

## 基于前递预取的 SoC 内存控制器精准仿真方法<sup>①</sup>

李作骏<sup>②</sup> 卢天越 陈明宇<sup>③</sup>

(中国科学院计算技术研究所先进计算机系统研究中心 北京 100190)

(中国科学院大学计算机科学与技术学院 北京 100190)

**摘要** 本文提出一种基于现场可编程门阵列(field programmable gate array, FPGA)的内存控制器性能精确仿真评估方法,通过高速可扩展接口(advanced extensible interface, AXI)总线前递、访存预取和数据缓存的方式解决了 FPGA 芯片内外访存时序需求不一致的问题,从而实现了在真实处理器系统应用仿真场景下对内存控制器的精确性能评估。与香山开源第 5 代精简指令集计算机(reduced instruction set computer-five, RISC-V)处理器雁栖湖架构硅后芯片对比, SPEC CPU2006 基准测试程序的执行时间平均偏差为 1.29%,最大偏差为 3.45%。该方法解决了因为内存控制器模型不准确而导致 FPGA 片上系统(system of chip, SoC)原型系统中真实应用仿真性能评估与流片后实际性能存在较大偏差的问题,同时无需进行大量修改就能用于任何支持 AXI 和双倍数据速率物理层接口(DDR PHY interface, DFI)协议的标准内存控制器精确仿真。

**关键词** 内存控制器;现场可编程门阵列;性能评估;双倍数据速率;动态随机访问存储器

近年来,随着通用处理器核心数、频率、架构的逐步提升,大数据、人工智能时代的到来,各类高并发度、大数据量应用不断涌现,由此而来的对大规模数据处理和存储的需求对现代处理器的访存性能提出了更高的要求,访存系统已成为影响整个处理器性能提升的关键部分之一<sup>[1]</sup>。而现代处理器片上系统(system on chip, SoC)研制过程中,由于流片周期和价格等因素的影响,流片前的仿真性能评估对于整个芯片的设计和验证非常重要<sup>[2]</sup>,这一流程涉及到的内存系统特别是芯片外动态随机访问存储器(dynamic random access memory, DRAM)的性能评估更是其中的关键。因此,在实际流片之前,能有一种能够快速、准确地对内存控制器性能进行评估的方案是不可或缺的。

在现有的内存控制器性能评估过程中,通常使

用软件仿真、大型仿真加速器和现场可编程门阵列(field programmable gate array, FPGA)原型系统 3 种方案对整体性能进行评估,而现有真实系统中的大数据集应用的出现更是对现代处理器设计中的实际应用性能评估提出了更高的要求。使用 VCS<sup>[3]</sup>或 Verilator<sup>[4-5]</sup>等软件仿真工具虽然能够同时对处理器和内存控制器性能进行精确建模和评估,但由于其执行速度较慢,通常只适合于小规模程序片段的性能评估;大型仿真加速器虽然能够在一定程度上满足真实应用的访存性能评估需求,但其价格通常极为高昂;FPGA 原型系统的方案由于其相对高速且成本较低,更适合于多数场景下基于真实应用对内存控制器性能进行仿真评估。

但是在 FPGA 芯片的内部缓存容量难以满足真实应用中产生的大量数据集的存储问题,只能借助

① 国家重点研发计划(2022YFB4500403)资助项目。

② 男,1990 年生,博士生;研究方向:计算机系统结构,内存与存储系统;E-mail:lizuojun@ict.ac.cn。

③ 通信作者,E-mail:cmy@ict.ac.cn。

(收稿日期:2024-03-21)

片外 DRAM 进行辅助存储。这就引入了一个新的挑战:在 FPGA 原型系统中部署的芯片代码包括内存控制器通常是以一个等比例缩放频率来执行的,这就使得做精确仿真时在 FPGA 中的内存控制器与外部存储芯片 DRAM 的交互也必须按照这个等比例缩放的频率来执行。然而,对于现有的标准双倍数据速率(double data rate, DDR)内存控制系统而言<sup>[6]</sup>,由于时序、工艺等的限制,其只能保证在一个较高的频率下运行,这就导致了 FPGA 内部低频仿真时序与外部高频访存的一个时序不匹配的问题。对于这个问题,国际上目前并没有一个成熟的方案。通常采用的增加固定访存延迟的方法并不能完整地反映出 DDR 内存系统中的复杂时序要求;使用现有 DDR 协议中的 DLL-off 低频工作模式<sup>[6]</sup>只能局限在特定的频率范围且其时序与真实系统访存存在一定差异。FireSim<sup>[7]</sup>方案模拟了内存控制器的时序和功能,并通过解耦合的方式对读写访存时序进行调度,但该方案需要对整个处理器的设计作一定的修改,很难对第三方标准的 DDR 访存子系统进行适配。

本文提出了一种在 FPGA 原型系统中精确仿真评估内存控制器性能的方案,通过采用高速可扩展接口(advanced extensible interface, AXI)<sup>[8]</sup>总线前递、访存预取以及数据缓存的方式,实现了对于内存控制器性能的精确评估,同时通过地址映射和协议转换保证了待评测系统的内存控制器无关性,从而使其能够用于评估任何支持标准双倍数据速率物理层接口(DDR PHY interface, DFI)<sup>[9]</sup>的内存控制器。

本文在 AMD FPGA 平台上设计并实现了一个内存控制器精确仿真方法,适配香山第 5 代精简指令集计算机(reduced instruction set computer-five, RISC-V)开源高性能雁栖湖版本的处理器<sup>[10-11]</sup>,并实现了全系统精确仿真系统。基于 SPEC CPU2006 基准测试集,在相同软件环境下对 FPGA 精确仿真系统和真实芯片进行了对比测试。结果表明,该系统在各类基准测试中准确率均高于传统 FPGA-固定延迟方案,且与真实芯片系统的性能测试结果平均误差在 1.29% 以内,最大误差为 3.45%。

## 1 背景和相关工作

### 1.1 处理器与内存控制器

内存控制器是现代计算机系统中不可缺少的一部分,是处理器和内存介质之间沟通的桥梁,负责调度数据在内存介质中的存取,保证数据能够正确、完整地储存在内存介质中。

在现代处理器设计中,由于内存墙问题的存在<sup>[12]</sup>,内存控制器的设计会对 SoC 整体性能产生非常大的影响,无法忽略不计。处理器发来的读写请求的调度、访存地址与内存地址的映射方式及内存中数据的缓存预取等设计选择均会对内存控制器的性能进而对 SoC 整体的性能产生相当的影响。如何精确地量化评估这些设计对整体性能的影响已成为研制人员需要考虑的关键问题之一。

#### 1.1.1 现有内存控制器性能评估方案

在对处理器 SoC 性能进行评估的过程中,研究人员通常使用 SPEC CPU 基准测试集合<sup>[13]</sup>作为一种必要的测试手段。而内存控制器作为处理器 SoC 系统的一个关键部分,不仅需要满足自身在芯片层面上的各类时序测试要求,更应当能够应对基于实际应用的大数据集系统级测试。虽然不同内存控制器的固定延迟、带宽等时序表现受标准协议影响通常差距不大,但由于其内部架构的不同,其对 SoC 系统级性能测试的表现差距会达到 10.00% 以上。因此,在对内存控制器进行性能评估的过程中,通常会在相同的处理器条件下,对各类不同的内存控制器进行基于实际应用的系统级测试。这一类测试中,最常用也是最能体现内存控制器性能指标的是 SPEC CPU 基准测试。

为了保证这一类基准测试的全系统仿真能够正常运行,在带有内存控制器的系统仿真时,通常有软件仿真、大型仿真加速器和 FPGA 原型系统 3 类方法。

#### 1.1.2 软件仿真

软件仿真方式就是使用 VCS、Verilator 等软件来编译执行寄存器传输级(register-transfer level, RTL)或者网表级的代码,通过仿真全局电路信号

的输入、输出和中间状态,来调试、验证和评估整个内存控制器的设计方案。其优点在于操作简单、易于上手,但缺点是执行时间相对较慢、运行频率通常在 1 ~ 100 kHz 之间。

RTL 全系统仿真能够将完整的内存控制器仿真模型和标准 DDR 双列直插式内存模块(dual inline memory modules, DIMM)模型集成到一起来进行稳定和准确的系统级仿真。对于 VCS 全系统仿真而言,由于仿真过程需要保证全局信号多种状态的稳定性,对于一个 SoC 系统 SPEC CPU 基准测试等实际应用的测试,其执行时间通常需要数日到数周甚至以月计。

因此,在实际应用中,也有采用辅助于 gem5<sup>[14-15]</sup>、DramSim<sup>[16]</sup>、Ramulator<sup>[17]</sup>、ZSim<sup>[18]</sup>等模拟器的方式来对软件仿真进行简化。其中,gem5 是全系统模拟器,DramSim、Ramulator、ZSim 是访存模拟器,它们能够准确模拟整个片外访存系统,包括内存控制器和片外 DDR DIMM。虽然 DramSim3 这类访存模拟器是周期准确的访存模拟器,但它们使用的内存控制器实现方式与实际芯片设计仍然会有一些区别,并不能完全准确地评估内存控制器的性能。

香山 RISC-V 开源处理器雁栖湖架构在前期评测中采用了 Verilator + DramSim3 的评测方案,该方案中运行单个完整的 SPEC CPU2006 基准测试就需要一周甚至更长的时间。

### 1.1.3 大型仿真加速器

大型仿真加速器的测试方法是通过定制化的专用软硬件设计,使用软硬件结合的方式加速整个仿真过程。这种方式可有效地缩短整个芯片的验证时间,同时能够评估规模较大的应用,运行频率通常在 100 kHz ~ 10 MHz 之间。

Cadence 的 Palladium 仿真加速器<sup>[19-20]</sup>能够将现有的 10 亿门电路级别的设计通过快速编译以及软硬件结合的设计,将单个完整的全系统仿真任务拆分为多个不相关的仿真任务,然后分配到各个硬件仿真节点上去,以实现仿真性能和资源利用的最大化。

但这类设备的价格通常极为高昂,无法在普通的 SoC 设计过程中采用。

### 1.1.4 FPGA 原型系统

FPGA 原型系统是通过将整个设计移植到符合测试环境的 FPGA 开发板中进行电路级验证。在电路时序能够保证且运行频率相同的情况下,其各个信号的输入输出等状态几乎可以与实际芯片的状态相一致,基本可以作为完整验证芯片预流片之前的实际功能和性能验证手段。如 Synopsys 的 HAPS 系列<sup>[21-22]</sup>。这种方式对于绝大多数设计都有比较快速、准确的仿真测试效果,且因为 FPGA 原型系统的规模可控,其成本在可以接受的范围之内,因此在普通 SoC 设计中被广泛采用。

但是 FPGA 芯片的片内存储资源有限,这就对真实应用的内存控制器的精确性能评估提出了挑战。对于真实应用的测试而言,由于 FPGA 原型系统上的资源限制,处理器核部分的时钟几乎只能保持在 100 MHz 以下,因此研究人员通常采用全系统运行频率按比率缩放的方式来保证精确性。由于 FPGA 芯片内部较小的存储容量不能满足真实应用的性能评估需求,对于 DDR 内存控制器的性能评估必须借助芯片外部大容量的 DRAM 来实现,而 FPGA 芯片对于片外 DRAM 的正常访问是受到 DDR4 内存总线的时序约束的,无法通过缩放频率来实现。如果直接将 FPGA 芯片内部的处理器核通过内存控制器直接对片外的 DRAM 进行访问,则必然会出现外部访存速率远大于处理器执行速率的“倒挂”状况,这必然会导致实际应用测试结果的不准确,甚至是出错的情况,如图 1 所示。

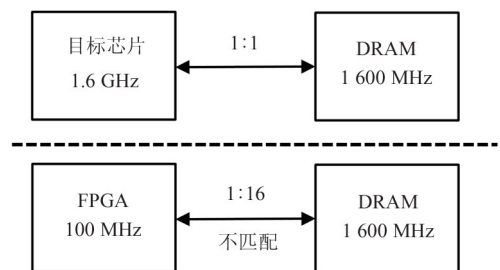


图 1 FPGA 原型系统评测内存控制器性能时的频率不匹配现象

这 3 种可用于实际应用性能评估的仿真测试环境的运行速度、使用难度和成本对比如表 1 所示。

从表 1 可以得知,软件仿真的优势在于其使用

起来较为方便,且价格偏低,但仿真速度较为缓慢;大型仿真加速器的优势在于在保证较高仿真速度的前提下,使用难度并没有显著提高,但其价格也是较为昂贵;而 FPGA 原型系统的方案虽然具备较高的运行速度,且价格在可以接受的范围内,但也仍然需要面对片内外访存时序不匹配的问题。

表 1 软件仿真、仿真加速器和 FPGA 原型系统的对比表

| 评估方法      | 运行频率             | 使用难度 | 成本 |
|-----------|------------------|------|----|
| 软件仿真      | 1 ~ 100 kHz      | 低    | 低  |
| 仿真加速器     | 100 kHz ~ 10 MHz | 一般   | 高  |
| FPGA 原型系统 | 10 ~ 100 MHz     | 一般   | 一般 |

## 1.2 FPGA 内存精准仿真的技术挑战

在一般利用 FPGA 进行内存控制器仿真实现过程中,研究人员通常采用简单的增加固定访存时钟周期延迟的方式来匹配芯片外内存速率与 FPGA 运行频率之间的差异。但是受到 DDR 时序中的刷新、行激活以及预充电等特性的影响,在此场景中从片外 DRAM 返回数据与实际应用状况下返回数据的时间差异并不是完整的固定延迟关系。采用增加固定延迟方式只能保证平均数据返回延迟与真实应用条件下平均延迟的误差能够控制在一个较小的范围内,这对于评价真实应用场景的访存行为会存在一定的误差。

要实现精准的内存控制器仿真和应用性能评估,研究人员将面临以下 2 个挑战。

### (1) 及时性问题

根据 DDR4 标准协议,从内存控制器向 DDR4 总线发出的读请求必须在十余个周期内返回数据。将内存控制器置于 FPGA 原型系统中进行性能评估时,由于 DDR4 标准协议规定的从被测内存控制器下游发出的读请求必须在十余个周期内返回数据;而下游的控制芯片外 DRAM 访问的物理内存控制器受到刷新、预充电、行激活等非读写命令的影响,尽管其运行在较高的频率下,也不能完全保证在上述非读写命令的干扰下及时将数据读取回被测内存控制器端。因此,应当如何处理芯片外访存延迟与被测内存控制器所需的访存延迟之间的差异是需要解决的关键技术之一。

### (2) 精准时序同步问题

在验证过程中使用的解决方案必须能够完整反映真实 SoC 访存的行为,即需要确保 FPGA 原型系统中对片外内存的访问不会影响处理器核原有的预期访存延迟、带宽等行为特性,需要保证 FPGA 原型系统中被测系统的各个访存请求都与预期系统的访存请求在时钟周期上完全一致。一方面,访问片外内存可能存在早到达、乱序到达情况,需要进行调整适配。另一方面,需要确保在评估过程中对芯片外内存的周期性非读写命令不会影响被测内存控制器的正常读写请求延迟,同时要保证这些额外的维护性命令周期也能按计划完成。

## 1.3 现有 FPGA 原型系统中的评估方法

面对上述挑战,已存在一些解决方案,包括 DLL-off 模式和功能-时序分离方法等,但这些方案仍有不足之处。

DDR4 标准协议<sup>[6]</sup>提供了一个供 DRAM 测试用的 DLL-off 模式。在 DLL-off 模式下,DRAM 的运行频率范围可以被缩减至 50 ~ 150 MHz,基本上可以和被测系统同步运行。但 DLL-off 模式下各个时序参数只能保证在特定的数值下运行,不能灵活调整,这往往与真实应用场景存在偏差。此外,由于 DRAM 的物理特性决定其刷新频率是固定的,不能因为降频就减少刷新的频度。如果不对被测内存控制器进行刷新参数上的修改,就会导致数据读写出错,使得其与真实应用场景下的运行情况出现一定的差距。因此这种方案只可作为片外访存和 FPGA 原型系统的速率不匹配问题受限的一个解决方案。现有的 FPGA 上的内存控制器如 AMD 的内存接口生成器(memory interface generator, MIG)不支持 DLL-off 模式,如果要完整支持 DLL-off 模式,对于整个内存控制器的修改和测试的工作量较大,因此很少有实际的应用。

FireSim<sup>[7]</sup>提出了一种 FPGA 原型系统上模拟内存控制器功能并对整个系统进行性能评估的框架。它利用抽象 RTL 通过模块解耦的方式构建了一个包含内存控制器时序模型和功能模型的完整访存模型。通过功能模型确定读写请求队列的发送,将处理器传来的访存内容经功能模型调度处理后发往

片外 DRAM;之后再借助时序模型确定数据返回的时间,当片外 DRAM 的数据返回时按照真实情况下确定的时序返回给处理器,以此来模拟一个标准的内存控制器以及片外 DRAM 的访存通路。FireSim 解决上述的及时性问题,依靠的是暂停系统仿真时钟等待内存数据返回的方法。

首先,这种方法必须采用专门开发的内存控制器时序模型,限制了对第三方的标准内存控制器的验证能力。其次,在真实系统中的高性能内存控制器的性能评估过程中,随着处理器带来的访存压力的上升以及后端片外访存的刷新等问题,势必会频繁导致其出现内部缓存队列占满以至于需要停止整个系统时钟的问题,这无疑增加了整个系统的运行时间。此外,FireSim 采用的这种功能-时序分离方法,在实际测试之前需要对原有内存控制器及处理器核心代码都进行修改及验证。

本文针对上述 2 种方案的缺点提出了一种基于预取的新方案。该方案能够解决被测系统频率与芯片外 DRAM 访存频率不匹配的问题,不需要修改原有访存通路、对被测系统的修改量较小,且适用于现有的各类支持标准接口的第三方内存控制器的精准仿真评估。

## 2 设计与实现

在 FPGA 原型系统中评估内存控制器性能时通常需要将数据缓存在片外 DRAM 中,如图 2 所示。因此,解决上述 2 个挑战的关键在于:(1)需要解决芯片外部访存延迟过长导致被测内存控制器正常读写请求不能及时返回的问题;(2)需要保证被测系统的各个访存请求都与原有系统的访存请求在时钟周期上完全一致。

### 2.1 系统总体设计

图 2 为整个 FPGA 原型系统的结构图。被测内存控制器为已经完成设计等待测试和评估的内存控制器。处理器的读写请求通过其上游的片上总线 AXI 接口发送,经过内部一系列的调度、访存时序关系处理等操作后,通过下游的标准 DFI 接口向 DDR 物理层模块发送和接受读写请求信息。通常 FPGA

上被测系统运行的时钟频率低于 100 MHz。这种低频的 DFI 接口无法直接和 FPGA 的 DDR 物理层模块相连,而是首先转化为对 FPGA 原型系统上的物理内存控制器请求。FPGA 的物理内存控制器与其 DDR 物理层相配合,可以运行在标准高速 DDR4 速率下。因其内部的读写调度方式较为简单,可在较短时间内返回读数据信息,其运行频率通常是被测系统使用的时钟频率 10 倍以上。

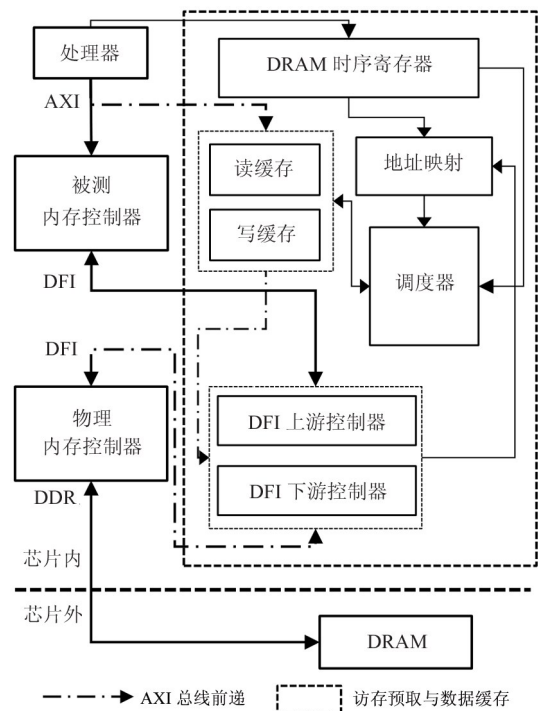


图 2 整个 FPGA 原型系统的结构图

本文提出的方案通过一个访存性能精确评估模块实现,其位于 FPGA 物理内存控制器和被测内存控制器之间,作为内部低频访存与外部高频访存之间的桥梁,通过 AXI 总线前递的方式解决了芯片内外访存延迟不匹配的问题、采用访存取和数据缓存及调度保证了被测系统各项功能和时序的正确性,其完整的内部结构如图 2 所示。

后文中仿真时钟指 FPGA 被测内存控制器运行的低频时钟,物理时钟指片外 DDR4 命令地址总线使用的高频时钟。

### 2.2 AXI 总线前递

本文面对的第 1 个挑战,即 FPGA 芯片内外访存频率不匹配的问题,体现在整个访存通路上从被

测内存控制器发出的访存请求必须保证在 DDR 协议规定的较短的十几个仿真周期内返回给被测内存控制器。这就使得无论物理内存控制器面对任意刷新、预充电、行激活、校准等特殊情况,都必须能及时将数据返回给被测内存控制器。考虑到内存控制器内部存在的地址转换、缓存调度以及时序控制等功能需要占用一定的时间周期进行处理,如果需要保证数据正确性,就必须给被测内存控制器读数据的返回预留出尽可能多的时间。因此,本文采用 AXI 总线前递的方式预先将访存请求不经过被测内存控制器直接转发给物理内存控制器提前进行处理,避免芯片外访存时可能遇到的各种非读写命令产生的特殊时延对读数据实时性的影响,从而影响被测内存控制器性能评估的准确性。

以香山开源 RISC-V 处理器雁栖湖架构使用的内存控制器为例,被测内存控制器使用低频命令地址总线时钟,物理内存控制器使用高频命令地址总线时钟。如图 3 所示,从被测内存控制器的访存读请求自 DFI 接口发出到读数据返回需要 13 个仿真时钟周期。而根据 DDR4 标准协议,从下游物理内存控制器接收到请求到读数据返回再考虑预充电、刷新、阻抗校准、行激活等可能出现的额外非读写命令,在单次访存过程中最大延迟的读请求需要完整经历“预充电-刷新-阻抗校准-行激活-读数据”5 个步骤的组合。以镁光型号 MT40A1G8SA-062E 的 8 Gb DDR4-1600 速率 DRAM 芯片为例,根据 DDR4 标准协议中规定的各项时序参数,预充电步骤是将已经开行的数据写回到 DRAM 中,需要 10 个物理时钟周期 ( $t_{RP}$ );刷新步骤是维护 DRAM 数据正确

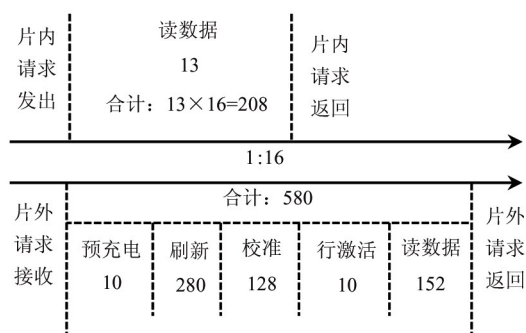


图 3 FPGA 原型系统中内存控制器芯片内外的读请求处理时间流

性的周期性命令,其执行时间通常为 350 ns,即 280 个物理时钟周期 ( $t_{RFC}$ );阻抗校准步骤是为了校准运行过程中温度等因素对终端电阻的影响,需要 128 个物理时钟周期 ( $t_{ZQCS}$ );行激活步骤则是 DRAM 读写前的打开行的操作,即将数据预先读取到行缓冲区,等待后续带有列地址的读写命令访问,需要 10 个时钟周期 ( $t_{RCD}$ );最后的读数据步骤则是正常从物理内存控制器发出请求到数据返回的时间,这一部分是在该请求地址已处于开行状态时,经实际测量需要 152 个物理时钟周期。因此,整个流程共计需要 580 个物理时钟周期。即便仿真时钟和物理时钟比值达到 1:16,也只有等效于 192 个物理时钟周期用来等待芯片外数据的返回,显然无法满足最差情况下可能需要的 580 个物理时钟周期的等待时间,使得最终的测试结果产生一定的偏差。

根据实际测试,单个读请求从被测内存控制器的 AXI 接口接收到下游的 DFI 接口发出的最短时间为 58 个仿真时钟周期,而采用 AXI 总线前递之后,如图 4 所示,这些请求可预先发送给芯片外的高速内存控制器,这就使得原本仅有的 13 个仿真时钟周期的等待时间被扩展到了 71 个仿真时钟周期。这就使得原本 192 个等效物理时钟周期的余量被扩展到了 1 136 个,即便考虑跨时钟域、在等待数据返回期间的请求排队及其他数据处理等额外等待时间,这也完全能够满足最差情况下需要的 580 个物理时钟周期的芯片外数据返回时间。

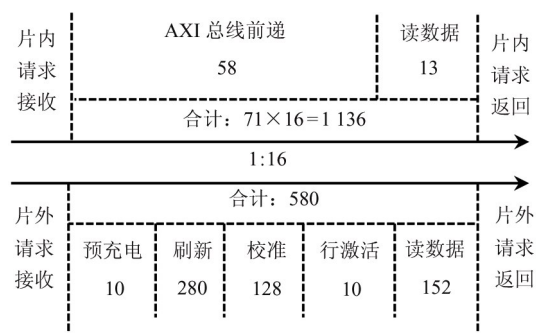


图 4 使用 AXI 总线前递后的 FPGA 原型系统中内存控制器芯片内外的读请求处理时间流

因为被测内存控制器中存在因调度而产生乱序的可能,还需要考虑的一种情况是 AXI 前递的访存并不是从 DFI 发出的第一个访存操作。但由于刷新

和校准这 2 个非读写命令为周期性命令,其 2 次执行间隔通常在数千个物理时钟周期以上,因此该最大延迟的读请求访存过程并不会连续出现。在通常状况下,单次对芯片外 DRAM 的读请求数据返回时间在 152 ~ 172 个物理时钟周期之间,具体数值取决于该请求是否需要行地址的切换从而需要额外的预充电和行激活命令。因此经过扩展后的 1 136 个周期余量,也足够满足一次 580 周期最长延迟和一次普通访问延迟的处理。

### 2.3 访存取和数据缓存

本文面对的第 2 个挑战,即需要保证 FPGA 原型系统中被测系统的各个访存请求都与原有系统的访存请求在时钟周期上完全一致,这就使得从被测内存控制器发往下游的全部访存请求都必须在 DDR4 时序规定时钟周期时刻返回。由于 AXI 前递以及物理内存控制器的高频率低延迟特性,从物理内存控制器返回的数据往往会在早于被测内存控制器需求的时间点到达。因此,从物理内存控制器返回的数据根据其访问地址会缓存起来,在被测内存控制器正常运行时需要的特定时间点返回数据,以保证整个被测系统的性能表现与其预期的实际表现相一致。

如图 4 所示,采用 AXI 总线前递之后,访存请求可以在被测内存控制器尚未发出的情况下预先访问芯片外的 DRAM。对于读请求而言,即便在最差的情况下,其返回的数据最晚会在被测内存控制器从处理器接收到访存请求后的 580 个等效物理时钟周期内到达被测内存控制器,而此时被测内存控制器接收到的读请求命令还未经过 DFI 接口发出。且由于现代高性能内存控制器均存在访存命令调度上的优化,其发出的访存请求与接收到的访存请求存在乱序的现象,这就使得在正式的请求来临之前预先到达的读数据必须与获取它的访存请求相对应,使其地址和数据以配对的形式缓存起来。此外,在被测内存控制器正式发出访存请求时,要想保证 DDR 时序的准确性,必须在剩余的 13 个仿真时钟周期内将数据准时返回到被测内存控制器。

因此,本文采用访存取结合数据缓存的方式将预先返回的数据暂存在数据缓存中,并与请求发

出时的地址缓存相对应。而当读请求从被测内存控制器的 DFI 接口发出时,经由 DFI 命令识别、行列地址转换为物理地址、缓存物理地址查找、缓存读数据返回 4 个步骤,能够保证第 13 个仿真时钟周期将数据准时返回到被测内存控制器中。

### 2.4 FPGA 原型系统的实现

在实际设计中,本文在被测内存控制器和物理内存控制器之间构建了一条额外的 AXI 总线旁路,通过二者之间的读缓存和写缓存模块实现跨时钟域和请求缓存的效果,使其能够满足芯片内外访存频率不匹配的问题,如图 2 所示。其中,FPGA 内部各个模块的功能如下所述。

调度器模块可以通过对读写请求进行调度,确保写请求和写数据能够以最快的方式从写缓存模块发出,而读请求能够在正确的时序关系下从读缓存模块中返回数据。该模块会根据 DFI 接口发来的刷新、预充电、行激活等命令通过内部状态机和计数器模拟正常的 DRAM 中所处的状态,并监测物理内存控制器的队列及数据返回情况,以此来保证读请求所需的数据严格按照 DDR4 时序中需求的时钟周期数返回,从而实现与真实状况下的访存特征保持完全一致。

地址映射模块是保证其与被测内存控制器使用同一种地址译码方式,使得预先发来的 AXI 访存请求物理地址信息能够与 DFI 访存请求行列信息相匹配,从而保证读写请求地址的正确性。由于现代的高性能内存控制器通常经过内部调度后会出现乱序的问题,即被测内存控制器从经过 AXI 前递接收的请求与从 DFI 接口发出的请求地址会出现不一致的情况。这种乱序带来的读数据返回与读请求发出在顺序上不匹配的现象使得调度器在处理返回的读数据时,必须将返回的数据与请求的地址相对应。此外,DFI 接口涉及到的地址是经转换后内存的行列地址,AXI 涉及到的地址则是系统物理地址,在实际应用中,也需要通过地址映射的方式将其一一对应。根据 DDR4 协议,DRAM 会通过行激活命令打开其内部特定的行地址,再通过读写命令传输其需要访问的列地址。因此,本文根据“行激活命令-缓存行,读写命令-缓存列”的方式将获取到的行列地址进行

缓存,再根据获取到的被测内存控制器中设定的行列地址与物理地址的映射关系将其转换为与上游发来的访存请求相一致的物理地址。

DFI 上游控制和 DFI 下游控制模块为与被测内存控制器进行通讯的模块,主要负责将被测内存控制器发来的 DFI 请求转化为模块内部能够识别的读写请求,并将模块内部返回的读数据信息即时发送给被测内存控制器,以保证在无修改的情况下其能够适配各类常用的内存控制器。

DRAM 时序寄存器包含 2 个部分:一部分是用于保证其与被测内存控制器中对于 DRAM 相关访存参数的同步;另一部分则是用于对调度器模块中的调度方式进行控制。

### 3 实验验证

软件仿真方式的耗时为 FPGA 原型系统的 1 000 倍以上,通常并不适用于 SoC 预流片阶段需要进行的大量真实应用场景下的测试,而仿真加速器方式的成本对于一般小规模项目而言通常难以接受。因此,本文在 FPGA 原型系统中与通常采用的简单固定延迟方案以及实际流片后的真实芯片在同一操作系统中使用标准的处理器性能评测基准测试集 SPEC CPU2006 进行实验,并将执行时间结果归一到相同时钟频率下后进行对比。

#### 3.1 测试方法

为了对内存控制器在真实场景下的性能进行评估和对比,本文选用以下 3 个对照平台。

(1) 流片后的香山开源 RISC-V 处理器雁栖湖版本为基准测试平台,CPU 主频为 1.6 GHz,乱序单核处理器,使用某商用高性能内存控制器,运行频率为 DDR4-1600,容量为 8 GB,运行兼容 RISC-V 的 Linux 操作系统。

(2) 携带 AMD 的 VU19P 芯片的 FPGA 原型系统平台,部署了与(1)相同的处理器及操作系统版本,其时钟频率为 100 MHz,以及芯片外运行频率为 DDR4-1600 的 8 GB 内存条,采用简单的 90-周期固定延迟的方案来弥补芯片内外频率不匹配的问题。

(3) 采用与(2)相同的 FPGA 原型系统平台,加

入了本文提出的精准仿真方案替换原有的 90-周期固定延迟的方案。

其中,香山开源 RISC-V 处理器雁栖湖版本为真实处理器 SoC 芯片。90-周期固定延迟的方案为在芯片外高速内存数据返回给 FPGA 内部处理器的通路中加入 90 周期延迟补偿,以弥补 FPGA 芯片内外因频率不匹配导致的访存延迟和带宽差异问题。这里的延迟周期数受 71 个访存时钟周期的固定延迟以及访存带宽影响,同时经过多次 Dramsim3 模拟器测试确定。

性能评估测试的基准程序选用部分 SPEC CPU2006 基础测试集,以各个应用的完整运行时间归一到相同频率下作为对比测试。

#### 3.2 实验结果与分析

实验结果如图 5 所示,3 个平台对比结果,以香山雁栖湖芯片测试结果为基准,可见本文提出的精确仿真模块对性能评估的准确性明显高于同一平台下 90-周期固定延迟的方案。与实际芯片对比,本文提出的精确仿真模块的测试结果最大误差为 3.45%,平均误差为 1.29%。其中,dealII 为计算密集型程序,访存密度不高,运行时间误差最小,而 90-周期固定延迟的方案采用的访存延迟已接近系统平均访存延迟;Gems FDTD 和 lbm 均为访存密集型程序,其访存密集度较大,90-周期固定延迟的方案由于芯片内外频率不匹配放大了被测系统在面对这些应用时本应受限的访存带宽,导致整体带宽运行时间的误差已达到完全无法忽视的程度,最大可达 20.00%。可见采用简单固定周期延迟的仿真方案无法体现访存带宽限制对系统整体性能的影响。

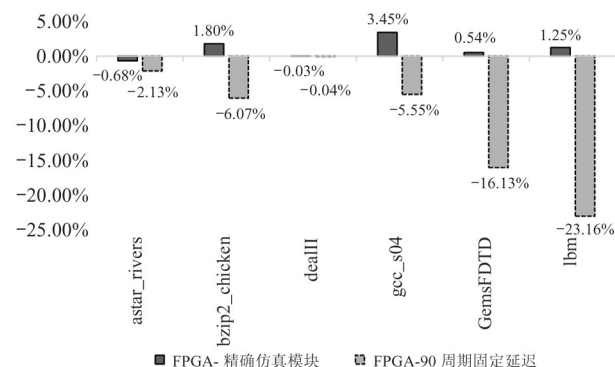


图 5 相同频率下 FPGA-精确仿真模块、FPGA-90 周期固定延迟方案与实际芯片测试的运行时间对比

## 4 结论

由于运行速度、价格和易用性的优势,对于与内存控制器相关的 SoC 性能评估通常在 FPGA 原型系统平台上进行。但芯片内外访存频率不匹配的问题,给精确评估系统整体性能带来了一定的挑战。本文通过 AXI 总线前递、访存预取和数据缓存的方式解决了 FPGA 芯片内外对于访存延迟需求不一致的问题,同时保证了被测系统与预期系统访存时序上的一致性,且不需要对被测内存控制器产生修改,适用于任意标准 AXI 和 DFI 接口的标准内存控制器。本文基于香山开源 RISC-V 处理器平台,与实际芯片及相同平台下的固定 90-周期延迟方案做了 SPEC CPU2006 基准测试集的对比测试,测试结果表明本文提出的内存控制器性能精确评估方法在各类程序上准确性明显优于 90-周期延迟的方法,且与实际芯片的运行结果最大误差为 3.45%,平均误差为 1.29%,能够较为准确地反映真实芯片对于各类应用的运行性能表现。

本文提出的内存控制器性能精确评估方案可直接用于各类支持 AXI 和 DFI 接口的第三方标准内存控制器中,无需对内存控制器本身作出任何修改,可以支持黑盒测试。且本文方法对整个被测系统的改动极小,所占用的 FPGA 资源如查找表和触发器等均不到 VU19P 总量的 1.00%,极大地扩展了对于内存控制器性能评估的易用性。

随着 DDR5 等新一代内存的推出和广泛应用,AXI 和 DFI 协议也会有所更新。在数据读写方式不发生较大改动的情况下,本文提出的内存控制器性能精确评估方案只需对相关接口转换模块和输入参数作出较小的修改就仍能满足这部分需求。

### 参考文献

- [ 1 ] MOITRA A, BHATTACHARJEE A, KUANG R, et al. SpikeSim: an end-to-end compute-in-memory hardware evaluation tool for benchmarking spiking neural networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023,42(11):3815-3828.
- [ 2 ] WOUDEBERG J, GROSSMANN P, VARNA A, et al. Invited: pre-silicon side channel and fault analysis [C]// 2023 60th ACM/IEEE Design Automation Conference. San Francisco, USA: IEEE, 2023:1-4.
- [ 3 ] SYNOPSIS. VCS functional verification solution[EB/OL]. (2024-03-18) [2024-03-18]. <https://www.synopsys.com/verification/simulation/vcs.html>.
- [ 4 ] 余子濠, 刘志刚, 李一苇, 等. 芯片敏捷开发实践: 标签化 RISC-V[J]. 计算机研究与发展, 2019,56(1): 35-48.
- [ 5 ] GEIER J, MUELLER-GRITSCHNEDER D. vRTLmod: an LLVM based open-source tool to enable fault injection in verilator RTL simulations [C] // Proceedings of the 20th ACM International Conference on Computing Frontiers. Bologna, Italy: ACM, 2023:387-388.
- [ 6 ] JOINT ELECTRON DEVICE ENGINEERING COUNCIL. DDR4 SDRAM standard: JESD79-4D [S]. Arlington: Joint Electron Device Engineering Council, 2021.
- [ 7 ] NIKIFOROV D, DONG S, ZHANG C, et al. RoSe: a hardware-software co-simulation infrastructure enabling pre-silicon full-stack robotics SoC evaluation [C] // Proceedings of the 50th Annual International Symposium on Computer Architecture. Orlando, USA: ACM, 2023:1-15.
- [ 8 ] ADVANCED RISC MACHINES. AMBA AXI protocol specification: ARM IHI 0022 [S]. Cambridge: Advanced RISC Machines, 2023.
- [ 9 ] DDR PHY INTERFACE GROUP. DDR PHY interface version 5.1 [S]. Austin: DDR PHY Interface Group, 2021.
- [ 10 ] XU Y, YU Z, TANG D, et al. Towards developing high performance RISC-V processors using agile methodology [C] // 2022 55th IEEE/ACM International Symposium on Microarchitecture. Chicago, USA: IEEE, 2022:1178-1199.
- [ 11 ] 王凯帆, 徐易难, 何伟, 等. 香山开源高性能 RISC-V 处理器敏捷设计实践[J]. 集成电路与嵌入式系统, 2022, 22(12):4-6.
- [ 12 ] GARZÁN E, LANUZZA M, TEMAN A, et al. AM4: MRAM crossbar based CAM/TCAM/ACAM/AP for in-memory computing [J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2023,13(1):408-421.
- [ 13 ] BUCEK J, LANGE K, KISTOWSKI K. SPEC CPU2017: next-generation compute benchmark [C] // Companion of the 2018 ACM/SPEC International Conference on Performance Engineering. Berlin, Germany: ACM, 2018: 41-42.

- [14] ROGERS S, SLYCORD J, BAHARANI M, et al. Gem5-SALAM: a system architecture for LLVM-based accelerator modeling [C] // 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture. Athens, Greece: IEEE, 2020:471-482.
- [15] 赵紫微, 涂航, 刘芹, 等. 针对 gem5 指令集实现及其功能测试的自动代码生成 [J]. 计算机研究与发展, 2023, 60(7):1678-1691.
- [16] LI S, YANG Z, REDDY D, et al. DRAMsim3: a cycle-accurate, thermal-capable DRAM simulator [J]. IEEE Computer Architecture Letters, 2020, 19(2):106-109.
- [17] LUO H, TŪGRUL Y, BOSTANCI F, et al. Ramulator 2.0: a modern, modular, and extensible DRAM simulator [EB/OL]. (2023-08-21) [2024-03-20]. <https://arxiv.org/pdf/2308.11030>.
- [18] HERRUZO J, FERNANDEZ I, GONZÁLEZ-NAVARRO S, et al. Enabling fast and energy-efficient FM-index exact matching using processing-near-memory [J]. The Journal of Supercomputing, 2021, 77:10226-10251.
- [19] CADENCE. Palladium Emulation [EB/OL]. (2024-03-18) [2024-03-18]. [https://www.cadence.com/en\\_US/home/tools/system-design-and-verification/emulation-and-prototyping/palladium.html](https://www.cadence.com/en_US/home/tools/system-design-and-verification/emulation-and-prototyping/palladium.html).
- [20] KAPOOR S, BATTA K, NAGPAL J. Emulation: accelerating simulation for rapid verification of modern processor-based subsystems [C] // 2023 3rd International Conference on Intelligent Technologies. Hubli, India: IEEE, 2023:1-8.
- [21] SYNOPSYS. HAPS-80 prototyping solution [EB/OL]. (2024-03-18) [2024-03-18]. <https://www.synopsys.com/verification/prototyping/haps-80.html>.
- [22] RAO C, YU H, WAN H, et al. ICARUS: a specialized architecture for neural radiance fields rendering [J]. ACM Transactions on Graphics, 2022, 41(6):1-4.

## A precise simulation method for SoC memory controller based on forwarding and prefetching

LI Zuojun, LU Tianyue, CHEN Mingyu

(Center for Advanced Computer Systems, Institute of Computing Technology,  
Chinese Academy of Sciences, Beijing 100190)

(Computer Science and Technology Academy, University of Chinese Academy of Sciences, Beijing 100190)

### Abstract

In this paper, an accurate simulation method of memory controller performance evaluation based on field programmable gate array (FPGA) is proposed, which solves the problem of inconsistency between internal and external memory access timing requirements of FPGA chip through advanced extensible interface (AXI) bus forwarding, memory prefetch and data cache. It can be implemented to evaluate the accurate memory controller performance under precise processor prototype with real applications. Compared with the post-silicon chip of the Xiangshan open-source RISC-V processor Yanqihu architecture, the SPEC CPU2006 benchmark runtime shows an average deviation of 1.29% and a maximum deviation of 3.45%. This method reduces the large deviation of real application performance evaluation results between pre-silicon FPGA system of chip (SoC) prototype and post-silicon test, which is primarily caused by inaccurate memory controller model. Furthermore, it can be used for accurate simulation of any standard memory controller supporting AXI and DFI protocol with minor modification.

**Key words:** memory controller, field programmable gate array, performance evaluation, double data rate, dynamic random access memory