

CMT 模拟器的设计与实现

杨 华, 崔 刚, 吴智博, 刘宏伟

(哈尔滨工业大学计算机科学与技术学院, 哈尔滨 150001)

摘 要: 片上多线程(CMT)是未来高性能处理器的发展方向, 而软件模拟是处理器体系结构研究和设计中不可或缺的技术手段。该文基于 SimpleScalar 工具集设计并实现了 CMT 节拍级模拟器——OpenSimCMT, 对 CMT 体系结构的设计和评测提供支持。OpenSimCMT 特点如下: (1)支持同时多线程(SMT)和片上多处理器(CMP)的模拟; (2)架构开放, 配置灵活, 可根据具体研究目标随时进行扩展, 添加新的模拟内容及相关统计; (3)功能全面, 对线程间资源竞争与共享、各功能单元、流水段、分支预测、多级 cache 等全方位模拟, 模拟结果准确。

关键词: 片上多线程; 模拟; 微处理器; 高性能

Design and Implementation of CMT Simulator

YANG Hua, CUI Gang, WU Zhibo, LIU Hong-wei

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

【Abstract】 Chip multithreading (CMT) represents the direction for future high-performance processors, and simulation is an integral part of the processor architecture research and design process. This paper presents the design and implementation of a SimpleScalar-based cycle-level simulator—OpenSimCMT, which serves for designing and evaluating the CMT architectures. OpenSimCMT features that: (1) supporting the simulation of simultaneous multithreading (SMT) as well as chip multiprocessors (CMP); (2) open framework and well configurable, being extendible to accommodate a given research goal, and readily to append new simulation contents and relative statistic; (3) all-around simulation, supporting inter-thread resource sharing and competition, various function units, pipeline, branch prediction, multi-level caches, etc, facilitating accurate simulation.

【Key words】 CMT; simulation; microprocessor; high-performance

1 概述

伴随晶体管集成密度的日益提高, 处理器的发展将逐渐进入片上多线程(CMT)时代^[1]。CMT通过开发线程级并行(TLP)来克服指令级并行(ILP)的不足, 显著提高了处理器的整体处理能力。同时多线程(SMT)^[2]和片上多处理器(CMP)^[3]是两类典型的CMT体系结构, 自提出以来就受到广泛的研究和关注。SMT在宽发射超标量(superscalar)的基础上增加多个硬件线程上下文(hardware context)控制, 使多线程同时“驻留”于处理器中, 不同线程的指令可同时发射, 竞争并共享处理器内部的各种资源, 有效地提高了资源利用率和整体性能。CMP将多个相同的处理核集成在同一芯片中, 共享大容量L2(或更低)cache, 每个核可看作一个独立的处理器, 分别运行不同的程序/线程。CMP旨在简化处理器设计, 将整个片上资源静态划分给多个独立的核, 其缺点是各自的资源无法交叉利用, 当TLP不足时会严重浪费资源。相比之下, SMT中大部分资源是“竞争式共享”, 在TLP和ILP间动态转换, 提高资源的整体利用率, 但也增加了设计和实现的复杂度。

处理器的设计、实现和验证过程日益复杂。软件模拟已成为处理器的先期设计、验证和评估的关键一环, 也是对新体系结构思想进行验证和量化评价的重要手段。模拟不但节省成本, 而且非常灵活, 帮助探索未知的设计空间, 对不同的设计思路和配置选项进行效果评测。由于性能、功耗、造价、可靠性、兼容性、可扩展性等方面的限制和要求, 高效实用的处理器体系结构通常来自对各种设计选项的权衡和取舍(trade-off), 这需要大量的先期模拟实验, 是一个反复比较、

逐渐求精的过程。如果没有模拟器的支持, 评测工作要等到电路级设计(Verilog 验证、FPGA 仿真等)、甚至是实际芯片出来后才能进行, 这是不切实际的。依赖于模拟器的详细程度和设计水平, 模拟结果与实际会有不同程度的偏差, 但这并不妨碍模拟器作为体系结构研究的重要技术手段、对各种设计选项进行相对客观的评测和比较。本文介绍了在 CMT 研究过程所设计并实现的节拍级模拟器——OpenSimCMT。

2 相关工作

SimpleScalar^[4]是由Austin设计的超标量处理器模拟工具集, 包含指令行为验证(sim-safe)、分类统计(sim-profile)、分支行为(sim-brped)、cache行为(sim-cache)、整体性能模拟(sim-outorder)等不同侧重点和详细程度的模拟; 还包括与之配合的交叉编译工具, 用于将高级语言程序(C和Fortran)编译链接成Pisa指令(类似MIPS, 用于研究目的)的可执行程序。由于SimpleScalar的功能比较全面、开放源码、可移植性好(C语言开发)等方面的优点, 一直以来被学术界广泛地扩展和移植, 作为新体系结构思想的验证和评测工具。目前为止, SimpleScalar已发展到 4.0 版(MASE), 而最广泛采用的仍是 3.0 版。除Pisa外, 还出现了支持Alpha、ARM、PowerPC、IA-32等指令集的版本, 并且被成功移植到多种UNIX、Linux和

基金项目: 国家自然科学基金资助项目(60503015)

作者简介: 杨 华(1974 -), 男, 博士研究生, 主研方向: 片上多线程体系结构, 容错处理器体系结构; 崔 刚, 教授、博士生导师; 吴智博, 博士、教授; 刘宏伟, 博士、副教授

收稿日期: 2006-10-18 **E-mail:** yangh@fcl.hit.edu.cn

Windows平台上运行。

SMTSIM^[2]由Tullsen等设计,是最早的SMT性能模拟器。虽然对SMT的支持比较完备,但存在如下一些不足之处:支持范围小(只支持Alpha指令集);统计项较少、且可配置性较差(参数较少,经常要修改源代码来进行配置),给大规模实验造成不便;运行平台少(最初只支持Alpha + Unix,最近才被移植到X86 + Linux上),制约了其适用范围。

3 OpenSimCMT 的设计与实现

CMT是新兴的、也是最复杂的处理器体系结构之一,影响因素很多。其研究过程需要大量的实验评测,迫切需要一个准确、信息量大、运行速度快、配置方便、扩展灵活的模拟器。为此,对SimpleScalar 3.0进行扩展,设计并实现了OpenSimCMT。

3.1 设计要点

OpenSimCMT的设计目标是准确、实用的基础实验平台,为各种CMT设计思路提供验证和评测支持。图1给出了OpenSimCMT的基本结构——CMT控制逻辑作用于8级流水线,并加强主存、各级cache、各功能单元、寄存器文件等重要资源的管理。其设计要点在于:

(1)清晰、开放的架构:根据不同的研究目的,需要随时加入对新设计选项的支持,保持整个模拟器架构的清晰和开放至关重要。除了继承SimpleScalar本身的开放性,OpenSimCMT对新增的结构和函数采用功能明确,结构简单的模块化设计。对原SimpleScalar中容易含混的地方重新设计,并将过于复杂的任务模块拆分成多个函数完成。例如:原SimpleScalar中的寄存器重命名过程比较隐晦(宏实现,对物理寄存器数量无明确限制),而OpenSimCMT明确添加了用于寄存器重命名过程的数据结构、函数和相关统计。

(2)可配置性良好:模拟过程往往涉及大量选项有或无、多或少、粗调和微调等配置细节,方便灵活的配置选项是实现高效率模拟工作的关键。除了继承SimpleScalar良好的可配置性,OpenSimCMT中添加CMT的相关配置选项,可通过命令行或配置文件的形式传递实验配置,无需修改源代码,这与SMTSIM相比有很大优势。

(3)评测信息全面:模拟实验是为了提供直观而全面的评测信息,其准确和全面与否直接影响研究人员的判断和取舍。OpenSimCMT增加了大量的相关统计项,全面反映CMT各线程的运行、对各类资源的占用和竞争情况。

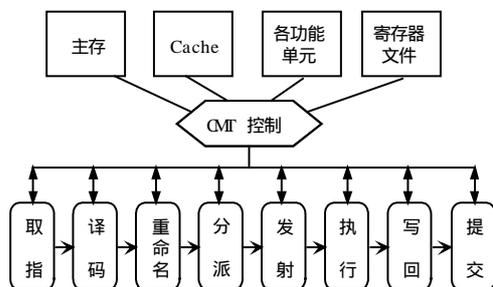


图1 OpenSimCMT的基本结构

3.2 实现过程

鉴于CMT及其模拟实现的复杂性,在OpenSimCMT实现过程中采用稳扎稳打、步步为营的策略。每个步骤完成之后都要经历大量的测试,确保其准确和高效,为进一步开发打下基础。大致的实现过程如下:

(1)将原SimpleScalar的RUU(register update unit)结构解

耦合为ROB(reorder buffer)和RS(reservation station)两部分,配置和使用更加灵活,符合大多数处理器的结构习惯。明确了寄存器文件的结构和实现,使寄存器分配过程更清晰可控。增加了指令等待队列(IQ)结构及相关控制。

(2)复制线程上下文,初步支持多线程的同时运行。将SimpleScalar中涉及线程上下文的数据结构进行数组化处理,并扩展装载函数(loader),支持多线程的同时装载。

(3)增强存储层次结构(memory hierarchy)的管理,包括各线程虚地址空间的相对偏移(测试程序(如SPEC 2000)运行在模拟器之上,无操作系统调度,需将各线程的地址空间错开,真实反映出各线程对各级cache、分支预测、TLB等资源的占用和竞争情况。)、各级cache、分支预测、I-TLB和D-TLB等机构的分立或共享策略。基本实现对CMP体系结构的模拟。

(4)增加对SMT的支持。加入包括Round-Robin和ICount取指、OLDEST_FIRST发射、cache和流水线资源的竞争和共享等SMT体系结构的基本特性。

(5)添加CMT相关信息的统计;运行SPEC 2000进行大规模测试,综合对比并调整,消除可能的评测偏差。优化数据结构和函数的实现,提高模拟速度。对主要选项给出缺省配置(反映主流趋势),方便设置过程。增加部分设置开关,可根据不同研究目的有选择地打开或关闭,提高模拟器运行效率。例如:在研究多线程的cache竞争行为时,可以关掉内部流水线资源的相关记录和统计选项,加快模拟速度。

3.3 应用实例

OpenSimCMT功能比较完备、架构开放,提供了CMT模拟实验的基础平台,实际应用时可根据具体研究目的方便灵活地进行扩展,已经将其用在一系列CMT研究工作中,取得令人满意的效果。应用实例如下:(1)提出并验证2L-MuRR机制^[5],缓解多线程并行对寄存器文件的设计压力;(2)提出并验证TSRR机制^[6],提升SMT对寄存器文件和整体资源的利用效率;(3)提出并验证SD-SRT机制^[7],消除原SRT中冗余的取指过程,缓解取指部件和I-cache的压力。

4 结束语

OpenSimCMT实现对CMT体系结构准确而高效的模拟。其设计、实现、及成功应用表明:

(1)保持设计上的兼容性非常重要,既方便继承前人成果,还有助于使用和推广,OpenSimCMT继承SimpleScalar工具集的优点,支持Alpha和Pisa指令集,并已在X86 + Linux和UltraSPARC + Solaris平台上测试和应用。

(2)模拟器的设计和实现过程中,保持清晰、开放架构至关重要,这样才能根据实际的模拟需求方便灵活地扩充,达到实用快速的模拟评测。

(3)CMT体系结构复杂、设计选项众多,评测一类设计远比评测一个设计重要,这增加了OpenSimCMT的实现难度,但有助于给出相对客观的比较结果,为判断取舍提供有力依据。可以预见,随着CMT体系结构的日益复杂及研究的不断深入,对模拟器的依赖程度继续增加,设计并实现高效、准确、灵活的模拟器将更加重要。

参考文献

1 Spracklen L, Abraham S G. Chip Multithreading: Opportunities and Challenges[C]//Proc. of the 11th Int'l Symposium on High-performance Computer Architecture, San Francisco, 2005.

(下转第255页)