

## 基于 FPGA 模拟片上多核处理器的新方法<sup>①</sup>

陈新科<sup>②\*</sup> 黄 帅<sup>\*\*\*\*</sup> 王焕东<sup>\*\*\*\*</sup> 吴瑞阳<sup>\* \*\* \*\*</sup> 曾 露<sup>\* \*\* \*\*</sup>

( \* 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)

( \*\* 中国科学院计算技术研究所 北京 100190)

( \*\*\* 中国科学院大学 北京 100049)

( \*\*\*\* 龙芯中科技术有限公司 北京 100190)

**摘 要** 为了解决使用现场可编程门阵列(FPGA)进行大规模片上多核处理器模拟的容量限制难题,提出了一种新颖的 FPGA 模拟方法。该方法通过混合真实的处理器核与伪造的处理器核,使用 1 个或 2 个 FPGA 即可模拟整个片上多核处理器,而且可以有效克服 FPGA 的容量限制问题,同时又不过多损害对多核处理器行为特征的有效模拟。用此方法实现了周期精确的全芯片模拟,并使用流片后的片上多核处理器芯片对此模拟方法进行了有效性验证。实验很容易地实现了 50MHz 以上的模拟速度,比基于相同设计的软件仿真快 10 万倍以上。模拟速度的大幅度提升,使得可以启动未经修改的 Linux 操作系统和运行完整的多用户 SPEC CPU2006 train 测试集。这种混合真实处理器核与伪造处理器核的模拟方法为片上多核处理器的功能验证和性能评估提供了一种简单高效的途径。

**关键词** 模拟, 仿真, 模型, 现场可编程门阵列(FPGA), 片上多核处理器, 伪造的处理器核

## 0 引 言

计算机架构设计方法的优劣和设计参数的取舍很难通过建立模型进行理论推导,运用模拟的方法,通过运行实际应用程序对目标性能进行定量分析是非常重要的环节。工业界和学术界一般使用基于软件的模拟器,如 simple-scalar<sup>[1]</sup>, simOS<sup>[2]</sup>, gem5<sup>[3]</sup>等。基于软件的模拟器具有透明、易于使用的优点,并且可以做到周期精确的模拟,但其致命缺点是模拟速度太慢,一般只有 1kHz 到 100kHz 的速度,这样的模拟速度很难运行真实的应用程序。为了弥补模拟速度的缺陷,人们一般使用采样和缩小工作集的方法来评估实际应用<sup>[4]</sup>。但文献[5, 6]的研究表明,测试工作集的大小以及模拟的充分性对模拟结果的准确度影响很大。

面对软件模拟的速度瓶颈<sup>[7]</sup>,越来越多的研究人员把目光转向利用现场可编程门阵列(field pro-

grammable gate array, FPGA)进行模拟加速<sup>[8]</sup>。现代 FPGA 已经可以运行在几十 MHz 到几百 MHz 的时钟频率,这样的运行速度已足够胜任一个完整的操作系统和较大规模的应用程序了。随着集成电路按照摩尔定律发展, FPGA 的容量也越来越大,已经可以放下一个中等规模的处理器核,但要把一个完整的片上多核处理器映射到单个 FPGA 中仍然是不现实的。为了解决 FPGA 的容量限制,人们常用以下方法:(1)时序模型与功能模型分离<sup>[5, 9]</sup>,其缺点是功能模型与时序模型之间的通信延迟成为限制模拟速度提升的一个瓶颈因素。(2)时分复用。它采用类似于超线程处理器核的技术,只用 FPGA 实现一个物理处理器核,通过分时使用的方法虚拟出多个处理器核供软件使用。文献[6, 10, 11]都采用了这种方法。这种方法的一个缺点是降低了模拟速度,因为需要使用多个 FPGA 时钟周期来模拟一个目标处理器时钟周期,随着模拟核数的增多,模拟速度会线性下降,同时,时分复用技术需要处理器核架构支

① 国家“核高基”科技重大专项课题(2009ZX01028-002-003, 2009ZX01029-001-003, 2010ZX01036-001-002, 2012ZX01029-001-002-002), 国家自然科学基金(61221062, 61100163, 61133004, 61173001, 61232009, 61222204), 863 计划(2012AA010901, 2012AA011002, 2012AA012202, 2013AA014301)资助项目。

② 男, 1987 年生, 博士生; 研究方向: 计算机体系结构, 片上多核处理器; 联系人, E-mail: chenxinke@ict.ac.cn  
(收稿日期: 2013-11-06)

持多线程技术,对于不支持多线程技术的处理器核架构来说,实现难度较大。(3)使用多个 FPGA 构建整个系统<sup>[8,12-14]</sup>。这样做的好处是准确,可以做到模拟系统和目标系统完全等同,且相对简单,不需要对目标系统进行重新构建。但使用多片 FPGA 进行实现时,需要对原芯片进行分区,以便把各个部分映射到不同的 FPGA 中,工程复杂度较高,并且多片 FPGA 之间较大的通信延迟以及由于引脚个数限制而必须采用的引脚复用方法,都会带来模拟速度的损失。为了解决使用 FPGA 进行大规模片上多核处理器模拟的难题,本文提出了一种新的 FPGA 模拟方法。该方法通过真实的目标处理器核与伪造的目标处理器核的混合,并使用真实的目标处理器核运行实际应用,来评估片上多核处理器的性能,有效地

克服了 FPGA 的容量限制问题。

## 1 背景介绍

片上多核处理器的典型结构如图 1 所示。它一般包含几个对称的处理器核,它们各自拥有私有的一级/二级缓存,通过互联总线连接到共享的最后一级大容量缓存上,通过集成的内存控制器访问片外内存资源,并通过片间互联接口进行多个芯片之间的高速通信以及连接输入输出(input/output, I/O)设备。共享资源的争夺以及多个处理器核之间复杂的交互行为给片上多核处理器的性能评估带来了很大困难。

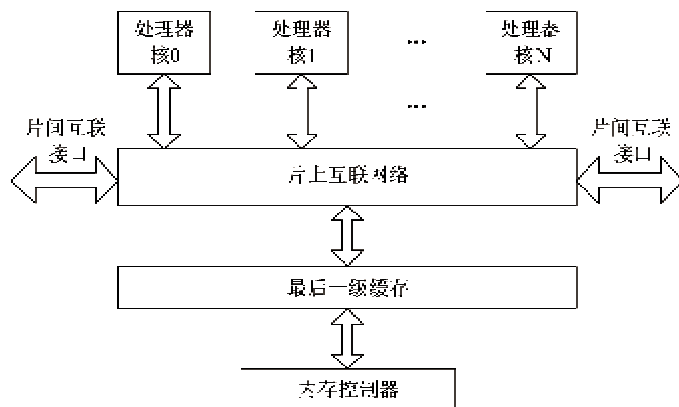


图 1 一个典型的片上多核处理器结构

使用历史痕迹(trace)进行性能评估是人们常常使用的另一种方法。使用 trace 的方法可以屏蔽掉不关心的模块,只对目标模块进行单独模拟。这样做的好处是:简单、易于使用,同时由于减少了模拟量,模拟速度也大大提高。但使用 trace 进行模拟的一个重要缺点是 trace 是静态的,而在实际系统中,多次运行相同应用得到的 trace 并不总是相同的。

为了确保处理器核功能的正确性,人们常常使用 FPGA 进行流片前的功能验证。面对 FPGA 的容量限制,人们一般对原系统进行剪裁,以便将整个系统映射到少数几个 FPGA 中。由于一个复杂的高性能多发射乱序执行处理器核需要占用大量的 FPGA 资源,因此人们一般不得不只保留一个处理器核。对于主要目的在于验证处理器核功能正确性的设计人员来说,由于片上多核处理器的多个处理器核一般是完全对称的,这样做并没有太大的问题。但如

果想要验证评估缓存一致性以及互联网络、内存控制器等共享功能部件的性能,这样做就无法达到目的。因为这些共享资源在单个处理器核独占和多个处理器核共享的情形下会表现出不一样的行为特征,体现出较大的性能差别,从而无法用单核环境下的性能来预测多核环境下的性能。

为了利用 FPGA 模拟的速度优势,同时为了准确模拟片上多核处理器,本文提出了一种将 FPGA 单核功能验证与 trace 模拟结合起来的片上多核处理器模拟方法。

## 2 片上多核处理器模拟方法

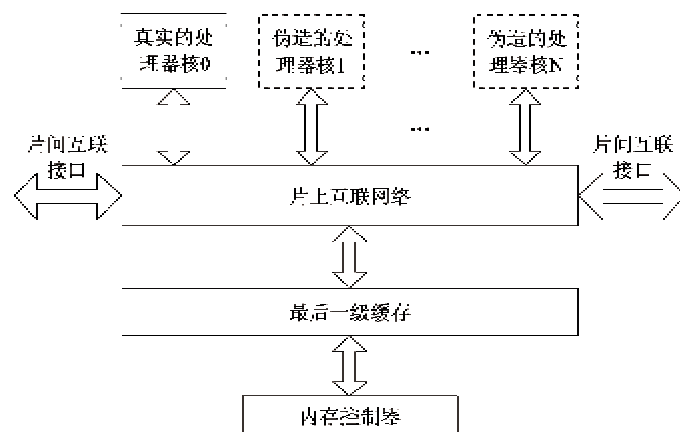
本文提出的片上多核处理器模拟方法如图 2 所示。其主要思想在于去除片上多核处理器中的重复部件——等同的多个处理器核,然后用伪造的处理器核来代替。由于伪造的处理器核不需要实现详细

的流水线等结构,他们占用的 FPGA 资源非常少,因此我们可以添加很多伪造的处理器核到目标系统中,并把整个系统全部映射到一个或两个 FPGA 中。同时由于我们保留了一个真实的目标处理器核,因此可以使用这个真实的目标处理器核来运行实际应用,产生可信的运行结果。

与单核 FPGA 功能验证相比,由于本文方法添加了多个伪造的处理器核,从而构造了一个多核环境,因此可以有效地评估共享部件在多核环境下的性能。同时,与完全基于 trace 流的方法相比,由于我们的系统中有一个真实的处理器核,它可以精确

地反映在共享资源竞争的情况下,真实的处理器核会受到的影响,并动态地反映到实时的 trace 流中,而这是传统的 trace 方法做不到的。

例如,如果需要评估缓存一致性,我们可以实现一个伪造的一级缓存,该缓存对一致性消息的处理过程与真实的一级缓存完全一样,但不需要存储数据内容,只需要存储地址 tag 和其它的控制位,对于缓存行容量为 64 字节的缓存来说,其面积大约为真实缓存的 1/8 左右。我们只需要伪造一些激励让伪造的缓存主动发出与其它处理器核地址相同的共享访问,就可以模拟缓存一致性的行为。



core0 是真实的目标处理器核,core1 到 coreN 是伪造的目标处理器核,它们并没有流水线、运算单元、一级/二级缓存等功能部件,它们只是用来产生与真实处理器核类似的访问请求,也就是说它们是一个处理器核 trace 生成器

图2 混合真实的处理器核与伪造的处理器核的片上多核处理器模拟方法

对于互联网络、内存控制器等共享资源来说,由于它们的主要任务分别是数据路由和数据存取,其收到的访问请求是来自真实的处理器核还是伪造的处理器核,对它们来说并没有差别。因此,添加伪造处理器核的方法可以很好地模拟多核环境来评估这些共享资源的性能。

在片上多核处理器中,处理器核对外的接口主要是数据访问接口和一致性消息传递接口。伪造的处理器核的功能结构如图3所示。为了使伪造的处理器核尽量真实,本文需要对模拟的应用程序进行行为特征提取。应用程序行为特征提取可以采用两种方法:(1)将真实处理器核的接口信号作为输入提供给伪造的处理器核,然后让伪造的处理器核生成与真实处理器核特征相同的 trace,以模拟典型的多用户相同进程的情况。(2)先统计出待评估应用的行为特征,例如访存密度、一级缓存缺失率、读写比例等特征信息,然后让伪造的处理器核生成符合这些特征信息的 trace 流,以模拟多用户混合进程的情况。

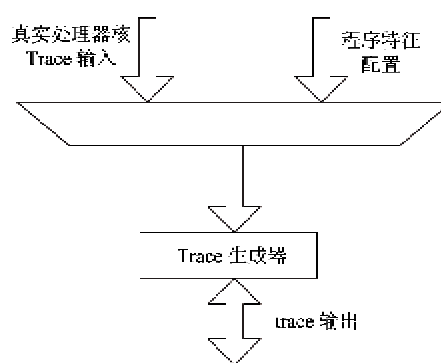


图3 伪造的处理器核的结构示意图

### 3 模型验证与实现

为了验证本文提出的多核处理器模拟方法的有效性,本文使用流片后的真实芯片作为参照系统,通过将参照系统和 FPGA 模拟系统配置为尽量相同的软硬件参数,让它们运行相同的测试程序,然后比较结果的相关度。

本文使用的参照系统是一款四核片上对称多核

理器,它的基本架构与图 1 类似。它的硬件配置参数见表 1。

表 1 目标系统硬件配置

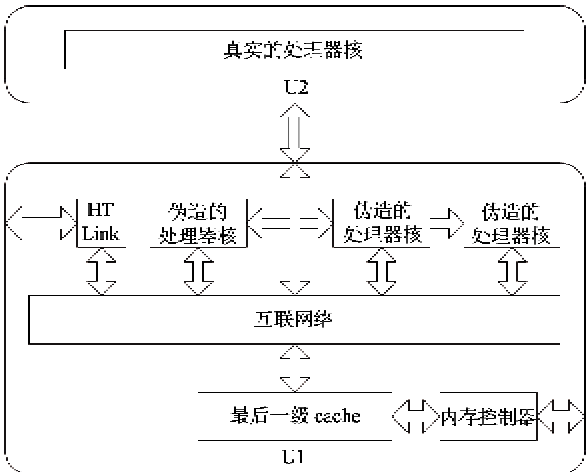
| 部件     | 配置                                          |
|--------|---------------------------------------------|
| 处理器核   | 四个 4 发射乱序执行超标量处理器核,9 级流水线,处理器核可动态关闭         |
| 一级指令缓存 | 私有的 64kB,4 路组相连                             |
| 一级数据缓存 | 私有的 64kB,4 路组相连                             |
| 二级缓存   | 私有的 128kB,8 路组相连                            |
| 最后一级缓存 | 共享的 4M 字节,4 路组相连,4 个 Bank 体,每个 Bank 体 1M 字节 |
| 片上网络   | AXI 交叉开关                                    |
| 内存控制器  | DDR2/3 乱序调度内存控制器,32 项命令队列                   |
| 片外内存   | DDR3 UDIMM,2GB,1Rank                        |

本文使用和真实芯片完全相同的源代码进行 FPGA 综合实现,FPGA 平台系统的架构与真实芯片完全相同,除了将处理器核 1 到 3 用伪造的处理器核来代替以及由于 FPGA 容量限制将最后一级缓存的容量缩小为 512kB。为了与 FPGA 平台等同,我们将真实芯片的最后一级缓存配置为只使用 1 个 Bank 体,也就是说真实芯片的最后一级缓存的容量缩小为 1MB。

为了防止伪造的处理器核产生的访存 trace 影响真实处理器核运行的正常程序的功能,我们保留了一段内存地址空间供伪造的处理器核使用,操作系统看到的地址空间只是整个硬件内存地址空间的一部分。在本实验中,操作系统使用内存地址空间的低地址部分,伪造的处理器核使用高地址部分。

本文使用 S2C 公司的 TAI Logic Module 作为 FPGA 模拟平台,它提供了两片 Xilinx Virtex6 FPGA 作为用户可编程资源,板上自带了 DDR2/3 SDRAM 标准 SODIMM 接口,并可以通过扩展引脚与用户自定义的子板相连。本文使用 HyperTransport 接口与自己定制的子板相连,子板提供南北桥以及 I/O 资源。

由于整个芯片的规模较大,无法用一个 FPGA 完全实现,因此把整个芯片分为两部分,其中,将真实的处理器核 core0 单独放在其中的一个 Virtex6 FPGA(U2)中,将芯片的剩余部分放在另外一个 Virtex6 FPGA(U1)中,它们之间通过 AMBA AXI 总线相连。AXI 总线除了进行处理器核主动发出的读写数据传送外,还负责进行一致性消息包传送。伪造的处理器核也全部放在 U1 中,以减少 U1 和 U2 之间的互连引脚数量。FPGA 系统分割图见图 4。



U2 中包含真实的处理器核(包括一级/二级缓存);U1 中包含伪造的处理器核、片上互连网络、HyperTransport 控制器、共享的最后一级缓存、内存控制器等资源;U1、U2 是相同的 Xilinx Virtex6 FPGA,这里为了节省空间,对 U2 进行了缩小

图 4 FPGA 系统的结构划分图

使用 FPGA 工具进行自动综合和物理实现后的 FPGA 资源利用情况见表 2。可以看到,添加 3 个伪造的处理器核后的面积开销相比原始代码只是增加了很小的比例。图 5 是 FPGA 平台原型图。

表 2 Xilinx Virtex6 资源利用情况

| FPGA 资源    | U2  | U1  | U1(添加 3 个伪造的相似核) | U1(添加 3 个伪造的 stream 核) |
|------------|-----|-----|------------------|------------------------|
| Registers  | 6%  | 11% | 13%              | 13%                    |
| LUTs       | 38% | 27% | 31%              | 32%                    |
| Block RAMs | 21% | 22% | 24%              | 23%                    |

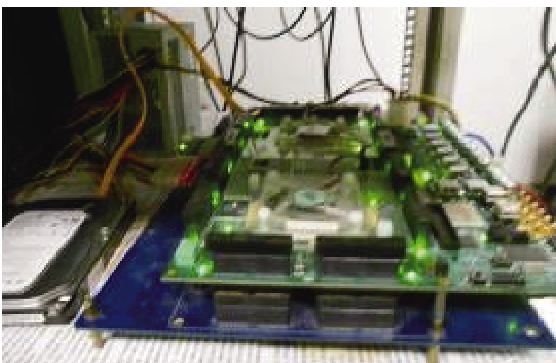


图 5 FPGA 平台原型,上层的绿色的 PCB 板是 S2C 公司的 Tai LM FPGA 平台,下层的蓝色的 PCB 主板是本文制作的子板,它提供北桥、网卡、硬盘接口、鼠标、键盘、启动 flash 以及串口等资源

由于时间限制,我们没有对专用集成电路(application specific integrated circuit, ASIC)设计的寄存器传输级(register transfer level, RTL)源码进行针对 FPGA 实现的手工优化,直接使用工具进行自动综合和布局布线,目前实现的 FPGA 平台的处理器核域的运行频率为 50MHz(包括处理器核、一级缓存、二级缓存、互连网络、最后一级缓存等)。由于真实芯片的内存控制器目前运行在 533MHz 的频率,因此将真实芯片的处理器核域的运行频率设置为 1.06GHz,使得处理器核/内存控制器具有 2:1 的频率比。同样地,我们将 FPGA 平台的内存控制器设置为 25MHz,使得 FPGA 平台和真实芯片具有相同的处理器核/内存控制器频率比。为了弥补 FPGA 平台较低的时钟频率,同样降低了 FPGA 平台上操作系统的时钟中断频率。

本文运行未修改的 Linux 2.26.36 内核,使用标准性能评估公司(SPEC)的评估处理器在真实应用中的计算性能的有效工具 SPEC CPU2006 和访存密集型测试程序 stream 作为测试程序集。通过运行不同个数的 SPEC CPU2006 进程以及混合运行 SPEC CPU2006 与 stream 程序,来评估 FPGA 模拟系统的有效性。

## 4 实验结果与分析

本实验分别模拟了 SPEC 多个用户进程同时运行以及混合运行 SPEC CPU2006 程序与 stream 程序的情况。模拟多个用户进程同时运行时,本实验将真实处理器核 core0 的接口信息作为输入提供给伪造的处理器核,伪造的处理器核对 core ID、AXI ID、访问地址等信息进行处理,然后输出给互连网络,这样伪造的处理器核产生的 trace 与真实处理器核基本相同。模拟 stream 程序时,首先在真实的系统中,使用 oprofile 工具统计出二级缓存的缺失率、访存密度以及读写比例,然后将这些特征信息提供给伪造的处理器核,让其产生符合这些特征信息的访存 trace。

由于 FPGA 模拟系统的运行速度仍然相对较慢,因此我们使用了 SPEC CPU2006 的 train 测试集,运行一遍完整的 train 测试集仍然需要花费将近 6 天的时间。

在 SPEC CPU2006 的测试试验中发现,多次运行或者使用不同的测试顺序都会对实验结果带来偏差。经过分析,我们认为是程序运行前的缓存状态

对程序的运行时间产生了影响。因此,编写了一个 cache\_clean 程序,它对一段大于最后一级缓存容量的连续地址空间进行只读操作,以便将所有的缓存行更新为 clean 态,这样当新的程序开始执行时,它面对的缓存状态都是相同的。通过修改 SPEC CPU2006 的运行脚本,在每次运行测试程序前,先运行一遍 cacheclean 程序。经过这样的修改,多次测试 SPEC CPU2006 的结果非常接近,误差在 1% 以内。

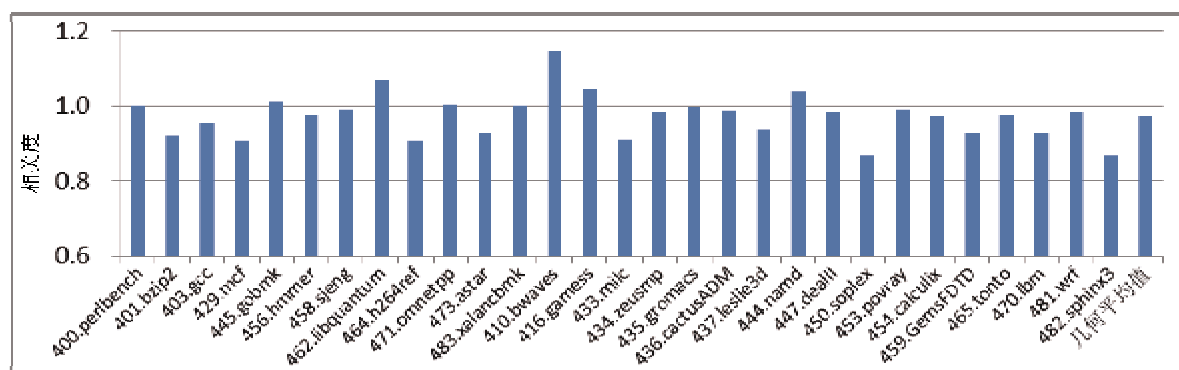
在 FPGA 模拟系统中,模拟双核系统对应于添加 1 个伪造的处理器核,模拟四核系统对应于添加 3 个伪造的处理器核。我们总是使用真实处理器核 core0 来运行 SPEC CPU2006 测试程序,使用它的运行时间作为性能衡量标准。伪造的处理器核模拟 SPEC CPU2006 进程时,直接使用真实处理器核 core0 的接口信息作为输入,产生与 core0 相似的 trace 流。伪造的处理器核模拟 stream 程序时,使用预提取的 stream 特征信息作为输入,产生特定的访存模式。

对于真实芯片平台和 FPGA 平台,首先运行单用户的 SPEC CPU2006 程序,得到单核下的基准运行时间,然后运行多用户 SPEC CPU2006 程序以及运行混合的 SPEC 与 stream 程序,得到多核下的 SPEC 运行时间,然后除以单核下的基准运行时间,计算出多核下的性能下降比例。然后用 FPGA 平台的性能下降比例除以真实芯片平台的性能下降比例,以表征两个平台多核模拟的相关度。比值越接近 1 表明两个平台的模拟相关度越高,也就是说 FPGA 平台的模拟准确度越高。

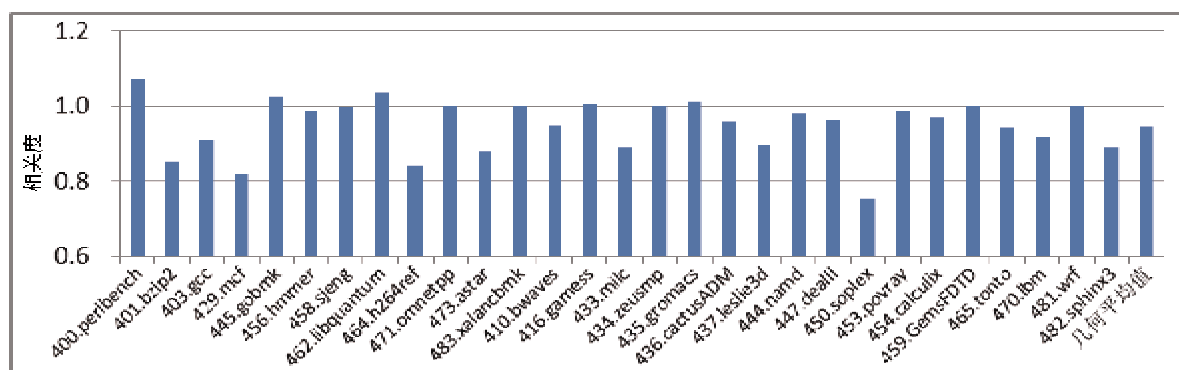
我们分别测试了 2 个 SPEC CPU2006 进程、4 个 SPEC CPU2006 进程以及混合 1 个 SPEC CPU2006 进程和 1 个 stream 进程、混合 1 个 SPEC CPU2006 进程和 3 个 stream 进程的情况。我们使用了 SPEC CPU2006 train 测试集,每种测试组合都运行了两次以上,然后取它们的算术平均值作为最终结果。实验结果见图 6。

从图 6(a)(b)可以看到,对于模拟多用户 SPEC CPU2006 进程来说, FPGA 平台和真实芯片平台的相关度很高,绝大部分 SPEC CPU2006 程序的相关度约为 0.9~1.1,模拟 2 进程与 4 进程的相关度的几何平均值分别达到了 0.97 和 0.95。说明通过复制真实处理器核运行过程中的实时 trace 来模拟多核芯片下同时多用户进程的方法是有效的。

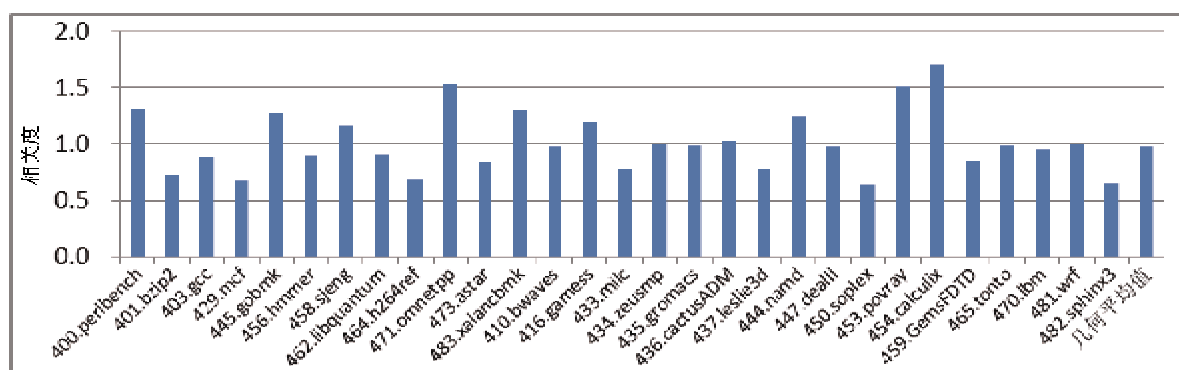
从图 6(c)(d)可以看到,对于模拟混合 SPEC



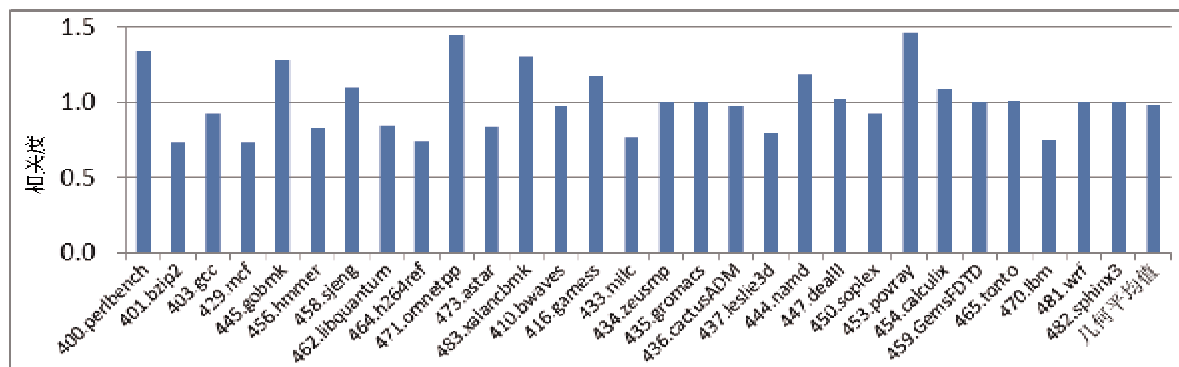
(a) 2 个 SPEC CPU2006 进程



(b) 4 个 SPEC CPU2006 进程(为了减少模拟时间,浮点部分使用了 test 测试集)



(c) 混合 1 个 SPEC CPU2006 进程与 1 个 stream 进程



(d) 混合 1 个 SPEC CPU2006 进程与 3 个 stream 进程(为了减少模拟时间,浮点部分使用了 test 测试集)

图 6 真实芯片与 FPGA 模拟平台的相关度

CPU2006 进程与 stream 进程来说,大部分 SPEC CPU2006 程序的相关度接近于 1,少部分程序的相关度偏差较大,最小值和最大值分别达到了 0.64 和 1.7。有趣的是,即使部分程序的偏差较大,但所有程序相关度的几何平均值非常接近于 1,模拟 2 核与 4 核处理器分别为 0.98 和 0.99。混合 stream 进程时,FPGA 平台模拟相关度下降的原因与程序特征信息提取和我们目前实现的 trace 生成机制有关。目前我们实现的是最简单的 trace 生成方法。在本例中,我们使用 oprofile 工具统计出的 stream 程序的二级缓存缺失率大约为 1/64 个缺失每个时钟周期,于是我们的 trace 生成方法为:每 64 个时钟周期发出一个访存命令,访存地址生成方法为简单的线性增加,而没有考虑程序启动时和结束前的访存非平均分布情况以及程序运行过程中存在的多个地址流情况。通过实现更复杂与精巧的 trace 生成方法,模拟准确度将会进一步提高。

## 5 结 论

本文提出了一种新颖的基于 FPGA 的片上多核处理器模拟方法,并通过与真实芯片的比较验证了它的有效性。实验证明,该方法可以很容易地实现 50MHz 以上的全系统模拟速度,使得能够在几天的时间内运行完整的多用户 SPEC CPU2006 train 测试集,如果使用基于 RTL 软件模拟的方法,则需要花费几个月,甚至几年的时间,这样的时间代价是设计人员无法承受的。

本文提出的 FPGA 模拟方法,可以有效用于片上多核处理器片上网络、共享缓存、内存控制器等共享部件的功能验证和性能评估。我们下一步将运用本文提供的平台进行片上多核处理器高性能内存系统的研究和设计工作。

### 参考文献

- [ 1 ] Austin T, Larson E, Ernst D. SimpleScalar: an infrastructure for computer system modeling. *Computer*, 2002, 35(2): 59-67
- [ 2 ] Rosenblum M, Bugnion E, Devine S, et al. Using the SimOS machine simulator to study complex computer systems. *ACM Transactions on Modeling and Computer Simulation*, 1997, 7(1): 78-103
- [ 3 ] Binkert N, Beckmann B, Black G, et al. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 2011, 39(2): 1-7
- [ 4 ] Hamerly G, Perelman E, Lau J, et al. Simpoint 3.0: faster and more flexible program phase analysis. *Journal of Instruction Level Parallelism*, 2005, 7(4): 1-28
- [ 5 ] Tan Z, Waterman A, Cook H, et al. A case for FAME: FPGA architecture model execution. In: *Proceedings of the 37th ACM/IEEE International Symposium on Computer Architecture*, Saint-Malo, France, 2010. 290-301
- [ 6 ] Pellauer M, Adler M, Kinsy M, et al. HAsim: FPGA-based high-detail multicore simulation using time-division multiplexing. In: *IEEE 17th International Symposium on High Performance Computer Architecture*, San Antonio, USA, 2011. 406-417
- [ 7 ] Miller J E, Kasture H, Kurian G, et al. Graphite: a distributed parallel simulator for multicores. In: *IEEE 16th International Symposium on High Performance Computer Architecture*, Bangalore, India, 2010. 1-12
- [ 8 ] Wawrzynek J, Patterson D, Oskin M, et al. RAMP: research accelerator for multiple processors. *Micro, IEEE*, 2007, 27(2): 46-57
- [ 9 ] Chiou D, Sunwoo D, Kim J, et al. Fpga-accelerated simulation technologies (FAST): fast, full-system, cycle-accurate simulators. In: *Proceedings of the 40th Annual IEEE/ACM international Symposium on Microarchitecture*, Chicago, USA, 2007. 249-261
- [ 10 ] Chung E S, Papamichael M K, Nurvitadhi E, et al. Prototflex: towards scalable, full-system multiprocessor simulations using fpgas. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 2009, 2(2): 15
- [ 11 ] Tan Z, Waterman A, Avizienis R, et al. RAMP gold: an FPGA-based architecture simulator for multiprocessors. In: *Proceedings of the 47th Design Automation Conference*, Anaheim, USA, 2010. 463-468
- [ 12 ] Schelle G, Collins J, Schuchman E, et al. Intel nehalem processor core made FPGA synthesizable. In: *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, USA, 2010. 3-12
- [ 13 ] Asaad S, Bellofatto R, Brezzo B, et al. A cycle-accurate, cycle-reproducible multi-FPGA system for accelerating multi-core processor simulation. In: *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, USA, 2012. 153-162
- [ 14 ] Wee S, Casper J, Njoroge N, et al. A practical FPGA-based framework for novel CMP research. In: *Proceedings of the 2007 ACM/SIGDA 15th International Symposium on Field Programmable Gate Arrays*, Monterey, USA, 2007. 116-125

## A novel method for simulating chip multi-core processors using FPGA

Chen Xinke<sup>\* \*\* \*\*\*</sup>, Huang Shuai<sup>\*\*\*\*</sup>, Wang Huandong<sup>\*\*\*\*</sup>, Wu Ruiyang<sup>\* \*\* \*\*\*</sup>, Zeng Lu<sup>\* \*\* \*\*\*</sup>

(<sup>\*</sup> State Key Laboratory of Computer Architecture(Institute of Computing Technology,  
Chinese Academy of Sciences), Beijing 100190)

(<sup>\*\*</sup> Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(<sup>\*\*\*</sup> University of Chinese Academy of Sciences, Beijing 100049)

(<sup>\*\*\*\*</sup> Loongson Technology Corporation Limited, Beijing 100190)

### Abstract

To solve the capacity limitation of field programmable gate arrays (FPGAs) when using them to conduct large scale simulations of chip multi-core processors, a novel FPGA-based simulation method was put forward. By mixing real processor cores with pseudo processor cores, the method can simulate a whole chip's multi-core processors by using only one FPGA or two, and can overcome the FPGA capacity limitation effectively while alleviating the harmfulness to the characterization of chip multi-core processors behavior. A cycle-accurate full system simulation was carried out by using the method, and the correctness of the proposed method was verified by using the corresponding taped out chip. A simulation speed beyond 50MHz was achieved easily, which was 100,000 times faster than the logic level software simulation with the same design. Under the vast increase in simulation speed, the unmodified Linux OS can be started and the complete multi-user SPEC CPU2006 benchmark can be run by using the train test suite. The proposed simulation method of mixing real cores with pseudo-cores can be easily, effectively used for chip multi-core processors's logic verification and performance validation.

**Key words:** emulation, simulation, modeling, field programmable gate arrays (FPGAs), chip multi-core processor, pseudo processor cores