


电子系统EDA新技术丛书

Xilinx FPGA

权威设计指南

—— Vivado 2014集成开发环境

何 宾◎编著



将最新的Xilinx设计理论贯穿于具体的设计实现中，理论与应用并重



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

电子系统 EDA 新技术丛书

Xilinx FPGA 权威设计指南

——Vivado 2014 集成开发环境

何 宾 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书全面系统介绍了 Xilinx 新一代集成开发环境 Vivado 2014.3 的设计方法、设计流程和具体实现。全书共 11 章, 内容包括: Xilinx UltraScale 结构、Vivado 集成设计环境导论、Vivado 工程模式基本设计实现、Vivado 非工程模式基本设计实现、创建和封装用户 IP 核流程、Vivado 高级约束原理及实现、Vivado 调试工具原理及实现、Vivado 嵌入式系统设计实现、Vivado 模型设计原理及实现、Vivado HLS 原理及实现、Vivado 部分可重配置原理及实现。本书参考了 Xilinx 提供的大量 Vivado 最新设计资料, 理论与应用并重, 将 Xilinx 最新的设计理论贯穿在具体的设计实现中。

本书作为使用 Xilinx 集成开发环境 Vivado 进行 FPGA 设计的工程技术人员的参考书, 也可作为电子信息类专业高年级本科生和研究生的教学和科研用书, 还可作为 Xilinx 公司 Vivado 相关培训的培训教材及工程技术人员的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

Xilinx FPGA 权威设计指南: Vivado 2014 集成开发环境 / 何宾编著. —北京: 电子工业出版社, 2015.2
(电子系统 EDA 新技术丛书)

ISBN 978-7-121-25400-0

I. ①X… II. ①何… III. ①可编程序逻辑器件-系统设计-指南 IV. ①TP332.1-62

中国版本图书馆 CIP 数据核字 (2015) 第 009749 号

责任编辑: 王敬栋 (wangjd@phei.com.cn) 文字编辑: 张 迪

印 刷: 北京京师印务有限公司

装 订: 北京京师印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 29 字数: 742.4 千字

版 次: 2015 年 2 月第 1 版

印 次: 2015 年 2 月第 1 次印刷

印 数: 3 000 册 定价: 79.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

全球知名的可编程逻辑元器件生产厂商——美国 Xilinx 公司，于 2012 年发布了新一代 Vivado 集成开发环境，使得新一代 FPGA 的设计环境和设计方法发生了重大变化。2014 年年初，Xilinx 新一代 UltraScale 结构的 FPGA 也进入量产阶段。这些都标志着未来在高性能数据处理方面，FPGA 将发挥越来越重要的作用。

Xilinx 新一代集成开发环境 Vivado 突出基于知识产权 (Intellectual Property, IP) 核的设计方法，更加体现系统级设计的思想，进一步增强了设计者对 FPGA 底层布局和布线的干预能力，以及允许设计者通过选择不同的设计策略，对不同的实现方法进行探索，从中找到最佳的解决方案。这些设计思想和设计方法，大大提高了 FPGA 的设计效率。

本书是在《Xilinx FPGA 设计权威指南——Vivado 集成设计环境》(清华大学出版社，2014) 基础上，针对读者提出的意见和建议，对原书进行了大幅度修订。该书从原来的 Vivado 2013.3 集成开发环境升级到 Vivado 2014.3 集成开发环境，并增加了 IP 核设计的内容。此外，还大幅度增加了对高级约束内容的讲解，并对原书章节的顺序进行了调整，以方便读者的学习。全书从原来的 8 章增加到 11 章。内容包括：Xilinx UltraScale 结构、Vivado 集成设计环境导论、Vivado 工程模式基本设计实现、Vivado 非工程模式基本设计实现、创建和封装用户 IP 核流程、Vivado 高级约束原理及实现、Vivado 调试工具原理及实现、Vivado 嵌入式系统设计实现、Vivado 模型设计原理及实现、Vivado HLS 原理及实现、Vivado 部分可重配置原理及实现。每章内容要点如下：

(1) 第 1 章主要介绍了 UltraScale 结构的特点，以及芯片内各个模块的原理和功能。

(2) 第 2 章主要介绍了 Vivado 系统级设计流程、Vivado 功能和特性、Vivado 中电路结构的网表描述、Vivado 中工程数据的目录结构、Vivado 中 Journal 文件和 Log 文件功能、Vivado 两种设计流程模式、Vivado 中 XDC 文件、Vivado 集成开发环境的启动方法、Vivado 集成设计环境主界面、Vivado 设计主界面及功能。

(3) 第 3 章主要介绍了创建新的设计工程、创建并添加一个新的设计文件、RTL 详细描述和分析、设计综合和分析、设计行为级仿真、创建实现约束、设计实现和分析、设计时序仿真、生成编程文件、下载比特流文件到 FPGA。

(4) 第 4 章主要介绍了非工程模式基本命令和功能、Vivado 集成开发环境分析设计、修改设计路径、设置设计输出路径、设置设计输出路径、读取设计文件、运行设计综合、运行设计布线、生成比特流文件。

(5) 第 5 章主要介绍了 Vivado 定制 IP 流程、创建新的用于创建 IP 的工程、设置定制 IP 的库名和目录、封装定制 IP 的实现、创建新的用于调用 IP 的工程、设置包含调用 IP 的路径、创建基于 IP 的系统、系统行为级仿真、系统设计综合、系统实现和验证。

(6) 第 6 章主要介绍了时序检查概念、时序约束概念、生成时序报告、添加时序约束、

物理约束原理、布局约束实现、布线约束实现、修改逻辑实现、配置约束原理、增量编译。

(7) 第 7 章主要介绍了设计调试原理和方法、创建新的 FIFO 调试工程、添加 FIFO IP 到设计中、添加顶层设计文件、使用 HDL 例化添加 FIFO 到设计中、添加约束文件、网表插入调试探测流程方法及实现、使用添加 HDL 属性调试探测流程、使用 HDL 例化调试核调试探测流程。

(8) 第 8 章主要介绍了简单硬件系统设计、在 PL 内添加外设、创建和添加定制 IP、编写软件程序、软件控制定时器和调试、使用硬件分析仪调试。

(9) 第 9 章主要介绍了 FPGA 信号处理方法、FPGA 模型设计模块、System Generator 运行环境的配置、信号模型的构建和实现、编译 MATLAB 到 FPGA、FIR 滤波器的设计与实现。

(10) 第 10 章主要介绍了高级综合工具概述、高级综合工具调度和绑定、Vivado HLS 工具的优势、C 代码的关键属性、时钟测量术语说明、HLS 关键优化策略、Vivado HLS 数字系统实现。

(11) 第 11 章主要介绍了可重配置导论、可重配置的实现。

参加本书编写的人员有李宝隆、张艳辉、彭勃、王纲领、常晓磊、王瑜、刘一民和何军。其中，李宝隆参与第 3 章和第 6 章设计实例的验证，张艳辉参与第 7 章设计实例的验证。全书由何宾统稿和定稿。

本书的编写得到 Xilinx 大中华区大学计划经理谢凯年先生和 Xilinx 亚太区传媒总监张俊伟女士的大力支持和帮助，以及美国 Digilent 公司的大力支持和帮助，他们为本书的编写提供了大量的资料和硬件设计平台。正是由于他们的无私帮助和鼎力支持，才能使作者顺利完成本书的编写工作。同时，也要感谢电子工业出版社的编辑和相关工作人员，他们的辛勤工作保证了本书的高质量出版。

由于 FPGA 技术发展迅速，作者水平有限，书中难免会有疏漏之处，欢迎读者批评指正。

作者

2014 年 11 月于北京



反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为以及歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，本社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市海淀区万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

| | | |
|--------------|---------------------------------------|----|
| 第 1 章 | Xilinx 新一代 UltraScale 结构 | 1 |
| 1.1 | UltraScale 结构特点..... | 1 |
| 1.2 | 可配置逻辑块..... | 2 |
| 1.2.1 | 可配置逻辑块的特点..... | 2 |
| 1.2.2 | 多路复用器..... | 3 |
| 1.2.3 | 进位逻辑..... | 5 |
| 1.2.5 | 分布式 RAM (只有 SLICEM)..... | 7 |
| 1.2.6 | 只读存储器 (ROM)..... | 8 |
| 1.2.7 | 移位寄存器 (只有 SLICEM)..... | 9 |
| 1.3 | 时钟资源和时钟管理模块..... | 10 |
| 1.3.1 | 时钟资源..... | 10 |
| 1.3.2 | 时钟管理模块..... | 12 |
| 1.4 | 块存储器资源..... | 13 |
| 1.5 | 专用的 DSP 模块..... | 15 |
| 1.6 | 输入/输出块..... | 16 |
| 1.7 | 高速串行收发器..... | 17 |
| 1.8 | PCI-E 模块..... | 18 |
| 1.9 | Interlaken 集成块..... | 19 |
| 1.10 | Ethernet 模块..... | 19 |
| 1.11 | 系统监控器模块..... | 19 |
| 1.12 | 配置模块..... | 20 |
| 1.13 | 互联资源..... | 20 |
| 第 2 章 | Vivado 集成设计环境导论 | 22 |
| 2.1 | Vivado 系统级设计流程..... | 22 |
| 2.2 | Vivado 功能和特性..... | 24 |
| 2.3 | Vivado 中电路结构的网表描述..... | 25 |
| 2.4 | Vivado 中工程数据的目录结构..... | 26 |
| 2.5 | Vivado 中 Journal 文件和 Log 文件功能..... | 26 |
| 2.5.1 | Journal 文件 (Vivado.jou)..... | 26 |
| 2.5.2 | Log 文件 (Vivado.log)..... | 27 |



| | | |
|--------------|--------------------------|-----------|
| 2.6 | Vivado 两种设计流程模式 | 28 |
| 2.6.1 | 工程模式和非工程模式不同点比较 | 28 |
| 2.6.2 | 工程模式和非工程模式命令的不同 | 29 |
| 2.7 | Vivado 中 XDC 文件 | 30 |
| 2.7.1 | XDC 的特性 | 30 |
| 2.7.2 | XDC 与 UCF 比较 | 31 |
| 2.7.3 | 约束文件的使用方法 | 32 |
| 2.7.4 | 约束顺序 | 32 |
| 2.7.5 | XDC 约束命令 | 33 |
| 2.8 | Vivado 集成设计环境的启动方法 | 34 |
| 2.9 | Vivado 集成设计环境主界面 | 35 |
| 2.10 | Vivado 设计主界面及功能 | 38 |
| 2.10.1 | 流程处理主界面及功能 | 38 |
| 2.10.2 | 工程管理器主界面及功能 | 40 |
| 2.10.3 | 工作区窗口 | 42 |
| 2.10.4 | 设计运行窗口 | 42 |
| 第 3 章 | Vivado 工程模式基本设计实现 | 44 |
| 3.1 | 创建新的设计工程 | 44 |
| 3.2 | 创建并添加一个新的设计文件 | 48 |
| 3.3 | RTL 详细描述和分析 | 52 |
| 3.4 | 设计综合和分析 | 55 |
| 3.4.1 | 综合过程的关键问题 | 55 |
| 3.4.2 | 设计综合选项 | 55 |
| 3.4.3 | Vivado 支持的属性 | 58 |
| 3.4.4 | 执行设计综合 | 66 |
| 3.4.5 | 综合报告的查看 | 69 |
| 3.5 | 设计行为级仿真 | 70 |
| 3.6 | 创建实现约束 | 75 |
| 3.6.1 | 实现约束的原理 | 75 |
| 3.6.2 | I/O 规划器功能 | 75 |
| 3.6.3 | 实现约束过程 | 76 |
| 3.7 | 设计实现和分析 | 81 |
| 3.7.1 | 设计实现原理 | 82 |
| 3.7.2 | 设计实现选项 | 82 |
| 3.7.3 | 设计实现及分析 | 87 |
| 3.7.4 | 静态时序分析 | 93 |
| 3.8 | 设计时序仿真 | 97 |

| | |
|-------------------------------------|------------|
| 3.9 生成编程文件 | 98 |
| 3.9.1 执行生成可编程文件 | 98 |
| 3.9.2 生成编程文件选项 | 98 |
| 3.10 下载比特流文件到 FPGA | 100 |
| 第4章 Vivado 非工程模式基本设计实现 | 104 |
| 4.1 非工程模式基本命令和功能 | 104 |
| 4.1.1 非工程模式基本命令列表 | 104 |
| 4.1.2 典型 Tcl 脚本的使用 | 105 |
| 4.2 Vivado 集成开发环境分析设计 | 106 |
| 4.2.1 启动 Vivado 集成开发环境 | 106 |
| 4.2.2 打开设计检查点的方法 | 107 |
| 4.3 修改设计路径 | 107 |
| 4.4 设置设计输出路径 | 108 |
| 4.5 读取设计文件 | 108 |
| 4.6 运行设计综合 | 109 |
| 4.7 运行设计布局 | 110 |
| 4.8 运行设计布线 | 112 |
| 4.9 生成比特流文件 | 114 |
| 4.10 下载比特流文件 | 115 |
| 第5章 创建和封装用户 IP 核流程 | 117 |
| 5.1 Vivado 定制 IP 流程 | 117 |
| 5.2 创建新的用于创建 IP 的工程 | 118 |
| 5.3 设置定制 IP 的库名和目录 | 119 |
| 5.4 封装定制 IP 的实现 | 120 |
| 5.5 创建新的用于调用 IP 的工程 | 124 |
| 5.6 设置包含调用 IP 的路径 | 124 |
| 5.7 创建基于 IP 的系统 | 125 |
| 5.8 系统行为级仿真 | 129 |
| 5.9 系统设计综合 | 132 |
| 5.10 系统实现和验证 | 133 |
| 第6章 Vivado 高级约束原理及实现 | 135 |
| 6.1 时序检查概念 | 135 |
| 6.1.1 基本术语 | 135 |
| 6.1.2 时序路径 | 135 |
| 6.1.3 建立和保持松弛 | 137 |

| | | |
|--------------|-------------------------------|------------|
| 6.1.4 | 建立和保持检查 | 138 |
| 6.1.5 | 恢复和去除检查 | 141 |
| 6.2 | 时序约束概念 | 142 |
| 6.2.1 | 时钟定义 | 142 |
| 6.2.2 | 时钟组 | 148 |
| 6.2.3 | I/O 延迟约束 | 151 |
| 6.2.4 | 时序例外 | 154 |
| 6.3 | 生成时序报告 | 168 |
| 6.4 | 添加时序约束 | 175 |
| 6.4.1 | 时序约束策略 1 | 175 |
| 6.4.2 | 时序约束策略 2 | 177 |
| 6.5 | 物理约束原理 | 182 |
| 6.5.1 | 网表约束 | 182 |
| 6.5.2 | 布局约束 | 183 |
| 6.5.3 | 布线约束 | 185 |
| 6.6 | 布局约束实现 | 186 |
| 6.6.1 | 修改综合属性 | 187 |
| 6.6.2 | 布局约束方法 | 188 |
| 6.7 | 布线约束实现 | 191 |
| 6.7.1 | 手工布线 | 191 |
| 6.7.2 | 进入分配布线模式 | 192 |
| 6.7.3 | 分配布线节点 | 194 |
| 6.7.4 | 取消分配布线节点 | 194 |
| 6.7.5 | 完成并退出分配布线模式 | 194 |
| 6.7.6 | 锁定 LUT 负载上的单元输入 | 195 |
| 6.7.7 | 分支布线 | 195 |
| 6.7.8 | 直接约束布线 | 197 |
| 6.8 | 修改逻辑实现 | 198 |
| 6.9 | 配置约束原理 | 199 |
| 6.10 | 增量编译 | 199 |
| 6.10.1 | 增量编译流程 | 199 |
| 6.10.2 | 运行增量布局和布线 | 200 |
| 6.10.3 | 使用增量编译 | 202 |
| 6.10.4 | 增量编译高级分析 | 204 |
| 第 7 章 | Vivado 调试工具原理及实现 | 205 |
| 7.1 | 设计调试原理和方法 | 205 |
| 7.2 | 创建新的 FIFO 调试工程 | 206 |

| | | |
|--------------|-------------------------|------------|
| 7.3 | 添加 FIFO IP 到设计中 | 207 |
| 7.4 | 添加顶层设计文件 | 210 |
| 7.5 | 使用 HDL 例化添加 FIFO 到设计中 | 211 |
| 7.6 | 添加约束文件 | 216 |
| 7.7 | 网表插入调试探测流程的方法及实现 | 218 |
| 7.7.1 | 网表插入调试探测流程的方法 | 218 |
| 7.7.2 | 网表插入调试探测流程的实现 | 220 |
| 7.8 | 使用添加 HDL 属性调试探测流程 | 225 |
| 7.9 | 使用 HDL 例化调试核调试探测流程 | 227 |
| 第 8 章 | Vivado 嵌入式系统设计实现 | 232 |
| 8.1 | 简单硬件系统设计 | 232 |
| 8.1.1 | 创建新的工程 | 232 |
| 8.1.2 | 使用 IP 集成器创建处理器系统 | 234 |
| 8.1.3 | 生成顶层 HDL 和导出设计到 SDK | 240 |
| 8.1.4 | 创建存储器测试程序 | 243 |
| 8.1.5 | 验证设计 | 245 |
| 8.2 | 在 PL 内添加外设 | 247 |
| 8.2.1 | 打开工程 | 248 |
| 8.2.2 | 添加两个 GPIO 实例 | 248 |
| 8.2.3 | 连接外部 GPIO 外设 | 256 |
| 8.2.4 | 设计综合 | 257 |
| 8.2.5 | 生成比特流和导出硬件到 SDK | 258 |
| 8.2.6 | 生成测试程序 | 258 |
| 8.2.7 | 验证设计 | 262 |
| 8.3 | 创建和添加定制 IP | 263 |
| 8.3.1 | 创建定制 IP 模板 | 263 |
| 8.3.2 | 修改定制 IP 设计模板 | 266 |
| 8.3.3 | 使用 IP 封装器封装外设 | 271 |
| 8.3.4 | 打开工程和修改设置 | 274 |
| 8.3.5 | 添加定制 IP 到设计 | 275 |
| 8.3.6 | 添加 BRAM | 277 |
| 8.3.7 | 添加约束 xdc | 279 |
| 8.4 | 编写软件程序 | 280 |
| 8.4.1 | 打开工程 | 280 |
| 8.4.2 | 创建应用工程 | 281 |
| 8.4.3 | 为 LED_IP 分配驱动 | 284 |
| 8.4.4 | 分析汇编目标文件 | 286 |

| | | |
|--------------|---------------------------------|------------|
| 8.4.5 | 验证设计 | 288 |
| 8.5 | 软件控制定时器和调试 | 288 |
| 8.5.1 | 打开工程 | 288 |
| 8.5.2 | 创建 SDK 软件工程 | 289 |
| 8.5.3 | 在硬件上验证操作 | 294 |
| 8.5.4 | 启动调试器 | 295 |
| 8.6 | 使用硬件分析仪调试 | 297 |
| 8.6.1 | ILA 核原理 | 298 |
| 8.6.2 | VIO 核原理 | 301 |
| 8.6.3 | 打开工程 | 302 |
| 8.6.4 | 添加定制 IP | 303 |
| 8.6.5 | 添加 ILA 和 VIO 核 | 303 |
| 8.6.6 | 标记和分配调试网络 | 306 |
| 8.6.7 | 生成测试程序 | 307 |
| 8.6.8 | 验证和调试 | 310 |
| 第 9 章 | Vivado 模型设计原理及实现 | 317 |
| 9.1 | FPGA 信号处理方法 | 317 |
| 9.2 | FPGA 模型设计模块 | 318 |
| 9.2.1 | Xilinx Blockset | 318 |
| 9.2.2 | Xilinx Reference Blockset | 319 |
| 9.3 | System Generator 运行环境的配置 | 319 |
| 9.4 | 信号模型的构建和实现 | 320 |
| 9.4.1 | 信号模型的构建 | 320 |
| 9.4.2 | 模型参数的设置 | 324 |
| 9.4.3 | 信号处理模型的仿真 | 326 |
| 9.4.4 | 生成模型子系统 | 326 |
| 9.4.5 | 模型 HDL 代码的生成 | 327 |
| 9.4.6 | 打开生成设计文件并仿真 | 329 |
| 9.4.7 | 协同仿真的配置及实现 | 330 |
| 9.4.8 | 生成 IP 核 | 333 |
| 9.5 | 编译 MATLAB 到 FPGA | 334 |
| 9.5.1 | 模型的设计原理 | 334 |
| 9.5.2 | 系统模型的建立 | 336 |
| 9.5.3 | 系统模型的仿真 | 338 |
| 9.6 | FIR 滤波器的设计与实现 | 339 |
| 9.6.1 | FIR 滤波器设计原理 | 339 |
| 9.6.2 | 生成 FIR 滤波器系数 | 339 |



| | | |
|-------------|---------------------------|------------|
| 9.6.3 | 构建 FIR 滤波器模型 | 341 |
| 9.6.4 | 仿真 FIR 滤波器模型 | 344 |
| 9.6.5 | 修改 FIR 滤波器模型 | 345 |
| 9.6.6 | 仿真修改后 FIR 滤波器模型 | 345 |
| 第10章 | Vivado HLS 原理及实现 | 347 |
| 10.1 | 高级综合工具概述 | 347 |
| 10.1.1 | 高级综合工具的功能和特点 | 347 |
| 10.1.2 | 不同的命令对 HLS 综合结果的影响 | 348 |
| 10.1.3 | 从 C 中提取硬件结构 | 349 |
| 10.2 | 高级综合工具调度和绑定 | 352 |
| 10.2.1 | 高级综合工具调度 | 352 |
| 10.2.2 | 高级综合工具绑定 | 353 |
| 10.3 | Vivado HLS 工具的优势 | 353 |
| 10.4 | C 代码的关键属性 | 354 |
| 10.4.1 | 函数 | 355 |
| 10.4.2 | 类型 | 356 |
| 10.4.3 | 循环 | 362 |
| 10.4.4 | 数组 | 364 |
| 10.4.5 | 端口 | 364 |
| 10.4.6 | 操作符 | 365 |
| 10.5 | 时钟测量术语说明 | 366 |
| 10.6 | HLS 关键优化策略 | 367 |
| 10.6.1 | 延迟和吞吐量 | 367 |
| 10.6.2 | 循环的处理 | 375 |
| 10.6.3 | 数组的处理 | 378 |
| 10.6.4 | 函数内联 | 383 |
| 10.6.5 | 命令和编译指示 | 384 |
| 10.7 | Vivado HLS 数字系统实现 | 386 |
| 10.7.1 | 基于 HLS 实现组合逻辑 | 386 |
| 10.7.2 | 基于 HLS 实现时序逻辑 | 400 |
| 10.7.3 | 基于 HLS 实现矩阵相乘 | 406 |
| 第11章 | Vivado 部分可重配置原理及实现 | 426 |
| 11.1 | 可重配置导论 | 426 |
| 11.1.1 | 可重配置的概念 | 426 |
| 11.1.2 | 可重配置的应用 | 427 |
| 11.1.3 | 可重配置的特点 | 430 |

| | | |
|--------|------------------|-----|
| 11.1.4 | 可重配置术语解释 | 432 |
| 11.1.5 | 可重配置的要求 | 433 |
| 11.1.6 | 可重配置的标准 | 433 |
| 11.1.7 | 可重配置的流程 | 435 |
| 11.2 | 可重配置的实现 | 435 |
| 11.2.1 | 查看脚本 | 436 |
| 11.2.2 | 综合设计 | 437 |
| 11.2.3 | 实现第一个配置 | 437 |
| 11.2.4 | 实现第二个配置 | 444 |
| 11.2.5 | 验证配置 | 445 |
| 11.2.6 | 生成比特流 | 446 |
| 11.2.7 | 部分重配置 FPGA | 448 |

第1章 Xilinx 新一代 UltraScale 结构

UltraScale 结构是业界首款采用最先进的 ASIC 架构优化的 All Programmable 结构。本章主要对 UltraScale 结构的 Kintex 和 Virtex 器件特性进行说明, 并对其内部所提供的设计资源进行详细的说明和必要的分析。通过这些分析, 帮助读者在 Vivado 集成开发环境中, 更加高效地开发基于 UltraScale 结构的 FPGA 应用。

1.1 UltraScale 结构特点

UltraScale 结构能从 20nm 平面 FET 结构扩展至 16nm 鳍式 FET 晶体管技术, 甚至更高的技术, 同时还能从单芯片扩展到 3D IC。

通过 Xilinx Vivado 设计套件的分析型协同优化方法, UltraScale 结构可以提供海量数据的布线功能, 同时还能智能地解决先进工艺节点上的头号系统性能瓶颈。这种协同设计可以在不降低性能的前提下达到实现超过 90% 的利用率。

UltraScale 架构不仅能解决系统总吞吐量扩展和时延方面的局限性, 而且还能直接应对先进工艺节点上的头号系统性能瓶颈, 即互连问题。UltraScale 新一代互连架构的推出体现了可编程逻辑布线技术的真正突破。

Xilinx 致力于满足从多 Gb 智能包处理到多 Tb 数据路径等新一代应用需求, 即必须支持海量数据流。在实现宽总线逻辑模块(将总线宽度扩展至 512 位、1024 位甚至更高)的过程中, 布线或互连拥塞问题一直是影响实现时序收敛和高质量结果的主要制约因素。过于拥堵的逻辑设计通常无法在早期器件架构中进行布线; 即使工具能够对拥塞的设计进行布线, 最终设计也经常需要在低于预期的时钟速率下运行。而 UltraScale 布线架构则能完全消除布线拥塞问题。结论很简单: 只要设计合理, 就能进行布线。

表 1.1 给出了 UltraScale 结构的 Kintex 器件特性, 表 1.2 给出了 UltraScale 结构的 Virtex 器件特性。下面将对 UltraScale 器件内部设计资源进行进一步说明, 以帮助读者充分了解 UltraScale 器件所提供的设计资源。这样, 在使用 Vivado 集成开发环境进行 FPGA 的设计时, 可以更加充分高效地利用这些资源, 从而进一步提高设计效率。

表 1.1 UltraScale 结构的 Kintex 器件特性

| | XCKU035 | XCKU040 | XCKU060 | XCKU075 | XCKU100 | XCKU115 |
|-----------------|----------|----------|----------|----------|----------|-----------|
| 逻辑单元 | 355,474 | 424,200 | 580,440 | 756,000 | 985,440 | 1,160,880 |
| 块 RAM (Mb) | 19.0 | 21.1 | 38.0 | 41.8 | 59.1 | 75.9 |
| DSP 切片 | 1,700 | 1,920 | 2,760 | 2,592 | 4,200 | 5,520 |
| PCI-E 块 | 2 | 3 | 3 | 4 | 6 | 6 |
| GTH 16 Gb/s 收发器 | 16 | 20 | 32 | 52 | 64 | 64 |
| I/O 引脚 | 520 | 520 | 624 | 728 | 832 | 832 |
| I/O 电压 | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V |

表 1.2 UltraScale 结构的 Virtex 器件特性

| | XCVU065 | XCVU080 | XCVU095 | XCVU125 | XCVU160 | XCVU190 | XCVU440 |
|--------------------|----------|----------|----------|-----------|-----------|-----------|-----------|
| 逻辑单元 | 626,640 | 780,000 | 940,800 | 1,253,280 | 1,621,200 | 1,879,920 | 4,432,680 |
| 块 RAM (Mb) | 44.3 | 50.0 | 60.8 | 88.6 | 115.2 | 132.9 | 88.6 |
| DSP Slices | 600 | 672 | 768 | 1,200 | 1,560 | 1,800 | 2,880 |
| PCI-E 块 | 2 | 4 | 4 | 4 | 4 | 6 | 6 |
| Interlaken | 3 | 6 | 6 | 6 | 9 | 9 | 0 |
| 100G 以太网 | 3 | 4 | 4 | 6 | 7 | 9 | 3 |
| GTH 16 Gb/s 收发器 | 20 | 32 | 32 | 40 | 52 | 60 | 48 |
| GTY 32.75 Gb/s 收发器 | 20 | 32 | 32 | 40 | 52 | 60 | 0 |
| I/O 引脚 | 520 | 832 | 832 | 1,040 | 1,040 | 1,040 | 1,456 |
| I/O 电压 | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V | 1.0~3.3V |

1.2 可配置逻辑块

可配置的逻辑块 (Configurable Logic Block, CLB) 是主要的逻辑资源, 用于实现时序和组合逻辑电路。

1.2.1 可配置逻辑块的特点

UltraScale 结构的 CLB 提供了高性能和低功耗的可编程逻辑; 每个 CLB 连接到一个开关矩阵, 用于访问通用的布线资源。一个 CLB 包含一个切片 (Slice), 每个切片提供 8 个 6 输入的查找表, 16 个触发器, 切片中的查找表 (Look Up Table, LUT) 按列排列。UltraScale 架构中包含两种类型的切片, 即 SliceL 和 SliceM。如图 1.1 所示, 给出了 SLICEL (L 表示逻辑) 的内部结构, 其中:

- (1) 图中左侧为 8 个 6 输入的 LUT, 从下到上依次用 A、B、C、D、E、F、G、H 标记。
- (2) 图中右侧为 16 个锁存/触发器资源。
- (3) 图中包含 F7、F8 和 F9 类型的多路复用器。
- (4) 图中在 LUT 右侧的部件为一个 8 位的进位链。
- (5) 图中进位链后的第一列为 F7 类型的多路复用开关。从下到上依次用 F7MUX_AB、F7MUX_CD、F7MUX_EF 和 F7MUX_GH 表示。
- (6) 图中 F7 类型多路复用开关后, 为上下两个 F8 类型的多路复用开关, 从下到上依次用 F8MUX_BOT 和 F8MUX_TOP 表示。
- (7) 图中 F8 类型多路复用开关后, 为一个 F9 类型的多路复用开关, 用 F9MUX 表示。



注

读者可打开器件结构图, 查看这些设计资源。

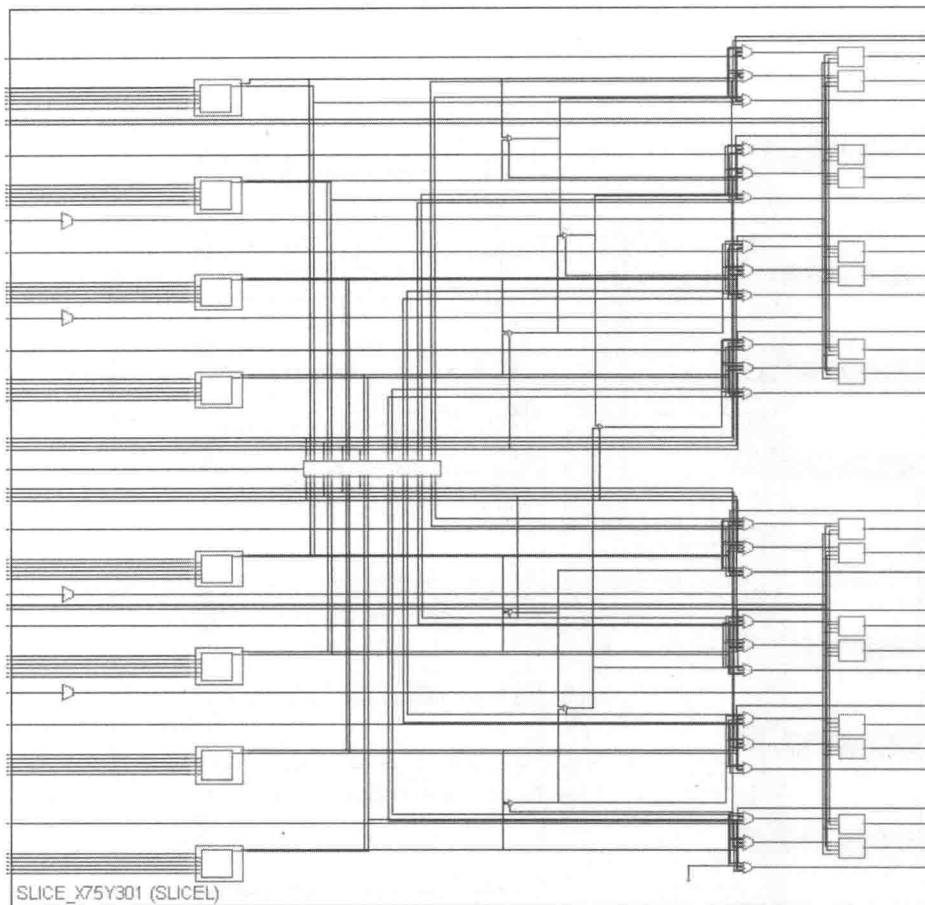


图 1.1 Kintex UltraScale SLICEL 内部结构图

如图 1.2 所示，给出了 SLICEM（M 表示存储器）的内部结构，其中 LUT 能配置为：

- (1) 一个查找表。
- (2) 64 位的分布式存储器。
- (3) 一个 32 位的移位寄存器。

此外，LUT 可以配置为包含公共输入的两个 5 输入 LUT。这样，就能产生两个逻辑运算结果。图 1.3 给出了将 6 输入 LUT 配置为 2 个 5 输入 LUT 的结构图。

1.2.2 多路复用器

多功能多路复用器将 LUT 组合在一起，构成 7、8 或者 9 输入的任何函数功能，或者最多 55 个输入的一些函数功能。每个切片包含 7 个多路复用器，用于构建更多的函数功能。不同多路复用器的功能包括：

1. F7MUX_AB、F7MUX_CD、F7MUX_EF 和 F7MUX_GH

用于组合两个相邻的 LUT。其中，F7MUX 的后缀 AB、CD、EF、GH 表示该类型多路