

基于硬件仿真器的 PCIe 接口验证方法的研究和实现

郝 强

(上海高性能集成电路设计中心, 上海 201204)

摘要: PCIe 接口是 System on Chip (SoC) 芯片上使用非常广泛的一种高速接口。因此, 在 SoC 芯片的 Register Transfer Level (RTL) 级设计开发阶段, 对 PCIe 接口设计的验证显得尤为重要, 需要通过不同的验证平台保证 PCIe 接口设计的功能正确性和性能稳定性。对基于 Cadence 硬件仿真器创建的 PCIe 接口验证平台的方法进行研究, 并在某款 SoC 芯片上实现了该验证流程。实践表明, 使用该方法能够较快速地构建验证平台, 提供较高的仿真测试性能, 同时支持多种调试手段, 有效地完成验证目标。

关键词: 硬件仿真; PCIe; 集成电路验证

中图分类号: TN47

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.209806

中文引用格式: 郝强. 基于硬件仿真器的 PCIe 接口验证方法的研究和实现[J]. 电子技术应用, 2020, 46(8): 77-79.

英文引用格式: Hao Qiang. Research and implementation of verification method for PCIe interface based on emulator[J]. Application of Electronic Technique, 2020, 46(8): 77-79.

Research and implementation of verification method for PCIe interface based on emulator

Hao Qiang

(Shanghai Hi-Performance IC Design Center, Shanghai 201204, China)

Abstract: The PCIe interface is a kind of high-speed interface widely used on system on chip (SoC). In the register transfer level (RTL) design and development stage of the SoC, it is particularly important to verify the design of the PCIe interface. Different verification platforms are needed to ensure the functional correctness and performance stability of the design. This paper focuses on the method of PCIe interface verification platform based on Cadence emulator, and realizes the verification method with a chip. The practice shows that this method can quickly build the verification platform, provide high emulation performance, support a variety of debugging methods, and effectively complete the verification objectives.

Key words: emulation; PCIe; integrated circuit verification

0 引言

随着集成电路设计技术的不断发展, SoC 芯片的功能和性能得到极大的丰富和提升。其中, 总线技术的发展起着重大的推动作用。PCIe 总线作为系统总线的延伸, 使得 SoC 芯片可以与外部高速设备相连, 能够完成多种应用功能扩展^[1]。因此, PCIe 总线接口成为 SoC 芯片设计中最为常用的外设接口之一。然而, PCIe 总线协议相较于一般慢速的总线协议更为复杂, 这就给 PCIe 接口设计的正确性带了挑战。为此, 一般会寻求多种验证方法和手段来力求全面地保证 PCIe 接口设计的正确性。综合验证平台的高效性和验证场景的多样性考虑, 采用基于 Cadence 硬件仿真器平台的 PCIe 接口的验证方法突出了开发周期短、运行速度快以及调试手段多的优势, 日益成为 PCIe 接口设计验证最为重要的验证方法之一。本文描述了基于 Cadence 硬件仿真器对某款

SoC 芯片的 PCIe 接口设计进行验证的实现方法和验证环境的创建流程, 用以满足该设计的验证需求。

1 待测设计简介

本待测设计是一款 SoC 芯片, 集成多种 I/O 接口, 其中 PCIe 接口符合 PCIe 3.0 链路标准, 支持 $\times 4$ 、 $\times 8$ 和 $\times 16$ 灵活配置, 该设计接口配置为根部件 (Root Complex, RC) 工作模式。其结构如图 1 所示。该待测设计的核心通过路

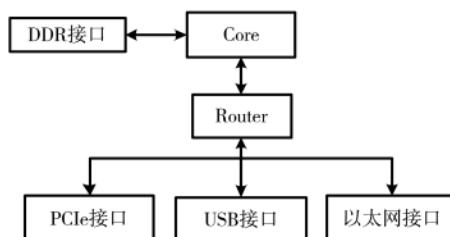


图 1

由部件与 PCIe 接口部件相连,可以实现访存和中断等请求。

2 传统的验证方法

为保证待测设计正确性,传统的验证方法包括软件模拟验证(Simulation)和 FPGA 原型验证(FPGA Prototyping)。

软件模拟验证方法,是通过构建软件仿真验证环境,使用软件模拟器 Simulator(例如 NC-sim)加载 RTL 级设计和 TestBench,并进行编译和施加测试用例来完成功能模拟。对于集成了 PCIe 接口的设计,还需要使用相应的 VIP(Verification IP)作为 EP(End Point Device),用来模拟实现访存和中断请求等验证。由于这些验证行为都是运行在软件层面,能够方便地完成波形数据收集和查错,但是,一旦待测设计规模较大,整个验证环境的运行速度下降明显,施加一个测试激励往往需要几个小时甚至数十个小时,运行的时间开销是难以忍受的。

FPGA 原型验证方法,是通过将 RTL 级设计使用 FPGA 资源实现,构建 FPGA 原型验证平台,进行实物验证。这种验证方法,运行速度快是其最大的优势,其运行频率最高能达到几十兆赫兹,比较适合进行连接真实设备对外设接口进行压力和稳定性验证。但是,FPGA 原型验证平台开发和调试周期较长,往往只能在芯片开发的中后期才能投入使用。而且,根据 FPGA 实现的一些特殊性,一般需要对 RTL 设计进行适应性修改。另一个重点不足在于调试手段有限,设计运行过程中一旦出现问题难以快速定位。

因此,对于本文的待测设计的验证,尤其是 PCIe 接口验证不能只依赖于软件模拟验证和 FPGA 原型验证,还需要寻求其他更加有效、便于调试的验证方法。

3 基于硬件仿真器的验证方法

硬件仿真器验证技术,相对传统验证技术形成较晚,但随着各大 EDA 公司大力地投入研发和推广,已经成为一种成熟的验证技术,受到行业内的接受和认可,这其中,以 Cadence 公司的 Palladium 系列的硬件仿真器产品的表现最为突出。基于硬件仿真器的验证方法,相较于软件模拟验证,具有更快的验证速度和更真实的外设测试平台。而相较于 FPGA 原型验证,它具有更短的开发周期和更丰富的调试手段。因此,本文所提的待测设计在研发周期内,开发了多种基于硬件仿真器的验证平台,运用于系统软件调试、应用课题仿真和外设接口验证等。

本文将研究基于硬件仿真器验证平台在对 PCIe 接口设计进行验证中所采用的验证方法。

3.1 验证平台的整体结构

如前所述,PCIe 接口属于高速总线接口,就 PCIe3.0 而言,可以支持 2.5 Gb/s、5.0 Gb/s 和 8.0 Gb/s 的传输速率,要直接在如此高的传输速率下进行仿真验证,对任何验证平台来说,都是一个不小的挑战。以 Cadence 公司 Palladium Z1 硬件仿真器为例,尽管其最高仿真运行频率可以达到 4 MHz^[2],但仍然不能直接满足 PCIe 接口的标准要求。为此,Palladium 平台提供了一套解决方案,通

CadenceLIVE 2020 中国用户大会优秀论文

过使用 SpeedBridge 进行了快慢两端的速率适配。基于该解决方案,需要对处理器的 PCIe 接口部件进行修改。

原始设计的 PCIe 接口应用场景是 PCIe Core 通过 PIPE 接口连接 PCIe PHY,再通过其高速串行接口连接板级的 PCIe 插槽,最终与 PCIe EP 真实设备进行连接。

由于 PCIe PHY 不可综合,不能直接加载到硬件仿真器上运行,因此,解决方案中提供了 SpeedBridge Wrapper 模块来进行替换,该 Wrapper 模块使用可综合代码实现,将其例化在处理器的 PCIe 接口设计中,其一端通过 PIPE 接口与设计进行连接,另一端输入输出端口映射到硬件仿真器的外部硬件接口引脚,再通过连接器电缆与 PCIe SpeedBridge 相连,PCIe SpeedBridge 与 PCIe EP 设备安插到支持两个 PCIe 插槽的第三方底板上。这样就构建出了基于硬件仿真器的 PCIe 接口设计的验证平台。其中,带 PCIe 接口的待测设计与 SpeedBridge Wrapper 运行在硬件仿真器上,以低速率通过 PCIe SpeedBridge 进行速率转换,实现了与外部高速率 EP 设备的通信。

在此解决方案中,运行在硬件仿真器上的 PCIe 接口设计可以与 SpeedBridge Wrapper 间使用 2.5 Gb/s、5.0 Gb/s 及 8.0 Gb/s 的传输速率进行仿真,因此,PCIe 接口设计可以得到充分的验证。需要指出的是,PCIe SpeedBridge 与外部 PCIe EP 设备只运行在 2.5 Gb/s 的速率,但这已经足够了。

3.2 验证环境内部实现

前面提到,运行在硬件仿真器上的 PCIe 接口设计,需要通过 SpeedBridge Wrapper 作为中间模块,与外部 SpeedBridge 进行连接,本节内容将进一步探究其实现方法。

PCIe 接口设计与 PCIe SpeedBridge Wrapper 在本待测设计中都是使用 Verilog 编写,只需要按照模块间的一般连接方式实现即可。需要注意的是,这两者应该直接通过 PIPE 接口直连。

SpeedBridge Wrapper 与硬件仿真器的外部硬件 Pin 脚是通过 terminal 映射方式,这就需要先将 Wrapper 的输入输出端口连接到验证环境顶层。通常的做法是,将 Wrapper 的端口类似待测设计的子模块端口一样,自下向上一层一层地传递后,连接到最外层,作为验证环境顶层的输入输出。然而,这种做法比较繁琐,需要大量修改待测设计各个关联层次的端口数量,而且,也会在一定程度上增加了待测设计代码的后期修改维护的工作量。因此,硬件仿真器提供了另一种实现方式,即在预编译之前,通过 globalNet 命令将需要引出的内部信号声明为 global net,这样就可以在环境顶层直接使用,省去了模块间的穿透,这大大减少了代码修改的工作量,这是比较推荐的做法。

定义了 globalNet 之后,就可以通过 cableConnection 指定硬件仿真器与外部 Target 系统的连接接口类型,即与外部 PCIe SpeedBridge 连接的硬件接口类型,在 Palladium Z1 硬件仿真器上,需要使用 TPOD 接口。

选择硬件接口之后,就需要对 terminal 进行物理关系映射,同时,设置合适的时序约束。

如图 2 所示,在 targetLocation 配置时,除了首选的 TPOD 资源配置之外,可以额外配置几组以备选,便于在运行阶段,按照可用资源进行灵活选择,避免了再次分配资源进行重复编译。

在完成了 PCIe SpeedBridge 相关的硬件接口配置之后,就可以进行硬件仿真器验证环境的编译和后续运行了。需要注意的是,工作模式需要设置为电路仿真 (In-Circuit Emulation, ICE) 模式。

3.3 验证调试方法

PCIe 设计接口的验证需要面向多种应用场景,这也为查错和调试带来难题。基于硬件仿真器的验证平台,在为用户提供较高的工作频率下,也为用户开发多种调试手段成为可能。

在该待测设计中,主要经历了三种调试阶段。

(1) 基于操作系统内核和用户测试程序信息的粗粒度调试阶段。正是由于硬件仿真器的运行速度快的优势,使得在处理器设计开发初期,就可以运行操作系统和用户测试程序。这样,通过操作系统内核信息和用户测试程序调试信息就可以判断出错位置,为后续调试指明方向。这个阶段可称之为粗粒度的调试阶段。

(2) 基于状态描述的动态监测中粒度调试阶段。状态描述语言(State Description Language,SDL)是 Palladium 硬件仿真器所支持的调试脚本语言,它可以调用相关命令控制硬件仿真器的运行和交互^[2],比如,暂停待测设计的运行、设置触发条件、捕获波形以及显示内部设计信号值等。通过 SDL 实现的监测器,可以实时监测 PCIe 设计接口重要的内部状态和事务数据报文,便于进一步问

```
targetLocation -add PCIe_1T3_1T4 PCIeG3 emu_host {1_T3 1_T4} -info {First PCIe SB instance}
targetLocation -add PCIe_1T0_1T1 PCIeG3 emu_host {1_T0 1_T1} -info {alternate connection1}
targetLocation -add PCIe_1T2_1T3 PCIeG3 emu_host {1_T2 1_T3} -info {alternate connection2}
targetLocation -add PCIe_1T1_1T4 PCIeG3 emu_host {1_T1 1_T4} -info {alternate connection3}
```

图 2 target 配置多选示例

```
Time == 96327.995842 ns PCIE3 SB LTSSM ==> 02
Time == 96327.995842 ns PCIE3 Slow Side Link Status: Trying to Establish link with the DUT: Polling .....
Time == 129215.874782 ns PCIE3 SB LTSSM ==> 04
Time == 129979.929044 ns PCIE3 SB LTSSM ==> 05
Time == 130207.81474 ns PCIE3 SB LTSSM ==> 06
Time == 130879.852656 ns PCIE3 SB LTSSM ==> 07
Time == 130911.899082 ns PCIE3 SB LTSSM ==> 08
Time == 130947.881034 ns PCIE3 SB LTSSM ==> 09
Time == 131555.826098 ns PCIE3 SB LTSSM ==> 0a
Time == 131671.830412 ns PCIE3 SB LTSSM ==> 10
Time == 131671.830412 ns PCIE3 Slow Side Link up : Gen1 @ 2.5 Gbps
Time == 133991.916692 ns PCIE3 SB LTSSM ==> 0c
Time == 134152.898446 ns PCIE3 SB LTSSM ==> 0b
.....
Time == 134580.93375 ns PCIE3 SB LTSSM ==> 29
Time == 134588.804802 ns PCIE3 SB LTSSM ==> 0e
Time == 134620.851228 ns PCIE3 SB LTSSM ==> 10
Time == 134620.851228 ns PCIE3 Slow Side Link Up: Gen3 @ 8.0 Gbps
```

图 3 SDL 状态监测示例

CadenceLIVE 2020 中国用户大会优秀论文

题定位。但受硬件资源限制,监测的信号量还是有限,因此,这个阶段称之为中粒度的调试阶段。SDL 状态监测示例如图 3 所示。

(3) 基于波形分析的细粒度调试阶段。通过前两个粗粒度、中粒度的调试阶段,问题已经能够锁定一个出错区间了。此时,可以在硬件仿真器验证环境下设置触发条件,捕获波形,来精准分析出错现场。支持全波形记录,使得硬件仿真器在调试方法上能与软件模拟验证相媲美,能够完成细粒度的分析调试。

4 结论

通过使用基于 Cadence 硬件仿真器进行 PCIe 接口验证的解决方案,大大缩短了待测设计 PCIe 接口验证平台的开发周期,基本上能与软件模拟环境开发保持同步,而相比软件模拟验证的运行速度,该验证平台可以提供近 1 000 倍的验证加速。同时,在设计初期就能够开展驱动软件层面的调试,为后期的 FPGA 原型验证减少了软件调试时间,对整个设计开发和验证工作的推动效果显著。当然,本文中对基于硬件仿真器的 PCIe 接口验证方法的研究还比较粗浅,需要进一步深挖和改进,以待能全部发挥 Cadence 硬件仿真器的优秀性能。

参考文献

- [1] 王奇.PCI Express 体系结构导读[M].北京:机械工业出版社,2010.
- [2] VXE user guide(Product Version 15.1)[EB/OL].(2016-06-XX).<http://www.cadence.com>.

(收稿日期:2020-06-22)

作者简介:

郝强(1988-),男,本科,工程师,主要研究方向:集成电路系统级设计与验证。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所