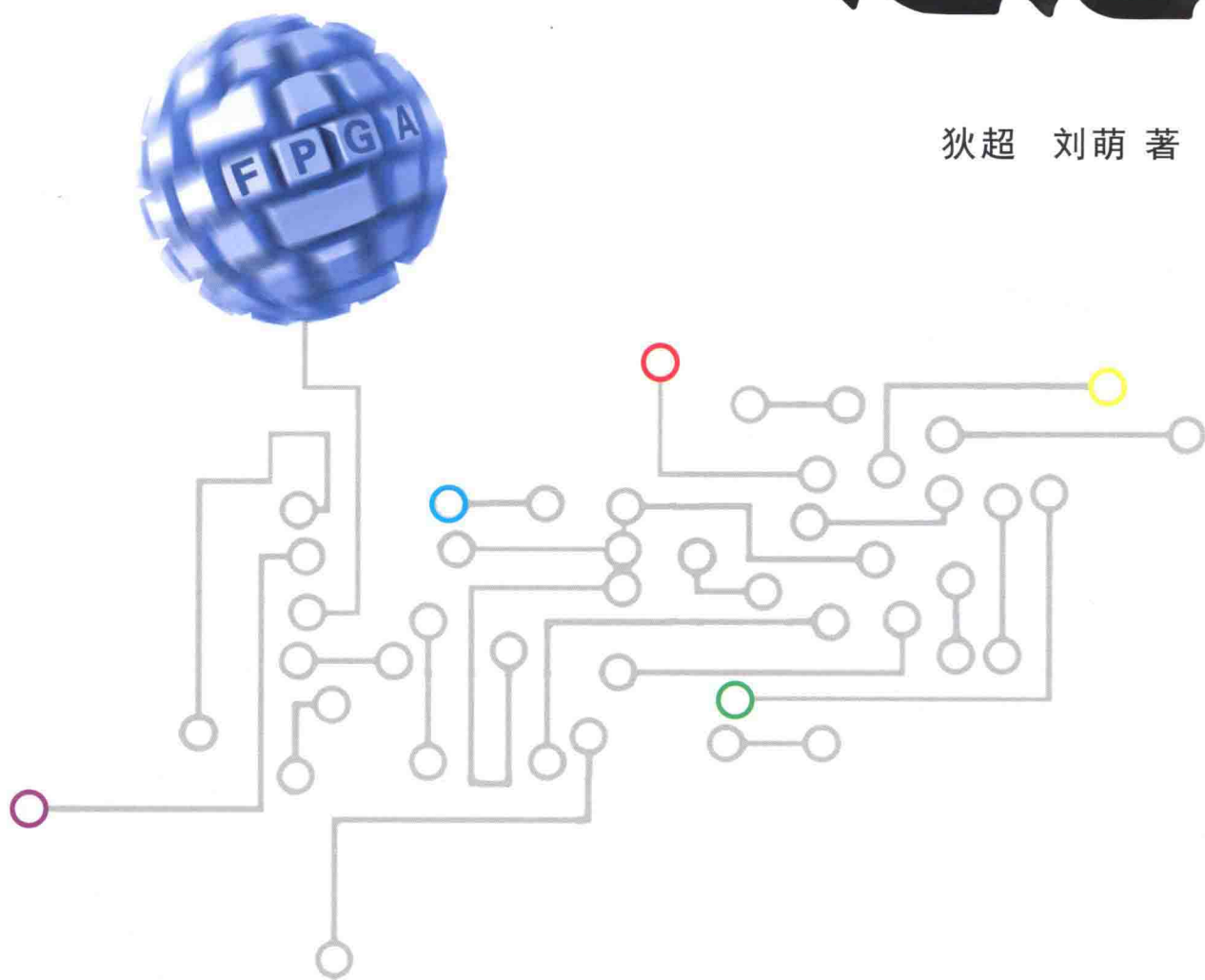


FPGA

之道

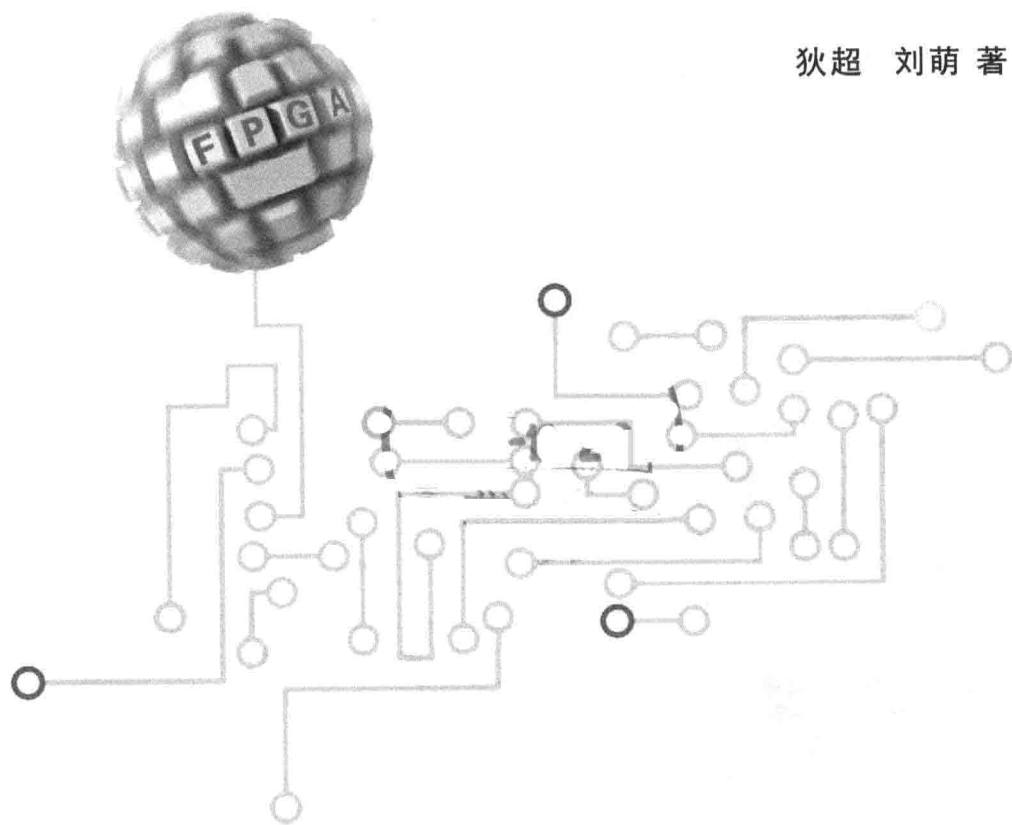
狄超 刘萌 著



FPGA

之道

狄超 刘萌 著



 西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

本书是一本针对 FPGA 技术进行全面、深入讲解的书籍,内容涵盖了数字电路相关基础理论的介绍、FPGA 芯片的构成及工作原理、FPGA 项目开发的全流程追踪、主流 HDL 与 HVL 语法的详细讲解与阐述等,尤其是针对 FPGA 项目开发流程中最为重要的三个环节——程序设计、功能仿真、时序分析——进行了深度剖析,通过丰富的思路阐述与实例介绍相结合的形式,力求让读者对 FPGA 技术能够“知其然”,且“知其所以然”。本书不仅是教材,更是工具书,适用于所有跟 FPGA 有缘的人阅读,无论你是本科生、研究生、老师或者工程师,无论你在校园、国企、外企或私企,只要在你的工作或生活中需要接触 FPGA,相信本书就是一本对你有帮助的书。本书是笔者从事 FPGA 事业十年经验的总结、两年心血的结晶,因此本书带给大家的绝不是空洞的口号,而是实实在在的 FPGA 技术。

图书在版编目(CIP)数据

FPGA 之道/狄超著. —西安:西安交通大学出版社,
2014.5
ISBN 978-7-5605-6189-9

I. ①F… II. ①狄… III. ①可编程序逻辑器件
IV. ①TP332.1

中国版本图书馆 CIP 数据核字(2014)第 094160 号

书 名 FPGA 之道
著 者 狄 超 刘 萌
责任编辑 杨 璠

出版发行 西安交通大学出版社
(西安市兴庆南路 10 号 邮政编码 710049)
网 址 <http://www.xjtupress.com>
电 话 (029)82668357 82667874(发行中心)
(029)82668315 82669096(总编办)
传 真 (029)82668280
印 刷 北京京华虎彩印刷有限公司

开 本 787mm×1092mm 1/16 印张 75.75 字数 1891 千字
版次印次 2014 年 8 月第 1 版 2014 年 8 月第 1 次印刷
书 号 ISBN 978-7-5605-6189-9/TP·617
定 价 338.00 元

读者购书、书店添货、如发现印装质量问题,请与本社发行中心联系、调换。

订购热线:(029)82665248 (029)82665249

投稿热线:(029)82669097 QQ:8377981

读者信箱:lg_book@163.com

版权所有 侵权必究

序

拿到本书样稿之时,惊讶之余,更多是惊喜。这本厚度堪比字典的关于 FPGA 设计的书,把 FPGA 入门、设计、验证的方法及过程详实严谨地展现在我们面前,细节部分的讲解,不乏作者的独到用心,全书提纲挈领而具有可操作性,相信对于广大读者有正确的引导作用。

该书作者在我们研究中心 3 年读研期间,其 FPGA 设计的才能已崭露头角,后经 2 年 FPGA 设计验证及 3 年 FPGA 专业开发工作的历练,已成为 FPGA 方面不可多得的专业人才,其学术水准在我们的研究生中间也颇具威信。

我问作者,作为一个不受论文、专著指标约束的“体制外”的人,为何耗费精力编写并出版这样的书?作者回答说:“回顾自己多年的 FPGA 设计经历,感觉大部分时间都花在一些细节处的思考和实践上,若是书中能够多些指点,开发效率自然会提高许多。写书的目的就是想把自己多年的经验、教训和感悟告诉后来者,让他们能少走弯路。”

这是一本有感而发的兴趣书,也是一本开卷有益良心书!有幸得到的样稿,亦经被争相传阅,篇幅虽长,但处处都可圈可点。我认为,它应该成为从事 FPGA 设计研究生及工程师的手边书,有困惑难解之处,可以随时请教这位“良师益友”。

西安交大电信学院数据广播研究中心 沈雪峰

2013 年 9 月 25 日

目 录

1 与我同行	(1)
1.1 个人简介	(1)
1.2 “道”的含义	(1)
1.3 写作目的	(2)
1.3.1 技术人生,有总结才有提高.....	(2)
1.3.2 从入门到精通,带你走上 FPGA 之路	(2)
1.3.3 即是教材,也是参考.....	(3)
1.3.4 一起出生吧!	(3)
1.4 内容综述	(4)
1.5 默认共识	(5)
1.6 3,2,1,我们开始	(6)
2 共同语言	(7)
2.1 什么是 FPGA?	(7)
2.1.1 名词解释	(7)
2.1.2 发展简史	(7)
2.1.3 应用方向	(8)
2.2 模拟与数字.....	(10)
2.2.1 模拟的与数字的.....	(11)
2.2.2 模拟信号与数字信号.....	(11)
2.2.2.1 模拟信号的概念.....	(12)
2.2.2.2 数字信号的概念.....	(12)
2.2.2.3 模拟信号转换为数字信号.....	(13)
2.2.3 模拟电路与数字电路.....	(14)
2.2.3.1 电信号的本质.....	(14)
2.2.3.2 模拟电路简介.....	(15)
2.2.3.3 数字电路简介.....	(16)
2.2.4 模拟信号处理电路系统与数字信号处理电路系统.....	(17)
2.2.4.1 模拟信号处理电路系统.....	(17)
2.2.4.2 数字信号处理电路系统.....	(17)
2.2.4.3 模拟系统与数字系统的优缺点比较.....	(18)
2.2.4.4 模拟系统与数字系统之间的接口.....	(19)
2.2.4.5 数字系统之间的接口.....	(20)
2.3 数字逻辑电路基础知识.....	(24)

2.3.1	数与数制简介	(24)
2.3.1.1	常见数制简介	(24)
2.3.1.2	数制间的转换	(25)
2.3.1.3	二进制编码简介	(30)
2.3.2	数字逻辑的基本运算	(32)
2.3.2.1	集合与数字逻辑	(32)
2.3.2.2	数字逻辑的基本运算	(33)
2.3.2.3	真值表	(34)
2.3.2.4	数字逻辑运算的基本公式与规律	(35)
2.3.2.5	数字逻辑运算中的三个规则	(36)
2.3.3	数字逻辑的化简	(37)
2.3.3.1	公式化简法	(37)
2.3.3.2	卡诺图化简法	(38)
2.3.3.3	系统化简法	(46)
2.3.4	数字逻辑功能单元	(49)
2.3.4.1	基本逻辑功能单元简介	(50)
2.3.4.2	小规模集成组合逻辑单元简介	(57)
2.3.4.3	时序逻辑基本单元简介	(71)
2.3.5	数字逻辑的载体	(93)
2.3.5.1	半导体器件简介	(93)
2.3.5.2	CMOS 门电路举例	(98)
2.4	硬件描述语言	(102)
2.4.1	图形化设计方法简介	(102)
2.4.2	从图形化设计到 HDL	(103)
2.4.3	HDL 简史	(103)
2.4.3.1	VHDL 简史	(103)
2.4.3.2	Verilog HDL 简史	(104)
2.4.4	软件编程思路与 FPGA 编程思路的变革	(105)
2.4.4.1	名称对比分析	(105)
2.4.4.2	抽象层级对比	(107)
2.4.4.3	编译原理对比	(107)
2.4.4.4	执行方式对比	(108)
2.5	硬件扩展	(110)
2.5.1	电源转换芯片系列	(110)
2.5.2	数模转换芯片系列	(110)
2.5.3	数据存储芯片系列	(111)
2.5.4	微处理器芯片系列	(112)
2.6	最自由的开发者	(113)

3 知己知彼	(114)
3.1 可编程逻辑器件的原理架构	(114)
3.1.1 集成电路简介	(114)
3.1.2 PLA 原理架构	(115)
3.1.3 PAL 原理架构	(115)
3.1.4 GAL 原理架构	(116)
3.1.5 FPLA 原理架构	(118)
3.1.6 CPLD 原理架构	(119)
3.1.7 FPGA 原理架构	(119)
3.2 FPGA 产品介绍	(121)
3.2.1 Xilinx 公司 FPGA 产品简介	(121)
3.2.2 Altera 公司 FPGA 产品简介	(121)
3.2.3 其他公司 FPGA 产品简介	(121)
3.2.4 FPGA 产品的工业等级简介	(122)
3.2.5 FPGA 产品的速度等级简介	(122)
3.3 FPGA 内部资源介绍	(123)
3.3.1 逻辑资源块	(123)
3.3.2 时钟网络资源	(126)
3.3.3 时钟处理单元	(127)
3.3.3.1 PLL	(127)
3.3.3.2 DCM	(129)
3.3.4 BLOCK RAM	(130)
3.3.5 DSP	(130)
3.3.6 布线资源	(132)
3.3.7 接口资源	(133)
3.3.8 专用高速接口资源	(135)
3.3.9 To be continued	(135)
3.4 FPGA 芯片的供电	(135)
3.4.1 外部端口供电机制	(135)
3.4.2 内部逻辑供电机制	(136)
3.4.3 专有电路供电机制	(136)
3.4.4 电源稳定性讨论	(137)
3.5 FPGA 芯片的配置方法	(137)
3.5.1 主动配置模式	(137)
3.5.2 被动配置模式	(138)
3.5.3 JTAG 配置模式	(139)
3.5.4 应用模式简介	(139)
3.6 IP 核介绍	(140)
3.6.1 IP 核概述	(140)

3.6.2	嵌入式微处理器概述	(141)
3.7	组合逻辑电路与时序逻辑电路	(141)
3.8	同步逻辑电路与异步逻辑电路	(142)
4	开发流程	(145)
4.1	背景知识的分析与研究	(145)
4.2	项目方案的设计与制定	(146)
4.2.1	写清楚项目背景	(147)
4.2.2	写清楚项目需求	(147)
4.2.3	写清楚方案框架	(147)
4.2.4	写清楚算法细节	(148)
4.2.5	确保逻辑完备性	(148)
4.2.6	确保实现无关性	(148)
4.2.7	确保书面易懂性	(149)
4.3	算法可行性仿真与验证	(149)
4.3.1	Why	(149)
4.3.2	When	(151)
4.3.3	How	(151)
4.4	FPGA 设计方案的制定	(151)
4.4.1	编写 FPGA 设计方案的好处	(152)
4.4.2	如何编写 FPGA 设计方案	(152)
4.5	FPGA 功能代码的编写	(153)
4.6	FPGA 设计的功能仿真	(154)
4.7	FPGA 顶层模块的门级仿真	(156)
4.7.1	门级仿真简介	(156)
4.7.2	形式化验证简介	(157)
4.7.3	形式化验证在 FPGA 设计中的应用	(158)
4.8	FPGA 设计的用户约束	(159)
4.9	FPGA 设计的时序分析	(160)
4.10	FPGA 顶层模块的时序仿真	(163)
4.11	FPGA 设计的实现过程	(164)
4.11.1	编译概述	(165)
4.11.2	编译流程之综合	(165)
4.11.2.1	综合的输入	(165)
4.11.2.2	综合的输出	(168)
4.11.2.3	综合的工具	(169)
4.11.3	编译流程之翻译融合	(169)
4.11.3.1	翻译融合的输入	(169)
4.11.3.2	翻译融合的输出	(170)

4.11.3.3	翻译融合工具	(171)
4.11.4	编译流程之映射	(171)
4.11.4.1	映射的输入	(171)
4.11.4.2	映射的输出	(173)
4.11.4.3	映射工具	(174)
4.11.5	编译流程之布局布线	(174)
4.11.5.1	布局布线的输入	(174)
4.11.5.2	布局布线的输出	(176)
4.11.5.3	布局布线工具	(176)
4.11.6	编译流程之配置生成	(176)
4.11.6.1	配置生成的输入	(177)
4.11.6.2	配置生成的输出	(177)
4.11.6.3	配置生成工具	(178)
4.11.7	高级编译方法简介	(178)
4.11.7.1	Partition	(178)
4.11.7.2	手动布局布线	(181)
4.11.7.3	智能化实现指导	(182)
4.11.7.4	时序收敛方案	(182)
4.12	开发板制作	(182)
4.13	FPGA 设计的上板调试	(183)
4.13.1	“实践是检验真理的唯一标准”	(183)
4.13.2	如何解决问题	(183)
4.13.3	如何找到问题	(185)
4.13.4	“时间是检验真理的第二标准”	(186)
4.14	项目总结 备份与后期维护	(187)
4.15	开发流程中的团队分工	(187)
5	程序设计	(189)
5.1	编程语法	(189)
5.1.1	VHDL 基本语法	(189)
5.1.1.1	VHDL 基本程序框架	(189)
5.1.1.2	VHDL 注释语法	(198)
5.1.1.3	VHDL 的 signal、variable 与 constant	(199)
5.1.1.4	VHDL 数据类型	(200)
5.1.1.5	VHDL 初始化	(206)
5.1.1.6	VHDL 的操作符号	(207)
5.1.1.7	VHDL 的并行语句	(215)
5.1.1.8	VHDL 的串行语句	(225)
5.1.1.9	编写自己的 VHDL 库文件	(234)

5.1.1.10	VHDL 编写注意事项	(240)
5.1.2	Verilog 基本语法	(246)
5.1.2.1	Verilog 基本程序框架	(246)
5.1.2.2	Verilog 注释语法	(253)
5.1.2.3	Verilog 数据类型	(254)
5.1.2.4	Verilog 初始化	(260)
5.1.2.5	Verilog 的操作符号	(261)
5.1.2.6	Verilog 的并行语句	(271)
5.1.2.7	Verilog 的串行语句	(285)
5.1.2.8	Verilog 中的编译指令	(294)
5.1.2.9	Verilog 中的编写注意事项	(296)
5.1.3	VHDL 与 Verilog 的比较	(301)
5.1.3.1	语法比较	(301)
5.1.3.2	语言比较	(304)
5.2	编程思路	(305)
5.2.1	代码风格	(306)
5.2.1.1	代码风格四步走	(306)
5.2.1.2	HDL 选择	(306)
5.2.1.3	命名规则	(307)
5.2.1.4	语法结构	(311)
5.2.1.5	空格空行	(316)
5.2.1.6	注释编写	(317)
5.2.1.7	模块设计	(320)
5.2.1.8	文件结构	(324)
5.2.2	三种描述方式	(324)
5.2.2.1	结构化描述方式	(324)
5.2.2.2	数据流描述方式	(326)
5.2.2.3	行为级描述方式	(328)
5.2.3	顶层设计模块	(330)
5.2.3.1	概念浅析	(330)
5.2.3.2	特点说明	(331)
5.2.4	设计的分类	(332)
5.2.4.1	按功能分	(332)
5.2.4.2	按面向分	(332)
5.2.4.3	按速度和规模分	(333)
5.2.5	编写纯净的组合或时序逻辑	(334)
5.2.5.1	组合逻辑描述方法	(334)
5.2.5.2	纯时序逻辑描述方法	(336)
5.2.5.3	清晰的时序逻辑描述方法	(338)

5.2.5.4	有隐患的混写逻辑	(340)
5.2.6	正确的变量访问思路	(345)
5.2.6.1	变量访问思路概述	(345)
5.2.6.2	赋值冲突	(348)
5.2.6.3	总线是怎么回事	(353)
5.2.7	数字电路中的隐患	(361)
5.2.7.1	寄存器输出的不稳定态	(361)
5.2.7.2	组合逻辑中的竞争与险象	(369)
5.2.8	时钟及时钟域	(385)
5.2.8.1	时钟,时序逻辑的心跳.....	(385)
5.2.8.2	时钟信号基本特征	(385)
5.2.8.3	时钟信号的分类	(386)
5.2.8.4	时钟域	(389)
5.2.8.5	跨时钟域问题	(394)
5.2.9	DCM 与 PLL	(403)
5.2.9.1	PLL 模块基本端口简介	(403)
5.2.9.2	DCM 模块基本端口简介	(406)
5.2.9.3	应用场合	(409)
5.2.10	复位的设计.....	(416)
5.2.10.1	为什么 FPGA 设计中要有复位	(416)
5.2.10.2	复位方式的分类.....	(417)
5.2.10.3	复位的设计方法.....	(418)
5.2.11	数据的存储.....	(429)
5.2.11.1	为什么需要数据存储.....	(429)
5.2.11.2	数据存储的载体.....	(429)
5.2.11.3	数据存储的形式、实现及应用场合	(430)
5.2.11.4	寄存器的 HDL 描述与用法	(433)
5.2.11.5	单口 RAM 的 HDL 描述与用法	(434)
5.2.11.6	双口 RAM 的 HDL 描述与用法	(441)
5.2.11.7	ROM 的 HDL 描述与用法	(443)
5.2.11.8	同步 FIFO 的 HDL 描述与用法	(446)
5.2.11.9	异步 FIFO 的 HDL 描述与用法	(448)
5.2.11.10	FIFO 使用小技巧之冗余法	(461)
5.2.11.11	STACK 的 HDL 描述与用法.....	(465)
5.2.11.12	外部存储芯片的 HDL 描述与用法	(467)
5.2.11.13	数据存储的使用思路	(467)
5.2.12	状态机,FPGA 的灵魂	(469)
5.2.12.1	状态机的概念.....	(469)
5.2.12.2	状态机的模型.....	(472)

5.2.12.3	状态机的设计	(477)
5.2.12.4	状态机的 HDL 模板	(492)
5.2.12.5	状态的编码方式	(511)
5.2.12.6	状态机的实现方式	(513)
5.2.12.7	显式状态机与隐式状态机	(517)
5.2.13	关于外界接口的编程思路	(523)
5.2.13.1	按传递方向分类	(523)
5.2.13.2	按电气特性分类	(528)
5.2.13.3	按功能特性分类	(531)
5.2.13.4	按时钟特性分类	(532)
5.2.13.5	按数据位宽分类	(551)
5.2.13.6	按速度分类	(555)
5.2.14	数的表示与数的运算	(557)
5.2.14.1	FPGA 中数的存储形式	(557)
5.2.14.2	整数在 FPGA 中的表示形式	(557)
5.2.14.3	整数的加、减运算	(563)
5.2.14.4	整数的乘法运算	(572)
5.2.14.5	整数的除法运算	(651)
5.2.14.6	整数的其他类型运算简介	(667)
5.2.14.7	非整数运算浅析	(674)
5.2.15	时空变换	(684)
5.2.15.1	时空变换之基本概念	(684)
5.2.15.2	时空变换之时域优化	(686)
5.2.15.3	时空变换之空域优化	(702)
5.2.15.4	时空变换之时间换空间	(707)
5.2.15.5	时空变换之空间换时间	(719)
5.2.16	“万能”的查表法	(732)
5.2.16.1	正弦波发生器示例	(732)
5.2.16.2	Gamma 校正示例	(735)
5.2.17	代码中的约束信息	(741)
5.2.17.1	为什么要在 HDL 中加入约束信息	(741)
5.2.17.2	HDL 中的常用约束示例	(741)
5.2.18	原语的使用	(756)
5.2.18.1	什么是原语	(756)
5.2.18.2	需要使用原语的情况	(759)
5.2.18.3	用原语表示 IP 核的好处	(763)
5.2.18.4	UDP 简介	(767)
5.2.19	提高设计的综合性能	(768)
5.2.19.1	提高设计的鲁棒性	(768)

5.2.19.2	提高设计的自测性	(775)
5.2.19.3	提高设计的重用性	(790)
5.2.19.4	提高设计的易改性	(790)
5.2.19.5	提高设计的移植性	(806)
5.2.19.6	提高设计的保密性	(807)
5.2.20	设计方法学讨论	(808)
5.2.21	FPGA 程序设计的境界	(809)
6	功能仿真	(811)
6.1	仿真原理	(811)
6.1.1	串行模仿并行思路分析	(811)
6.1.2	有限模仿无限思路分析	(812)
6.1.3	组合逻辑仿真原理	(812)
6.1.4	时序逻辑仿真原理	(813)
6.1.5	HDL 的仿真原理	(813)
6.1.6	仿真时间与物理时间	(814)
6.2	仿真语法	(815)
6.2.1	Graphic Waveform	(815)
6.2.1.1	数字波形简介	(815)
6.2.1.2	从实际到仿真	(817)
6.2.1.3	“Hello world”之 Graphic Waveform	(818)
6.2.1.4	一些绘制波形的操作	(818)
6.2.1.5	一些观察波形的操作	(820)
6.2.1.6	波形仿真结果分析及重要注意事项	(820)
6.2.2	VHDL Test Bench	(822)
6.2.2.1	“Hello World”之 VHDL Test Bench	(822)
6.2.2.2	继承描述语法	(825)
6.2.2.3	时间相关语法	(825)
6.2.2.4	时钟激励语法	(827)
6.2.2.5	循环仿真语法	(829)
6.2.2.6	进程语句体	(831)
6.2.2.7	文件操作语法	(832)
6.2.3	Verilog Test Fixture	(837)
6.2.3.1	“Hello world”之 Verilog Test Fixture	(837)
6.2.3.2	继承描述语法	(839)
6.2.3.3	时间相关语法	(839)
6.2.3.4	时钟激励语法	(843)
6.2.3.5	循环仿真语法	(846)
6.2.3.6	程序块语句体	(847)

6.2.3.7	字符显示语法	(848)
6.2.3.8	系统存储载入函数	(849)
6.2.3.9	系统文件操作函数	(850)
6.2.3.10	系统屏幕输出函数	(855)
6.2.3.11	系统随机数生成函数	(855)
6.2.3.12	系统仿真时间函数	(857)
6.2.3.13	系统仿真进度控制任务	(858)
6.2.4	System Verilog SVT	(859)
6.2.4.1	“Hello World”之面向过程 SV Test	(859)
6.2.4.2	“Hello World”之面向对象 SV Test	(868)
6.2.4.3	测试平台相关语法	(878)
6.2.4.4	System Verilog 的数据类型	(894)
6.2.4.5	System Verilog 的数组类型	(906)
6.2.4.6	System Verilog 的操作符号	(920)
6.2.4.7	System Verilog 的过程语句	(922)
6.2.4.8	System Verilog 的进程语句	(930)
6.2.4.9	System Verilog 的任务和函数	(930)
6.2.4.10	System Verilog 的线程及线程间通信	(938)
6.2.4.11	System Verilog 的系统函数、任务语法	(960)
6.2.4.12	System Verilog 的面向对象编程语法	(969)
6.2.4.13	System Verilog 的随机化语法	(995)
6.2.5	Assertion	(1016)
6.2.5.1	Why Assertion?	(1017)
6.2.5.2	SVA 分类	(1017)
6.2.5.3	SVA expression	(1023)
6.2.5.4	SVA sequence	(1026)
6.2.5.5	SVA property	(1033)
6.2.5.6	SVA assert	(1035)
6.2.5.7	并发断言进阶	(1036)
6.3	仿真环境	(1048)
6.3.1	待测设计准备	(1049)
6.3.1.1	寻找所需的模块	(1049)
6.3.1.2	寻找所需的文件	(1050)
6.3.1.3	Clean 工程的概念	(1051)
6.3.2	仿真库文件准备	(1051)
6.3.2.1	建立 Xilinx 平台相关的仿真库	(1052)
6.3.2.2	建立 Altera 平台相关的仿真库	(1069)
6.3.3	仿真平台的结构设计	(1069)
6.3.4	仿真平台的变形	(1070)

6.3.4.1	直接观察波形法	(1071)
6.3.4.2	用多仿真平台替换任务生成模块	(1071)
6.3.4.3	用参考设计替换记分板模块	(1071)
6.3.4.4	添加逆变换模块	(1073)
6.3.4.5	第三方器件模拟	(1074)
6.4	仿真思路	(1075)
6.4.1	如何围绕 FPGA 项目展开仿真	(1075)
6.4.1.1	仿真介入的时机	(1075)
6.4.1.2	自底向上的仿真	(1076)
6.4.1.3	自顶向下的仿真	(1076)
6.4.1.4	回归测试	(1077)
6.4.2	仿真模块的实现	(1077)
6.4.3	测试用例的分类	(1078)
6.4.4	定位问题的手段浅析	(1079)
6.4.4.1	找出所有的问题	(1079)
6.4.4.2	如何具体进行问题排查	(1080)
6.4.4.3	如何加速问题的出现	(1081)
6.5	仿真覆盖率	(1081)
6.5.1	覆盖率浅析	(1081)
6.5.2	代码覆盖率简介	(1082)
6.5.2.1	代码覆盖率的种类	(1083)
6.5.2.2	代码覆盖率的示例	(1086)
6.5.3	功能覆盖率简介	(1096)
6.5.3.1	覆盖组简介	(1097)
6.5.3.2	覆盖组数据的收集规则	(1102)
6.5.3.3	覆盖组的选项	(1119)
6.5.3.4	在仿真中进行覆盖率统计	(1123)
6.5.4	覆盖率的合并	(1124)
7	时序分析	(1128)
7.1	基本概念介绍	(1128)
7.1.1	常用时间参数介绍	(1128)
7.1.2	线延迟与门延迟	(1129)
7.1.3	影响延迟的因素	(1130)
7.1.3.1	温度与电压	(1130)
7.1.3.2	三种工况	(1131)
7.1.4	时钟信号的偏差描述	(1132)
7.2	时序分析的原理	(1133)
7.2.1	一道时序分析的例题	(1133)

7.2.1.1	解答一:能否正确工作分析	(1134)
7.2.1.2	解答二:最大时钟速率分析	(1136)
7.2.1.3	延伸二:最小时钟速率?	(1137)
7.2.1.4	解答三:保持时间不足情形分析	(1138)
7.2.2	同步时序逻辑的分析原理	(1138)
7.2.2.1	逻辑锥的概念	(1138)
7.2.2.2	逻辑锥的划分	(1138)
7.2.2.3	逻辑锥的求解	(1139)
7.2.3	其他常见逻辑情形的应对方法	(1143)
7.2.3.1	内部时钟相关时序分析	(1143)
7.2.3.2	外部接口相关时序分析	(1151)
7.3	常用时序约束介绍	(1167)
7.3.1	时序环境约束	(1168)
7.3.2	分组时序约束	(1168)
7.3.3	常用时序约束	(1169)
7.3.2.1	周期约束	(1169)
7.3.2.2	输入约束	(1171)
7.3.2.3	组间约束	(1175)
7.3.2.4	输出约束	(1177)
7.4	根据时序报告修改设计	(1179)
7.4.1	时序分析报告示例	(1179)
7.4.1.1	待分析设计	(1179)
7.4.1.2	时序约束文件	(1180)
7.4.1.3	时序报告简介	(1181)
7.4.2	常见问题及修改	(1189)
7.4.2.1	可以不改的一些问题	(1189)
7.4.2.2	常见时序收敛的手段	(1189)
8	梦想启航	(1193)
8.1	你懂了吗?	(1193)
8.2	趁热打铁吧	(1193)
8.3	常回来看看	(1193)
8.4	共同成长	(1193)
8.5	祝你好运	(1194)
8.6	致谢	(1195)
参考文献		(1196)

1 与我同行

1.1 个人简介

笔者是陕西西安人,本科与研究生都就读于西安交通大学通信与信息系统专业,毕业后先后辗转于北京、上海、西安三地,期间一直从事 FPGA 相关项目的开发与验证工作。截止到写本书之日,笔者接触 FPGA 共有将近八年的时间。而在这八年的时间里,笔者通过不断的学习和项目的磨练,在 FPGA 技术方面也终有一定积累,拥有了比较丰富的项目经验,掌握了一定的学习和科研方法,也总结了不少技巧。当然,学海无涯,技术的发展也是无止境的,笔者在今后的时间里仍然会继续沿着 FPGA 的道路坚持下去!

1.2 “道”的含义

本书名为——《FPGA 之道》,这其中最关键的就是一个“道”字,它有三层含义:一、道路;二、方法;三、精神。这也是贯穿本书始终的三大主线。

道路,是本书的第一条主线。它指的是 FPGA 的技术之路,主要是指笔者从事 FPGA 事业的这八年所走过的道路,感受到的风景,体会到的滋味。其次,它指的也是此时此刻正在阅读本书的“你”的道路,作为本书的读者,想必你也一定是有着 FPGA 相关的项目要做,或者未来打算做 FPGA 相关的项目吧。那么,从无到有,从不懂到精通,从搭建软、硬件环境到成功拿下项目,这之间都是需要我们老老实实、一步一个脚印的沿着 FPGA 技术之路走下来。只要坚持不懈,走得越远,收获就会越大,万里长征路漫漫,请务必加油,大家共勉!

方法,是本书的第二条主线。它指的是 FPGA 的方法学,这其中包括学习 FPGA 的方法和使用 FPGA 的方法。虽然俗话说“勤能补拙”、“笨鸟先飞”,但是如果飞的方向错了,那么可能永远也到达不了目的地,并且悲观情绪的慢慢滋生很可能会令你丧失继续坚持下去的勇气。因此,光靠使蛮劲、耗时间是不一定能学好 FPGA 的,掌握必要的、科学的、可行的方法,是助你在 FPGA 技术之路上少走弯路的必要条件!

精神,是本书的第三条主线。它指的是一名沿着 FPGA 技术之路勇往直前的奋斗者的境界和觉悟!这是一种求知欲的表达和一份对 FPGA 技术的热爱!只有达到这样一种精神层面,才能在当前这个纷繁复杂、物欲横流的世界中找到属于自己的一片“净土”。而只有这片“净土”,才是能够让你静下心来、解放思想、发散思维、不断创新的源泉之地。因此,这也应该是每一个在 FPGA 道路上的长征的“你”所追求的境界!