doi:10.11887/j.cn.201803027 http://journal.nudt.edu.cn

众核处理器访存链路接口的FPGA验证*

周宏伟[1],徐 实[2],王忠奕[1],杨乾明[1],冯权友[1],邓让钰[1],窦 强[1]
(1. 国防科技大学 计算机学院,湖南 长沙 410073;2. 湖南大学 信息科学与工程学院,湖南 长沙 410082)

摘 要:面向众核处理器提出一种访存链路接口的现场可编程门阵列(Field-Programmable Gate Array,FPGA)验证平台,用于对处理器访存链路关键部件进行功能及可靠性测试。提出片上读写激励自动产生与检查机制、以太网接口硬件用户数据报协议(User Datagram Protocol,UDP)协议栈和FPGA芯片间多通道并行链路三项关键技术并进行设计实现。实验结果表明提出的各项关键技术功能正确,不仅丰富了功能验证中随机激励产生及结果验证的手段,而且实现了对链路数据检错和多lane间延迟偏斜纠正逻辑的可靠性测试与评估。经过该平台验证的访存链路接口在实际芯片中通过了功能正确性测试,证明了验证的有效性。

关键词:众核处理器;访存;链路;现场可编程门阵列;验证
中图分类号:TP302.1 **文献标志码**:A **文章编号**:1001-2486(2018)03-176-07

FPGA verification for memory link interface of many-core processor

ZHOU Hongwei[1], *XU Shi*[2], *WANG Zhongyi*[1], *YANG Qianming*[1], *FENG Quanyou*[1], *DENG Rangyu*[1], *DOU Qiang*[1]
(1. College of Computer, National University of Defense Technology, Changsha 410073, China;
2. College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China)

Abstract: An FPGA(field-programmable gate array) verification platform for memory link interface of many-core processor was proposed to test the function and reliability of the main components of the processor's memory access link. Three key technologies, the on-chip read-write requests automatic generation and result checking mechanism, the hardware UDP(user datagram protocol) protocol stack in Ethernet interface and the multi-lane parallel link between FPGA chips were proposed and implemented. Experiments on the platform show that the proposed technologies are correct, they not only enrich the ways of the random request generation and result checking for functional verification, but also can test and evaluate the logics of link errors detection and lane-to-lane deskew. The proposed platform has been used to verify a real many-core processor in which the function of memory link interface is correct, so the validity of the verification is proved.

Key words: many-core processor; memory access; link; field-programmable gate array; verification

为了提高仿真速度,目前被广泛采用的针对大规模专用集成电路(Application Specific Integrated Circuit, ASIC)芯片的验证手段有两种:一种是基于专用处理器的仿真加速器[1-2],另一种是基于现场可编程门阵列(Field-Programmable Gate Array, FPGA)的硬件原型平台[3]。基于专用处理器的仿真加速器既具有软件模拟的灵活性又具有硬件加速的高性能,与软件模拟相比,其模拟速度能够提高10^3~10^5倍,通常用于芯片流片前的设计验证和软件验证,能够节省大量开发时间[4-5]。基于FPGA的硬件原型平台能够获得兆赫兹(MHz)级别的寄存器传输级仿真速度,能够在最接近真实逻辑上进行系统级验证[6-7]。

众核处理器丰富的片上计算资源对访存带宽提出了更高的需求。由于处理器片上面积和功耗受限,将最后一级Cache(Last Level Cache,LLC)以及存储控制器放置在片外,通过访存链路接口进行内存扩展成为趋势。IBM Power8处理器[8]的访存通过串行链路连接4个片外存储器缓冲芯片(Memory Buffer Chip, MBC),实现了16个DDR3-1600访存通道。每个MBC上集成增强动态随机存取存储器(enhanced Dynamic Random Access Memory, eDRAM)作为LLC,缓存来自内存的数据以提高访存性能。美国镁光公司提出了混合存储立方(Hybrid Memory Cube,HMC)存储器[9],将DRAM存储器与控制芯片通过三维堆叠工艺封装在一起,通过串行链路和处理器相连。考虑到访存链路传输的报文以短报文为主、对传

* **收稿日期**:2016-12-31
基金项目:国家自然科学基金资助项目(61303069,61472432,61602498);核高基重大专项基金资助项目(2015ZX01028101)
作者简介:周宏伟(1980—),男,陕西宝鸡人,副研究员,博士,E-mail:zhou.hongwei@139.com

输延迟要求高的特点,其课题组提出了一种基于多通道并行技术的访存链路接口(Multi-lane Parallel Memory Link Interface,MP-MLI):8 ~ 10 位数据与一个时钟信号作为一个通道(lane),在时钟的上升沿和下降沿控制下进行双倍数据速率(Double Data Rate,DDR)传输,多个并行通道通过延迟校准后同步传输数据。每个 lane 内部采用源同步传输技术,不需要串行链路的时钟编解码过程,在满足传输带宽的前提下降低传输延迟。

由于 FPGA 平台能够构造真实的板级通信环境以仿真芯片之间的互联,因此非常适合对访存链路接口进行原型验证。针对众核处理器访存链路接口的特点,本文提出了一种针对多通道并行访存链路接口 MP-MLI 的 FPGA 原型验证平台,仿真速度达到了 100 MHz,能够与仿真加速器平台互为补充,充分验证芯片间的高速链路接口的各种功能,有效加速验证速度。

1　存储控制芯片总体结构

在集成 64 核心的众核处理器中,为了满足计算对访存带宽的需求,处理器分为集成处理器核心、片上网络的主处理器芯片与 8 个片外存储控制芯片(Memory Control Chip,MCC)。主处理器芯片和每个 MCC 之间通过 MP-MLI 进行互联通信,如图 1 所示。每个 MCC 集成两个 DDR3 - 1600 控制器和 16 MB 的 L3Cache。MCC 中的访存请求首先经过 L3Cache 流水线,如果命中,则可以直接从 L3Cache 中返回数据;如果失效,则需要访问两个存储控制器中的某一个,通过片外双列直插存储模块(Dual-Inline-Memory-Module,DIMM)存储器获得所需数据。为了减少 MCC 的输入/输出(Input/Output,IO)引脚数,同时保证互连带宽和访存带宽匹配,主处理器与 MCC 之间构建两条非对称的独立物理通路:命令通路和数据通路。主处理器集成存储调度单元(Memory Schedule Unit,MSU)和主访存链路接口(MLI_Master),MSU 负责接收来自处理器芯片中片上网络的访存请求,进行一定的读写调度优化,然后通过命令通路发送访存命令到 MCC(对于写请求还携带写数据)。MCC 集成从访存链路接口(MLI_Slave),通过数据通路返回读响应数据给主处理器。命令和数据通路均使用多通道并行的方式进行位宽扩展。为了降低链路延迟,每个 lane 采用基于源同步技术的 8 位并行总线构建。命令通路使用 4 个 lane,数据通路使用 8 个 lane。访存链路接口单元包含链路层和物理层两个层次:链路层完成链路层报文的发送和接收、链路层报文的流控、循环冗余校验(Cyclic Redundancy Check,CRC)和报文出错重传机制;物理层完成物理链路的管理、物理微包的 DDR 方式的发送和接收、lane 与 lane 之间的延迟偏斜校准(lane-to-lane deskew)以及 DDRIO 单元根据功耗、电压和温度变化的自校准功能等。

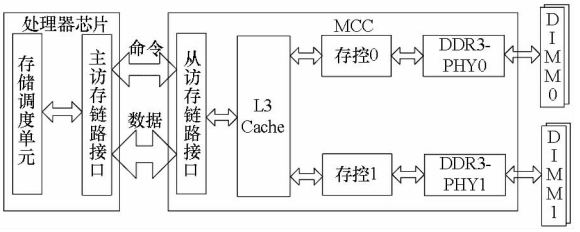


图 1　MCC 总体结构图

Fig. 1　Overall architecture of MCC

2　面向访存链路接口的 FPGA 验证平台

基于具有 4 颗 Xilinx V7 芯片的 FPGA 开发板构建验证平台。该平台集成的 FPGA 芯片内部包含 DDR3 控制器和对应的物理层(PHYsical layer, PHY)、千兆以太网控制器、IO 缓冲(buffer)和 IO 延迟单元等。

2.1　总体结构

为了仿真真实的芯片间通信,验证平台采用了两块 FPGA 芯片:一片为主 FPGA 芯片,构建仿真主处理器的芯片,内部集成以太网控制器、自动读写激励发生器、返回数据自动比较器等和用于配置、激励产生和结果检查的联合测试工作组(Joint Test Action Group, JTAG)标准协议接口逻辑,还集成了需要被测试的 MSU 和 MLI_Master;另一片为从 FPGA 芯片,构建仿真 MCC 的芯片,内部集成 MLI_Slave、L3Cache 模型、DDR3 存储控制器 IP 和对应的 PHY IP 等。L3Cache 逻辑并没有包含在从 FPGA 芯片中,因为 L3Cache 的存储体太大,一片 FPGA 无法完全放入,另外 L3Cache 的功能能够在仿真加速器平台中得到验证,因此不放入 FPGA 平台能够简化验证环境、缩短验证时间。验证平台总体结构如图 2 所示。

主 FPGA 芯片中的 MSU、MLI_Master、I2C 主控制器、链路 IO 和 I2C IO 是被测逻辑,MSU 负责访存请求调度,MLI_Master 负责链路管理和数据传输,I2C 主控制器负责通过 I2C 接口访问 MCC 芯片的寄存器。

为了产生读写请求激励,设计了两种激励产生方式,一种比较灵活的方式是通过测试计算机

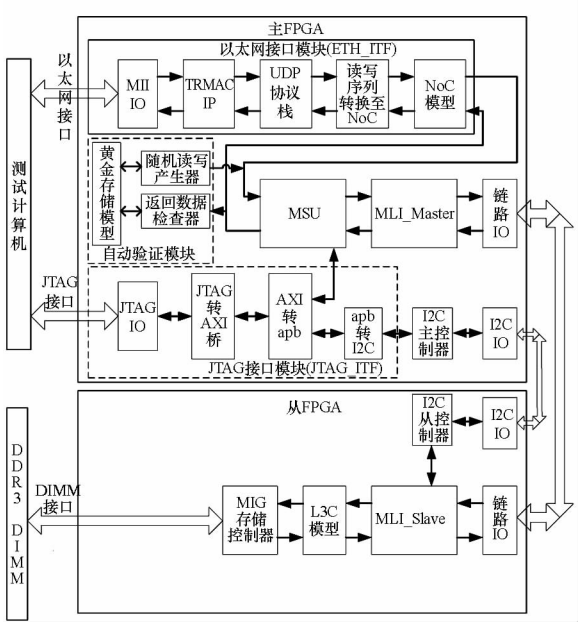


图 2 FPGA 验证平台总体结构图

Fig. 2 Overall structure of FPGA verification platform

按照测试意图产生读写请求序列,通过以太网接口以用户数据报协议(User Datagram Protocol, UDP)报文的方式发送到主 FPGA。主 FPGA 内部设计有以太网接口模块(ETH_ITF),该接口逻辑包含:介质无关接口(Media Independent Interface, MII)IO、三态以太网控制器(TRMAC IP)、UDP 协议栈(负责解析网络中的 UDP 报文并转换为读写序列)、读写序列转换至片上网络(Network on Chip, NoC)报文格式模块和 NoC 模型(负责将读写报文发送至 MSU)。测试计算机通过以太网和主 FPGA 相连,模拟处理器核发送读写请求序列,并将这些序列打包成 UDP 报文,通过以太网传输到主 FPGA 芯片。主 FPGA 接收到 UDP 报文后,通过解析出读写请求序列,并转换为 NoC 报文格式,通过 NoC 模型发送到被测的 MSU 模块。读响应数据同样通过以太网传回到测试计算机,测试计算机对返回的数据的正确性进行验证。为了实现被测逻辑中配置寄存器和状态寄存器的读写,主 FPGA 芯片中设计有 JTAG 接口模块(JTAG_ITF),该接口逻辑包含多个协议转换桥,用于读写主 FPGA 及从 FPGA 芯片中相关部件的寄存器。

另一种产生激励的方式不需要测试计算机,激励的产生和返回数据的检查均通过 FPGA 内部集成的自动验证模块完成,该模块包括:随机读写产生器、返回数据检查器和黄金存储模型(Golden Memory)。其中 Golden Memory 主要负责记录所有随机产生的读写请求地址对应的准确数据,用于比较从被测逻辑返回的读数据和记录的数据,以判断返回数据的正确性。由于 FPGA 内部的静态随机存储器(Static Random-Access Memory, SRAM)资源受限,因此 Golden Memory 能够记录的数据量有限,随机读写产生器随机产生的地址数目也有限。这种激励产生与结果比较的方法不依赖于外部测试计算机,也不需要以太网接口模块,逻辑规模小,测试速度快,但是被测地址数量受限,因此需要和前一种方法结合使用,进一步提高验证覆盖率。

从 FPGA 芯片主要用于验证 MCC 的关键逻辑,包括从访存链路接口 MLI_Slave、I2C 从控制器、链路 IO 和 I2C IO。由于 FPGA 片上 SRAM 资源受限,L3C 模型用于模拟 L3Cache 的基本数据通路,不具备缓存数据的功能。存储控制器由 Xilinx V7 芯片提供的存储接口产生器(Memory Interface Generator, MIG)生成,通过 FPGA 芯片集成的 PHY 连接开发板上的 DDR3 DIMM 内存。由于每个 FPGA 芯片仅支持一个访存通道,因此将原 MCC 中的存储控制器从双通道简化为单通道。

2.2 片上读写激励自动产生与检查

FPGA 片上集成的读写激励自动产生与检查机制,不仅能够产生随机激励,还能够对返回结果进行检查以判断返回数据的正确性。该机制的核心是黄金存储模型,该模型保存被测地址的最新访存数据以及每一个未完成的读请求应当返回的正确数据。随机地址产生器负责产生随机地址,由于 MSU 最多支持 16 个未完成的读请求,因此使用 4 位的 ID 号用于标识已发出但未完成的读请求。一旦某个 ID 号的读响应返回,则释放该 ID 号为空闲状态,可以分配给下一个请求。16 个地址寄存器用于记录对应的第 0 个到第 15 个 ID 号的读写请求的地址,地址 0 保存 ID 号为 0 的读写请求,依次类推。与 16 个地址对应的有 16 个数据队列,每个队列深度为 16,能够保存对应地址的 16 个未完成的读请求。每个数据队列的写入数据来自于与之一一对应的“最新数据寄存器”,每个最新数据寄存器保存对应 ID 号的地址的最新数据。

2.3 以太网接口硬件 UDP 协议栈

为了能够全流水化处理以太网报文以提高协议栈的性能,本文设计实现了一种硬件 UDP 协议栈,不需要软核即可实现与测试计算机之间的高带宽网络通信,其结构如图 3 所示。

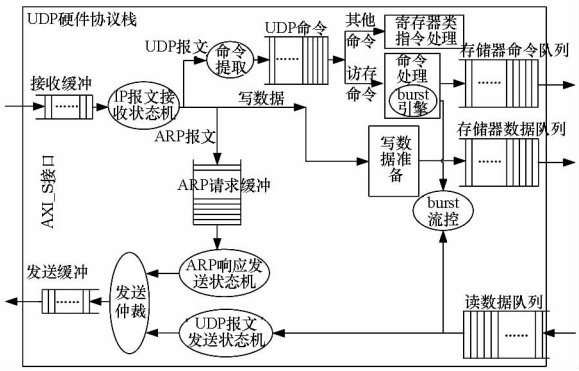


图 3　硬件 UDP 协议栈总体结构图

Fig. 3　Overall structure of hardware UDP protocol stack

主 FPGA 芯片通过硬件逻辑支持网络层地址解析协议(Address Resolution Protocol, ARP),用于建立测试计算机与主 FPGA 芯片之间的网络连接。TRMAC IP 核与硬件协议栈之间通过 AXI-S 接口连接。网络层 IP 报文接收状态机负责对 IP 报文进行接收和报文解析。对解析出的 ARP 报文,ARP 发送状态机负责产生 ARP 响应报文。对解析出的 UDP 报文,通过“命令提取”模块从 UDP 报文的数据负载区域获得请求信息,将请求命令缓存到 UDP 命令队列中,同时将写请求数据经过“写数据准备”逻辑缓存到写数据队列中。突发(burst)读引擎负责将突发读请求转换成连续的读命令。读响应返回的数据在 UDP 报文发送状态机的控制下发送到 AXI-S 发送先进先出(First Input First Output, FIFO)队列中。UDP 报文发送状态机支持报文分片功能,能够将 burst 长度超过 IP 报文最大长度的 UDP 报文转换为多个 IP 报文分片传输出去。

一个 UDP 负载中能够连续放置最多 64 个命令,每个命令的格式为 4 位的操作码和 64 位对齐的命令或数据,主 FPGA 依次对它们进行解析执行。操作字节定义了以下几种基本操作:0—空操作,1—更新读基地址,2—更新写基地址,3—更新读 burst 长度,4—启动 burst 读,5—寄存器读操作,6—寄存器写操作。

2.4　FPGA 芯片间多通道并行链路

众核处理器的主处理器芯片与 MCC 之间采用了多通道并行链路技术。每个 lane 包含差分 IO 实现的 clk 信号和单端 IO 实现的 8 ~ 10 个数据信号。命令通路使用 4 个 lane,数据通路使用 8 个 lane,每个通路各个 lane 之间的数据对齐采用了 lane 到 lane 的延迟偏斜纠正技术。Xilinx V7 的 FPGA 芯片中的 IO buffer 包括用于单端 IO 的 IOBUF 和用于差分 IO 的 IOBUFDS,分别用于替换原始设计中的单端和差分 DDR IO buffer。FPGA 芯片中的可调节延迟值的 IO 延迟单元能够用于替换原始设计中用于延迟偏斜纠正逻辑中粗粒度延迟调节所需的延迟单元。假设每个 lane 的工作频率为100 MHz,则可以调节的延迟值的最大值约为 1/4 个时钟周期。为了验证链路可靠性控制逻辑,在主 FPGA 芯片中增加了随机数据干扰模块,该模块能够在发送方对每个 lane 的多位数据中的某些位随机地进行翻转,在接收方检查是否能够校验出发送方注入的错误,以及链路接口中的链路重传机制能否正确重传数据、最终将正确的数据送达接收方。为了验证多个 lane 之间存在延迟偏斜时逻辑的功能正确性,在主 FPGA 芯片中为每个 lane 增加了额外的级联的寄存器站,能够对每个 lane 的所有信号统一增加 0 ~ 8 个时钟周期的延迟。

3　FPGA 验证平台搭建与实验

基于集成 4 颗 FPGA 芯片的开发板搭建提出的 FPGA 验证平台,如图 4 所示。主 FPGA 与从 FPGA 之间的链路采用集成在测试板内部的 FPGA 芯片间高速互连信号线,也可以通过测试版两侧的 Dedicated IO 使用电缆互连。控制与状态子板用于系统复位、开启错误注入功能、错误模式选择以及错误状态信息显示等功能。测试计算机通过以太网 PHY 子板上的以太网接口进行访存命令发送与结果接收,通过 JTAG 接口对主 FPGA 和从 FPGA 的配置寄存器进行访问。为了实现测试计算机通过以太网对主 FPGA 进行测试激励的发送和测试结果的正确性检查,使用

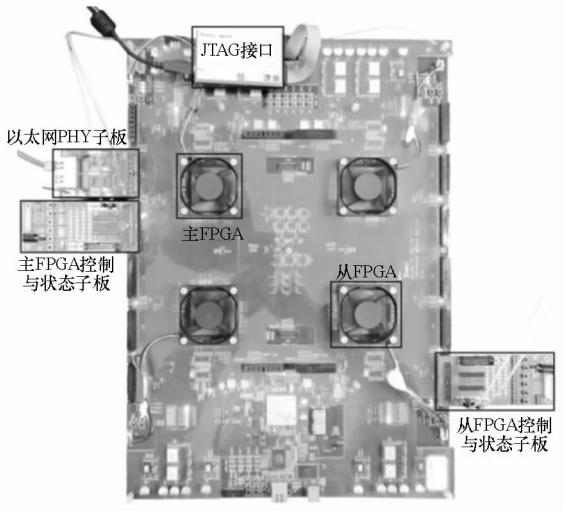


图 4　FPGA 验证平台实物图

Fig. 4　Physical map of FPGA verification platform

Python 语言开发了一套定制的存储调试软件(Custom Memory Debug Software, CMDS)。由于 Python 的基本套接字接口采用了标准库的 socket 模块,支持 TCP/IP 和 UDP 套接字类型,因此基于其开发的调试软件具有良好的跨平台特性。CMDS 支持第 2.3 节中定义的通过 UDP 报文传送的所有操作类型。

3.1 片上读写激励自动测试

在测试过程中对自动测试产生的随机地址和请求数目进行统计。模拟环境中能够设置请求发送概率:概率为 1 表示激励产生器每个时钟周期都会产生一个读写请求,只要 MSU 有信用,产生的请求就会被发送给 MSU;概率为 1/16 表示激励产生器每个时钟周期能够产生一个读写请求的概率为 1/16,如果产生读写请求的同时 MSU 有信用,才能将请求发送给 MSU。实验中,分别在两种概率下运行 400 万个时钟周期,统计总的访存数目和产生的访存地址的数目,并计算每个地址的平均访问次数,结果如表 1 所示。

表 1 读写激励自动产生的访存数和访存地址数

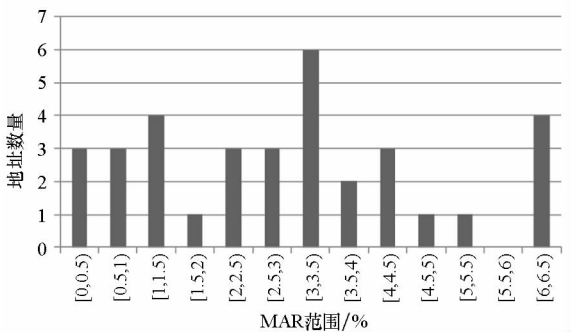
Tab. 1 Number of memory accesses and addresses generated by read/write requests automatic generator

请求发送概率	访存数目	访存地址数目	每个地址平均访问次数
1	285 927	34	8410
1/16	118 258	49 865	2.37

对于请求发送概率为 1 的情况,总的访存地址数目为 34,自动验证模块黄金存储模型中每个地址寄存器中的地址被平均更新了 2 次;每个地址被访问的平均次数为 8410,访存地址相关性很高。原因如下:当请求发送概率为 1 时,由于只要 MSU 有信用,就会发送一个读写请求到 MSU,因此,黄金存储模型中第一批随机产生的 16 个访存地址会被频繁访问,对于其中任意一个地址,在前一个读请求完成之前,又会产生对该地址的下一个读请求,因此该地址会被频繁访问,不会被新的请求地址替换。

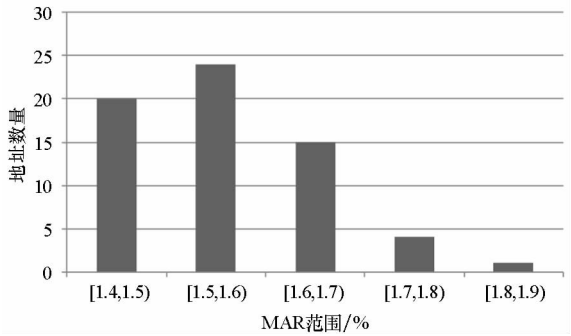
当请求发送概率为 1/16 时,访存地址的总数为 49 865,每个地址的平均访问次数为 2.37,访存地址相关性低。原因如下:当请求发送概率降低时,产生请求的速度变缓,对于一个访存地址来说,前一个访问未完成又产生对该地址的新的访问的概率大大降低,因此保存在地址寄存器中的地址被更新的频率显著提高,每个地址平均被访存的次数显著减少。请求发送概率为 1 时的访存数目约为概率为 1/16 时的访存数目的 2.4 倍,而不是 16 倍,原因是:存储器写请求报文包含 1 个命令微片和 4 个数据微片,需要 5 个时钟周期发送,因此 16 个时钟周期内,最多发送 3 个写请求;另外,MSU 对于相关地址的调度性能低于对不相关地址的调度。

定义一个访存地址的访存次数占总访存次数的比率为该地址的访存次数比率(Memory Access Ratio, MAR)。图 5 为不同请求发送概率下随机产生的各访存地址的 MAR 值的分布情况。请求发送概率为 1 时,34 个访存地址的 MAR 分布如图 5(a)所示:不同地址的 MAR 差别较大,有 3 个地址的 MAR 值小于 0.5%,4 个地址的 MAR 值大于 6%,MAR 的差异较大。当请求发送概率为 1/16时,统计了 ID 为 0 的地址寄存器前 64 个访存地址的 MAR 值,其分布如图 5(b)所示:不同地址的 MAR 差别较小,大部分地址的 MAR 值位于 1.4% ~1.7%之间,只有 1 个地址的 MAR 值达到最大的 1.8%,所有地址的最大和最小的 MAR 值



(a) 发送概率为 1 时 MAR 分布情况

(a) MAR distribution when requests sending probability is 1



(b) 发送概率为 1/16 时 MAR 分布情况

(b) MAR distribution when requests sending probability is 1/16

图 5 不同请求发送概率下 MAR 分布情况

Fig. 5 MAR distribution with different requests sending probability

的差异约为 28.6%。综上,实验结果说明通过调整请求发送概率,能够控制访存地址的数量与访问频度,实现对访存地址相关性和每个访存地址的访存次数比率分布的控制,产生不同特点的访存流以提高测试覆盖率。

FPGA 验证平台核心逻辑的工作频率达到了 100 MHz。在片上读写激励随机测试中,发现了一个自动验证模块的设计错误,在 FPGA 验证平台上复现错误的时间约为 20 s,而在软件模拟环境中花费了约 48 h,复现错误所用时间比为 1∶8640。FPGA 平台加速效果显著。

3.2　多 lane 并行链路可靠性测试

3.2.1　链路注入错误测试

为了测试链路数据被干扰导致数据翻转的情况,试验中利用位于主 FPGA 内部的数据干扰模块对每一个 lane 内部的数据线进行错误注入。为了验证数据链路传输报文头微片时的可靠性,对发生数据翻转的线的数目从 1 到 8 情况下,测试 CRC 校验逻辑能够检测出错误的概率。测试采用四种不同的模式:随机位模式(翻转位的位置随机选择);连续相邻位模式(翻转位的位置相邻);连续奇数位模式(翻转位的位置位于连续的奇数位置);连续偶数位模式(翻转位的位置位于连续的偶数位置)。

实验中使用“随机错误数据干扰模块”对整个报文微片进行错误注入。在各模式测试中,当发生翻转的位的个数小于等于 4 时,对所有符合模式定义的翻转位的位置进行遍历,对所有情况进行测试并统计结果,获得 CRC 校验的正确率。当发生翻转的位的个数多于 4 时,若对所有可能出错的位置进行排列组合,组合数目非常大,无法进行全遍历测试。因此,实验中对注入 5 到 8 位随机错的情况进行了简化:分别固定微片数据的第 1 位到第 4 位为翻转位,剩余的翻转位的位置按照全排列的方式在微片数据剩余的位置中选择。接收方设置一个专门的计数器统计检测出错误的总次数,用于计算校验准确率。

测试结果如下:对于所有的 1 位错、2 位错和 3 位错,各种模式下错误均能够被准确地校验出来,校验准确率为 100%。对于随机注入 4 到 8 位错的情形,CRC 算法具有一定的漏检情况,即不能在所有情况下正确地判断发生了错误。统计发现,在随机注入 4 到 8 位错时,对于随机位模式,99.9% 以上的注入错误能够被准确检出,即校验准确率超过 99.9%;对于连续相邻位、连续奇数位以及连续偶数位三种模式,校验准确率仍为 100%。

3.2.2　多 lane 之间的延迟偏斜容忍能力测试

延迟偏斜的容忍能力和链路的发送时钟与接收方接收时钟的频率比有关,设置发送与接收时钟频率比为 0.9∶1,1∶1,1∶1.5,1∶2 和 1∶4 五种情形。由于 lane 与 lane 之间的延迟偏斜纠正逻辑基于异步 FIFO 实现,FIFO 深度为 16,异步时钟域转换开销为 4 个时钟周期,因此,在发送和接收时钟频率相同的情况下,理论上能够容忍的延迟偏斜大小应该不超过 FIFO 深度减去时钟域转换时间,即 12 个发送方时钟周期时间。当接收时钟频率更高时,由于 FIFO 的读取比发送更快,此时能够容忍的延迟偏斜值会适量增大。

实验发现,当发送和接收时钟频率比为 0.9∶1时,不论延迟偏斜值的大小如何设置,由于 FIFO 的写入速度大于读出速度,因此 FIFO 总会在一段时间后溢出,无法正常进行偏斜纠正工作,而且延迟偏斜越大,FIFO 越早发生溢出。当频率比为 1∶1 时,只要延迟偏斜值小于 12,就不会发生 FIFO 溢出的情况,延迟偏斜纠正逻辑能够正常工作。由于实际电路中时钟频率会发生少量抖动,因此频率比为 1∶1 时,延迟偏斜为理论上能够正常工作的值 12 时 FIFO 仍有可能发生溢出导致纠正失败。在其他几种频率比情况下,FIFO 的写入速度小于读出速度,只有当延迟偏斜值超过一定范围时,FIFO 才会发生溢出,导致无法正确纠正。综上,为了确保延迟偏斜纠正逻辑正常工作,必须满足以下两个条件:①发送时钟频率要小于等于接收时钟频率;②链路数据通路中 lane 与 lane 之间最大的延迟偏斜小于 12 个发送方的时钟周期时间。

4　结论

本文提出了一种面向众核处理器访存链路接口的 FPGA 验证平台,用于对处理器访存链路关键部件进行功能及可靠性测试,提出了片上读写激励自动产生与检查机制、以太网接口硬件 UDP 协议栈、FPGA 芯片间多 lane 并行链路。实验结果表明:①片上读写激励自动产生与检查机制功能正确,通过调整请求发送概率,能够控制访存地址的数量与访问频度,实现对访存地址相关性和每个访存地址的访存次数比率分布的控制;②硬件 UDP 协议栈功能正确,能够在定制的存储调试软件的配合下进行随机地址访存测试;③支持多 lane 并行链路可靠性测试,支持统计和分析各种链路错误注入情况下的校验准确率以及多 lane

链路之间的延迟偏斜容忍能力。

参考文献(References)

[1] Cadence Incorporation. Palladium XP verification computing series[EB/OL]. (2016 - 05 - 14)[2016 - 12 - 15]. https://www.cadence.com/content/cadence-www/global/en_US/home/tools/system-design-and-verification/acceleration-and-emulation/palladium-xp.html.

[2] 周宏伟, 邓让钰, 李永进, 等. 多核处理器验证中存储数据错误快速定位方法[J]. 国防科技大学学报, 2012, 34(6): 1 - 6.
ZHOU Hongwei, DENG Rangyu, LI Yongjin, et al. A fast location mechanism on memory data error for multi-core processors verification[J]. Journal of National University of Defense Technology, 2012, 34(6): 1 - 6. (in Chinese)

[3] S2C Inc. Prodigy complete prototyping platform[EB/OL]. (2016 - 12 - 10)[2016 - 12 - 20]. http://www.s2cinc.com/products/prodigy-complete-prototyping-platform.

[4] 郭阳, 李暾, 李思昆. 微处理器功能验证方法研究[J]. 计算机工程与应用, 2003, 39(5): 35 - 37.
GUO Yang, LI Tun, LI Sikun. Functional verification methodology for microprocessor[J]. Computer Engineering and Applications, 2003, 39(5): 35 - 37. (in Chinese)

[5] 张珩. 通用处理器设计中硬件仿真验证[J]. 计算机工程与应用, 2007, 43(8): 1 - 3, 233.
ZHANG Heng. Hardware simulation for microprocessor verification[J]. Computer Engineering and Applications, 2007, 43(8): 1 - 3, 233. (in Chinese)

[6] 黄晓林, 潘红兵, 易伟, 等. 基于多FPGA的NoC多核处理器验证平台设计[J]. 计算机工程与设计, 2012, 33(1): 180 - 185.
HUANG Xiaolin, PAN Hongbing, YI Wei, et al. Design of multi-FPGA-based multi-core processor NoC verification platform[J]. Computer Engineering and Design, 2012, 33(1): 180 - 185. (in Chinese)

[7] 郭阳, 李思昆, 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.
GUO Yang, LI Sikun, QU Wanxia. Verification of on-chip multi-core processor: challenges, status, forecasts[J]. Journal of Computer-Aided Design & Computer Graphics, 2012, 24(12): 1521 - 1532. (in Chinese)

[8] Starke W J, Stuecheli J, Daly D M, et al. The cache and memory subsystems of the IBM POWER8 processor[J]. IBM Journal of Research & Development, 2015, 59(1): 1 - 13.

[9] Pawlowski J T. Hybrid memory cube (HMC)[C]// Proceedings of the 23th Hot Chips Conference, 2011: 1 - 24.