

赛灵思 中国通讯

Xilinx News

第五十六期 2015年 夏季刊 Issue 56 Summer 2015

All Programmable Abstraction: 你的编程你做主

使用 FPGA 优化视频水印操作的 OpenCL 应用

全面掌控频域

实现用于专业视频的 JPEG 2000 网络

Vivado 2015.1 版本的最新消息



在低成本 FPGA
开发板上实现
Oberon 系统

30

 **XILINX**
ALL PROGRAMMABLE™

<http://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。

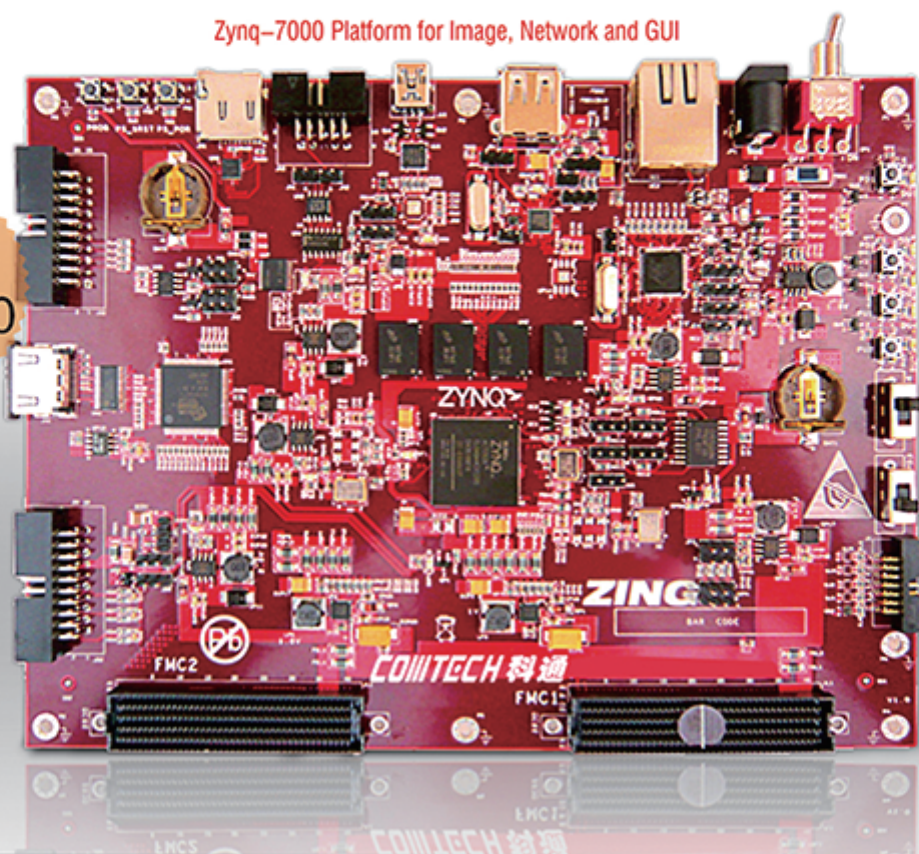


ZING Board

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

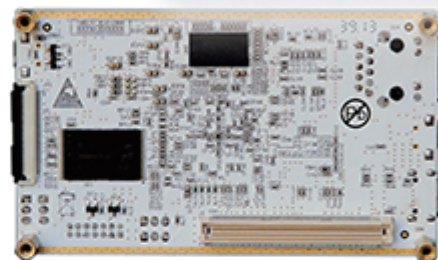
融合软/硬件协同编程，开启定制化SoC设计时代

SNOWLeo

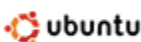


主要特性

- 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

Xcell journal

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

为更高的设计团队生产力做好准备

欢迎订阅2015年春季刊《赛灵思中国通讯》。在本期诸多佳作中，您将了解到赛灵思“全可编程抽象化（All Programmable Abstraction）”战略的三个_{方面}。所谓“**All Programmable Abstraction**”，是指赛灵思及其联盟成员推出的全新高级设计入门环境，帮助我们在FPGA设计中使用熟悉的软件编程模式。这些开发环境便于设计团队提高生产力，甚至能帮助那些从未使用赛灵思全可编程（All Programmable）FPGA或SoC编程的设计人员在没有硬件工程师的协助下使用这些器件。

在封面报道中，我将介绍赛灵思开发环境的发展演进，更好地支持高级设计入门。“**All Programmable Abstraction**”，始于2012年赛灵思Vivado HLS高层次综合工具的发布，以及MathWorks和NI等赛灵思联盟成员第三方设计环境的推出。多年来，该计划不断发展，最近赛灵思还推出了SDx系列开发环境的三个全新成员：**SDNet™**、**SDAccel™**和**SD-SoC™**。

利用上述全新的开发环境，整个设计团队能进一步提升生产力，预先构建出完整架构级系统。这有助于节约后端时间，帮助团队在软件开发环节占得先机。不过令人印象更深刻的一点，或许是这些环境能帮助新用户_{在系统组合中添加赛灵思FPGA和SoC}，从而无需FPGA专家帮助就推出新的应用或创新。近期业界分析认为，全球软件工程师和硬件工程师的人数为10:1。因此，SDx环境将帮助更广泛的人群打造出基于**All Programmable**器件的创新产品。

本期内容还包含SDAccel开发环境的两个实际应用实例。您将在第38页和第42页上看到这两个实例。

我们也很高兴在本期中展示计算机工程设计领域传奇人物Niklaus Wirth所做出的贡献。他开发的高级语言移植到了Spartan®-3 FPGA平台上。没听说过Wirth教授吗？他发明了Pascal语言和_{一些后续语言}，能帮助编程人员变成系统发明者。

Wirth教授现已退休，他修订并更新了自己编写的《Oberon项目》一书，帮助学术界向下一代计算机专业_{人士讲授系统编程知识}。原书使用的处理器现在已停产。由于找不到合适的商用替代品，Wirth用Digilent的低成本Spartan-3开发板设计了自己的处理器，因此价格也很适合学生使用。

他的文章讲述了使用Spartan-3开发板更新Oberon编程语言的设计经验，希望能激励下一代创新者。

最后，在第54页的技术文章中，赛灵思的Daniel Michek将帮助读者一步步使用Vivado™ DSP系统生成器工具，打造最佳硬件平台。

希望您喜欢本期的内容。



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 Xilinx, Inc. 保留所有权利。本文包含的赛灵思、赛灵思徽标和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

发行人致语

为更高的设计团队生产力
做好准备... 4



实际应用中的出色表现

广播领域的出色表现

实现用于专业视频的JPEG 2000网络... 14

科学应用领域的出色表现

“大白兔”：当每纳秒都变得重要... 18

无线通信领域的出色表现

RF-DAC多频带发射器线性评估... 26

Xperiment

在低成本FPGA开发板上实现Oberon系统... 30

在数据中心领域的出色表现

清除通向基于FPGA的OpenCL数据中心服务器的障碍... 38

在数据中心领域的出色表现

使用FPGA优化视频水印操作的OpenCL应用... 42



封面 专题

8

All Programmable Abstraction:

你的编程你做主



THE XILINX XPERIENCE FEATURES

手把手课堂：FPGA 101

全面掌控频域... 48

手把手课堂：FPGA 101

将SoC平台设计与
DSP系统生成器相集成... 54

手把手课堂：FPGA 101

重新思考快速宽频
ADC中的数字下变频... 58



42



XTRA READING

号外，号外

Vivado 2015.1版本的最新消息... 64

实例

Xilinx联盟计划合作伙伴的
最新最佳技术... 66



All Programmable Abstraction: 你的编程你做主

赛灵思的全新 SDx 软件定义环境配合 Vivado IPI、HLS 和备受青睐的系统级设计工具。



作者: Mike Santarini

赛灵思杂志出版商

赛灵思公司

mike.santarini@xilinx.com



为将设计团队的生产力提升到全新的水平, 并将 All Programmable FPGA、SoC 和 3D IC 推向更广泛的软件工程师用户, 赛灵思近期新增了两款 SDx™ 开发环境系列成员。新推出的 SDAccel™ 开发环境使没有任何 FPGA 经验的数据中心设备编程员都能够运用 OpenCL™、C 或 C++ 语言针对数据中心和云计算基础设施对赛灵思 FPGA 进行编程, 设计出的 FPGA 设备性能功耗比远超基于 GPU 和 CPU 的设备。赛灵思还推出了 SDSoC™ 开发环境, 支持同样没有 FPGA 经验的软件开发人员用 C 或 C++ 语言基于赛灵思及第三方平台开发商推出的 Zynq®-7000 All Programmable SoC 和 UltraScale+™ MPSoC 平台创建各种系统。

SDx 环境是赛灵思 All Programmable Abstraction 计划的最新产品。该计划旨在帮助软件工程师和系统架构师方便地编程赛灵思器件, 同时开发环境可根据他们的需求量身定制。

赛灵思公司企业战略与市场营销高级副总裁 Steve Glaser 指出: “SDNet、SDAccel 与 SDSoC 这一完美组合将为系统和软件工程师提供熟悉的类似 CPU、GPU 和 ASSP 的编程环境, 这些工程师将首次尽享 All Programmable 器件带来的独特优势,

诸如定制加速功能, 性能功耗比提升 10 倍乃至 100 倍, 任意设备间连接, 以及提供下一代智能系统所需的保密性和安全性。赛灵思正助力下一代系统做到进一步互联, 实现软件定义和虚拟化, 同时还必须支持基于软件的分析, 更多云端计算功能, 这通常是无处不在的视频和嵌入式视觉应用所需要的。这就要求 SDx 软件定义编程环境和异构多处理技术采用新型 UltraScale FPGA 和 MPSoC。”

2014 年春季 SDNet™ 开发环境 (详见《赛灵思中国通讯第 52 期》) 推出后不久就推出了 SDAccel 和 SDSoC 这两款新的开发环境。

SDx 新环境在帮助软件工程师和系统架构师对赛灵思器件的 FPGA 部分进行编程的同时, 也能让拥有硬件工程设计资源的设计团队的生产力更高, 且更快集成到优化的系统上, 进而加速产品上市进程。有了可行的系统设计, 硬件工程师能集中精力优化 FPGA 布局 and 性能, 实现更高的系统效率, 同时软件工程师则能进一步优化应用代码。

All Programmable Abstraction

赛灵思于 2008 年在新任 CEO Moshe Gavrielov 领导下开始规划 7 系列 All Programmable 系列 FPGA、3D IC 和 Zynq-7000 All Programmable SoC 时, 就明显发现 7 系列和未来产品线的所有产品均具有丰富的功能, 完全能让客户将赛灵思器件作为最新最具创新性产品的核心。

这些 All Programmable 器件比赛灵思早些年代的胶合逻辑 FPGA 高级得多, 实现了任何其他架构所无法企及的系统功能和最终产品差异化。为了最大化这些最新器件的价值, 在竞争

中脱颖而出，管理层认识到赛灵思必须开发出相关工具和方法，帮助系统架构师乃至嵌入式软件开发员（而仅仅只有 FPGA 专家），去进行赛灵思最新器件的编程工作。此外，公司还必须针对高增长市场，为软件工程师开发出设计环境，并根据设计人员习惯使用的工具和流程来定制这些环境。此外，加强与 MathWorks 和国家仪器（NI）等公司联盟也势在必行，这些公司已经构建了各种环境，使非传统 FPGA 用户能够充分利用赛灵思 All Programmable 器件的功率效率和灵活性优势。

对老客户来说，如果能为设计团队的每个成员都提供相应的设计环境，那就能确保高效性，缩短产品上市时间。如果这些环境足够高级，就能真正推广 All Programmable FPGA 和 Zynq SoC 设计，让那些没有 FPGA 设计经验的架构师和软件工程师无需硬

件设计人员帮助就能进行器件编程。全球软件工程师人数与硬件工程师人数比为 10:1。因此，提供这种开发环境（或支持这种开发环境的硬件平台）的赛灵思及其联盟计划合作伙伴既能扩大用户群，又能增加营收。

该战略和支持软件工程师及系统架构师利用根据其设计需求量身定做的环境进行赛灵思器件编程的后续开发工作就是赛灵思所谓的 All Programmable Abstraction（如图 1 所示）。

VIVADO HLS和IPI: 设计抽象的第一步

2011 年，设计抽象迈出了重大的第一步，那就是赛灵思收购了私企 AutoESL 公司的高层次综合（HLS）工具。继并购之后，赛灵思于 2012 年公开发布了集成到 ISE® 设计套件和 Vivado® 设计套件工具流程中的 HLS

技术。根据 Berkeley Design Automation 的详细调研（结果表明 AutoESL 的 HLS 工具最简便易用，而且是 EDA 产业所有 HLS 工具中能提供最优秀结果质量（HLS），赛灵思选择了 AutoESL 技术。HLS 源于 EDA 世界，这也意味着工具使用模型面向的是 ASSP 和片上系统（SoC）架构师以及拥有 C 和 C++ 编程经验、同时对硬件描述语言（Verilog 和 VHDL）及芯片设计需求有一定工作了解的成熟设计团队。

利用 Vivado HLS，具有丰富技能的架构师和设计团队能用 C 和 C++ 创建算法，同时利用 Vivado HLS 来编译并转换那些算法为 RTL IP 模块。随后，FPGA 设计人员能将这个模块及其它模块集成到创建或授权的 RTL 中，并用赛灵思的 IP Integrator（IPI）工具将 IP 汇编到设计中。随后，FPGA 设计人员能将汇编设计一步步通过 Vivado 流程，执行 HDL 仿真、时序与功耗优化、布局布线等。最终，设计人员生成网表 / 比特文件并配置目标 All Programmable 器件的硬件。如果设计包含处理器（Zynq SoC 或 MPU 软核），配置后的器件随即可供嵌入式软件工程师编程。

为了帮助嵌入式软件工程师应对繁杂的编程工作，赛灵思提供了基于 Eclipse 的集成设计环境，即赛灵思软件开发套件（SDK），其包含编辑器、编译器、调试器、驱动程序和面向 Zynq SoC 或 FPGA 的库，其内嵌了赛灵思的 32 位 MicroBlaze™ 软核。该 SDK 于 10 多年前推出，随着赛灵思器件上处理器集成的发展变化而得到了大幅提升，此前器件上集成硬化的 DSP Slice 和软核 MCU 和 MPU（8

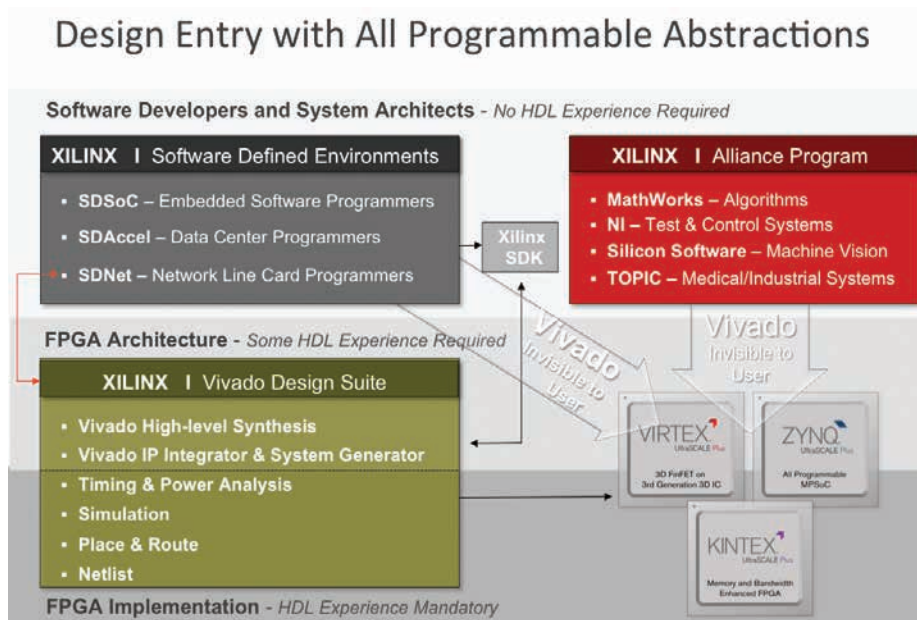


图 1 – 利用新的 SDx 设计环境和联盟成员提供的环境，赛灵思正帮助更多创新者充分发挥赛灵思 All Programmable 器件的作用，为下一代产品带来更大增值。

位、16位和32位)，到 Virtex®-4 和 Virtex®-5 FPGA 时已转变为集成硬化的 32 位 PowerPC®；到 Zynq SoC 已转化为集成 32 位 ARM® 处理器；而到即将推出的 Zynq UltraScale+ MPSoC 则发展到集成 64 位 ARM 处理器。

联盟成员为更多用户带来增值

10 多年来，赛灵思与国家仪器 (NI) 和 MathWorks 公司一直保持密切合作。这两家公司致力于为其服务的特定用户提供量身定制的独特高级开发环境。

美国国家仪器 (NI) (德克萨斯州奥斯汀) 提供的硬件平台深受控制和测试系统创新者欢迎。赛灵思的 FPGA 和 Zynq SoC 为 NI RIO 平台提供动力。NI 的 LabVIEW 开发环境是用户友好型图形化程序，底层运行 Vivado 设计套件，因此 NI 的客户无需了解任何 FPGA 设计细节。一些人或许根本不知道 RIO 产品的核心其实是赛灵思器件。他们只需在 LabVIEW 环境中对其系统编程即可，NI 硬件能提升其开发的设计性能。

就 MathWorks (马萨诸塞州纳蒂克) 而言，该公司 10 多年前为其 MATLAB®、Simulink®、HDL Coder 和 Embedded Coder 添加了 FPGA 支持，底层运行赛灵思的 ISE 和 Vivado 工具，而且完全自动化。因此，公司的用户——主要是数学算法开发人员——能开发算法并大幅提升算法性能，而且能在 FPGA 结构上简便运行算法。

10 年前，赛灵思为其 ISE 开发环境添加了 FPGA 架构级工具——系统生成器 (System Generator)，近期又给

Vivado 设计套件添加了，以帮助具备 FPGA 知识的团队进一步调节设计，实现算法性能增益。MathWorks 和赛灵思两家公司技术的强强组合，有助于客户公司推出成千上万种创新产品。

近期，其他公司也开始为赛灵思环境做出贡献。赛灵思近期迎来 TOPIC Embedded Systems 和 Silicon Software 这两家欧洲公司加入其联盟计划，特别是欢迎其针对医学和工业市场的高级开发环境。

[TOPIC Embedded Systems](#) (荷兰埃因霍温) 拥有独特的开发环境 DYP-LO, 《赛灵思中国通讯第 54 期》对其进行了详细介绍。该环境面向 Zynq SoC，很快也将面向 Zynq MPSoC，对系统级设计采用统一方法，从而使系统架构师能够用 C 或 C++ 创建系统设计结果，并在 Zynq SoC 的双核 ARM Cortex™-A9 处理系统上运行该结果。一旦用户发现设计某些部分在软件中运行太慢，可以将这些部分拖放到窗口中，将 C 语言转化为 FPGA 逻辑 (底层运行 Vivado HLS)，并将其放入 Zynq SoC 的可编程逻辑中。来回交换代码片段，直到达到最佳系统性能。TOPIC 最初用于医疗设备开发，目前也开始与工业设备制造商合作。

[Silicon Software](#) (德国曼海姆) 是支持软件工程师利用赛灵思 All Programmable 器件的最新联盟成员。公司的 VisualApplets 图形图像处理设计环境可帮助面向 Zynq SoC 平台的系统架构师和软件工程师打造工业机器视觉应用创新，而且无需硬件工程师的协助。在 2013 年 SPS Drives 行业展会的赛灵思展台上，Silicon Software 展示了其用 VisualApplets 环境开发

的光学检查系统。演示显示通过采用 VisualApplets 环境将图形处理任务从 Zynq SoC 的处理系统转交给器件的 FPGA 逻辑来执行，能将系统性能提升 10 倍。

借助 SDx 推广 FPGA 和 SOC 设计

借助 SDx 软件定义开发环境，赛灵思将面向关键市场的软件开发人员和系统架构师构建一系列高级的设计入门环境。SDAccel 和 SDSoc 在底层自动运行整个 Vivado 流程，无需直接使用 Vivado 工具，也无需硬件工程师的协助。SDNet 不会访问 Vivado HLS，针对线路卡架构师，其提供了独特的“两步使用模型”。

在第一步中，线路卡架构师采用直观的类型 C 语言的高级语言而不是晦涩的微代码来设计需求，并制定网络线路卡规格描述。SDNet 开发环境根据规格描述生成 RTL 版设计。流程随后需要硬件工程师在目标 FPGA 中实现 RTL。

在第二步中，SDNet 还允许网络公司用高级语言测试和更新协议，并升级线路卡的功能，即便在现场部署后也能做到，而且无需硬件设计参与。这个流程有助于企业快速创建并更新线路卡，这种灵活性对软件定义网络而言非常重要。

SDAccel 和 SDSoc 这两个新设计环境将 SDx 的理念推广到新的应用领域。

SDACCEL 可优化数据中心性能功耗比

《数据中心学刊》(Data Center Journal) 2014 年 3 月的一篇文章指出，谷歌、Facebook、亚马逊、领英等公司核心部分——数据库——“消耗的电力占

全球电力 3% 以上，同时产生 2 亿公吨 CO²”。如此巨大的能耗使得其数据中心一年电费超过 600 亿美元。即便对最大规模网络公司来说，功耗也会严重削减盈利性，而且对环境也会产生不可估量的影响。

随着越来越多的企业期望采用云计算和大数据分析技术；视频和流媒体的全球性普及，以及越来越多的人加入无线网并准备升级到未来 5G 网络，对数量更多、性能更好的数据中心的需求呈几何级无止境增长趋势。《数据中心学刊》文章指出，从当前趋势看，到 2017 年数据中心流量预计将达 7.7ZB，这意味着数据中心的功耗如果不能得到有效控制将会出现大幅增长。当今大多数数据中心的功耗主要源于其基础组件——Intel x86 处理器。目前 MPU 能提供良好但不是最佳的性能，而且功耗很高。

目前全球庞大的的软件工程师队伍均发现 MPU 是最容易编程的器件。要解决数据中心性能问题，许多公司一直在设法构建采用图形处理单元（GPU）或通过 GPU 实现 CPU 系统加速的设备。GPU 的性能大大超越数据中心的 CPU，但不幸的是功耗要高很多。性能虽然很高，但功耗巨大。

为集两者之大成，许多企业均转而采用以 FPGA 为中心的方法，将 FPGA 与其它处理器结合使用，以实现数据中心设备性能最大化。

许多数据中心设备厂商已证实，分立式 FPGA 结合分立式 CPU 使用，虽然每个卡的功耗会增加一点点，但性能却得到大幅提升，从而可大幅提升性能功耗比。还有人认为，通过采用一颗在单个 SoC 上集成 x86 处理器内核和 FPGA 逻辑的芯片，有望进一步提升性能功耗比。还有人认为，在

单个 SoC 上集成 FPGA 逻辑和 64 位 ARM 处理器 IP 似乎能带来更低功耗但同样高性能的解决方案。

在数据中心使用 FPGA 的主要障碍就是编程问题。数据中心开发人员习惯于 x86 架构编程，这些开发人员通常只有纯软件编程背景。帮助开发人员将 CPU 程序转向更快的 GPU 的第一步就是业界 OpenCL 语言的开放开发。过去 2 年来，OpenCL 取得进一步发展，使客户能够针对 FPGA 进行编程，从而为未来数据中心设备架构乃至无所不在网络创造了新的机遇。

通过推出 SDAccel 环境，赛灵思弥补了编程鸿沟，并为数据中心工程师使用 OpenCL、C 或 C++ 编程 FPGA 平台铺平了道路，而且无需硬件工程师介入。赛灵思公司设计方法市场高级总监 Tom Feist 表示，针对 OpenCL、C 和 C++ 的 SDAccel 开发环境使数据中心编程人员能够打造出比 CPU 和 GPU 系统性能功耗比高出 25

倍的设备。

Feist 指出，在 SDAccel 流程中（见图 2，SDAccel 开发环境演示），x86 CPU 结合运行赛灵思 20nm Kintex[®] UltraScale FPGA 的加速卡，软件开发人员能发现需要加速的应用，并用 OpenCL 编码和优化内核，再为 CPU 编译和执行应用。随后他们再来评估并调试内核，直到找到周期精确的模型。接下来，再用 SDAccel 编译代码，并自动实现在 FPGA 中（可能因为 Vivado 在底层运行）。随后就能运行应用，验证通过卡加速后的应用性能以及与单纯用 CPU 运行的性能对比。Feist 指出：“他们能循环运行迭代，直到找到性能和功耗之间的最佳平衡，相对于 CPU 和 GPU 实现方案，性能功耗比可提升多达 25 倍。”

在本期《赛灵思中国通讯》中，Feist 和他的同事详细介绍了 SDAccel 环境，后续文章则介绍 SDAccel 的工作原理（见第 38 页和 42 页）。

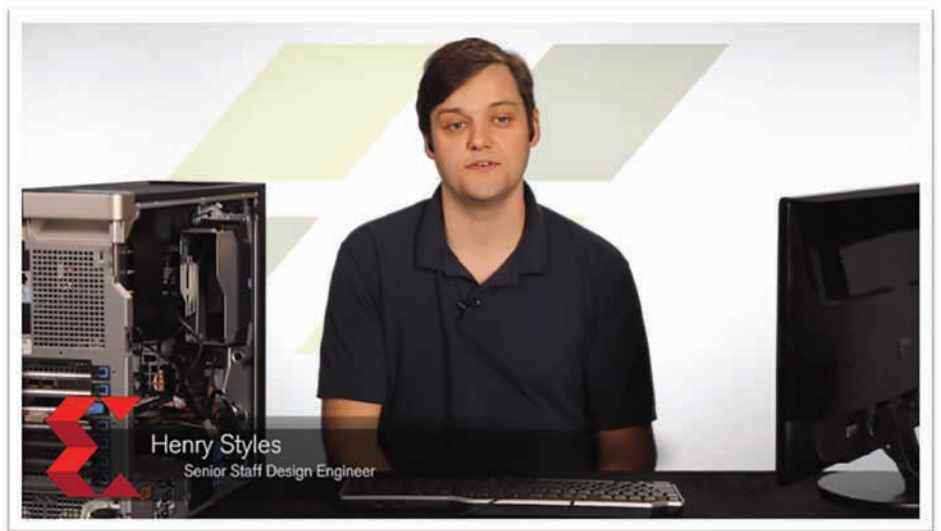


图2 – 在本SDAccel开发环境演示中，工程师 Henry Styles介绍了如何通过使用内含Alpha Data ADM-PCIE-7V3加速器的标准64位x86工作站来运用SDAccel开发环境实现加速。



图3 – 在本SDSoC开发环境演示中，首席工程师Jim Hwang用SDSoC构建简单的图形处理流水线以检测运动，并在以60fps速率运行的实时HD 1080p视频中插入运动边缘。

SDSoC让嵌入式系统创新者辈出

SDNet 让线卡开发人员能够利用独特的软定义方式快速创建下一代网络，而 SDAccel 则能让数据中心设备厂商的下一代数据中心设计的实现最佳性能功耗比。新的 SDSoC 开发环境或许能为赛灵思用户群带来最广泛的影响。这是因为 SDSoC 针对广泛的嵌入式系统设计团队，特别是软件工程师，这些软件工程师设计面向的是赛灵思 Zynq SoC 所服务的大部分市场。SDSoC 环境现在能让用户配置逻辑——不光是对运行 Zynq SoC 硬件平台的嵌入式系统处理器编程——而且能使用 C 和 C++ 语言。

SDSoC 产品经理 Nick Ni 指出：“软件开发人员习惯于对主板、ASSP 平台和 ASIC 进行编程，且无需硬件工程师参与。有了 SDSoC 开发环境，他们就用类似 ASSP 的方式对 Zynq SoC 和 MPSoC 平台进行编程。SDSoC 的独特之处在于现在我们能 Eclipse IDE 环境中用 C 或 C++ 语

言面向 Zynq SoC 和 Zynq UltraScale+ MPSoC 平台创建完整的系统设计。

Nick 表示，使用 SDSoC 环境，嵌入式软件开发人员能运用 C 或 C++ 语言创建设计并进行测试，以发现 Zynq SoC 处理系统上哪些部分运行得不佳。一旦发现可疑代码，便突出显示并命令 SDSoC 环境自动将该代码分区到 Zynq SoC 的可编程逻辑中，以加速系统性能。Nick 指出，SDSoC 环境只需点击按钮就能把软件功能移到 FPGA 逻辑中，无需硬件工程师介入。SDSoC 中的编译器将生成整个 Vivado 项目，并为目标硬件平台生成可引导的软件映像。

Nick 指出：“利用针对我们 Zynq SoC 的 SDSoC 环境，我们主要是让我们的嵌入式软件工程师变成系统工程师。”（见图 3，SDSoC 开发环境演示）

SDSoC 环境利用带有用户指定的 C/C++ 代码的宏编译器，可在 Zynq SoC 逻辑中实现加速。通过在底层运行 Vivado 设计套件，该环境可自动将

该代码返回 IP 模块并将该模块配置为器件的逻辑，从而自动生成软件驱动程序。

SDSoC 可为基于 Zynq All Programmable SoC 的开发板（包括 ZC702 和 ZC706），以及第三方和特定市场平台（诸如 ZedBoard、MicroZed、ZYBO 以及视频与图像开发平台）提供板支持包 (BSP)。

Nick 表示：“我们今后几个月将为 SDSoC 添加更多 BSP，尤其是随着越来越多第三方平台公司用 Zynq SoC 开发系统。SDSoC 有助于赛灵思以及采用 Zynq SoC 开发平台的公司扩大用户群。”

Nick 同时指出，SDSoC 环境的主要目的是让大量嵌入式软件设计人员能用赛灵思 Zynq SoC 创建完整的系统，不过有传统 FPGA 背景的用户和设计团队也能从该环境受益匪浅。

Nick 指出：“它能帮助设计团队快速用 C 和 C++ 语言设计系统架构，然后尝试不同的配置，以获得所需的最佳性能。如果他们团队里确实有 FPGA 设计人员，则能让其通过 Vivado 工具来进一步优化模块和逻辑布局。”

如欲了解联盟成员对 SDSoC 环境支持的更多信息，敬请参阅本期 Xpedit 部分（第 66 页）。关于赛灵思 All Programmable Abstraction 的更多信息，敬请访问：<http://china.xilinx.com/products/design-tools/all-programmable-abstractions.html>。

广播领域的出色表现

实现用于专业视频的 JPEG 2000网络

作者: Jean-Marie Cloquet

Barco Silex公司图像处理业务部经理

jean-marie.cloquet@barco.com



赛灵思携手 Barco Silex 推出的最新参考设计支持通过互联网协议网络传输 JPEG 2000 视频。

由于画质出众，JPEG 2000 已成为高画质视频压缩（包括电视广播商节目收集网络视频传输）的标准选择。因此视频设备供应商已经开始在各类传输解决方案中添加 JPEG 2000 编/解码器，以支持各种接口和在某些情况下使用专有协议。

但这个趋势意味着视频服务提供商只有一个或者几个厂商的产品可供选择。随着 2013 年 4 月视频服务论坛（VSF）的 TR-01 互联网协议网络视频传输建议书的出台为设备互操作性提供规范，这一困局才有了解决方案。赛灵思及其联盟计划认证成员 Barco Silex 迅速携手支持这一互操作性规范。

Barco Silex 已经完成 VSF TR-01 建议书的参考设计。这种多通道 JP2000 IP 视频传输解决方案最近已经公布在赛灵思网站上。它以赛灵思和 Barco Silex 提供的 IP 核为基础，方便广播设备 OEM 厂商定制和集成。鉴于 Barco Silex 的贡献，美国国家电视艺术科学院（National Academy of Television Arts and Sciences）为 Barco Silex 颁发了 2014 年技术与工程艾美奖（见图 1）。

寻求一流的视频压缩

JPEG 2000 取代了旧的 JPEG 标准，与其前身或 MPEG 等其他常见格式相比，能提供众多优势。2004 年 JPEG2000 通过美国好莱坞数字影院倡导组织 (DCI) 规范要求，成为数字影院图像压缩的事实标准。由于具备画质无损压缩能力，JPEG2000 成为安保、存档和医疗应用的理想选择。

广播行业也对此表示关注。广播和视频服务企业拥有大量直播视频，需要通过他们所谓的节目收集网络 (contribution network) 中传输至后期制作和流媒体设施 (图 2)，要求无延迟、画质无损。因此专业视频行业对画质无损压缩尤为感兴趣。即压缩方案需既能保持画质，又能实现高效存储和传输。

此外 JPEG2000 的其他创新还意味着广播行业的飞跃式发展。与 MPEG 把各帧压缩为帧组不同，JPEG2000 视频流中的每一帧都单独压缩为静止帧。这种单帧压缩技术既能实现低时延，还能便于逐帧进行后处理和编辑。JPEG2000 视频流还能部分解压和观看，从而允许用同一视频流开展不同应用并提供不同观看体验。

另一大优势是视频流的传输容差能力强。与其它编解码器相比，纵使无法用前向纠错 (FEC) 纠正传输误差，这些误差在解码后对画质的影响也非常小。最后 JPEG2000 在多次编/解码流程后还能保留原有画质，这对存在多个视频管理阶段的节目收集网络而言，意义十分重要。

鉴于这种需求，设备供应商迅速启动自己视频设备上的 JPEG2000 编/解码器的实现工作。不过异地之间的传输，他们仍然有多种实现方案可

供选择，诸如专有协议等。对视频服务提供商而言，不利之处在于他们必须在一家或几家厂商的产品中做出选择，而非构建匹配度最好、性价比最高的基础设施。

视频传输标准化建议

因此服务提供商提出了明确的传输标准化需求，以确保现有和未来设备之间实现更理想的互操作性。他们所需要的，是一种能够通过 IP 网络理想组织的传输技术，因为 IP 网络正在成为主流网络架构，能提供随时可供高吞吐量数据传输使用的标准化设备。自 2007 年起，美国电影电视工程师协会 (SMPTE) 就公布了 IP 视频传输的标准，并自那时起一直不断丰富完善。SMPTE 2022 就包括针对 MPEG-2 传输流中恒定比特率视频信号的 IP 协议 (用于压缩视频的 SMPTE 2022 1&2；以及用于未压缩视频的 SMPTE 2022 5&6)。

以这些标准为基础，视频服务论坛于 2013 年发布了 VSF TR-01 文档，即《基于 IP 的 MPEG-2 TS 的 JPEG 2000 广播配置文件视频传输》的建议书。VSF 是一家由服务提供商、用户和制造商组成的国际协会，致力于发展视频网络技术的互操作性、质量指标和培训。

任何符合 VSF TR-01 的设备都会从 SDI (串行数字接口) 获取输入信号，而 SDI 是广播行业无压缩点对点视频传输的既有标准。该设备将提取有效视频、音频和辅助数据 (例如字幕)，然后把视频压缩成 JPEG 2000 格式。得到的视频流与音频数据和辅助数据经多路复用为 MPEG-2 传输流。该传输流随即按 SMPTE 2022 封装为实时传输协议 (RTP) 流，并通

过 IP 发送给接收设备。接收器将对 RTP/IP 流进行解封装操作，解复用 MPEG-2 传输流，解码 JPEG 2000，然后将视频、音频和辅助数据以 SDI 信号方式输出。

实现基于FPGA的参考解决方案

2012 年 9 月，就在 VSF 建议书发布之前，赛灵思和 Barco Silex 宣布结成合作伙伴，共同开发 IP 视频传输解决方案，旨在提供包含经硬件验证的 IP 核、参考设计和系统集成服务在内的综合性平台。在合作中，Barco Silex 肩负系统集成商的作用，将赛灵思提供的内核（SMPTE 2022、SMPTE SDI、以太网 MAC）与自己的高性能 JPEG 2000 和 DDR3 存储器控制器内核完美匹配在一起。其目标是支持广播设备 OEM 厂商加速产品开发，把最先进的 IP 视频传输功能添加到现有以及正在开发的产品中。

在此框架内，合作伙伴目前已经完成由四通道发送器 - 接收器平台组成的参考设计（图 3）。发送器能接收四路 SDI 高清（HD）流（1080p30），有选择地用 JPEG 2000 压缩，然后按 VSF TR-01 标准通过 1Gbps（经压缩）

或 10Gbps（未压缩）以太网传输。接收器平台则负责接收 IP 流、并对其进行解封装和解压，然后将信号发给四路 SDI 高清链路。

在发送器平台上，赛灵思 SMPTE SDI 内核接收输入的 SDI 视频流。在无压缩路径上，这些 SDI 流先由赛灵思 SMPTE 2022-5/6 IP 视频传输发送器内核进行多路复用并将其封装为固定大小的数据报，然后通过赛灵思 10Gb 以太网 MAC（10GEMAC）和 10G PCS/PMA 内核发送出去。

在压缩路径上，SDI 流首先进入 JPEG 2000 编码器完成压缩。接着它们按照 VSF TR-01，由 BarcoSilex 设计的专用 TS 引擎内核封装为 MPEG-2 传输流。最后 SMPTE 2022-1/2 IP 视频传输发送器内核将这些流打包为固定大小的数据报，并通过 1G TEMAC 发送出去。此外，压缩后的流也可以使用 10GEMAC 和 10G PCS/PMA 内核在 10Gb 链路上与未压缩的视频进行多路复用。

在接收器平台上，先在 10GEMAC 上收集未压缩流的以太网数据报，然后 SMPTE 2022-5/6 IP 视频传输接收器内核过滤数据报、

解封装，解复用为单独的流，通过 SMPTE SDI 内核输出 SDI 视频。在 10GEMAC 上收集经压缩的视频流的以太网数据报，然后由 SMPTE 2022-1/2 IP 视频传输接收器内核和 TS 引擎解封装，并馈送至 JPEG 2000 解码器。解码器输出的视频则转换为 SDI，并发送给 SMPTE SDI 内核。

对四个通道的每一个通道而言，都可以独立于其他通道选择是作为压缩路径还是未压缩路径。

为互操作性解决方案奠定基础

两家公司把参考设计实现在两个不同平台上，一个使用 Zynq®-7000 All Programmable SoC，另一个使用 Kintex®-7 FPGA。不过使用的模块可集成到各种解决方案中，以满足从低成本大批量应用到最严苛的高性能应用的全系列 OEM 系统要求。使用的 IP 核（如赛灵思 SMPTE 2022）和以太网 MAC LogiCORE™ 模块，可用于直至 UltraScale™ 水平的全系列赛灵思 FPGA 系统。

在编 / 解码方面，参考设计采用了 Barco Silex 的 JPEG 2000 编码器 / 解码器 IP 核。这些 IP 属于经硬件验证被广泛采用的单芯片 FPGA 解决方案，可供高性能同步多通道 720p30/60、1080i、1080p30/60 和 2K/4K/8K JPEG 2000 编码 / 解码使用。这些内核也支持目前市场上现有的最丰富多样的 JPEG 2000 选项。在把多路视频流拼接为平滑高数据速率系统中发挥关键作用的是来自 Barco Silex 的 DDR 存储器控制器。这款高度可定制的控制器的优化可实现高带宽，能够把访问重新排序，混合之后发送给 SDRAM 的不同 Bank。

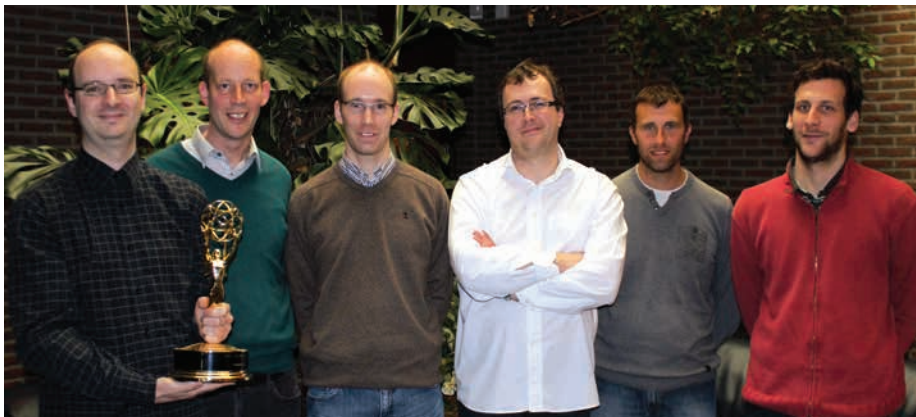


图 1 - 负责参考设计的 Barco Silex 视频团队因实现 JPEG 2000 互操作性的标准化和量化荣誉 2014 年技术与工程艾美奖。从左至右分别是：Luc Ploumhans、Sake Buwalda、François Marsin、Jean-François Marbehant、Jean-Marie Cloquet 和 Vincent Cousin。

使用更大的视频显示器。以四 SDI 模式使用参考设计的四个输入通道（通过四条 SDI 线缆传输 4K），就可以输入 4K 信号并通过 IP 网络传输 4K 信号。这样参考设计就能够用于高达 4K 的视频分辨率。

FPGA推动视频行业发展

赛灵思和视频专业厂商 Barco Silex 的合作的目的是在专业视频市场中充分发挥基于 FPGA 的平台的功能和灵活性。通过将 Barco 的 JPEG 2000 内核与赛灵思的传输内核结合使用，OEM 厂商可以迅速生产和更新标准化广播设备，让他们的产品在工艺方面满足未来需求。

视频行业的 IP 网络应用方兴未艾，参考设计的推出恰逢其时。OEM 厂商能否在新兴市场分一杯羹取决于其产品上市速度。采用基于赛灵思 FPGA 的可重编程解决方案，他们就能够标准尚在发展演化中把产品推向市场。

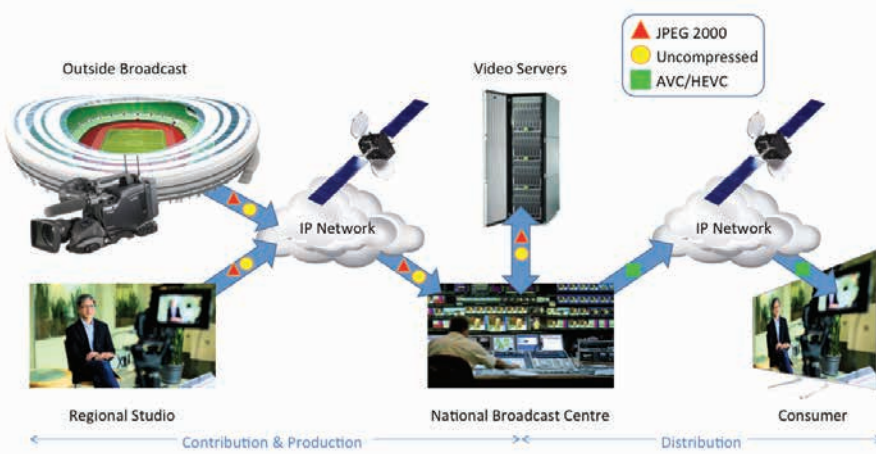


图2 - 广播企业的节目收集网络

两公司在 2014 年 2 月在弗吉尼亚州阿林顿举办的年度 VidTrans 大会上以公开互操作性演示的形式展示了该参考设计的第一代。在 VSF 组织的这次测试中，10 家公司（Artel、Barco Silex、Ericsson、Evertz、Imagine Communications、IntoPIX、Media Links、Macnica、Nevion 和赛灵思）提供的技术和设备连在一起，

演示使用 JPEG 2000 编码器和解码器实时压缩 720p30 和 1080i60 高清内容并实时传输。

数月之后，Barco Silex 展示该参考设计已具备处理 4K 和超高清（UHD）信号的能力。作为建议用作新一代视频分配的主要标准之一，4K 视频能承载的像素数量是 1080p 视频的四倍，能进一步提升图像分辨率和

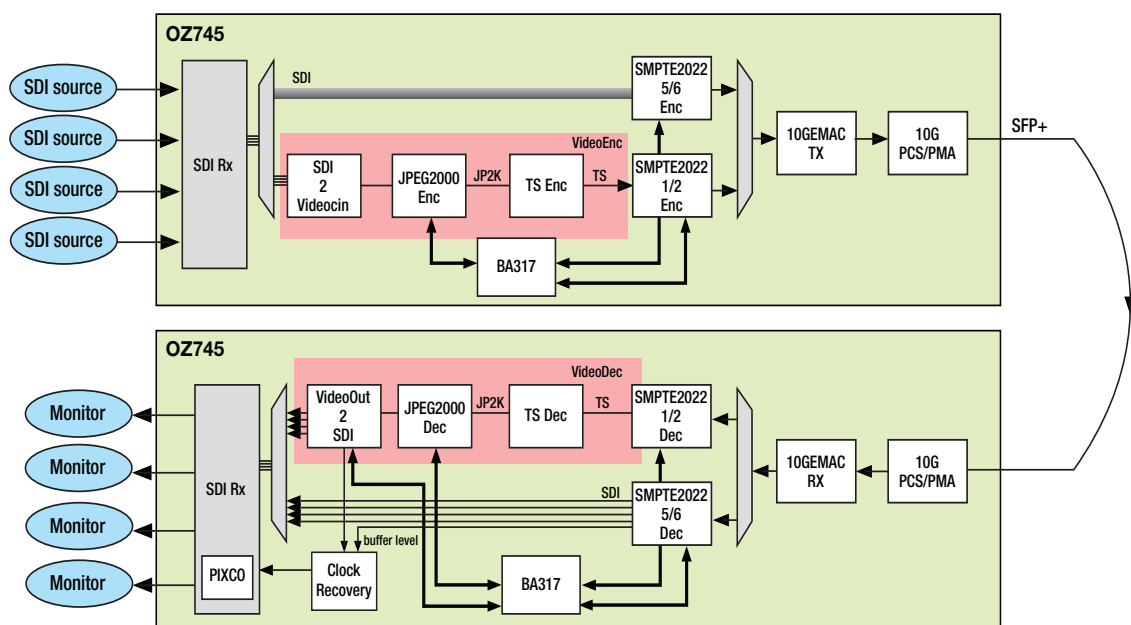


图3 - 由发送器-接收器平台组成的参考设计的原理图

“大白兔”： 当每纳秒都变得重要

作者：

Javier Díaz博士，

Seven Solutions SL首席执行官
javier@sevensols.com

Rafael Rodríguez-Gómez，

Seven Solutions SL首席技术官
rrodriguez@sevensols.com

Eduardo Ros，

Seven Solutions SL首席运营官
eduardo@sevensols.com

在围绕赛灵思 FPGA 及 SoC 器件构建的产品中，一种基于以太网的高精度时序解决方案正在进入市场。



电信和信息科学的最新发展正在推动对工业时间传输的要求几近科研应用的水平。例如，即将到来的 100G 以太网网络及 5G 移动通信要求时序精度在几纳秒范围内，而用于配电网的智能电网则要求亚微秒精度。高频率交易的时间戳（一般指股票交易）需要可靠的机制将时间从认证机构分配给业务中心。最后，GPS 或伽利略等采用 GNSS 技术的定位服务都可通过高精度同步机制获得优势。

一种诞生于欧洲核研究机构 CERN、名为“大白兔”的以太网技术旨在满足这些应用及其他应用的高精度时间需求。以《爱丽丝仙境奇遇记》中着迷于时间的野兔命名的“大白兔”技术基于并兼容于 PTPv2 (IEEE-1588v2) 和同步以太网等标准机制，但经过适当修改，可实现亚纳秒精度。“大白兔”不仅天生可通过远距离链路执行自校准，而且还能够以极小的退化将时间分配给极大数量的设备。

我们的分支机构 Seven Solutions SL 自 2009 年“大白兔”技术创世以来就一直在开发该技术，其始终致力于使用赛灵思 All Programmable 解决方案将大白兔产品带入市场。我们的最新产品是 ZEN (Zynq® 嵌入式节点) 电路板，这是一款旨在保持高精度参

由于价格原因，基于芯片级原子钟等高精度时钟的解决方案对于大规模采用而言成本太高。

考时钟的时序电路板，不仅可为其它节点提供时序信息，同时还可在“大白兔”网络框架中进行自我同步。

时间简史

物理学家一直都理解时间的重要性，多年来已经发明了各种测量时间的方法。从简单的天空扫描技术（日规、测星仪）到依靠亚原子世界（原子钟）属性的复杂机制，科学家一直都在为开发精确时钟而孜孜不倦地工作。现有时钟在大约3亿年内不会有1秒的误差，这种精度在许多应用中至关重要，例如维持国家度量衡实验室时标。

不过，这些极为精确的时钟非常昂贵、脆弱，并且会占据大量的物理空间。因此它们并不适合许多现实场景。实际上大多数应用通常依赖电子产品，其包含低成本时钟（晶体振荡器）。只需花上几美元，我们就可在一大批规格各异的振荡器中做出选择。

对于简单的应用而言，振荡器的精确度已经足够了。但在许多其它需要同步通信或全局时间概念来同步工作的应用领域（分布式仪器）中，这些彼此互不相连的“自由运行时钟”就无法使用了。虽然设计人员可通过安装较好的振荡器来局部解决该问题，但在技术上这样做并非总是可行。单独的时钟仍然没有同步，即便小小的频

率偏差，也会让这种方法失效。

然而由于价格原因，基于高精度时钟（如芯片级原子钟或CSAC）的解决方案对于大规模采用而言成本太高。在这些情况下，替代方法是将时钟信息从基准时钟（高稳定、一般价格昂贵）分配给网络中所有其它需要准确同步的组份。问题是我们如何才能做到？

时间传输技术

分配时间可选择多种方法。请注意分配频率（涉及通过导线发送振荡器信号）与分配相位（当事件在网络的所有组份中以极为一致的瞬间触发时）不同。

例如，我们可通过使用同轴电缆或光纤传输时钟振荡来解决第一个问题（频率分配）。在第二个场景（相位分配）中，我们不仅可在导线上编码一个脉冲，进行每秒传输，而且还可将这个脉冲作为参考，知道新的每秒开始的时间。这种技术一般被称为每秒脉冲（PPS）信号。

另外，还存在第三个问题。我们可能还需要提供时间，这不仅要让各项工作按相同时间运行，或是提供有关何时开始计数（相位）的相同参考，而且还要确保我们在所有设备中具有相同的时间。因此，可通过从中央时间服务器传播时间信息，然后测量该消息的传播时间并在每个节点上

对其进行标注的方法来分配时间值。有了频率、相位（PPS）和时间这三个元素，我们就可以说网络是同步的。

目前的工业解决方案以不同的方法提供这些属性。例如，GPS设备通过提供参考频率（10至50MHz）、PPS信号和串行码来提供时间（一般基于NMEA协议）。这种方法广泛用于大量需要准确同步的系统中，因为不同仪器可以便捷连接至不同的GPS接收器。但它使用大量低级信号。在电网应用中，这些值通过名为IRIG-B的简单协议提供，其可提供时间和PPS信息。过去，IRIG-B方法一直能充分满足同步电网的需求。然而今天它不能处理“智能电网”，因为这种电网正在变得空前复杂，而且还包含需要更高精度的新型能源监控应用。

随着数据包网络的基本普及，交换网络以往使用的机制已经经过发展演化，适应了数据包网络。在基于精确时间协议（PTPv2或IEEE-1588v2）加同步以太网（SynCE）的解决方案中，SDH/SONET技术也在逐渐转变。PTPv2是网络时间协议（NTP）的工业演化版，该协议被互联网用于同步整个网络中的计算机。PTPv2依靠硬件时间戳机制显著改善时间同步的精度。

第二种机制SynCE可实现在数据载体上编码时钟信号。采用这种对

用户透明的方法，我们可将时钟信息和频率分配给所有设备。将 PTPv2 与 SyncE 配对，有助于我们针对电信无缝使用数据包网络。这种组合是当前电信、电网和自动化应用中最常见的解决方案。请注意，一些与相位传播及系统可扩展性有关的关键问题仍然十分重要，而且尚待解决。

科学应用及更高层次应用

许多应用都需要把参考时钟源信息传播到不同目的地。科研机构可能是需要高精度时间分配最为严格的基

础设施。从 CERN 的 LHC 加速器到 CTA、SKA 或 KM3NeT 等大型射电天文分布式设施，所有这些都需要超高精度的时间与频率分配。

但新一代 IT 及通信应用还需要使用目前标准方法无法实现的极高精度时间传输。例如在 GPS 应用领域，测量卫星信号传播时间等同于测量距离，因此定位与时间精度测量密切相关。一般来说，GNSS 容易受阻塞或电子欺骗问题的影响。所以在用于时间分配时，建议重要基础设施将地面替代方案（基于光纤）用作互补冗余

机制。

“大白兔”解决方案

“大白兔” (<http://www.whiterabbitsolution.com/>) 是精确时序以太网的扩展和外延。它是 CERN 于 2009 年构思的一个开放式协作软硬件项目，技术行业已急切开始参与了其发展。源代码提供在开放式硬件资源库 (OHWR, <http://www.ohwr.org>) 中，以鼓励不同企业及研究机构进行开发。

从一开始起，位于西班牙格拉纳达的 Seven Solutions (www.sevensols.com)

大白兔应用

电信

Advanced-LTE 等新一代电信应用需要高精度时序来正确处理通过多个天线/基站传送的高频视频/音频流。其他相关应用包括雷达和信号智能。

控制与获取

工业同步应用成为大规模获取与控制机制的关键。

智能电网

电网监测是一项必要工作。由于大白兔精度很高，能够帮助预测可能破坏系统的高峰值。

高能物理

大白兔提供一种分布式模块化技术，使用自动校准功能以低维护成本准确地同步各种设备。它是用户研究中心内时钟分配的一种简便方法。

高频率交易

大白兔为低延迟交易和高速决策提供高效率解决方案，能在全球范围内同步采集市场估值。

地理定位

大白兔是支持或改善机场中现有 GPS 信号及雷达设备的理想技术。

大白兔的特性

- ✓ 低于纳秒级传输时间
- ✓ 相位/频率分配
- ✓ 兼容 (PTPv2、SyncE)
- ✓ 动态校准
- ✓ 模块化、可扩展
- ✓ 有确定性



图1 — 大白兔应用概要

就一直是在协作设计各种大白兔产品，其中包括电子产品、固件以及门控件。此外，该公司还可提供基于该技术的定制解决方案和交钥匙解决方案。

作为以太网的扩展，”大白兔”技术正在接受评估，以加入高精度配置文件框架中的新一代高精度时间协议标准（IEEE-1588v3）中。标准化将有助于“大白兔”集成至未来各种不同技术中，如图 1 所示。

“大白兔”技术的深入介绍

“大白兔”整合大量机制，可优化其位于以太网扩展框架内的时序精度，因此可保留以太网通信结构。此外，“大白兔”还集成 PTP、同步以太网以及数字双路混合器时间差异（DMTD）相位跟踪技术。

Seven Solutions 提供的全新 ZEN 电路板将展示“大白兔”的主

要组份如何在产品中结合（图 2）。ZEN 电路板以赛灵思 Zynq-7000 All Programmable SoC 为基础，内含“大白兔”内核以及可提供高精度时钟的千兆位以太网 MAC 实现方案。在“大白兔”内核中实现的同步机制包括以下组份：

- 频率同步（同步）：这可通过使用 SynCE 实现，其可编码数据载体中的时钟信号。为确保所有节点都使用相同的频率，我们采用一个基于本地振荡器的机制，该振荡器可通过从光链路恢复的外部时钟进行控制。
- 相位同步：节点的物理时钟可转发至主设备组份，或从主设备组份转发至节点，以便主设备能够将该信号（来自从设备）的相位与自己的相位相比较。偏差应该

与信号通过光纤的传播时间相同（使用 PTP 进行正确测量）。有了该信息，主设备就可确定其自己的时钟与来自从设备的时钟之间的相位差，并要求从设备将其相位转换成与主设备完全相同的值。该过程可通过在 FPGA 门控件中实现一个数字 DMTD 来数字化完成。

- 时间同步：这是使用 PTPv2 协议的结果，该协议不仅可测量链路传播时间，而且还可提供全局时间概念。此外，“大白兔”还考虑了因在双向光纤中为每次通信传输（在环路中向前和向后）使用不同波长而产生的有关传播时间的不对称性，从而可提高标准 PTP 协议的精度。由于频率和相位之前已经同步，因此我们可以在“大白兔”网络中确保所有设备的全局时间概念。

所有这些工作均可在“大白兔”内核中实现，一部分使用适当的 FPGA 门控件，一部分使用嵌入式软件内核。“大白兔”产品包括执行这些不同时钟工作所需的适当振荡器、PLL 和时序电子产品。

作为案例研究，下面我们将更详细了解 ZEN 电路板。该电路板采用双路“大白兔”内核（D-WRC），这是 Seven Solutions 用我们最新赛灵思 7 系列产品开发的原始“大白兔”内核的修订版。D-WRC 能够同步两个“大白兔”节点，也可作为菊花链网络中的中间链路。此外，ZEN 电路板还包含由 D-WRC 控制的高精度、低抖动、温度补偿时钟资源。

此外，Zynq SoC 在 Linux 操作系统下运行的双核 ARM® Cortex™-A9 处理器可促进用户应用的开发。板载

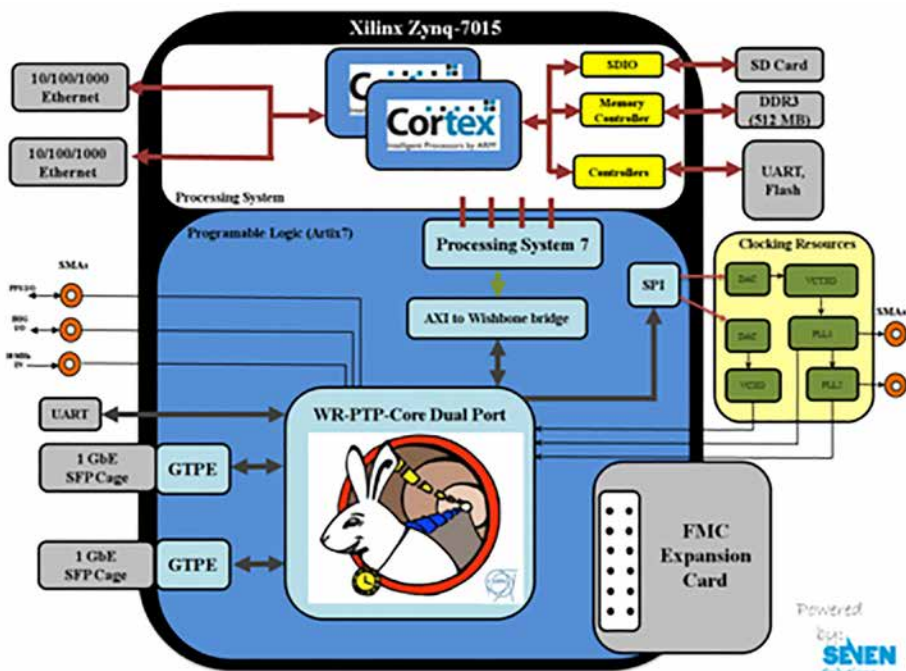


图2 — 基于Xilinx Zynq SoC器件的“大白兔”门控组份

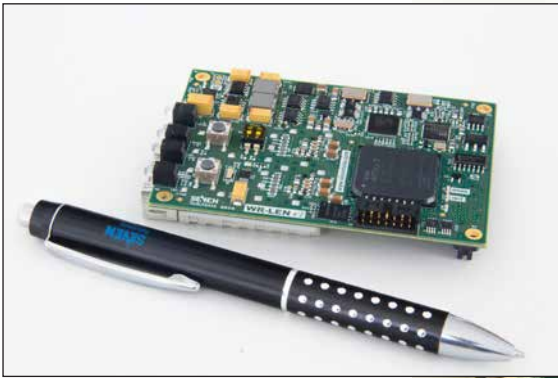


图3 — “大白兔” LEN电路板（上图）和ZEN（右图）电路板。它们分别由 Artix FPGA和Zynq SoC器件提出支持，采用支持工业应用的适当外壳。



提供 Linux，有助于使用令人关注的全新特性，例如适用于配置的 Web 服务、状态监控的 SNMP 支持以及远程固件加载与更新。

ZEN 电路板旨在用作提供高精度时间，它可提供多样化连接与扩展带来的大量可能性：

- RIG-B I/O 是 ZEN 电路板使用的当日时间，它既可作为主设备，也可作为从设备。
- 两个连接至 ARM 处理器的 10/100/1000 以太网端口可用于各类网络应用和协议（NTP、sNTP、PTPv2 以及管理等）。
- 提供两个 SFP 模块，用于插接符合“大白兔”规范的链路。
- SMA 连接器可让 ZEN 电路板与更精确的时钟同步（例如 GPS 源或高稳定振荡器），提供由“大白兔”同步的各类时钟。
- FMC 连接器可插入一款采用“大白兔”项目框架开发的夹层板，

或是市场现有的任何其它工业电路板。这些 FMC 卡不仅可增强 ZEN 的发展潜能，而且还可实现大量的产品配置。

- 存储器资源包括 SD、DDR3 和闪存。
- 包含两个 UART-USB 连接器，可在 D-WRC 和 Cortex 处理器中进行管理和调试。

简言之，ZEN 电路板可为最终用户提供一节点，其不仅可实现亚纳秒同步并以菊花链方式工作，同时还可尽其所能地提供最佳的 Zynq SoC 与全新水平的系统设计功能。

“大白兔”设备

“大白兔”技术肇始于 CERN 提倡的开放式硬件社区（开放式硬件库，OHWR）。为加速该技术的学习进程，Seven Solutions 开发出了一款由两片名为 SPEC 的 Spartan®-6 电路板组成的“大白兔”入门套件。其中一个可配置为主设备，另一个则可配置

为从设备。目的是鼓励用户执行几个早期评估实验。

这项技术最复杂的组份是交换机。我们通过与 CERN、GSI 以及其他合作伙伴合作，共同开发了 18 端口“大白兔”交换机，从而设计了采用 MicroTC 尺寸规格的主板。核心组份是一款 Virtex®-6 (LX240T) FPGA。我们将该器件与一款运行嵌入式 Linux 操作系统的外部处理器（ARM926E）进行配对，执行了系统更新和文件管理等各项高级工作。该交换机使用了 18 个针对 SFP 的 GTX 链路以及 40 个面向通用任务（LEDS、SFP 检测等）的 GPIO。这是一种非常复杂的产品，既能分配时间，也能在使用标准电信工具的同时处理数据包。

近期，Seven Solutions 将“大白兔”内核移植到了 LEN 电路板中的赛灵思 Artix® FPGA 系列中（图 3），实现了比现有 OHWR 器件成本更低、能效更高的解决方案。此外，我们刚刚开发出了一款基于 Zynq SoC 器件

的“大白兔”产品。该 WR-ZEN 节点（之前的名称）代表一个完整的通用片上系统方法，其中节点和计算机都整合在同一电路板中。该解决方案可在降低成本，改善系统灵活性的同时，实现更便捷的维护。

Seven Solutions 目前开发的工业产品可为管理、配置和监控提供标准接口，不仅能充分利用“大白兔”技术的各项优势，同时还可提供更强大的特性、支持和文档。

“大白兔”应用

“大白兔”技术的第一个目标是科学应用。近期该技术已经在一些设备及研究项目中整合，用于高能物理及分布式射电天文设施的框架中。“大白兔”技术已经在多个粒子加速器（CERN、

GSI 以及其他机构）中使用，而且 KM3NeT 与 HISCORE 等国际科研计划也正在考虑使用该技术。因此，“大白兔”方法已在各种要求严格的应用中得到验证，这类应用要求通过大型设施上的分布式仪器实现精确的时序与频率转移。

2014 年，“大白兔”还由 [荷兰 VSL 进行了 125km 的远距离链路测试](#)，并由 [芬兰 MIKES 进行了 1,000km 的远距离链路测试](#)。

在科研领域之外的广泛应用中也需要精确时序。智能电网要求准确可靠的时序，而高频交易也离不开 [经过认证的精确时序](#)。

许多此类应用领域目前依赖的是 GPS 时序信号，这种信号生来就容易受到影响（由于环境条件或者由于意

外或恶意阻塞及电子欺骗）。GPS 不应该用于安全关键性基础设施（2010 年 1 月 20 日《美国空军首脑对过度依赖 GPS 的警告》，摘自《Inside GNSS》杂志新闻）。在这一点上，“大白兔”代表着一种替代性解决方案，其可实现通过地面光纤进行高精度时序及频率传输。可通过部署标准电信网络来降低这种方法的成本。

除“大白兔”的功能特性外，Seven Solutions 还正在面向要求高可用性的重要应用开发全新工业产品。冗余电源、热插拔风扇、延期振荡器以及其它技术允许其部署于不能承受系统故障或长时间维修过程的设施。

图 4 是如何将 WR-LEN 用作分布式机制，以简单低成本的方式提供时序信息。该系统能够以类似 GPS 的

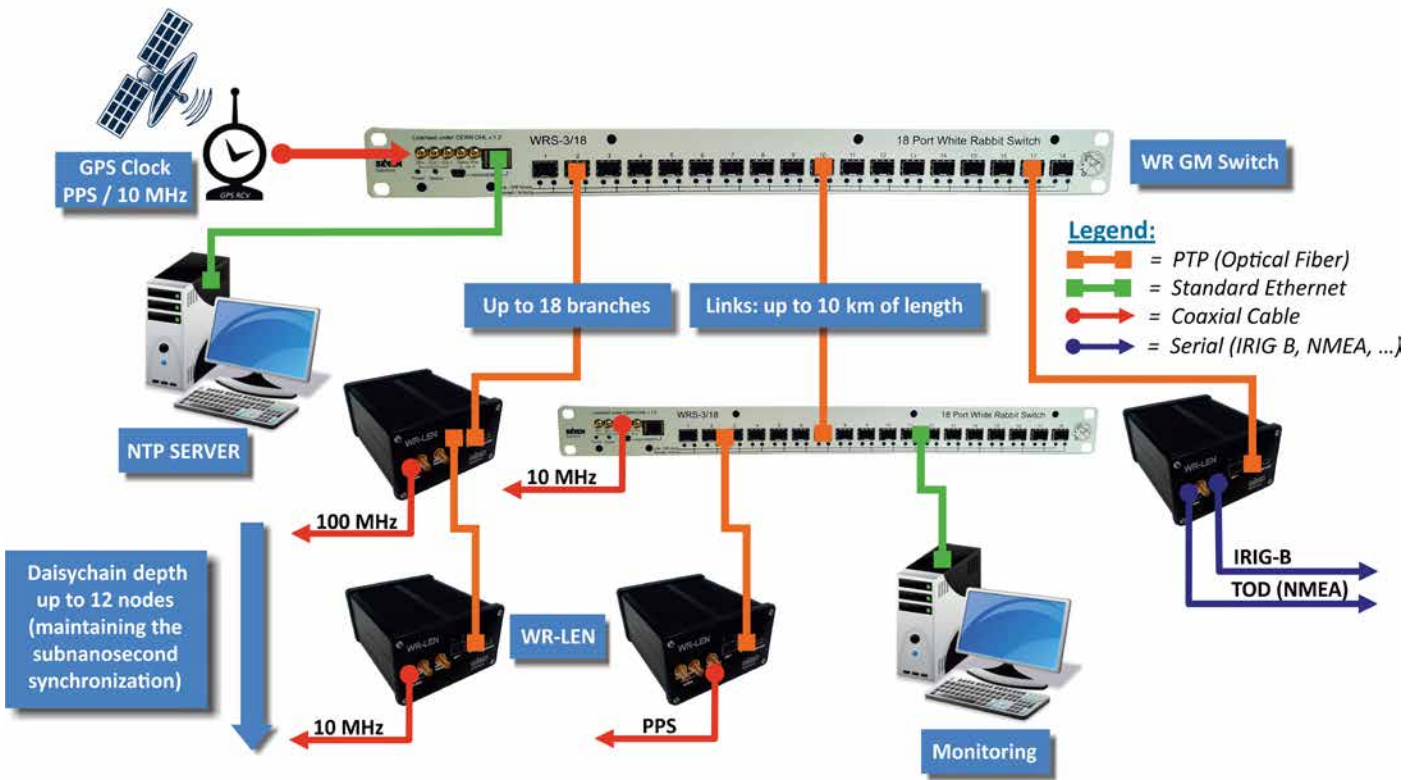


图4 — 为不同节点提供精确时序的“大白兔”网络。菊花链配置允许通过WR-LEN节点。

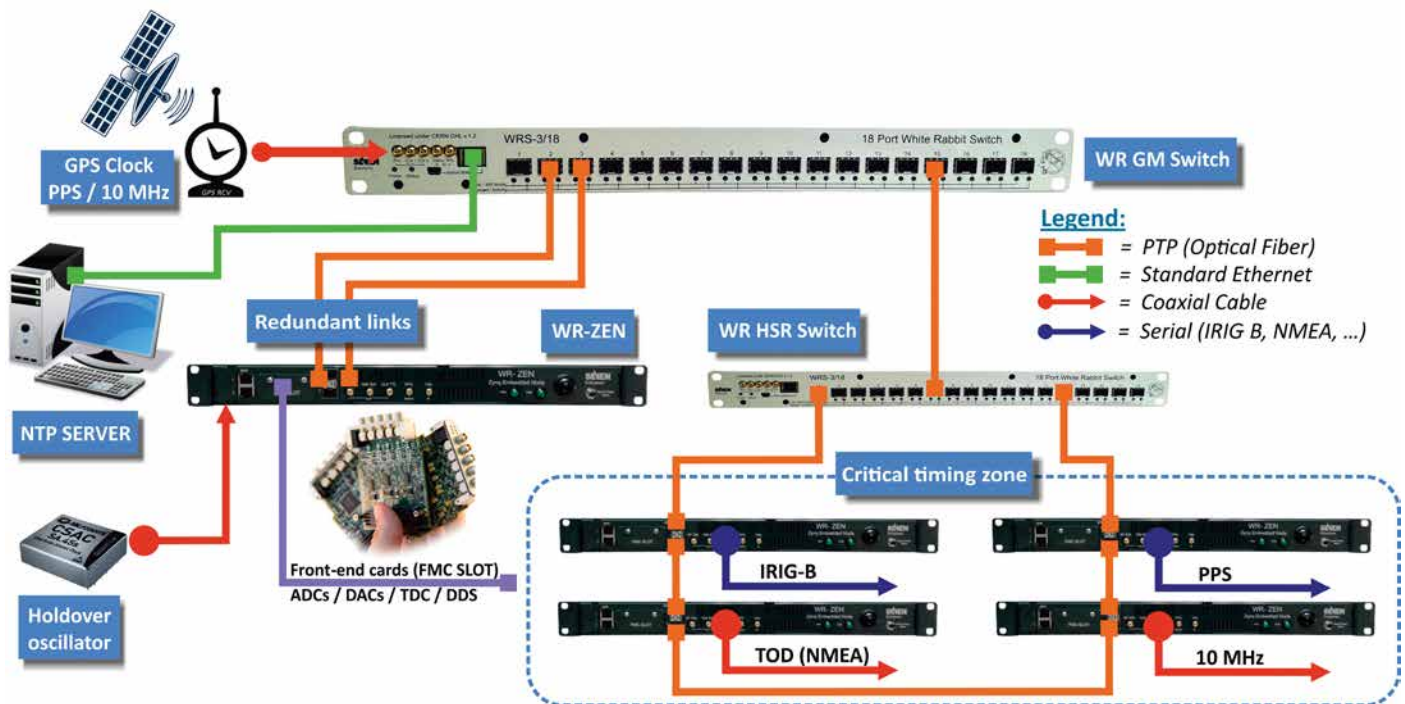


图5 基于ZEN时间提供技术、面向安全关键性系统的网络配置

方式分配时序。它使用适用于电网应用的 IRIG-B 输出格式提供时序。此外，PTPv2 也是一种可用的接口，是一种有效选项，因为它能够与现代电网设施上使用的 PTP 网络集成。

图 5 是基于 ZEN 时间提供技术的网络配置。该实例中显示了更多功能强大的特性。关键组份上的冗余电源、冗余网络拓扑及延期 CSAC 振荡器都可实现。此外，FMC 扩展端口还可实现添加夹层卡，进一步开发更多传感与控制应用。

未来发展

“大白兔”是一项前景光明的技术，能够解决智能电网、电信以及高频率交易等各类终端应用领域的同步需求问题。“大白兔”不仅能解决相位同步等问题，同时还能以亚纳秒精度为分布

在遥远距离（数百公里）的数以万计的设备同步时钟。因此，“大白兔”还能实现超高精度时间传输以及无损的全面数据传输。

这些特性加上“大白兔”的可扩展性，将有助于开发整个世界范围的地面同步机制，其不仅可用作 GPS 解决方案（基于地面卫星接收站天线）的后台技术，而且还能开启自动驾驶汽车或室内导航等全新应用的大门。即将来临的 100G 电信网络将通过高精度服务质量评估机制获得巨大优势，而 5G 无线技术则可使用“大白兔”解决众所周知的相位同步问题。

这些只是“大白兔”可能会在其中产生巨大影响的未来应用的部分示例。此外，由于“大白兔”可能会在 IEEE-1588v3 配置文件（目前正在考虑）中得到标准化，因此它将非常容

易适应多家厂商。而且我们认为最具挑战性的应用还未出现。

Seven Solutions 正在提供第一批基于“大白兔”技术的工业产品。我们的交钥匙解决方案不仅可便捷集成在基于标准电信接口的用户应用中，而且只需使用 Web 服务或 SNMP 等标准软件便可完成配置 / 读取。下一步我们将针对分配 RF 生成（无需发送参考频率）或触发事件采集整合各种特性，提供大量电信应用所需的高精度。

RF-DAC多频带 发射器线性评估

贝尔实验室的研究人员展示了如何利用赛灵思 FPGA、IP 核和 MATLAB 为 RF DAC 的快速评估创建一个灵活平台。

Lei Guan

技术人员

Alcatel Lucent Ireland公司贝尔实验室
lei.guan@alcatel-lucent.com

无线通信行业已经进入了一个全新的一体化时代；每个网络运营商都在寻求更紧凑、多频带基础架构解决方案。新兴射频类数据转换器——RF DAC和RF ADC——在架构上使创建紧凑的多频带收发器成为可能。但这些新兴器件固有的非线性将成为这一发展趋势的绊脚石。

例如，频域中射频器件的非线性包括带内和带外两种情况。带内非线性是指TX频带内不需要的频率成分（frequency term），而带外非线性则是指TX频带外不需要的频率成分。

对于正在使用RF DAC对多频带发射器进行原型设计的系统工程师而言，确保关键组件符合标准线性要求是非常重要的。因此，在早期原型设计阶段，从根本上需要一个灵活的测试平台，以正确评估AR DAC在多频带应用中的非线性性能。

在爱尔兰贝尔实验室，我们已经创建了一个灵活的软硬件平台，可用于快速评估下一代无线系统潜在备

选设备RF DAC。这个研发项目的三个关键因素分别是：赛灵思高性能FPGA、赛灵思IP和MATLAB®。

在开始这段工程设计传奇故事之前，我们还要强调几点。在设计中，我们试图尽量减少FPGA资源的占用，同时尽可能保持系统灵活，所以我们只需要集中精力实现必要的功能。为建立完整的测试系统，我们选用ADI公司的最新RF-DAC评估板（AD9129和AD9739a）和赛灵思ML605评估板。ML605评估板配套提供Virtex®-6 XC6VLX240T-1FFG1156 FPGA器件，其包括快速切换I/O（频率高达710 MHz）和SERDES单元（频率高达5 Gbps），用于连接RF DAC。

现在，让我们仔细看看如何使用赛灵思FPGA、IP和MATLAB创建这个简单而又功能强大的测试平台。

系统级要求与设计

该评估平台的主要目的是通过各种用户自定义的测试数据序列来激励RF

DAC。为此，我们设计了两个测试策略：连续波（CW）信号测试（xDDS）和宽频带信号测试（xRAM）。

多频连续波（CW）测试一直是RF工程师对RF元件非线性进行特性描述的首选。遵循相同的测试理念，我们创建了一个基于直接数字综合器（DDS）的可调四音逻辑内核，实际上是采用一对双音信号在两个独立频带上激励RF DAC。通过独立调谐四音，我们可以评估RF DAC的线性性能-即频域内的互调位置与功率。

连续波（CW）信号测试是一种固有窄带操作。为进一步评估RF DAC的宽频带性能，我们需要通过并发多频、多模信号（如分别为2.1 GHz和2.6 GHz的双模UMTS和LTE信号）对其进行激发。为此，我们创建了一个基于片上BRAM阵列的数据存储内核；该内核有两个子组，可以为重复测试存储各自的双频用户数据。

图1显示了简化的系统级平台设计图。可以看到，我们采用简单直观的设计策略，构建尽量简单的平台并

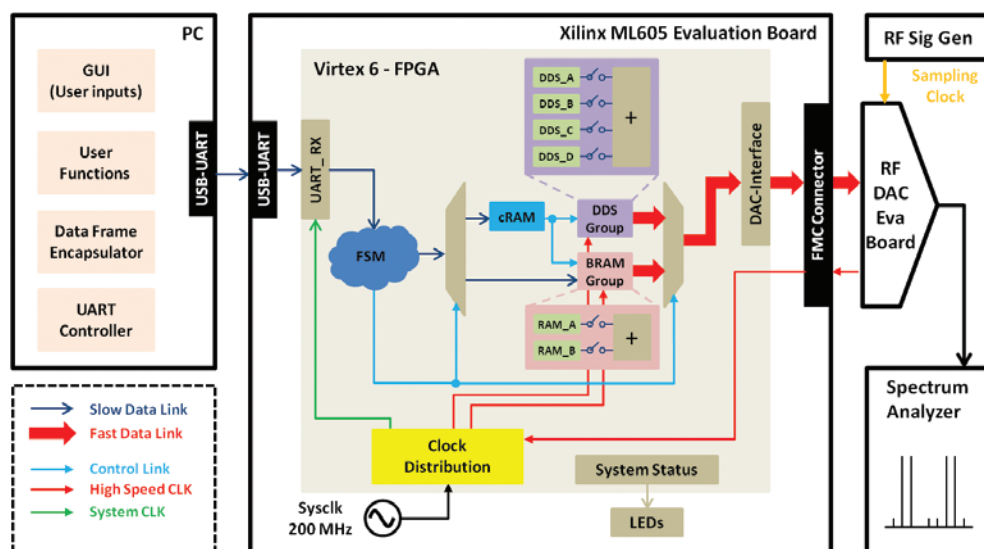


图1 - 简化的系统级平台方框图

通过升级功能对其进行模块化。

硬件设计：赛灵思 FPGA 内核

图 1 中的 FPGA 部分列出了系统基本需要实现的逻辑单元。包括时钟分布单元、基于状态机的系统控制单元和基于 DDS 内核的多音生成单元，以及嵌入在 RAM 周围的两个单元：基于 BRAM 的小型控制消息存储单元（cRAM 内核）和基于 BRAM 阵列的用户数据存储单元（dRAM 内核）。还包括连接 PC 的 UART 串行接口和连接 RF DAC 的高速数据接口。

图 3 所示）。

系统中基本上有两种类型的数据帧。带报头“FF01”的数据帧（cRAM 帧）用来为 DDSes 和系统控制消息传输相位增量值。带报头“FF10”或“FF11”的其他数据帧（dRAM 帧）用来传输用户自定义的数据。状态帧“S1x”只处理带报头“FF01”的数据，用以更新相位增量值和执行控制指令。状态帧“S2x”和“S3x”分别为两个频带接收并存储用户自定义数据。占线信号用来连续锁存数据，直至看到数据序列末尾的最后停止

音组合在一起，并通过管道将这些频音输送至下一级。由于 DDS 内核输出是二进制补码格式，如果 RF DAC 需要另一种数据格式，如偏移二进制码，则需要格式转换单元。

一般来说，高性能片上 BRAM 通常是创建中小型用户存储系统的首选。例如，在这个平台上，我们利用赛灵思 模块存储生成器（Block Memory Generator）内核为两个频带创建两个独立的数据存储 RAM。每个 RAM 的宽度为 16 位，深度为 192k。

对于 PC 与 FPGA 之间的通信，我们创建了一个 UART 串行接口单元并将其设置为相对较低的速度，即 921.6 kbps（相当于 115.2 字节 / 秒）。传输 cRAM 数据帧（18 字节）和 dRAM 数据帧（约 384k 字节）分别需要约 0.16 毫秒和 3.33 秒时间。

器件厂商通常会以 VHDL 或 Verilog 格式提供芯片高速数据接口的实例设计。对于经验丰富的 FPGA 工程师而言，复用或定制参考设计并不是很难。例如，就我们系统的 AD9739a 和 AD9129 RF DAC 而言，ADI 公司会提供并行 LVDS 接口的参考设计。顺便提一下，如果无法从芯片厂商处获得实例设计，赛灵思有几款简单易用的高速接口芯片，如 CPRI 和 JESD204B。

软件设计：MATLAB DSP 功能与图像用户界面 (GUI)

我们选择 MATLAB 作为软件主机，只是因为它在数字信号处理（DSP）性能方面具备诸多优势。另外，MATLAB 还为图形用户界面（GUI）的布局提供一种称为 GUIDE 的简便易用的工具。所以现在，对于这个项

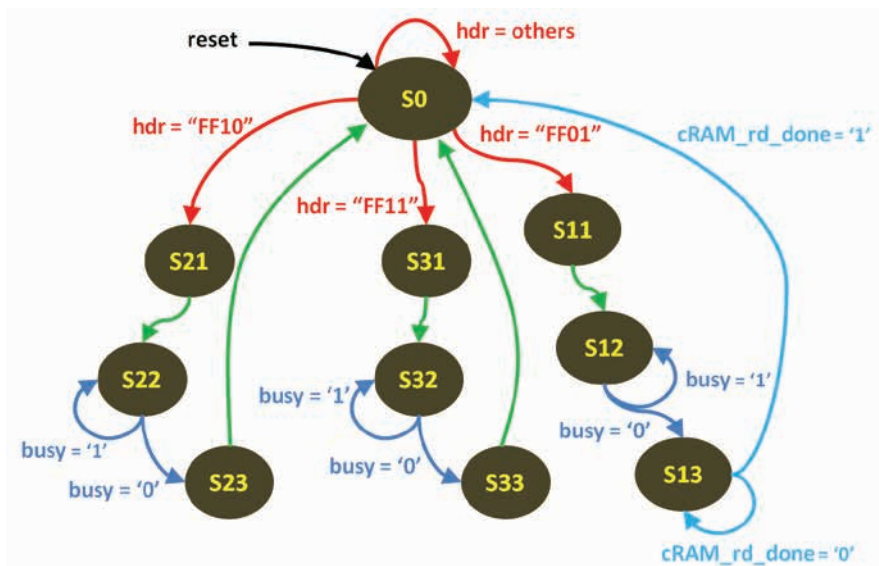


图2 - 关键状态机详细设计图

时钟是 FPGA 的生命脉搏。为确保多款时钟在 FPGA Bank 上正确分配，我们选用赛灵思时钟管理内核，为时钟的定义和指定提供一种简单的交互方式。

嵌入状态机周围的小型指令内核用作系统控制单元。如图 2 所示，在初始状态（S0）下，报头检测器单元工作，负责监测并过滤来自 UART 接收器的输入数据字节。数据字节被生成并封装在 MATLAB 数据帧内（如

位。控制消息——例如调用单个 / 多个 DDS 或用户数据序列——存储在 cRAM 数据帧的最后两个字节内。它们将在 cRAM_rd_done 信号上升沿处执行。

然后，我们举例说明四个采用赛灵思 DDS 内核的独立的频音生成单元，并将其配置为相位增量模式。特定频率的相位增量值在 MATLAB 上生成并通过 cRAM 数据帧下载到 FPGA。通过混频器，我们将多个频

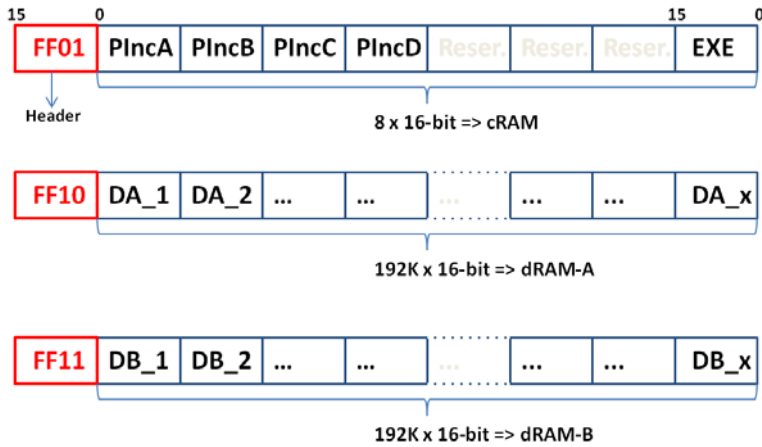


图3 - 数据帧封装例解

目，我们需要从MATLAB 获得什么？

事实上，我们需要与低级 DSP 功能和数据流控制功能相关的用户界面。所需 DSP 功能为相位增量值计算器、基带数据序列发生器和数字上变频器。控制功能为数据帧封装器、UART 接口控制器和系统状态指示器。

图 4 显示了我们为该平台创建的图像用户界面 (GUI)。应首先定义 RF DAC 关键参数——采样率，然后才能选择 xDDS 模式或 xRAM 模式激励器件。然后，在各个子面板上，

我们可以自定义参数，以调用相应 MATLAB 信号处理功能。在 xDDS 模式下，可以通过简单方程式， $phase_incr = fc * 2^{nbits} / fs$ ，计算频音 fc 与采样率 fs 的相位增量值。其中， $nbits$ 表示 DDS 用来综合频率的二进制位数量。按下“启动”按钮，生成的相位增量值会转化成定点格式并封装在带不同报头和消息的 2 字节数据帧内 (如图 3 所示)，然后通过 UART 发

至 cRAM 单元并在 FPGA 内执行。在 xRAM 模式下，我们生成基

带数据序列，将其标准化为满刻度 (带符号的 16 位) 并将其上变频为在 MATLAB 下所需的频率。通过 UART 将处理过的数据下载到 dRAM 之后，按下启动按钮，我们就可以调用宽频带信号测试。切记，要用 FPGA 侧所用相同协议参数在 MATLAB 配置 UART 串行接口。

最后，我们采用信号发生器——R&S SMU200A——来提供采样时钟，从而从逻辑上“开启”RF DAC。我们还将 RF DAC 输出连接至频谱分析仪，来评估频域内 RF DAC 的线性性能。

快速评估

在原型设计的早期阶段，关键 RF 组件的线性性能评估是一个关键问题，但通过我们的软硬件平台，在不影响性能的条件下可以快速进行这项评估。然后，可以添加 RF 功率放大器并使用所建议的平台来评估级联系统的线性。在确定非线性之后，可以执行一些数字失真算法来消除级联系统不必要的非线性。

在 FPGA 设计中合理使用赛灵思 IP 核可以大大缩短开发周期并提升数字系统的稳健性。展望未来，我们预计会将平台上的数据接口模块升级至 JESD204B 标准，以支持更高数据传输速率，从而满足多个同步 RF DAC 需求。同时，我们正在将 FPGA 主机从赛灵思 ML605 迁移至 Zynq®-7000All Programmable SoC ZC706 评估套件。Zynq SoC 设计是在单台 PC 上创建无需任何外部 DSP 和控制功能的独立解决方案的一个很好的选择。

如需了解有关平台和数字失真的更多信息，敬请发电子邮件至 lei.guan@alcatel-lucent.com，与作者取得联系。

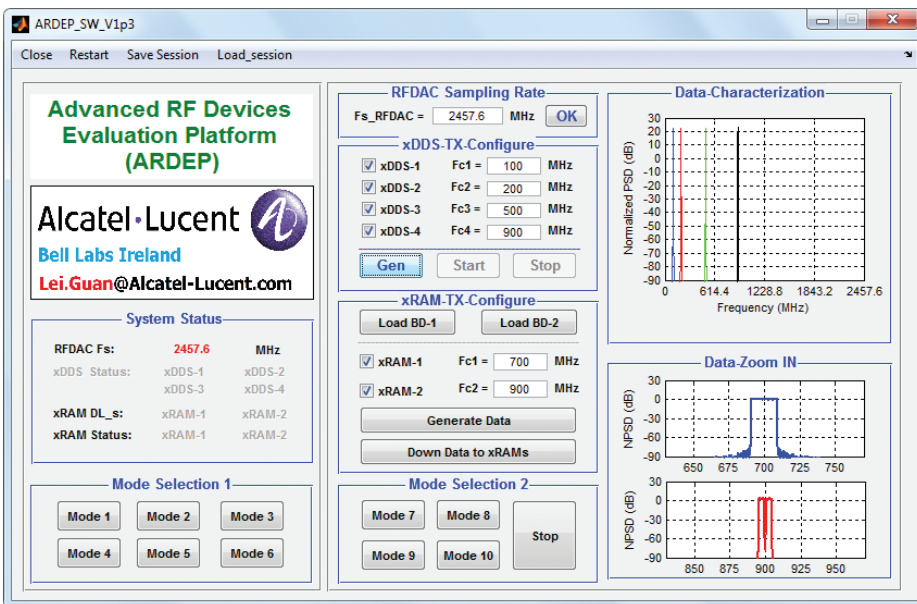


图4 - 图形用户界面截图

在低成本FPGA开发板上实现Oberon系统

Niklaus Wirth

教授（已退休）

瑞士苏黎世瑞士联邦理工学院（Swiss Federal
Institute of Technology）（ETH）

wirth@inf.ethz.ch

赛灵思 Spartan-3 开发板成为作者改 进其软件教育用的 Oberon 编程语言和 编译器的基础。

1988年，Jürg Gutknecht 和我一起完成并出版了 Oberon 编程语言 [1,2]，其为我职业生涯早期开发的另外两种编程语言 Pascal 和 Modula-2 的后续版本。与 Modula-2 相比，我们最初设计 Oberon 编程语言力求更加精简和高效，以便能够更好地帮助教学人员为计算机科学专业的学生教授系统编程。我们再接再厉，于1990年针对可采用 windows 系统和具有字处理能力的工作站开发了 Oberon 操作系统 (OS)，以此作为工作站的现代实现方式。接着，我们出版了一本书，名为《Project Oberon》，详细讲解了 Oberon 编译器以及与之同名的操作系统。此外，书中还包括详细的指令和源代码。

几年前，我的朋友 Paul Reed 建议我重新修订并再次出版这本书，不仅因为这本书对系统设计教学具有重要价值，同时还因为这本书可以作为良好的切入点，帮助那些想要成为创新者的人们从零基础构建可靠的系统。

然而，我当时实际上是遇到了很大的困难。我最初开发的编译器是将已基本消失了的处理器作为目标。因此，我的解决办法就是为现代处理器重新编写编译器。但在做了不少研究之后，我无法找到一款能够在清晰度、规律性和简洁性上符合我标准的处理器。因此，我自己设计了这款编译器。而我之所以能够将该想法付诸实现，都是因为现代 FPGA 能够帮助我设计硬件

《赛灵思中国通讯》很荣幸能发表这篇由业内具有传奇色彩的 Niklaus Wirth 所撰写的文章。Niklaus Wirth 发明了 Pascal 和几种派生编程语言并开创了一些针对计算机和软件工程的经典方法。作为 ACM 图灵奖和 IEEE 计算机先驱奖的获得者，Wirth 教授虽已从教学生涯中退休，但他仍然在继续帮助教育工作者培养并启发未来的创新者。



选择赛灵思 FPGA能帮我更新系统， 同时让设计尽可能接近自1990年 以来的原始版本。

以及系统软件。更重要的是，选择赛灵思 FPGA 能帮我更新系统，同时让设计尽可能接近自 1990 年以来的原始版本。

实现在低成本 Digilent Spartan®-3 开发板上的新型处理器 RISC 负责托管 1MB 静态 RAM (SRAM) 内存。我唯一添加的系统硬件就是一个鼠标接口和一个用来替换旧系统中硬盘驱动器的 SD 卡。

这本书和面向整个系统的源代码可在 projectoberon.com[3,4,5] 中查阅，

也可在该网站上名为 S3RISCinstall.zip 的单个文件中进行查阅。该文件包含指令、SD 卡文件系统图像和 FPGA 配置比特文件（对于 Spartan-3 开发板的 Platform Flash，此为 PROM 文件形式），以及 SD 卡 / 鼠标接口硬件的构造详图。

RISC处理器

该处理器由算术逻辑单元、由 16 个 32 位寄存器组成的阵列和带指令寄存器、IR 及程序计数器 PC 的控制单元

组成。Verilog 模块 RISC5 就是该处理器的典型代表。

该处理器具有 20 种指令：4 种用于移动、偏移和旋转；4 种用于逻辑运算；4 种用于整数运算；4 种用于浮点运算；2 种用于存储器访问；2 种用于分支。

RISC5 通过运行环境 RISC5Top 导入。该运行环境包括到各种（内存映射）器件和 SRAM(256M×32 位) 的接口。整个系统（图 1）包括以下 Verilog 模块（见行统计）：

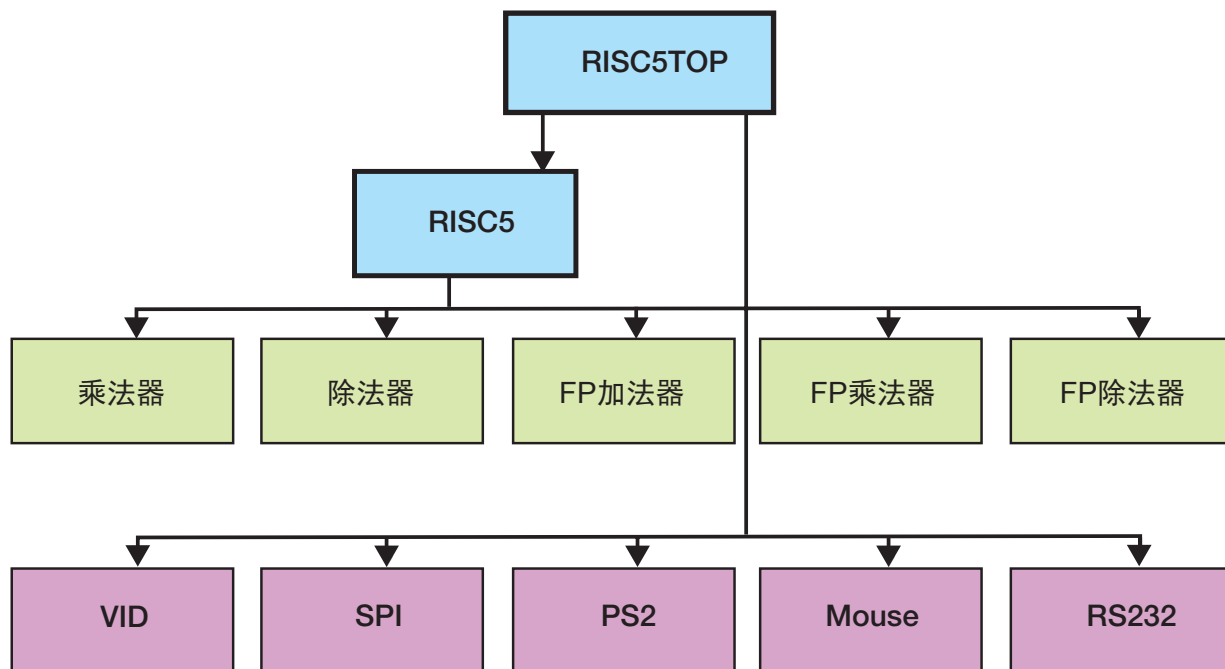


图1 – 该系统及其所含Verilog模块的方框图

RISC5Top	运行环境	194
RISC5	处理器	201
乘法器	整数运算	47
除法器		24
FP 加法器	浮点运算	98
FP 乘法器		33
FP 除法器		35
SPI	SD 卡和发送器 / 接收器	25
VID	1024×768 视频控制器	73
PS2	键盘	25
鼠标	鼠标	95
RS232T	RS232 发送器	23
RS232R	RS232 接收器	25

我将内存映射到黑白 VGA 显示器中，这样它只占用 1024×768×1 位 / 像素 = 98304 字节，基本上占 1 MB 可用主内存的 10%。该 SD 卡将取代初始系统中 80MB 的硬盘驱动器，其可通过能够接受并序列化字节或 32 位字的标准 SPI 接口进行存取。键盘和鼠标通过标准 PS-2 串行接口连接。此外，还提供一根串行异步的 RS-232 线和一个通用 8 位并行的 I/O 接口。模块 RISC5Top 还带有一个每毫秒采用增量式计数的计数器。

OBERON操作系统

该操作系统软件由包括内存分配器（带垃圾回收器）的内核、文件系统以及引导载入程序、文本系统、浏览器系统和文本编辑器组成。

名为“Oberon”的模块是中心任务调度程序，而“System”是基础命令模块。通过点击显示器上任何浏览器文本“M.P”上的中间按钮即可触发动作，其中 P 是模块 M 声明的程序名。如果 M 不存在，则会自动加载。但是，大多数文本编辑命令是通过简

单的鼠标点击触发的。其中，左边一栏按钮用来设置“脱字”符，标记文本位置，右边一栏按钮用来选择文本字段（text stretch）。

“Kernel”模块包括磁盘存储管理和垃圾回收器。我保证观察浏览器是平铺的，不重叠。标准布局显示了多个浏览器的两条垂直轨迹。只需拖动标题栏，就可以放大、缩小或移动它们。图 2 显示了在显示器上运行的用户界面以及 Spartan-3 开发板、键盘及鼠标。

加载时系统占用模块空间 112640 字节（21%），占堆（heap）的 16128 字节（3%）。系统包括所以下几个模块（见行统计），如图 3 所示：

内核	271（内核）
文件目录	352
文件	505
模块（引导载入程序）	226
浏览器	216（外壳）
文本	532
Oberon	411
菜单浏览器	208
文本帧	874
系统	420
编辑	233

值得注意的是，该系统在加电或重置时，完成初始化仅需两秒钟。这包括文件目录中垃圾回收扫描。

OBERON编译器

系统自带的编译器采用简单的自上而下递归下降分析法。用户使用 *ORP.Compile @* 命令即可激活模块选定的源文本上的编译器。包解析器通过扫描仪输入各种符号，包括识别符、数字和特殊符号（如 BEGIN、END、+ 等）。该方案已被证明在许多应用中有效且均表现不凡。这点在我著的《编译器结构》[6,7] 中有详细说明。

该包解析器调用代码生成程序

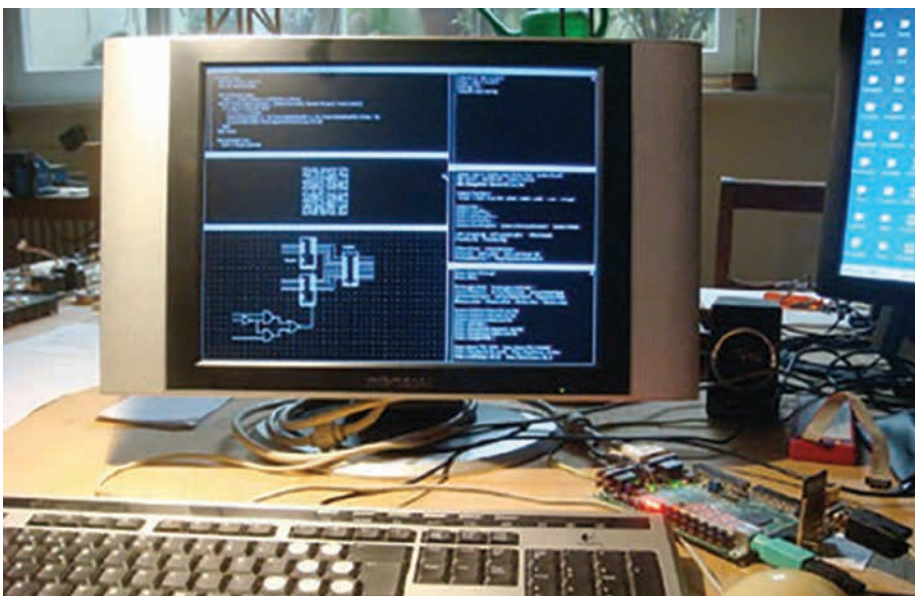


图2 - 显示用户界面的显示器（Spartan-3开发板在右侧）

模块中的程序。这些程序直接将指令添加在代码阵列上。如果已知所有分支目的地，向前跳转指令（forward-branch instructions）在模块编译结束时则会提供跳转地址（修正）。

所有可变地址都与基址寄存器关联。这就是用于局部变量（运行时设置在程序输入）的 R14（堆栈指示器）

或用于全局和输入变量的 R13。

基址地址按要求通过地址保存在寄存器 R12 内的系统全局模块表载入。R15 用于 RISC 架构确定的返回地址（链接）。因此，R0-R11 可用于表达式评估和传递过程参数。

整个编译器由 4 个相对较小的有效模块组成（见行统计）：

ORP	包解析器	968
ORG	代码生成器	1120
ORB	base def	435
ORS	扫描仪	311

编译器占用 115912 字节（22%）的模块空间和 17508 字节（4%）的堆空间（编译之前）。其源代码长约 65KB。编译器自身的编译在 25 MHz

Lola HDL及Verilog翻译

名为 Lola 的硬件描述语言（HDL）于 1990 年作为硬件设计基础教学的一种方式开发。这一时期，文本定义开始替代电路图，首个 FPGA 开始应用，尽管尚未达到主流设计。Lola 由生成位流文件（被加载到 FPGA）的编译器执行。位流文件格式由 Algotronix 公司和 Concurrent Logic 公司共同开发。这两家公司提供的存储单元结构都比较简单，看起来好像是自动布局布线的最佳选择。

在我的项目之后，紧接着是在 FPGA 上重新实现 Oberon 系统，现在突然出现这种想法以唤醒 Lola。由于赛灵思 FPGA 存储单元结构相当简单，所以我们没有冒险实现布局布线，更何况赛灵思因专利原因而拒绝透露其位流文件格式这一事实。

很明显，要建立不生成专有位流文件但翻译成赛灵思能为其提供综合工具的语言的 Lola 编译器。我们选择了 Verilog。这种解决方法意味着迂回绕路相当浪费：首先，Lola 模块已经经过包解析，然后翻译，然后再次包解析。通过所有这些措施，我们确信 Lola 编译器具有适当的错误报告和类型一致性检查功能。

为促进 Lola-2 的开发，我们决心用 Lola 重新定义所有 RISC5 处理器模块。现在，这已经实现了。

LOLA语言

Lola 是一种 Oberon 风格的短小而精

炼的程序语言（见 <http://www.inf.ethz.ch/personal/wirth/Lola/Lola2.pdf>）。

为简单起见，我们在这里只展示一个简单的 Lola 文本实例（图 1）。源文本单元被称为模块。其报头规定了名称及其输入和输出参数，以及各参

```

MODULE Counter0 (IN CLK50M, rstIn: BIT;
                 IN swi: BYTE; OUT leds: BYTE);

TYPE IBUFG := MODULE (IN I: BIT; OUT O: BIT) ^;
VAR clk, tick0, tick1: BIT;
    clkInBuf: IBUFG;
REG (clk) rst: BIT;
    cnt0: [16] BIT; (*half milliseconds*)
    cnt1: [10] BIT; (*half seconds*)
    cnt2: BYTE;

BEGIN leds := swi.7 -> swi : swi.0 -> cnt1[9:2] : cnt2;
    tick0 := (cnt0 = 49999);
    tick1 := tick0 & (cnt1 = 499);
    rst := ~rstIn;
    cnt0 := ~rst -> 0 : tick0 -> 0 : cnt0 + 1;
    cnt1 := ~rst -> 0 : tick1 -> 0 : cnt1 + tick0;
    cnt2 := ~rst -> 0 : cnt2 + tick1;
    clkInBuf (CLK50M, clk)
END Counter0.
    
```

图1 - Lola文本（显示计数器计数毫秒和秒，并在面板LED上显示秒数）

RISC 处理器上只需几秒钟 [8]。

编译器通常使用值为 NIL 的指示检查数组索引和基准值。如果违反规定，这会造成陷阱。这种技术具有高度的安全性，防止错误和崩溃。事实上，只能通过采用伪模块 SYSTEM 中的操作，即 PUT 和 COPY，才能防碍系统的完整性。这些操作必须局限于

接入器件接口的驱动模块。在输入列表中，通过 SYSTEM 很容易识别它们。整个系统采用 Oberon 编程，无需使用汇编码。

我选择 Digilent Spartan-3 开发板是由于其成本低、操作简便，这使其适于教育机构，以获得整套课堂教学套件。一个重大优势就是该开发板

上有静态 RAM，使得接口连接非常简单直观（甚至用于字节选择）。遗憾的是，所有新开发板均采用动态 RAM，虽然存储空间更大，但接口连接复杂得多，因此刷新和初始化（校准）需要电路。这一电路与带静态 RAM 的整个处理器一样复杂。即使控制器以单片式提供，这有悖于我们

数的名称和类型。报头之后是一段包含本地对象（如变量和寄存器）的声明。之后一段通过赋值规定变量和寄存器数值。BYTE 表示一个 8 位阵列。

LOLA编译器

编译器采用简单的自上而下递归下降分析法。通过 LSC.Compile @ 命令在所选 Lola 源文本上激活该编译器。包分析器包解析器通过扫描仪输入各种符号，包括识别符、数字和特殊符号（如 BEGIN、END、+ 等）。该方案已被证明在许多应用中有效且表现不凡。这在我著的书《[编译器结构](#)》（[第 1 部分](#)和[第 2 部分](#)）中有详细说明。

而不是直接即时生成 Verilog 文本，作为语法分析的一种意外结果，包分析器中会出现语句，从而生成树形结构，以适当方式表示输入文本，以便进一步处理。这种结构具有一个优点，即通过调用不同翻译程序很容易生成各种不同的输出。其中一个就是 Verilog 翻译程序。命令为 LSV. List outputfile.v。另一个命令可能翻译至 VHDL 或简单地列出树形结构。但也有可能生成网表，以便通过布局器和布线器进行进一步处理。

因此，整个编译器至少包括 4 个相对较小的高效模块组成（线数显

示）：

LSS	扫描仪	159
LSB	基址	52
LSC	编译器 / 包解析器	503
LSV	Verilog 生成器	215

可在 <http://www.inf.ethz.ch/personal/wirth/Lola/Lola-Compiler.pdf> 查阅从 Lola 翻译至 Verilog 的指令。

软件与硬件‘程序’之间的差异

在过去，已进行许多努力使 HDL 看起来像“普通”编程语言（PL）。我们还注意到，HDL 在编程语言集中通常有一个对应程序，各自采用不同风格。因此，Verilog 在 C、VHDL 在 Ada，以及 Lola 在 Oberon 中都有其各自的原型。我们认为，认识到两个类别之间的根本差异是非常重要的，特别是存在句法相似或几乎相同的时候。这些根本差异是什么？

为了简化说明，我们仅分析同步电路 - 即，所有寄存器与时钟同样发出滴答声的电路。总体来说，这确实是一种良好的设计模式，如果可能的话，附在同步电路上。那么，很明显，一个电路的所有元件同时（或几乎同时）工作。每一个变量和寄存器

通过唯一一种表达式（组合电路）进行定义。多重赋值没有任何意义。可以想象，每个 HDL 程序包含在一个大的永远重复子句中，因为寄存器和变量赋值会在每一个同步时钟脉冲周期内重复。

这是 John von Neumann 提出带定序器的处理器体系结构的巧妙构思。定序器含有指令寄存器，据此，（各个周期中）某些电路会被选用而其他电路会被忽略，从而巧妙地重新利用（ALU）不同部件。现在，周期或步骤已成为固有顺序，并有可能将数值重新分配到同一个变量，因为程序计数器将它们与程序中和指令序列中的某些次序关联起来。真是这种定序器构思使得通过相对简单电路执行庞大程序成为可能。

总之，Lola-2 是 PL Oberon 风格下的 HDL。这里所述编译器将 Lola 模块翻译为 Verilog 模块。Lola 的优点在于语言的简单、规则结构以及编译器对类型检查和改进错误诊断的突出。整个 RISC 处理器模块已通过 Lola 表示（见 <http://www.inf.ethz.ch/personal/wirth/Lola/index.html>）。

— Niklaus Wirth

的开放检查原则。

最后的想法

40 多年前，C.A.R. Hoare 说，在科学

技术的所有分支学科中，学生在被要求用实验尝试自己的想法之前，容易受到许多示范性设计实例的影响。编程和软件设计与这种明智模式形成了

鲜明对比。这里，在阅读任何实例之前，要求学生从一开始就编写程序。

这种可怕事实的原因就是：几乎不存在大小合适的示范实例。因此，我决定稍微修正这种情形，我在 1975 年编写了《算法与数据结构》一书。随后（与 J. Gut-knecht）负责操作系统教学任务，我设计了 Oberon 系统（1986-88）。

此后，编程教学并没有明显提高，而系统的规模和复杂性显著增加。虽然开源的努力颇受欢迎，但它并没有真正改变这种形势，因为大多数程序已经建成“运行”，但却没有真正投入使用。

我继续大胆提议，应重新设计所有程序，不光是计算机，还有读取方式。这些程序应是可发行的。即便这种方法正确而有效，但这是一项比创建可执行程序更为艰难的任务。这意味着，没有任何部分必须用汇编码指定。

忽略在这种“人为因素”的结果就是，在许多地方，新的应用没有得到精心设计，而是通过消除程序中错误来实现，其结果有时是令人担忧的。要做到“易懂”的关键就是坚持简单性和规律性，放弃不必要的修饰，避免花里胡哨，正确区分传统与便捷。

该系统所占空间小，是小资源实现大作为的见证。虽然 Oberon 系统包含一个文件系统、文本编辑器和浏览器（Windows）管理程序，但相比多数现代操作系统而言，其占用的空间少得可怜。意外的是它仅仅依靠一些简单规则，因此便于学习如何使用。

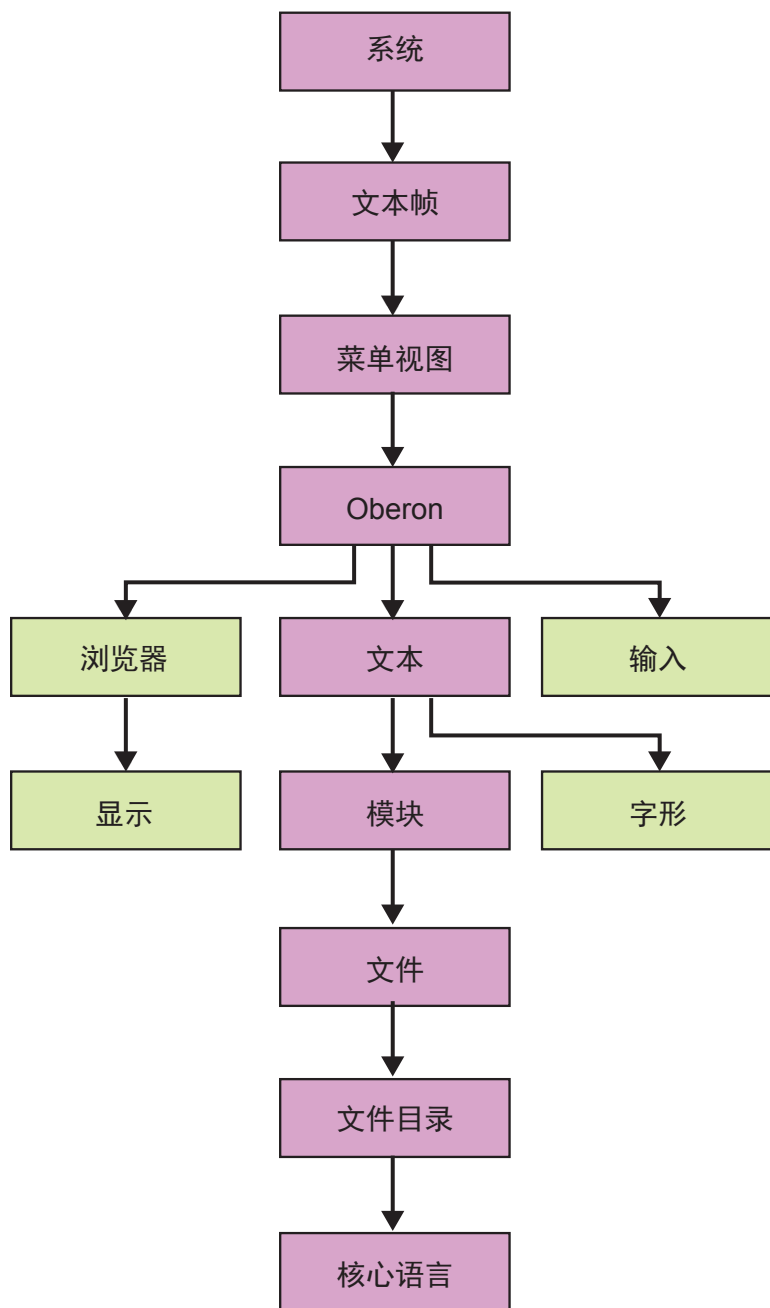


图3 - 系统及其模块

最后，这种简单的系统的另一大优势就是，用户可以在该基础系统上安全构建所需程序，而不必担心是否存在未知特征与功能，如后门程式（back door）。从系统完整性易受到攻击的角度看，这是一个基本属性，是关键安全型应用不可或缺的。值得注意的是，我们的系统硬件没有这种隐藏元件。毕竟，在如此之大而又没人能够完全了解的基础上构建的任何系统也不能予以保证。🌈

致谢

我衷心感谢 Paul Reed 所做的宝贵贡献。他建议我重新编撰《Oberon 项目》一书，还建议在 FPGA 上重新实现整个系统。Paul 一直激励着我。他想到过用 SD 卡替代磁盘，他为 SPI、PS-2 和 Verilog 中的 VID 接口做出了卓越贡献。


参考文献

1. <http://www.inf.ethz.ch/personal/wirth/Oberon/Oberon07.Report.pdf>
2. <http://www.inf.ethz.ch/personal/wirth/Oberon/PIO.pdf>

Oberon/PIO.pdf

3. www.inf.ethz.ch/personal/wirth/ProjectOberon/index.html
4. www.inf.ethz.ch/personal/wirth/Oberon/PIO.pdf
5. www.inf.ethz.ch/personal/wirth/Oberon/Oberon07.Report.pdf
6. <http://www.inf.ethz.ch/personal/wirth/CompilerConstruction/CompilerConstruction1.pdf>
7. <http://www.inf.ethz.ch/personal/wirth/CompilerConstruction/CompilerConstruction2.pdf>
8. <http://www.inf.ethz.ch/personal/wirth/ProjectOberon/PO.Applications.pdf>(Ch.12)

Xilinx All Programmable Low-End Portfolio




Xilinx 低端产品系列的产品列表和产品选择指南

对于需要低成本、低密度可编程解决方案的应用而言，Xilinx 全可编程低端产品系列可为各种标准的集成度、性能及功耗提供最丰富的特性集以及最低成本的解决方案。

如此多选项，该如何选择？我们可帮助您把您的应用需求映射至器件功能，让选择变得简单易行。这一简单易用的指南不仅将重要器件指标进行了细分，而且还以图形方式显示 Xilinx 低端 FPGA 及全可编程 SoC 产品系列的主要产品特性。每个器件系列都能为目标应用提供最大价值：

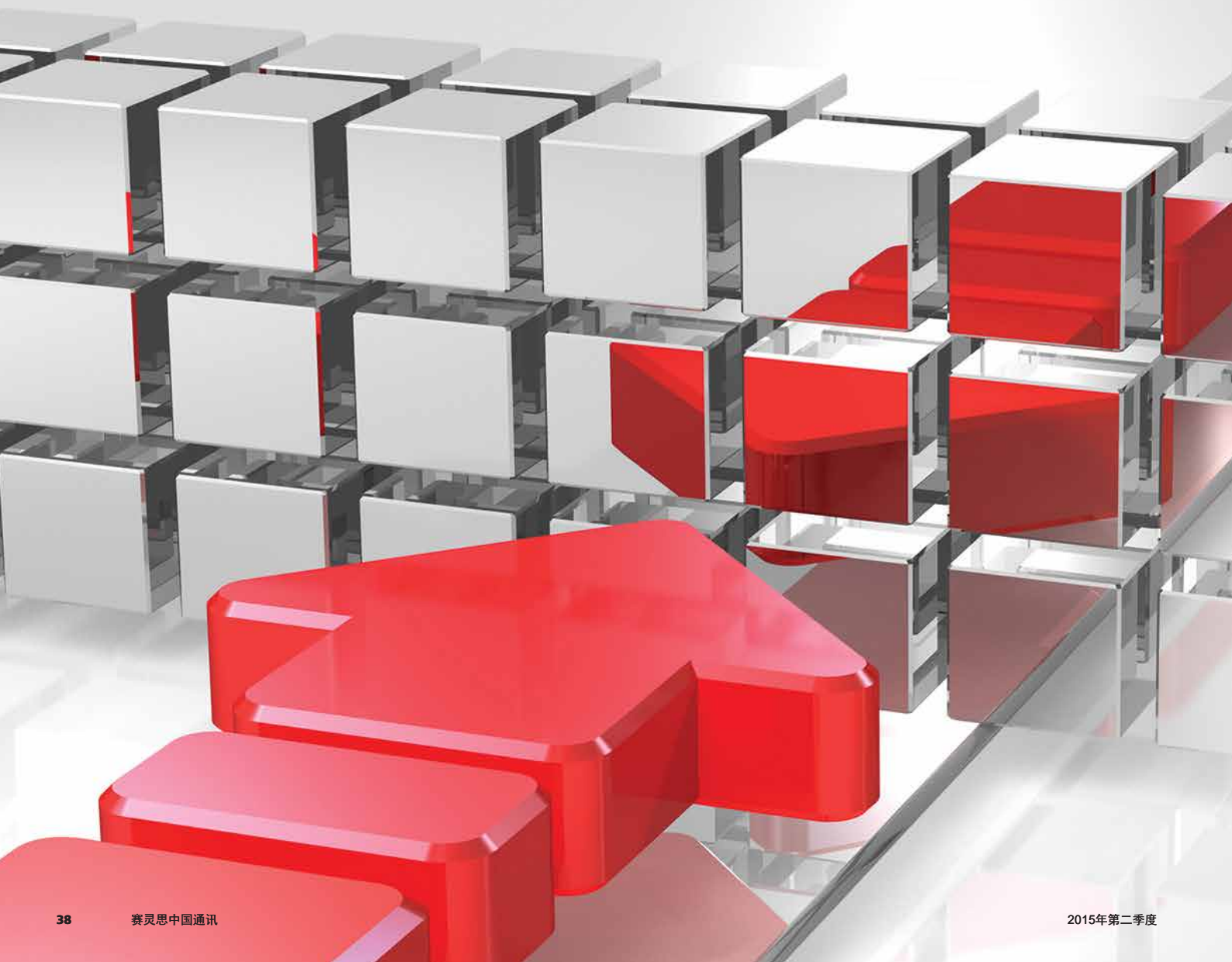
- Artix[®]-7 FPGA：专为收发器而优化
- Spartan[®]-6 FPGA：专为 I/O 而优化
- Zynq[®]-7000 全可编程 SoC：面向系统集成和优化

了解有关 Xilinx 全可编程低端产品系列产品列表及产品选择指南的更多详情。



清除通向基于FPGA的 OpenCL数据中心服务器 的障碍

赛灵思 SDAccel 环境能在 FPGA 上提供类似 CPU 的开发与运行时间体验，减轻数据中心设计负担。



作者:

Devadas Varma,

赛灵思公司SDAccel和Vivado高层次综合事业部高级工程总监。

Dvarma@xilinx.com

Tom Feist

赛灵思公司设计方法市场营销部高级总监

Tfeist@xilinx.com

从支持中小机构运作的服务机房，到支持美国大型企业和提供云计算服务接入的企业级数据中心，数据中心在现代经济中发挥着骨干作用。根据自然资源保护委员会的统计，数据中心是美国用电量最大、增长最快的用电大户之一。

2013年美国数据中心估计耗电约910亿千瓦时，超过纽约市全部居民用电的两倍以上，而且到2020年用电量有望达到1400亿千瓦时[1]。很明显，降低用电对于扩展数据中心，提升可靠性和降低运营成本具有重要意义。

根据具体的服务器应用，数据中心各个不同。许多服务器长期不间断运行，使得硬件可靠性和耐用性极为重要。虽然服务器可以使用商用计算机部件组装，但关键任务型企业级服务器往往使用以硬件加速为目的的专用硬件，如图形处理单元（GPU）和数字信号处理器（DSP）。现在许多企业寻求使用现场可编程门阵列（FPGA），因为FPGA是一种高度并行的架构，而且功耗相对较低。赛灵思的新款SDAccel™开发环境为开发人员提供了一种熟悉的CPU或类似CPU的环境，避免编程给FPGA在此类应用中使用时造成的障碍。

提升性能功耗比

Amazon Web 服务、Google Compute、Microsoft Azure 和中国的百度等公有云（public cloud）拥有巨大的图片库，需要极快的图像识别能力。在一个实现方案中，谷歌科技人员将16000个计算机处理器连接为一个实体，创建出了一个极为庞大的机器学习神经网络，然后投放到互联网上，让其自主学习。这项研究是新一代计算机科学的代表。这一代计算机科学以充分利用大型数据中心中的大

量计算机集群的运算能力为目标。潜在应用包括让图像搜索、语音识别和机器语言翻译能力迈上新的台阶。但是对数据中心设计而言，仅依靠充分利用CPU并非是一种高能效的做法。要提高速度、降低功耗，还需要其他解决方案。

中国最大的搜索引擎百度借助于深度神经网络处理技术来解决语音识别、图像搜索和自然语言处理方面的问题。百度迅速判定，如果在在线预测中使用神经网络反向传播算法，FPGA解决方案在降低功耗的同时，还能以比CPU/GPU简便得多的方式进行扩展[2]。

因在数据中心服务器的主机卡和线路卡中集成了FPGA，新一代28nm和20nm高集成度FPGA系列（如赛灵思7系列和UltraScale™器件）正在改变数据中心动态发展状况。性能功耗比可以轻松达到CPU/GPU的20倍以上，同时在某些应用中与传统CPU相比，时延可降低50至75倍。

但是对FPGA硬件资源有限或缺乏的开发团队而言，由于需要使用RTL（VHDL或Verilog）开发专业知识才能充分发挥FPGA的性能优势，因此过渡到FPGA难度较大。为解决这一问题，赛灵思已引入开放计算机语音（OpenCL™）作为减轻编程负担的方法。

OpenCL代码移植性

由苹果公司开发并经Khronos集团推广的OpenCL[3]有助于异构设计中的CPU、GPU、FPGA和DSP模块集成。为增强用于编写可在异构平台上运行的程序的OpenCL框架，赛灵思等业界领先的CPU、GPU和FPGA厂商都在为这种语音及其API的开发做出努力。

SDAccel编译器相对CPU性能提高10倍，且功耗仅为GPU的1/10。

OpenCL 被 CPU/GPU/FPGA 厂商、服务器 OEM 厂商以及数据中心管理人员等日益广泛地接受，说明各方都已经认识到一个严峻的现实：用于单处理器架构的 C 语言编译器在服务器机架内部只能实现小幅总体功耗降低，即便是在处理器采用低于 20nm 的工艺技术并添加特殊省电状态后依然如此。

OpenCL 是一种用于编写可在由 CPU、GPU、DSP、FPGA 及其它处理器构成的异构平台上运行的程序的框架。OpenCL 包含基于 C99 的编程语言和应用编程接口（API），以控

制平台和在目标器件上执行程序。OpenCL 使用基于任务和基于数据的并行机制提供并行计算功能。

针对OpenCL的赛灵思SDAccel开发环境

在开发特定领域规格描述环境方面，赛灵思有近十年的工作经验。数据中心管理人员和服务器 / 交换机 OEM 厂商对数据中心性能的担忧，迫使开发环境向统一环境方向纵向发展，以在数据中心应用中实现设计优化。在此情况下，一种用于应用加速的 OpenCL 开发环境 — SDAccel™ — 应

运而生。

最新赛灵思 SDAccel 环境（如图 1 所示）为数据中心应用开发人员提供完整的基于 FPGA 的硬件和 OpenCL 软件。SDAccel 内含一个快速且架构优化编译器，能高效利用片上 FPGA 资源以及用于代码开发、特性分析和调试的基于 Eclipse 集成设计环境（IDE）的熟悉软件开发流程。该 IDE 可提供类似 CPU/GPU 的工作环境。

此外，SDAccel 运用赛灵思的动态可重配置技术，可提供针对将即时换入换出的不同应用优化的加速器内核。这些应用能够在运行时间内将多个内核换入或换出 FPGA，且不会给服务器 CPU 和用于持续应用加速的 FPGA 之间的接口造成扰动。

SDAccel 架构优化的编译器支持软件开发人员优化和编译流、低延迟定制数据路径应用。SDAccel 编译器支持使用 C、C++ 和 OpenCL 任意组合的源代码，主要面向赛灵思高性能 FPGA 器件。SDAccel 编译器与高端 CPU 相比，性能提升 10 倍，功耗仅为 GPU 的 1/10，同时保持代码兼容性和传统软件编程模式，便于应用移植并有助于降低成本。

SDAccel 是唯一基于 FPGA 的开发环境，内含各种用于应用加速并针对 FPGA 优化的库。该库包含 OpenCL 内置函数、任意精度数据类

SDAccel™开发环境
在FPGA上实现类似CPU/GPO的开发体验

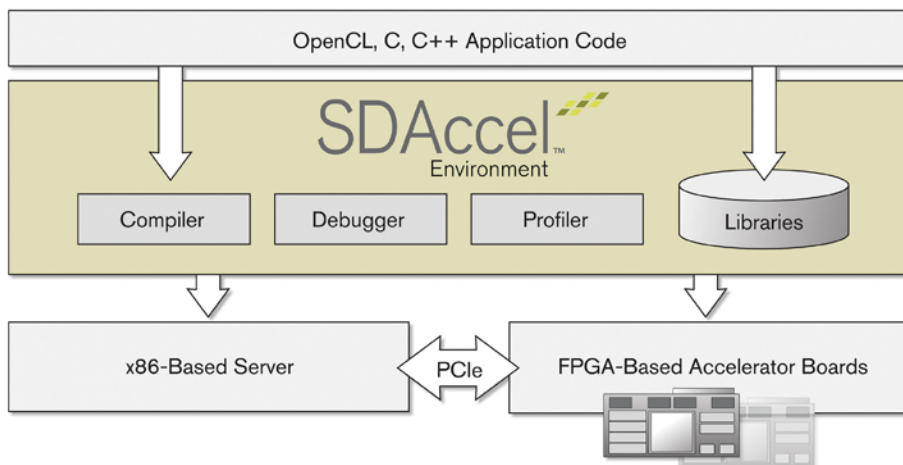


图1 - SDAccel环境包含架构优化编译器、库、调试器和分析器，可提供类似CPU/GPU的编程体验。

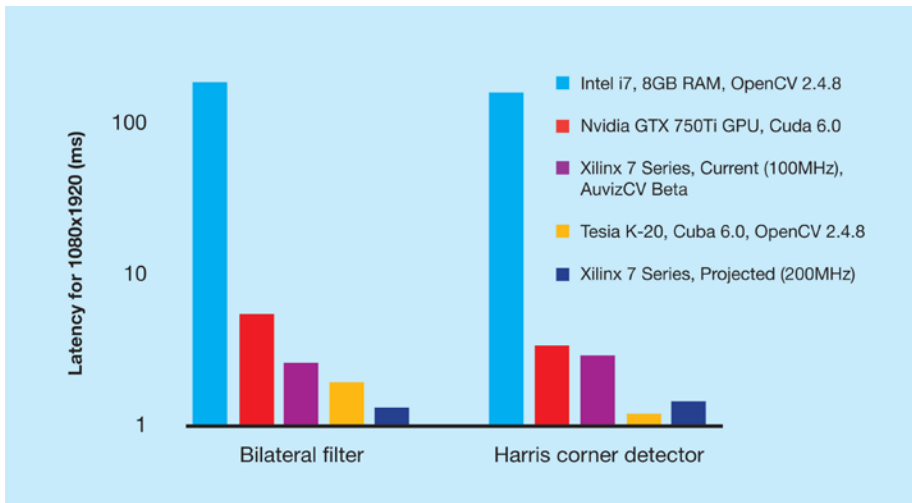


图2 - 运用OpenCL语言为CPU、GPU和FPGA架构编写的视频处理算法在FPGA上运行速度更快

(Auviz使用AuvizCV库完成的基准测试)

型(定点)、浮点、math.h、视频、信号处理和线性代数函数。

在诸如带有复杂嵌套数据路径(nested datapath)的视频处理等实际计算工作负载上，FPGA架构的内在灵活性使其相对于CPU和GPU固定架构而言，在性能和功耗方面具有明显的优势。如图2中所示的基准测试结果表明，SDAccel编译的FPGA解决方案在性能上超过相同代码的CPU实现方案，并提供可与GPU实现方案相媲美的卓越性能。

双边滤波器和Harris角点检测器均使用标准的OpenCL设计模式编码，使用器件的全局存储器在内核之间传输数据。SDAccel生成的FPGA实现方案通过创建片上存储器组供高带宽存储器传输和低时延计算使用，可达到优化存储器访问的目的。创建和使用这些专用存储器组代表SDAccel编译器的部分架构感知功能。

软件工作流

FPGA一直有望超越CPU和

GPU实现方案，拥有更高的算法性能以及更低的功耗范围。但直到现在因为编程模式未能如愿以偿。而这一编程模式又是有效利用FPGA所必须的。SDAccel支持具备系统内即时可重配置功能的软件工作流，能最大化数据中心的硬件加速ROI，从而克服这一障碍。SDAccel是一种独特而完整的基于FPGA的解决方案，它的功能和简便易用性远远超越了竞争对手的工具。更多详情，敬请访问下列网站：<http://china.xilinx.com/products/design-tools/sdx/sdaccel.html>。

参考资料：

1. <http://www.nrdc.org/energy/data-center-efficiency-assessment.asp>
2. <http://www.pcworld.com/article/2464260/microsoft-baidu-find-speedier-search-results-through-specialized-chips.html>
3. <https://www.khronos.org/opencv>

Xilinx 发布新版 SDAccel 开发环境加速数据中心应用 拥有更强大平台、库和设计服务生态系统

完整的 OpenCL、C 和 C++ 开发环境，将数据中心的性能功耗比提升高达 25 倍

2015年5月27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))今天宣布推出面向 OpenCL™、C 和 C++ 的 2015.1 版本 SDAccel™ 开发环境。SDAccel 是面向系统和软件工程师而打造的 SDx™ 系列开发环境成员之一，可将利用 FPGA 实现数据中心应用加速的性能功耗比提升高达 25 倍之多。新版 SDAccel™ 开发环境增强了 SDAccel 集成开发环境 (IDE) 的特性，扩展了 OpenCL 标准合规性，并拥有了一个由 SDAccel 认证联盟成员所组成的更强大的生态系统，用以提供平台、库和设计服务。

SDAccel 开发环境增强功能

SDAccel 开发环境 2015.1 版本新增了调试和特性分析功能，让基于 Eclipse 的 IDE 变得更简单易用，可加速 OpenCL、C 和 C++ 内核的开发与部署。此外，新版 SDAccel 开发环境扩展了 SDAccel 的 Khronos 标准合规性，现可支持新的 OpenCL ICD (Installable Client Driver) 客户端驱动程序。扩展 ICD 支持后，OpenCL 的多种实现方案可在同一系统中共存，从而使应用开发人员能够在 CPU、GPU 和 FPGA 之间实时切换，加速运行时间并降低功耗。

不断向平台、库和设计服务供应商扩展的生态系统

在 2014 年 11 月宣布推出的 SDAccel 认证生态系统基础上，赛灵思已经迎来更多提供开发板、特定市场专用库和设计服务的新联盟成员。新的开发板包括赛灵思 Kintex UltraScale KCU105 评估套件、支持混合存储立方体 (HMC) 的美光 (Micron) Pico Computing SB-850 开发板，Alpha Data ADM-PCIE-KU3 开发板和 4DSP CES820 现成商用 (COTS) 小型嵌入式系统。

全新的和更新的赛灵思最佳库包括 OpenCV、BLAS 和 Auviz Systems 提供的机器学习深度神经网络 (DNN) 库。此外，新联盟成员 ArrayFire 也提供了机器学习库。

除了加速终端应用开发所需的算法、库和 FPGA 内核外，赛灵思还打造了一个由设计服务联盟成员构成的全球生态系统，帮助全球设计团队推动 OpenCL、C 和 C++ 应用的 FPGA 加速。新的设计服务成员包括 ArrayFire、Cluster Technology Limited、Impulse Accelerated Technologies、Instigate Design、Irish Centre for High-End Computing (ICHEC) 和 MulticoreWare 公司。

使用FPGA优化视频水印操作的OpenCL应用

赛灵思 SDAccel 开发环境为内存限制问题提供优化方法

作者:

Jasmina Vasiljevic

多伦多大学研究员

vasiljev@eecg.toronto.edu

Fernando Martinez Vallina博士

赛灵思公司软件开发经理

vallina@xilinx.com



视频流和下载通常会耗掉消费者绝大部分互联网流量，同时也是云计算技术发展的主要推动力。对视频流和下载需求的持续增长，正在驱动视频处理应用迈出专业系统领域，步入数据中心。这一应用模式的转变需要具备快速扩展能力的计算节点来满足视频内容制作和分发的各个不同高计算强度阶段的需求，如转码需求和水印需求。

我们近期使用赛灵思 SDAccel™ 开发环境来编译和优化专为 FPGA 加速卡采用 OpenCL™ 编写的视频水印应用。视频内容提供商使用水印起到广告和内容保护的作用。我们的目的是设计一种能处理运行在 Alpha Data ADM-PCIE-7V3 卡上，吞吐量为 30fps，分辨率为 1080p 的高清（HD）视频的水印应用。

SDAccel 开发环境能让设计人员先用 OpenCL 编写应用，然后在无需了解底层 FPGA 实现工具的情况下把应用编译到 FPGA 中。可以以这种视频水印应用为例来介绍 SDAccel 中的主要优化技巧。

带标识插入功能的视频水印

该视频水印算法的主要功能是在视频

流的特定位置覆盖一个标识。用于水印的标识可以是活动的，也可以是静止的。活动标识一般采用简短的重复性视频片段来实现，静止标识则采用静止图像。

广播企业宣传自己视频流最常用的方法是把企业标识用作静止水印，因此成为我们实例设计的目标。该应用根据下列等式，以逐像素粒度插入静止标识。

$$\begin{aligned} \text{out}_y[x][y] &= (255-\text{mask}[x][y]) * \text{in}_y[x][y] + \text{mask}[x][y] * \text{logo}_y[x][y] \\ \text{out}_{cr}[x][y] &= (255-\text{mask}[x][y]) * \text{in}_{cr}[x][y] + \text{mask}[x][y] * \text{logo}_{cr}[x][y] \\ \text{out}_{cb}[x][y] &= (255-\text{mask}[x][y]) * \text{in}_{cb}[x][y] + \text{mask}[x][y] * \text{logo}_{cb}[x][y] \end{aligned}$$

输入和输出帧为二维阵列，像素使用 YCbCr 色域表达。在该色域中，每个像素用三个分量表达。Y 表示亮度分量，Cb 表示色度蓝色色差分量，Cr 表示色度红色色差分量。每个分量都用一个 8 位值表达，因为每个像素为 24 位。

该标识是一个包含待插入内容的二维图像。掩膜也是一个图像，但只包含标识的轮廓图。掩膜的像素可以是白色或黑色。掩膜的白色像素表

示标识的插入位置，黑色像素则表示原始像素未被触及的地方。图 1 所示的，就是这种视频水印算法的运算方式实例。

目标系统和初始实现方案

我们运行该应用的系统如图 2 所示。该系统由 Alpha Data ADMPCIE-7V3 卡组成，该卡通过 PCIe® 链路 with x86 处理器通信。在该系统中，主机

处理器从磁盘提取输入视频流，将其传输到设备全局内存中。设备全局内存位于 FPGA 卡上，可供 FPGA 直接访问。除把视频帧存放到设备全局内存中外，标识和掩膜也从主机传输到 FPGA 加速器卡上并存入片上内存中，以充分利用 BRAM 内存的低时延优势。因为本应用使用的是一个静止标识，只需在片上内存中存储静止图像和布局位置数据。

创建数据后，主机处理器会给

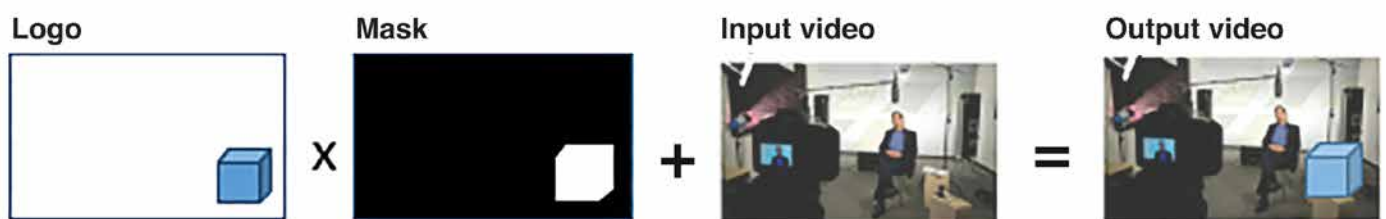


图1 - 工作中的视频水印算法

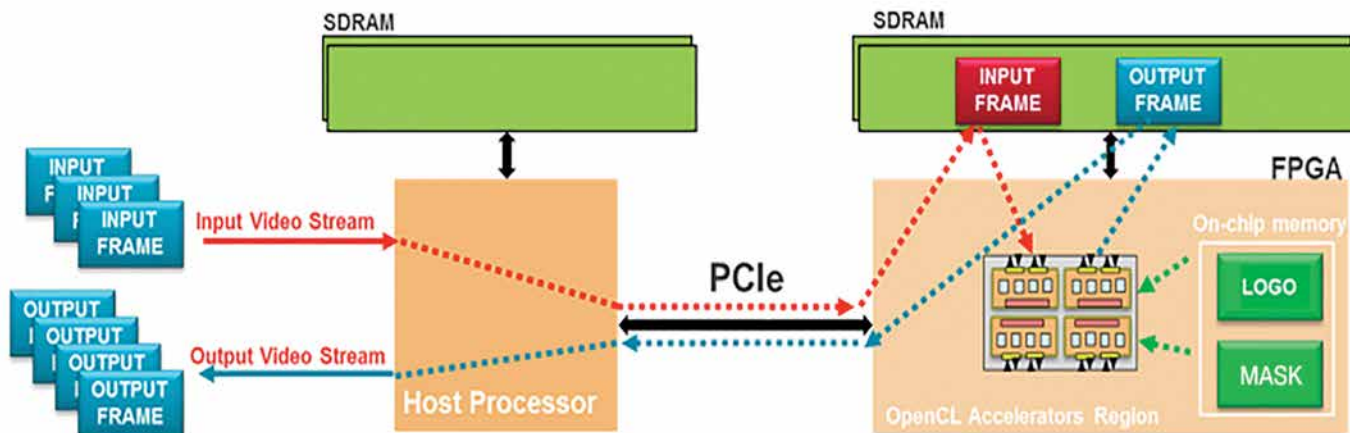


图2 - 视频水印应用系统总览图

FPGA 架构中的水印内核发送一个启动信号。该信号触发内核完成三件工作：开始从设备全局内存获取输入视频帧；在掩膜定义的位置插入标识；将处理过的帧传输回设备全局处理器，等待处理器调用。

视频流中每帧的数据传输与计算的协调工作使用图 3 所示的代码完成。

该代码运行在主机处理器上，负责发送视频帧到 FPGA 加速器卡，启动加速器，然后从 FPGA 加速器卡取回处理后的帧。

FPGA 水印算法的首个实现方案如图 4 所示。这是一个功能正确的应用实现方案，但没有进行任何性能优化或为充分利用 FPGA 架构的功能进行考虑。因此该代码在 SDAccel 中编译完成后，在 Alpha Data 卡上运行得到的最大吞吐量仅为 0.5fps。

从图 4 的代码中可以看到，这种水印算法不是一种高计算强度的设计。大多数时间花在访问内存，读取

和写入视频帧上。因此我们在优化实例设计时，把重点放在优化内存带宽上。

使用矢量化优化内存访问

与其他软件可编程架构相比，FPGA 架构的优势之一在于灵活性强，能配置连接内存的总线。SDAccel 能根据具体的应用内核创建用于连接内存的定制化数据路径和架构。通过修改代码，一次可以处理多个像素，从而能够从内核中调用更高的内存带宽。这个过程称之为矢量化。

矢量化的程度是否合适，取决于具体应用和所使用的 FPGA 加速器卡。以 Alpha Data 卡为例，设备全局内存接口宽度为 512 位，这与 SDAccel 为内核提供的最大 AXI 互联宽度一致。鉴于最大带宽为 512 位，该应用调整为每次处理 20 个像素（24 位 / 像素 × 20 像素 = 504 位）。SDAccel 完全支持矢量数据类型。因此就本应用而言，代码的矢量化非常简单，就是把所有阵列的数据类型修

改为 char20（如图 5 所示），这样吞吐量就能达到 12fps。

使用突发模式优化内存访问

虽然矢量化能显著改善应用性能，但仍不足以实现 30fps 的吞吐量目标。该应用仍然受内存局限，因为内核每次只能向内存传输 20 个像素。为减轻内存限制对应用造成的影响，我们不得不修改内核代码，以生成到内存的突发读取 / 写入操作，从而实现大于 20 个像素的数据集。修改后的内核代码见图 6。

代码内核首先修改的是在内核中定义片上存储，以便每次存储像素块。片上内存用内核代码中声明的阵列来定义。为启动到内存的突发事务处理，该代码实例化 memcpy 命令，以将数据块从 DDR 移到内核内的 BRAM 存储系统中。根据片上内存资源的大小和待处理数据的量，一个视频帧可分割成 20 个 1920×54 像素块（如图 7 所示）。

当 memcry 命令把数据块放置到内核阵列中，该算法就会在数据块上执行水印算法，然后把结果放回内核阵列。数据块处理的结果随后使用 memcry 命令传送回 DDR 内存。反复执行这个操作 20 次，直至给定帧中所有的数据块处理完毕。通过修改内核代码，系统性能达到了 38fps，超过了既定的 30fps 目标。

Host Code

```

SEND FRAME {
for (i=0; i<FRAMES; i++) {
    // Send a video frame to the FPGA device
    err = clEnqueueWriteBuffer(commands, d_frin, CL_TRUE, 0,
        sizeof(int) * LENGTH_FRAN, h_frin, 0, NULL, &writeEvent0;
    clWaitForEvents(1, &writeEvent);

    // Run logo insertion on the input frame
    COMPUTE FRAME {
    err = clEnqueueTask(commands, kernel_load_block, 0, NULL,
        &kernelEvent);
    clWaitForEvents(1, &kernelEvent);

    // Read the output frame back to CPU
    RECEIVE FRAME {
    err = clEnqueueReadBuffer( commands, d_frou, CL_TRUE, 0,
        sizeof(int) * LENGTH_FROUT, h_frou, 0, NULL, &readEvent);
    clWritForEvents(1, &readEvent);
    }
}
}

```

图3 - 用于协调每帧数据传输和计算的代码

```

for (y=0; y<FRAMES_HEIGHT; y++){
    for (x=0; x<FRAMES_WIDTH/20; x++) {

        i = y*FRAME_WIDTH/20 + x;
        out_y[i] = (255-mask[i]) * in_y[i] + mask[i] * logo_y[i];
        out_cr[i] = (255-mask[i]) * in_cr[i] + mask[i] * logo_cr[i];
        out_cb[i] = (255-mask[i]) * in_cb[i] + mask[i] * logo_cb[i];

    }
}

```

图4 - 水印内核的初始实现方案


```

for (y=0; y<FRAMES_HEIGHT; y++) {
  for (x=0; x<FRAMES_WIDTH/20; x++) {

    i = y*FRAME_WIDTH/20 + x;

    out_y[i] = (255-mask[i]) * in_y[i] + mask[i] * logo_y[i];
    out_cr[i] = (255-mask[i]) * in_cr[i] + mask[i] * logo_cr[i];
    out_cb[i] = (255-mask[i]) * in_cb[i] + mask[i] * logo_cb[i];
  }
}
    
```

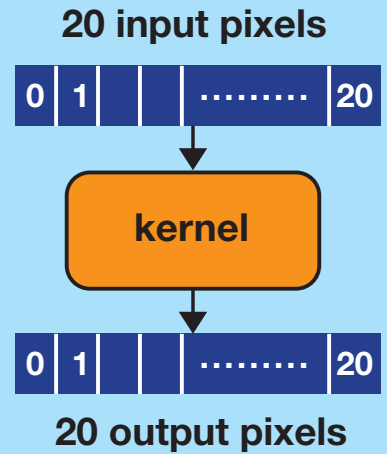


图5 - 矢量化后的内核代码

```

char20 l_in_y[BLOCK_WIDHT*BLOCK_HEIGHT/20];
char20 l_in_cr[BLOCK_WIDHT*BLOCK_HEIGHT/20];
char20 l_in_cb[BLOCK_WIDHT*BLOCK_HEIGHT/20]; } On-chip memory storage for the input block

char20 l_out_y[BLOCK_WIDHT*BLOCK_HEIGHT/20];
char20 l_out_cr[BLOCK_WIDHT*BLOCK_HEIGHT/20];
char20 l_out_cb[BLOCK_WIDHT*BLOCK_HEIGHT/20]; } On-chip memory storage for the ioutput block

for (block_id=0; block_id<FRAME_HEIGHT/BLOCK_HEIGHT; block_id++){ ← Loop through 20 blocks

  memcpy(l_in_y, in_y + block_id*BLOCK_WIDHT*BLOCK_HEIGHT/20, BLOCK_WIDHT*BLOCK_HEIGHT*3);
  memcpy(l_in_cr, in_cr + block_id*BLOCK_WIDHT*BLOCK_HEIGHT/20, BLOCK_WIDHT*BLOCK_HEIGHT*3);
  memcpy(l_in_cb, in_cb + block_id*BLOCK_WIDHT*BLOCK_HEIGHT/20, BLOCK_WIDHT*BLOCK_HEIGHT*3);

  for (y=0; y<BLOCK_HEIGHT; y++) {
    for (x=0; x<BLOCK_WIDTH/20; x++) {

      i = y*BLOCK_WIDTH/20 + x;

      l_out_y[i] = (255-mask[i]) * l_in_y[i] + mask[i] * logo_y[i];
      l_out_cr[i] = (255-mask[i]) * l_in_cr[i] + mask[i] * logo_cr[i];
      l_out_cb[i] = (255-mask[i]) * l_in_cb[i] + mask[i] * logo_cb[i];
    }
  }

  memcpy(out_y, l_out_y + block_id*BLOCK_WIDHT*BLOCK_HEIGHT/20, BLOCK_WIDHT*BLOCK_HEIGHT*3);
  memcpy(out_cr, l_out_cr + block_id*BLOCK_WIDHT*BLOCK_HEIGHT/20, BLOCK_WIDHT*BLOCK_HEIGHT*3);
  memcpy(out_cb, l_out_cb + block_id*BLOCK_WIDHT*BLOCK_HEIGHT/20, BLOCK_WIDHT*BLOCK_HEIGHT*3);
}
    
```

PHASE 1
PHASE 2
PHASE 3

图6 - 针对突发数据传输优化的内核代码

应用前景广泛

使用 SDAccel 开发本文介绍的这类应用时所进行的必要优化属于软件优化。因此这些优化工作与从其他处理架构中（如 GPU）获取性能所开展的优化类似。使用 SDAccel 后，让 PCIe 链路工作、驱动程序、IP 布局和互联等细节都不是问题，使我们就像设计人员一样只需集中精力开发目标应用。

我们在水印应用中所做的优化适用于使用 SDAccel 编译过的所有应用。事实上视频水印应用就是一个很棒的技术讲解案例，详细介绍了赛灵思 SDAccel 中推出的优化方法。

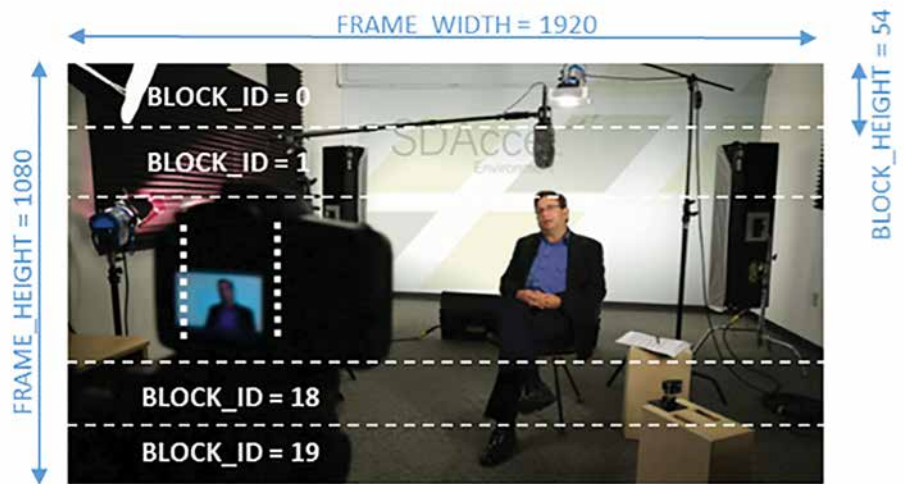


图7 - 把视频帧分区成数据块

Xilinx发布Vivado 2015.1版本加速系统验证

新版套件包括 Vivado 实验室版本、加速仿真流程、交互式 CDC 分析和并提供先进的 SDK 系统性能分析

2015年5月5日，中国北京 - All Program-mable 技术和器件的全球领先企业赛灵思公司（Xilinx, Inc. (NASDAQ:XLNX)）今天宣布推出可加速系统验证的 Vivado® 设计套件 2015.1 版。该版本具备多项可加速全可编程 FPGA 和 SoC 开发及部署的主要先进功能。全新版本包含 Vivado 实验室版本（Vivado Lab Edition）、加速的 Vivado 仿真器与第三方仿真流程、交互式跨时钟域（CDC）分析以及赛灵思软件开发套件（SDK）提供的系统性能分析。

全新的 Vivado 实验室版本

Vivado 实验室版本是一款免费的轻量级 Vivado 设计套件的编程与调试版本。该实验室版本包含 Vivado 器件编程器、Vivado 逻辑与串行 I/O 分析器以及内存调试工具，专门针对无需全功能 Vivado 设计套件的实验室环境。Vivado 实验室版本是比全功能 Vivado 设计套件小 75% 的简易版，大幅缩短了实验室的设置时间，并减少了系统内存需求。对需要通过以太网进行远程调试或编程的设计团队来说，Vivado 设计套件 2015.1 版还提供了独立的硬件服务器，其大小还不到完整版 Vivado 设计版本的 1%。

Vivado 仿真器及第三方仿真流程

Vivado 设计套件 2015.1 版还提升了仿真流程，可将 LogiCORE™ IP 编译时间缩短 2 倍以上，让整体仿真性能比此前版本快 20%。新版本还全面集成了赛灵思联盟计划成员 Aldec、Cadence Design Systems、Mentor Graphics 和新思科技（Synopsys）所提供的仿真流程。

Aldec 公司 CEO Stanley Hyduke 博士表示：“利用赛灵思 Vivado 工具指令语言（Tcl）库的基础架构，Aldec 现在能在 Vivado 设计套件中全面集成 Riviera-PRO 和 Active-HDL。这种独特的集成功能将给客户带来极佳的简便易用性优势。”

交互式跨时钟域分析

赛灵思还提供交互式 CDC 分析功能，进一步扩展了其先进的验证功能组合。该功能支持设计人员在设计早期阶段调试 CDC 问题，从而减少了昂贵的系统内调试周期，提升了生产力。结合 Vivado 设计套件的交互式时序分析和交叉探测特性，CDC 分析功能可提供强大的时序分析和调试功能，并加速产品上市进程。

赛灵思 SDK 加入先进的系统内性能分析与验证功能

为加速开发 Zynq®-7000 全可编程 SoC，赛灵思针对裸机和 Linux 应用扩展了其系统性能分析工具套件。赛灵思 SDK 现使嵌入式软件开发人员能够分析其 SoC 设计的性能和带宽，包括处理器子系统（PS）的关键性能参数，以及 PS、可编程逻辑（PL）和外部存储器之间的带宽分析。采用 AXI 流量生成器的系统建模设计则可用于 Zynq-7000 全可编程 SoC ZC702 和 ZC706 评估板。

全面掌控频域

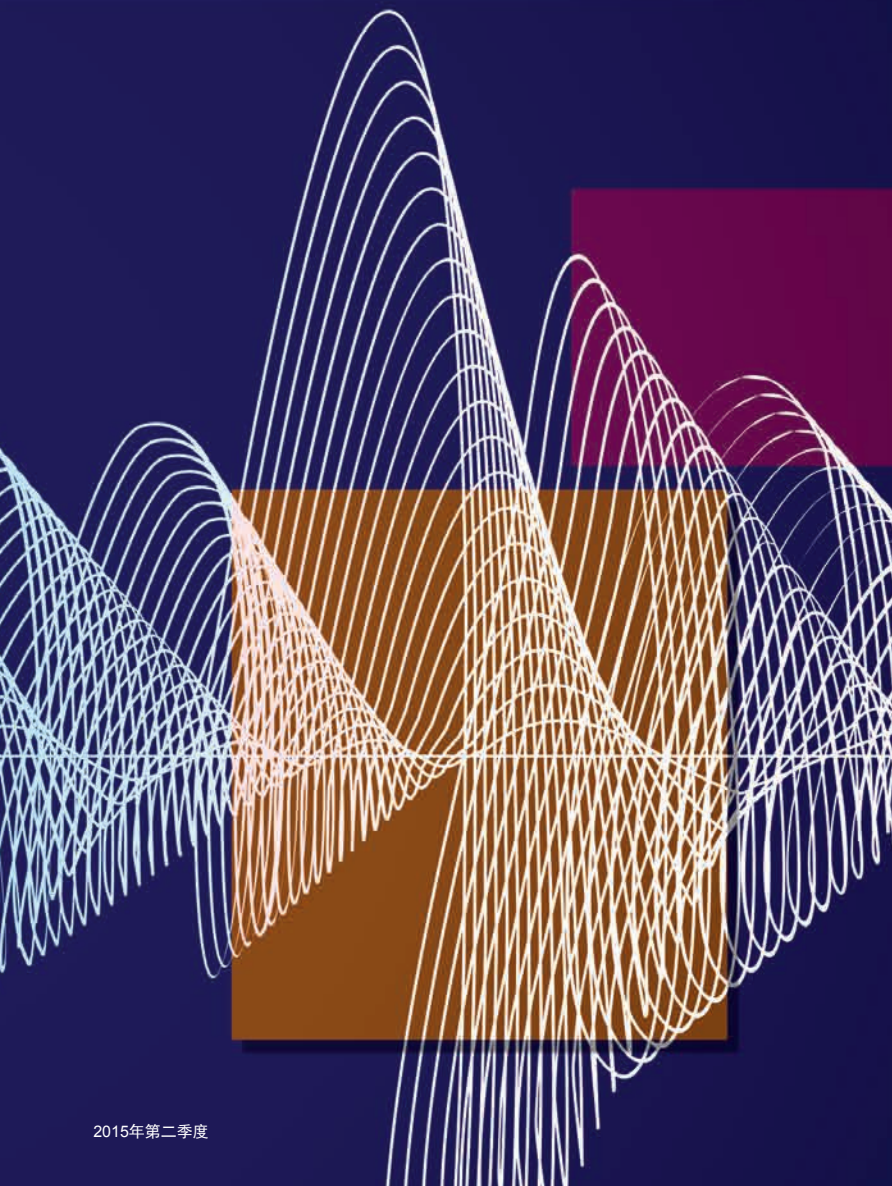
作者：

Adam P. Taylor

e2v公司首席工程师

aptaylor@theiet.org

许多应用都要求能够在频域内开展工作。本文将介绍如何处理 FPGA 设计中的频域问题。



对许多工程师而言，在频域中开展工作不如在时域中开展工作那么自然，可能是因为频率与复杂的数学运算有关。但是要充分发挥赛灵思 FPGA 解决方案的潜力，您需要在这两个域中自由切换。

令人欣慰的是详细地掌握频域并不像您最初想象的那样令人生畏。不管是您自己设计的定制模块还是市场上现有的 IP 模块，都能帮助您轻松转入转出频域。同时在频域中实现高速处理的方法也不乏其数。

时域还是频域？

工程师既能在时域中检验和处理信号，根据时间分析信号，也能在频域中检验和处理信号，根据频率分析信号。项目对工程师的主要要求之一，就是应该知道什么时候应该开展哪种分析。

一般在电子系统中，需要考查的信号是一个不断变化的电压、电流或频率。它可以是传感器的输出，也可能是系统其他部分生成的输出。在时域中，您可以测量信号的幅度、频率和周期，以及信号上升或下降时间等更有意义的参数。实验室环境中观察时域信号常用的是示波器或逻辑分析仪。

但是信号的一些参数体现在频域内。必须在频率中分析这些参数，才能解读其中包含的信息。在频域中可以识别信号的频率分量、各自的幅度和每种频率的相位。由于在

根据信号的类型，如重复性还是非重复性、离散还是非离散，有多种方法可用于时域和频域之间的变换

频域中便于开展卷积运算，因此在频域中工作也能大大简化信号处理。卷积是一种将两种信号混合成第三种信号的数学方法。和时域分析一样，如果想要在实验室环境中观察频域信号，可以使用频谱分析仪。

对某些应用而言在时域中工作比较合适，例如用于监测更大型系统的电压或温度的系统。虽然噪声可能是个问题，在许多情况下取一定数量样本的平均值就可以满足要求。但对其他应用来说，最好是在频域中开展工作。例如需要从另一信号中过滤某种信号的信号处理应用，或是需要将信号与噪声源分离的信号处理应用，最

好在频率中开展分析。

由于采样是在时域中完成的，在时域中工作基本不需要对量化数字信号进行后处理。与此相反，在频域中开展工作首先要求应用将量化数据从时域变换到频域。同样，从频域输出后处理过的数据时，需要从频率转回到时域。

如何进行变换？

根据信号的类型，如重复性还是非重复性、离散还是非离散，有多种方法可用于时域和频域之间的变换，包括傅里叶级数、傅里叶变换和 Z 变换。尤其是在电子信号处理和 FPGA

应用中，最常用到的变换是离散傅里叶变换（DFT），傅里叶变换中的一种。工程师使用 DFT 分析具有周期性和离散性的信号，即由一定数量均匀分布在采样频率中的 n 位样本组成的信号。在许多应用中这种信号一般来自系统内的 ADC。

简单地说，DFT 的工作原理就是把输入信号分解成代表信号正弦分量和余弦分量的两个输出信号。因此对由 N 个样本组成的时域序列，DFT 会返回两组 $N/2+1$ 个余弦波样本和正弦波样本，分别被称为实分量和虚分量（图 1）。实样本和虚样本的宽度为 n 位输入信号宽度的 $n/2$ 。

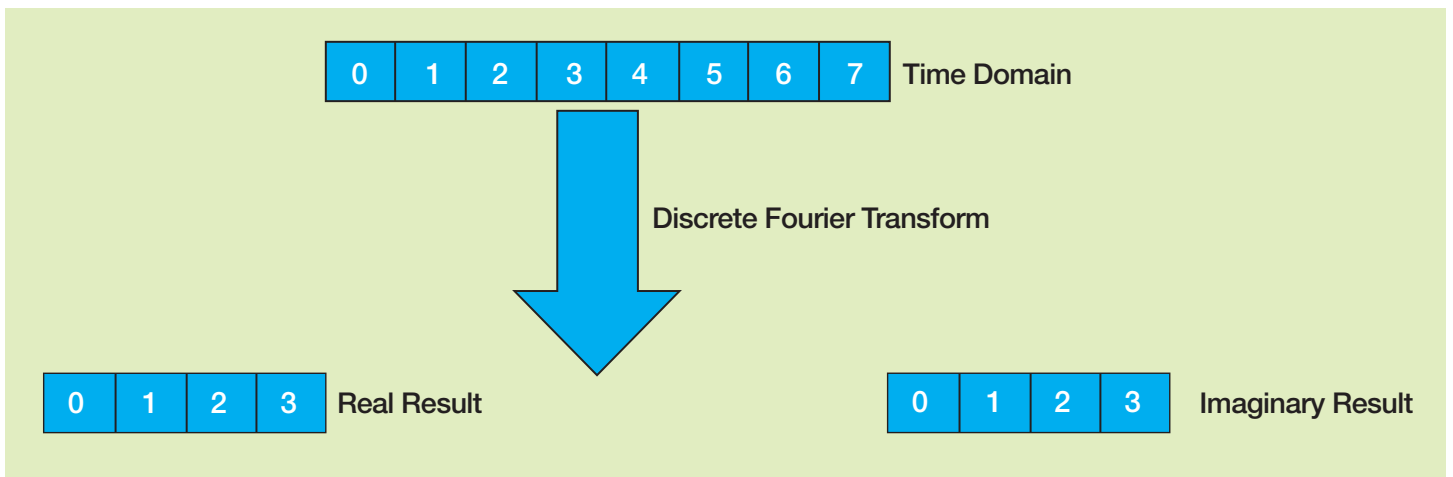


图1 - 时域中的n位被变换为频域中n/2实位和n/2虚位

计算 DFT 的算法十分简明，如下式所示：

$$\begin{aligned} \text{Re}X[k] &= \sum_{i=0}^{N-1} x[i] \cos\left(\frac{2\pi ki}{N}\right) \\ \text{Im}X[k] &= -\sum_{i=0}^{N-1} x[i] \sin\left(\frac{2\pi ki}{N}\right) \end{aligned}$$

其中， $x[i]$ 代表时域信号； i 的取值为 0 到 $N-1$ ； k 的取值为 0 到 $N/2$ 。这种方法被称为关联法，其功能是将输入信号与对应迭代的正弦波或余弦波相乘，以确定其幅度。

当然，您可能想要在应用中的某点从频域变换回时域。为此可以使用综合公式，它将实波形和虚波形结合起来，重建时域信号，即：

$$x[i] = \sum_{k=0}^{N/2} \text{Re}\bar{X}[k] \cos\left(\frac{2\pi ki}{N}\right) + \sum_{k=0}^{N/2} \text{Im}\bar{X}[k] \sin\left(\frac{2\pi ki}{N}\right)$$

但 $\text{Re}X$ 和 $\text{Im}X$ 是正弦波和余弦波的缩放结果，因此需要进行缩放。除 $\text{Re}X[0]$ 和 $\text{Re}X[N/2]$ 之外的所有情况， $\text{Re}X$ 和 $\text{Im}X$ 的值均为 $\text{Re}X[k]$ 或 $\text{Im}X[k]$ 除以 $N/2$ 。对 $\text{Re}X[0]$ 和 $\text{Re}X[N/2]$ ，则除以 N 。出于明显的理由，这种运算被称为傅里叶反变换 (IDFT)。

通过探讨确定 DFT 和 IDFT 所使用的算法，可能有助于了解如何加以利用这些算法。

您可以使用 Octave、MATLAB[®] 乃至 Excel 对所采集的数据进行 DFT 计算。许多实验室工具，如示波器，也能根据需要执行 DFT 计算。

不过需要指出的是上述的 DFT 和 IDFT 被称为实 DFT 和实 IDFT，因为输入的是实数而非复数。很快就会清

楚为何要了解这一点。

需要进行变换的地方

从电信到图像处理、雷达和声纳，难以想象有比傅里叶变换更强大、更适合并可实现 FPGA 中的分析技术。实际上 DFT 是最常用 FPGA 应用之一（即，生产有限输入响应 (FIR) 滤波器系数）的基础（参阅[赛灵思中国通讯第 43 期](#)《数字滤波器设计与实现详情》）。

但其用途不局限于滤波。DFT 和 IDFT 还可用于电信处理中，用于执行电信通道的信道化和重组。在频谱监测应用中，它们可用来判断监测带宽内的频率构成。在图像处理中，它们用于处理图像卷积，供滤波器内核开展图像模式识别等操作。所有这些应用在实现过程中一般使用比上文介绍的更有效的算法来计算 DFT。

总而言之，了解 DFT 并将其实现在 FPGA 中是每个 FPGA 开发人员的必备技能。

基于FPGA的实现方式

上文介绍的 DFT 和 IDFT 的实现一般采用嵌套循环，每个循环完成 N 次计算。因此实现 DFT 计算所需的时间为：

$$DFT_{time} = N * N * K_{dft}$$

其中， K_{dft} 表示完成每次迭代的处理时间。显然这种方法实现起来极为耗时。为此 FPGA 中实现 DFT 一般使用快速傅里叶变换算法 (FFT)。FFT 常常被称为我们时代最重要的算法，因为它对许多行业都产生根本性

的影响。

FFT 与 DFT 算法稍有不同，它计算的是复数 DFT，即它需要实时域信号和虚时域信号，得到的结果的宽度是 n 位而非 $n/2$ 位。这意味着如果需要计算实 DFT，必须首先把虚部设为 0，然后把时域信号转移到实部。如果要在赛灵思 FPGA 实现 FFT，可以有两种选择。您可以使用您选择的 HDL 重新编写 FFT，也可以使用 Vivado[®] 设计套件 IP Catalog 或其他来源提供的 FFT IP。除非有不得已的原因不能使用 IP，为缩短开发时间，一般都应该选择 IP。

FFT 的基本方法是将时域信号分解为一系列单点时域信号。因为样本被重新排序，这个过程常被称为位反转。如果没有位反转算法这一捷径可走，那么创建单点时域信号的级数可以用 $\text{Log}_2 N$ 计算得到，其中， N 表示位数。

随后使用这些单点时域信号计算每个点的频谱。这一计算相当简单，因为频谱与单点的时域相等。

这些单频点的重组是 FFT 算法最复杂环节。必须每一级重组一次这些频点，恰好是时域分解的反向操作。因此要重建频谱需要 $\text{Log}_2 N$ 级，这里就是著名的蝶形 FFT 发挥作用的地方。

与 DFT 的执行时间相比，FFT 用时为：

$$FFT_{time} = K_{fft} * N \text{Log}_2 N$$

与计算 DFT 相比，执行时间明显缩短。

在 FPGA 中实现 FFT 算法时，还

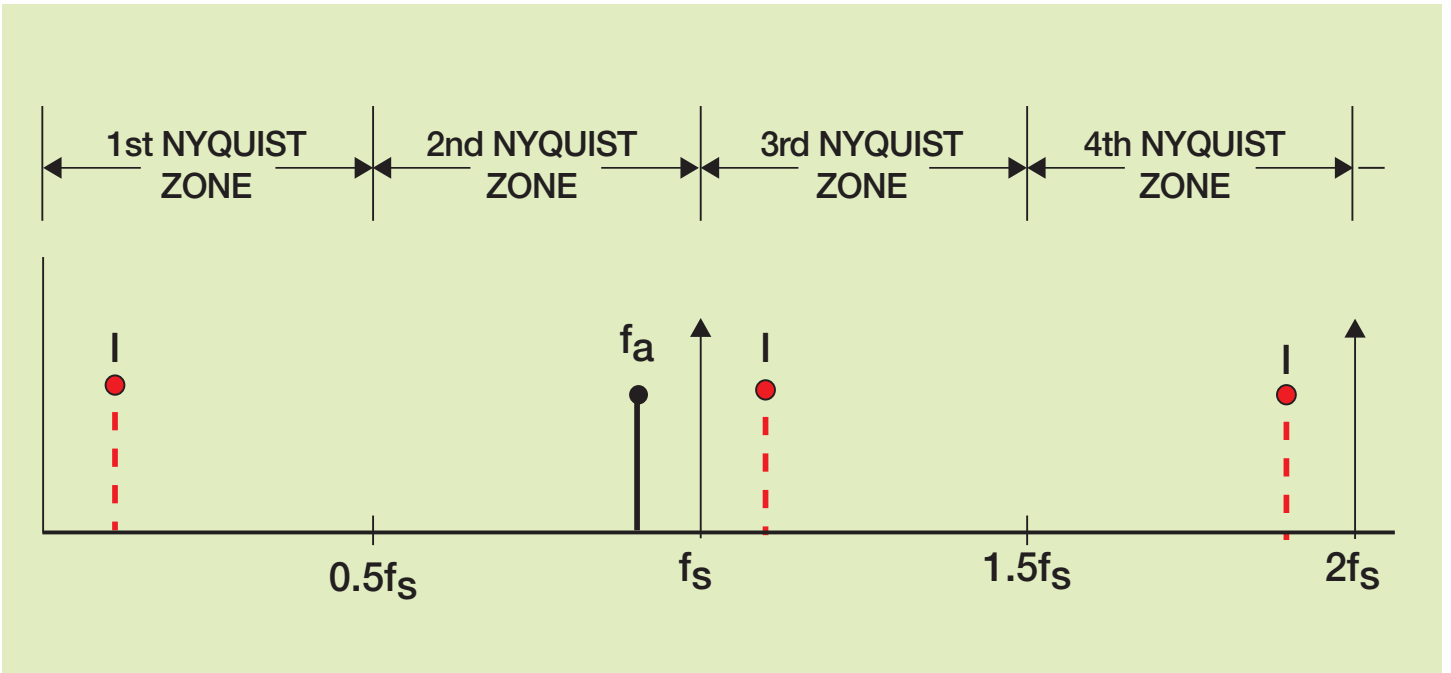


图2 - 奈奎斯特域与混叠

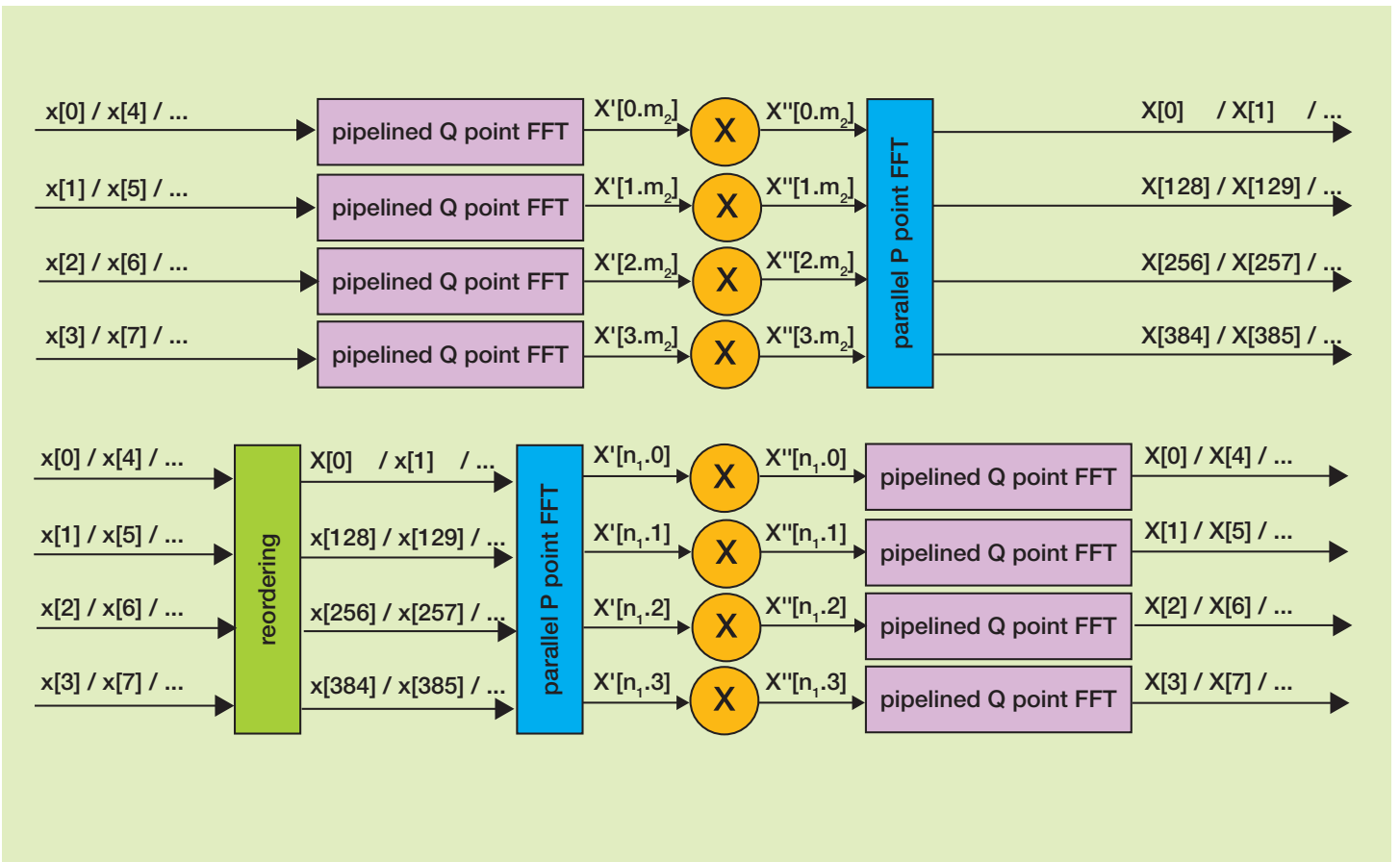


图3 - 分解与重组FFT结构

必须考虑 FFT 采样数。FFT 采样数将决定本底噪声，在此之下将无法再看到有用信号。FFT 采样数还将决定频点之间的间距。使用下面的公式可以确定 FFT 采样数：

$$FFTNoise\ Floor\ (dB) = 6.02n + 1.77 + 10\log_{10}\left(\frac{FFTsize}{2}\right)$$

其中，n 为时域中量化位的数量，FFTSize 为 FFT 采样数。对基于 FPGA 的设计，这一般是 2 的幂，如 256、512、1,024 等。频点可采用下列公式计算出其等间隔：

$$Bin\ Width = \frac{Fs}{FFTsize}$$

举个非常简单的例子，在采样频率 (FS) 为 100 MHz 的情况下，使用 128 个 FFT 样本可实现 0.39 Hz 的频率精度。这意味着如果频率彼此之间的间隔小于 0.39 Hz 将无法分辨。

高速采样

许多 FPGA 中的 FFT 应用和较高性能的系统都工作在非常高的频率下。高频运行会产生自己特有的设计难题。

因为在高频下无法维持奈奎斯特采样率（每个周期两个样本），因此需要不同的方法。例如使用 2.5 GHz 采样率采样 3 GHz 全功率带宽模拟输入。运用奈奎斯特频率法则，高于 1.25 GHz 的信号将被混叠到有用的第一个奈奎斯特域中。这些混叠镜像将成为基本信号的谐波分量，因此包含与非混叠信号相同的信息，如图 2 所示。

要判断得到的谐波或谐波成分的频率布局，可使用下列算法：

$$\begin{aligned} F_{harm} &= N \times F_{fund} \\ IF(F_{harm} = \text{Odd Nyquist Zone}) \\ F_{loc} &= F_{harm} \text{ Mod } F_{fund} \\ Else \\ F_{loc} &= F_{fund} - (F_{harm} \text{ Mod } F_{fund}) \\ End \end{aligned}$$

其中，N 表示有用谐波的整数。

继续前述例子，如果采样率为 2,500 MHz，基频为 1,807 MHz，则会在第一奈奎斯特域中产生 693 MHz 的谐波成分，供在 FFT 中进一步处理。

在掌握了频谱的基本知识之后，下一个需要考虑的关键因素是如何将这些 ADC 和 DAC 器件与 FPGA 连接在一起。在上面所述的实例中，无法从 ADC 接收频率为采样频率一半（采样率为 2.5 Gbps）的数据。为此，高性能数据转换器使用多路复用的数字输入输出，该输入输出工作在较低频率下，一般为转换器的采样频率的四分之一或二分之一（即 FS/4 或 FS/2）。

FPGA 通过多个数据流接收到数据后，如果希望执行 DFT，那么下一个要解决的问题就是如何在 FPGA 内处理数据。包括电信和射电天文在内的多种应用常用的方法是使用重组或分解 FFT 结构（如图 3 所示）。

虽然这种应用与直接 FFT 相比更复杂，使用这种方法能够实现高速处理。

正您所见，在频域中工作并非像最初想象的那样困难，特别是有 IP 模块帮助转入转出频域的情况下。此外还有多种方法可供选择，帮助您实现高速处理。

Xilinx SDAccel和SDNet双双荣膺Lightwave创新奖

光网络行业高度肯定赛灵思为系统工程师和软件工程师所提供的出色的技术支持，帮助他们充分利用全可编程逻辑和 SoC 器件的优势

2015 年 3 月 31 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)) 今天宣布其 SDAccel™ 和 SDNet™ 软件定义开发环境凭借其出色的技术成就双双荣膺 2015 年 Lightwave 创新奖。Lightwave 创新奖旨在表彰光网络行业中的顶尖产品和解决方案，所有奖项均由第三方光纤网络专家组成的独立评审团所选出。

凭借卓越的技术成就，赛灵思 SDAccel 和 SDNet 环境在设计工具类奖项中脱颖而出。所有产品都进行五分制打分，得分 4 分以上表明评委认为该产品提供的技术特性和性能具有明显本质上的优势。赛灵思每个产品的得分均达到了 4.5 分。

Lightwave 发行人 Tim Hermes 表示：“获得 4.5 分的产品和技术就表示巨大的成功，显然今年的赛灵思赢得了这样的成功。将硬件复杂性抽象出来，帮助软件工程师充分利用可编程技术的优势，这是实现系统开发和差异化设计的关键因素，评委们一致认为这两种开发环境颠覆了传统设计方法。”

赛灵思公司有线和数据中心业务部副总裁 Hemant Dhulla 指出：“得到众多网络领域专家的好评，这真是可喜可贺。这两款设计环境是专门为满足系统和软件工程师设计网络系统和加速数据中心应用的编程要求而全新开发的。我们为这两款产品感到骄傲，他们让更多领域的工程师可以跟容易提早运用和体验 FPGA 的众多优势，同时也为下一代系统设计带来更多的可能性。”

赛灵思于 2014 年 3 月推出面向网络的软件定义规范环境 (SDNet)，也是赛灵思 SDx 系列的首个成员。借助该环境，系统设计师和软件架构师能够通过高级用户自定义规范和高度优化的 FPGA 技术轻松地创建高性能包处理系统。

赛灵思于 2014 年 11 月再为 SDx™ 系列推出第二款产品，那就是用于加速的 SDAccel 软件定义开发环境的。采用该环境，系统工程师和软件工程师能通过 FPGA 为数据中心创建出功耗优化的、可重配置的加速计算环境。SDAccel 可在通用开发环境中将针对 OpenCL™、C 和 C++ 的编程环境、优化编译技术，以及动态可重配置的加速器完美结合在一起。而赛灵思 SDx 系列的第三大成员 SDSoC 于 2015 年 3 月初宣布推出，可支持更广泛的嵌入式软件及系统设计人员充分利用 Zynq 全可编程 SoC 和 MPSoC 的优势。

关于 SDx

SDx (china.xilinx.com/sdx) 是专门面向系统和软件工程师而推出的一系列软件定义开发环境。SDx 让具备很少甚至没有 FPGA 专业知识的开发人员，也能通过运用高级编程语言受益于带有业界标准处理器（无论该处理器是位于开发板上，还是嵌入在 Zynq 全可编程 SoC 和 MPSoC 上）的可编程硬件的强大功能。

关于 LIGHTWAVE

30 多年来，Lightwave (www.lightwaveonline.com) 一直为全球光学通信领域的高级决策者提供值得信赖的技术、应用及商业信息与洞见。Lightwave 竭诚服务于技术厂商、通信运营商和大型企业，为他们提供全面具体的光通信商业环境情况。Lightwave 是重要的信息来源，能提供 FTTx、网络、设备设计、MSO 光学器件以及测试测量领域的相关信息，可满足企业高管、开发负责人、项目经理、网络工程师以及其他战略决策者的需求，这些信息对确保企业成功至关重要。Lightwave 隶属于 PennWell 公司旗下。

将SoC平台设计与 DSP系统生成器相 集成

作者：

Daniel E. Michek,

赛灵思公司系统级产品营销高级经理，
daniel.michek@xilinx.com

Vivado 系统生成器
工具能方便地接入平
台设计，从而可充分
利用开发板接口和处
理系统。

FPGA 的应用不断拓展，同时 FPGA 设计流程也随之不断演进。我们不再将 FPGA 用作简单的胶合逻辑，甚至不再作为信号处理链的核心，用以将 IP 与专有后端接口集成。相反，FPGA 正在转变为可编程片上系统，其中包含作为处理器外设的硬件以及在强大 APU 上运行的高级软件。这种架构就是我们所说的赛灵思 All Programmable SoC。

为了充分发挥这种全新流程的优势，我们需要将设计方法从 FPGA 早期的自上而下 RTL 转变为以 IP 开发和标准化连接（例如 ARM® 的高级可扩展接口（AXI））为中心的自下而上流程。随着接口从定制接口发展为通用接口，我们就可以花更少的精力来验证数据路径与平台设计之间的交互。

赛灵思 DSP 系统生成器（System Generator for DSP）同样也得到了发展。该工具是 Vivado® 设计套件的一部分，其通过将 DSP 数据路径整合到用 Vivado IP Integrator 工具构建的

平台设计中，打造出全新的自下而上设计方法。我们仔细看看系统生成器（System Generator）实现的设计自动化如何帮助高性能设计充分利用平台连接功能。

构建 All Programmable 平台框架

通过定义用来存放数据路径的平台框架，我们可以启动全新的设计流程。Vivado 工具套件能识别开发板；我们将利用开发板自动化功能构建新的平台设计。

如图 1 所示，平台设计或平台框架的基本组成包括：处理器级和开发板级接口，以及将其组合在一起的逻辑。我们将平台框架作为系统级设计的基础（即外壳），这给我们的数据路径提供了空间。模块和连接功能自动化通过 IP 外设将处理系统连接到板级接口。封装在 IP Catalog 中的 DSP 数据路径或软件加速器利用赛灵思 Designer Assistance 自动化功能方便地连接到我们的处理器平台框架中，进而连接到外部器件接口。

创建作为可输入 IP 的数据路径

我们的最终目标是让数据路径能进入 All Programmable 平台框架。如果要从头开始，我们可以用标准化接口创建数据路径。如图 2 所示，只需将网关端口迅速标记为 AXI4-Lite 接口，或者对端口进行命名以匹配标准连接（如 Simulink® 图上的 AXI4-Stream），这样，System Generator 为 Vivado IP Catalog 打包设计时就会为设计添加额外逻辑并将通用信号收集到接口内。

不过，我们利用一种新设计方法，使用平台框架定制一个可集成到 All Programmable 设计的插件。我们使用自动化功能确定平台设计中现有哪些接口，哪些接口与开发板关联，以及哪些接口可为 DSP 数据路径创建插件。既然我们的目标是将数据路径转换为可连接至平台框架的 IP，因此我们不需要关注板级接口，而是关注标准化 AXI 接口。开发板上的每个不关联接口转变为 System Generator 网关。这些网关在 System Generator 中

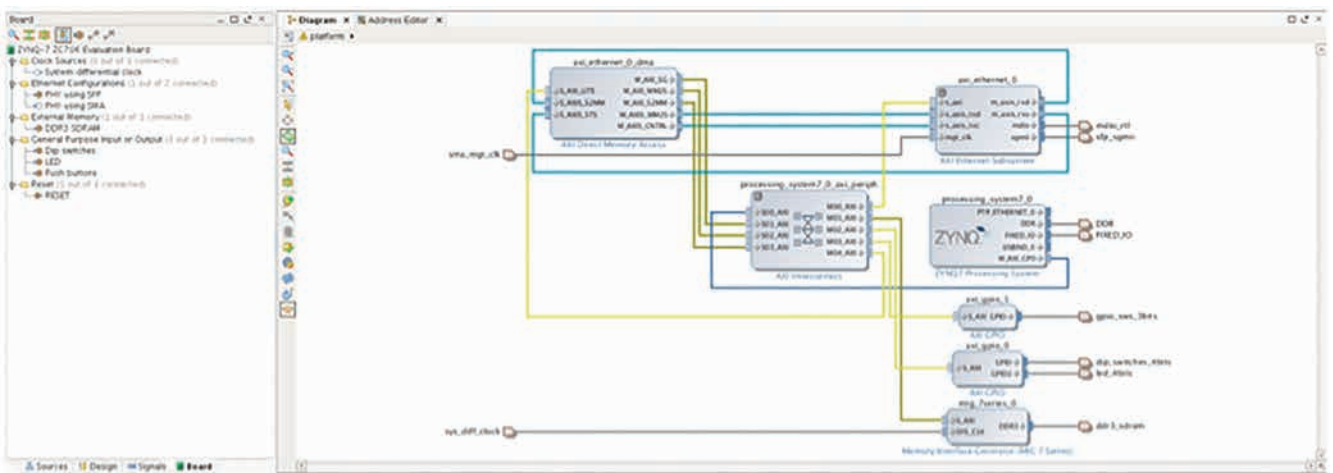


图1—将处理系统连接到板级接口的平台框架实例

充当简单信号，当我们将平台设计导出到 IP Catalog 时，这些网关会生成用以连接平台设计的 AXI 接口。

如实例所示，AXI4-Lite 接口创建独立的读 / 写信号，在导出到 Vivado 工具套件时共享通用的可寻

址寄存器接口。简单的复制 - 粘贴使我们在地址偏移位置通过相同接口获得更多用于处理器的直接寄存器。同

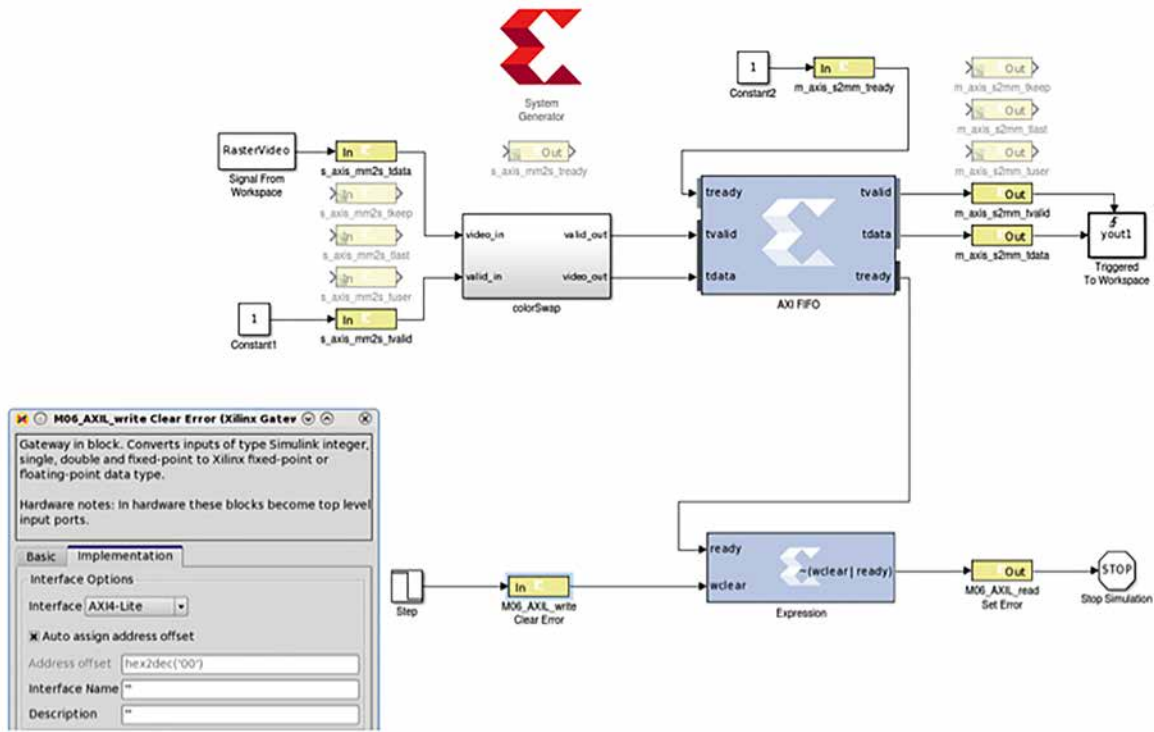


图2 - 自动将网关导入AXI4-Lite和AXI4-Stream接口

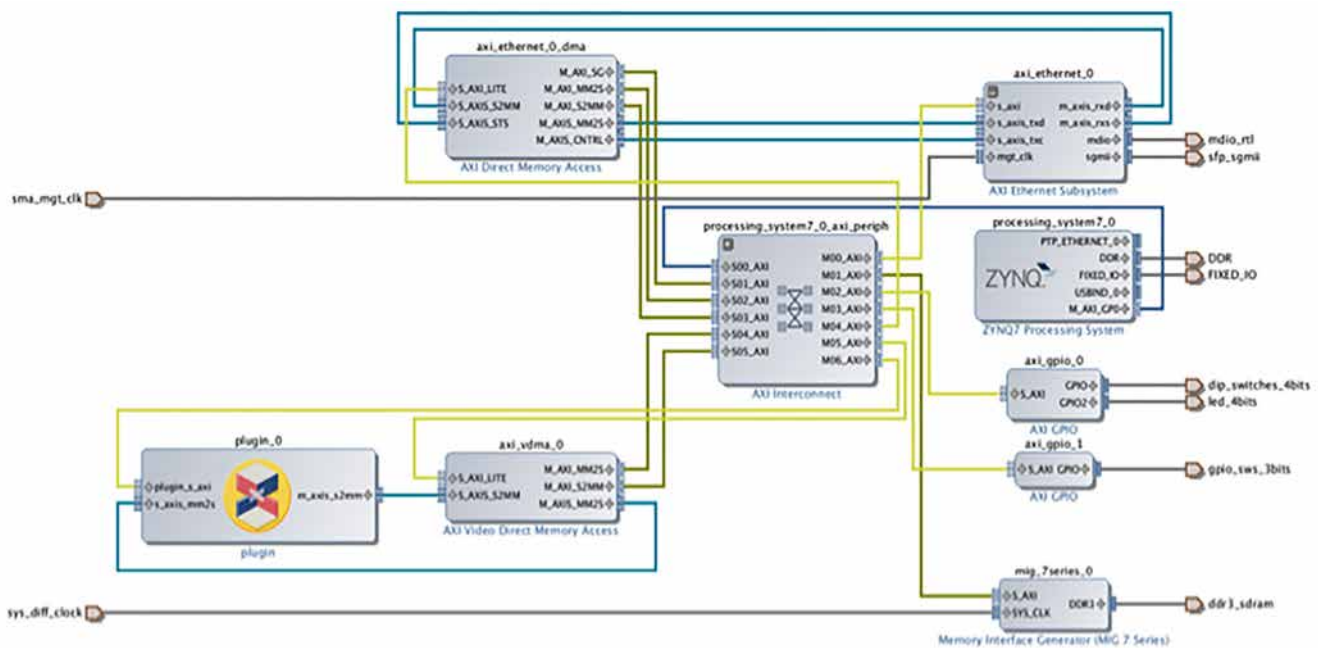


图3 - 将DSP数据路径连接到平台框架的平台系统

时，我们可自动生成软件驱动 API 以读/写寄存器。

如果平台设计中提供 AXI4-Stream 接口，System Generator 会向模型添加正确匹配的网关。AXI4-Stream 接口极其灵活并包含很多信号。ACLK 时钟源与该接口相关联，但该信号直接作为这部分数据路径的抽象系统时钟。TVALID 信号表示接口有效。其他信号为可选信号。System Generator 会将初始流接口中的信号添加到我们的模型，不过我们可以通过删除或添加信号来满足内部要求。

在图 2 所示的模型中，我们的数据路径只关心 TDATA（经接口发送的数据）和 S_AXIS 接口上的 TVALID。为删除不必要的信号，我们针对该模型对没用的网关添加注释，因为默认值将会驱动 IP Integrator 中的信号连接。

AXI4-Lite 和 AXI4-Stream 信号能利用自下而上方法轻松进行仿真和验证。AXI4-Lite 接口建模为用于访问大量 Simulink 模块的简单网关，实现网关一侧到另一侧数据传送的抽象。同样，AXI4-Stream 接口只是一系列遵循简单握手规则的信号，以将数据从一个 IP 核传递到另一个。

仿真建模的唯一挑战在于接口上所用的可选端口。如果我们每个周期都接收数据并通过数据路径无中断地进行处理，我们就不需要 TREADY 握手信号。简化模型通过 TDATA 网关将矢量的每个元素从 Simulink 的信号发送到 Workspace 令牌。当需要完整握手时，我们用 AXI4-Stream FIFO 进行建模以缓冲数据，如图 2 中的 M_AXIS 接口所示。

这种定制的自动化功能可作为创建连接平台框架的 IP 的起点。不过，

System Generator 很灵活，允许我们添加或删除部分甚至整个 AXI 接口。最终可将数据路径转换为可在多个系统级设计中重用的 IP。

添加逻辑后，最后一步是将使用 DSP 系统生成器构建的数据路径导出到 Vivado IP Catalog。这种操作能实现接口的简单连接，无论使用 RTL 还是在 IP Integrator 中都是如此。此外，我们还生成在 SDK 中使用的驱动器，并将具有最佳测试向量数据的仿真模型连接到 IP。由于我们在创建 DSP 数据路径时事先知道平台框架的情况，因此能自动组合模型和平台设计，如图 3 中的完整系统所示。

降低仿真风险

生成包含硬件加速器、DSP 数据路径或定制逻辑的完整片上系统很具有挑战性。采用自下而上方式进行仿真以确定数据路径按预期工作，这种方法风险很大，而且维持可支持数据路径的平台接口带宽同样很困难。

我们通过使用标准化接口开发能降低仿真风险的 IP。这是因为接口级的交互被抽象化，使我们专注于验证内部数据路径。最后，通过利用开发板、模块和连接的智能自动化功能，我们可生成基于平台的系统，从而满足我们的需求并集成自定义数据路径。

如需了解有关 DSP 系统生成器的更多详情，敬请访问 china.xilinx.com/products/design-tools/vivado/integration/sysgen.html。如果您有任何问题或意见，请致电电话 (858) 207-5213，或发送电子邮件至 daniel.michek@xilinx.com，联系作者 Daniel Michek。

Xilinx UltraScale 20nm 器件助力打造 JDSU ONT 400G 以太网 测试平台

最新 JDSU 测试平台采用赛灵思 UltraScale 20nm FPGA，加速了 400G 产品的开发、验证与互操作性

2015 年 3 月 35 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)) 今天宣布其 Virtex® UltraScale™ 20nm FPGA 已应用于 JDSU ONT 400G 以太网测试平台。该新平台可提供 400G 带宽，并确保真正基于数据包到数据包的精确分析，从而可以满足各种先进的 400G 应用的需求和复杂性。赛灵思 Virtex® UltraScale™ VU095 器件是 ONT 400G 以太网测试平台的关键组件，能提供前所未有的高性能、高系统集成度和高带宽，实现通过单一路径架构将芯片到芯片之间的损耗降到最低。

JDSU 公司产品线经理 Paul Brooks 表示：“我们的 ONT 400G 解决方案采用了 Virtex UltraScale 器件，这使得我们能够提供加速 400G 产品开发、验证和互操作所需的强大功能和灵活性。”

JDSU ONT 400G 以太网测试平台采用 JDSU 公司的 ONT 架构，该架构的理念是使用更少的设备实现全面的模块化压力测试。这一业界领先的技术已被用来支持和验证各种先进的 400G 转发器和 400G 可插拔模块的复杂结构与要求。

赛灵思公司测试测量与评估细分市场总监 Hanneke Krekels 指出：“赛灵思 UltraScale FPGA 不断帮助像 JDSU 这样的行业领先者交付新一代测试解决方案。凭借这一最新平台，JDSU 可以满足 400G 以太网早期试用客户的需求，并进一步加快该生态系统的发展。”

该解决方案曾于 2015 年 3 月 24 至 26 日在洛杉矶举行的 2015 美国光纤通讯展览会及研讨会 (OFC) 上进行过展示。如需观看演示视频，敬请访问：<http://v.youku.com/vshow/idXOTMwMDA3NDE2.html>。

重新思考快速宽频 ADC中的数字下变频

作者：Ian Beavers

Analog Devices公司技术专家

ian.beavers@analog.com

高性能 GSPS ADC 为基于赛灵思 FPGA 的设计解决方案带来板载 DDC 功能



宽带每秒数千兆个样本 (GSPS) 模数转换器 (ADC) 为高速采集系统带来众多性能优势。这些 ADC 在高采样率和输入带宽下提供较宽的可见频谱。然而，有些情况需要宽带前端，有些则要求能够滤波并调谐为较窄的频带。

当应用只需要较窄带时，用 ADC 采样、处理和传送宽带频谱本身就低效，而且还耗能。当数据链路占用赛灵思 FPGA 中的大量高速收发器，只为在后续处理中对宽带数据进行抽取和滤波时，就会产生不必要的系统负担。赛灵思 FPGA 收发器资源可以得到更好的分配，以接收所需的低带宽并疏导来自多个 ADC 的数据。可在 FPGA 的多相滤波器组信道器中针对频分复用 (FDM) 应用进行额外滤波。

高性能 GSPS ADC 现将数字下变频 (DDC) 功能在信号链中进一步提升，以使其位于基于赛灵思 FPGA 的设计解决方案的 ADC 之中。该方案为高速系统架构师提供了多种新的设计选择。然而，由于该功能对 ADC 来说相对比较陌生，因此工程师可能就 DDC 模块在 GSPS ADC 中的运行存在一些设计相关问题。让我们理清一些最常见的问题，以便设计人员能够更有信心地使用这种新技术。

为了充分获得DDC的性能优势，设计中还要包含滤波器-混频器组件以作为抽取的补充。

什么是抽取？

最简单的定义，抽取就是只观察 ADC 输出样本中具有周期性的子部分，而忽略其他部分。结果就是通过下采样来有效降低 ADC 采样率。例如，ADC 的 M 抽取模式只输出第 M 批样本中的第一个，舍弃之间的所有其他样本。对每个 M 的倍数，重复该方法。

样本抽取本身只能有效减小 ADC 采样率，并相应地作为低通滤波器。如果没有频率变换和数字滤波，抽取

只会在频域中将基波的谐波以及其他杂散信号相互叠加。

DDC的作用是什么？

既然抽取本身无法阻止频带外信号的叠加，那么 DDC 是如何做到的？

为了充分获得 DDC 的性能优势，设计必须包含滤波器 - 混频器组件作为抽取功能的补充。数字滤波能从狭义上的频带（由抽取比率设定）中有效消除带外噪声。DDC 的典型数

字滤波器实现方案是一个有限脉冲响应 (FIR) 滤波器。由于没有反馈，这种滤波器只与过去的输入有关。滤波器的通带应匹配抽取后的转换器有效频谱。

DDC滤波器应该多宽？

DDC 的抽取比率通常基于整数因数，即 2 的幂次方（2, 4, 8, 16 等）。不过，抽取比率实际上可以是基于 DDC 架构的任意比率，包括小数抽取。对

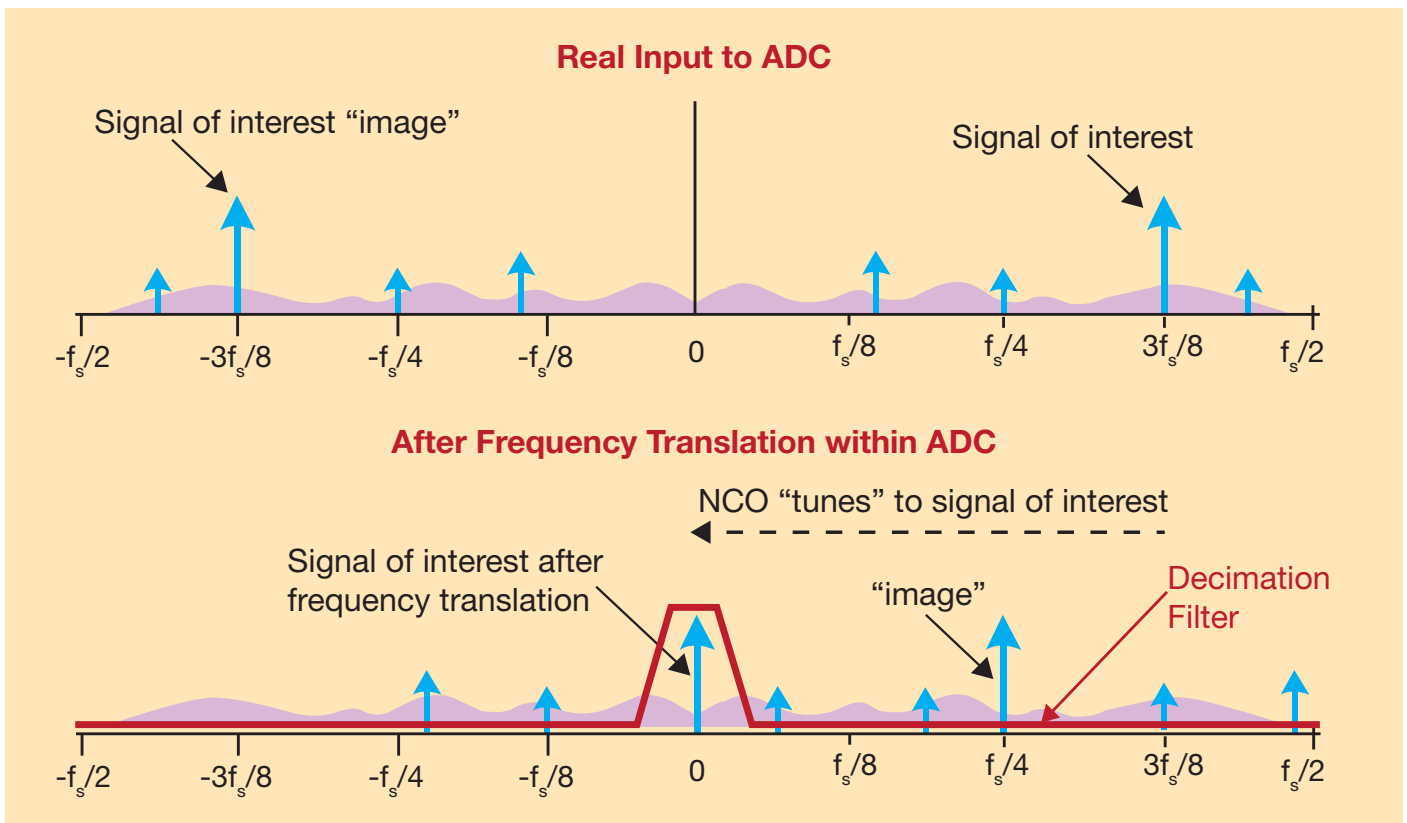


图1 - 采用低通滤波器和NCO的频率变换可在所需频率下有效实现带通滤波器。频率规划能确保不想要的谐波、尖刺和图像落在频带以外。

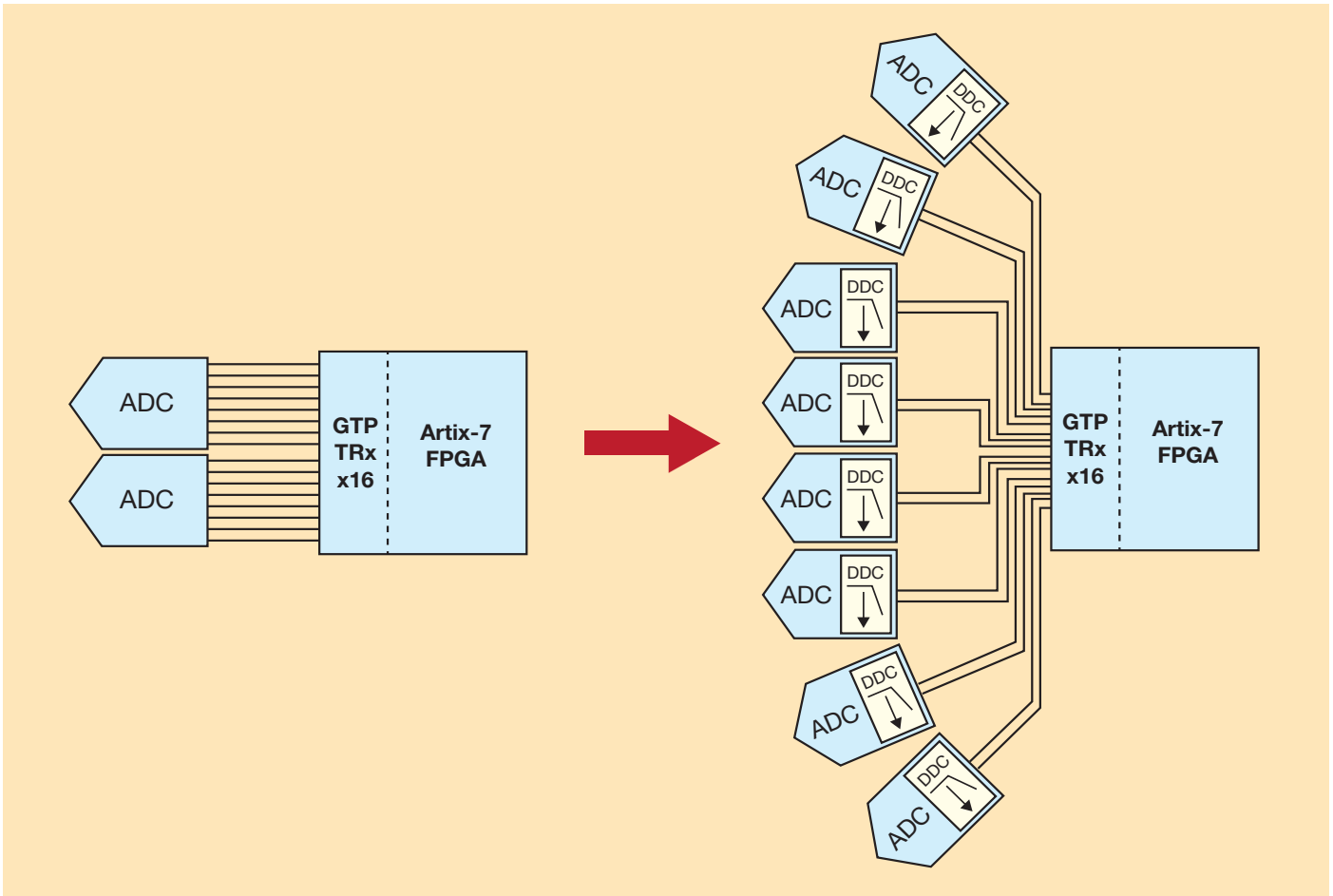


图2 – 抽取比率为8的DDC能让赛灵思Artix-7的16 GTP 6.6Gbps收发器支持八个ADC，每个都通过两条JESD204B通道传送抽取后的I/Q数据，反之只能支持两个ADC，每个通过八条通道输出完整带宽。

于小数抽取的情况，在抽取前通常需要一个插值计算模块来实现有理分数比率。

理想情况下，数字滤波器应准确匹配抽取频率带宽并滤去频带以外的一切干扰。然而，实际的有效滤波器带宽无法准确匹配抽取比率的整个带宽。因此，滤波器带宽将是抽取频率的一定百分比，例如 85% 或 90%。举例来说，抽取因数为 8 的滤波器的有用带宽实际上可能是采样率除以 10 或 $f_s/10$ 。DDC 滤波级必须具备较低的通带纹波和较强的阻带混叠抑制能力。

频率是固定的吗？

下一个问题是 DDC 滤波器的频率是固定的，还是能进行调谐并集中于某个所需的特定频带。

我们已经讨论了 DDC 的抽取和滤波级。不过，只有在所需频率处于从 DC 开始的滤波器通带之内时才有意义。如果不是这样，我们需要采取方法将滤波器调谐到不同的频谱部分以观察有用信号。可利用数控振荡器 (NCO) 在第一个或第二个奈奎斯特区域内调谐窄带。NCO 用来将滤波器频带调谐和混合到宽带频谱的不同部分 (图 1)。

数字控制字提供采样率的小数分频器，频率布置分辨率由数字控制字中所使用的位数来设定，可实现对有用频带的混合。控制字具备相应的调谐范围和分辨率，以便将滤波器放在所需的位置。典型的 NCO 控制字可能多达 48 位分辨率，跨越采样频率的两个奈奎斯特频带，这对大多数应用来说足够了。

NCO 带有一个混频器。该器件工作方式很像模拟正交混频器，可将 NCO 频率作为本地振荡器，以执行对真实、复杂输入信号的下变频。

滤波器紧跟在频率变换级后面。

单个8抽取DDC能够使赛灵思 Artix-7 FPGA系统可支持的ADC数量提高至四倍。

在所需的载波频带向下调谐到 DC 以后，滤波器就能有效降低采样率，同时能有效抑制在调谐后的有用带宽周围由临近的无用载波产生的混叠现象。

将输入信号通过混频降至基带时，由于过滤了负像，因而会出现 6 dB 的信号损失。NCO 还会额外产生一个小的插入损耗。因此，降频至基带后的输入信号总损耗通常略高于 6 dB。NCO 允许将输入频谱调谐至 DC，这样便可由后续的滤波器模块进行有效滤波，以防止混叠。DDC 还可能包含独立控制的数字增益级。增益级让系统实现 +6 dB 或更高的增益，以在输出位数的整个范围内集中信号的动态范围。

处理器间中断

采用 ADC 样本抽取后便无需向信号链下游发送最终会被舍弃的无用信息。由于这类数据被滤除，因此降低了 ADC 后端所需的输出数据带宽。这个减少量被 I/Q 输出数据量的增加所抵消。例如，具有 I 和 Q 数据的 16 抽取滤波器会使宽带输出数据减少 8 倍。

数据速率的最小化能减少 ADC 的 JESD204B 输出通道数量，进而降低系统布局的复杂性。ADC 输出带

宽的减小有助于设计小型化系统，否则这是无法实现的。例如，受系统功耗和尺寸的限制，电路板只能使用一个 FPGA，对于这种情况，系统所支持的高速串行收发器数量会在不使用 DDC 时限制 ADC 的数量。

当系统内只能观察到很窄的带宽时，ADC 内的抽取有助于消除这种局限性。使用单个 8 抽取 DDC 可将 ADC 的输出带宽减少至两个输出数据通道，以让赛灵思 Artix[®]-7 FPGA 系统支持的 ADC 数量提高至四倍。对于这种情况，我们利用 Artix-7 FPGA 中现有的 16 GTP 收发器设计出采用 DDC 的 8 ADC 结构（图 2）。这样能更高效使用赛灵思 FPGA 资源，成为一组 FDM 通道的多通道数字接收器。

DDC滤波器是否影响SNR和SFDR?

下一个需要研究的问题是当 DDC 滤波器打开和关闭时，信噪比 (SNR) 和无杂散动态范围 (SFDR) 这两个模拟性能如何变化。

由于转换器的宽带噪声被滤除而且只能观察到较窄的频谱，我们应该看到信号功率与观察到的噪声之比更高。ADC 的动态范围在滤波器的通带内应该更好。对宽带频谱进行抽取和滤波的固有优势在于利用 DDC 改进

SNR。

DDC 实现的数字滤波用来滤除较窄带以外的噪声。ADC 的 SNR 计算必须包含一个考虑被滤除噪声处理增益的滤波校正因子。使用完美数字滤波器，带宽每减小的 2 的幂次方，被滤除噪声引起的处理增益将会增加 +3 dB：

$$\text{理想 SNR (具有处理增益)} = 6.02 * N + 1.76 \text{ dB} + 10 \log_{10}(f_s / (2 * BW))$$

使用 DDC 的一个明显优势是能够使基波信号的谐波落在所需频带的外面。通过适当的频率规划，数字滤波可以防止谐波在窄 DDC 带宽内看到，从而提高系统的 SFDR 性能。

在只需要窄带的系统中，DDC 通过滤掉宽带噪声来提供 ADC 处理增益。这样能提高有用带宽内的信噪比。另一个优势是，通过合理的频率规划，通常占主导的第二和第三次谐波会落到调谐后的有用带宽之外并被数字滤除。这能提高系统的 SFDR。

采样定理指出谐波或其他高阶系统尖刺可能回折到每个奈奎斯特频带末尾的周围。这对于 DDC 同样适用，其第二或第三次无用谐波有可能回折到通带内并降低 SFDR。因此，为了研究这类采样问题，应该为 DDC

通带滤波器宽度和 NCO 调谐位置实施系统频率规划。

是否需要外部滤波器？

使用内部 DDC 的系统 ADC 也可以使用附加的模拟滤波器，就像没有 DDC 滤波时那样。对于宽带系统，DDC 能够缓和 ADC 前端的滤波要求。

DDC 中的数字滤波能分担一些工作，否则就要在前端放置严格的防止混叠模拟滤波器。不过，宽频带前端能实现多种用途，可让 DDC 同时观察多个频带，或者利用 NCO 扫描有用的频带以找到变化的输入信号。

ADC能否提供多个DDC？

工程师考虑用 FPGA 实现内部数字下变频时提出的最后一个问题是，ADC 是否只提供一个 DDC。答案是否定的；事实上可以观察到多个频带。

就 ADC 中的多个 DDC 而言，每个都有自己的 NCO，用来通过调谐将频带在奈奎斯特区域中分开。这种方法能同时观察多个频带，并减小系统 FPGA 收发器和抽取模块负荷，以将它们分配给其他处理活动，例如针对 FDM 系统将多个 ADC 实现信道化。

高速 ADC 现已具备足够的处理能力将 DDC 功能带入信号链。如果

系统不需要使用宽频带奈奎斯特率 ADC 的完整带宽，则可通过 DDC 滤除不想要的数据和噪声。这样能改善信号采集的 SNR 和 SFDR。较低的带宽能降低 FPGA 收发器（例如 Artix-7）的数据接口负担，并有助于设计更复杂的信号采集系统。🌈

TRACE32[®]

Debugging Xilinx's Zynq™ -7000 family with ARM® CoreSight™

- ▶ RTOS support, including Linux kernel and process debugging
- ▶ SMP/AMP multicore Cortex®-A9 MPCore™s debugging
- ▶ Up to 4 GByte realtime trace including PTM/ITM
- ▶ Profiling, performance and statistical analysis of Zynq™'s multicore Cortex®-A9 MPCore™



www.lauterbach.com

LAUTERBACH
DEVELOPMENT TOOLS

Vivado 2015.1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2015.1版本可从赛灵思下载中心下载：china.xilinx.com/download。

VIVADO 设计套件 2015.1 版本亮点

最新版 Vivado 设计套件包括最新 Vivado 实验室版本（Vivado Lab Edition）、互动时钟域交叉（CDC）分析、加速仿真流、赛灵思软件开发套件（SDK）中的高级系统性能分析，以及 XCVU440 等新型器件。

Vivado 实验室版本

最新 Vivado 实验室版本是 Vivado 设计套件的免费轻量级编程与调试版本，包括 Vivado 器件编程器、Vivado 逻辑与串行 I/O 分析器以及存储器调试工具。实验室版本旨在用于无需功能齐全的 Vivado 设计套件的实验室环境中，因此其相比完整的 Vivado 设计套件，尺寸要小 75%，从而大幅缩短了实验室设置时间，也降低了系统存储器要求。对需要远程调试或以太网编程的设计团队来说，Vivado 设计套件 2015.1 还提供了独立的硬件服务器，所占用的资源还不到完整 Vivado 设计版本的 1%。

互动时钟域交叉分析

互动 CDC 功能可在设计周期的早期阶段调试 CDC 问题，减少了成本不菲的系统内调试环节。结合互动时序分析和交叉探测特性，Vivado 设计套件能带来无与伦比的时序分析和调试功能，从而加速产品上市进程。

Vivado 仿真器和第三方流程

仿真流程改进将 LogiCORE™ IP 编译时间缩短一半以上。整体仿真性能比此前版本高出 20%。Vivado 仿真流程与赛灵思联盟计划成员 Aldec、Cadence Design Systems、Mentor Graphics 和 Synopsys 等公司的流程全面集成。

赛灵思 SDK 高级系统性能分析

赛灵思进一步扩展了 SDK 的功能，使其能够分析 Zynq®-7000 All Programmable SoC 设计的性能与带宽情况，包括处理系统（PS）的关键性能参数分析，以及 PS、可编程逻辑（PL）和外部存储器之间的带宽分析。采用 AXI 流量生成器的系统建模设计主要面向 ZC702 和 ZC706 评估板提供。

器件支持

新器件
本版本推出以下 UltraScale™ 器件：

- Virtex® UltraScale 器件：
XCVU125、XCVU190、XCVU440
常规使用

- Kintex® UltraScale 器件：
XCKU035、XCKU060、XCKU115

- Virtex UltraScale 器件：
XCVU065

早期试用（请联系您本地的赛灵思销售代表）

- Virtex UltraScale 器件：XCVU160

许可证

Vivado 设计套件 2015.1 支持最新许可证。客户端通过 Vivado License Borrow 可借用一个浮动席位并绑定到一个设备，以便在特定时段非联网使用 Vivado 工具（仅适用于激活许可证）。此外还提供激活型许可证的虚拟机支持。如果客户端（节点绑定）许可证使用 Vivado License Manager，并连接到因特网，那么 Vivado License Manager 会自动下载并安装激活许可证。

IVADO 设计套件： 设计版本更新

部分重配置和 Tandem 配置

部分重配置 IP 现已开始面向 7 系列、Zynq SoC 或 UltraScale 器件中的所有部分重配置用户供货。该 IP 是 PR 系统核心，在发生软硬件触发事件时从存储器读取并交付给配置端口部分比特流。最新版本扩展了对 UltraScale 器件的支持，同时还可支持 KU115、VU125 和 VU190 器件，以及此前支持的 KU040、KU060 和 VU095 器件的实现。Tandem PROM 和 Tandem PCIe[®] 提供给相同的 UltraScale 器件，现在也具有 PR 支持（KU115、VU125 和 VU190）。

[更多信息，敬请参见：PG193。](#)

Vivado IP Integrator

本版本更新包括自下而上的综合流程选项，可加快设计迭代。每个 IP 自己综合，只有 IP 需要再次综合时才修改。Vivado IDE 提供专门针对 IP Integrator 优化的最新布局，从而将包括设计生成与验证，以及版本控制易用性改进等在内的项目流程时间缩短达一半之多。该版本还支持将设计保存为验证状态，这样生成时就无需再次运行验证。“Add IP …”窗口的搜索得到改进，现在能快速存取 IP 细节。

[更多信息，敬请参见：Vivado 设计套件 2015.1 版本说明。](#)

VIVADO 设计套件： 系统版本更新

Vivado 高层次综合

系统版本新增了可综合的 C++ 库函数，专注于软件定义无线电应用：数控振荡器（NCO）、QAM 调制器和解调制器。[更多信息，敬请参见：UG902。](#)

DSP 系统生成器

本版本新增了对硬件协同仿真突发模式的高级支持，能加速仿真，从而将性能提升 100 倍。更出色的时序分析能通过交叉探测快速发现伪路径。新功能可对 Vivado IP Integrator 的 SoC 平台设计进行解析，以便量身定制配套系列网关，从而确保简单精确的 IP 开发。增强对多个 AXI4-Lite 接口的支持，有助于独立寄存器与时钟域保持一致。最后，该工具现可支持 MATLAB[®] 2015A。

[更多信息，敬请参见：Vivado 设计套件 2015.1 版本说明。](#)

赛灵思知识产权（IP）更新

赛灵思推出了新一代因特网协议视频传输（video-over-Internet Protocol）连接和专业视频解决方案，支持“任何媒体通过任意网络传输”，满足广播和专业音视频市场要求。

因特网协议视频传输连接

赛灵思为发布分布网络定义和部署互联网协议视频传输，满足 SMPTE ST 2022-1,2,7 和 SMPTE ST 2022-5,6,7 内核及参考设计要求。ST 2022-1,2,7 和 ST 2022-5,6,7 内核现已开始由赛灵思供货。更多信息，敬请参见：<http://china.xilinx.com/products/intellectual-property/ef-di-smpte2022-12.html> and <http://china.xilinx.com/products/intellectual-prop-erty/ef-di-smpte2022-56.html>。

Vivado 设计套件 2015.1 版本提供 Video-over-IP FEC Engine 1。赛灵思为所有内核提供免费的系统级设计，并配套提供应用指南，请见：http://china.xilinx.com/esp/broadcast/refdes_listing.htm。

新系列“任意设备间”视频连接解决方案

该版本包括 HDMI 1.4/2.0、支持 HDCP 的 DisplayPort 1.2 和新版 6G/12G SDI。赛灵思开发和支持的这些接口将帮助开发人员满足 4K/Ultra-HD 等最新业界标准要求。

赛灵思 Vivado 设计套件 2015.1 版本提供视频连接解决方案，Artix-7、Kintex-7、Vir-tex-7、Virtex-7X 和 Kintex UltraScale FPGA 以及 Zynq-7000 All Programmable SoC 将支持该解决方案。面向 Kintex-7 和 Kintex Ultra-Scale FPGA 以及 Zynq SoC 的视频处理套件现已开始由 Omnitek 供货。更多信息，敬请访问：<http://omnitek.tv/sites/default/files/OSVP.pdf>。

了解更多

快速入门视频教程

Vivado 设计套件快速入门视频教程是一系列详细介绍 Vivado 设计套件和 UltraFast[™] 设计方法特性与功能的操作说明视频。新课题包括：

- Vivado 2015.1 的最新消息
- 利用 Vivado 中的 Cadence IES 进行仿真
- 利用 UltraScale Memory IP 进行设计
- 利用 IP Integrator 自动化使用开发板

如需观看所有快速入门视频，敬请访问：china.xilinx.com/training/vivado。

培训

讲师指导的有关 Vivado 设计套件、UltraFast 设计方法等培训课程，敬请访问：china.xilinx.com/training。

欢迎立即从以下网址下载 Vivado 设计套件 2015.1：<http://china.xilinx.com/download>。

Xilinx联盟计划合作伙伴的最新最佳技术

本实例 (*Xpedit*) 重点介绍了赛灵思联盟计划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

DAVE 的 BORA 现可支持 SDSO 开发环境

DAVE Embedded Systems (位于意大利波西娅) 一直与赛灵思开展合作，旨在确保其 BORA 模块支持赛灵思的 SDSoC 开发环境。BORA 模块基于赛灵思的 Zynq®-7000 All Programmable SoC。既然 BORA 可支持 SDSoC，BORA 用户则能够轻松快速地开发自己的软件算法。虽然 BORA 的硬件加速功能在实现在可编程逻辑中，但它明显能从运行于 Zynq SoC 的双核 ARM® Cortex™-A9 处理系统上的软件应用中调用。用 C、C++ 或 SystemC 编写的加速函数能从软件领域转移到

位于现有实现方案顶部的 FPGA 架构上。

DAVE 于今年 2 月在德国纽伦堡举行的 2015 年嵌入式世界大会 (Embedded World) 上演示了 BORA 系统。所演示的 IP (即 LCD 控制器) 采用经典工具开发，并通过 SDSoC 设计环境生成。

如需了解更多信息，敬请访问：
<http://www.dave.eu>。

XYLON 使用 SDSOC 来实现基于 MICROZED 的视觉平台

Xylon (位于克罗地亚萨格勒布) 开发

的 logicBRICKS IP 核旨在帮助客户走在图形处理和计算机视觉领域的科技创新前沿。为了提供能顺利集成到赛灵思 All Programmable SoC 和 MPSoC 中，并可实现不断发展演进的视频处理、对象检测和视频分析算法的可重用 IP，Xylon 的设计人员已成为使用最新赛灵思设计工具和技术的专家。赛灵思在开发其创新性 SDSoC 开发环境——作为手动 RTL 编码的替代方案——的过程中，Xylon 一直与赛灵思保持密切合作。

借助该最新开发环境，Xylon 只用两周时间就能为基于 MicroZed 开发板的视觉平台开发出了板支持包

(BSP)。通过将传统 logicBRICKS IP 用作 C 语言可调用的 RTL IP, Xylon 设计出了一款实时面部特性跟踪系统, 该系统曾在 2015 年嵌入式世界大会上进行过演示。

如需了解更多信息, 敬请访问:
<http://www.logicbricks.com/>。

IVEIA利用SDSOC开发边缘检测算法

利用赛灵思的 SDSoC 开发环境, iVeia (位于美国马里兰州安纳波利斯) 在嵌入式世界大会上展示了其用于处理高清视频的纯 C/C++ 参考设计。利用 iVeia 的 Atlas-I-Z7e 模块上系统和视频开发套件, 公司能在 GigE Vision 的高清摄像头实时运行精细的边缘检测算法。

两个相同的 C/C++ 代码实现方案并行运行展示, 一个用标准的 C/C++ 编译器进行编译, 另一个则用 SDSoC 开发环境的全系统优化编译器编译。通过比较, SDSoC 开发环境实现方案的性能比标准实现方案高出几乎两个数量级。使用 SDSoC 环境对设计进行特性分析和分区, 这样被识别为有高重复性、高计算强度的函数就能用可编程逻辑处理, 而更复杂的统计算法则继续由处理系统处理。

如需了解更多信息, 敬请访问:
<http://www.iveia.com/>。

ADI用SDSOC打造无线电参考设计

任何基于 SoC 的软件无线电 (SDR) 平台都面临三大挑战, 那就是软件和 HDL 之间的设计分区、在 HDL 代码中实现数据处理算法, 以及将 HDL 代码集成到主系统设计中。赛灵思 SDSoC 开发环境通过集成开发环境应对上述挑战, 用户在该环境中可用 C/

C++ 代码实现整个设计, 选择哪些部分用可编程逻辑执行, 并让代码工具生成设计的软件和 HDL 部分, 最后将各部分集成在一起, 创建最终的系统设计。

Analog Devices (位于马萨诸塞州诺伍德) 已开始采用 SDSoC 开发环境为其基于 AD9361/AD9364 RF 捷变收发器的 SDR FMCOMS2/3/4/5 平台创建参考设计。首个发布的参考设计展示如何在 SDSoC 开发环境中实现直接数字综合器 (DDS) IP 核并整合所生成的 IP 到 Analog Devices FMCOMMS HDL 平台和 Linux 环境中。这样, 就能发送 DDS 的输出并由 FMCOMMS 卡无线接收, 进而在 Analog Devices IIO Scope Linux 应用中展示。

DDS IP 根据赛灵思的高层次综合技术 (HLS) 完全用 C 代码实现, DDS 函数在 C 主循环中调用, 生成 IP 与设计其他部分之间的数据流绑定。根据该代码, SDSoC 开发环境能综合 DDS IP, 将其集成到 Analog Devices HDL 平台中, 并生成所有必要的文件, 从而能在 Zynq-7000 All Programmable SoC 上运行 Analog Devices Linux 分布。Analog Devices 在嵌入式世界大会上展示了 DDS HLS IP 和 Zynq SoC SDR 套件。

DORNERWORKS推动XEN HYPERVISOR支持ZYNQ ULTRASCALE+ MPSOC设计

赛灵思已选择与 DornerWorks 合作, 让其为赛灵思客户提供开源 Xen hypervisor 解决方案、全面的客户支持和工程设计服务。Xen hypervisor 是一款比较成熟的虚拟机监控器, 能同时运行多个操作系统, 这就使得 Xen 成为今后基于 Zynq UltraScale+™

MPSoC 产品的明确之选, 能满足下一代云端、数据中心和无线及有线网络计算的要求。Xen 既可提供应用虚拟化服务, 也可提供运行于 Zynq MPSoC 四核 ARM Cortex-A53 处理器上的客户操作系统的虚拟化。

赛灵思产品系列和解决方案市场副总裁 Hugh Durdan 指出: “DornerWorks 为 Xen hypervisor 解决方案提供了业经验证的成熟技术及专业知识, 而且拥有嵌入式设计和 FPGA 逻辑设计的强大实力。我们与 DornerWorks 开展 Zynq UltraScale + MPSoC 的 Xen hypervisor 解决方案合作将帮助我们的客户快速采用虚拟化安全型 OS 平台加速产品的开发进程。”

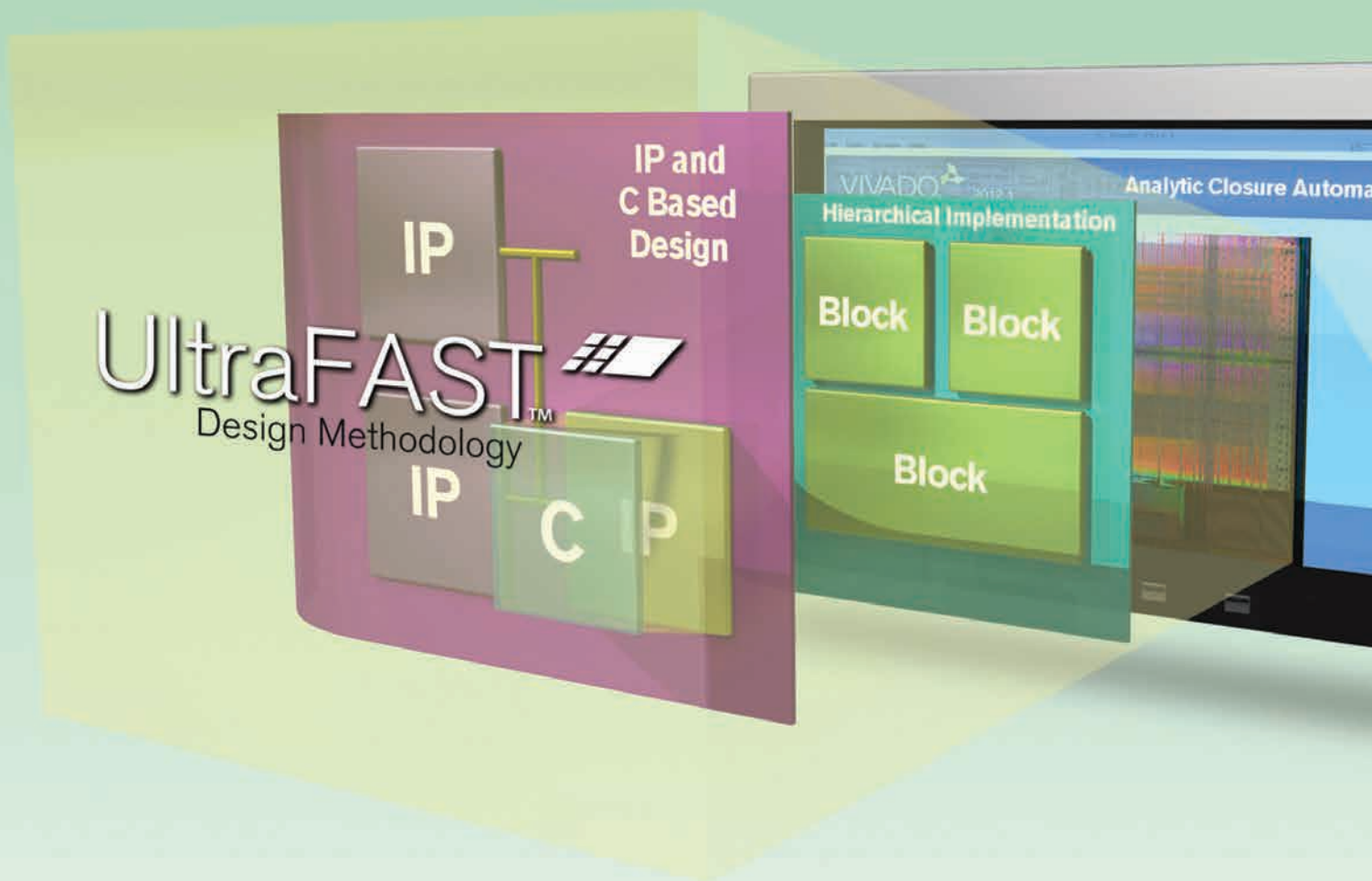
Hypervisor 的开发不仅需要 Xen 内核, 也需要配置 dom0 (享有特权的系统域) 和使用客户操作系统的配置客户域。DornerWorks 将提供完整包和文档的样本发布, 介绍如何宣传您自己的定制系统。

DornerWorks 还为客户提供了一系列的服务, 包括 FPGA 设计服务等, 从而打造完整的 hypervisor 生态系统。

如需了解更多详情, 敬请访问:
<http://dornerworks.com/services/XilinxXen>.

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。