

赛灵思

中国通讯 Xilinx News

第五十四期

2014年

冬季刊

Issue 54

Winter

2014

利用 Xilinx 的 UltraScale 架构大幅提升生产力

利用 FPGA 对大规模 MIMO 信道进行特性描述

利用 Xilinx FPGA 实现
高效并行实时上采样

充分发挥 PicoBlaze
微控制器的优势

Vivado 2014.3 版本的最新消息

利用 Zynq SoC 简化 4K 电视的开发



 **XILINX**
ALL PROGRAMMABLE™

请即浏览
网络版的全部精彩内容
china.xilinx.com/xcell



新品速递!

欢迎选用本系列产品!

- ✓ 低成本开发工具
- ✓ 支持Linux的開箱即用
- ✓ 现成的SOM解决方案
- ✓ 从原型机转向规模化生产非常容易

ZYNQ

www.microzed.org

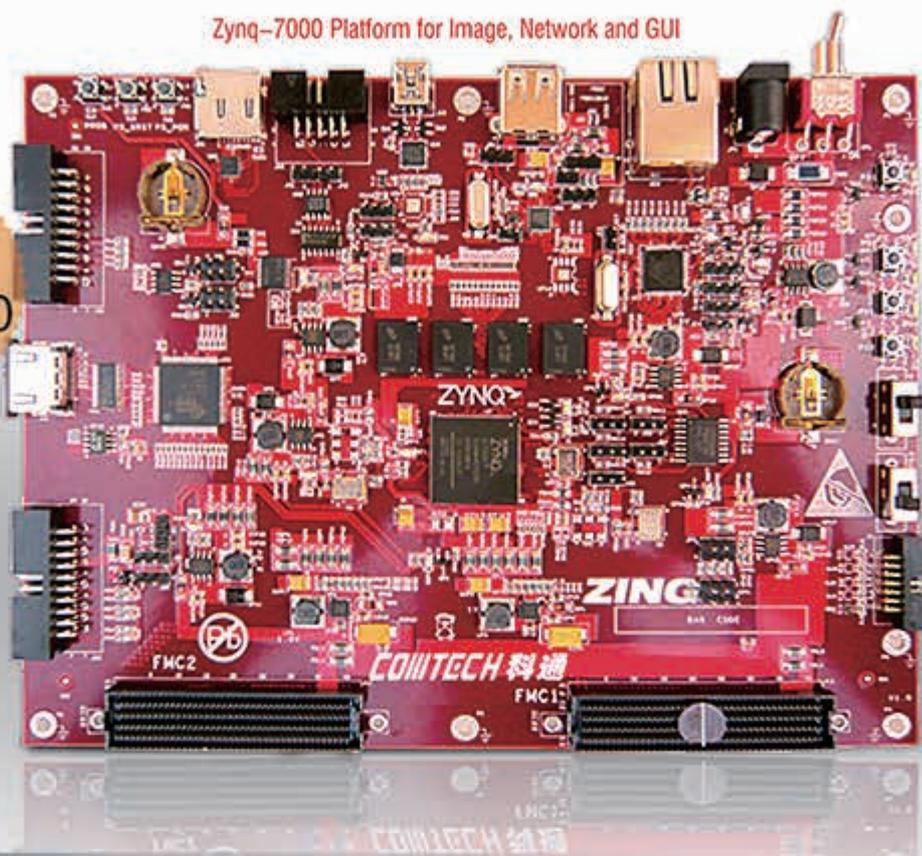
MicroZed™ 是一种基于Xilinx Zynq®-7000 全可编程系统级芯片(SoC) 的低成本开发电路板。它的独特设计让其既可以被用作单独的基本SoC实验用的评估板，也可以与基板组合，用作嵌入式系统化模块(SOM)。这种独立工作/SOM相结合的方法可以很快地将设计构思从概念转到投产，使MicroZed 成为基于SoC应用的理想平台。MicroZed以一个网上社区平台为依托，用户可以在那里下载套件的相关文档和参考设计，也可以与其他进行Zynq设计的工程师合作。



ZING Board 是基于 Xilinx Zynq™-7000 SoC 的开发套件，给高性能系统设计带来帮助，加速设计者创新产品的诞生。

Zynq-7000 Platform for Image, Network and GUI

ONLY
US\$350



ZING 开发套件

- Zing 评估开发板
包含 XC7Z020CLG484-1C
- 参考设计、设计范例以及演示文件
- 开发板设计文档
- 包含所有软件和参考设计、演示以及文档
帮助您快速入门
- 技术文档
- 电缆和电源
- TF卡

融合软/硬件协同编程，开启定制化SoC设计时代

SNOWLeo



主要特性

- Xilinx ZYNQ 7010/7020 AP SoC
- 双核ARM Cortex-A9 主频高达800MHz
- 28nm FPGA百万门可编程逻辑
- 512MB DDR3 SDRAM
- 千兆以太网/USB 2.0 OTG/USB UART
- HDMI 高清输出，支持3D 1080p显示
- TF卡和Nand Flash 双重启动模式
- 高速扩展IO连接器



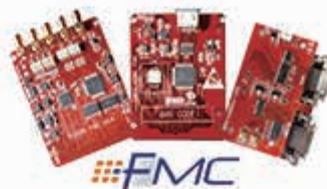
操作系统支持



ZingSoM：体积最小的ZYNQ核心模块

专为OEM和小批量高附加值产品客户量身定制的ZYNQ最小系统模块，集成了ZYNQ系统所需的全部组件，包括DDR3，FLASH，GigaE和USB PHY芯片。预装Linux/Android 操作系统和常用IP，开箱即用。

► ZingSoM 已经开始现货供应



丰富的FMC子卡支持行业应用
提供IP定制设计服务

更多详细信息请联系: xilinx_enquiry@comtech.com.cn

COMTECH 科通

www.comtech.com.cn

出版商 Mike Santarini
mike.santarini@xilinx.com
408-626-5981

编辑 Jacqueline Damian

艺术总监 Scott Blair

设计/制作 Teie, Gelwicks & Associates
1-800-493-5551

广告销售 Dan Teie
1-800-493-5551
xcelladsales@aol.com

国际 Melissa Zhang, Asia Pacific
melissa.zhang@xilinx.com

Christelle Moraga, Europe/
Middle East/Africa
christelle.moraga@xilinx.com

Tomoko Suto, Japan
tomoko@xilinx.com

订购往期期刊 1-800-493-5551

嵌入式方法应用指南助力设计团队交付 Zynq SoC 设计 —— UltraFast

确定为下一个设计项目采用 Zynq[®]-7000 All Programmable SoC 的设计人员，我将为你们带来重大好消息：赛灵思刚刚发布了题为“[UltraFast 嵌入式设计方法指南](#)” ([UG1046](#)) 的方法必读手册，并提供免费下载。

这篇篇幅长达 211 页的手册是赛灵思于 2013 年所出版“[Vivado 设计套件的 UltraFast 设计方法 \(UG949\)](#)”的配套文档，我们曾在《赛灵思中国通讯》第 50 期的封面专题中进行过介绍。

赛灵思方法市场营销高级总监 Tom Feist 指出：“Vivado 设计套件的 UltraFast 设计方法”重点在于帮助所有 Vivado 用户通过 Vivado 工具获得最佳实践，而全新推出的“UltraFast 嵌入式设计方法指南”则旨在帮助整个设计团队最大限度地利用赛灵思丰富的嵌入式工具套件。根据手册指南，设计团队能够优化其设计实践，进而加速其 Zynq SoC 设计的上市进程。

Feist 指出：“‘Vivado 设计套件的 UltraFast 设计方法’获得了巨大的成功。实践证明，其不仅使用户能够加速开发进程（从数月缩短至数周），同时还有助于众多设计团队充分利用 Vivado 的 ASIC 级特性。‘UltraFast 嵌入式设计方法指南’将进一步帮助嵌入式设计团队善用我们的 Zynq SoC。”

通常情况下，嵌入式设计团队由系统架构师、硬件工程师以及软件工程师等具有不同技能的人才组成，团队成员各施其能。Feist 说：“各类工程师都有自己的独门绝技，会使用特定的软件或工具开展设计工作。当然，技能水平也存在差异，而这本方法手册是面向所有人的。”

Feist 指出，本手册将帮助设计人员在整个团队中更好地组织其 Zynq SoC 项目，并能让每个人的 Zynq SoC 设计技能获得锐意提升。Feist 说：“其让嵌入式设计团队的每位成员都能清晰了解重要原则，明确什么该做、什么不该做，而且知道能用什么样的最佳实践来优化设计进程。”

手册内容基于赛灵思专家和成功向市场投放若干 Zynq SoC 产品的设计团队提供的最佳实践。手册根据不同工程设计领域相应分为不同章节。不过 Feist 指出，所有团队成员只要阅读全文都可获得宝贵信息，并能更游刃有余地组织管理项目。

本手册包括六章：系统级考虑事项、硬件设计考虑事项、软件设计考虑事项、硬件设计流程、软件设计流程和调试。此外，手册还设计了流程之间的切换和关系。

我由衷希望“UltraFast 嵌入式设计方法指南”能为您提供全面有益的信息。同时也希望您能喜欢本期赛灵思中国通讯刊载的其他众多佳作。祝您阅读愉快！



Mike Santarini
发行人



www.xilinx.com/xcell/

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124-3400
Phone: 408-559-7778
FAX: 408-879-4780
www.xilinx.com/xcell/

© 2014 年赛灵思公司版权所有。保留所有权利。本文包含的赛灵思、赛灵思徽标和所有其他指定品牌均为赛灵思的商标。所有其他商标是其各自所有者的财产。

本期文章、信息和其他材料仅出于为读者提供方便目的而提供。赛灵思对上述任何文章、信息和其他材料及其使用不做任何明示、暗示或规定性担保，因此用户对其使用带来的风险承担全部责任。任何使用上述信息的人或实体均不得因使用上述信息造成伤害、损失、成本而向赛灵思提出索赔。

领先一代

业界首款 ASIC级架构FPGA

现已提供详细器件选型表、文档、设计工具和方法

- 消除DSP和包处理的瓶颈
- 显著提升定点及浮点运算性能与效率
- 集成二代3D IC系统及全新的3D IC宽存储器优化接口
- 海量I/O与存储器带宽，大幅降低时延
- 大幅降低功耗



视点

发行人致信

嵌入式方法应用指南助力设计团队交付 Zynq SoC 设计——UltraFast... 4

XCELLENCE BY DESIGN APPLICATION FEATURES

ULTRASCALE领域的出色表现

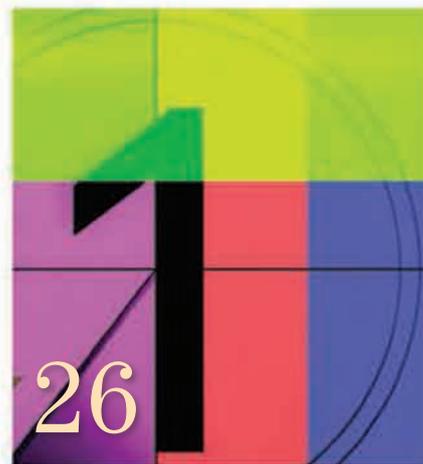
UltraScale为高带宽设计带来板载 Interlaken...14

无线通信领域的出色表现

利用FPGA对大规模MIMO信道进行特性描述...18

视频处理领域的出色表现

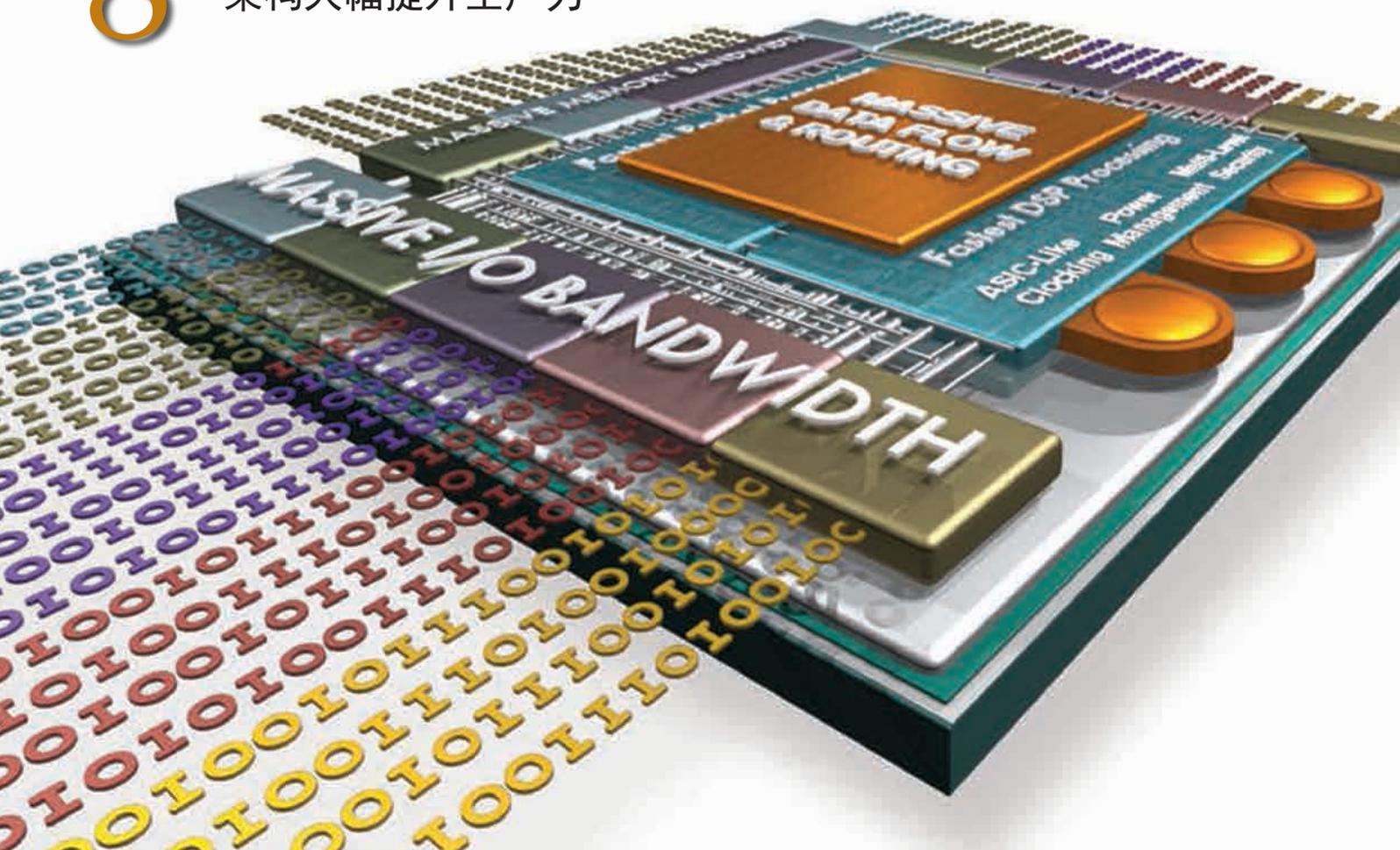
利用Zynq SoC简化4K电视开发...26



封面专题

8

利用Xilinx UltraScale
架构大幅提升生产力



THE XILINX XPERIENCE FEATURES

手把手课堂：FPGA 101

利用Xilinx FPGA实现高效并行实时上采样...32

手把手课堂：FPGA 101

FPGA设计规划框架...40

手把手课堂：FPGA 101

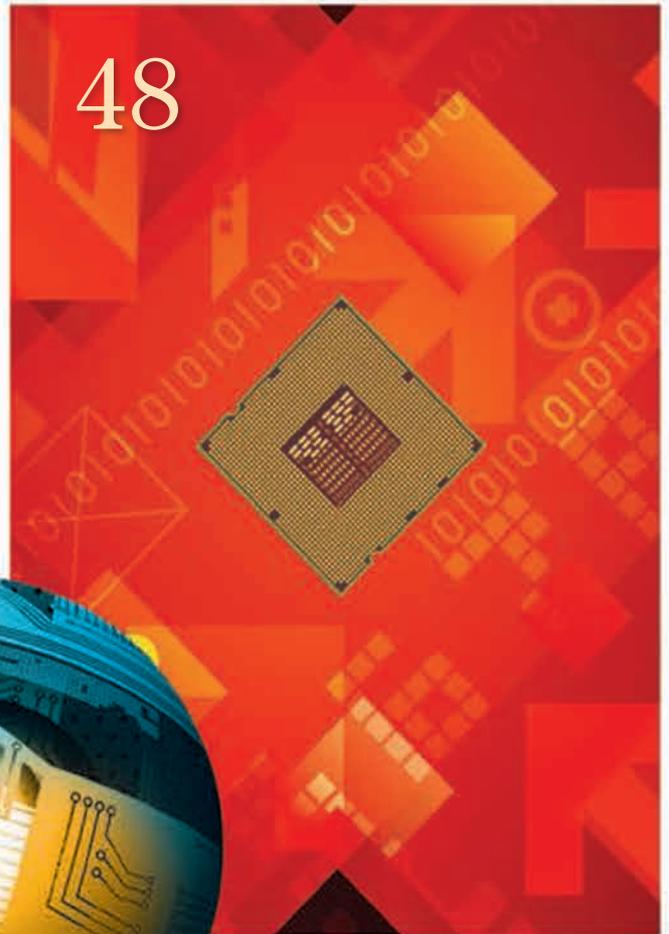
用Vivado IPI和Xilinx IP实现
更快速的设计输入...44

手把手课堂：FPGA101

充分发挥PicoBlaze微控制器的优势...48

专家园地

通过实时改变使用率研究
FPGA功耗行为...54



48



40



54

XTRA READING

号外，号外

Vivado 2014.3版本的最新消息... 60

实例

Xilinx联盟计划合作伙伴的最新最佳技术... 62



利用Xilinx UltraScale 架构大幅提升生产力

作者: Nick Mehta

赛灵思公司高级技术市场经理

nick.mehta@xilinx.com

赛灵思UltraScale架构的增强功能结合Vivado设计套件的省时工具能帮助您更快打造出色系统。

许

多市场和应用都对系统带宽和处理功能需求显著增长。无论是有线或无线通信、数字视频还是图像处理，更高的数据吞吐量要求都实现相同的结果，那就是所有系统组件的流量和要求都在提升。通过并行和串行 I/O 到达片上的数据也越来越多。数据必须经过缓存，然后再次以 DDR 存储器形式通过并行 I/O 或以串行存储器标准（如混合内存立方体 (HMC) 和 MoSys 带宽引擎）形式再次通过串行 I/O。数据从并行和串行 I/O 返回后，在传送到下个目的地之前必须在逻辑和 DSP 中进行处理。

由于诸多原因，系统处理要求越来越复杂。以更高数据速率传输的更大数据包要求并行数据总线更宽，而且频率更高。为高效处理数据，往往要在单一器件上构建整个系统，从而避免因两个 FPGA 之间发送大量数据引起的相关时延和功耗问题。因此需求更密集的具有更多功能的 FPGA。重要的是，在提升高功能 FPGA 利用率的同时，必须保持能以最高性能运行，以避免器件填充时性能下降。

让复杂的高容量器件实现高利用率听起来可能对设计人员来说是一项很艰巨的任务。赛灵思提供了大量解决方案，旨在缩短设计时间，让设计人员集中精力实现产品差异化，从而从市场中脱颖而出。

结合使用高性能并行SelectIO和高速串行收发器连接功能，可在UltraScale FPGA之间实现数据传输。

ULTRASCALE 架构

为了解决普遍存在的市场难题，赛灵思近期推出了 UltraScale™ 架构（如图 1 所示），可提供前所未有的系统集成度、高性能和出色的功能。赛灵思用这款全新架构创建了两款高性能 FPGA 系列产品。赛灵思 Virtex®

UltraScale 与 Kintex® UltraScale 系列相结合，能满足多种不同系统要求，着重通过大量技术创新与进步降低总功耗。UltraScale 技术共享大量构建块，能提供可扩展的架构，并针对多种不同的市场需求精心优化。

增加系统带宽

在进行任何信号处理或数据操作之前，数据首先要到达目的地。目前已有大量串行和并行协议和标准专门满足目标应用的特定需求。大多数标准的一个共同宗旨就是提升整体数据吞吐量，从而支持大量信息以越来越高

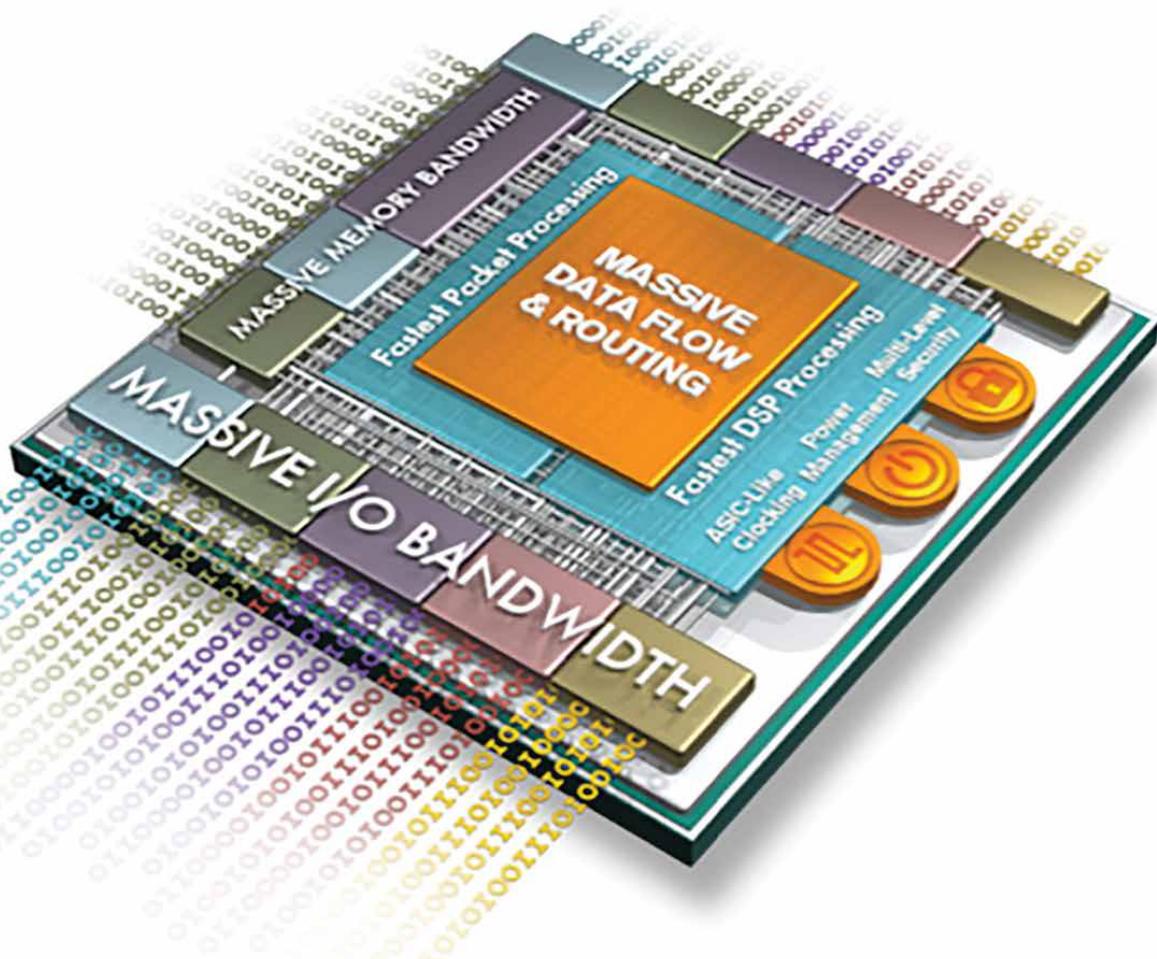


图1 -UltraScale架构的主要优势在于速度和带宽。

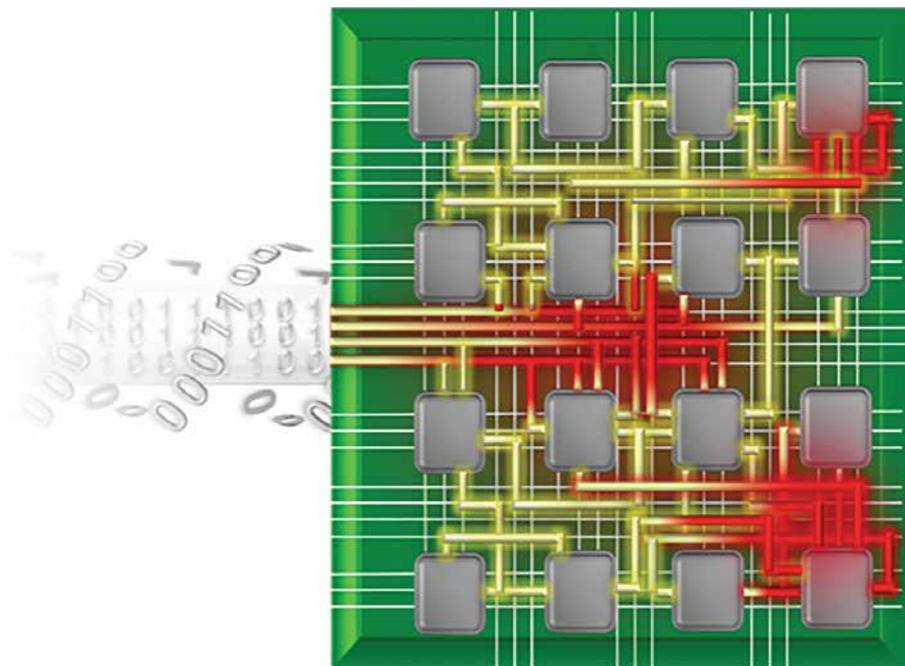


图2 - UltraScale架构能处理大量数据。

的数据速率进行系统传输。

结合使用高性能并行 SelectIO™ 和高速串行收发器连接功能，可在 UltraScale FPGA 之间实现数据传输。I/O 模块通过灵活的 I/O 标准和电压支持可实现先进的存储器接口和网络协议。UltraScale 架构中的不同串行收发器数据传输速率高达 16.3 Gbps，能提供主流串行协议所需的所有性能，最高支持 32.75 Gbps，能满足 25G+ 背板设计的要求，相对于前一代收发器而言能大幅降低单位比特的功耗。UltraScale FPGA 中的所有收发器都支持 PCI Express® Gen3 和 Gen4 所需的数据速率，而 PCI Express 的集成块则支持基于 UltraScale 架构的 FPGA，最高能支持 x8 Gen3 端点和 Root Port 设计。

数据时钟功能与缓冲

所有同步系统都要依靠一个或多个时

钟信号来实现电路同步。系统性能提升要求以更大器件容量提高时钟频率，并要求改进时钟灵活性，降低总时钟功耗。

UltraScale 架构采用功能强大的经过重新架构设计的时钟管理电路系统，包括时钟综合、缓冲和路由组件，能提供功能强大的框架，满足设计要求。时钟网络支持 FPGA 中极为灵活的时钟分布，从而能最大限度地减少与时钟信号相关的偏移、功耗和延迟。此外，时钟管理技术紧密结合专用存储器接口电路系统，可支持 DDR4 等高性能外部存储器。时钟细分和新型门控时钟粒度技术相对于现有的 FPGA 而言能进一步加强对时钟功耗的控制。

相对于前一代 FPGA 以及同类竞争性 FPGA，全局功能时钟缓冲器数量的大幅增加有助于显著提升设计人员的生产力。此前，全局缓冲器的使

用要非常节省，FPGA 中心只有 32 个全局时钟缓冲器。UltraScale 架构能在架构中自由分布全局功能时钟缓冲器，从而在需要时就能提供资源，不用再那么节省了。此外，赛灵思相对于前一代 FPGA 而言大幅简化了时钟缓冲器类型，且保留了所有时钟切换、时钟分频和时钟使能功能，从而获得大量灵活的高功能时钟缓冲器，能在需要时提供全部功能。

数据的存储、处理和路由

任何系统的关键都在于其处理、操作和转换接收到的数据的能力（如图 2 所示）。系统复杂性的提升，要求将通用架构与更专业的功能（专门针对具体类型的数据处理）相结合。

如今的 FPGA 架构集成众多组件：含有六输入查询表（LUT）和触发器的可配置逻辑块（CLB）、带 27x18 乘法器的 DSP Slice、带内置 FIFO 和 ECC 支持的 36Kb Block RAM 等。这些资源通过大量的高性能、低时延互联紧密连接在一起。

除了逻辑功能之外，CLB 还提供移位寄存器、乘法器和进位逻辑功能，并能配置 LUT 为分布式存储器来配合高功能可配置 Block RAM。DSP Slice（包括全新的 96 位宽 XOR 功能、更宽的 27 位预加法器和 30 位输入）执行大量独立功能，其中包括乘法累加、乘法加法和模式检测等。除了器件互联之外，在采用第二代 SSI 3DIC 技术支持的器件中，信号可采用专用低时延接口模块在超级逻辑区域之间传输。这种整合的路由资源能方便地支持下一代数据总线宽度，从而让器件利用率达到 90% 以上。

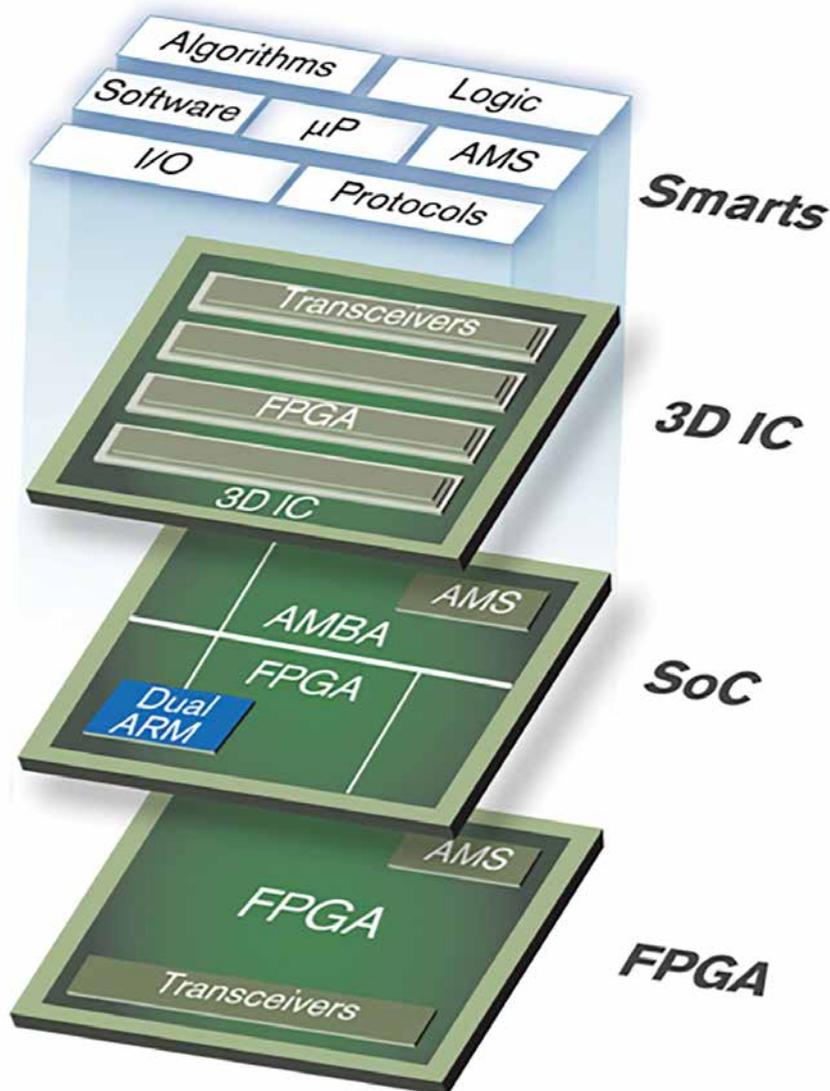


图3 – UltraScale器件为业界领先的科技添加常用功能。

简化设计挑战

UltraScale 架构带来的架构增强功能可帮助设计人员在相同区域打包更多设计方案，但同时器件尺寸也在增大。其结果就是单器件能载入更多设计，这是一大优势，但也给设计团队带来了巨大压力，必须快速实现指定设计，从而让最终产品尽可能快的上市。利用 UltraScale 架构和协同优化的 Vivado® 设计套件，赛灵思能够打造出多个省时、生产力更高的解决方案。

集成核心功能

高度灵活的可编程性这一极具价值的功能与诸多事情一样，要想有所得就要付出代价。利用可编程资源构建的功能可能比专门针对该功能的模块尺寸更大，甚至可能速度更慢。当然，FPGA 本质上明显的优势就是具有可编程性，但赛灵思 FPGA 实现了专门功能、集成 IP 的适当平衡，能帮助用户快实现常用功能（图 3）。UltraScale 架构包含针对常见通信协议的集成模块。Kintex UltraScale

和 Virtex UltraScale 器件中包含针对 PCI Express、100G Ethernet 和 150G Interlaken 的多个集成模块，且都经过全面测试和验证，能确保实现功能。

除了通信协议之外，每个 I/O Bank 都包含一个可编程存储器 PHY，能用存储器接口生成器 (MIG) 工具进行配置。这充分说明了根据需要进行集成的特性。存储器 PHY 和一些控制逻辑可创建为可编程专用函数，但存储器接口的数字部分用器件架构构建，针对不同模式提供所有必需的定制与支持，而这则是专门电路所难以实现的。

在器件架构中有大量其它模块，专门设计用于执行特定功能，同时保持可编程性。设计人员能以不同深度和广度来配置这些模块存储器，以级联成更大型的低功耗阵列。DSP Slice 具有许多模式，能让用户根据所选功能访问模块的不同组件。因此除了门和寄存器之外，UltraScale 架构上能实现大量丰富的功能。

可定制、可重复的 IP 能提高生产力

每个设计都包含许多不同架构构建块，相互连接构成系统。行业内有些功能已经非常完备了，从经济性角度讲完全可以将它们做成专用的固定功能模块。然而，最佳设计方法是采用可编程逻辑构建功能，并经过验证，随时按需重用。这种 IP 概念已经存在了很多代，但赛灵思近期推出了一些生产力增强特性（图 4）。

即插即用 IP

2012 年，赛灵思采用 ARM® AMBA® AXI4 接口作为即插即用 IP 的标准接口。采用统一的标准接口相比过去能大幅简化 IP 集成，将多种不同接口整

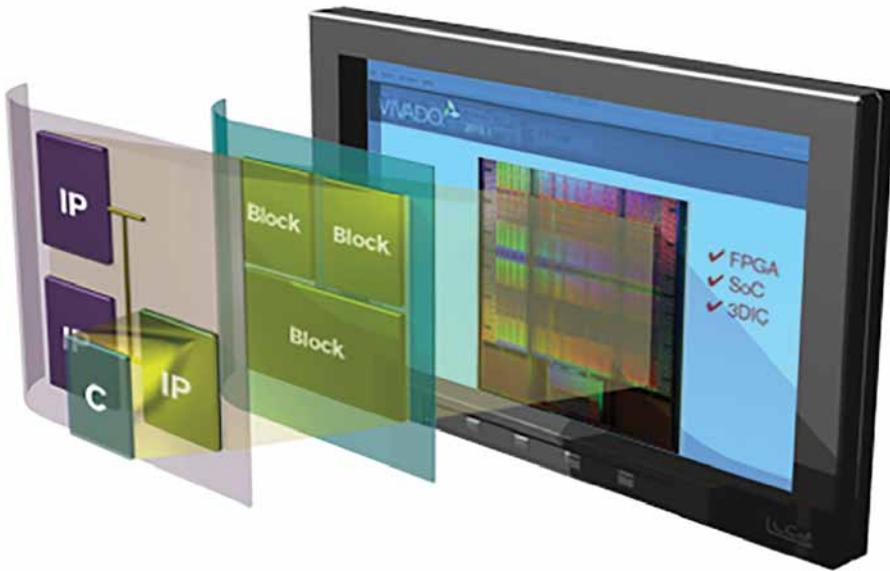


图4 – Vivado工具加速复杂设计的创建和实现。

合到一个，设计人员也不必去掌握多种不同的接口。UltraScale 架构继续受益于 AXI4 互联的灵活性和可扩展性，从而帮助设计人员实现最快的产品上市进程，同时采用 AXI4-Lite 和 AXI4-Stream 等不同 AXI4 互联协议优化 IP 性能、面积和功耗。

Vivado IP Packager 和 IP Catalog 采用 IP-XACT 标准（该标准最初由 SPIRIT 联盟作为工具流程中封

装、集成和重用 IP 的标准结构推出）。IP-XACT 现已获得 IEEE 的批准（IEEE1685-2009）。Vivado IP Packager 采用在本地或共享驱动器上可扩展 IP Catalog 中提供的约束、测试平台和文档来创建设计。采用 Vivado IP Catalog，用户能够将自己的 IP、赛灵思的 IP 以及第三方 IP 无缝集成在一起，从而确保设计团队轻松一致地分享所有 IP。

VIVADO IPI

VIVADO IP 集成器（Vivado IPI）（图 5）是以 IP 为中心的设计流程，可加速系统集成进度，更轻松快捷地将不同组成部分组装成系统。利用交互式图形用户界面，IPI 提供 IP 接口的智能自动连接、一键式 IP 子系统生成和强大的调试功能，可帮助设计人员轻松快捷地连接其 IP Catalog 中的任何乃至全部 IP。这种功能使得设计人员能快速组装复杂系统，包括来自多种来源的设计资源——一些免费的、一些购买的、一些内部创建的——并且能清楚掌握所有构建块都能得到正确配置。从概念到调试从未如此神速。

简而言之，UltraScale 架构在许多关键方面进行了架构创新，能成功满足下一代高性能设计的严格要求。像 UltraScale 这样，能确保使用系统频率越来越高的宽数据总线来实现设计，是成功开展设计工作的必要组成部分。不过，随着器件尺寸和复杂度的提升，帮助设计人员不断提高生产力变得至关重要。赛灵思提供集成块和预验证 IP 组合，能为设计人员带来加快实现出色解决方案所需的全部工具。

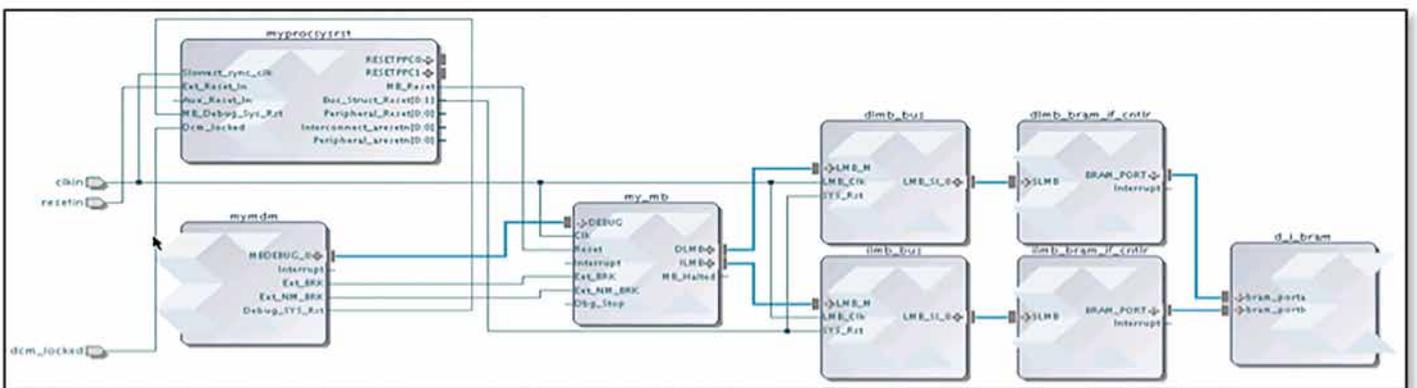


图5 – 在 IPI 中构建设计方案和连接 IP 块一样简单。

UltraScale 为高带宽设计带来板载 Interlaken

赛灵思在其 UltraScale FPGA 上集成了 Interlaken IP，可简化包处理系统互联。

作者：Martin Gilpatric

赛灵思公司收发器技术市场营销经理

martin.gilpatric@xilinx.com

带宽

带宽是个有趣的东西。十年前，大多数人都不清楚什么是“每秒字节”。而今天，在线视频、智能手机和各种现代化互联社会不可或缺的装备让带宽成为所有人关注的问题。这些应用告诉人们，带宽非常重要，人们也清晰地意识到需要更高的带宽。

遗憾的是，高级堆栈顶层观点会制约相关技术的发展，阻碍为家庭以

及便携式设备提供更多比特与字节。所有这些数据都要求我们的基础设施必须跟上发展要求。过去依靠 10Gbps 光学技术的互联机制现在已经发展到 40 乃至 100 Gbps，而不久的将来还有望推进到 400Gbps。

当今产业中的 INTERLAKEN

在整个过程中，基础包处理任务和架构并没有发生巨大变化。同样，100Gbps 系统面对的许多封装、功耗和热性能限制与其前代技术也一样。这意味着系统所有组件都需要大幅提高速度。更重要的是，这些器件之间的互联必须能够很容易扩展。系统中使用的 FIC、NPU、MAC 及其它 ASSP 来自众多不同的供应商，这导致问题进一步复杂化。这些器件采用各种互联宽度和速率来实现目标吞吐量。

现在，赛灵思和 Interlaken 协议已着手协助解决包处理功能之间的社交互联瓶颈问题。由思科和 Cortina 合力打造的 Interlaken 旨在应对这些挑战。赛灵思 FPGA 可通过高性能可编程逻辑和高性能收发器轻松实现这项标准。以下三个方面使得 Interlaken 成为应对这些难题的最佳协议：其具备高度可扩展性；Interlaken 非常适用于 OTN 和以太网系统；它是一项开放式协议。

从最基本的层面说，Interlaken 旨在最大限度地提升灵活性和可扩展性。该规范没有强行要求预定义的线路速度或信道宽度。举例来说，这意味着您实现 150Gbps 的接口就能对 100G 以太网数据路径（带其它包头）进行封装，12 条信道每条线路 12.5 Gbps 或者 6 条信道每条分别为

Interlaken 在包处理过程中的潜在位置

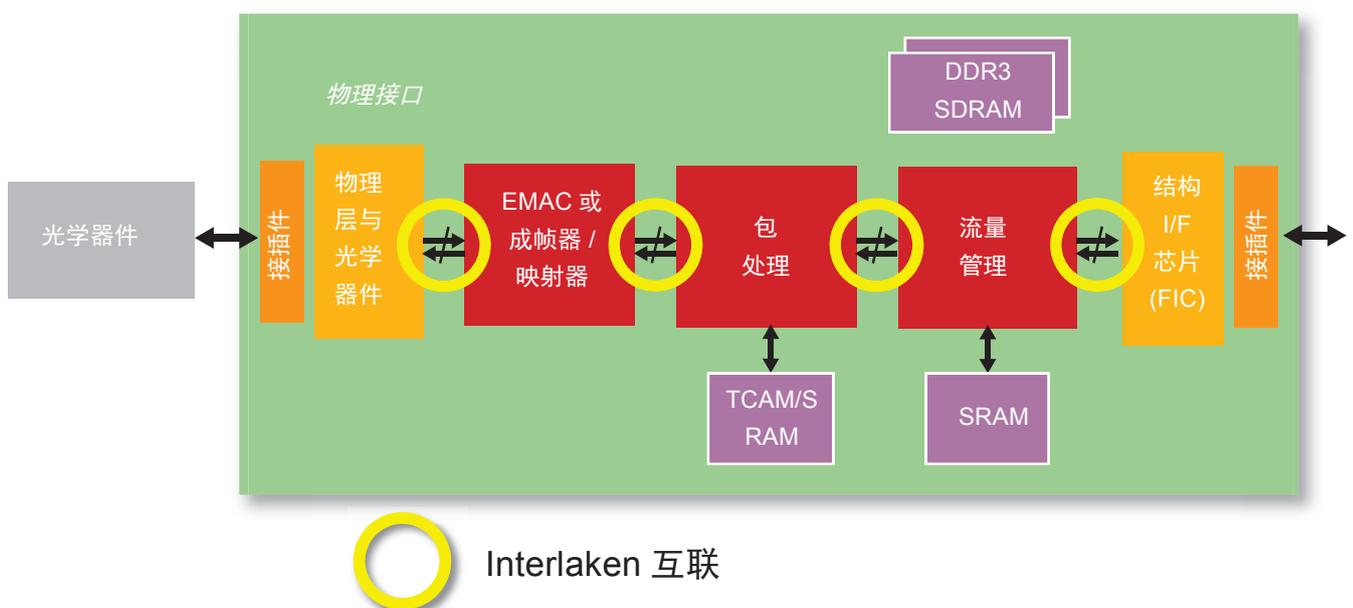


图1 – 利用Virtex UltraScale的32.5-Gbps GTY高性能收发器（如本视频所示）和集成式Interlaken模块，客户能够在不影响灵活性的同时创建性能更高、功耗更低、集成度更紧密的网络。

25 Gbps。无论哪种情况，Interlaken 的逻辑接口均可得到保持，同时简化了其它设计。此外，这种灵活性也使 Interlaken 适用于细分市场应用，如与 TCAM 协同工作的后备接口。

无论数据通过多少条信道传输，都要在这些信道上进行分段和拆分。这种方法的优势在于能够减少时延，同时能很好地支持 OTN 流或以太网数据包。在分段过程中，Interlaken 可支持信道化，让设计人员能够灵活地实现数据包优先化等特性，同时能在协议现有的特性集中工作。

Interlaken 具有丰富的特性集，而且无需注册，可免费使用，因此非常适合推广。此外，原始定义中包含了各种利益相关方公司，这意味着 Interlaken 的特性集也能方便地满足多种不同需求。由于能迅速获得各种 ASSP 和 IP，赛灵思 FPGA 的使用和推广都变得更方便。

赛灵思和 INTERLAKEN： 一对完美的组合

赛灵思素以帮助设计人员更方便地采用最新高性能标准而闻名。在向 100 Gbps 乃至更高技术转移的过程中，实践证明赛灵思器件在包处理系统和测试设备方面发挥了重要的主导作用。由于 100G 以太网 MAC、OTU4 和 Interlaken 解决方案的高质量 IP 随时可供使用，因此配合赛灵思的高速 FPGA 结构和世界一流的收发器能够为客户带来灵活而强大的解决方案。

从 Virtex[®]-5 代开始就面向赛灵思 FPGA 提供了 Interlaken IP，一开始是通过第三方厂商提供，而在收购了 Sarance 之后，则是由赛灵思自己提供。UltraScale™ 系列 FPGA 是提供 Interlaken 解决方案的第四代赛灵思器



图2 - 利用 Virtex UltraScale 的 32.5 Gbps GTY 高性能收发器（如本视频所示）和集成式 Interlaken 模块，客户能够在不影响灵活性的同时创建性能更高、功耗更低、集成度更紧密的网络。

件，其取得了一项重要发展。

在 UltraScale FPGA 中，赛灵思将 Interlaken IP 整合到芯片中。通过将 Interlaken 设为固定的特性，赛灵思能够释放出更多结构逻辑，减少软实现技术带来的时序约束，从而将设计人员从繁重的工作中解脱出来。此外，这款一体化解决方案还能节约动态和静态功耗，而且不会对灵活性造成不良影响。统一的 Interlaken 模块能以任何线路速度实现多达 12 条信道，每条信道速度高达 12.5 Gbps，或者实现六条信道，整体速度高达 25 Gbps。

赛灵思在 Interlaken 的优势基础上进一步发展

集成式 Interlaken IP 只是让赛灵思 UltraScale FPGA 具有巨大吸引力的其中一个因素。赛灵思 UltraScale FPGA 拥有可支持不同线路速度的大量特性丰富的收发器、高级高速可编程逻辑以及用于 MAC 解决方案的集成 IP 软核，并能实现新兴标准，任何其它 FPGA 或 ASSP 都无法达到类似的水平。

UltraScale FPGA 中的 GTH 和 GTY 收发器具备许多特性，能使其在各种不同条件下工作。GTH 的线路速度从 500 Mbps 到 16.3 Gbps 不等，GTY 的运行速度可达 32.75 Gbps，可确保 FPGA 能够支持链路合作伙伴所需的任何线路速度（[敬请在优酷上观看图 2 所示的视频演示](#)）。这些收发器的均衡功能（包括连续时间线性均衡和决策反馈均衡），能帮助用户将板上其它器件的任何元素与光学器件或背板远端的器件相连接。

为了更方便地启动这些高速互联，无论是 100G 以太网、OTU 4.4 还是 Interlaken 宽接口，所有收发器均衡特性均为自适应。这就意味着在 GTY 中 CTLE 的三个抽头和 DFE 的 15 个抽头之间，收发器自身就能处理由此形成的数百万种组合。设计人员无需手动调节每条链路并保持工艺、电压和温度变化条件下的链路容量。Interlaken 可支持无限宽的接口，因此自适应性能确保启动工作简单而稳健。

UltraScale 集成式 100G 以太网

MAC 或 CMAC 与集成式 Interlaken 模块相辅相成。CMAC 具备与集成式 Interlaken 模块相同的优势，能够减少功耗、降低复杂性并缩短实现时间，同时可为今天的 100G 光学技术（CFP、CXP、CFP2、CFP4 和 QSFP28）提供直接可用的接口。连接这些光学接口需要采用 10x10.3125-Gbps 或 4x25.78-Gbps 串行接口。CMAC 和 UltraScale 收发器可支持两种接口：10.3125 Gbps 的 GTH 和适应两种线路速度的 GTY。CMAC 只提供与线路卡断开接口的可能协

议。赛灵思提供的 IP 能在结构中实现，以支持 OTN 应用或以太网标准组合：1G、10G、40G、100G 乃至新兴 400G 标准。

接采用大量 Interlaken 配置的 ASSP 并实现 MAC 功能是 FPGA 的两大重要用途。不过，FPGA 设计还有很多值得进一步探索的潜力。精明的设计人员已经开始智能优化系统，让 FPGA 处理选用 ASIC 或 ASSP 时执行成本较高的任务。这些功能包括不同层次的包处理、TCAM 后备接口实现或其它帮助产品超越竞争对手的

特性。

与科技发展趋势一样，高带宽领域也保持不断前进。这意味着吞吐量会不断提升，新标准会逐渐出现，而赛灵思 UltraScale 器件在这两方面始终战无不胜。通过将 Interlaken 集成到 UltraScale FPGA 上，赛灵思能够支持现有的 100G 系统和所有未来系统，方便地采用新标准，增强现有架构的特性，并更好地全面集成系统。🌈

Xilinx 扩展 20 nm Kintex UltraScale 产品阵容 锁定要求最严苛的数据中心加速、视频和信号处理应用 *Kintex UltraScale KU115 FPGA 器件现已出货*

2014 年 10 月 6 日，中国北京 - All Programmable 技术和器件的全球领先企业赛灵思公司（Xilinx, Inc. (NASDAQ:XLNX)）今天宣布 Kintex® UltraScale™ KU115 FPGA 器件正式出货，从而使得其进一步扩展了其 20 nm 产品阵容。作为 Kintex UltraScale 系列的旗舰产品，KU115 不仅在单个可编程器件中集成了最多的 DSP 数量，更是增加了一倍可用的 DSP 资源。经过 DSP 优化的 KU115 FPGA 器件，主要针对包括数据中心、视频与医疗成像、广播系统以及雷达等在内的数据中心计算加速和信号处理应用。KU115 现已开始向多家客户发货，这是赛灵思向市场成功交付的第四款 20 nm UltraScale 器件系列。UltraScale 系列的采用势头以及用户的反馈表明，该系列器件能够给赛灵思客户带来巨大价值，比如 Google 地图工程师就发现此款器件助力了其世界地图的绘制。

Kintex UltraScale KU115 FPGA 针对从浮点到定点等各种 DSP 密集型运算进行了精心优化。除了可为对称滤波应用带来高达 8,181 GMAC 的卓越性能，其嵌入式 DSP 模块增强了对高清视频编码、有线通信系统的前向纠错 (FEC) 与循环冗余校验 (CRC)，以及无线通信和航空航天应用领域常用的复杂滤波与算法等应用的功能。KU115 每次运算都能更有效地利用资源，具有最佳的单位功耗性能，从而可满足处理密集型系统严格的功耗和热性能要求。结合 Vivado® 高层次综合和赛灵思的 OpenCL 软件定义开发环境等高级抽象工具，KU115 系列可在设计和实现 DSP 密集型算法时提供完整的解决方案，并最大限度地减少计算瓶颈问题。

Kintex UltraScale FPGA 可提供多达 116 万个逻辑单元、5,520 个优化的 DSP Slice、76 Mb BRAM、16.3Gbps 背板收发器、PCIe® Gen3 硬模块、集成式 100Gb/s 以太网 MAC 与 150 Gb/s Interlaken IP 核，以及运行速率高达 2,400 Mb/s 的 DDR4 存储器接口。Kintex 器件系列凭借其最低的功耗和最佳的性价比打造出了全新的 28 nm 中端产品，并可在 20 nm 工艺下提供数量最多的 DSP。

关于赛灵思

赛灵思是 All Programmable 器件、SoC 和 3D IC 的全球领先供应商。赛灵思公司行业领先的产品与新一代设计环境以及 IP 核完美地整合在一起，可满足客户对可编程逻辑乃至可编程系统集成的广泛需求。如需了解更多信息，敬请访问赛灵思中文网站：<http://china.xilinx.com/>。

利用FPGA对大规模MIMO信道进行特性描述

作者: Patrick Murphy
Mango Communications公司总裁
patrick@mangocomm.com

Clayton Shepard
莱斯大学研究生
cws@rice.edu

Lin Zhong
莱斯大学副教授
lzhong@rice.edu

Chris Dick
赛灵思DSP首席科学家
chris.dick@xilinx.com

Ashutosh Sabharwal
莱斯大学教授
ashu@rice.edu

由24个FPGA、96个天线以及一个802.11定制模块构建的系统可对多用户MIMO传播环境进行实时分析。

多用户 MIMO (MUMIMO) 是一种无线通信技术, 采用基础架构节点 (例如基站和接入点) 上的多个天线为多个客户同时提供服务。MU-MIMO 是未来无线标准中必不可少的组成部分, 有望为繁忙的网络带来显著的性能提升。

人们预想随着无线系统的更新换代, 基站的天线数量将会不断增加, 最终形成“大规模 MIMO”系统。大规模 MIMO 方案将 MU-MIMO 基站上的天线数量增至数十个甚至数百个, 以求在提高性能的同时简化基站的信号处理功能。有一种可扩展的大规模 MIMO 技术被称为共轭波束形成^[1]。该技术的一种早期实现方案显示出实际性能提升潜力^[2]。

多用户 MIMO 技术离不开无线传输环境的准确信息。一个 MUMIMO 基础架构节点可同时服务多个用户, 前提条件是具备有关每个用户无线信道的准确的最新测量信息。实时收集信道信息极具挑战性, 而且陈旧或不准确的信道信息对性能影响可能非常严重。

我们设计出一款用于描述大规模 MIMO 信道特性的集成系统, 可帮助研究人员实时分析信道的动态特性。该系统的核心部分采用基于赛灵思 FPGA 的 WARP 硬件平台和 [Mango Communications](#) 的 [802.11 参考设计](#), 可通过莱斯大学 (Rice University) 的 Argos 平台将 24 个 FPGA 连接至 96 个天线^[2]。Mango Communications 开发的 Python 定制框架可用于实时控制和收集来自阵列中每个节点的数据。Mango 和莱斯大学的这两款工具结合使用可提供深入的无线协议栈信息, 包括对大规模 MIMO 进行特性描述所需的原始信道数据。

Mango Communications 802.11 定制模块的重要功能是实时发送来自各个接收天线的底层基带参数, 例如 AGC 增益、信道估计值和原始数据包内容 (甚至是含有错误的数据包)。参考设计的这项功能使莱斯大学的 Argos 阵列作为符合 802.11 标准的接入点 (AP) 使用, 为商用 Wi-Fi 设备 (例如智能手机、平板电脑或笔记本电脑) 提供互联网, 同时实时收集阵列天线与每个客户

我们的大规模MIMO系统Argos利用FPGA实时地对数据进行本地处理，可显著减少上游处理器的负担。

端之间的信道数据。赛灵思 FPGA 是实现每个天线上的实时处理的关键。它们将来自各个天线的的数据简化成可由定制模块发送和分析的每客户端信道特性信息。

让我们来具体了解一下 WARP 硬件平台、Mango 的 802.11 模块以及针对 MU-MIMO 的共轭波束形成策略。最后，我们还会检查特性分析过程，包括实时从 Wi-Fi 客户端收集无线信道测量数据，以及处理信道数据以估计可达到的 MU-MIMO 性能。

系统组件

无线开放访问研究平台（WARP）是一款全新构建的可扩展、可编程无线平台，用于高级无线网络的原型设计。WARP 能够将高性能可编程硬件与包含参考设计和支持资料的开放源代码库结合在一起。

WARP 项目于 2006 年由莱斯大学教授 Ashu Sabharwal 创立，最初由美国国家科学基金会（National Science Foundation）提供资金支持，目前由赛灵思持续提供支持。自此，

该项目已经发展成为能够自我维持的开放源码平台，其用户遍布世界各地。Mango Communications 于 2008 年从莱斯大学 WARP 项目中拆分出来，最初的目的是制造和销售莱斯 WARP 硬件。2012 年，Mango 发布了经过彻底重新设计的 WARP v3 硬件。如今，Mango 工程师是 WARP 资源库和论坛最活跃的贡献者，持续提供开发和开放源码 WARP 设计支持。

大规模 MIMO 信道测量系统的核心组成部分是 Mango Communications 的 WARPv3 硬件平台。WARP v3 适用于快速、实时构建新型无线系统的原型设计。在硬件方面集成了一个赛灵思高性能 Virtex[®]-6 FPGA，两个灵活的射频接口以及包括 DDR3 DRAM 和两个 1Gbps 以太网接口在内的多个外设。WARP v3 板卡可通过 Mango 的双无线电 FMC 模块扩展至 4 个射频接口。图 1 中的硬件配置可提供具有 FPGA 独立数字基带连接的 4 个完全可编程的射频接口。

为了研究大规模 MIMO 系统，应该让多个 WARPv3 节点位于相同位置并共享电源、时钟和以太网连接。[莱斯大学 Argos 项目](#)能够满足这一要求。Argos v2 阵列包含 24 个四天线 WARP v3 节点，如图 2 所示。Argos 阵列旨在支持各种大规模 MIMO 实验，而且特别适合同步收集所有 96



图1 – 具有双无线电FMC模块的WARP v3硬件可提供一个FPGA、四个RF接口、存储器和两个以太网连接。



图2 – 莱斯大学Argos v2阵列包含24个四无线电WARP v3节点，具备共享时钟和以太网连接。

个阵列天线的信道测量数据。

Argos 阵列中每个 WARP v3 节点上的 FPGA 都可提供接近射频接口的强大处理能力。在诸如 Argos 这样的大规模 MIMO 配置中，有大量数据需要处理。例如，当接收 40MHz 的带宽时，WARP v3 上的每个射频接口

都会产生 960Mbps 的采样流（双 12 位 40Msps 的 ADC）。完整 Argos 阵列的总量是该值的 96 倍，远远超出可发送到 PC 和实时处理的最高值。相反，该系统能够利用 FPGA 实时对这些数据进行本地处理，并显著减少上游处理器的负担。这种实时处理能

力对于我们的大规模 MIMO 信道特性描述设计来说非常重要，因为这样系统就可以不断地测量信道，并可靠地观测小于毫秒的信道特性变化。执行该处理的 FPGA 定制设计正是针对 WARP v3 的 Mango Communications 802.11 参考设计。

该参考设计是 802.11a/g 介质访问控制层（MAC）和物理层（PHY）的实时 FPGA 实现方案。该设计用来与标准 Wi-Fi 设备进行交互操作，起到接入点（用于 Wi-Fi 客户端）、客户端（访问 Wi-Fi 接入点）或者监控器（只能接收的网络活动无源观测器）的作用。您可以对 MAC 和 PHY 进行定制，以探索从标准配置衍生出的变体方案。这种互操作性与可扩展性的完美结合能够实现各种规模的无线通信和网络实验。WARP v3 硬件用户可免费获得 802.11 参考设计的完整源文件。

图 3 给出了参考设计架构。该设计使用两个赛灵思 MicroBlaze™ 内核在软件中实现高电平和低电平 MAC 协议。MAC 连接两个 FPGA 核，分别用于实现 PHY 发送器和接收器。我们在赛灵思系统生成器中实现了这些 PHY 内核。发送器内核实现了完整的字节至波形流水线，可从 MAC 读取数据包有效载荷，创建 OFDM 波形，并将波形驱动至射频接口 DAC。该流水线包含编码、扰码、交织、IFFT 和前同步码插入。MAC 指定每个数据包的调制和编码速率；支持 802.11a/g 规定的全部 8 种数据速率。

接收器设计则实现了完整的波形至字节流水线，包括 AGC、同步、FFT、信道估计、均衡、检测和解码。接收器使用数据包 SIGNAL 字段中的 RATE 值自动针对每个数据包配置解调和解码模块。接收器能以足够

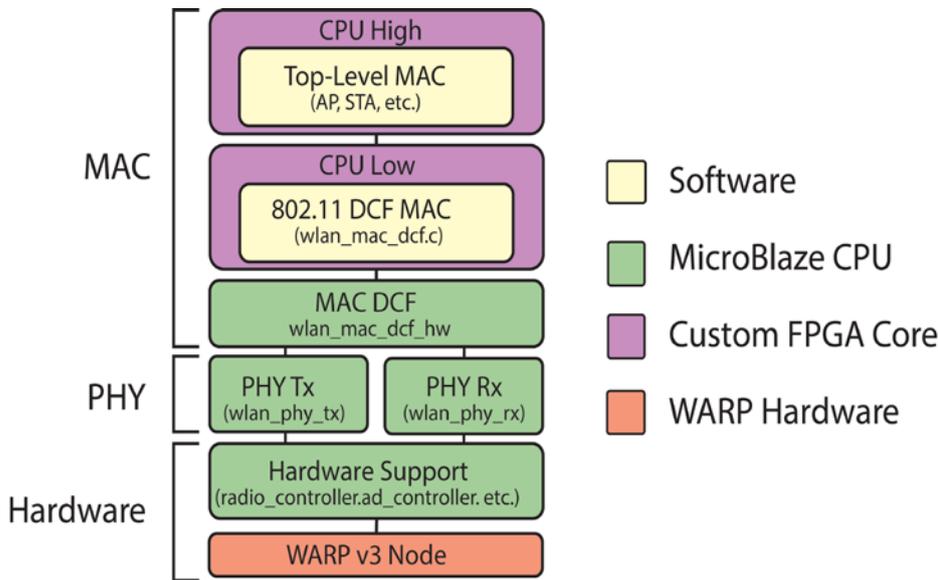


图3 – Mango 802.11参考设计架构包含两个用于MAC的赛灵思MicroBlaze CPU以及PHY发送器和接收器系统生成器定制内核。

快的速度解码任何速率的数据包，并在接收后发出确认字符 (ACK)，以满足标准对于接收至发送 (Rx-to-Tx) 之间严格的转换要求。

接收器中的信道估计子系统对大规模 MIMO 信道的特性描述来说特别重要。在标准 OFDM 接收器中，信道估计器生成每个副载波的复杂信道系数。均衡器使用这些估计系数为每个接收到的数据符号校正信道幅度和相位衰减问题。此外，我们的设计还将每个接收数据包信道估计值副本保存到片上存储区。MAC 将这些信道估计值作为关于接收帧和标准信息（例如接收器功耗、AGC 增益选择、校验和状态以及天线选择等）的额外元数据。然后，信道估计值会被复制到更高水平的 MAC 以备进一步处理。我们的特性描述平台从 Argos 阵

列中各个节点所接收到的每个数据包中收集这些估计值，以构成对大规模 MIMO 传播环境的实时视图。

WARPNET 实验框架

大规模 MIMO 特性描述系统的最后一个部分是 WARPnet 框架，用于运行大型 WARP 节点网络的实验。WARPnet 是一个 Python 定制软件包，使用专用控制连接功能连接至多个 WARP 节点。该框架允许 Python 脚本在 PC 上运行，以便实时地远程配置实验参数并检索实验数据。WARPnet 通过每个 WARP v3 板卡上的次级以太网连接与 Mango 802.11 参考设计进行交互。上部的 MicroBlaze 器件用于处理 WARPnet 命令，使框架能够直接访问节点的高电平 MAC 状态以及从低电平 MAC 和 PHY 传送

上来的全部数据。

WARPnet 框架针对我们的大规模 MIMO 信道特性描述设计，维持着与 Argos 阵列中每个节点的连接。每个节点都被配置为 802.11 监控器，可捕获来自每个接收数据包信道估计值，并通过以太网下载这些数据包以备进一步分析。

WARPnet 的完整 Python 源代码在 WARP 库中以开放源码方式提供。

理解 MU-MIMO

采用 MU-MIMO 技术的基站需要为众多发射天线创建波形，这些天线与无线信道相结合时可同时为多个用户发送数据。创建多用户波形需要在基站上执行精细复杂的处理。我们现已提出很多种 MU-MIMO 技术。对 MU-MIMO 设计的普遍要求是准确掌握每个基站天线到每个客户设备间的无线传播特性。

有一种被称为“迫零”的 MU-MIMO 方法，与单用户技术相比，其无论从理论上还是从近期的实践上^[1]都可实现显著的性能提升。迫零方法可最大程度地提高每个客户接收天线上的信号与干扰加噪声比 (SINR)。将 SINR 实现最大化需要最大程度地提高达到用户天线的波形中的用户有效载荷的信号功率 (即“S”)，同时尽量减小其他用户的有效载荷功率 (即“I”)。迫零法需要在基站上进行非常精细复杂的处理。采用迫零法，在计算特定基站天线的传输波形时需要了解每个用户的有效载荷以及从其它天线到每个用户的无线信道。计算复杂性会随着基站天线数量的增加而显著提高。

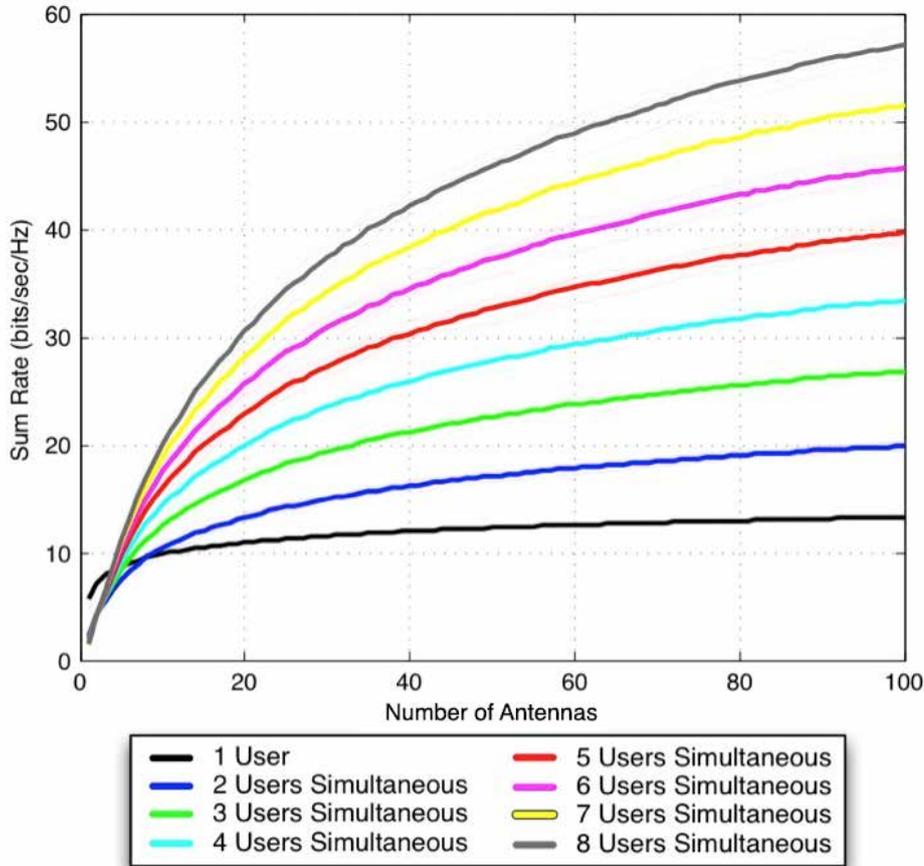


图4 - 多用户MIMO网络的仿真表明, 当接入点安装足够多的天线时可为多用户实现显著的速度提升。

共轭波束形成^[1]是另外一种 MU-MIMO 替代技术。使用这种方法时, 基站会最大程度地提高发送到每个客户端设备的有效信号功率, 而不会主动将干扰功率最小化。共轭波束形成法通过将信号功率最大化 (SINR 中的“S”) 同时忽略干扰功率 (SINR 中的“I”) 的方式增加每个用户的 SINR, 理论上这种方式会随天线数量的增加而改善。此外, 进行每个传输天线波形的共轭波束形成计算时, 无需知道其它天线的信道特性。以上因素综合起来使得共轭波束形成法特别适用于大规模 MIMO 系统, 其中基站天线数量比用户要多得多。

考虑一下经典的 Shannon 信道容量公式 $C = \log(1 + \text{SINR})$ 。无线信道的容量 (比特 / 秒 / Hz) 与 SINR 成对数增长关系。当系统添加更多用户和天线时, 多用户共轭波束形成法存在两个相互矛盾的因素。首先, 多个天线的存在会增加接收信号功率, 因为每个天线都可旋转相位, 使传输信号在用户接收器端被有机地结合起来。第二, 独立用户的多传输会增加干扰功率。叠加的干扰信号会随机组合。随着天线数量的增加, 有机组合的信号功率增长速度超过随机组合的干扰功率, 从而增大整体的 SINR。

图 4 中的仿真结果显示, 在使

用共轭波束形成法时, 基站天线数量的增加对整体网络容量的影响。

仿真假设网络中有一个基站和八个用户, 用独立且恒等分布的瑞利 (Rayleigh) 衰落信道对无线信道建模。仿真实验给出了同时服务一至八个用户时的总体网络速率与基站所用天线数量的对比情况。当天线数量很少时, 我们发现一次对一个以上的用户采用共轭波束形成法并没有好处。如果基站只限制在几个天线, 那么带分时功能的传统单用户波束形成法可能优于多用户共轭波束形成法。随着天线数量的增加, 可以支持更多的用户, 以获得整体网络速率的显著提升。

该仿真实验使用理想的信道模型展示说明多用户共轭波束形成法可实现性能提升。性能提升是否能够在真实系统中实现取决于基站与客户端设备之间真实的无线信道。我们的 MU-MIMO 信道特性描述平台可实时测量基站与真实用户设备之间的信道, 为评估 MU-MIMO 技术的真实性能提供了强大的工具。

完美集成

既然我们已经了解了测量大规模 MIMO 信道的目的, 以及 Rice Argos 阵列、WARP 硬件和 Mango 802.11 参考设计所提供的工具, 我们接下来就看一看如何将这内容组成完整的实时大规模 MIMO 信道特性描述平台。

Argos 阵列中的 24 个 WARP v3 节点由 Mango 802.11 定制参考设计版进行配置。这个版本以只接收监控器模式工作, 使节点的四个天线都接收数据包。每次接收数据包时, 节点都会为每个副载波估算复杂信道系数。

对数据包进行解码并通过以太网发送数据包头和信道估计值以进行分析。这个处理流程在阵列中所有 24 个并行工作的节点中执行。

为了与标准 Wi-Fi 设备通信，信道测量平台必须实现一个标准的 802.11 接入点。使用另一个 WARP v3 节点来达到此目的，使 Mango 802.11 参考设计运行于 AP 模式。这个 AP 节点作为 Argos 阵列中的第 25 个节点。AP 可提供开放的 Wi-Fi 网络，接受商用 Wi-Fi 设备的关联，并通过其主以太网连接提供互联网接入。

这是 Mango 802.11 参考设计中 AP 的标准特征。为了实现实时信道测量，AP 需要实施一个附加功能。每次有 Wi-Fi 客户端加入或离开无线网络时，AP 节点会通过 WARP v3 板卡上的副以太网连接发送一个以太网数据包。信道分析应用程序（下面将介绍）使用这些关联更新来维护活动客户端本地列表。

客户端传输

在从 Wi-Fi 商用设备接收到的数据包中收集信道估计值时所面临的一个重要挑战是确保设备的信息发送频率足够高。现代 Wi-Fi 设备经常采用严格的节电方案，在没有应用程序请求网络接入的情况下会关闭 Wi-Fi 连接。这些设备会定期与接入点报到，但其频率有可能不足以确保获得最新的阵列信道估计信息。

我们用两种方法解决客户端传输频率不足的问题。首先，修改由平台 AP 发送的信标中的流量指示图 (TIM) 字段，以通知所有连接客户端有新的数据包正排队等待。TIM 字段一般用来帮助客户端实现节电效果，使客户端在纯接收模式下短暂唤醒，以接收信标，解码 TIM，并在无流量等候时恢复到低功耗模式。通过在每个信标中的 TIM 字段中列出每个节点，可以让节点较少地进入休眠模式。

第二种方法是使用客户端设备

发送的 ACK 数据包来请求客户端传输。该阵列可以从客户端发送的任何数据包（包括短 ACK）中提取信道估计值。不过，802.11 ACK 数据包只包含一个目标 MAC 地址，通常会阻止阵列识别正在传输的客户端。

我们在 802.11 MAC 规范中使用一个特殊方法来解决这一问题。该标准要求 802.11 器件在成功接收发送到该客户端的单播数据包后发送一个肯定的确认数据包。即使数据包源地址无法识别，“强制 ACK”要求仍然适用。因此，AP 可发送一个采用唯一虚假源地址的数据包，以便让包含唯一标识符的客户端触发一个 ACK 发送信号。客户端收到数据包后会将 ACK 发送至 AP 所使用的唯一虚假源地址。阵列节点接收 ACK，并明确地将得到的信道估计值与发送客户端进行关联。这个技巧非常好用，能够在阵列上为信道估计触发频率更新。之所以能使用这种方法，是因为 Mango 802.11 参考设计具备完全可编程性。

实时分析

大规模 MIMO 信道测量平台的最后一个组成部分是定制应用程序，该应用程序能够收集阵列信道估计值，计算出可达到的多用户容量，并实时显示结果。我们采用 Objective C 开发该应用，使用本地 UDP 插口连接阵列的 WARP v3 节点，并利用 OS X 图形框架绘出结果。

该应用程序有两个主视图。第一个视图显示每个阵列天线针对各个副载波所收集的信道数量，共 4,992 个数据点（52 个副载波 x 96 个阵列天线）。该视图显示由阵列收集的原始信道数据，并主要负责传达由每个阵列天线观察到的大量信道值。图 5 给



图5—定制MU-MIMO信道分析应用中的信道数量视图。每个条形图代表每天线每副载波的信道数量

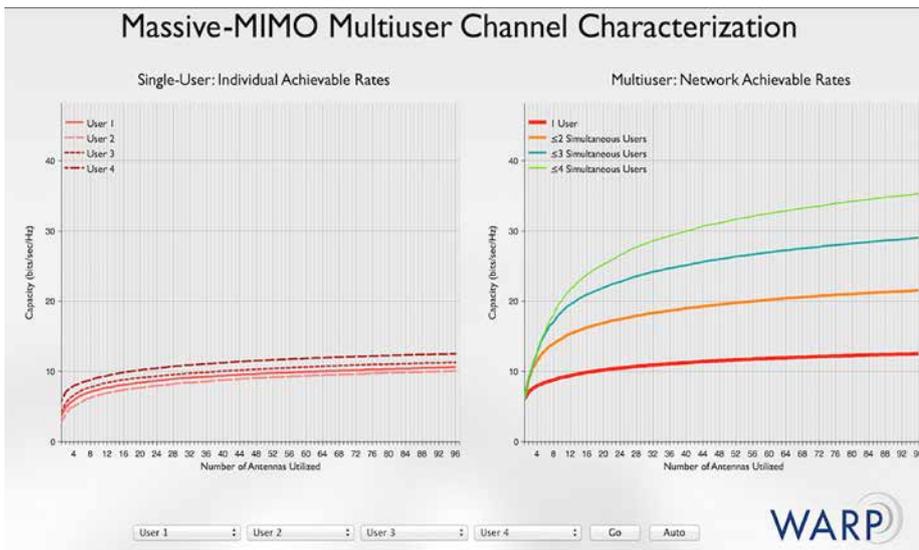


图6 – 利用阵列收集的真实MU-MIMO信道测量结果估算单用户和多用户方案的网络容量，并显示在我们的定制分析应用程序中。

出了该视图的屏幕截图。实际上，该视图会进行实时更新（10 帧每秒更新活动 Wi-Fi 客户端）。

应用程序的第二个视图显示了根据阵列信道估计值得到的容量计算结果。在图 6 中给出该视图。执行两个容量计算。第一张图描绘了每个用户的容量与所用阵列天线数量的对比情况。图中的每条线近似等于可达到的单用户下行链路容量，并假设阵列在传统的单用户波束形成配置下使用部分天线。每条容量曲线斜度随天线数量增加而减小，这清楚地说明在使用传统单用户无线技术时多天线优势会减小。

第二张图显示了阵列使用部分天线实现下行链路多用户波束形成技术时的总体网络容量。四条曲线的走势清楚体现了在使用多用户技术时更多天线数量所带来的优势。当服务更多用户时斜率增加体现了采用多用户波束形成技术时所实现的“日志外

(outside the log)”（常在 MIMO 中用“prelog”表达）网络容量增益。

我们在 2 月份举行的 2014 赛灵思新兴技术座谈会 (ETS) 上演示了大规模 MIMO 信道测量平台。演示具体内容，包括视频和补充材料链接均在 <http://warpproject.org/XilinxETS> 中提供。

参考资料

1. T.L. Marzetta, “Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, 2010
2. C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang and L. Zhong, “Argos: Practical Many-Antenna Base Stations,” *Proceedings of ACM MobiCom*, pp. 53–64, 2012
3. Q. Yang, X. Li, H. Yao, J. Fang, K. Tan, W. Hu, J. Zhang and Y. Zhang, “Bigstation: Enabling Scalable Real-time Signal Processing in Large MU-MIMO Systems,” *Proceedings of ACM SIGCOMM*, pp. 399–410, 2013

Xilinx携手NXP降低无线基础架构无线电的资本和运营支出

2014 年 11 月 25 日，中国北京 - All Programmable 技术和器件的全球领先企业赛灵思公司 (Xilinx, Inc. (NASDAQ:XLNX)) 和恩智浦半导体公司 (NXP Semiconductors (NASDAQ:NXPI)) 联手宣布，他们将携手降低无线基础架构无线电的资本支出 (CapEx) 和运营支出成本 (OpEx)。两家公司一直保持通力合作，致力于让客户能够快速方便地将赛灵思最新峰值因数抑制 (CFR) 和数字预失真 (DPD) SmartCORE™ IP 与 NXP 的 Gen9 LDMOS 射频高效功率放大器技术完美集成。NXP 先进的功率放大器器件和赛灵思的 All Programmable 器件及无线电 IP 的无缝组合，使客户能够实现更小、更轻、可靠性更高的无线电设备，非常适用于新一代无线基础架构设备。

高效功率放大器实现了同样出色的无线电输出功率，而且通过使用额定功率较低的设备以及减少冷却机械装置数量，还能够节省资本支出。赛灵思的 Zynq®-7000 All Programmable SoC 是业界首款支持最新一代 CPRI (线路速率为 10.1 Gbps) 和 JESD204B (线路速率为 12.5 Gbps) 标准的软硬件可编程器件。Zynq 器件还支持 CFR 和 DPD 等高级信号处理算法与 ARM® 处理器控制层软件的集成，可满足远程无线电、分布式天线系统和中继器产品等的需求。此外，较小的散热片能够减少电源的复杂性和重量，有助于设计人员在单个芯片上实现完整的数字无线电。

除了以单个 RAT 配置支持 MC-GSM 跳频或以多个 RAT 配置支持高达 75 MHz 的射频带宽外，赛灵思的最新 CFR 和 DPD IP 现在还能以单个 RAT 配置支持高达 100 MHz 的射频带宽。赛灵思的无线电 IP 结合 NXP 面向小型、高效、高性能 LTE 基站的 Gen9 系列射频功率晶体管，使客户能够在更短的时间内用更少的资源实现先进的无线电基础架构，同时在 2.6GHz 带宽下将功率放大器效率提高到 50%。

视频处理领域的出色表现

利用Zynq SoC简化 4K电视开发

作者: Roger Fawcett

OmniTek 总经理

roger.fawcett@omnitek.tv

赛灵思All Programmable技术堪称4K视频系统设计人员的福音。相关工具、IP和参考设计将为FPGA设计新手助一臂之力。

超高清（UHD）电视因其分辨率等级同时也被称为4K电视。这种电视已经广泛推出，而且事实证明，4K技术比3D电视更受消费者青睐。但是，相关标准却落后于4K电视的增长趋势。美国电影电视工程师协会（SMPTE）支持4K60视频的6 Gbps和12 Gbps SDI标准才刚刚发布；支持相同分辨率的HDMI™ 2.0和DisplayPort接口尚处于应用的早期阶段。由于4K UHD电视的消费需求巨大，许多专案标准已经蜂拥而来，试图填补这一空白。

事实上，有关4K UHD电视的很多部分正处于不断变化的状态，因此系统一定要有足够的灵活性来适应发展中的标准。确保这种灵活性的方法是将在此类设计中使用已久的传统芯片组和ASSP替换为FPGA和All Programmable片上系统，例如赛灵思Zynq®-7000 All Programmable SoC。这种解决方案不仅可提供所需的灵活性，同时还能实现与ASIC相媲美的性能水平。

同时，最新FPGA和SoC的尺寸和性能也存在相当大的设计挑战，尤其对于不是特别擅长FPGA的工程师来说更为如此。硬件设计与FPGA实现方案之间虽有很多相似之处，但是基于FPGA的系统通常要涉及更多的组件。此外，固件设计中固有的灵活性也增加了更多复杂性。

幸运的是，赛灵思为4K电视设计人员提供了很多帮助，与从头开始设计系统相比能节省大量时间和金钱成本。不过在具体介绍如何在4K领域应用FPGA技术之前，我们首先看一看4K系统如何如此快速地变得如此受欢迎，以及任何4K系统都必须解决的问题。

4K的优缺点

自电视问世以来，业界一直在通过增大分辨率、帧率或图像动态范围（即所能达到的亮度）持续改进画质，实现更大、更好、更快速的视频，让图像更逼真。当然，还包括实现真正的3D效果，至少更有一种身临其境

的感觉。

增大分辨率可以使图像更精致，而且在更大屏幕上显示时不会让像素纹理变得明显。更大的屏幕有更强的沉浸式感受。这些改进很容易得到消费者认可，因此也就愿意在这方面花钱。通过增大帧率（更流畅的动作）或动态范围（光线更亮，黑色更暗）实现的改善效果虽然很出色，但到目前为止在吸引消费者注意力方面所起到的效果相比之下并不明显。

最新 4K UHD 电视像素是此前广受欢迎的高清标准的四倍。对消费者来说最重要的一点是利用 4K 技术升级到更大尺寸电视，可以享受更好的沉浸式效果，而且不会对图像质量产生任何明显影响。

然而，在开发支持 4K 视频的系统时面临着诸多固有技术挑战。首先，高达 60Hz 帧率下提供的 3,840 x 2,160 像素的帧尺寸相当于 600MHz 的像素率。因此，需要使用非常高性能的系统对其进行实时处理。其次，针对 4K 定义了不同传输配置，所有配置都涉及多个数据流，有些在相同线缆上以多路复用方式传输，有些在

不同线缆上传输，另外还出现很多不同传输技术，包括：4x3G；6G-SDI 和 12G-SDI；HDMI 1.4 和 2.0，DisplayPort 1.2，以及 V-by-One HS。

设计人员面临的另一个问题是任何系统不仅需要处理 4K 标准，而且还要处理当前使用的很多或全部视频标准（包括 SD）。此外，系统必须支持不同标准之间的转换，会遇到很多相关问题：上/下/交叉转换、不匹配的色彩空间、颜色校正、隔行扫描与去隔行扫描以及 Cadence 处理。另一个复杂问题是在完成上转换后通常需要紧接着使用所谓“超分辨率”增强技术来抵消不可避免的图像平滑问题。

其它处理还包括降噪、裁剪和调整大小，这些都需要实时完成。有些系统可能还需要处理高带宽数字内容保护（HDCP）。

此外，如果想要确定广播传输的质量还需要生成适当的眼图和抖动显示图，但相应技术在高比特率下越来越难以实现。

第一层辅助：4K IP 核

设计任何系统时的第一步都是找到可

运用到设计中的现成模块。在 FPGA 领域，等同于构建块就是可内置到 PCB 设计中的各种芯片，即 IP 核。因此，第一步要确定您的 4K UHD 设计中可以使用什么 IP 核。

OmniTek 对于各种类型的视频系统设计而言都是很好的 IP 核供货源。该公司是赛灵思联盟计划的认证成员，起初作为专门从事自主开发测试测量系统的开发商，具有丰富的视频处理经验。这些系统需要专用硬件，因此也就需要开发专用的固件模块。固件模块现在也以 IP 核的形式提供。OmniTek 在推出最新测试测量系统（即新的 Ultra 4K Tool Box）之后，就开发出多种支持 4K 技术的 IP 核，现已向第三方开发商供货。

有两种内核对 4K 系统设计人员来说尤其有用，它们就是 OmniTek 的 OSVP v2 可扩展视频处理器（OSVP v2 Scalable Video Processor）和多通道流媒体 DMA 控制器（Multi-Channel Streaming DMA Controller）。这两种内核均采用 ARM® AMBA® AXI4 系统互联标准，都可用于赛灵思 7 系列 FPGA 和 Zynq SoC。

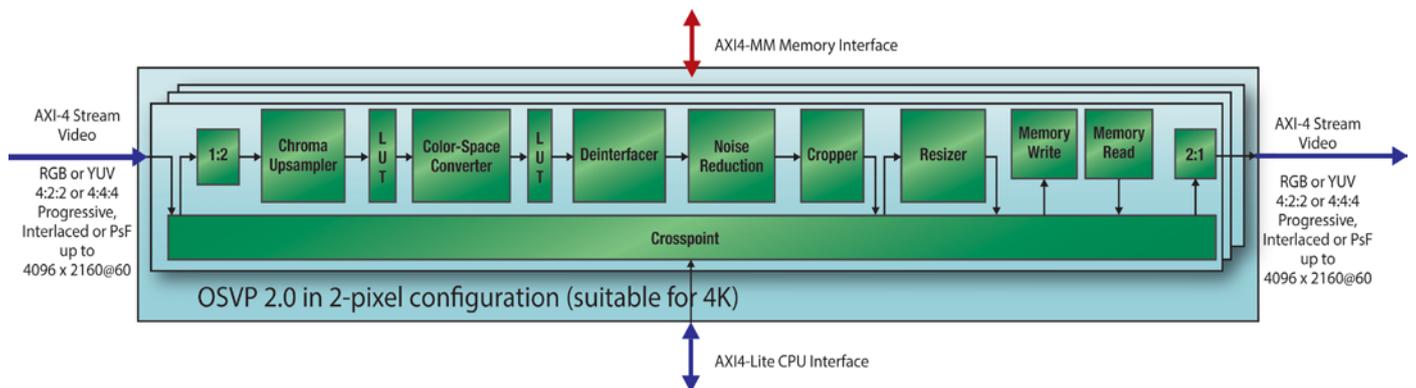


图1- OmniTek的OSVP v2可扩展视频处理器内核的输入通道架构

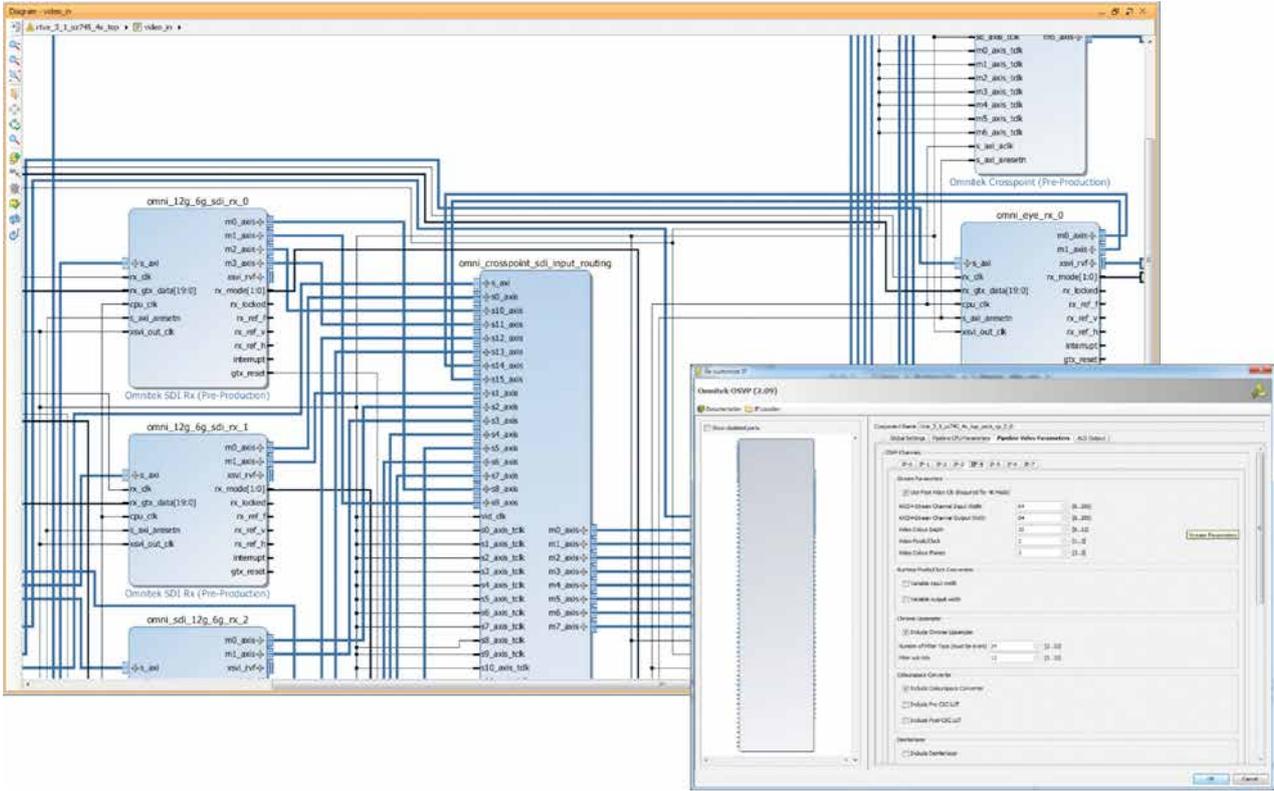


图2 – 在Vivado IPI中布局的RTVE 3.1视频设计以及配置窗口

OSVP v2 的功能包括：六轴色彩校正；运动及边缘自适应去隔行（包含 3:2 和 2:2 Film Cadence 检测与处理）；可通过图像锐化和平滑调整大小和进行裁剪；以及降噪等。图 1 给出了该内核的方框图。您可以在编译时间对所包含的处理功能的选项进行配置，同时，由 OSVP v2 内核所执行的处理工作的更详细内容可在运行时间内或这从软件中设置。

OSVP v2 内核是套件中的一部分。该套件还包含用来合并多个视频流的合并器；用来产生隔行格式输出的隔行扫描器；专用交叉点；以及用来在 4:4:4、4:2:2 和 4:2:0 YCbCr 之间转换的色度重采样。（单个色度重采样能够从 4:4:4 转换至 4:2:2，从 4:2:2 转换至 4:2:0，从 4:2:2 转换至 4:4:4，或从 4:2:0 转换至 4:2:2。）

单个 OSVP v2 内核能处理多个视

频通道。但这会受到实现内核所用的 FPGA 或 SoC 能提供的资源，以及可用 SDRAM 带宽的限制。例如，您可对采用 Kintex®-7 XC7K325T FPGA 实现的 OSVP 内核进行配置以支持八个输入，处理采用八种不同 HD 视频标准或八种色彩空间的视频。同时，您可对输出模块进行配置以提供 16 个渐进 HD 输出。此外，您还可以让输出模块提供单个 4K 通道，或者四通组，以便共同提供 Square Division (“quad”) 或双像素采样交错格式组成的 4K 视频。

设计复杂 4K 系统时的另一个挑战是对处理视频时所需的众多高带宽存储器访问进行管理。有时候，所需的视频操作功能与视频处理模块一起提供。例如，OSVP v2 内核包含一个多端口视频 DMA 模块，该模块提供一个用来操作视频输入和输出的高效引擎。

然而通过 PCI Express® 采集和播放单通道或多通道 4K60 视频，需要一个 DMA 控制器，用于优化处理 PCIe® 接口传输的流式数据。OmniTek 的多通道流式 DMA 控制器有两项功能在这里很有用。首先是基于 FIFO 的 DMA (FDMA)，这样就无需将数据从存储器转入和转出。第二个是一系列的设计优化，使得控制器可以高效利用 PCIe 带宽，例如预取分散 / 聚集模式描述符，以及背靠背 TLP 数据包打包。

OmniTek 针对 4K UHD 视频处理开发了另一个 IP 核，可以将双采样交错格式组成的 4K 视频分解成不同数据流。另外，对基本 MIG SDRAM 控制器进行了替换，从而进一步提高了 UHD 电视视频应用的性能。

可编程优势

赛灵思对基于 FPGA 和 SoC 的 4K 视频系统设计人员从三方面做了进一步支持。

第一大优势在于 Zynq SoC，其为高性能视频或图像处理提供了强大的软硬件组合处理能力。Zynq SoC 在单个器件中集成了功能丰富的双核 ARM Cortex™-A9 处理系统以及 7 系列（28nm）FPGA 可编程逻辑。用户既可在 ARM 处理器上运行处理算法，也可在需要加速时将处理任务转给 FPGA 硬件，实现实时运行。

Kintex-7 FPGA 和 Zynq SoC 的可编程逻辑提供的 300MHz 可持续视频处理速度，结合 1,600 Mbps 的 64 位 DDR3 存储器性能，对于操作 4K 视

频处理和 4K 帧缓冲非常重要。Zynq SoC 中具备丰富 DSP 功能的可编程逻辑架构为 DSP 设计人员提供了一个用以实现信号处理算法的高度灵活的平台，同时，处理器与可编程逻辑之间的紧密耦合有助于开发跨越两个领域的编解码算法。以 Zynq SoC 作为设计的基础可以将组件集成到单个器件中，而无需使用多个 ASSP，因此能够节省功耗和成本。

赛灵思还为 4K 视频系统开发提供强大的连接支持，包括 FPGA 和 SoC 中的多个内置收发器，以及众多自己开发的连接 IP。例如，Zynq 7045 SoC 提供多达 16 个 12.5Gbps 收发器，并符合 12G-SDI、6-Gbps HDMI 2.0、5.4-Gbps DisplayPort 1.2 以及 10-

Gbps 以太网标准。

赛灵思所做的第三个重要贡献是与 Vivado® 设计套件有关的 IP Integrator (IPI) 工具。如图 2 所示，有了 IPI 工具，连接 IP 模块的任务就变得类似于在印刷电路板上连接芯片。符合 AMBA AXI4 互连协议（赛灵思已经采用为接口标准）的 IP 块接口处理将相当简单（如连接 OmniTek OSVP 和 DMA 模块）。

赛灵思最新 UltraScale™（16-nm/20-nm）技术的推出还将提供更强大的功能。该技术支持的时钟速度高达数百 Gb/s，并被称为“ASIC 级”。（如需了解更多信息，敬请访问：china.xilinx.com/products/technology/ultrascale.html）。UltraScale

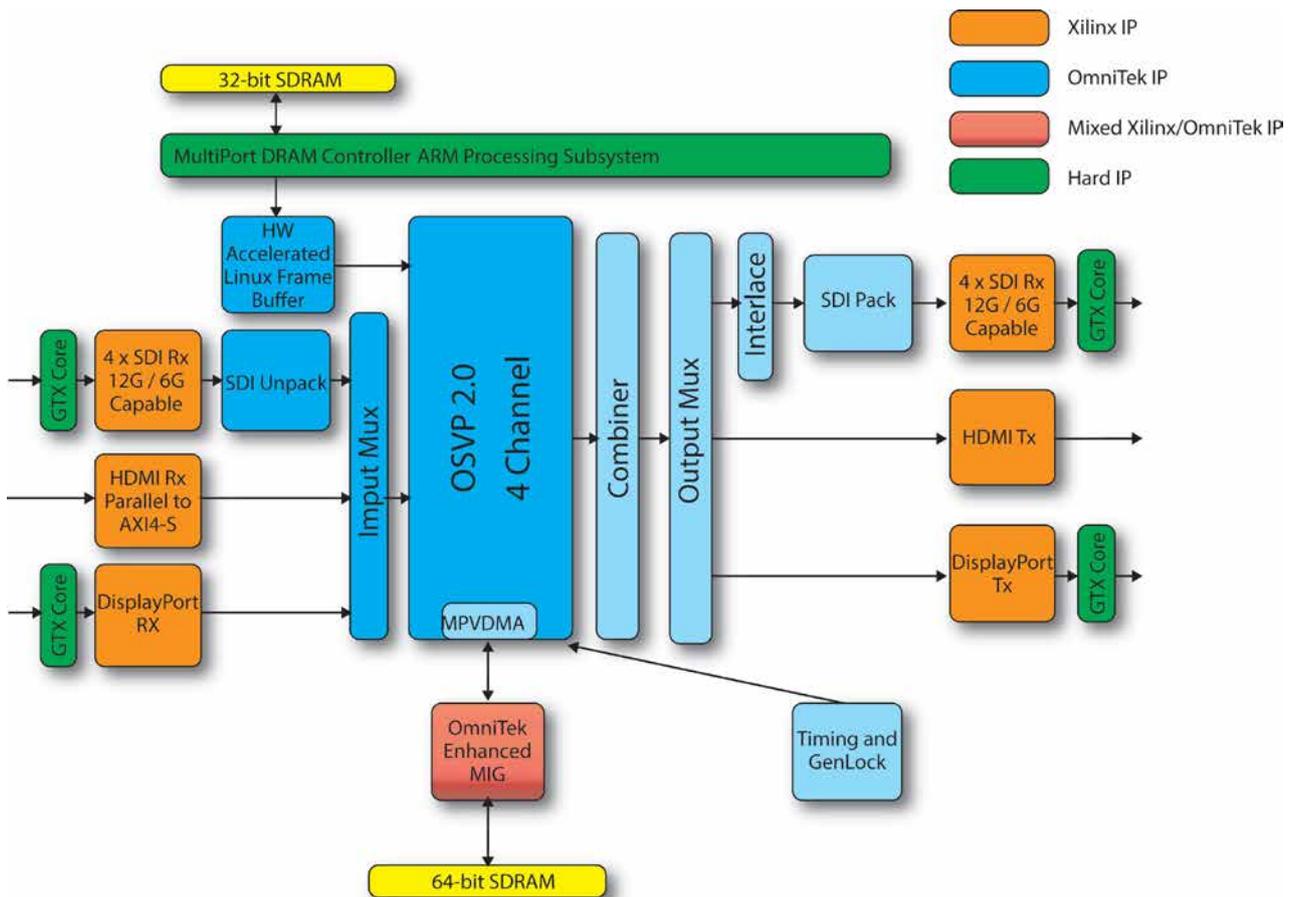


图3 - RTVE3.1参考设计的方框图



图4 – 打开后的OmniTek最新Ultra 4K Tool Box，展示里面的组成部分

架构不仅支持 4K 视频系统的开发，还将支持更高的 8K 系统。

可供修改的现成系统

尽管 IP 核所提供的构建块在简化视频系统设计的创建方面发挥了很大作用，然而，如果对现成的系统进行修改，则能获得更高起点。较长时间以来，赛灵思一直提供包含赛灵思视频与连接 IP 以及来自 OmniTek 的 IP 模块的实时视频引擎（RTVE）参考设计。这些参考设计提供针对广泛视频应用领域的高清晰度广播级视频处理功能。RTVE 参考设计不仅展现出这些 IP 模块的功能性，还能展示出方便的互操作性，因为这些模块都根据 AXI4 互联标准而设计。

RTVE 的每个新版本都会加入最新的 IP 模块，以扩展参考设计功能。最

新版本 RTVE 3.1 增加了对 4K 视频标准的支持，包括 SMPTE 425-5:2014、DisplayPort 1.2、6G-SDI 和 12G-SDI。图 3 给出了该设计的方框图。

RTVE 3.1 设计增加了以上介绍的 OmniTek 内核，以及 OmniTek 隔行扫描器、合并器和专用交叉点，以及赛灵思的一些关键组件。它还提供 API 和应用，从而可通过从基于网络的接口驱动 RTVE 引擎。RTVE 3.1 参考设计固件和应用软件都以源文件的形式提供给客户，既可用于演示如何使用这些工具设计系统，也可作为开发类似系统的起点。

另外，还为客户提供 RTVE 3.1 的硬件平台。该平台包含 OmniTek OZ745 开发套件（基于赛灵思 Zynq 7045 SoC）和 FMC 扩展卡。FMC 增加了符合 DisplayPort 1.2 标准

的输入和输出端口，以及两个 SD/HD/3G/6G-SDI 输入和输出。凭借这些 I/O 端口可将所支持的视频标准进行扩展，包括 6G 4K 和 12G 4K；3G 等级 A 和 3G 等级 B Square Division/Quad 4K；以及 3G 等级 A 和 3G 等级 B 双采样交错 4K。

OmniTek 的 Ultra 4K Tool Box（图 4）能够证明这些组件紧密结合加在一起能够构建出可行的商用系统。该 4K Tool Box 在基本架构中包含了 OmniTek 的 OZ745 开发套件、FMC 卡以及 RTVE 3.1 的固件及相关应用软件。该 4K Tool Box 不仅提供对所有视频标准（最高 4K60）的上/下和交叉转换以及相关图像校正功能，而且提供多种显示功能，包括为构成 4K 图像的所有数据流生成的眼图和抖动视图、色域视图和像素数据显示。

Ultra 4K Tool Box 是市场上的新产品，但已吸引所有 4K 处理领域中的广大客户购买，例如芯片组制造、测试测量以及广播领域的客户。这表明整个视频行业对最新 4K 标准的兴趣所在。

终极辅助

除了这些工具和 IP 以外，OmniTek 还提供咨询服务，以帮助客户建立和运行 4K 设计。赛灵思的领先芯片技术和软件工具，与 OmniTek 在视频处理和制造方面的专业知识的强强结合意味着视频系统设计人员可以在完整开发框架的基础上着手进行设计，并获得超过预期的更简便的集成功能和更强有力的支持。其结果就是可以将创新的高竞争力产品更加快速地推向市场。

利用Xilinx FPGA实现 高效并行实时上采样

作者：William D. Richard

圣路易斯华盛顿大学副教授

wdr@wustl.edu

本文介绍一种使用 Virtex-6器件和免费 WebPACK工具实现 实时四倍上采样的 方法。

许多信号处理应用都需要进行上采样。从概念上讲，对数据向量进行 M 倍上采样的最简单方法是用实际频率分量数的 $(M-1)$ 倍个零填充数据向量的离散傅里叶变换 (DFT)^[1]，然后将零填充向量转换回时域^[1,2]。但这种方法计算量很大，因此不能在 FPGA 内部高效实现。在本文介绍的高效并行实时上采样电路中，每个 ADC 时钟可产生 M 个上采样值，其中 M 是所需的上采样倍数。在我们赛灵思的 Virtex®-6 XC6VLX75T FPGA 实现方案中，上采样倍数 M 为 4，可以作为较普遍方法的一个实例。

我们并行上采样方法的总体概念源于某些作者所称的“窗口 SINC 插值”，这种方法在文献资料^[3,4]中有一些非常精彩的专文介绍。

为了更好的说明，现以图 1 中所示的 16MHz 模拟信号为例。该信号的表达式为：

$$f(t) = \cos(2\pi ft) * e^{(t * t)/constant}$$

等式 1

如果用 12 位 ADC 以 80MHz 的频率对图 1 所示的信号进行采样或量化，输入范围为 ADC 完整输入范围的 97.7%，则每个信号周期只能采样五次，产生的样本数据序列如图 2 所示。如果对该示例数据序列进行四倍上采样，则有效采样率为 320MHz，每个信号周期能够提供 20 个样本。虽然您可以使用本文介绍的方法进行更高倍数的上采样，但为了说明起见我们还是使用 $M=4$ 的上采样。

当然，还可以通过直接在 ADC 生成的数据序列的每个实际样本值之间插入 $(M-1)$ 个零来获得上采样数据向量及所需的样本数（公认效果较差）。该“零插入步骤”对应于复制频域中的原始信号频谱。通过对得到的“零填充”时域信号进行低通滤波，就能去除频域中所需

频谱的“复本”，从而获得上采样数据向量。

FIR 滤波器设计

频域中的理想（砖墙式）低通滤波器相当于在时域中用无限域 Sinc 函数作卷积。因此为近似化所需的卷积运算，可让前述的零填充时域信号通过速率为 ADC 时钟频率 M 倍的对称低通 FIR 滤波器，该滤波器拓扑结构与图 3 所示的示例 31 抽头 FIR 滤波器相同。用这种方法，我们可以实时生成上采样数据向量。图 3 中的 R1、R2、…、R31 代表速率为 ADC 时钟频率 M 倍的寄存器，C0、C1、…、C15 代表 FIR 滤波器的系数。

需要说明的是，图 3 所示的 FIR 滤波器中的大部分寄存器在任何特定时钟间隔内都会包含 0，而非实际的样本

数据。以 M=4 为例，当 R1 包含实际样本数据时，R2、R3 和 R4 将包含 0。当 R1 包含实际样本数据时，R5、R9、R13、R17、R21、R25 和 R29 也将包含实际样本数据，其余的寄存器将包含 0。在下一时钟间隔中，R2、R6、R10、R14、R18、R22、R26 和 R30 将包含实际样本数据。

如图 3 所示，由于在通过 FIR 滤波器的每 M 个样本中有 M-1 个样本为 0，就 M=4、使用 31 抽头 FIR 滤波器的情况而言（如图 4 所示），您可以将滤波器分解开，并行产生 M 个输出。采用这种实现方案，并行 FIR 滤波器的工作频率为 ADC 的基本时钟频率，而非 ADC 时钟频率的 M 倍。

如图 4 所示，您可以设定窗口 Sinc 函数的系数 $C_w(n)$

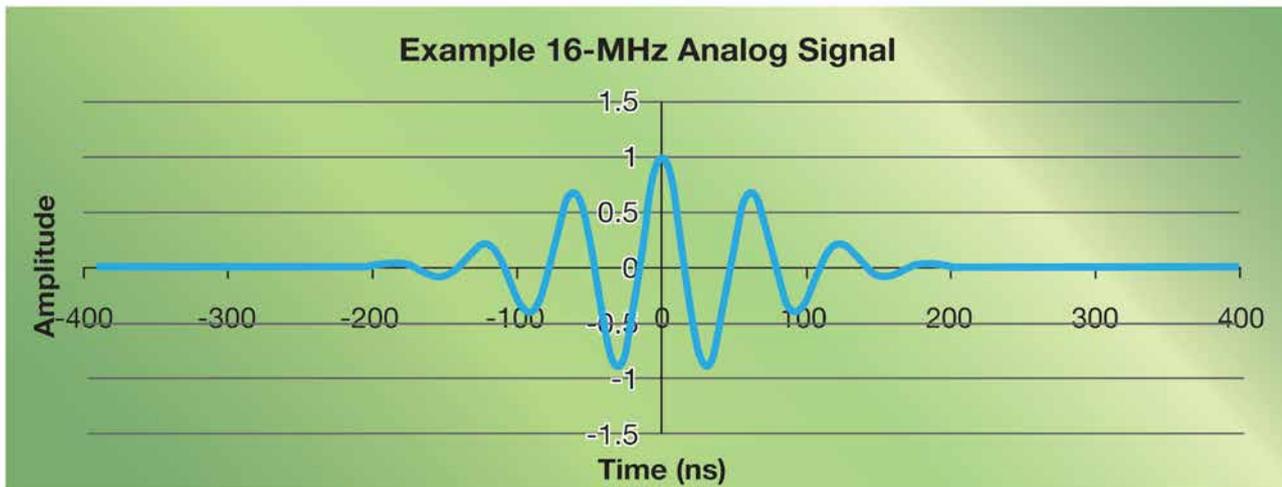


图1 - 展示上采样过程的16MHz信号实例

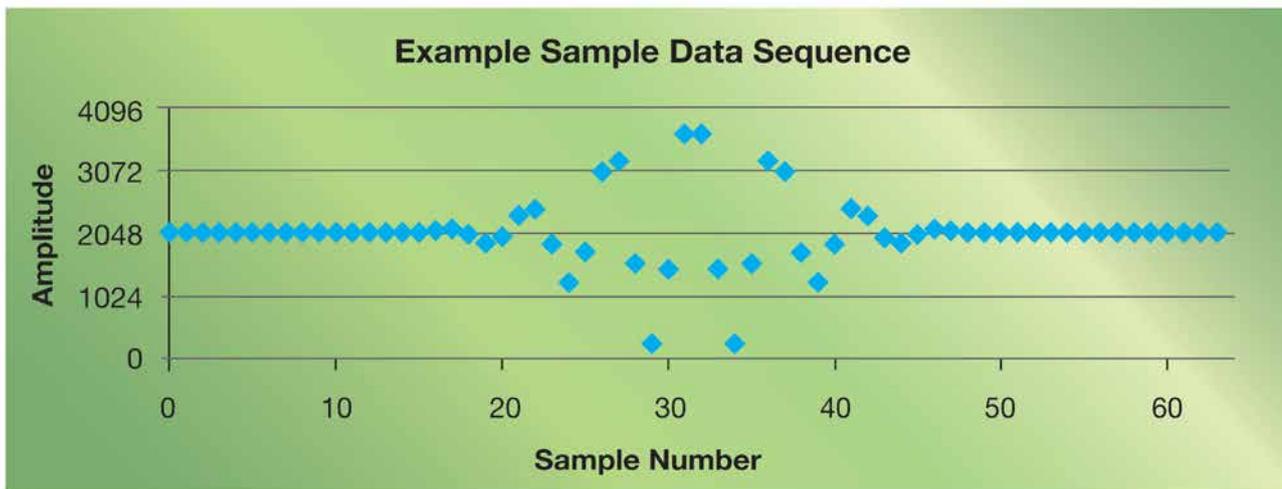


图2 - 本图是12位ADC，输入范围为ADC完整输入范围的97.7%，用80MHz或每周期五次频率对图1的模拟信号采样得到的样本数据序列实例。

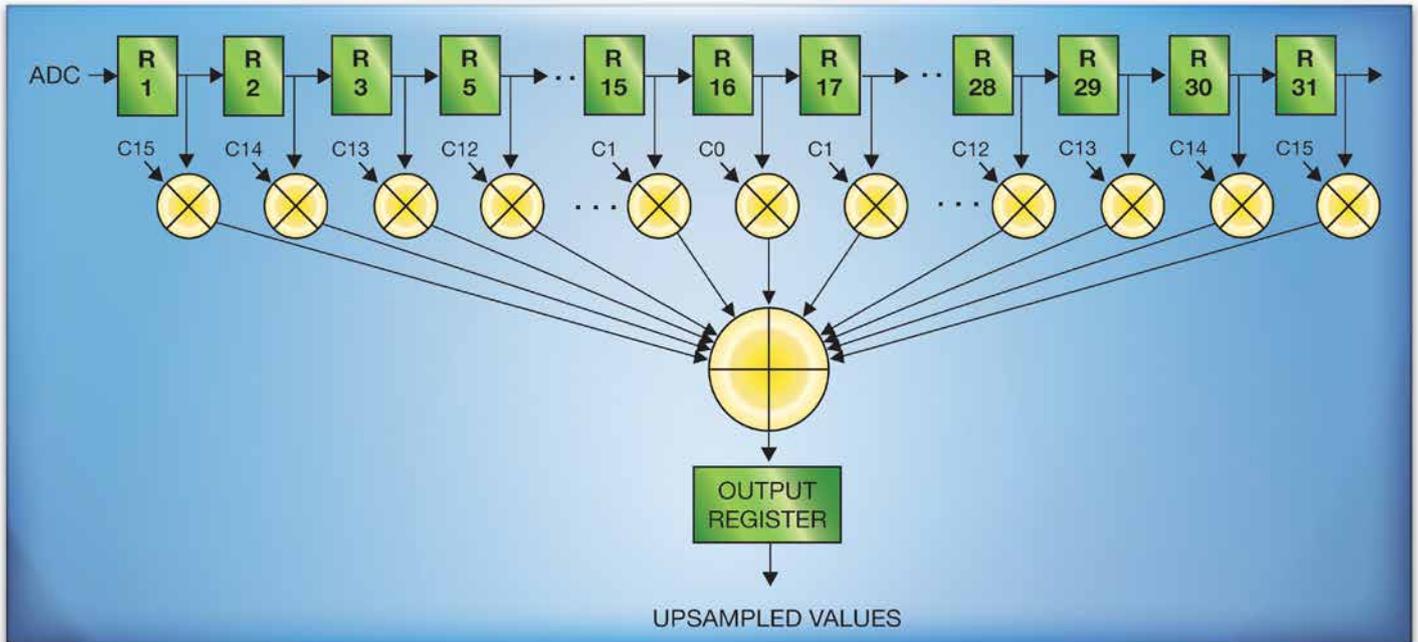


图3 - 当滤波器工作在基本ADC时钟频率的M倍频率时，零插入步骤中使用31抽头FIR滤波器每时钟周期可生成一个上采样数据值

来最大限度地减少实现该 FIR 滤波器所需的乘法器数量。对于 T 个抽头的低通 FIR 滤波器而言，可通过下式得出最佳系数：

$$C(n) = \text{Sinc}[(n * \pi) / M], n = 0 \text{ to } (T-1)/2.$$

等式 2

这里的汉宁窗系数的计算方法为：

$$H(n) = [1 - \text{COS}(2 * \pi * (n + ((T-1)/2)) / (T-1))] / 2, n = 0 \text{ to } (T-1)/2.$$

等式 3

窗口 Sinc 函数系数 $C_w(n)$ 随即通过对应的 $C(n)$ 和 $H(n)$ 值相乘即可得到，如：

$$C_w(n) = C(n) * H(n), n = 0 \text{ to } (T-1)/2.$$

等式 4

在 $M=4$ 时，如果按上面介绍的方法计算 31 抽头 FIR 滤波器的系数，令 $C_0=1.0$ ， $C_4=C_8=C_{12}=C_{15}=0$ ，则无需使用与图 4 中这些系数有关的 9 个乘法器。此外，由于生成 UPSAMPLED VALUE(1) 每个系数会使用两次，用户可以“折叠”该实现方案，比如在相乘之前让 R1 和 R8 相加，这样就可以去掉四个乘法器。最后得到的设计仅使用 18 个乘法器每时钟周期就可产生四个上采样值。需要注意

的是按上文介绍的设计方法，每个原始样本值会原样从并联滤波器输出。

我们使用图 5 所示的可综合 VHDL^[5] 模型评估图 4 所示电路的性能。该 VHDL 实现方案假定样本数据为 12 位，即可能来自美国模拟器件公司的 AD9670 八通道超声前端集成电路^[6]的数据。滤波器系数表达为 25 位定点常数，以与集成到 FPGA 芯片上的乘法器输入位数相匹配。来自 ADC 的输入样本先馈入与输入引脚相连的寄存器（图 4 中的 R1），上采样输出值则使用与输出引脚相连的寄存器。寄存器 R2 到 R8 属芯片内部寄存器。寄存器 R1 到 R8 故意设定为 15 位宽度，以便为综合后逻辑提供执行计算所需的动态余量。该设计能检查溢出或下溢，还能将结果钳制在有效范围以内。

无需流水线化

图 6 显示的是使用免费的赛灵思 WebPACK™ 工具^[8]中提供的 14.7 版 ISim 仿真器对该 VHDL 模型进行仿真并将馈送图 2 中的采样 / 量化 12 位数据序列后得到的上采样数据序列。每个原始的 12 位样本均保持不变，原因上文已述。原始波形中每个实际样本之间插入了三个新样本。

计算所得（上采样所得）值与原始模拟信号中理想值之间最大的误差为整个范围的 0.464%，平均误差为整个

布局布线设计使用19个DSP48E1模块，但占用的Virtex-6 Slice资源不足1%，无需流水线化即可运行在107MHz下。

范围的 0.070%。当然因初始量化步骤原因，在采样 / 量化 12 位源矢量数据值中存在 1/2 LSB 的误差（或整个范围的 0.012%）。

我们使用 WebPACK 工具 14.7 版在赛灵思 XC6VLX 75T-3FF484 Virtex-6 FPGA^[7] 上实现这一上采样器。布局布线设计占用该器件中 288 个 DSP48E1 模块中的 19 个，

但使用的 Slice 资源不足 1%。最终得到的上采样电路能够运行在 107MHz 下。无需让滤波器流水线化即可实现这一性能。此外，我们还开发出了用流水线实现的版本，可以工作在 217MHz 以上。

虽然 XC6VLX75T-3FF484 是赛灵思 Virtex-6 系列中的最低端成员，但芯片上仍集成有 288 个带有 25x18 位乘法

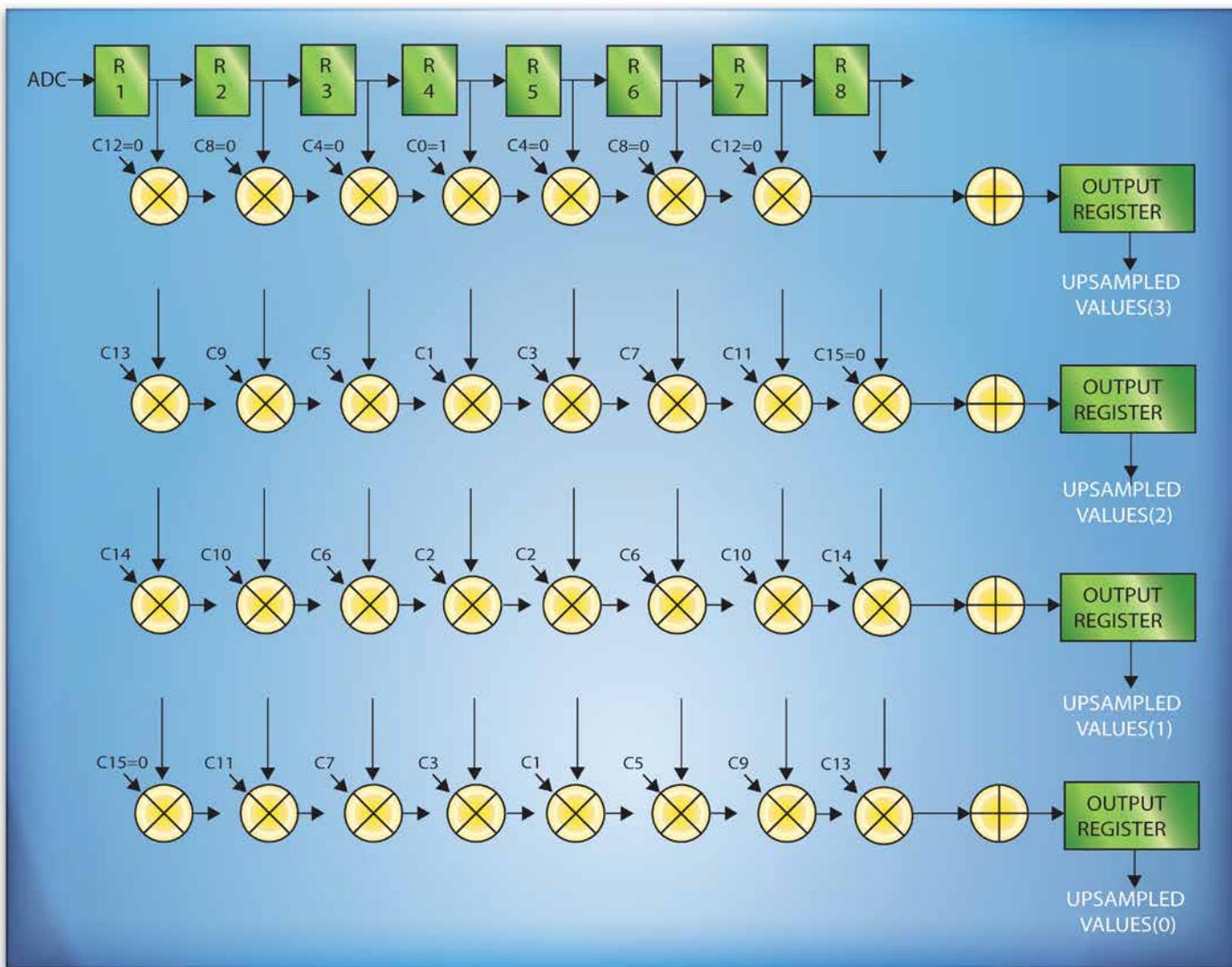


图4 - 通过在任何给定时钟周期内对图3中每四个寄存器中唯一一个包含非零数据的寄存器进行观察，可以拆解该滤波器，再让滤波器在以基础ADC时钟频率运行的情况下并行

```

LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
USE IEEE.STD_LOGIC_ARITH.ALL ;
USE IEEE.STD_LOGIC_UNSIGNED.ALL ;

ENTITY upsample IS
    PORT (clk          : IN STD_LOGIC ;
          r_ext       : IN STD_LOGIC_VECTOR(11 DOWNT0 0) ;
          d0,d1,d2,d3 : OUT STD_LOGIC_VECTOR(11 DOWNT0 0)) ;
END upsample ;

ARCHITECTURE mine OF upsample IS
    SIGNAL r1,r2,r3,r4,r5,r6,r7,r8          : STD_LOGIC_VECTOR(14 DOWNT0 0) ;
    SIGNAL d0int,dlint,d2int                : STD_LOGIC_VECTOR(39 DOWNT0 0) ;
    CONSTANT c1 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "1110001111110110011100110" ;
    CONSTANT c2 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "1001101111101110000000011" ;
    CONSTANT c3 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0100010101111101100111001" ;
    CONSTANT c5 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0010001010010010011110000" ;
    CONSTANT c6 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0010001110001110010111001" ;
    CONSTANT c7 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0001001000101111000010111" ;
    CONSTANT c9 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0000100011011001000000101" ;
    CONSTANT c10 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0000100000100110000100111" ;
    CONSTANT c11 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0000001101110111011000010" ;
    CONSTANT c13 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "000000011000100001100100" ;
    CONSTANT c14 : STD_LOGIC_VECTOR(24 DOWNT0 0) := "0000000001000001000111111" ;

BEGIN

    flops:PROCESS(clk)
    BEGIN
        IF (clk = '1' AND clk'EVENT) THEN
            r1 <= "000" & r_ext ;
            r2 <= r1 ;
            r3 <= r2 ;
            r4 <= r3 ;
            r5 <= r4 ;
            r6 <= r5 ;
            r7 <= r6 ;
            r8 <= r7 ;
            IF d0int(39) = '1' THEN
                d0 <= "000000000000" ;
            ELSIF d0int(38) = '1' OR d0int(37) = '1' THEN
                d0 <= "111111111111" ;
            ELSE
                d0 <= d0int(36 DOWNT0 25) ;
            END IF ;
            IF dlint(39) = '1' THEN
                d1 <= "000000000000" ;
            ELSIF dlint(38) = '1' OR dlint(37) = '1' THEN
                d1 <= "111111111111" ;
            ELSE
                d1 <= dlint(36 DOWNT0 25) ;
            END IF ;
            IF d2int(39) = '1' THEN
                d2 <= "000000000000" ;
            ELSIF d2int(38) = '1' OR d2int(37) = '1' THEN
                d2 <= "111111111111" ;
            ELSE
                d2 <= d2int(36 DOWNT0 25) ;
            END IF ;
            d3 <= r4(11 DOWNT0 0) ;
        END IF ;
    END PROCESS ;

    d0int <= r2*c11 - r3*c7 + r4*c3 + r5*c1 - r6*c5 + r7*c9 - r8*c13 ;
    dlint <= (r2+r7)*c10 - (r1+r8)*c14 - (r3+r6)*c6 + (r4+r5)*c2 ;
    d2int <= r2*c9 - r1*c13 - r3*c5 + r4*c1 + r5*c3 - r6*c7 + r7*c11 ;

END mine ;

```

图5 - 该VHDL源代码使用单进程和25位定点系数实现图4的滤波器拓扑结构。

这种简明的FIR滤波器设计方法无需借助复杂精密的滤波器设计工具就能提供优异的结果。

器的 DSP48E1 模块。换言之，在理论上足以实现 15 个图 4 所示的并行上采样 FIR 滤波器。我们制作的原型环形阵列超声系统使用了八套以 80MHz 的频率运行在 XC6VLX75T FPGA 上的上采样器，在波束成形之前对来自八通道 Analog Devices AD9670 超声前端芯片的数据进行上采样处理。在该系统中，上采样器按仿真预测的方式运行，在以 AD9670 ADC 的基本时钟频率 80MHz 运行的情况下，能使用上采样到 320MHz 的数据实时完成波束成形。

最大型赛灵思 Virtex-6 FPGA 器件 XC6VFX475T 包含有 2,016 个 25x18 位乘法器，在理论上一个芯片就可以实现 106 个图 4 所示类型的上采样滤波器。

只要滤波器使用本文介绍的高效并行拓扑结构进行设计，就能够使用实现在工作频率为 107MHz 的赛灵思 XC6VLX75T-3FF484 FPGA 上的 FIR 滤波器完成 $M=4$ 倍的实时上采样。原始数据样本将原封不动通过滤波器，并行产生 $(M-1)=3$ 个上采样值。这种简明的 FIR 滤波器设计方法无需借助复杂精密的滤波器设计工具就能提供优异的结果。本文介绍的思路稍加拓展，就可以使用更大的因

数进行上采样，或者是使用抽头数更多的 FIR 滤波器降低计算出的上采样值的误差。🌈

参考资料：

1. A.V. Oppenheim、R.W. Schaffer, 《离散时间信号处理》，Prentice Hall, 新泽西州恩格尔伍德克利夫斯, 1989 年。
2. H. Stark、J.W. Woods、I. Paul, 《使用直接傅里叶逆转换和最优插值法对计算机体层摄影术进行探讨》，IEEE 生物医学工程通讯第 28 期, 496 到 505 页 (1981 年)。
3. R.W. Schaffer、L.R. Rabiner, 《插值的数字信号处理方法》，IEEE 第 61 期会刊, 692 到 702 页 (1973 年)
4. R. Crochiere、L.R. Rabiner, 《多速率数字信号处理》，Prentice Hall, 新泽西州恩格尔伍德克利夫斯, 1983 年。
5. D. Pellerin、D. Taylor, 《轻松 VHDL 设计》，Prentice Hall, 新泽西州恩格尔伍德克利夫斯, 1997 年。
6. Analog Devices AD9670 八通道超声 AFE 及数字解调器产品说明书, Sp0 修订版, 美国模拟器件公司, 2013 年。
7. Virtex-6 系列总览 2.3 版 (DS150), 赛灵思公司, 2011 年。
8. ISE 深入教程 13.1 版 (UG695), 赛灵思公司, 2011 年。

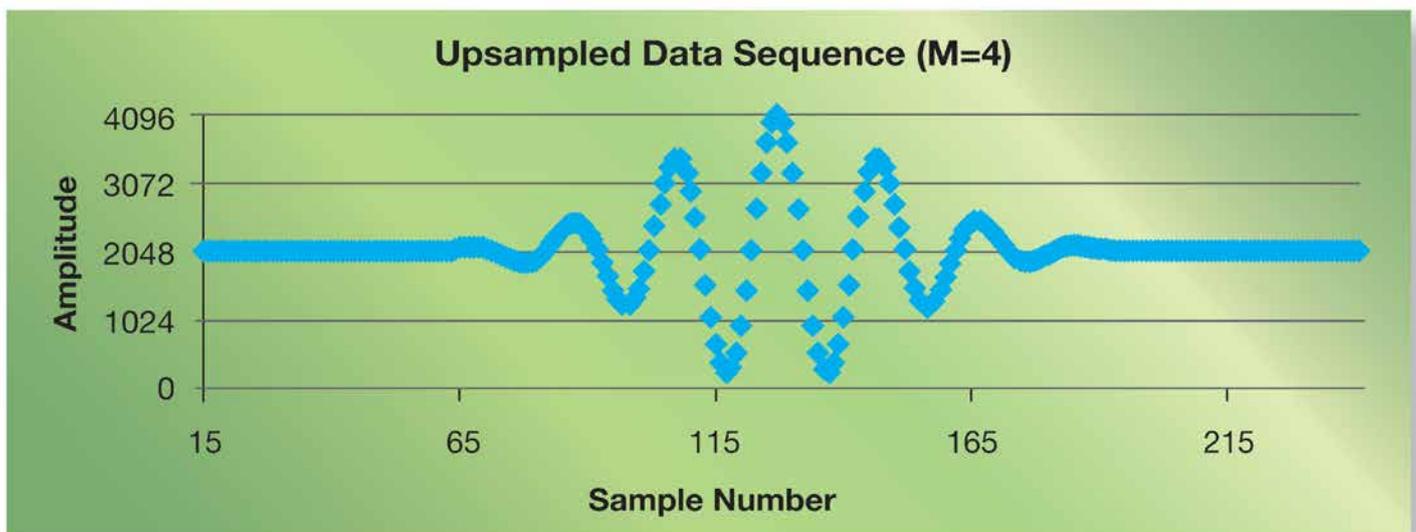


图6 - 该图表显示的是VHDL模型生成的上采样数据序列

依元素科技培训课程时间表 2015/1 至 2015/3

培训课程	培训时间	1月	2月	3月
Vivado设计套件工具流程	1天	14日 深圳 21日 上海	4日 武汉 11日 北京	4日 北京 11日 深圳
基于ISE软件工程师用户的Vivado设计套件	1天	15日 杭州	4日 成都	18日 北京
基于Vivado的FPGA设计基础	2天	12-13日 北京	2-3日 上海	19-20日 西安
Vivado设计套件的高级工具和技术	2天	19-20日 成都	12-13日 深圳	9-10日 北京
Vivado设计套件静态时序分析和Xilinx设计约束	2天	12-14日 深圳	12-13日 上海	12-13日 北京
面向软件设计人员的基于C语言HLS编码	1天	7日 西安		25日 北京
面向硬件设计人员的基于C语言HLS编码	1天	14日 武汉		18日 上海
基于C语言设计: Vivado HLS高层次综合	2天	29-30日 西安 19-20日 北京	12-13日 北京 5-6日 深圳	2-3日 上海 16-17日 上海
Zynq全面可编程SoC架构介绍	1天	21日 北京	4日 深圳	11日 上海
Zynq全面可编程SoC系统架构	2天	15-16日 深圳 19-20日 上海	5-6日 武汉 2-3日 西安	19-20日 北京 30-31日 成都
嵌入式系统开发	2天	12-13日 成都		26-27日 北京
嵌入式系统软件开发	2天	29-30 深圳		
嵌入式开放源码 Linux 开发	2天		11日 深圳	
高级嵌入式系统软件设计	2天	22-23日 上海		30-31日 深圳
基于ISE的FPGA设计基础	1天			25日 深圳
利用Vivado逻辑分析器的调试和验证	2天	26-27日 深圳	2-3日 北京	12-13日 上海
面向性能的设计	2天	8-9日 上海		
高级FPGA设计	2天	15-16日 武汉		
使用PlanAhead分析与设计工具进行基本设计	1天	28日 北京	11日 上海	11日 西安
使用PlanAhead分析与设计工具进行高级设计	2天	26-27日 武汉		23-24日 杭州
Xilinx部分重配置工具和技术	2天	22-23 北京		30-31日 上海
利用Spartan-6和Virtex-6系列进行设计	2天	8-9日 深圳		
使用7系列产品进行设计	2天	23-24日 深圳 27-28日 北京	2-3日 上海 12-13日 杭州	2-3日 北京 9-10日 上海
Xilinx FPGA的信号完整性和电路板设计	2天			26-27日 上海
设计LogiCORE PCI Express系统	2天	5-6日 上海		16-17日 北京
利用以太网MAC控制器进行设计	2天			26-27日 北京
利用千兆位串行I/O进行设计	2天	8-9日 北京		23-24日 深圳
利用System Generator进行DSP设计	2天	29-30日 北京		19-20日 天津
Xilinx FPGA的基本DSP实现技术	2天		9-10日 深圳	
利用VHDL进行设计	2天		9-10日 北京	
利用Verilog进行设计	2天	7日 深圳		26-27日 上海
Vivado设计套件的基本Tcl脚本	1天	21日 武汉		18日 深圳
UltraFast设计方法	1天	14日 北京	4日 上海	4日 西安
UltraScale架构设计	2天	29-30日 上海	5-6日 深圳	2-3日 成都

Xilinx在线培训课程系列 (WebEx)	培训课程	1月	2月	3月
以在线培训方式实施现场课堂教学和实验, 学员于线上学习。面向全国或是海外华人工程师参加的中文FPGA培训课程, 适合交通不便或工作繁忙不便参加现场培训的工程师。课程安排Q&A时间, 老师现场解答学员在学习和实验中遇到的问题, 提供最新的实验项目现场操作并进行专业辅导, 直接带给学员FPGA项目设计的亲身体验。授课老师都获Xilinx认证, 并具有丰富的FPGA系统项目经验。	FPGA设计基础 (免费)	7日	4日	4日
	面向性能的设计	12-13日		2-3日
	高级FPGA设计	15-16日		9-10日
	PlanAhead分析与设计	19-20日	2-3日	16-17日
	利用7系列产品进行设计	22-23日	5-6日	19-20日
	Zynq所有可编程SoC系统架构	26-27日	9-10日	23-24日
	Vivado设计套件工具流程 (免费)	28日	11日	25日
依元素科技高级FPGA培训课程系列	培训课程	1月	2月	3月
以FPGA应用方向与案例式教学为主的FPGA实战课程, 使用Xilinx最新的FPGA培训课程为基础, 理论部份针对工程师在设计上最常见需求和问题来安排培训内容, 实践部分结合实际项目案例培养动手能力并解决实际问题的调适能力, 是有一定FPGA设计基础工程师, 想提升设计能力的最好选择课程, 授课老师都获Xilinx认证, 并具有丰富的FPGA系统项目经验。	基于FPGA的网络设备开发实战			2-3日 北京
	基于Xilinx FPGA的DSP系统设计	8-9日 北京		23-24日 北京
	设计高速串行传输电路	26-27日 北京		26-27日 北京
	通用FPGA接口电路设计技术			9-10日 北京
	基于Xilinx FPGA的高速存储接口设计	15-16日 北京		16-17 北京
Xilinx FPGA设计高级进修班		13-14日 北京		
深入浅出FPGA在高速光网络中的开发应用				30-31日 北京

有关报名注意事项:

请联系: 北京: 电话: 010-8275-7632, 传真: 010-62166151
 深圳: 电话: 0755-86186715, 传真: 0086-755-86186700,
 电子邮件: training@e-elements.com
 地址: 北京市海淀区北三环西路32号恒润国际大厦801
 网址: www.e-elements.com

Xilinx宣布推出针对OpenCL、C和C++的SDAccel开发环境,

完美结合业界首款架构优化编译器、库、开发板, 在FPGA上首次实现完全类似于CPU/GPU的开发和运行时间体验

2014年11月18日, 中国北京--All Programmable技术和器件的全球领先企业赛灵思公司(Xilinx, Inc. (NASDAQ:XLNX))今天在2014国际超算大会(Super Computing 2014)上宣布推出针对OpenCL™、C和C++的SDAccel™开发环境, 将单位功耗性能提高高达25倍, 从而利用FPGA实现数据中心应用加速。SDAccel是赛灵思SDx™系列的最新成员, 将业界首款支持OpenCL、C和C++内核任意组合的架构优化编译器、库、开发板完美结合在一起, 在FPGA上首次实现了完全类似CPU/GPU的开发和运行时间体验。

戴尔公司平台架构与技术及CTO办公室执行总监Robert Hormuth指出:“基于FPGA的技术有了新的突破, 能支持优化的计算应用。在戴尔服务器部署的过程中, 简化编程是决定采用FPGA加速器的关键障碍。毫无疑问, 赛灵思开辟了一条正确的道路, 让开发人员能够借助一个软件环境, 提高FPGA平台用户的生产力。”
 IBM电源开发副总裁兼OpenPOWER基金会总裁Brad McCredie表示:“IBM高度赞赏赛灵思致力于实现其FPGA软件可编程性这一发展方向。利用C、C++和OpenCL创建优化型FPGA加速器的高度灵活性和可靠的结果质量, 能提升IBM为客户带来更大价值的价值。IBM坚信OpenCL对提高生产力大有裨益, 同时我们正在与赛灵思展开紧密合作, 将该技术应用到我们的OpenPOWER产品设计中。”

- 首款针对OpenCL、C和C++的架构优化编译器
- 在FPGA上首次实现完全类似CPU/GPU的开发体验
- 在FPGA上首次实现完全与CPU/GPU的运行时间体验



欢迎各位作出反馈讯息和建议
 传真: (010) 5939 3005
 电邮: webmarketing_apac@xilinx.com

赛灵思 中国/香港代表处

香港 电话: (852)2424 5200
 上海 电话: (86)21-5131 6060
 深圳 电话: (86)755-8660 6588
 北京 电话: (86)010-5651 7300

传真: (852)2494 7159
 传真: (86)21-5198 1020
 传真: (86)755-2583 0986
 传真: (86)10-5939 3005

电邮: ask-china@xilinx.com

FPGA设计规划框架

作者: Jeffrey Lin

赛灵思公司全球通信服务部高级经理

Jeffrey.lin@xilinx.com

这款久经考验的
FPGA开发框架是您
通向完美项目执行的
通途。

框架简介

该框架是一种在 FPGA 中设计硬件的自上而下的迭代设计方法。首先规划从系统架构层面出发决策 FPGA 功能。随后我们根据 FPGA 器件的已知功能和性能逐步优化 FPGA 设计的特性。

另外，大型 FPGA 设计的实现需要制定完备的开发、仿真和验证规划。该框架的作用就是帮助我们制定这些规划。简而言之，该框架可归纳为图 1 所示的流程图。本文的讨论将集中在规划和技术文档部分（顶部）。

系统架构

就本讨论范围而言，系统架构指在系统软件和硬件之间的功能划分。尤其，重点是将硬件功能细分到 FPGA 以及其它微芯片组件上（假设已经明确产品层面需求），例如市场营销或产品定义部门可能已经参与进来并明确产品需求。

在系统架构阶段，思路是如何明确地让这些产品需求在真实产品中得以实现。就 FPGA 来说，主要的决策围绕着应将哪些特性和功能在 FPGA 中实现，以及进一步，哪些特性和功能适合在 FPGA 中实现。

通过提前定义 FPGA 的高级要求，就能避免开发流程已接近尾声时进行成本不菲的设计及需求修改。在设计早期阶段明确系统架构有助于指导用户做出对开发时间和产品成本至关重要的多项重大决策。

在这个层面上探讨，只需要一般性地概略介绍 FPGA 特性。详细特性和实现需求将在 FPGA 需求定义过程

的后续阶段进行定义。本次探讨的参与者应包括熟悉系统级要求的、了解系统级架构设计的，以及熟悉 FPGA 特性与功能的等不同人员。

具体就 FPGA 而言，需要回答十大问题：

1. 要在 FPGA 中实现的特性列表是什么样的？
2. 在 FPGA 中实现特性与使用非 FPGA 组件实现特性相比，需在技术上做怎样的权衡取舍？
3. 在 FPGA 中实现与使用非 FPGA 组件实现相比，设计工作量 / 成本分别如何？
4. 需要什么样的定制特性或处理？
5. FPGA 的灵活性对功能有什么好处？
6. 您应该考虑的未来风险缓解措施是什么？
7. 能否把多个非 FPGA 组件的特性集中到 FPGA 中？
8. 根据准备实现的设计特性，如何选择 FPGA 器件？
9. 特性是否能在 FPGA 中实现？
10. 需要什么样的非 FPGA 器件，如何让这些非 FPGA 器件与 FPGA 接口相连？

FPGA 架构

FPGA 架构属于 FPGA 器件上物理层的微架构级和芯片级数据流设计。您的团队应与系统级架构同步设计该架构，以确定器件尺寸、选择器件和可行性。

定义 FPGA 架构的目的，是确保

长久以来新型 FPGA 的功能和性能已经为它们赢得系统中的核心位置，成为许多产品的主要数据处理引擎。

鉴于 FPGA 在如此多应用中的重要地位，采取正式且注重方法的开发流程来处理 FPGA 设计比以往更加重要。该流程旨在避免开发周期后期因发现设计缺陷而不得不进行费时费钱的设计修改，而且该缺陷还可能对项目进度计划、成本和质量造成灾难性影响。

赛灵思全球通信服务部一直以来都采用久经考验的设计框架来为其客户开发和交付一揽子 FPGA 设计，产品涵盖医疗图像处理工程到自学网络交换工程，应有尽有。该框架在我们设计、开发以及交付成百上千 FPGA 设计的过程中得到不断发展和完善。

我们使用的框架覆盖从系统架构考量到 FPGA 开发与测试规划等各个环节。我们从 FPGA 硬件的角度重点详细介绍该框架，通过介绍希望其他工程团队能够发现该框架在复杂的 FPGA 设计项目里面的优点。

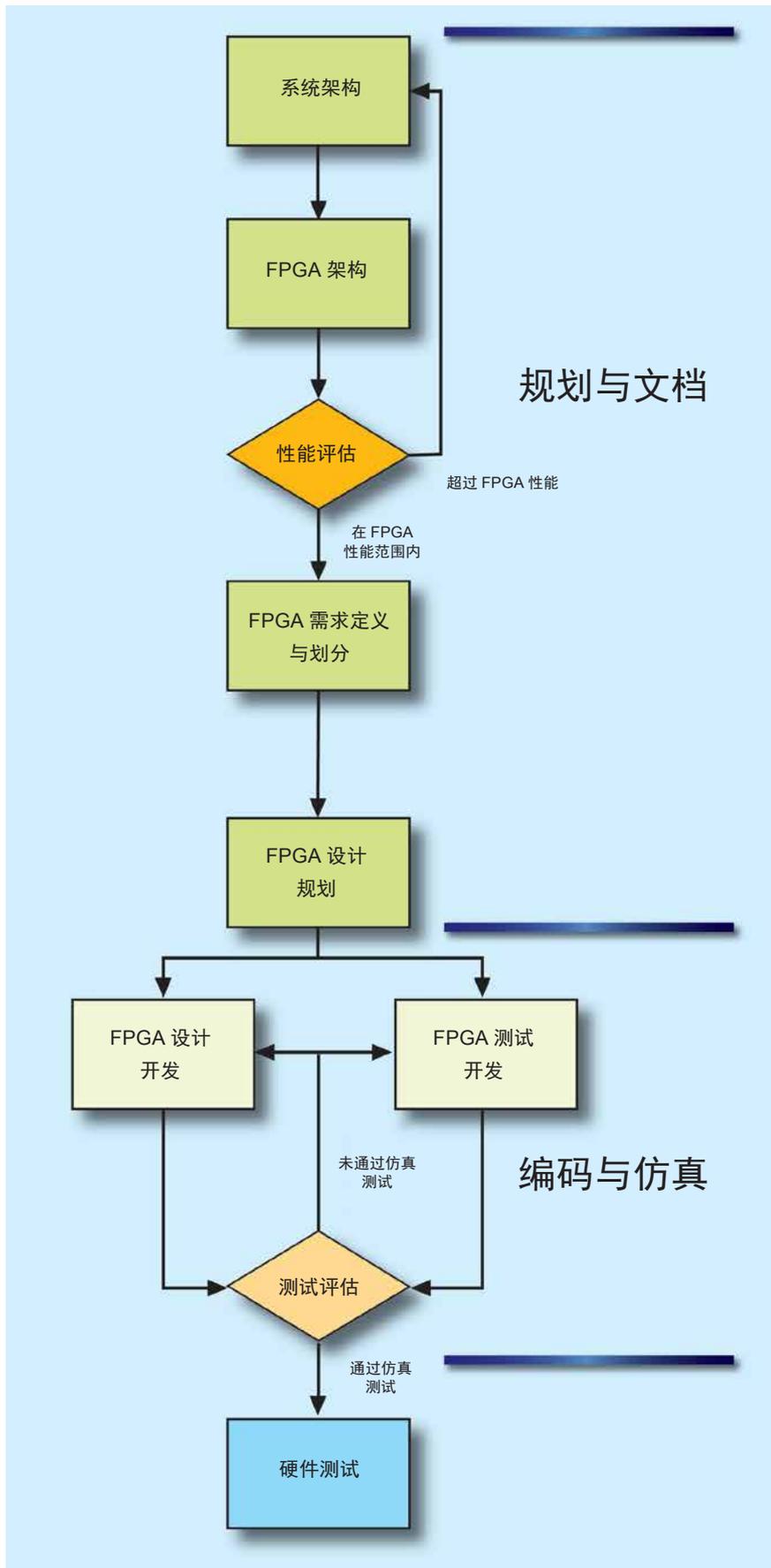


图1 - FPGA开发框架

系统架构需求是准确、现实、切实可行的设计要求，能够在FPGA中实现。

这个层面的探讨需要对FPGA架构和资源的特性与功能有深入的认识。因此应该由经验丰富的FPGA设计人员参与完成。在这个阶段，您必须考虑与FPGA性能目标、潜在风险因素和FPGA资源利用率。

在FPGA架构定义阶段，您有可能会发现系统级需求和架构无法实现，或是对实现在FPGA中而言存在高风险。在这种情况下您必须重新评估和更新系统架构，以创建在FPGA中可实现的高级需求列表。

您应问问自己现有IP哪些可供使用，还必须创建哪些IP。此外，您还需要检查I/O需求，以及把时钟域和时钟特性映射到FPGA时钟资源的方法。其他关键问题包括：如何在FPGA上布局千兆位收发器(GT)资源；在SSI器件中是否已考虑到交叉SLR数据流；目标时钟频率对设计功能而言是否实际。最后，您还必须评估您设计的目标性能对选择的FPGA而言是否实际。

FPGA需求定义与划分

FPGA需求定义和划分阶段与系统和FPGA架构定义紧密关联并受这两个阶段的决策左右。FPGA需求定义指定义准备在FPGA中实现的详细需求，并用作明确的特性清单以供设计和测试工程团队设计、测试和实现。FPGA需求定义与系统和FPGA架构需求定义的不同之处在于FPGA需求是精确的。该清单定义了FPGA的细致要求，而不仅仅是系统的不同组件之间的功能划分，或是穿越FPGA的数据流之间的功能划分。

这个阶段的目的是清楚地定义 FPGA 工程团队确切能实现和测试的内容。在这个阶段，用户将把高级系统和 FPGA 架构需求转换成用于实现的具体需求。这样做能够带来双重好处。首先，单独定义 FPGA 需求能突出强调系统和 FPGA 架构的任何局限以及之前未曾考虑到或未曾预见到的状况。其次，该步骤将为 FPGA 设计的开发和测试的顺利执行铺平道路。

为适当地描述 FPGA 需求，您必须建立清晰且简明扼要的定义，以便提炼成单独的需求。我们建议为各项需求添加标签或序号，使用易于判断其能否实现的简短描述来定义各项需求，避免用高级含糊的需求用语。只要足够清晰扼要，您可以使用任何行业标准格式或专有格式。

避免使用含混或界定不清的用语，例如“快速”或“小型”。应坚持使用有明确目标的用语，如“400MHz”或“4.2K 触发器”。这样定义的目的旨在确保将文档分配给之前不了解系统或 FPGA 架构的开发工程团队进行实施时不再需要反复澄清。您需要检查每个需求是否表述清楚、简洁、明确且是否包含了所有必需的信息以避免反复澄清的必要。此外，还应该注意需求中是否包括管脚和 I/O 定义？是否所有的高级需求都已经分解为基本设计要素？未参与早期系统架构定义的设计团队是否能够使用这些需求开发 FPGA？以及最后测试与验证团队是否能使用该文档开发测试平台和制定测试方案，明确验证每一项需求是否通过。

FPGA 设计规划

该框架的这个阶段用于为 FPGA 硬件

的实际开发制定规划，确保各项特性及开发工作的完成与整个产品开发的其他部分协调一致。

这个阶段的目标是适当地把当前的系统级需求、FPGA 级需求和架构体现到开发规划中。在通过前面介绍的规划阶段之后，现在开发团队一般会面临两种情况。

第一种情况是系统与 FPGA 架构以及需求理解良好，描述详尽，最终，FPGA 设计开发阶段（即 HDL 编码）和测试开发阶段（仿真、测试台）基本无需修改设计需求，顺利执行。

第二种情况是系统架构和 FPGA 需求仍然处在变动中。这样的设计会在设计开发周期和测试开发阶段面临众多变数和修改。

虽然人人都想得到第一种情况的结果，但往往却不能成功，最终落入第二种情况的境地。很明显，这种情况更加难于管理。

设计规划的总体目标应是在开发周期的这个阶段实现第一种情况。在第一种情况中，FPGA 的开发简单直观，只需为实现和测试设计特性安排进度计划。

在第二种情况中，最重要的管理工作是确保落实充分理解的流程，以便评估和决定应该进行哪些修改以及每种修改为总体开发进度计划带来的影响如何。这里可以运用多种项目管理理念和技巧。最重要的一点是完成上述修改评估和影响评估。

就专门的 FPGA 规划和开发而言，FPGA 的优势之一在于能够多次修订硬件平台和多次把硬件平台下载到原型 PCB 上。设计团队应充分发挥这一功能的优势。因此建议的开发规划是逐渐把特性添加到能够工作的设

计中。其思路是从能够实现主要通信接口的基本设计出发，无需实现全部需求就能工作。

这样做能够带来双重好处。首先可以确保随时有一个可工作的设计，您可用来调试 PCB 和更大型系统。其次，调试实际的 FPGA 设计会简单得多，因为可以检查新添加的特性，确保新添加的特性不会干扰或中断当前可工作的设计。

与 FPGA 设计开发同步，同样重要的是为实现的 FPGA 设计准备良好的仿真环境规划。投资开发稳健可靠的仿真环境，就可以复制现实数据流，在仿真中重现出错条件，迅速隔离和判断根本原因，从而不仅能减少设计缺陷，还能显著缩短实验室调试时间。

开发稳健可靠的测试仿真环境和开发 FPGA 本身一样复杂，需要视同开发 FPGA 一样规划和考量。

赛灵思全球通信服务部已经把我们的 FPGA 开发框架一致应用于成百上千个 FPGA 设计中并不断优化完善。结果得到一种切实可行，成效显著，易于理解，能广泛地适用于各种不同开发工作的设计方法。您使用该框架开发新型 FPGA 设计能获得诸多好处，其中包括准确的总体开发进度计划、快速硬件启动，而且最终能够按时发布产品。🌈

用Vivado IPI和Xilinx IP 实现更快速的设计输入

作者：Duncan Cockburn
赛灵思公司高级设计工程师
duncan.cockburn@xilinx.com

本文将介绍如何
优化赛灵思内核
以便在CPRI远
程无线电头端设
计中使用Vivado
IPI。

新

型基于 FPGA 的设计使用 IP 核的数量和种类日趋繁多。Vivado® 设计套件中的 IP 集成器 (IPI) 工具和赛灵思通信 IP 让设计人员能够更加轻松快速地将 IP 模块连接在一起。

为了更好地阐明 IPI 方法的强大之处，我们以远程无线电头端 (RRH) 为例。RRH 位于天线附近，构成蜂窝通信网络的一部分。它们通常通过光纤连接至上游的基带收发器站，并可以选择性地连接到下游的更多 RRH，从而构成多跳拓扑结构（图 1）。

通用公共无线电接口 (CPRI) 协议普遍用来连接这些 RRH。现在让我们来构建包含一个上行链路 CPRI 端口和三个下行链路 CPRI 端口且这些端口连接在一起的实例设计。我们可以利用 IPI 构建出整个设计中的主

体部分。然后我们选用 Kintex®-7 器件，因为该器件具备低功耗、低成本和高性能等优异特性，因此非常适合此实例设计。速度等级为 -2 的 All Programmable 器件 Kintex FPGA 和 Zynq®-7000 SoC 中的 GTX 收发器支持使用 9.8Gbps CPRI 线路速率。

图 2 展示了我们在 IPI 中将要创建的内容。我们可建立模块设计，并从 IP Catalog 中实例化所需的 IP。标准赛灵思 IP Catalog 中提供有 CPRI 内核。该内核经过优化，不仅可实现资源共享（可能情况下），而且还能简化在 IPI 中的使用。交换机属于自定义 IP。

IP 核资源共享

使用多个 IP 实例时客户遇到的挑战之一是如何高效共享资源。很多通信 IP 核都支持“共享逻辑”功能。对于 CPRI 内核，我们可用内核中的可共享逻辑资源配置 IP，或者省略这些共享资源。如果共享资源包含在内核之中，它们会提供必要的输出以便让我们将共享资源连接到不含这些逻辑的内核。

有特殊要求的用户可能希望在内核中移除这些共享逻辑并实现自

己的逻辑。在我们的实例设计中，我们将 CPRI 内核的运行速度配置为 9.8Gbps。在该线路速率下，有必要使用 LCtank 振荡器来提供收发器时钟。Kintex-7 器件中的收发器采用 Quad 布局，每个收发器 Quad 包含四个收发器通道和一个基于 LC-tank 的 Quad 锁相环 (QPLL)。所有内核都有必要共享 QPLL 以及由上行链路时钟信号生成的时钟。图 3 中给出了带有定制共享逻辑的上行链路内核上 QPLL 和时钟输出端口，该上行链路内核的共享逻辑连接到将不含共享逻辑的下行链路 CPRI 内核的相应输入端口。

CPRI 内核间的数据路由

我们还进行了 IQ 交换机和以太网交换机的实例化，以便让数据在内核之间实现路由。

CPRI 网络中的控制和管理数据通过以太网子通道发送。凭借系统中的以太网交换机，可以远程发布固件升级或命令，并将其发送至任意节点。IP 旨在使用尽量少的逻辑资源，因为这种情况没必要使用功能齐全的以太网交换机。

IQ 交换机能以确定的时延在 CPRI 内核之间路由任何 IQ 样本。多跳无线电系统的一个重要特性是可以准确测量链路延迟，而且 CPRI 标准定义了一种能够协助该测量的方法。

用 IPI 连接接口

IPI 总线接口将定义好的一组逻辑端口映射到 IP 上的特定物理端口。如果我们尽可能地使用接口，就可以从连接很多信号转变为连接少量接口。IP 上的通用总线接口是符合 ARM® AXI 标准（例如 AXI4-Lite 和 AXI4-Stream）的接口。这种抽象程度的提高使设计

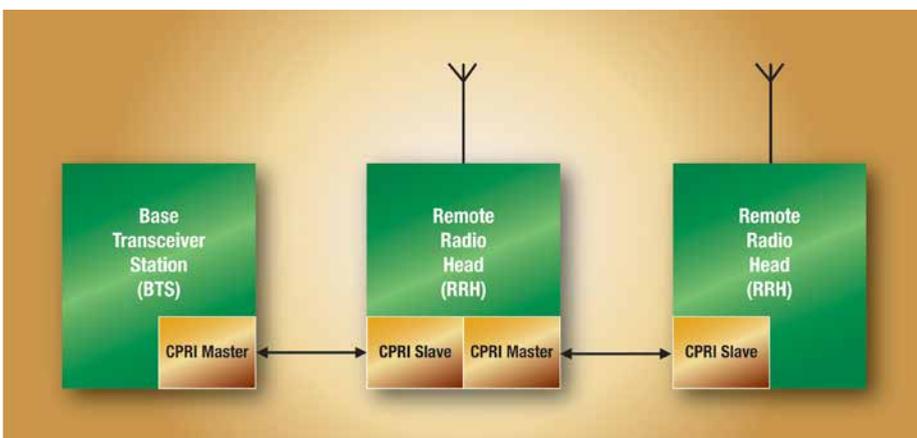


图1 - 多跳结构的方框图

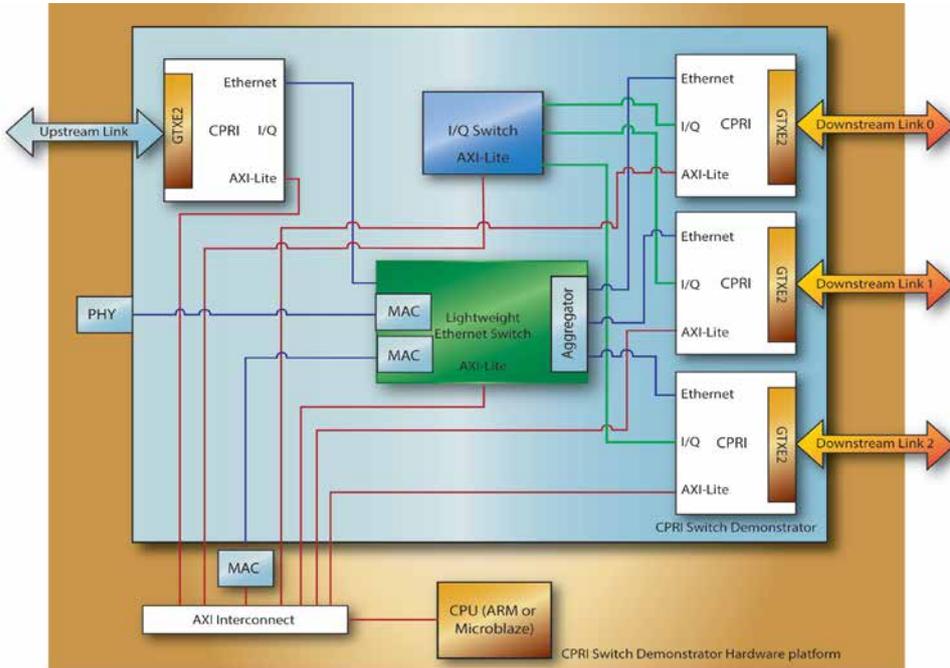


图2 – CPRI交换机硬件平台

输入更简单、更快速，而且使您能够充分利用接口的设计规则检查功能。

Vivado IP Packager 允许您在 IPI 中使用自己的 IP，并在自己的设计中利用这些接口。

IPI 便于将接口连接在一起。只需点击接口，IPI 就会指示其可以连接至具体内容。将连接线拖拽到所需的端点，就会建立起来连接。利用该方法，只需点击几次就可连接很多信号。

图 4 显示的以太网交换机提供若干 AXI4-Stream 接口，两个 GMII 接口，以及一个 AXI4-Lite 接口。数据流接口能直接连接到 CPRI 内核，这样无需在 CPRI 内核上进行内部缓冲。GMII 接口能连接到以太网 PHY，可供工程师在现场调试网络问题。AXI4-Lite 管理接口用来访问地址表格映射及其它配置选项，例如地

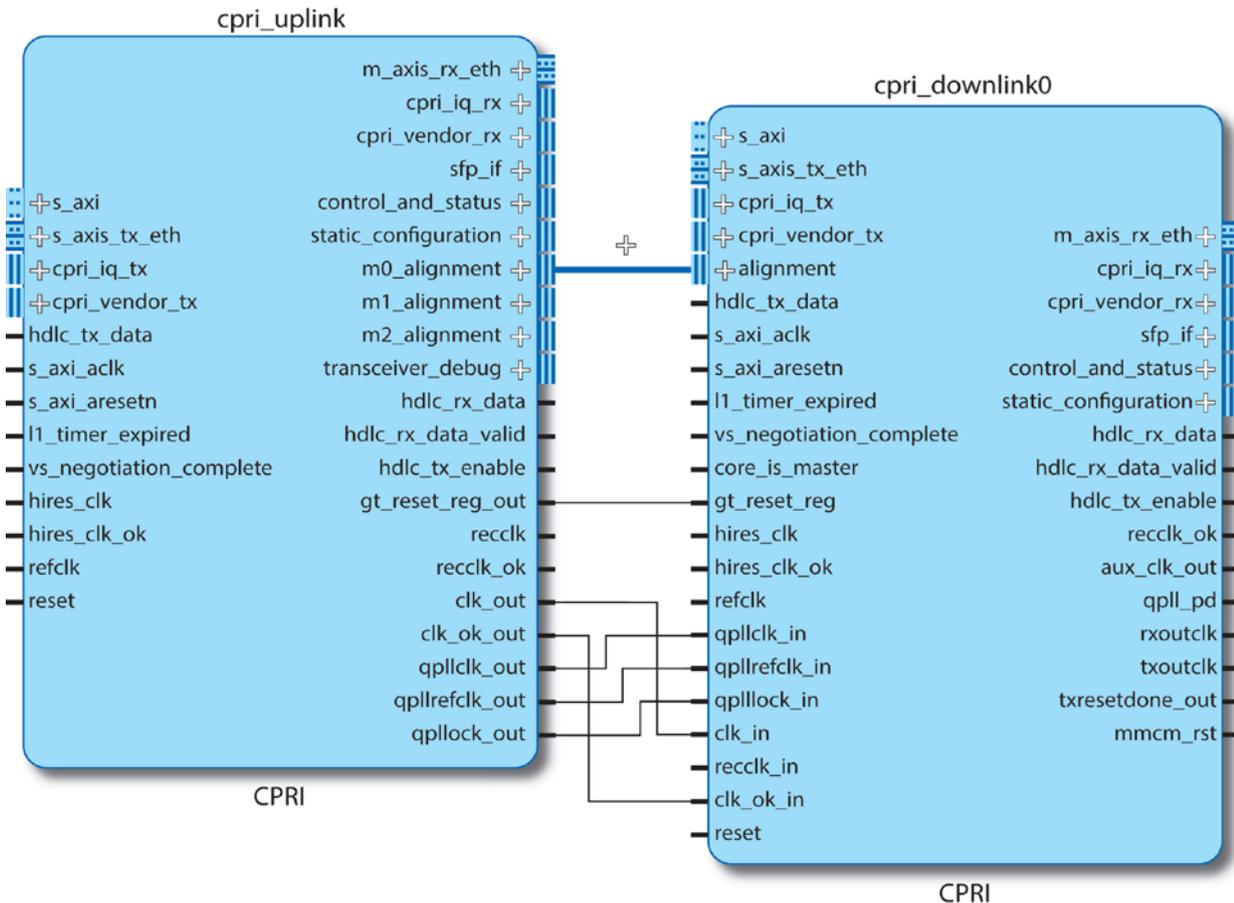


图3 – QPLL的共享逻辑连接

址表格老化间隔。

我们继续以这种方式增强系统，在 IPI 中连接各个接口。您可以灵活地使用任何最适合的输入方法。除了使用 GUI 连接接口外，还可以通过 Tcl 控制台直接发命令，或者从脚本获得命令。每次在 GUI 中做任何操作时，就会返回相应的命令。

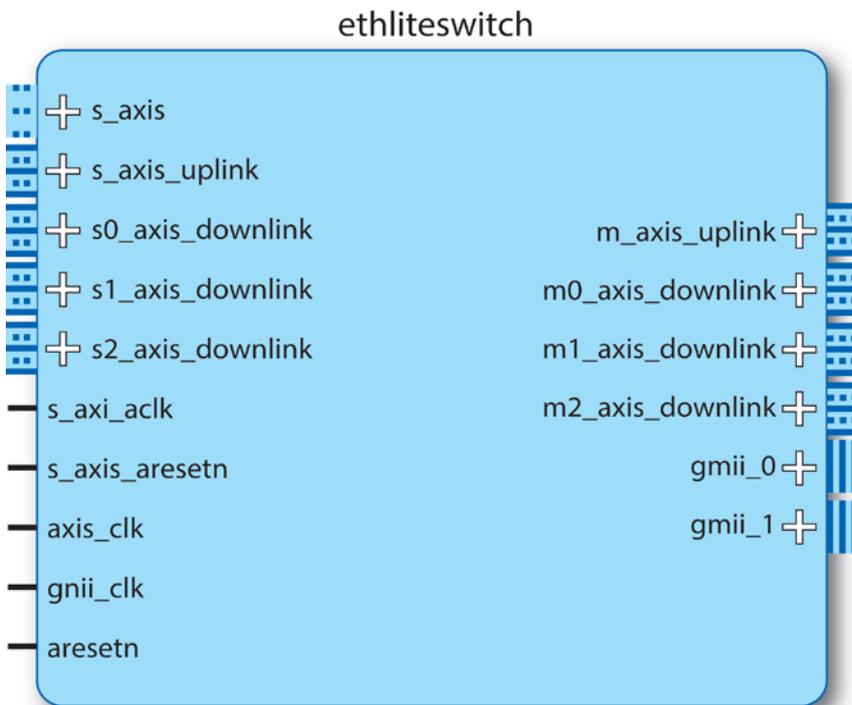
整个设计完成后，使用“write_bd_tcl”命令将整个设计导出。该命令创建一个可引用的 Tcl 文件，用以从头创建整个模块设计，或方便地作为脚本构建流程中的一部分。设计中的所有 IP 都提供一个 AXI4-Lite 管理接口，以便让内核连接到主机处理器。IPI 中内置的智能功能可以实现自动化连接。凭借这种机制，IPI 就能识别出 IP 上的 AXI4-Lite 接口将连接到 AXI 总线互联，并自动配置合适

的地址范围，为我们连接到总线。然后，您可借助 IPI 将该总线连接到主机处理器。本案例中的主机处理器是 MicroBlaze™，不过若使用 Zynq SoC 系列器件，就能方便地将主机处理器改为 ARM CPU。

更多优势接踵而至

Vivado IPI 功能正在快速增加，以后还将带来更多优势。通过使用正确的 IP，我们可以将整个子系统快速集成在一起，并从中受益。

如需了解有关 CPRI、以太网交换机或 IQ 交换机 IP 方面的更多信息，请发送邮件至：permind@xilinx.com，与赛灵思无线通信部的 Perminder Tumber 取得联系。



CPRI 6 Port Lightweight Ethernet Switch

图4 – 带接口的以太网交换机符号

Xilinx发布Vivado 2014.3、SDK及最新 UltraFast提升Zynq SoC 生产力

2014年10月9日，中国北京 - All Programmable 技术和器件的全球领先企业赛灵思公司 (Xilinx, Inc. (NASDAQ:XLNX)) 今天宣布推出可编程行业唯一 SoC 增强型设计套件 Vivado® 设计套件的 2014.3 版本、SDK 和最新 UltraFast™ 嵌入式设计方法指南，为 Zynq®-7000 All Programmable SoC 的生产力带来重大突破。伴随此款最新版 Vivado 设计套件推出的还包括其内含的 Vivado 高层次综合 (HLS) 和 IP 集成器的增强功能，以及最新性能监控与可视化功能。实践证明，这些加强功能与最新 UltraFast 嵌入式设计方法指南相结合，可将生产力提升 10 倍以上。

加速实现和验证：Vivado HLS 增强了从 C 语言综合的质量结果 (QoR, Quality-of-Results) 和 AMBA AXI-4 接口的自动推理功能，从而加速了集成的时间并提升了集成质量。利用 Vivado HLS，可以直接根据 C 算法规格描述创建和验证 IP，不仅可以快速实现可与手动编码 RTL 媲美的设计，而且验证速度比 RTL 仿真快好几个数量级。

加速集成：Vivado IPI 的增强功能包括：新增数据流和存储器映射 AXI 互联之间的自动连接功能，可促进并简化 IP 在 Zynq SoC 系统中的集成。赛灵思 Vivado 设计套件 2014.3 新增了 Xylon™ logicBRICKS™ 评估 IP 核，而在未来版本中将扩大和加入其他联盟计划成员的 IP。全新的 logicBRICKS IP 可快速评估有效的图像和视频处理 IP，并可进一步丰富 Vivado IP 目录。

加速系统设计和软件开发：赛灵思还扩展了其软件开发套件 (SDK) 功能，新增系统仪表与性能可视化功能，以便快速发现系统性能瓶颈，并运行假设情景流程。赛灵思 SDK 2014.3 版提供可在 FPGA 架构上运行的可配置 AXI 流量生成器，让设计人员在开发周期的早期阶段就可以提早进行嵌入式软件开发。

UltraFast 嵌入式设计方法指南：为了进一步补充和完善 Vivado 的 UltraFast™ 设计方法，赛灵思还推出了最新 UltraFast 嵌入式设计方法指南 (UG1046)。该最新指南为包括系统架构师、软件工程师和硬件工程师等在内的设计团队提供了利用 Zynq SoC 进行嵌入式系统设计的最佳实践，从而借助 Zynq SoC 实现可预见的成功并提升其嵌入式系统的生产力。

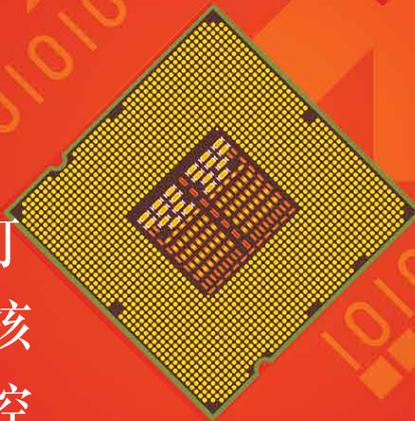
充分发挥PicoBlaze微控制器的优势

作者：Adam P. Taylor

e2v公司工程系统负责人

aptaylor@theiet.org

众多FPGA应用均可通过使用简单的软核处理器来简化时序控制结构的生成，从而深受其益。



PicoBlaze™ 是一款结构紧凑的 8 位软核微控制器，FPGA 工程师可在其所选的赛灵思 FPGA 中对其进行实例化。一旦实现，该处理器完全包含到仅使用逻辑 Slice 和 Block RAM 的 FPGA 架构中；无需外部易失性和非易失性存储器。

由于其实现方案封装面积小，FPGA 才可能容纳多个 PicoBlaze 实例化。每个实例化用于实现通常由状态机创建的控制结构。这样不仅能缩短开发时间，而且还可用标准化方法来控制结构生成。由于赛灵思 FPGA 结构潜在的高性能，PicoBlaze 实例化往往能够超越许多分离式 8 位微控制器。

下面让我们看看如何在我们的设计中充分利用这款便捷的器件。

PICOBLAZE 架构

在使用这个处理器之前，首先让我们了解一下它的架构。PicoBlaze 是一款基于 RISC 架构的非常简单的 8 位微

控制器（如图 1 所示）。该控制器有一个 12 位的地址端口，这意味着它可以处理多达 4096 个存储器位置。每个地址位置包括一个 18 位指令，用以定义该处理器必须执行的操作。该处理器的输入输出可能通过两个 8 位端口（一个输入端口，一个输出端口）。该控制器还提供了一个 8 位识别端口，可让多达 256 个外设通过该端口进行读写。此外，该控制器还提供一个 64、128 或 256 字节大小可选的高速暂存存储器。跟所有微处理器一样，PicoBlaze 包含一个算术逻辑单元并支持一个中断。有了这些功能，意味着该控制器能够为 FPGA 设计工程师带来众多优势。

PicoBlaze 最重要的一个方面是它具有高度的确定性，这意味着所有指令都需要两个时钟周期才能执行完毕，中断服务最多需要四个时钟周期。（您可以通过下载赛灵思用户指南，了解有关 PicoBlaze 架构的更多详情。）

如何使用 PICOBLAZE?

FPGA 应用通常需要结合并行和时序操作，其中数据流主要采用并行操作，而控制结构主要以时序结构实现，如状态机。（参见《赛灵思中国通讯》第 46 期，“[如何在您的 FPGA 中实现状态机](#)”）。然而，复杂的控制结构如果以状态机实现，可能会变得难以处理，不仅会延长验证时间，而且还会让开发周期的后期修改变得困难重重。复杂的状态机还需要花更多时间来开发，如果需要开发数个，那么花费时间就会相当长。

此外，您还可以使用 PicoBlaze 控制通过 RS232、I²C 和 SPI 接口的串行通信。事实上，您采用典型 8 位微处理器实现的任何方案均可在 PicoBlaze 中实现，而且具有更高的性能。工程师们已经用 PicoBlaze 在控制系统中实现 PID 控制器。他们结合使用 PicoBlaze 与 I²C、SPI 或并行 DAC 来创建方形、锯齿和三角形以及更复杂的正弦/余弦等各种参考波形（使用移位加 CORDIC 算法）。在您

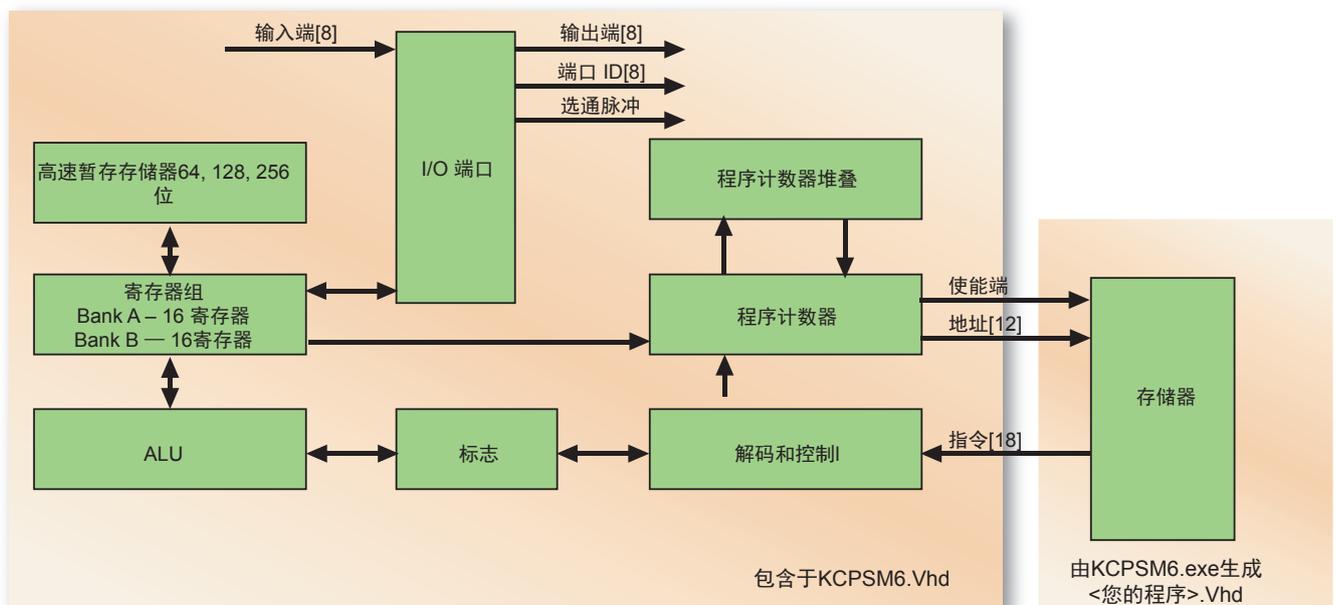


图1 - PicoBlaze架构，处理器（左）和存储器（右）

的FPGA中实例化PicoBlaze微控制器来实现这些时序功能，不仅可缩短开发时间，而且还允许在开发周期的后期阶段进行简单的修改。当然，作为软核，PicoBlaze还有助于解决过时的时间问题，而且在您开发ASM模块时还支持设计重用。

首个PICOBLAZE实例化

遵循以下几个简单步骤，您就可以在您的设计中快速实现PicoBlaze。首先，确保您将用于目标器件中的微控制器是最新版本。您可从赛灵思[PicoBlaze 下载区](#)下载这些微控制器产品，这里既提供支持最新7系列器件的版本，又提供能与旧版Spartan®-3和Virtex®-4器件协同使用的微控制器版本。

当下载了正确的处理器版本后，将这些文件解压到您的工作目录下并确保您能读取“read-me”文件，同时根据需要对路径的设置以及赛灵思的环境变量。在您的工作目录中，您将注意到以下文件或应用，以及有用的read-me文件、许可证文件和用户指南。

- KCPSM6.Vhd: 这是一个真正的PicoBlaze源代码。
- KCPSM6.exe : 这是一个汇编程序，您可运用其生成所需的机器码和存储器文件。
- ROM_Form.vhd : 这是一个可执行汇编程序，使用该文件可以生成VHDL文件，您创建的程序将存储在这里。
- KCPSM6_design_template vhd : 这是一个PicoBlaze处理器模板实例。
- All_kcpsm6_syntax.psm : 这个文件是关于所有汇编程序命令和语法的定义。

```
NAMEREG s0,led ;rename S0 register to led
;As 8 bit processor we need four delay loops 256 * 256 * 256 *
256 = 4294967296
CONSTANT max1, 80 ;set delay
CONSTANT max2, 84 ;set delay
CONSTANT max3, 1e ;set delay
CONSTANT max4, 00 ;set delay
main: LOAD led, 00; load the led output register with 00
flash: XOR led, FF; xor the value in led register with FF i.e.
toggle
        OUTPUT led,01; output led register with port ID of 1
        CALL delay_init; start delay
        JUMP flash; loop back to beginning
delay_init: LOAD s4, max4;
            LOAD s3, max3;
            LOAD s2, max2;
            LOAD s1, max1;
delay_loop: SUB s1, 1'd; subtract 1 decimal from s1
            SUBCY s2, 0'd; carry subtraction
            SUBCY s3, 0'd; carry subtraction
            SUBCY s4, 0'd; carry subtraction
            JUMP NZ, delay_loop;
            RETURN
```

Figure 2 – 用于项目闪存LED的汇编程序代码片段

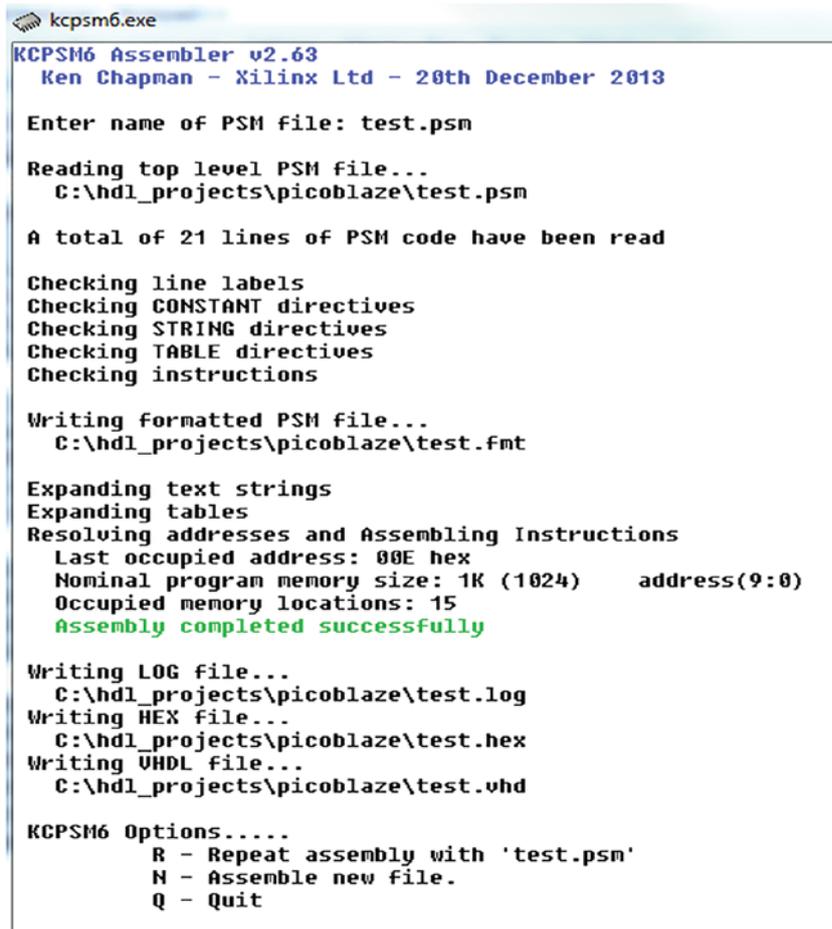


图3 –使用KCPSM6汇编程序生成存储器文件

在非常简单的层面,您只需在您的设计中 声明两个组件

对于我们的实例设计,如果你不想将 PicoBlaze 及其程序存储器添加到现有项目中,那么最后一步您应在我们能实例化 PicoBlaze 及其程序存储器的 ISE® 设计套件中创建一个新的项目。

一旦完成上述步骤,我们就准备开始在我们的应用中创建 PicoBlaze 处理器。在此最简单的层面上,您只需在您的设计中声明两个组件:一个是处理器本身,另一个是程序存储器,如图 1 所示(处理器位于左侧方框中,存储器位于右侧方框中,用以提供上下文环境)。当然,如果您需要实现一个以上的实例化,那么您应该提供若干个存储器组件,用于存储不同的程序。然而,首先需要我們做的是了解典型项目的开发流程。

开发流程

创建首个 PicoBlaze 实例其实很简单。第一步,使用 Notepad++ 之类的编辑器创建一个空白的文本文件。该文件应该具备一个 .PSM 的文件扩展名,比如像“test.psm”这样的。您可以用 PicoBlaze 汇编程序对微控制器进行编程。

赛灵思在 All_kcpsm6_syntax.psm 文件中对该语法进行了详细介绍,您可以自行下载。无论怎样,这个语法很容易理解和掌握。图 2 是汇编程序代码片段的示例,这是一个简单程序,用以闪存运行频率为 2Hz、时钟频率为 40MHz 的 LED。

一旦您对自己的汇编程序感到满意后,就可以进行下一步:通过您下载的汇编程序可执行文件运行该程序。

这样将会生成一个存储器文件(您的 FPGA 中会用到的 VHDL)、一个日志文件和一个十六进制文件,我们将在后文介绍它们的用途。图 3 显示了针对上述代码片段所运行的汇编程序流程。一旦运行汇编程序,您现在就能够在您的 FPGA 内实例化 PicoBlaze。

您现在拥有两个所需的 VHDL 文件:KCPSM6.vhd 和由汇编程序创建的包含您的应用的 VHDL 文件(在本例中为 test.vhd)。第二步:在您的 VHDL 设计中声明两个组件(KCPSM6 和 Memory)并对他们进行实例化(如图 4 所示)。在图 5 所示的代码片段中可以看到这个简单的 VHDL 示例。这个 VHDL 示例实现的 PicoBlaze 可以将 LED 闪存到 LX9 Spartan® 开发板上。

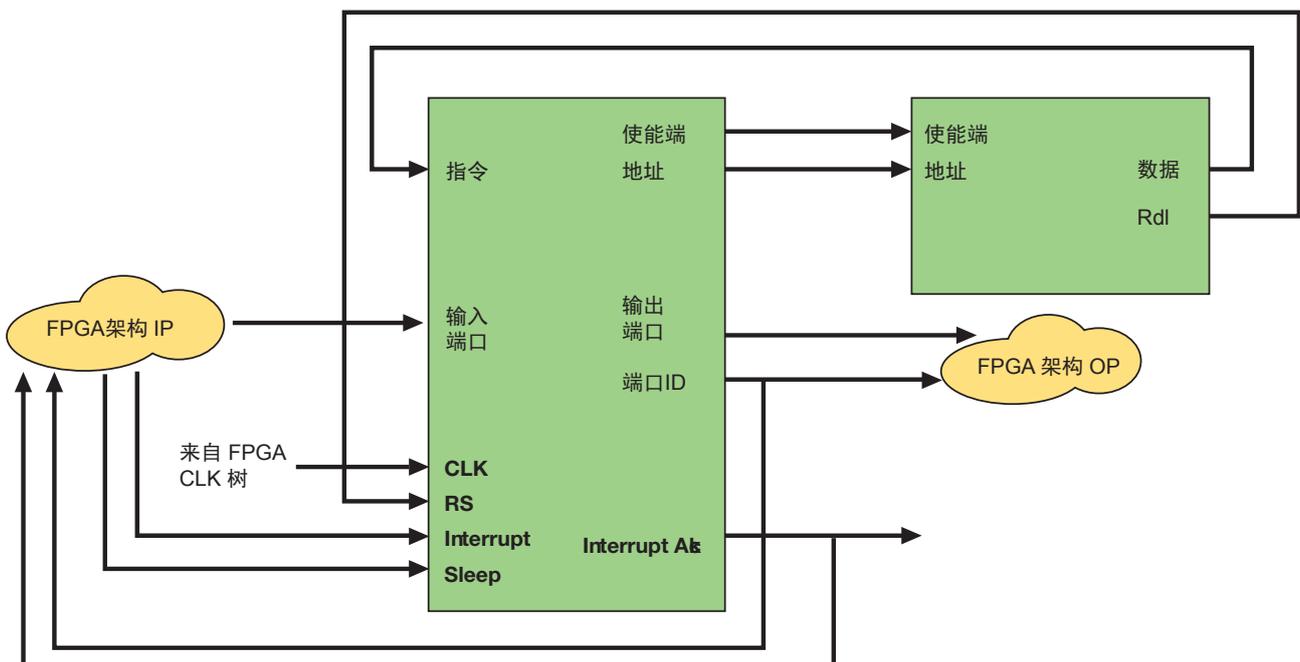


图4 - PicoBlaze上下文方框图

仿真与验证

一旦您在应用中完成了设计文件的实例化，您肯定希望能在综合和实现之前在仿真环境中验证一下该系统或模块的性能。由于 PicoBlaze 使用的是逻辑 Slice 和 Block RAM，所以您在 Mentor Graphics 的 Mod-elSim 或赛灵思 ISE 中的 ISim（如果您是在 Vivado® 设计套件中的 Xsim 中实现您的 PicoBlaze，也可使用 Xsim）等程序中进行仿真就极其简单。

由于 Block RAM 包含有针对您程序的指令，所以仿真工作自然就简单。基本上，您只需要提供一个时钟以及实例化要求的其他输入输出。图 6 显示了在 ISim 中进行的 PicoBlaze 仿真结果，以及指令加载之间的两个时钟周期。

升级您的程序

在 FPGA（以及比特文件）中内置 PicoBlaze 的最大优势之一就是在 FPGA 配置完成后 PicoBlaze 开始直接在其 RAM 中运行程序。然而，在某些情况下，您可能需要修改该处理器核正在执行的程序。虽然根据设计余下部分的复杂程度，您可能会重新运行 Implementation 阶段（包括更新了存储器文件），但这会非常耗时，尤其是您只在实验室里尝试各种可能的情况下。因此，在重新运行 Implementation 阶段之前可以用 JTAG loader 更新处理器核的程序存储器，来调试你的软件程序。其中 JTAG loader 也能通过下载获得。

在使用 JTAG loader 之前，首先需要您的设计设置中启用该加载程序。在一个程序存储器的实例化中使用通用 C_JTAG_LOAD-ER_ENABLE : integer := 1。注意您一次只能为您设计中的一个内存实例设置该参数。

在您的设计中启用该程序后，您必须从 JTAG_loader 目录中选择和你

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity pico_wave_top is
    Port ( clk : in STD_LOGIC;
          led : out STD_LOGIC_VECTOR (3 downto 0));
end pico_wave_top;

architecture Behavioral of pico_wave_top is

component kcpsm6 is
    generic( hwbuidl : std_logic_vector(7 downto 0) := X"00";
            interrupt_vector : std_logic_vector(11 downto 0) := X"3FF";
            scratch_pad_memory_size : integer := 64);
    port ( address : out std_logic_vector(11 downto 0);
          instruction : in std_logic_vector(17 downto 0);
          bram_enable : out std_logic;
          in_port : in std_logic_vector(7 downto 0);
          out_port : out std_logic_vector(7 downto 0);
          port_id : out std_logic_vector(7 downto 0);
          write_strobe : out std_logic;
          k_write_strobe : out std_logic;
          read_strobe : out std_logic;
          interrupt : in std_logic;
          interrupt_ack : out std_logic;
          sleep : in std_logic;
          reset : in std_logic;
          clk : in std_logic);
end component kcpsm6;

component test is
    generic( C_FAMILY : string := "S6";
            C_RAM_SIZE_KEYWORDS : integer := 1;
            C_JTAG_LOADER_ENABLE : integer := 1);
    Port ( address : in std_logic_vector(11 downto 0);
          instruction : out std_logic_vector(17 downto 0);
          enable : in std_logic;
              rd1 : out std_logic;
              clk : in std_logic);
end component test;

SIGNAL instruction : std_logic_vector(17 DOWNTO 0);
SIGNAL address : std_logic_vector(11 DOWNTO 0);
SIGNAL enable : std_logic;
SIGNAL rd1 : std_logic;
SIGNAL kcpsm6_output : std_logic_vector(7 downto 0);
SIGNAL port_id : std_logic_vector(7 downto 0);
SIGNAL write_strobe:std_logic;
begin

ram_inst : test PORT MAP (
    address => address,
    instruction => instruction,
    enable => enable,
    rd1 => rd1,
    clk => clk);

```

```

pico_inst : kcpsm6 PORT MAP (
  address => address,
  instruction => instruction,
  bram_enable => enable,
  in_port => (OTHERS =>'0'),
  out_port => kcpsm6_output,
  port_id => port_id,
  write_strobe => write_strobe,
  k_write_strobe => open,
  read_strobe => open,
  interrupt => '0',
  interrupt_ack => open,
  sleep => '0',
  reset => rd1,
  clk => clk);

output_ports: process(clk)
begin
if rising_edge(clk) then
if write_strobe = '1' then
-- 4 LEDs at port address 01 hex Spartan LX9
  Development board
    if port_id(0) = '1' then
      led <= kcpsm6_output(3 DOWNTO 0);
    end if;
end if;
end if;
end process;
end Behavioral;
    
```

图5 –将LED闪烁存入LX9 Spartan开发板上的PicoBlaze的代码片段

的操作系统匹配的版本并将其复制到您的工作目录下（即十六进制文件所在位置）。现在您可以打开一个命令窗口并导航到您的工作目录下，然后使用以下命令。

```
jtagloader -l <Your Project Name>.hex
```

注意：我已经我的操作系统重新命名为可执行文件版本，即 jtagloader.exe。

当您在您的最新 PSM 文件上运行汇编程序时，此项操作将下载所创建的十六进制文件，结果如图 7 所示。下载该文件后，您将注意到 JTAG loader 会停止处理器执行并在释放处理器复位之前下载新的程序到存储器中，此时处理器开始运行新的程序。

一旦您对 PSM 文件的更新行为感到满意，您就可以重新运行实现方案和比特文件生成，从而确保下次该器件经过配置，能够执行更新的程序。

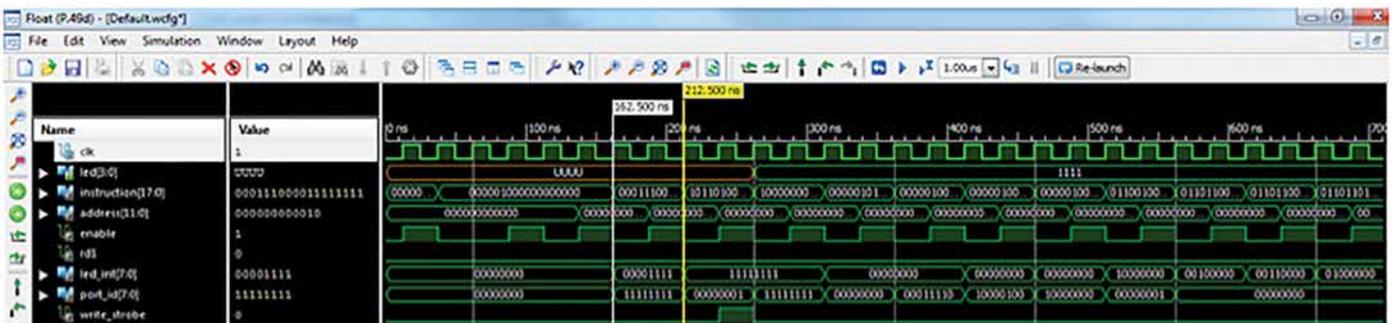


图6 – ISim仿真结果

```

c:\hdl_projects\picoblaze>jtagloader -l test.hex

          _ _ _ _ _
         / / / / /
        / / / / /
       / / / / /
      / / / / /
     / / / / /
    / / / / /
   / / / / /
  / / / / /
 / / / / /
/ / / / /

JTAG Loader by Kris Chaplin, Xilinx UK
Build : Date: Sep 17 2012, Time: 14:40:50
Target: Microsoft Windows 7 Service Pack 1, 64-bit.
Use the -h option if you need help

Selection: Load filename set to test.hex
Load filename: test.hex
Info:=====
Info:CABLE name used in scanChain: auto, cableArgCount=0
Info:=====
Info:Digilent Plugin: Plugin Version: 2.4.4
Info:=====
    
```

图6 – ISim仿真结果

通过实时改变使用率 研究FPGA功耗行为

电源管理一直是FPGA设计中的关键问题。现有一种新方法可用于测量真实FPGA器件的功耗估计值。

作者：Ahmet Caner Yüzügüler

Aselsan公司硬件设计工程师
acyuzuguler@aselsan.com.tr

Emre Sahin

Aselsan公司高级硬件设计工程师
emresahin@aselsan.com.tr

现

代的FPGA芯片能够开发高性能应用，但在这些设计中电源管理通常是一大限制因素。FPGA器件的资源使用最能决定设计的容量和处理速度，但是增加资源就会提高功耗。更高的功耗会提高运行成本、面积要求和结温，而设计人员必须用更多的气流和冷却系统来解决结温问题。

由于开发板或系统的总功耗极其重要，因此设计人员必须设置一个功耗预算，在资源使用和功耗问题方面兼顾平衡。所以，事先预测系统潜在功耗的能力可帮助设计人员获得先优势。

对于实现前的功耗评估，赛灵思可提供一些工具根据用户输入内容或综合报告进行功耗估计。其中一款工具就是赛灵思功耗估计器（Power Estimator, XPE）电子数据表。这款基于Excel的功耗估计工具能让您输入资源使用、翻转率和时钟频率等设计属性，并根据器件信息计算估计的功耗值。另一款常用的工具是赛灵思功耗分析器（Power Analyzer, XPA）。在布局和布线后，XPA会导入已生成的NCD文件，并利用实现详细信息和仿真结果（而非用户输入）来更加准确地估计功耗。

作为备选方案，我们设计出一种新方法来测量真实器件上FPGA设计的功耗估计值。为模拟不同的实现方案，我们创建了一个独立于器件的通用VHDL设计，其有可能实现在FPGA运行时通过串行通道改变激活的资源（即DSP slice、Block RAM和slice寄存器）的数量及相应的运行条件（结温、时钟频率和翻转

率)。我们的这种技术可以同时监控电源的电流和电压大小，便于我们轻松观察器件在不同资源使用和环境条件下的动态功耗特性。

我们已在赛灵思 KC702 评估板上实现了该设计。不过，只要您的 FPGA 器件支持设计中所使用的 IP 核，您只需做轻微修改也可在任何其它器件上实现该设计。

另一种使用该技术的方法可能是

作为面向 FPGA 板的 VHDL 测试设计。我们假设最近设计了一个通用型 Kintex®-7 开发板。客户可在不断变化的环境条件下利用未知的资源容量在开发板上实现任何设计。为了确保开发板的稳定性和稳健性，我们需要在最高和最低所需的环境温度下强加器件的工作极限，并验证 FPGA 能够支持不同的资源使用情况。然而，测试每种资源使用方案需要从头开始开发

一款全新的 VHDL 设计，这会耗费太多时间。我们推荐的方法可帮助设计人员灵活地根据需要对测试设计进行实时控制。

实现细节

我们设法让实现方案尽可能的简单，以避免生成作为逻辑使用的 slice LUT，因为我们无法对其功耗进行控制。实现 slice 寄存器最简单的方法是

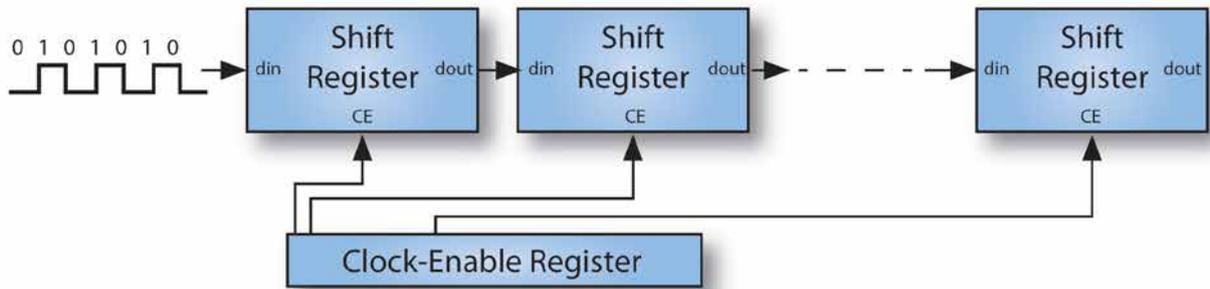


图1a

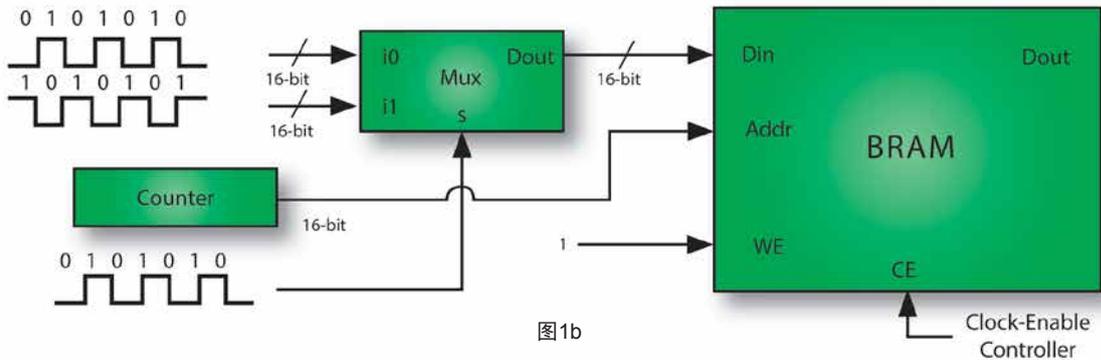


图1b

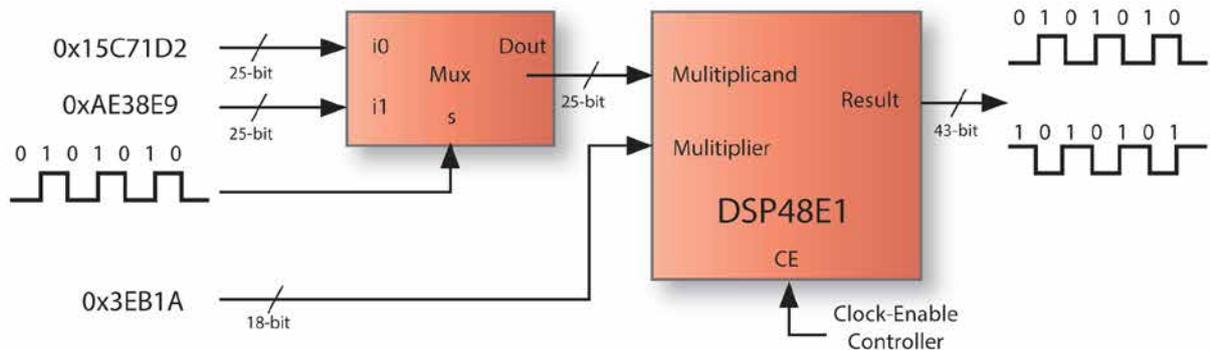


图1c

图1 - 实现方案方框图：(a) slice寄存器、(b) Block RAM和(c) DSP slice

将它们组合为移位寄存器模块。图 1a 中的方框图给出了 slice 寄存器的实现方法。这里需要注意一个简单的问题，当我们尝试创建 slice 寄存器时，综合工具常常将 slice LUT 作为 32 位移位寄存器（SRL32）使用，而不是使用 slice 寄存器。您也可以利用以下 VHDL 属性来强制综合软件使用 slice 寄存器：

```
attribute SHREG_EXTRACT : string;
attribute SHREG_EXTRACT of <signal_name> : signal is "no";
```

LogiCORE™ 模块存储器生成器创建作为单端口 RAM 的 Block RAM。将一个数位持续翻转的 16 位字不停写入激活 BRAM 的随机地址中，从而使它们保持被占用状态。图 1b 是单个 BRAM 组件的方框图。同样，我们使用 DSP slice 将一个 25 位被乘数与一个 18 位乘数相乘，得到一个 43 位输出，这就是单个 DSP48E1 可以处理的最大字宽。所有 DSP 组件的被乘数会被定期修改，从而使 DSP 消耗动态功耗，如图 1c 所示。然而，综合工具试图在综合过程中删除资源，因为这些模块的输出未与任何输出引脚连接。您可以使用以下属性使资源保持不变：

```
attribute KEEP : string;
attribute KEEP of <signal_name>: signal is "TRUE";
```

在该方案中，您可通过控制时钟使能信号来激活资源。时钟禁用后的资源只消耗非常小的功耗，可在设计中忽略不计。具体来说，具有 100% 翻转率的 50 个 DSP slice 的功耗为 0.112 瓦，而当它们的时钟使能信号取消断言时功耗为 0.001 瓦。不同使用

率下的结果几乎相同。因此，每个资源组件都由时钟使能信号通过串行通信进行控制的设计无需采用新的设计和实现步骤即可立即仿真出改变使用率时的情况。

资源的平均翻转率是显著影响功耗的重要因素。翻转率是指特定资源在每个时钟周期内输出信号的跳变次数。您可以交替改变资源的输入以使

输出在每个周期都改变其状态，从而实现恒定的 100% 翻转率。例如，要调整 DSP slice 的翻转率，您可以选择两个被乘数（连续交替）和一个乘数，这样输出的每位在每个周期中都会发生变化。输入信号的交替速率决定资源的翻转率；因此，我们通过串行通道就能即时控制某个资源类型的翻转率。

此外，功耗还与时钟频率成正比。我们使用混合模式时钟管理器（MMCM）生成具有可变频率的时钟。MMCM 输出的频率、相位和占空比由一组寄存器决定，而且一般只当实现设计和生成 bit 流文件时才进行初始化。然而，MMCM 的动态重

配置端口使我们能在 FPGA 运行时改变输出时钟等特性。时钟的相位和占空比不在考虑范围内，因为相位并不影响功耗，而占空比在大多数情况下也并不会改变。

另一方面，频率与功耗高度相关。MMCM 的内部工作机理是，

VCO 频率由 CLKOUT0_DIVIDE 寄存器的值进行分频，以获得所需的输出频率。我们根据[赛灵思应用指南 XAPP888](#)中介绍的状态机算法为寄存器赋予新值，实现无毛刺转换。与此同时，我们根据图形用户界面（GUI）中的用户输入实时改变设计的时钟频率。然后，我们能方便地观察不同频率下设计的功耗特性。

高温条件下的功耗行为是另一项需要认真观察和验证的重要问题。芯片的内核温度取决于开发板设计、处理速度、环境温度、散热片和风扇的气流。在我们的设计中，我们利用简单的 PWM 开关控制器改变位于 FPGA 芯片顶部的风扇的速度，从而局部地控制结温。此外，您也可以使用热风枪等外部加热工具来缩短加热时间。

用片上传感器测量内核温度，并通过 LogiCORE XADC 向导生成模数转换器 XADC。当您在 GUI 中输入参考温度值时，GUI 通过串行通道将参考值发送到器件，并与片上传感器测量到的内核温度进行比较。通过控制风扇的速度将结温稳定在所需的水平上。这样，您就可以观察功耗情况并绘制与内核温度的走势图，从而验证是否满足设计的功耗预算，以及功耗在所需的加热特性下是否保持在关键限值以内。

图形用户界面可用来与 FPGA 器件通信，以改变上述参数。图 2 给出了 GUI 的屏幕截图。我们首先通过标准 UART 利用其中一个可用端口连接到器件。然后，我们打开赛灵思功耗估计器将估计值与测量值进行比较。GUI 立即为使用默认参数值的估计功耗和实际功耗绘制功耗与时间关系图。您可在相关面板中修改这些参

数，并在资源（Resource）面板中利用翻转率指定所用的 slice 资源数量。使用扫描（Utilization Sweep）面板在等间隔下从 0% 至 100% 扫描特定资源的使用率，并利用每个间隔的测量功耗值绘制功耗与使用率关系图。

同样，您可在时钟选择（Clock Select）面板中改变 MMCM 的输出并扫描时钟频率。在下一栏的 Vccint 面板中可测量和连续绘制出 FPGA 芯片的电源电压。同时，内核温度（Core Temperature）面板显示并绘制出内核温度。用户可在提供的对话框中输入参考温度以改变结温。

您可利用集成在 KC702 开发板上的德州仪器（Texas Instruments）UCD9248 数字 PWM 系统控制器测量电源的电压和电流电平。这种用于电源转换器的多轨和多相位 PWM 控制器支持电源管理总线（PMBus）通信协议。其 PWM 信号可驱动用于调节 Vccint 电源电压的 UCD7242 集成电

路。一组 PMBus 命令可用于配置 IC 功能。UCD7242 包含片上电压与电流感应电路，并与 UCD9248 进行通信。我们的 GUI 软件不断向该器件发送相应的 PMBus 命令，同时接收标准 PMBus 数据格式的 DC/DC 转换器的电压和电流值。然后，我们将收到的字节转换为实际数值，并获得电源的电压和电流电平。

在高温下运行复杂、拥塞的设计可能会扰乱电源电压电平。不幸的是，FPGA 芯片对于输入电压的容差范围较小。超过该电平范围会导致芯片上的功能失效或者永久损坏。因此，您在设计完开发板后，还需要测试严苛工作条件下电压电平的稳定性。Vccint 面板可用来观察和验证设计开发板的电源系统是否能够承受高温下进行高速处理时的电压变化。

设计人员需要始终谨记的重要一点是：由于制造工艺变化，每个器件都会有不同的功耗。例如，在 Virtex-

7 FPGA 芯片制造完成后，会在测试阶段确定足够让芯片正确运行的最小电压源电平并写入电熔丝。将稳压器输出调节至这个最小电压电平可以改变静态功耗。因此，静态功耗的偏移会导致 XPE 与我们的设计在结果上出现差异。例如，XPE 中估计的 Kintex-7 芯片的静态功耗在典型情况与最差情况时计算相差 0.7 瓦。

完美匹配 XPE

我们在很多具有可变因素的不同设计方案中对我们的设计进行了测试，以确保其精确性和可靠性。同时，还在测量值的示意图中绘出 XPE 的估计结果，从而将功耗估计值与实际测量值进行比较。

通过最初将资源使用率从零至设计最大值扫描的设计实现可观察 Kintex-7 器件的功耗行为。图 3 给出了扫描一种资源类型的使用率过程中功耗的变化情况。直线和虚线分别代表实际功耗测量值和 XPE 的估计值。我们可以看到即使在使用率和功耗处于高值时两条曲线依然靠得非常紧密。

这些结果表明赛灵思功耗估计器计算值与实际测量值一致，可以准确地预测功耗。此外，我们还可以断定我们推荐的方法能够起到预期效果，并且可以作为 XPE 的替代方案。更好的选择是将两种方法结合使用。同时采用 XPE 和我们的方法来测试设计可对结果进行双重检查，确保您在 XPE 中修改数字时不会犯任何错误。例如，如果您没有在 XPE 中正确输入时钟的平均扇出数字，或者误解了一个

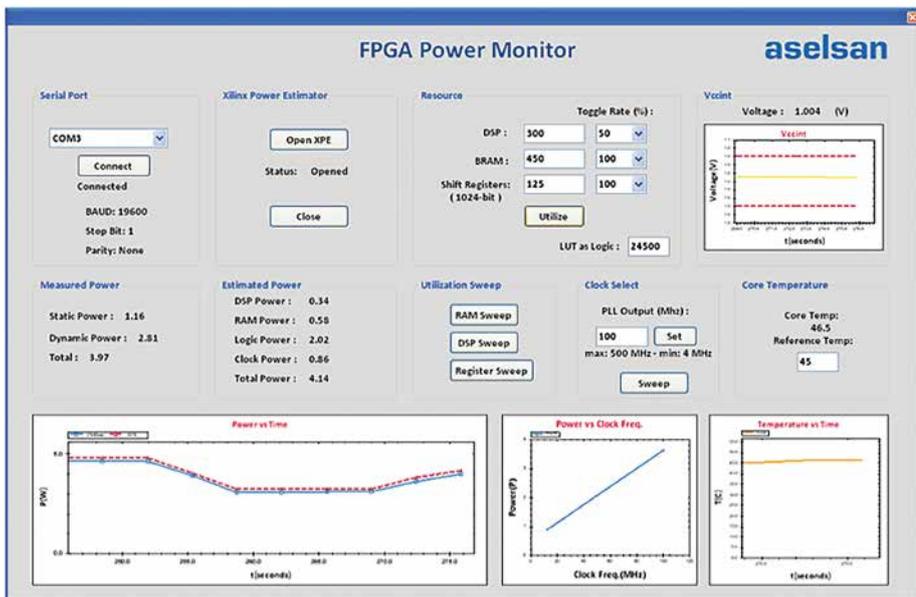


图2 – 我们的图形用户界面

概念并错误填写对话框，那么得到的示意图就无法重合，即表明有错误存在，这样您就可以校正输入值，防止出现误导性的估算结果。

FPGA 开发板的额定规范，例如最大功耗或允许的内核温度范围，总是包含在开发板的数据手册中。不过，具有较高资源使用率和时钟频率的设计会超出这些额定值。因此，必须确认功率调节器能为器件提供足够的电流，

而且散热系统足以将温度保持在 FPGA 设计的临界值以内。我们的方法可通过增加资源使用和时钟频率来测试开发板的可靠性能否满足高性能要求。这种测试有助于确定开发板上实现的设计所具备的最高信号处理速率。此外，我们还可以了解是否需要

升级开发板的功能和系统的冷却机制。

测试结果表明了我们的方法能够可靠地控制资源使用和评估功耗，因此工程师在设计前期阶段就多了一种电源管理选项。在采取进一步功能改进后，例如增加 JTAG 接口、完全独立于器件的 VHDL 代码以及通用的电流感应系统，这项设计将会成为 FPGA 项目的关键工具。🌈

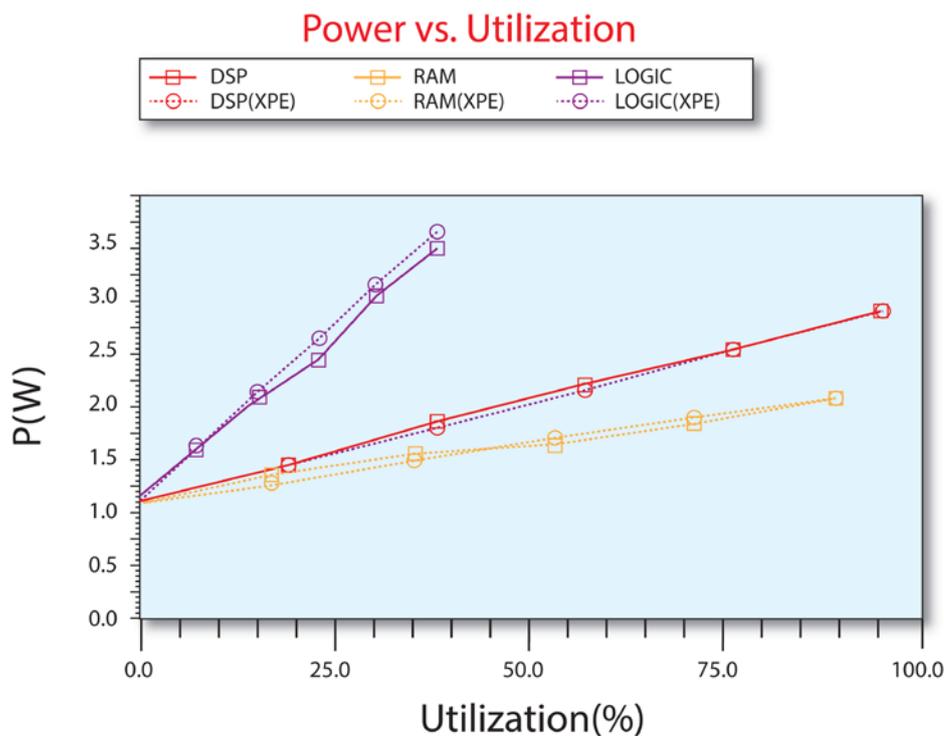


图3 – 在这个功耗与使用率关系图中，直线代表实际的功耗测量值，虚线代表XPE的估计值。

Xilinx推出业界首款FPGA 低时延25G以太网IP 解决数据中心应用吞吐量 难题

全新 25G 以太网 MAC 和 PCS
LogiCORE IP 在 2014 年国际超
算大会上亮相并演示

2014 年 11 月 17 日，中国北京 - All Programmable 技术和器件的全球领先企业赛灵思公司 (Xilinx, Inc. (NASDAQ:XLNX)) 今天宣布推出业界首款 FPGA 低时延 25G 以太网 IP，用以解决数据中心应用所面临的吞吐量难题。这款低时延 25G 以太网 MAC 和 PCS LogiCORE™ IP 解决方案不仅通过提供从 10G 到 25G 链路的迁移路径帮助降低数据中心的资本支出，还能将架顶式 (TOR) 开关和服务器之间的前面板带宽增加一倍以上 (10G 到 25G)，从而大幅提升性能。此外，这种独特的业界解决方案还支持新的 25G 以太网联盟规范，与业界持续发展的新一代处理器保持同步发展，从而使得未来数据中心服务器的性能可以实现翻番。

赛灵思将于 11 月 17 日至 20 日在美国路易斯安娜州新奥尔良市 Ernest N. Morial 会议中心举行的 2014 年国际超算大会上展示 25G 以太网 MAC 和 PCS LogiCORE IP 解决方案，届时欢迎莅临我们的 3903 号和 4006 号展台。现场演示采用两款 Virtex® UltraScale™ VCU107 开发板，其使用 5 米长的直连铜线和两个 QSFP+ 模块通过四通道 25G 以太网进行通信。

Vivado 2014.3 版本的最新消息

赛灵思不断改进其产品、IP和设计工具，努力帮助设计人员提高工作效率。本文将介绍赛灵思设计工具的当前最新更新情况，其中包括为加速赛灵思 All Programmable 器件设计而构建的以IP及系统为中心的全新一代革命性创新型设计环境 Vivado® 设计套件。如欲了解有关 Vivado 设计套件的更多信息，敬请访问：china.xilinx.com/vivado。

通过产品升级，赛灵思设计工具的功能将得到显著增强并新增一些新特性。保持及时更新升级是确保实现最佳设计结果的简单方式。

Vivado 设计套件 2014.3 版本可从赛灵思下载中心下载：china.xilinx.com/download。

VIVADO设计套件2014.3版本亮点

Vivado 设计套件的最新版本扩展了对 Virtex® UltraScale™ 器件的支持，并将 7 系列器件的编译时间缩短 20%。整体生产力增强特性包括：Vivado 高层次综合 (HLS) 的结果质量改进；Vivado IP Integrator 中流媒体和存储器映射 AXI 互联之间的自动连接；以及 Vivado IP Catalog 进一步丰富与扩展。

如需了解更多信息，请参阅 [Vivado 设计套件 2014.3 版本说明](#)。

常规使用

- Kintex® 和 Virtex UltraScale: XCKU035、XCKU040、XCKU060、XCKU075、XCKU100、XCKU115、XCVU065、XCVU080 和 XCVU095

早期试用

- Virtex UltraScale SSI 器件: XCVU125、XCVU160、XCVU190 和 XCVU440
- 如需使用这些器件，请联系赛灵思现场应用工程师。

VIVADO设计套件： 设计版本升级

Vivado IP Integrator

Vivado IP Integrator 将设计辅助功能提升到了全新的水平，能够在流媒体和存储器映射 AXI 互联之间实现自动连接，而且还进一步丰富了 Vivado IP Catalog。

部分重配置

赛灵思部分重配置技术现可对 Ultra-Scale 器件提供初步支持（仅支持 KU040 和 VU095 器件的实现）。在 ES2 芯片 (Virtex UltraScale) 或量产芯片 (Kintex UltraScale) 可用之前禁用比特流生成功能。

赛灵思还新增了比特流生成功能，包括每帧 CRC 选项，用于通过插入 CRC 值来实现部分比特流完整性检查。用户利用 `write_bitstream -cell` 可以请求仅提供特定的部分比特流。如需了解更多信息，敬请参阅：“Vivado 设计套件用户指南”的部分重新配置章节。

VIVADO设计套件： 系统版本升级

Vivado 高层次综合

赛灵思已增强了 Vivado HLS 的结果质量 (QoR)，同时将 fmax 平均提高了 10%，LUT 使用量减少了 5%，而且还支持 AMBA® AXI-4 接口调用。QoR 改善体现在：加强对 DSP 模块的支持；优化回路的控制结构，以及对操作延迟的高级用户控制。设计人员可利用最后这一特性对存储器端口延迟进行建模。

AXI4 存储器映射和流媒体接口的接口增强特性包括：改进对任意精度类型的建模。用户可访问有关尺寸资料。此外，主页与上下文有关，现可从指令编辑器直接访问。

DSP 系统生成器

该工具现可支持 MATLAB®2014B，并提供增强型以太网硬件协同仿真功能，包括自由运行时钟支持和板部分模块标签。通过亚稳态寄存器或用户定义的 RTL 扩展对多个异步时钟领域的支持，这样可以实现一个更完整的系统级设计。其它增强特性包括：

- DSP 系统生成器现可提供改进的 AXI4-Stream 接口，用于处理用户定义的 RTL 输入（黑盒）。
- 改进模型和波形查看器之间的交叉探测，加速调试与验证进程。
- 升级 CORDIC 和 DSP48 宏命令性能模型，可缩短仿真运行时间
- 改进缓存支持可加速复杂 IP 超级采样率 FIR 编译器的仿真建模，从而有能够实现 GHz 级采样率滤波器。

推出新ULTRAFast嵌入式 设计方法

赛灵思推出新的“UltraFast 嵌入式设计方法指南”，进一步丰富了 UltraFast™ 设计方法，使嵌入式设计团队在采用 Zynq®-7000 All Programmable SoC 创建智能系统时能够做出更明智的决策。该指南涵盖关键的原则、具体行为准则、最佳实践与技巧，避免走入误区。它还提供一些用户案例，归纳了从赛灵思内外部系统开发中吸取的经验与教训。

最佳实践主要面向由系统架构师和软件工程师以及硬件工程师组成的整个设计团队。现在可立即从下列网址中下载最新“UltraFast 嵌入式设计方法指南”：

China.xilinx.com/support/documentation/sw_manuals/ug1046-ultrafast-design-methodology-guide.pdf

VIVADO 快速入门教程

Vivado 设计套件快速入门视频教程是介绍 Vivado 设计套件内在特性与功能的示范视频。最新内容包括：2014.3 版本的最新消息；Vivado 激活浮动许可证生成、许可和激活简介；UltraScale 存储器控制器 IP 的使用，以及如何使用 report_cdc 来分析 CDC 结构问题。

在此观看所有快速入门视频：
china.xilinx.com/training/vivado

Vivado 培训

如需了解有关 Vivado 设计套件的讲师指导的培训课程，敬请访问：
china.xilinx.com/training

现在可从下列网址中立即下载 Vivado 设计套件 2014.3：
china.xilinx.com/download-load

Xilinx, Northwest Logic 和Xylon联合推出低成本 MIPI接口

基于 FPGA 的低成本 MIPI 接口，专门针对视频显示器和摄像头的。设计嵌入式系统 DSI 和 CSI-2 视频接口的用户现在即可采用低成本 MIPI 接口

2014 年 9 月 12 日，中国北京 - All Programmable 技术和器件的全球领先企业赛灵思公司 (Xilinx, Inc. (NASDAQ:XLNX)) 携手其高级联盟成员 Northwest Logic 和 Xylon 宣布推出基于赛灵思 FPGA 的低成本 MIPI 接口 IP，该产品专门针对成本敏感型视频显示器和摄像头而优化。MIPI 显示串行接口 (DSI) 和摄像头串行接口 2 (CSI-2) 已成为多种不同嵌入式系统中低成本视频显示器和摄像头的业界连接标准。赛灵思 FPGA 现可用来连接图像传感器和 ASSP，支持高带宽应用开发的 CSI-2 和 DSI 标准，满足 4K2K 乃至更高标准要求。此次针对 MIPI 支持的发布，进一步加强了赛灵思开发 Smarter Vision (智能视觉) 方案的承诺，至今已经可以支持先进的实时分析和沉浸式显示应用。

赛灵思公司广播和专业音视频业务细分市场负责人 Aaron Behman 指出：“MIPI DSI 和 CSI-2 现在是赛灵思 7 系列和 Spartan® 6 FPGA 连接低成本显示器、摄像头和视频应用处理器的首选标准。因此，当今越来越多基于 MIPI 的产品，尤其是支持 4K2K 乃至更高标准的产品，都在集成 FPGA。”

Xilinx联盟计划合作伙伴的 最新最佳技术

*Xpedite*重点介绍了赛灵思联盟合作伙伴生态系统的最新技术更新。

赛灵思联盟计划是指与赛灵思合作推动 *All Programmable* 技术发展的认证公司组成的全球性生态系统。赛灵思创建了这个生态系统，旨在利用开放平台和标准以满足客户需求并致力于帮助它取得长期成功。包括 IP 提供商、EDA 厂商、嵌入式软件提供商、系统集成商和硬件供应商等在内的赛灵思联盟成员助力提升您的设计生产力，同时最大限度地降低风险。下面为您分享一些精彩案例。

ALPHA DATA发布用于 FPGA 加速的赛灵思 ULTRASCALE开发板

认证联盟成员 [Alpha Data 公司](#) 与赛灵思合作提供面向大规模部署的商用产品中的首款 UltraScale™ FPGA。

Alpha Data 公司现成的开发板目录中现在包括赛灵思的 20nm Kin-tex® UltraScale 器件，可支持新型加速卡。Alpha Data 的 [ADM-PCIE-KU3](#) 是一款基于赛灵思 Kintex UltraScale KU060 ASIC 系列 FPGA 的高性能、可重配置、半长、低截面 x16PCIe® 板型开发板。

用户可以结合使用 Vivado® 设计

套件、高层次综合工具和 OpenCL 来创建开发板的应用。向客户提供的 IP 和参考设计能够帮助加速开发进程。在 Windows 或 Linux 系统中可以直接对该开发板的 Gen3PCIe/DMA、DDR3、自测试、开发板控制 / 状态等特性进行演示和评估。

ADM-PCIE-KU3 是 Alpha Data 公司以赛灵思 FPGA 为中心的大获成功的产品系列的最新成员，它是 AlphaData 与赛灵思十多年的经验和合作的成果。DM-PCIE-KU3 采用带宽为 25G 字节 / 秒的 16GB 双独立通道 DDR3-1600 ECC 存储器。该产品具有两个支持 10G/40G 以太网的 QSFP+ 端口、两个 SATA 连接器、电

压 / 温度 / 电流控制与监测装置，以及有源风冷散热器，所有这些组件均集成在 6.6 x 3.8 英寸的小型 PCIe 开发板上。

ADM-PCIE-KU3 极具价格优势，现可以立即订购供货。目前该产品可配套提供 Kintex UltraScale KU060-2 ES 器件。2015 年年中将配套提供 KU060-2 量产芯片。此外，该开发板还可配套提供包括实例设计和 PCIe/DMA IP 核评估的综合软件开发套件 (SDK)。

根据许可证还可提供其它 IP 和完整 PCIe/DMA 内核。如欲了解有关 ADM-PCIE-KU3 产品详情和订购选项的更多详情，敬请访问以下网址：
<http://www.alpha-data.com/ku3>

赛灵思 ZYNQ SOC助力美国国家仪器公司的最新系统级模块

联盟成员美国国家仪器公司 (NI) 基于赛灵思 Zynq-7000 All Programmable SoC 的 [sbRIO-9651](#) 系统级模块 (SOM) 将经过全面测试验证的硬件设计与完整的中间件解决方案以及 NI Linux 实时操作系统完美结合在一起。这一强大组合显著缩短了开发时间和产品上市时间，并大大降低了设计风险。此外，NI SOM 还提供了一种可替代硬件描述语言 (HDL) 的备选方案，从而可简化 I/O 连接和数据通信的任务。该产品使设计团队能够定制 SOM，且不会增加开发定制软件的时间和风险。

不像其它的 SOM 只提供有限的部署用软件，NI sbRIO-9651 集成了经验证的板级支持包 (BSP)、器件驱动程序和 NI Linux 实时操作系统。作为 HDL 的备选方案，NI [LabVIEW](#) 系统设计软件针对处理器和 FPGA 逻辑开发提供了带有数千种功能和 IP 块的图形开发环境。其主要优势包括：

- NI SOM 配套提供开箱即用的完整中间件解决方案，有效避免了开发嵌入式操作系统、定制软件驱动程序以及其它通用软件组件相关的时间和风险问题。
- LabVIEW FPGA 的集成度高，设计团队无需具备 HDL 专业知识就可充分发挥 FPGA 技术的强大功能。
- NI Linux Real-Time 是一个稳健可靠的基于 Linux 的实时操作系统 (RTOS)，使设计团队能够访问广泛的应用和 IP 社区。
- 完整的开发套件包括参考载板以及可以重复使用的设计文件。

- 设计团队可以使用 NI 的 CompactRIO 技术快速对其应用进行原型设计，然后用原型设计使用的相同代码进行部署，这样既省时又省力。
- SOM 产品建立在工业级 Zynq-7020 SoC 的基础上，经过精心设计、测试和验证，可长期可靠地用于部署和工业应用中。

TOPIC和赛灵思推出用于医疗系统的完整设计平台

赛灵思及其高级联盟成员 [Topic Embedded Systems](#) 公司联合推出了一款用于医疗系统的开发和部署平台。该平台完美组合了 Topic 的 [Dyplo](#) 框架和 Zynq SoC 的功能。医疗设备制造商可以在处理器和 FPGA 架构之间拖放传感器算法，即时查看性能和功耗的平衡。

Topic 的 Dyplo 框架与 Vivado 设计套件集成在一起，可以利用高层次综合和部分重配置功能来生成所有必需的硬件和软件编程文件。然后 Dyplo 可作为操作系统的中间件组件，从应用层中抽象出硬件并在运行时间管理动态硬件和软件配置。产生的设计可实现系统级系统，同时相对纯处理器实现方案而言可大幅降低功耗。

该平台包括 Topic 的 [Miami 系统级模块](#)，配套提供 Zynq XC7015 或 ZC7030 器件。Florida 载板具有一个 10 英寸的 TFT 触摸屏、SD 卡槽、Wi-Fi、蓝牙、以太网和 HDMI 等功能。板上 TI ADS1298 模数转换器支持 16 个 24 位 32kHz ECG/EMG/EEG 电极通道。硬件符合 ISO13485、EN 55021 A 类和 IEC 61000-4-2 EMC 标准，显著降低了开发风险。

完整开发套件可由 Topic 公司供

货，起价为 3,450 美元。该套件包含 ZC7030 Miami SOM、Florida 载板、8 个电极、Dyplo 的试用许可证、医疗参考设计以及 Linux 内核，通过可靠的空运供货。此外，Miami SOM 也可单独购买。如需了解更多详情，请联系：info@TopicEmbeddedProducts.com。

简化IP评估请求，加快XYLON IP核的设计进程和集成速度

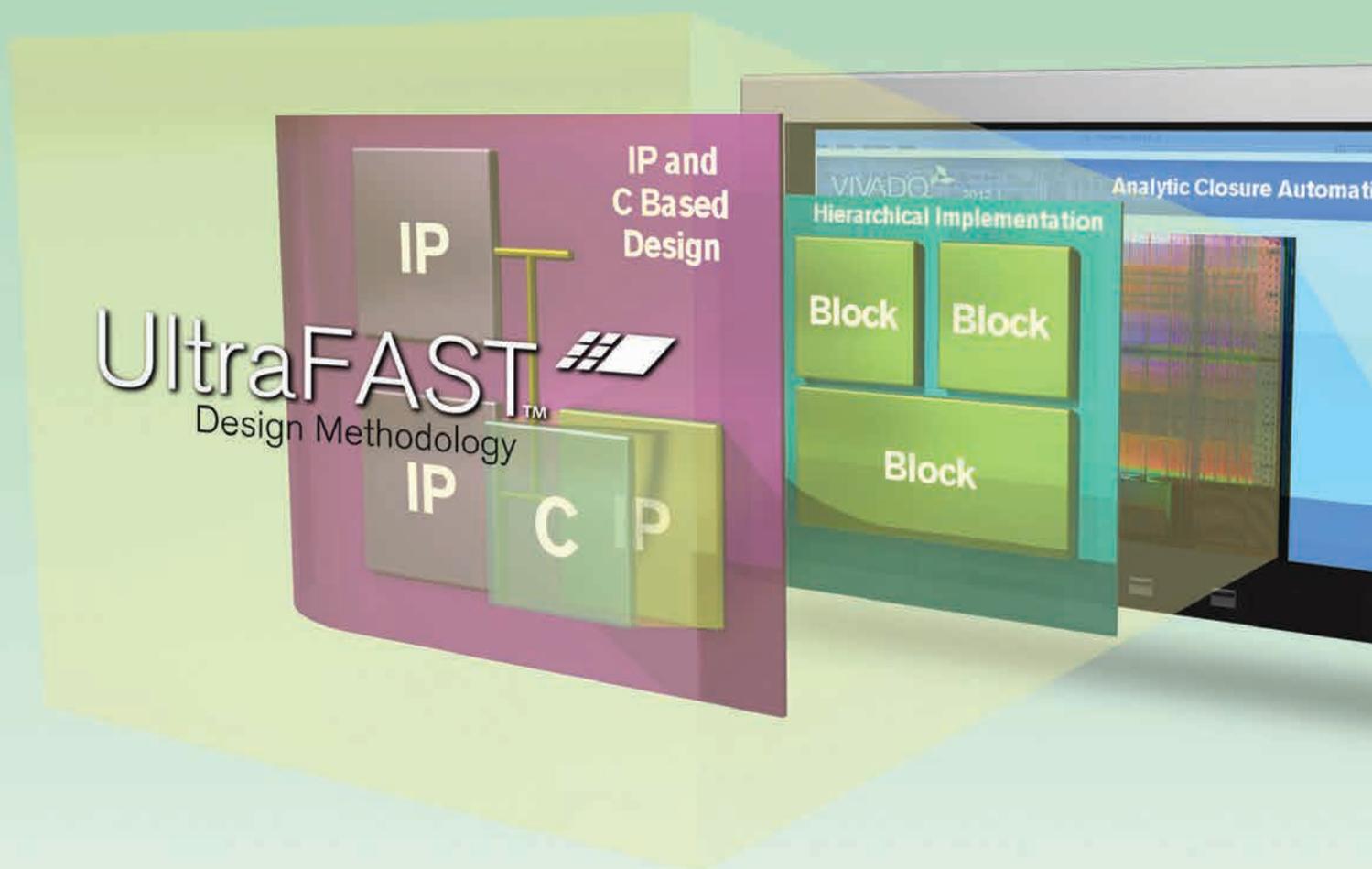
最新版 Vivado IP 集成器针对赛灵思联盟成员的 IP 核提供一键式 IP 评估请求功能。使用 2014.3 版本的客户现在可使用赛灵思高级联盟会员 [Xylon](#) 提供的 logicBRICKS 评估 IP 核，在未来版本中还将添加其他联盟成员提供的 IP。

经过验证，有 7 款 Vivado 优化型 logic-BRICKS IP 核可与 Vivado UltraFAST™ 兼容，并针对赛灵思 All Programmable FPGA 和 SoC 进行了精心优化。Vivado IP Catalog 列出了上述 7 款 IP 核以及相应的链接，以供下载评估版本。这样可以对高效的图像处理、视频处理以及接口 IP 进行快速评估，从而进一步丰富 Vivado IP Catalog 并缩短整个设计周期，这些内核包括：

- [logiVIEW 透视变换和镜头校正图像处理器](#)
- [logi3D 可扩展的 3D 图形加速器](#)
- [Bitmap 2.5D 图形加速器](#)
- [logiCVC-ML 小型多层视频控制器](#)
- [logiI2C 串行总线控制器](#)
- [logiI2S Audio I2S 发送器 / 接收器](#)
- [logiSDHC SD 卡主机控制器](#)

赛灵思推出

Vivado[®] 设计套件的 UltraFast[™] 设计方法



赛灵思的 UltraFast[™] 设计方法可加速设计进程并可预测设计周期。