

面向异构多核处理器的 FPGA 验证

李小波^{1,2,3} 唐志敏^{1,2,3} 李文³

¹(计算机体系结构国家重点实验室(中国科学院计算技术研究所) 北京 100190)

²(中国科学院大学计算机科学与技术学院 北京 100049)

³(上海处理器技术创新中心 上海 200120)

(lixiaobo17b@ict.ac.cn)

FPGA Verification for Heterogeneous Multi-Core Processor

Li Xiaobo^{1,2,3}, Tang Zhimin^{1,2,3}, and Li Wen³

¹(State Key Laboratory of Computer Architecture (Institute of Computing Technology, Chinese Academy of Sciences), Beijing 100190)

²(School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049)

³(Shanghai Processor Technology Innovation Center, Shanghai 200120)

Abstract With the development of processor architecture, high-performance heterogeneous multi-core processors are emerging. Since the design of high-performance heterogeneous multi-core processor is very complex, in order to reduce the design risk, shorten the verification cycle, carry out software development in advance, reproduce the post-silicon problems, we usually need to build a prototype verification platform of field programmable gate array (FPGA), and based on the FPGA platform to carry out a variety of software and hardware verification and debugging work with different functions. This paper presents a method of debugging and verifying heterogeneous multi-core high-performance processor based on homogeneous FPGA platform which effectively utilizes the architecture characteristics of heterogeneous multi-core processor and the symmetry characteristics of homogeneous FPGA platform, divides FPGA by hierarchical top down method, builds the platform from bottom to up. The combination of speed bridge, adaptive delay adjustment, embedded virtual logic analyzer and other technologies can quickly complete the FPGA platform bring-up and deployment. The proposed multi-core complementary, inter-core replacement simulation method with debug SHELL can verify the target high-performance heterogeneous multi-core processor quickly and completely. Through the FPGA prototyping platform, we have successfully completed the pre-silicon verification, software hardware co-development and testing, post-silicon bug reproduce and also provided a fast hardware platform for the next generation processor's architecture design.

Key words heterogeneous multi-core; field programmable gate array (FPGA) prototyping; speed bridge; adaptive delay adjustment; virtual logic analyzer (VLA); inter-core replacement simulation

收稿日期:2020-04-22;修回日期:2020-12-09

基金项目:国家重点研发计划项目(2018YFB1003501);国家自然科学基金项目(61732018,61872335);计算机体系结构国家重点实验室开放课题(CARCH201914)

This work was supported by the National Key Research and Development Program of China (2018YFB1003501), the National Natural Science Foundation of China (61732018,61872335), and the Open Program of the State Key Laboratory of Computer Architecture (CARCH201914).

摘要 随着处理器架构的发展,高性能异构多核处理器不断涌现。由于高性能异构多核处理器的设计十分复杂,为了降低设计风险,缩短验证周期,提前进行软件开发,复现硅后问题等,通常需要搭建现场可编程门阵列(field programmable gate array, FPGA)的原型验证平台,并基于 FPGA 平台开展种类繁多,功能各异的软硬协同验证和调试工作。提出的基于同构 FPGA 平台对异构多核高性能处理器的 FPGA 调试、验证方法,有效地利用了异构多核处理器的架构特征,同构 FPGA 的对称特点,以层次化的方法自顶向下划分 FPGA,自底向上构建 FPGA 平台。结合差速桥、自适应延迟调节、内嵌的虚拟逻辑分析仪(virtual logic analyzer, VLA)等技术可快速完成 FPGA 平台的点亮(bring-up)和部署。所提出的多核互补,核间替换模拟的调试 SHELL 等方法可以快速完整地对目标高性能异构多核处理器进行 FPGA 验证。通过该 FPGA 原型验证平台,成功地完成了硅前验证,软硬件协同开发和测试,硅后问题复现工作,并为下一代处理器架构设计提供了快速的硬件平台。

关键词 异构多核;FPGA 原型验证;差速桥;自适应延迟调节;虚拟逻辑分析仪;核间替换模拟

中图法分类号 TP302

近年来,伴随着处理器应用领域的不断拓展,实际应用对处理器应对复杂场景的功能需求不断提高,面向通用计算场景的传统通用处理器逐渐不能适应这些需求,因而面向复杂应用需求的异构多核处理器不断涌现。对于高性能的异构多核处理器,由于其核组众多,核内架构复杂,核间通信多样,高速外设接口丰富,整体设计规模十分庞大。随之而来的是待验证的状态空间成倍的增长^[1],对其进行完备的功能验证已经成为异构多核处理器设计工作中的瓶颈。

传统基于软件模拟验证(simulation)^[2]的方法具有灵活性强,使用简单的特点,但随着异构多核处理器逻辑单元规模增大,全芯片系统级模拟验证速度下降明显。虽然采用基于事务(transaction)^[3]的验证方法可提高验证对象的抽象层次,从而加快验证的速度,但对于待验证的目标设计,其运行速度也只能达到几十赫兹(Hz),甚至几赫兹,并时有软件模拟 EDA(electronic design automation)工具及环境崩溃,根本无法运行大量广泛的系统级测试程序,更别提进行复杂的软硬件混合验证。

基于硬件仿真加速器(emulator)的仿真验证^[4],其运行速度可达几百千赫兹(kHz),甚至优化后能到兆(MHz)级。但其运行环境要求高^[5],维护成本高^[6],价格十分昂贵;另一方面虽然其运行速度比模拟速度提升几百到几千倍,但其仍然达不到运行操作系统、编译器、系统级软件应用和性能、压力测试程序等的要求。

现场可编程门阵列(field programmable gate array, FPGA)原型验证通过使用商用 FPGA 实现处理器设计的大部分功能(PHY, (phase locked

loop, PLL))等模拟定制电路无法在 FPGA 上实现),可以进行高速的原型验证^[7]。由于其运行速度通常可达几十到百兆级,可用可以运行更多更广泛的真实的测试程序和系统软件,经常可以发现许多软件模拟验证难于发现的问题。不仅如此,模拟验证和硬件仿真等手段,一般只能连接测试模型或者虚拟设备,这可能带来验证环境与真实物理设备行为之间的差异,而 FPGA 平台却可以直接连接外围真实的物理设备,用于系统级软件开发、测试和性能分析等方面,从而广泛被采用于处理器的开发验证中,并发挥着十分重要的作用。

1 相关工作

目前,FPGA 验证平台和验证方法在不同架构的处理器验证中也都有成功应用^[8]。英特尔(Intel)公司^[9]基本在其每一代处理器^[10-11]设计验证、性能分析中都有采用 FPGA 平台,以弥补传统模拟验证的不足。国内龙芯团队^[12]采用了多 FPGA 验证平台完成了龙芯-2G 的 FPGA 验证工作。这些 FPGA 验证平台主要侧重于单一架构,单一处理器核心在 FPGA 上的原型验证,而对于多晶粒(Die)的异构多核高性能处理器芯片如何进行 FPGA 平台的构建和及其 FPGA 调试验证方法暂未涉及。此外,这些原型验证平台,并没有涉及系统级的 FPGA 验证环境实现,大部分解决方案是将 FPGA 连接到已有主板上^[11]。但是如果是第 1 代芯片的研发,主板系统不可能在芯片验证阶段准备就绪,这就需要我们独立构建基于 FPGA 的系统级验证平台。

本文提出的基于同构对称 FPGA 平台对异构

多核高性能处理器的 FPGA 验证、调试方法,有效地利用了异构多核处理器的架构特征,同构 FPGA 的对称特点,以层次化的方法自顶向下划分 FPGA 功能,自底向上构建 FPGA 平台。采用差速桥(speed bridge)和自适应时延(adaptive delay)调整采样等技术,结合内嵌的虚拟逻辑分析仪(virtual logic analyzer, VLA)调试工具可以快速地完成 FPGA 平台的点亮和部署。运用多核互补,核间替换模拟的调试 SHELL 等验证方法可以有效地完成对目标高性能异构多核处理器及整个 SOC 芯片系统级的 FPGA 验证。

本文的贡献主要有 6 个方面:

1) 结合目标处理器架构,采用了层次化灵活可配置的方法,以子功能模组、功能模组、异构多核模组、晶粒(Die),自底向上、逐级递进的方法,最后扩展到全芯片,完成了 FPGA 平台的构建;

2) 针对高速 I/O 接口协议速度快、频率高,且全系统 FPGA 实现时 FPGA 频率较难达到实际协议速率等问题,运用了差速桥方案,成功实现了高速 I/O 速率匹配,完成了系统级的高速 I/O 互联,从而实现了大量高速 I/O 接口的 FPGA 验证和软件驱动的提前开发;

3) 对于多晶粒的 FPGA 平台互联,由于每个 FPGA 平台独立配置,它们上电顺序和时钟相位无法保证同步对齐等问题,通过插入可配置的自适应时延控制逻辑,有效地解决了信号跨多个 FPGA 平台同步采样问题;

4) 在 FPGA 调试过程中,通过构建 VLA,有效地解决了 FPGA 平台调试难题,加速了平台点亮和后续验证调试进程;

5) 在 FPGA 验证过程中,根据异构多核处理器的特点,提出了大小核互换、多核间模拟替代、内嵌

调试 SHELL 等方法,可以方便快速地进行软硬件协同验证,巧妙地解决了软件任务如何在核间划分,负载均衡等工作;

6) 应用该 FPGA 验证平台,可极大加速异构多核处理器验证进程,明显缩短硅前(pre-silicon) FPGA 验证时间,有效地保证芯片一次性流片成功,同时在芯片设计阶段就为软件开发提供了一个高效的开发调试硬件平台,可提前在这个硬件原型系统平台上进行软件开发测试,及早发现软硬件协同设计问题,并且可以为硅后验证(post-silicon validation),及后续项目的架构设计提供有效可靠的平台保障。

2 目标异构多核处理器简介

2.1 目标异构处理器结构

本文需要进行 FPGA 验证的目标处理器是一款异构多核高性能处理器,其总体结构如图 1 所示。该处理器由若干个晶粒互联组合封装而成,单个晶粒约 10 亿门,规模十分巨大,设计复杂度高。该处理器既包含负责初始化启动和安全管理的精简指令集(reduced instruction set computer, RISC)处理器核(简称小核),也包括用于各种复杂计算处理的通用复杂指令集(complex instruction set computer, CISC)处理器核(简称大核),还包括用于 Die 间互联切换控制、功耗能效管理的数字信号(digital signal process, DSP)处理器核(简称控核)。同时还包括多通道最高速率可达 3 200 Mbps 支持 ECC 的 DDR4(double data rate 4)内存控制器,及大量高速外设接口,比如多个 16 Gbps 的 PCIe4.0、若干 6 Gbps 的 SATA 3.0,10 Gbps 的 USB3.1,10 Gbps Ethernet 等高速 I/O 接口,此外还有各种低速的调试、维护管理接口等。

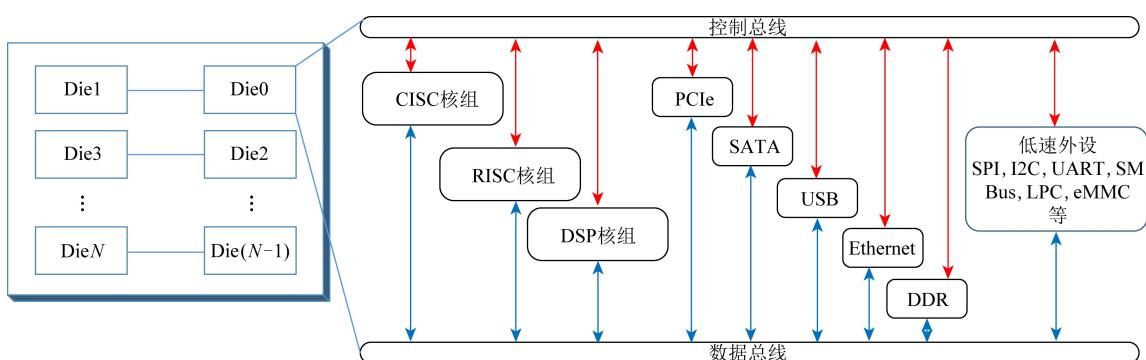


Fig. 1 The architecture of heterogeneous multi-core processor

图 1 异构多核处理器架构

2.2 目标异构多核处理器功能划分

图 1 处理器架构中从功能的角度出发,可以将整个处理器按功能划分为复杂指令集(CISC)处理器核组,精简指令集(RISC)处理器核组,数字信号(DSP)处理器核组,多路内存(DDR4)存储控制器组,各种高速互联接口组,低速调试接口组.接下来,从各个功能组团中选取具有代表性的一个核(模块)进行 FPGA 验证,最终我们选择了 1 个 CISC 核心,1 个 RISC 核心,1 个 DSP 核心,1 个 DDR4.0 通道,1 个 PCIe4.0 接口,1 个 SATA 3.0 接口,1 个 USB3.1 接口,1 个 Ethernet 接口和一些低速调试管理接口作为一个典型的最小配置.这样做有 2 方面原因:1)每个功能模组都有一个完备的功能模块,可以有效验证单个核/模块正确性;2)避免大量同质的复制,增大了 FPGA 规模,导致容量、成本、时间等一系列问题.

待最小配置的 FPGA 平台成功完成点亮并稳定运行后,再根据测试需求,可以添加更多核心和功能模块.比如当需要测试同构多核通信和缓存一致性,就可以添加 2 个或更多的处理器核心,让多个核心并发访问缓存页面,在测试例程检查结果同时,还可以通过后续第 4 节介绍的 VLA 记录总线上的访存序列,查看是否符合预期.如果需要并发压力测试多个 I/O 的吞吐率,可以集成更多的 I/O 模块,并通过调试 SHELL 驱动多个模块测试例程并发运行,以更真实地模拟实际芯片的工作情况,提高验证覆盖率.

2.3 目标异构多核处理器 FPGA 验证的挑战

虽然经过 2.2 节的功能核(模)组划分裁剪后,待验证处理器规模已经缩减了很大一部分,但其规模仍然很大,内部结构复杂,高速接口种类繁多,对其进行 FPGA 验证及软硬件协同开发调试等依然面临诸多挑战,主要包括:

1) 目标异构多核处理器架构内部互联复杂,逻辑规模庞大,高速 I/O 较多,如何划分映射到多片 FPGA 上,并满足目标芯片在模块级/系统级,单核/多核,单模块/多模块,单晶粒/多晶粒及全芯片等不同规模配置下的灵活快速点亮并实施有效验证,是一个巨大挑战;

2) FPGA 验证平台相比传统模拟验证和硬件加速器验证,信号可观察性、可控制性更差,调试十分困难,迭代周期长,如何设计一个模块化,灵活易用的虚拟逻辑分析仪,是一个亟待解决的问题;

3) 目标异构多核处理器,不同于传统同构多核

处理器,如何进行异构核间协同、软件任务划分、负载均衡、系统调度,如何有效利用异构多核的架构进行更有效快速验证,也是一个值得探讨方向.

本文后续章节将分别介绍如何应对这些挑战,解决本节提到的各种问题,最终完成该异构多核处理器的 FPGA 平台构建、调试、软硬件协同验证及硅后问题在 FPGA 平台上的复现.

3 FPGA 原型验证平台构建

本节将详细介绍如何根据目标异构多核处理器架构,构建灵活可扩展的 FPGA 平台.具体工作包括:设计裁剪及容量预估,FPGA 平台选型、逻辑分片及时分复用、差速桥集成、时钟等比缩放、多平台互联等.

3.1 设计裁剪及容量预估

如 2.2 节所述,由于待验证异构多核处理器规模巨大,现有最大的 FPGA 容量也相去甚远,如何裁剪设计的规模成为 FPGA 平台能否构建的前提.具体方法为:

1) 同质核心空壳(empty shell),由于本处理器采用了异构多核架构,每种架构处理器可能有一个或若干个,在不改变系统顶层情况下,最灵活方法是每种架构的核组只保留一个核心,其他同质核心全部替换为空壳(shell).在设计时我们经过精心考虑,在空壳核心输出信号连接合适初始化值(tie0/tie1)的状况下可以保证整个全片仍然能正常工作.

2) 保留必不可少模块,分步集成各核心和高速 I/O 模块,自底向上策略逐步扩大验证规模.待最小配置点亮后,再根据验证需求添加更多的处理器核心和功能模块.

3) 资源预估,对每个功能子模块进行面积、引脚、时钟等资源评估.

图 2 展示了待验证目标系统的最小配置下各功能模块的面积,在小饼图中我们可以注意到单个 CISC 核心面积(大核)是 RISC 核心(小核)和 DSP 核心(控核)的 20 倍以上,相差很大,这给 FPGA 资源分配和规划带来新的挑战,同时也是本文第 5 节中使用大小核模拟替换协同验证的重要原因.

3.2 FPGA 平台选取

根据不同芯片架构对 FPGA 验证需求的不同,会选适合待验证芯片架构的 FPGA 验证平台,有 FPGA 平台基于其自身架构的特点和验证的需要选取了子母板架构^[13].而基于我们待验证的处理器,

其异构多核对称多晶粒架构的特点以及为了满足本文第4节介绍的通用调试接口需求,经过多种权衡考量,最终采用了统一架构的FPGA平台。这样做有利于验证平台的层次化、模块化,具有灵活易扩展等特性,方便多个晶粒级联,并具有统一的外部接口。

FPGA型号方面,由于设计规模巨大,选择Xilinx公司^[14]目前较大容量的XCVU440芯片,每个验证板上包含4颗该芯片,引脚数为1456,分别连接到24个连接头上,每个连接头可以传送50个有效信号,其逻辑示意如图3所示。

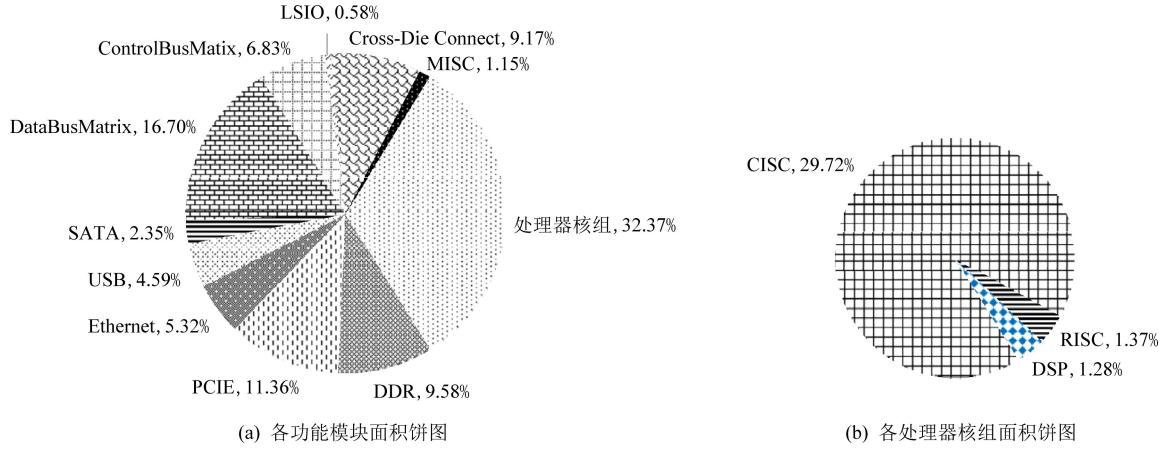


Fig. 2 The area pie of the each function module

图2 各功能模块面积饼图

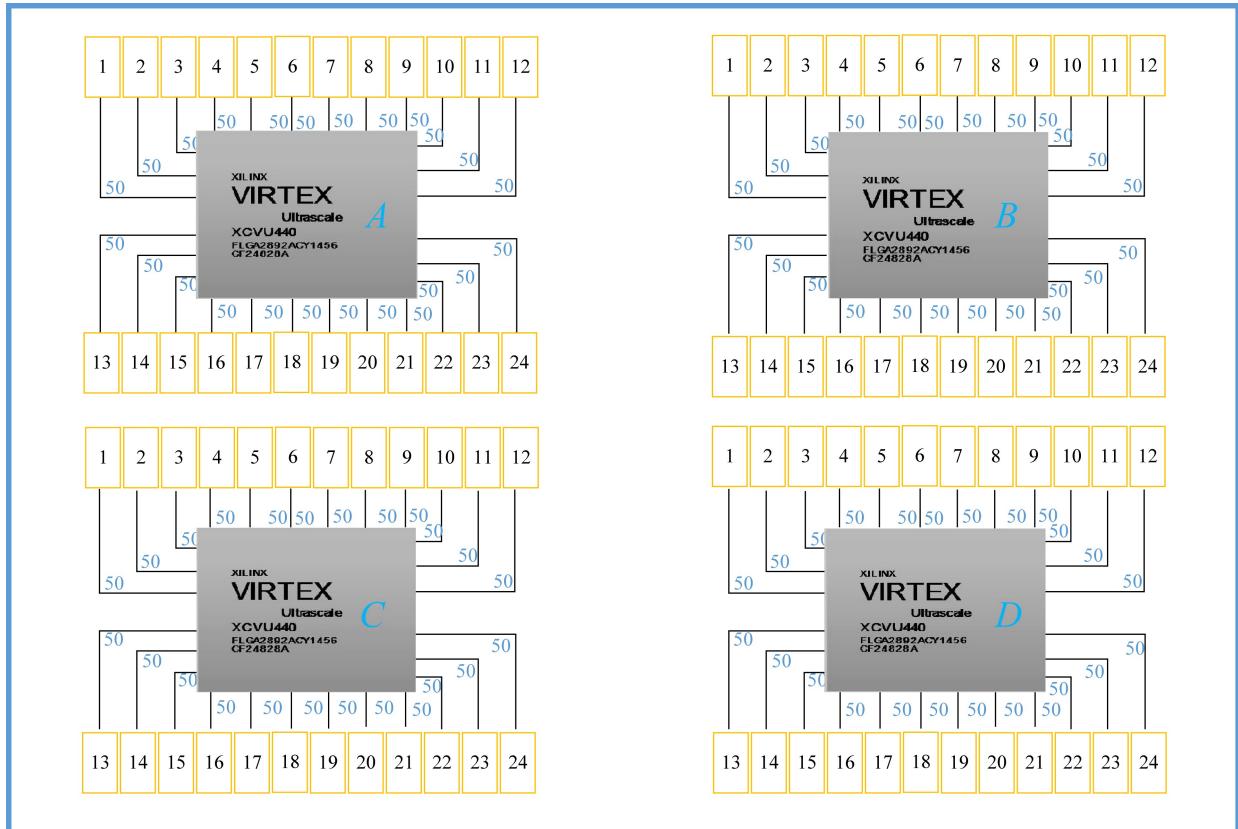


Fig. 3 FPGA board connection logic

图3 FPGA板连接逻辑示意图

3.3 分片及时分复用

对于单颗XCVU440虽然其等效ASIC达到了约

2600万门左右,但是相对于我们需要进行FPGA验证的目标复杂异构多核处理器规模,仍然相距甚远,

不可避免需要使用多片 FPGA 才能完成.对于待验证目标处理器到多片 FPGA 的逻辑分片,如果使用纯手工对设计划片,虽然可以得到比 FPGA 划分工具更好的结果,但这需要有对设计十分了解的有经验的工程师和较长的时间迭代;如果尝试使用 EDA 工具的自动分片,较难得到期望的理想结果,而且从资源预估结果看需要使用 80 多片 FPGA 才能实现,目前现有的 FPGA 工具难于处理如此大规模的设计.

为了解决这个难题,结合芯片架构特征我们采取了人工深度干预 FPGA 工具的混合划片方法:

1) 自顶向下,对称逻辑划片以晶粒为第 1 层次进行切割划分以减少综合、布局布线 FPGA 数量;

2) 晶粒内部,结合 3.1 节预估信息,裁剪部分同质核心/模块,减小规模后,同时根据芯片互联数据通路及微架构,首先人工固定部分需要外接到 I/O 引脚的功能片(tile)和需要特殊处理的功能模块后,再引入自动划片工具对其他剩余逻辑进行划分;

3) 基于文献[15]介绍的 K-L 划分方法,调整部分功能片,让紧耦合互联信号多的模块尽量放在同一片 FPGA 内,尽量前期减少互联信号总数;

4) 采用时分复用方法^[15],以高频时钟传输接口信号来满足 FPGA 片间传输较多信号超过连接

引脚(pin)数的需求.

通过这 4 个方法,可以将一个经典配置划分在 12 颗 FPGA 中,而纯 FPGA 工具需要 19 颗 FPGA,得到一个相对较优的划分结果.此外,根据 FPGA 板拓扑位置,将 DDR4 和低速调试 I/O 接口放在一个 FPGA 板中(Board 1),高速 I/O 接口统一划分在另一个 FPGA 板中(Board 3),跨晶粒互联全部位于 Board 2,形成非常规整的对称结构,这十分易于多 FPGA 平台互联实现.

3.4 差速桥实现

由于目标验证异构多核处理器内集成了 DDR4 存储控制器,PCIe4.0,SATA3.0,USB3.1,万兆 Ethernet 等众多高速 I/O 外设接口,为了验证它们的逻辑正确性,需要在 FPGA 平台上实现这些高速 I/O 接口.

如果根据标准协议,PCIe4.0 和 SATA3.0 等高速接口真实速度较快,对 FPGA 工作频率要求非常高.而目标芯片在划分互联后,芯片内部控制逻辑需要跨越多个 FPGA 板和多颗 FPGA 芯片,其频率已经很难达到诸如 PCIe4.0 的全速率要求,这就需要通过差速桥来匹配 FPGA 和真实接口之间的速率,使 PCIe4.0,SATA 3.0 等高速接口能在 FPGA 平台上以低速的形式实现.最终平台整体架构如图 4 所示:

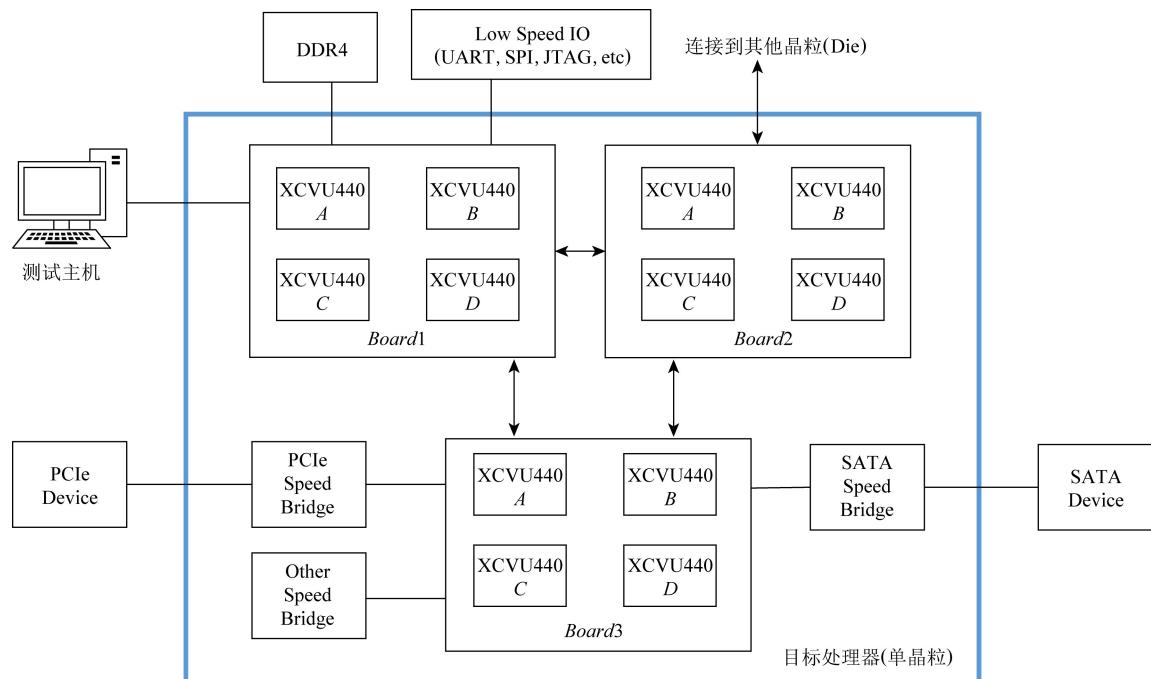


Fig. 4 The overall architecture of FPGA platform for target processor(single Die view)

图 4 目标处理器 FPGA 平台总体架构(单晶粒视图)

3.5 跨晶粒(Die)平台互联

目标验证异构多核芯片,采用了多晶粒封装架构,需要将图 1 的多晶粒组合起来,进行多晶粒互联验证和性能分析等.如果增加一个新的顶层,完全交给 FPGA 工具划分,由于工具处理能力有限,会导致 FPGA 工具崩溃;另一方面效率低下,我们以 4 晶粒互联为例,需要完成多达 48 颗 FPGA 的划分、综合、布局布线、生产下载文件的全流程,迭代一次需要一周以上.鉴于此,如果能利用到晶粒之间功能对称以及 FPGA 规整互联的特性,就可以极大简化这项工作.

由于晶粒内部逻辑功能完全一致,如果不需要考虑晶粒间的上电顺序和信号时钟同步(后面将介绍如何解决上电顺序和时钟采样同步),只需要通过改变FPGA引脚在启动时的上下拉高低值,就可以

区分主晶粒(master)和从晶粒(slave).引脚上拉的主晶粒在系统启动过程中,固件(firmware, FW)会根据当前晶粒的主从情况,选择不同初始化执行路径,在等待主从晶粒完成数据通路建立后(Step5),主晶粒固件就可以完成对其他从晶粒的配置工作.另一方面由于FPGA平台选取了统一的架构和相同型号的FPGA,就可以所有晶粒都采用一样的FPGA配置文件,这样只需要完成单晶粒的12颗FPGA平台实现以后就可以同时支持多晶粒的互联部署,这极大地提高了工作效率,缩短调试时间.

而多晶粒(多 FPGA 平台)的上电顺序和互联的信号和时钟如何同步,也是一个亟待解决难题。结合 Die 间互联协议和采样的要求,我们引入了自适应与软件可配置的时延控制机制,成功实现了接口采样的准确互联和信号同步,具体原理如图 5 所示:

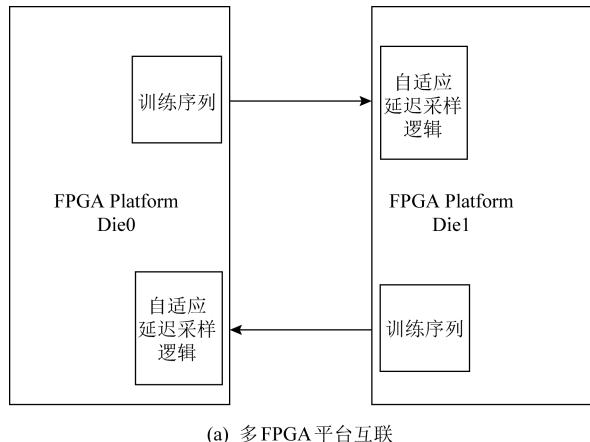


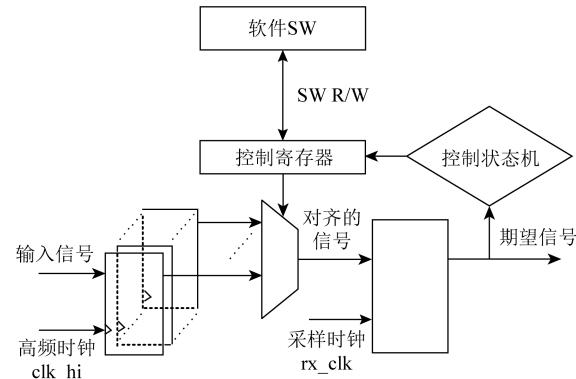
Fig. 5 Adaptive delay synchronized logic

图 5 FPGA 自适应延迟调节同步逻辑

自适应延迟电路嵌入接收端(RX),通过寄存器修正接收到信号延迟,本设计中根据延迟需求,采用800 MHz 高频时钟(clk_hi)控制采样时延,每个时延步长 1.25 ns,共 4 b 控制,延迟区间范围 0~20 ns,可以覆盖接口传输信号的全周期。

自适应时延调整主要分 6 个步骤：

- 1) FPGA 平台系统上电完成,各平台时钟启动、Reset 等释放;
 - 2) 发送端复位完成后,自动发送训练序列;
 - 3) 接收端状态机根据采样结果,不断修正控制选择寄存器,控制输入信号以 1.25 ns 为步长持续滑动,以调整接收数据和采样时钟 rx_clk 之间相位;
 - 4) 当接收信号符合预期后,控制状态机结束时延修正 并通知发送端停止发送训练序列,



(b) 自适应延迟采样逻辑实现

5) 固件在启动序列中,会轮询(polling)控制寄存器值,进行软硬件同步,确认 FPGA 平台间数据采样同步完成;

6) 软件进入下一步,控制 FPGA 平台接口间进行正常的数据传输,多晶粒 FPGA 平台同步完成。

4 FPGA 平台调试技术

如何提高 FPGA 的调试能力,是 FPGA 平台能否快速点亮部署和定位硬件/软件问题的关键.基于传统的 FPGA 自带工具抓取信号的方法,常常受限于 FPGA 内部存储(memory)容量限制,不能观察较多信号,较长时间(信号深度).另一种方法是将信号外接到引脚(pin)借助逻辑分析仪等调试设备,也

受限于可用引脚限制和采样深度,只能选取一些关键信号.而对于目标验证芯片,其内部数据总线的位宽较大,需要观察信号也非常多,利用 FPGA 的 I/O 管脚做调试引脚常常会 I/O 数量不足.另一方面,在多核处理器芯片调试过程中,常常由于异构多处理器架构叠加多核多线程和分支预测、推测执行等复杂技术,导致问题表现得一定随机性,有时候测试进

行几个小时就能发现问题,有时候需要几十个小时才能碰到问题,这就需要记录多个处理器核的多个线程的长时间运行指令序列(PC 值),以进一步分析和定位问题.针对这些调试难题,我们自研了基于以太网口的虚拟逻辑分析仪(VLA),其基本结构如图 6 所示.其主要分为 3 个功能模块:信号获取、信号传输、信号解析.

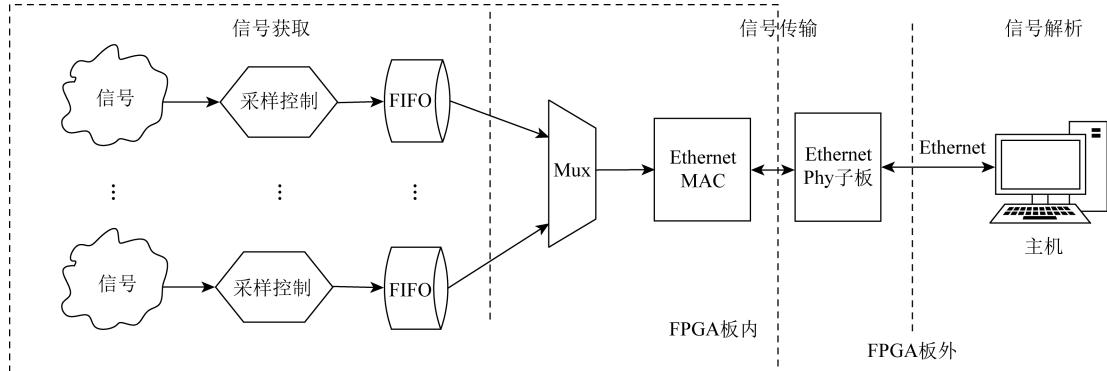


Fig. 6 Basic architecture of VLA

图 6 VLA 基本结构

4.1 VLA 信号获取

VLA 信号获取是一个标准化的模块,只需要实例化在设计中,然后将需要获取信号,或者调试总线等连接到该模块顶层端口上,并连接上一致性的时钟,复位和控制信号就可以完成信号的获取,使用起来十分简洁.

此外,还可以通过软件配置寄存器来灵活控制 VLA 的触发(trigger),可以在任意时刻或者软件发现错误时开始信号获取、暂停、再获取,结合软件配置选择调试信号的输出,灵活地控制需要获取时间段,信号深度等参数.

4.2 VLA 信号传输

最终的获取信号会通过一个或多个以太网(Ethernet)硬件控制器自动封包成符合 Ethernet 包(packet)格式的以太网包,并通过 FPGA 上以太网控制器发送出去,通过网络传输给任意目标主机.由于调试信号是实时传输到主机,几乎不占有 FPGA 自身存储空间,其信号的获取深度理论上可以无限大(只要主机存储空间足够大),这成功地解决了 FPGA 自带调试工具和外接逻辑分析仪等设备获取调试信号深度有限的问题.

4.3 VLA 信号解析

目标主机可以通过常用的抓包程序(如 Wireshark, Sniffer 等)捕获以太网包,并存储在特定文件中.结合信号的获取顺序和位宽,通过脚本简单处理

收到的以太网包中的数据,再呈现给设计、验证及软件人员.而且由于处理器系统级验证特殊性,在大部分情况下并不需要再看波形,只需要分析回放(replay)指令执行序列,软件工程师就可以直接定位出错的指令序列,十分易于调试,这极大地提高了工作效率.当然,VLA 不仅仅可以用于记录程序执行序列,根据调试验证模块的不同,还可以记录其他希望观测调试的信号.比如控制总线事务,DDR 读写序列(sequence),PCIe DMA 序列等信息,这十分便于对相关模块工作状态的检查和调试.

5 异构多核处理器验证方法探索

本节将介绍使用本文构建完成的 FPGA 平台,结合异构多核处理器的特点,开展相关 FPGA 验证工作.

5.1 异构处理器间协作验证动机

本目标异构多核处理器设计十分复杂,不同核组之间的规模和复杂度都相差较大,如图 2 所示.其中尤以 CISC 核为甚,该大核需要占用较多的 FPGA 资源,且其初始化启动过程繁复,用时较长,需要操作系统和驱动软件加载支持.如果在验证调试初期,希望能较快直接测试其他待测功能模块(例如 PCIe 接口),这时可以利用异构多核的特点,使用另外的一个异构的 RISC 核(简称小核)来完成 PCIe 的快

速初始化和直接测试.此外,对于芯片的硅后问题定位,也可以在更简单直接的小核系统中进行复现.

5.2 异构处理器间模拟替代

目标异构多核处理器,根据架构设计情况,在正常工作时,CISC,RISC,DSP核间各有任务分工.在FPGA验证过程中,为了更灵活地验证设计,如图1所示,这些核组都连接在芯片内部的数据和控制总线上,因此可以完成异构核间相互模拟替换,在更早阶段,以更少的资源就可以开展其他各个高速外设模块或处理器核间的交叉验证,这样既节约了FPGA资源,又简化平台点亮,极大地加快了原型验证测试进程.此外,采用大小核方式协调还十分有利于问题排除法定位,芯片硅后调试时也可以用于

开展各个模块(特别是各高速外设)的问题复现(reproduce).为此我们引入了专用于FPGA原型验证和硅后验证的调试解释外壳程序(调试SHELL).

5.3 调试SHELL工作流程

调试SHELL是一组专用于FPGA原型验证和硅后问题复现的测试程序集合及其构建的裸芯片(bare-metal)运行环境,它可以灵活地内嵌在异构多核处理器正常启动流程的适当阶段.如图7所示,调试SHELL作为工程开发模式一部分,在RISC处理器引导程序完成后,可以选择是否进入调试SHELL,调试SHELL一旦开启,就可以以更灵活直观方式对设计中的其他功能模块或处理器核心进行测试.

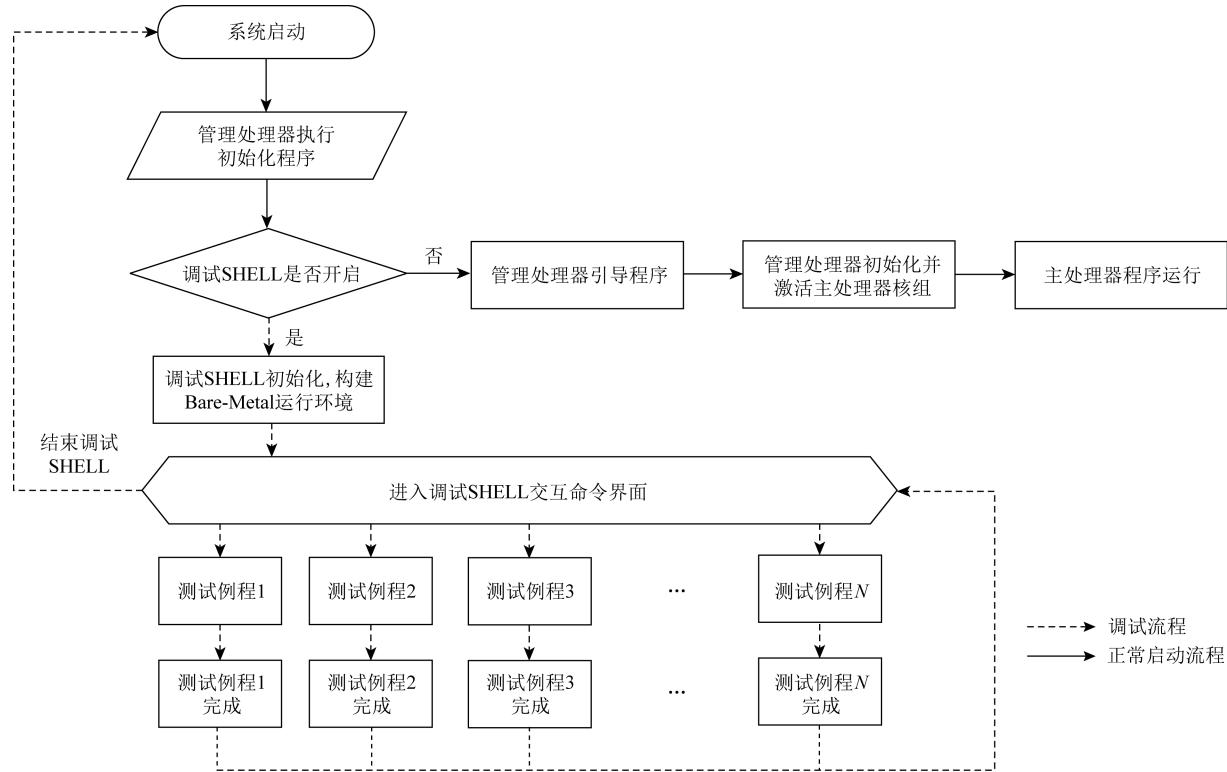


Fig. 7 The flow of Debug SHELL

图7 调试SHELL执行流程

此外,调试SHELL由于不依赖于软件库(lib)和操作系统(operating system, OS),可以嵌入任何一种架构的处理器核上执行,使用起来非常灵活.除管理测试例程之外,它还为测试例程提供裸芯片(无BIOS,OS支持)的运行环境,它主要有3方面重要功能:1)地址映射.CISC处理器、RISC处理器、DSP处理器数据通路差异,地址空间的映射不一样,这需要调试SHELL来加载测试例程和初始化对应处理器的堆栈空间等基本运行环境.2)存储管理.多个测

试例程存放位置、运行地址、内存和内部SRAM空间分配和管理等.3)多测试例程简单并发.调试SHELL可以宏观上“同时”启动多个测试例程,比如测试例程1主要测试PCIe,测试例程2测试USB,调试SHELL可以并发调度例程1和例程2执行,让PCIE,USB同时工作,从而能更真实模拟实际芯片工作场景,增加测试覆盖率和测试压力.

以调试SHELL工作在小核上为例,其主要工作划分8个步骤:

- 1) FPGA 平台系统上电启动;
- 2) 管理处理器核组(小核)解复位,开始 Bootcode 管理程序执行和系统初始化;
- 3) 管理程序执行完成,进入是否开启调试 SHELL 选项,如果选择开启,进入 4),否则进入正常启动执行 7);
- 4) 调试 SHELL 初始化,并进入交互界面,并选择测试例程 X 进入 5),或选择结束测试进入 1),重新启动系统;
- 5) 进入测试例程 X,并开启对应测试验证;
- 6) 测试用例 X 执行测试完成并反馈测试结果,返回 4);
- 7) 开启引导程序流程,并解复位主处理器核(大核);
- 8) 主处理器开始正常程序启动流程,完成系统启动.

6 FPGA 原型验证结果

通过构建灵活可配置的 FPGA 平台,可以实现多核异构处理器内各核组、各模块的 FPGA 原型验证,通过异构处理器大小核互补迭代,可以方便实现芯片不同规模,不同研制阶段,硅前、硅后验证及软件开发测试需求,并为后续下一代项目提供快速评估平台.结合我们项目经验,主要分 4 个 FPGA 验证的不同阶段,如图 8 所示,参考每个阶段发现的硬件设计问题、软件问题及对项目的贡献等因素可以计算 FPGA 在各个阶段收获的验证成果.

- 1) 流片前验证阶段(阶段 1).在流片前 FPGA 搭建、各个功能模块点亮、后续固件(FW),主机

BIOS、软件驱动的开发,标准性能测试(SPEC)过程中,总共运行了 1600 多支测试(test cases),产出了半的验证成果.由于该 FPGA 平台最高可以运行在 14 MHz 的频率下,能够进行完整操作系统(OS)的启动全流程,同时可以对各功能模块实施更多更广泛的功能验证和全系统的性能测试.以 OS 启动为例,只需要 40 min 可以完成 Linux 启动过程,而这需要几百亿个时钟周期,如果在传统基于模拟验证的方式下几乎无法在流片前完成.此外在这一阶段,发现了一些严重的设计问题(bug),并得以在流片前及时修复.

2) 流片后到芯片回来阶段(阶段 2).这一阶段一般持续约 9~16 周.在这个阶段虽然芯片已经流片,但芯片并未回来,可以在这一段较短时间内利用 FPGA 平台完成软件驱动的开发测试、芯片压力测试、性能分析等工作,并为芯片回来后的硅后验证做好准备.在我们实际项目中,在这一阶段内虽然未曾发现芯片设计问题,但却修复了多个系统软件和 I/O 驱动问题,整体贡献了约 10% 验证成果.

3) 硅后验证阶段(阶段 3).这一阶段芯片已经回来,大量的软硬件测试可以在芯片上直接进行, FPGA 平台主要提供调试支持性工作.但从我们的项目经验看,这一阶段 FPGA 平台却提供了约 25% 的验证成果.在硅后验证发现的问题中,其中有 3 个极难在软件模拟仿真中被复现,最终都是通过 FPGA 平台成功复现,并采用第 4 节介绍的虚拟逻辑分析仪 VLA,获取到指令执行序列和错误波形,找出了问题的根源.并基于这些信息,才构造了模拟仿真测试用例,用于芯片 ECO(engineer change order) 修复验证和后续项目的回归测试集.

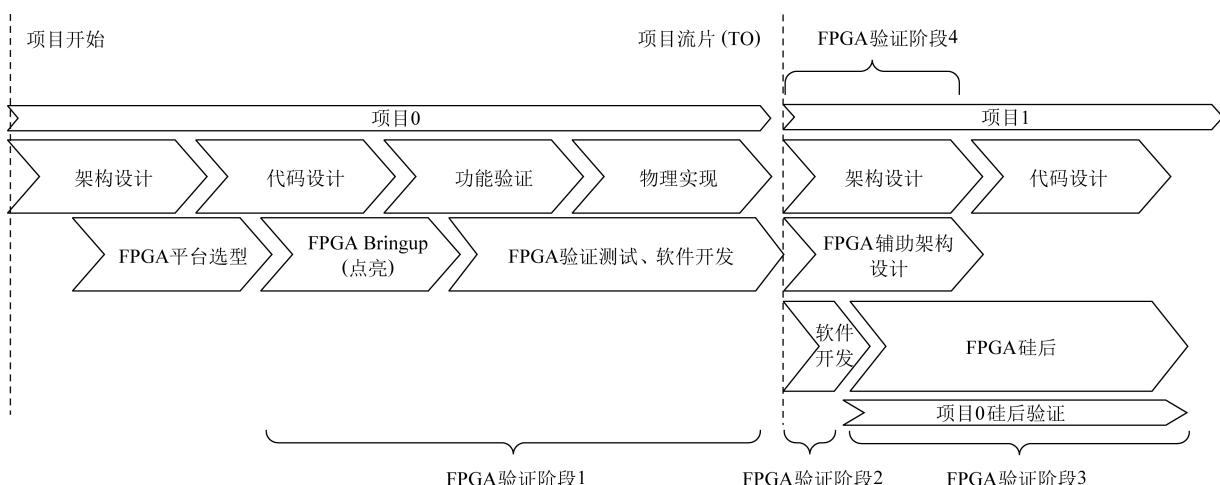


Fig. 8 Stage of FPGA verification

图 8 FPGA 各验证阶段示意图

4) 下一代芯片定义阶段(阶段4),由于同系列芯片的设计和验证天然具有继承延续性,当需要设计芯片下一代时,架构设计者经常需要快速评估一些特性的价值,FPGA平台由于其灵活可配置、迭代运行速度快等优势,能够有效地用于下一代芯片的评估分析中.可以在FPGA平台上快速修改部分处理器实现参数,例如增大或减小处理器一级、二级,三级缓存容量,调整发射队、提交队列深度、DDR 访存延迟、I/O 延迟等,这对于架构定义和性能优化都十分有益.在我们的处理器研发过程中,这一阶段在整个FPGA的生命周期中贡献了约15%的成果.

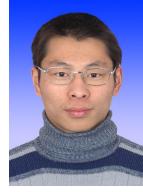
7 总 结

本文针对一款高性能异构多核处理器开展FPGA原型验证工作,选取了同构对称FPGA平台对异构多核高性能处理器的进行FPGA验证.以层次化的方法自顶向下划分FPGA功能,自底向上构建FPGA平台,极大地简化了FPGA构建流程.采用差速桥和自适应时延调整采样等技术,结合自研的内嵌虚拟逻辑分析仪调试工具快速完成FPGA平台的点亮和部署.运用多核互补、核间替换模拟的调试SHELL等方法可以快速完成目标高性能异构多核处理器的FPGA验证.因此,本文提供针对异构多核处理器的FPGA验证平台搭建和验证方法是可行且十分高效的,经过采用本文介绍的FPGA验证的目标芯片,达到了预期设计目标.通过该FPGA原型验证平台,成功地完成了该芯片的硅前、硅后验证,软硬件协同开发、测试、下一代芯片架构评估等工作,为该系列芯片的全周期提供了有效的平台保障,并可以推广应用于相类似的异构多核处理器的FPGA平台搭建和测试验证工作中.下一步我们将继续结合异构多核处理器架构特征,研究更有效的FPGA平台构建和调试技术,缩短平台点亮时间,不断提高FPGA平台的调试和验证能力.

参 考 文 献

- [1] Guo Yang, Li Sikun, Qu Wanxia. Verification of on-chip multi-core processor: Challenges, status, forecasts [J]. Journal of Computer-Aided Design and Computer Graphics, 2012, 24(12): 1521-1532 (in Chinese)
(郭阳, 李思坤, 屈婉霞. 片上多核处理器验证: 挑战、现状与展望[J]. 计算机辅助设计与图形学学报, 2012, 24(12): 1521-1532)
- [2] Evans A, Silburt A, Vrckovnik G, et al. Functional verification of larger ASICs [C] //Proc of the 35th Design Automation Conf. New York: ACM, 1998: 650-655
- [3] Beltrame G, Bolchini C, Fossati L, et al. ReSP: A non-intrusive transaction-level reflective MPSoC simulation platform for design space exploration [C] //Proc of 2008 Asia South Pacific Design Automation Conf. New York: ACM, 2008: 673-678
- [4] Ganapathy G, Narayan R, Jorden G, et al. Hardware emulation for functional verification of K5 [C] //Proc of the 33rd Design Automation Conf. New York: ACM, 1996, 315-318
- [5] Zhang Heng, Shen Haihua. Function verification of Godson2 processor [J]. Journal of Computer Research and Development, 2006, 43(6): 974-979 (in Chinese)
(张珩, 沈海华. 龙芯2号微处理器的功能验证[J]. 计算机研究与发展, 2006, 43(6): 974-979)
- [6] Gateley J, Blatt M, Chen D, et al. UltraSPARC-I emulation [C] //Proc of the 32nd ACM/IEEE Design Automation Conf. New York: ACM, 1995: 13-18
- [7] Ray J, Hoe J C. High-level modeling and FPGA prototyping of microprocessors [C] //Proc of the 11th ACM/SIGDA Int Symp on Field Programmable Gate Arrays(FPGA'03). New York: ACM, 2003: 100-107
- [8] Gschwind M, Salapura V, Maurer D. FPGA prototyping of a RISC processor core for embedded applications [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2001, 9(2): 241-250
- [9] Roland E, Wunderlich, Hoe J C. In-system FPGA prototyping of an Itanium microarchitecture [C] //Proc of IEEE Int Conf on Computer Design (ICCD'04). Piscataway, NJ: IEEE, 2004: 288-194
- [10] Lu S, Yiannacouras P, Kassa R, et al. An FPGA-based Pentium in a complete desktop system [C] //Proc of the 15th ACM/SIGDA Int Symp on Field Programmable Gate Arrays (FPGA'07). New York: ACM, 2007: 53-59
- [11] Wang P, Collins J, Weaver C, et al. Intel atom processor core made FPGA-synthesizable [C] //Proc of the 17th ACM/SIGDA Int Symp on Field Programmable Gate Arrays (FPGA'09). New York: ACM, 2009: 209-218
- [12] Hu Weiwu, Wang Huandong, Gao Xiang, et al. A multi-FPGA based platform for emulating a 100M-transistor-scale processor with high-speed peripherals [C/OL] //Proc of the 18th ACM/SIGDA Int Symp on Field Programmable Gate Arrays (FPGA'10). New York: ACM, 2010 [2020-03-17]. <https://dl.acm.org/doi/abs/10.1145/1723112.1723160>
- [13] Zhu Ying, Chen Cheng, Xu Xiaohong, et al. Design and implementation of FPGA verification platform for multi-core processor [J]. Journal of Computer Research and Development, 2014, 51(6): 1295-1303 (in Chinese)
(朱英, 陈诚, 许晓红, 等. 一款多核处理器FPGA验证平台的设计与实现[J]. 计算机研究与发展, 2014, 51(6): 1295-1303)

- [14] Xilinx. Xilinx virtex ultrascale+ FPGA product brief [OL]. [2019-01-18]. <https://www.xilinx.com/support/documentation/product-briefs/virtex-ultrascale-product-brief-v2.pdf>
- [15] Li Xiaobo, Zhang Heng, Zhang Fuxin, et al. FPGA verification for a kind of complicated chip [J]. Computer Engineering, 2006, 32(14): 243-245 (in Chinese)
(李小波, 张珩, 张福新, 等. 一类复杂芯片的 FPGA 验证 [J]. 计算机工程, 2006, 32(14): 243-245)



Li Xiaobo, born in 1982. PhD candidate. His main research interests include computer architecture, processor design and verification, hardware acceleration and FPGA verification.
李小波, 1982 年生. 博士研究生. 主要研究方向为计算机体系结构、处理器设计与验证、硬件加速与 FPGA 验证.



Tang Zhimin, born in 1966. PhD, professor, PhD supervisor. His main research interests include high performance computer architecture, processor design and digital signal processing.

唐志敏, 1966 年生. 博士, 研究员, 博士生导师. 主要研究方向为高性能计算机体系结构、处理器设计、数字信号处理.



Li Wen, born in 1976. PhD, associate professor. His main research interests include computer architecture, SOC design and verification.

李文, 1976 年生. 博士, 副研究员. 主要研究方向为计算机体系结构、SOC 设计与验证.