



# Xilinx FPGA

## 开发实用教程

田耘 徐文波 编著



清华大学出版社



内容简介

本书以Xilinx公司的FPGA器件为对象,详细介绍了FPGA器件的结构、原理、开发方法和应用。全书共分10章,第1章介绍FPGA器件的发展概况;第2章介绍FPGA器件的内部结构;第3章介绍FPGA器件的开发流程;第4章介绍FPGA器件的开发工具;第5章介绍FPGA器件的开发语言;第6章介绍FPGA器件的开发方法;第7章介绍FPGA器件的开发应用;第8章介绍FPGA器件的开发案例;第9章介绍FPGA器件的开发实验;第10章介绍FPGA器件的开发总结。

# Xilinx FPGA

## 开发实用教程

田耘 徐文波 编著

清华大学出版社  
北京

## 内 容 简 介

本书系统讲述了 Xilinx FPGA 的开发知识,包括 FPGA 开发简介、Verilog HDL 语言基础、基于 Xilinx 芯片的 HDL 语言高级进阶、ISE 开发环境使用指南、FPGA 配置电路及软件操作、在线逻辑分析仪 ChipScope 的使用、基于 FPGA 的数字信号处理技术、基于 System Generator 的 DSP 系统开发技术、基于 FPGA 的可编程嵌入式开发技术、基于 FPGA 的高速数据连接技术和时序分析原理及时序分析器的使用 11 章内容,各章均以实例为基础,涵盖了 FPGA 开发的主要方面。

本书适合从事 Xilinx 系列 FPGA 设计和开发的工程师,以及相关专业的研究生和高年级本科生使用。

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

Xilinx FPGA 开发实用教程/田耘,徐文波编著. —北京:清华大学出版社,2008. 11  
ISBN 978-7-302-18425-6

I. X… II. ①田… ②徐… III. 可编程逻辑器件—系统开发—教材 IV. TP332.1

中国版本图书馆 CIP 数据核字(2008)第 129399 号

责任编辑:王一玲

责任校对:白 蕾

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:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:北京密云胶印厂

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260 印 张:39.25 字 数:946 千字

版 次:2008 年 11 月第 1 版 印 次:2008 年 11 月第 1 次印刷

印 数:1~3000

定 价:59.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:030298-01

赛灵思(Xilinx)公司作为可编程器件(PLD)的领导厂商,占有超过50%的市场份额,为客户提供可编程逻辑芯片(CPLD、FPGA和PROM)、软件设计工具、不同等级的知识产权核(IP Core)以及系统级的完整解决方案。

随着FPGA工艺和设计水平的不断提高,其在数字系统中所扮演的角色也从逻辑胶合者提升到处理核心。目前,赛灵思公司的FPGA涵盖了逻辑应用、数字信号处理以及嵌入式三大应用领域,例如Spartan-3A/AN/E系列FPGA采用90nm工艺,广泛应用在中低规模系统中,包括机器视觉、机顶盒、DCD播放器以及广泛的多媒体处理等;Virtex-4/5系列FPGA分别采用90nm、65nm工艺,主要面向高端应用,如高速互连网络、无线通信、宽带接入以及汽车工业。此外,赛灵思公司的Virtex-5系列FPGA是目前业界主要的65nm工艺可编程器件的提供商,占据了超过90%的市场份额。

基于Xilinx公司的领先技术,更多的工程师和研究人员加入到赛灵思FPGA的开发队伍中来。在过去一年中,Xilinx公司通过大学和开源社区Openhard,举办了Xilinx杯开源硬件创新大赛以及多个网络研讨会,帮助广大技术人员、在校的研究生和高年级本科生尽快掌握Xilinx FPGA的开发流程,但切入点都比较零散,对于大多数开发人员来讲缺少一本合适的系统级书籍。《Xilinx FPGA开发实用教程》一书弥补了上述不足。

整体而言,本书具有以下3个特色:首先,从逻辑设计、数字信号处理、嵌入式系统设计和高速连接功能等4个方面系统地介绍了赛灵思FPGA开发应用,条理清晰、思路明确,符合FPGA目前和未来的发展趋势;其次,较为详细地介绍了Verilog HDL语言和Xilinx FPGA的开发技巧,融入了作者的工程开发经验,对于初学者和工程开发人员来讲都具有较强的可读性;第三,全面介绍了赛灵思公司的ISE、System Generator以及EDK开发软件,内容完整性高。

所以本书是一本较为理想的工程工具书和大学的教辅书籍,我郑重地将其推荐给大家,希望通过本书的出版,使更多的读者掌握赛灵思FPGA的开发技能,更好地促进FPGA开发技术的普及和推广。

赛灵思公司(Xilinx, Inc.)中国区大学计划经理

谢凯年博士

2008年9月

# 前言

## FOREWORD

2007年10月份,作者有幸聆听了 Xilinx 公司全球 CTO Ivo Bolsens 先生在清华大学题为“FPGA: The future platform for transforming, transporting and computing”的演讲,感触颇深。Ivo 先生指出了 FPGA 的三大应用领域:数字处理中的信号变换、高速交换中的数据收发以及求解中的复杂计算。作者本人虽然已有多年的 FPGA 开发经验,但还是第一次听到如此精辟的总结,随即想到深入了解并推广这种实用且精辟的 FPGA 开发理念。考察了许久,我们发现市场上没有此类相关书籍,且已有书籍比较偏重于单一软件的操作或 HDL 语言的讲解,因此就萌生了编写一本书,从系统开发的角度,以软、硬件结合的方式来阐述先进的 FPGA 开发理念。于是经过半年的思索、查阅资料、和相关专家大量讨论以及反复修改,便有了这本书的诞生,以期起到抛砖引玉的作用。

Xilinx 公司是最早也是最大的 FPGA 生产商,其芯片设计技术、开发软件和相关解决方案在业界属于顶级水平,拥有广泛的客户群。本书主要讲述了 Xilinx FPGA 的开发知识,包括 FPGA 开发简介、Verilog HDL 语言基础、基于 Xilinx 芯片的 HDL 语言高级进阶、ISE 开发环境使用指南、FPGA 配置电路及软件操作、在线逻辑分析仪 ChipScope 的使用、基于 FPGA 的数字信号处理技术、基于 System Generator 的 DSP 系统开发技术、基于 FPGA 的可编程嵌入式开发技术、基于 FPGA 的高速数据连接技术和时序分析原理及时序分析器的使用共 11 章内容,各章均以实例为基础,涵盖了 FPGA 开发的主要方面。由于篇幅所限,我们没有在本书中给出一个完整的工程实例。为了弥补这一缺陷,我们 Xilinx FPGA 开源社区 Openhard 网站中附带了本书所有的实例,期望本书能够帮助提高读者的工程开发能力。

全书各章由田耘、徐文波完成,孙霏菲参与了第 7 章的编写工作。此外,在成文过程中,我们参考了较多的书籍、论文和网络文献,向其作者表示深深的谢意。Xilinx 公司中国区大学计划经理谢凯年博士在百忙之中为本书作序,并提供了硬件实验环境;Xilinx 公司亚太区公共关系经理张俊伟女士一直关心、鼓励作者,并最终促成本书成稿。与非网科技的贺潇荃先生、陶丹博士等在成书过程中给予了我们诸多良好的建议和帮助;清华大学出版社的王一玲编辑为本书的修改付出了许多劳动,并给出许多中肯的修改意见,感谢他们为本书所做的贡献。

本书适合从事 Xilinx 系列 FPGA 设计和开发的工程师,以及相关专业的研究生和高级本科生使用。毫无疑问,市场上已经有很多关于 FPGA 设计的书籍,我们也不认为本书是其中最重要的一本,但我们意识到,FPGA 开发一定要结合芯片特点以及提供商的诸多建议和协议,只有这样才能真正掌握其开发之道。

书中的全部内容都是实际项目硬件和 Xilinx 公司各类文档、书籍的结合体,全部信息

几乎都可以从 Xilinx 网站以及 Google 上找到渊源,不过我们仍然向您推荐本书,因为网络的信息是分散的、杂乱的,且正确性不是 100%的,本书各章内容的安排是从大量的实践中总结出来的,循序渐进,条理清楚,且都经过作者验证。我们的目的就是 from Ivo Bolsens 先生的观点出发,结合项目开发,将网络上尽可能多的相关信息以相对较高的质量组合起来。

FPGA 技术博大精深且发展迅猛,不可能通过一本书进行全方位的详细介绍,更多的还需要读者自己动手实践。由于作者水平有限,加上时间比较仓促,书中不妥之处,敬请指正。在本书出版后,作者将继续在 Openhard 社区中维护书籍内容,进行修正和补充,详细网址为: <http://www.openhw.org/html/08-05/415531070314nup7.html>。

作者

2008 年 5 月

注:限于篇幅,本书部分程序的 RTL 级综合结果示意图不能全幅度显示,因而不清楚。读者可在本书配套网站上查看。

# 目录

## CONTENTS

<b>第 1 章</b>	<b>FPGA 开发简介</b>	1
1.1	可编程逻辑器件基础	1
1.1.1	可编程逻辑器件概述	1
1.1.2	可编程逻辑器件的发展历史	2
1.1.3	PLD 开发工具	2
1.2	FPGA 芯片结构	3
1.2.1	FPGA 工作原理与简介	3
1.2.2	FPGA 芯片结构	4
1.2.3	软核、硬核以及固核的概念	8
1.3	基于 FPGA 的开发流程	9
1.3.1	FPGA 设计方法概论	9
1.3.2	典型 FPGA 开发流程	10
1.3.3	基于 FPGA 的 SOC 设计方法	13
1.4	Xilinx 公司主流可编程逻辑器件简介	13
1.4.1	Xilinx FPGA 芯片介绍	14
1.4.2	Xilinx PROM 芯片介绍	21
1.5	本章小结	23
<b>第 2 章</b>	<b>Verilog HDL 语言基础</b>	24
2.1	Verilog HDL 语言简介	24
2.1.1	Verilog HDL 语言的历史	25
2.1.2	Verilog HDL 的主要能力	25
2.1.3	Verilog HDL 和 VHDL 的区别	26
2.1.4	Verilog HDL 设计方法	26
2.2	Verilog HDL 基本程序结构	27
2.3	Verilog HDL 语言的数据类型和运算符	28
2.3.1	标志符	29
2.3.2	数据类型	29
2.3.3	模块端口	31
2.3.4	常量集合	31



2.3.5	运算符和表达式 .....	32
2.4	Verilog HDL 语言的描述语句 .....	37
2.4.1	结构描述形式 .....	37
2.4.2	数据流描述形式 .....	38
2.4.3	行为描述形式 .....	38
2.4.4	混合设计模式 .....	46
2.5	Verilog 代码书写规范 .....	46
2.5.1	信号命名规则 .....	46
2.5.2	模块命名规则 .....	47
2.5.3	代码格式规范 .....	48
2.5.4	模块调用规范 .....	50
2.6	Verilog 常用程序示例 .....	50
2.6.1	Verilog 基本模块 .....	50
2.6.2	基本时序处理模块 .....	56
2.6.3	常用数字处理算法的 Verilog 实现 .....	62
2.7	本章小结 .....	83
<b>第 3 章 基于 Xilinx 芯片的 HDL 语言高级进阶 .....</b>		<b>84</b>
3.1	面向硬件电路的设计思维 .....	84
3.1.1	面向硬件的程序设计思维 .....	84
3.1.2	“面积”和“速度”的转换原则 .....	89
3.1.3	同步电路的设计原则 .....	90
3.1.4	模块划分的设计原则 .....	93
3.2	优秀的 HDL 代码风格 .....	94
3.2.1	代码风格的含义 .....	94
3.2.2	通用代码风格的介绍 .....	95
3.2.3	专用代码风格的简要说明 .....	103
3.3	Verilog 建模与调试技巧 .....	108
3.3.1	双向端口的使用和仿真 .....	108
3.3.2	阻塞赋值与非阻塞赋值 .....	111
3.3.3	输入值不确定的组合逻辑电路 .....	113
3.3.4	数学运算中的扩位与截位操作 .....	113
3.3.5	利用块 RAM 来实现数据延迟 .....	115
3.3.6	测试向量的生成 .....	118
3.4	Xilinx 公司原语的使用方法 .....	119
3.4.1	计算组件 .....	119
3.4.2	时钟组件 .....	121
3.4.3	配置和检测组件 .....	126
3.4.4	吉比特收发器组件 .....	128



3.4.5	I/O 端口组件	128
3.4.6	处理器组件	134
3.4.7	RAM/ROM 组件	134
3.4.8	寄存器和锁存器	139
3.4.9	移位寄存器组件	140
3.4.10	Slice/CLB 组件	141
3.5	本章小结	143
<b>第 4 章</b>	<b>ISE 开发环境使用指南</b>	<b>144</b>
4.1	ISE 套件的介绍与安装	144
4.1.1	ISE 简要介绍	144
4.1.2	ISE 功能简介	144
4.1.3	ISE 软件的安装	145
4.1.4	ISE 软件的基本操作	148
4.2	基于 ISE 的代码输入	153
4.2.1	新建工程	153
4.2.2	代码输入	154
4.2.3	代码模板的使用	155
4.2.4	Xilinx IP Core 的使用	157
4.3	基于 ISE 的开发流程	164
4.3.1	基于 Xilinx XST 的综合	164
4.3.2	基于 ISE 的仿真	169
4.3.3	基于 ISE 的实现	174
4.3.4	基于 ISE 的芯片编程	180
4.3.5	功耗分析以及 XPower 的使用	183
4.4	约束文件的编写	193
4.4.1	约束文件的基本操作	193
4.4.2	UCF 文件的语法说明	195
4.4.3	管脚和区域约束语法	196
4.4.4	管脚和区域约束编辑器 PACE	198
4.5	ISE 与第三方软件	204
4.5.1	Synplify Pro 软件的使用	204
4.5.2	ModelSim 软件的使用	212
4.5.3	Synplify Pro、ModelSim 和 ISE 的联合开发流程	216
4.5.4	ISE 与 MATLAB 的联合使用	217
4.6	Xilinx FPGA 芯片底层单元的使用	219
4.6.1	Xilinx 全局时钟网络的使用	220
4.6.2	DCM 模块的使用	221
4.6.3	Xilinx 内嵌块存储器的使用	227

4.6.4	硬核乘加器的使用 .....	232
4.7	本章小结 .....	240
<b>第5章</b>	<b>FPGA 配置电路及软件操作 .....</b>	<b>241</b>
5.1	FPGA 配置电路综述 .....	241
5.1.1	Xilinx FPGA 配置电路综述 .....	241
5.1.2	Xilinx FPGA 常用的配置管脚 .....	243
5.1.3	Xilinx FPGA 配置电路分类 .....	243
5.2	JTAG 电路的原理与设计 .....	245
5.2.1	JTAG 电路的工作原理 .....	245
5.2.2	Xilinx JTAG 下载线 .....	248
5.3	FPGA 的常用配置电路 .....	250
5.3.1	主串模式——最常用的 FPGA 配置模式 .....	251
5.3.2	SPI 串行 Flash 配置模式 .....	257
5.3.3	从串配置模式 .....	262
5.3.4	字节宽度外部接口并行配置模式 .....	265
5.3.5	JTAG 配置模式 .....	270
5.3.6	System ACE 配置方案 .....	273
5.4	iMPACT 软件使用 .....	277
5.4.1	iMPACT 综述与基本操作 .....	278
5.4.2	使用 iMPACT 创建配置文件 .....	280
5.4.3	使用 iMPACT 配置芯片 .....	289
5.4.4	FPGA 配置失败的常见问题 .....	289
5.5	从配置 PROM 中读取用户数据 .....	290
5.5.1	从 PROM 中引导数据简介 .....	290
5.5.2	硬件电路设计方法 .....	291
5.5.3	软件操作流程 .....	293
5.6	本章小结 .....	294
<b>第6章</b>	<b>在线逻辑分析仪 ChipScope 的使用 .....</b>	<b>295</b>
6.1	ChipScope 介绍 .....	295
6.1.1	ChipScope Pro 简介 .....	295
6.1.2	ChipScope Pro 软件的安装 .....	297
6.1.3	ChipScope Pro 的使用流程 .....	298
6.2	ChipScope Core Generator 使用说明 .....	299
6.2.1	ChipScope Pro 核的基本介绍 .....	299
6.2.2	ChipScope 核的生成流程 .....	301
6.3	ChipScope Core Inserter 使用说明 .....	305
6.3.1	Core Inserter 的用户界面 .....	305

6.3.2	Core Inserter 的基本操作 .....	306
6.4	ChipScope Pro Analyzer 使用说明 .....	311
6.4.1	ChipScope 分析仪的用户界面 .....	311
6.4.2	ChipScope Analyzer 的基本操作 .....	312
6.5	在 ISE 中直接调用 ChipScope 的应用实例 .....	314
6.5.1	在工程中添加 ChipScope Pro 文件 .....	315
6.5.2	在 ChipScope Pro 中完成下载和观察 .....	315
6.6	本章小结 .....	316
<b>第 7 章</b>	<b>基于 FPGA 的数字信号处理技术 .....</b>	<b>317</b>
7.1	数字信号概述 .....	317
7.1.1	数字信号的产生 .....	317
7.1.2	采样定理 .....	318
7.1.3	数字系统的主要性能指标 .....	319
7.2	离散傅里叶变换基础 .....	319
7.2.1	离散傅里叶变换 .....	319
7.2.2	频域应用 .....	320
7.2.3	FFT/IFFT IP Core 的使用 .....	322
7.3	XtremeDSP 模块功能介绍 .....	325
7.4	乘累加结构的 FIR 滤波器 .....	326
7.4.1	单乘法器 MAC FIR 滤波器 .....	326
7.4.2	对称 MAC FIR 滤波器 .....	330
7.4.3	MAC FIR 滤波器 IP Core 的使用 .....	334
7.5	半并行/并行 FIR 滤波器 .....	338
7.5.1	并行 FIR 滤波器 .....	338
7.5.2	半并行 FIR 滤波器 .....	339
7.5.3	FIR Compiler IP Core 的使用 .....	340
7.6	多通道 FIR 滤波器 .....	344
7.6.1	滤波器组的基本概念 .....	344
7.6.2	多通道 FIR 滤波器的基本原理 .....	345
7.6.3	多通道 FIR 滤波器组的 FPGA 实现 .....	346
7.7	本章小结 .....	349
<b>第 8 章</b>	<b>基于 System Generator 的 DSP 系统开发技术 .....</b>	<b>350</b>
8.1	System Generator 的简介与安装 .....	350
8.1.1	System Generator 简介 .....	350
8.1.2	System Generator 的主要特征 .....	351
8.1.3	System Generator 软件的安装和配置 .....	352
8.2	System Generator 入门基础 .....	354

8.2.1	System Generator 开发流程简介	354
8.2.2	Simulink 基础	356
8.2.3	AccelDSP 软件工具	358
8.3	基于 System Generator 的 DSP 系统设计	359
8.3.1	System Generator 快速入门	359
8.3.2	System Generator 中的信号类型	368
8.3.3	自动代码生成	369
8.3.4	编译 MATLAB 设计生成 FPGA 代码	370
8.3.5	子系统的建立和使用	373
8.4	基于 System Generator 的硬件协仿真	380
8.4.1	硬件协仿真平台的介绍与平台安装	380
8.4.2	硬件协仿真的基本操作	381
8.4.3	共享存储器的操作	385
8.5	System Generator 的高级应用	387
8.5.1	导入外部的 HDL 程序模块	387
8.5.2	设计在线调试	392
8.5.3	系统中的多时钟设计	394
8.5.4	软、硬件联合开发	397
8.5.5	FPGA 设计的高级技巧	399
8.5.6	设计资源评估	403
8.6	开发实例: 基于 FIR 滤波器的协仿真实例	403
8.7	本章小结	407
<b>第 9 章</b>	<b>基于 FPGA 的可编程嵌入式开发技术</b>	<b>408</b>
9.1	可编程嵌入式系统(EDK)介绍	408
9.1.1	基于 FPGA 的可编程嵌入式开发系统	408
9.1.2	Xilinx 公司的解决方案	409
9.2	Xilinx 嵌入式开发系统组成介绍	409
9.2.1	片内微处理器软核 MicroBlaze	410
9.2.2	片内微处理器 PowerPC	413
9.2.3	常用的 IP 核以及设备驱动	415
9.2.4	系统设计方案	424
9.3	EDK 软件基本介绍	426
9.3.1	EDK 的介绍与安装	427
9.3.2	EDK 设计的实现流程	429
9.3.3	EDK 的文件管理架构	431
9.4	XPS 软件的基本操作	435
9.4.1	XPS 的启动	435
9.4.2	利用 BSB 创建新工程	436

9.4.3	XPS 的用户界面 .....	441
9.4.4	XPS 的目录结构与硬件平台 .....	448
9.4.5	在 XPS 加入 IP Core .....	450
9.4.6	在 XPS 中定制用户设备的 IP .....	452
9.4.7	XPS 中 IP Core API 函数的查阅和使用方法 .....	468
9.5	XPS 软件的高级操作 .....	469
9.5.1	XPS 的软件输入 .....	469
9.5.2	XPS 中的设计仿真 .....	473
9.5.3	将 EDK 设计作为 ISE 设计的子系统 .....	481
9.5.4	XPS 对嵌入式操作系统的支持 .....	485
9.5.5	XPS 工程的实现和下载 .....	485
9.5.6	在线调试工具 XMD 的使用 .....	490
9.5.7	XPS 中 ChipScope 的使用 .....	496
9.5.8	软件平台 SDK 的使用 .....	504
9.6	EDK 开发实例——DDR SDRAM 接口控制器 .....	510
9.6.1	DDR SDRAM 工作原理 .....	510
9.6.2	DDR SDRAM 控制器的 EDK 实现 .....	510
9.6.3	DDR SDRAM 控制器的调试 .....	520
9.7	本章小结 .....	521
<b>第 10 章</b>	<b>基于 FPGA 的高速数据连接技术 .....</b>	<b>522</b>
10.1	高速数据连接功能简介 .....	522
10.1.1	高速数据传输的背景 .....	522
10.1.2	Xilinx 公司高速连接功能的解决方案 .....	523
10.2	实现吉比特高速串行 I/O 的相关技术 .....	523
10.2.1	吉比特高速串行 I/O 的特点和应用 .....	523
10.2.2	吉比特串行 I/O 系统的组成 .....	525
10.2.3	吉比特串行 I/O 的设计要点 .....	528
10.3	基于 Rocket I/O 高速串行技术 .....	530
10.3.1	Rocket I/O 技术简介 .....	530
10.3.2	Aurora 协议 .....	531
10.3.3	Rocket I/O 硬核模块的体系结构 .....	532
10.3.4	Rocket I/O 的时钟设计方案 .....	544
10.3.5	Rocket I/O 的开发要素 .....	548
10.3.6	Rocket I/O IP Core 的使用 .....	553
10.4	基于 Xilinx FPGA 的千兆以太网控制器的开发 .....	555
10.4.1	千兆以太网技术 .....	555
10.4.2	基于 FPGA 的千兆以太网 MAC 控制器实现方案 .....	556
10.4.3	Xilinx 千兆以太网 MAC IP Core .....	561

10.5	本章小结 .....	565
<b>第 11 章</b>	<b>时序分析原理以及时序分析器的使用 .....</b>	<b>566</b>
11.1	时序分析的作用和原理 .....	566
11.1.1	时序分析的作用 .....	566
11.1.2	静态时序分析原理 .....	567
11.1.3	时序分析的基础知识 .....	568
11.2	Xilinx FPGA 中的时钟资源 .....	573
11.2.1	全局时钟资源 .....	574
11.2.2	第二全局时钟资源 .....	577
11.3	时序约束 .....	578
11.3.1	使用约束文件添加时序约束 .....	578
11.3.2	使用约束编辑器添加时序约束 .....	582
11.4	ISE 时序分析器 .....	590
11.4.1	时序分析器简介 .....	591
11.4.2	时序分析器的文件类型 .....	591
11.4.3	时序分析器的调用与用户界面 .....	592
11.4.4	时序分析器的基本使用方法 .....	599
11.4.5	提高时序性能的手段 .....	602
11.5	本章小结 .....	606
<b>缩略语</b>	.....	<b>607</b>
<b>参考文献</b>	.....	<b>610</b>

FPGA(Field Programmable Gate Array)即现场可编程门阵列,属于可编程逻辑器件的一种,在 20 世纪 90 年代获得突飞猛进的发展。经过近 20 年的发展,到目前它已成为实现数字系统的主流平台之一。本章主要介绍 FPGA 的起源、发展历史、芯片结构、工作原理、开发流程以及 Xilinx 公司的主要可编程芯片,为读者提供 FPGA 系统设计的基础知识。

## 1.1 可编程逻辑器件基础

### 1.1.1 可编程逻辑器件概述

可编程逻辑器件(Programmable Logic Device,PLD)起源于 20 世纪 70 年代,是在专用集成电路(ASIC)的基础上发展起来的一种新型逻辑器件,是当今数字系统设计的主要硬件平台,其主要特点就是完全由用户通过软件进行配置和编程,从而完成某种特定的功能,且可以反复擦写。在修改和升级 PLD 时,不需额外地改变 PCB 电路板,只是在计算机上修改和更新程序,使硬件设计工作成为软件开发工作,缩短了系统设计的周期,提高了实现的灵活性并降低了成本,因此获得了广大硬件工程师的青睐,形成了巨大的 PLD 产业规模。

目前常见的 PLD 产品有编程只读存储器(Programmable Read Only Memory, PROM)、现场可编程逻辑阵列(Field Programmable Logic Array,FPLA)、可编程阵列逻辑(Programmable Array Logic,PAL)、通用阵列逻辑(Generic Array Logic,GAL)、可擦除的可编程逻辑阵列(Erasable Programmable Logic Array,EPLA)、复杂可编程逻辑器件(Complex Programmable Logic Device,CPLD)和现场可编程门阵列等类型。PLD 器件从规模上又可以细分为简单 PLD(SPLD)、复杂 PLD(CPLD)以及 FPGA。它们内部结构的实现方法各不相同。

可编程逻辑器件按照颗粒度可以分为 3 类:①小颗粒度(如“门海(sea of gates)”架构);②中等颗粒度(如 FPGA);③大颗粒度(如 CPLD)。按照编程工艺可以分为 4 类:①熔丝(Fuse)和反熔丝(Antifuse)编程器件;②可擦除的可编程只读存储器(UEPROM)编程器件;③电信号可擦除的可编程只读存储器(EEPROM)编程器件(如 CPLD);④SRAM 编程器件(如 FPGA)。在工艺分类中,前 3 类为非易失性器件,编程后,配置数据保留在器件上;第 4 类为易失性器件,掉电后,配置数据会丢失,因此在每次上电后需要重新进行数据配置。



## 1.1.2 可编程逻辑器件的发展历史

可编程逻辑器件的发展可以划分为4个阶段,即从20世纪70年代初到70年代中为第1阶段,20世纪70年代中到80年代中为第2阶段,20世纪80年代中到90年代末为第3阶段,20世纪90年代末到目前为第4阶段。

第1阶段的可编程器件只有简单的可编程只读存储器(PROM)、紫外线可擦除只读存储器(EPROM)和电可擦只读存储器(EEPROM)3种。由于结构的限制,它们只能完成简单的数字逻辑功能。

第2阶段出现了结构上稍微复杂的可编程阵列逻辑(PAL)和通用阵列逻辑(GAL)器件,正式被称为PLD,能够完成各种逻辑运算功能。典型的PLD由“与”、“非”阵列组成,用“与或”表达式来实现任意组合逻辑,所以PLD能以乘积和形式完成大量的逻辑组合。

在第3阶段,Xilinx和Altera公司分别推出了与标准门阵列类似的FPGA以及类似于PAL结构的扩展性CPLD。它们提高了逻辑运算的速度,具有体系结构和逻辑单元灵活、集成度高以及适用范围宽等特点,兼容了PLD和通用门阵列的优点,能够实现超大规模的电路,编程方式也很灵活,成为产品原型设计和中小规模(一般小于10 000)产品生产的首选。在这一阶段,CPLD、FPGA器件在制造工艺和产品性能方面都获得长足的发展,达到了 $0.18\mu\text{m}$ 工艺和系统门数百万门的规模。

第4阶段出现了SOPC(System On Programmable Chip,可编程的片上系统)和SOC(System On Chip,片上系统)技术。它们是PLD和ASIC技术融合的结果,涵盖了实时化数字信号处理技术、高速数据收发器、复杂计算以及嵌入式系统设计技术的全部内容。Xilinx和Altera公司也推出了相应的SOC FPGA产品,制造工艺已达到 $65\mu\text{m}$ ,系统门数也超过百万门。并且,这一阶段的逻辑器件内嵌了硬核高速乘法器、吉比特差分串行接口、时钟频率高达500MHz的PowerPC微处理器、软核MicroBlaze、PicoBlaze、Nios以及Nios II,这不仅实现了软件需求和硬件设计的完美结合,还实现了高速与灵活性的完美结合,使其已超越了ASIC器件的性能和规模,也超越了传统意义上FPGA的概念,使PLD的应用范围从单片扩展到系统级。目前,基于PLD片上可编程的概念仍在不断地向前发展。

## 1.1.3 PLD 开发工具

基于高复杂度PLD器件的开发,在很大程度上要依靠电子设计自动化(EDA)来完成。PLD的EDA工具以计算机软件为主,将典型的单元电路封装起来形成固定模块并形成标准的硬件开发语言(如HDL语言)供设计人员使用。设计人员考虑如何将可组装的软件库和软件包搭建出满足需求的功能模块甚至完整的系统。PLD开发软件需要自动地完成逻辑编译、化简、分割、综合及优化、布局布线、仿真以及对于特定目标芯片的适配编译和编程下载等工作。典型的EDA工具中必须包含两个特殊的软件包,即综合器和适配器。综合器的功能就是将设计者在EDA平台上完成的针对某个系统项目的HDL、原理图或状态图形描述,针对给定的硬件系统组件,进行编译、优化、转换和综合。适配器一般不用于FPGA设计。

随着开发规模的级数性增长,必须减短 PLD 开发软件的编译时间,提高其编译性能,并提供丰富的知识产权(IP)核资源供设计人员调用。此外,PLD 开发界面的友好性以及操作的复杂程度也是评价其性能的重要因素。目前在 PLD 产业领域中,各个芯片提供商的 PLD 开发工具已成为影响其成败的核心成分。只有全面做到芯片技术领先、文档完整和 PLD 开发软件优秀,芯片提供商才能获得客户的认可。一个完美的 PLD 开发软件应当具备下面 5 点:

- 准确地将用户设计转换为电路模块;
- 能够高效地利用器件资源;
- 能够快速地完成编译和综合;
- 提供丰富的 IP 核资源;
- 用户界面友好,操作简单。

Xilinx 公司的 ISE、Altera 公司的 Quartus II 和 Maxplus II 是业界公认的优秀集成 PLD 开发软件。此外,综合软件 Synplify 和仿真软件 ModelSim 等诸多第三方 EDK 开发软件也满足上述要求。

## 1.2 FPGA 芯片结构

### 1.2.1 FPGA 工作原理与简介

如前所述,FPGA 是在 PAL、GAL、EPLD、CPLD 等可编程器件的基础上进一步发展的产物。它是作为 ASIC 领域中的一种半定制电路而出现的,既解决了定制电路的不足,又克服了原有可编程器件门电路有限的缺点。

由于 FPGA 需要被反复烧写,它实现组合逻辑的基本结构不可能像 ASIC 那样通过固定的与非门来完成,而只能采用一种易于反复配置的结构,查找表可以很好地满足这一要求。目前,主流 FPGA 都采用了基于 SRAM 工艺的查找表结构,也有一些军品和宇航级 FPGA 采用 Flash 或者熔丝与反熔丝工艺的查找表结构。可通过烧写文件改变查找表内容的方法来实现对 FPGA 的重复配置。

根据数字电路的基本知识可以知道,对于一个  $n$  输入的逻辑运算,不管是与或非运算还是异或运算,最多只可能存在  $2^n$  种结果。所以,如果事先将相应的结果存放于一个存储单元中,就相当于实现了与非门电路的功能。FPGA 的原理也是如此,它通过烧写文件去配置查找表的内容,从而在相同的电路情况下实现了不同的逻辑功能。

查找表(Look-Up-Table)简称为 LUT,LUT 本质上就是一个 RAM。目前,FPGA 中多使用 4 输入的 LUT,所以每一个 LUT 可以看成是一个有 4 位地址线的  $16 \times 1$  的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后,PLD/FPGA 开发软件会自动计算逻辑电路的所有可能结果,并把真值表(即结果)事先写入 RAM。这样,每输入一个信号进行逻辑运算,就等于输入一个地址进行查表,找出地址对应的内容后输出即可。

下面给出一个 4 输入与门电路的例子来说明 LUT 实现逻辑功能的原理。