

乱序超标量处理器核的性能分析与优化*

孙彩霞, 隋兵才, 王 蕾, 王永文, 黄立波, 李文哲, 王俊辉
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘要:随着处理器微体系结构日益复杂,性能分析在处理器研制过程中的作用越来越重要。常用的性能分析方法是建立性能模型,该方法主要用于研制初期的设计空间探索,如果用于微体系结构级的分析和优化,速度和精度都会成为限制因素。因此,提出一种基于计数器的性能分析方法,该方法以项目组已经完成的一款处理器核的硬件实现代码为基础,在处理器核外部添加一个专用性能监测单元,收集微体系结构分析和优化需要的各种事件,并通过结果分析器对统计的事件进行分析,得到微体系结构实现的性能受限因素。采用此方法,在现场可编程门阵列原型系统上对 SPEC CPU2000 测试程序运行时的性能受限因素进行分析,并根据分析结果采取相应的优化措施,优化后的处理器核性能得到了明显提升。

关键词:性能分析;计数器;处理器核;微体系结构

中图分类号:TP302.1 **文献标志码:**A **文章编号:**1001-2486(2016)05-014-06

Counters based performance analysis and optimization of an out-of-order superscalar processor core

SUN Caixia, SUI Bingcai, WANG Lei, WANG Yongwen, HUANG Libo, LI Wenzhe, WANG Junhui
(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: With the ever-increasing design complexity in the processor micro-architecture, performance analysis becomes more and more important in the research and design of processors. Performance models are used widely in the performance analysis, which are more suitable for the design space exploration in the early stage. When used in micro-architecture optimizations, the accuracy and the speed of performance models are the limiting factors. Therefore, a performance analysis method based on counters was proposed. In this method, the RTL register transfer level code of a processor core was used as a baseline, and a specialized performance monitor unit was added to collect the events needed by the micro-architecture analysis and optimization. Then the collected events were sent to a result analyzer, where the factors affecting the performance were obtained. By adopting the method, we analyzed what affects the performance in running SPEC CPU2000 benchmarks on FPGA (field-programmable gate array) prototyping, and optimized the micro-architecture of processor core according to the analysis results. The performance of the optimized processor core is improved obviously.

Key words: performance analysis; counters; processor core; micro-architecture

为了追求更高的处理器性能,多发射、分支预测、寄存器重命名、乱序执行、前瞻执行等技术被广泛采用,使得处理器微体系结构变得日益复杂,仅仅依靠直觉或经验设计处理器变得几乎不可能。性能分析成为处理器研制过程中必不可少的一个环节。

性能分析可以分为硅前和硅后两个阶段。硅前分析主要进行处理器设计空间探索,确定满足期望目标的处理器实现,常用手段是使用性能模拟器。主流处理器生产厂商虽然没有过多披露他们在性能分析方面的实现细节,但是从公开的资料中能够了解到他们对性能分析的重视以及为此

所做的努力。IBM 公司开发了一整套用于 PowerPC 处理器微体系结构探索的环境和工具 (Micro-architecture Environment Toolset, MET)^[1], 2004 年 Singhal 等^[2]介绍了 Pentium4 处理器性能分析过程中使用的踪迹驱动的性能模型,中科院计算机所在龙芯系列处理器研制过程中开发了 Sim-Godson 性能模拟器^[3]。不仅在工业界,还在学术界发布了很多好用的开源性能模拟器,如 SimpleScalar^[4], GEMS^[5], M5^[6], Gem5^[7]等。性能模拟器的抽象层次通常较高,但精度有限。此外为了加快模拟速度,模拟时通常会采用一些加速技术,比如采样^[8],这些技术会带来精度上的损

* 收稿日期:2015-11-25

基金项目:国家自然科学基金资助项目(61103011,61170045,61402501)

作者简介:孙彩霞(1979—),女,黑龙江绥化人,副研究员,博士,硕士生导师,E-mail:cxsun@nudt.edu.cn

失。硅后分析主要根据芯片实际表现来验证和优化硅前分析方法或进行编译器的优化^[9-10],常用手段是使用硬件性能计数器。现代处理器几乎都在硬件上实现了性能监测单元^[11-12],以对一些事件进行统计,并通过性能计数器呈现出来。Intel VTune 性能分析器就是利用这些计数器进行应用的性能优化。但是受限于硬件实现代价,处理器通常只设置少量性能计数器,所统计的事件也非常有限,不足以指导微体系结构级的性能分析与优化。

本文提出在硅前利用性能计数器分析处理器核在运行应用程序时的性能受限因素,指导处理器核微体系结构的优化。该方法以项目组已经研制完成的一款处理器核(简称 XMC)的硬件实现代码为基础,在处理器核外部添加一个专用性能监测单元,收集微体系结构分析需要的各种事件,并通过结果分析器对统计的事件进行分析,得到微体系结构实现的性能受限因素,以指导下一代处理器核的设计。采用此方法,我们在 FPGA 原型系统上对 SPEC CPU2000 测试程序运行时的性能受限因素进行了分析,并根据分析结果采取了相应的优化措施,优化后的处理器核性能得到了明显提升。

1 XMC 处理器核微体系结构

图 1 给出了 XMC 处理器核的微体系结构。每个周期,取指单元(Instruction Fetch Unit, IFU)取出 128 位、4 条指令存入指令队列(instruction queue);每个周期最多译码 4 条指令,译码后的指令进入译码队列(decode queue);根据寄存器映射表(rename table),每个周期最多对 4 条指令进行重命名,重命名后的指令进入分派队列(dispatch queue)。如果没有足够的重命名寄存器,重命名过程会发生阻塞;每个周期最多分派 4 条指令,分派后的指令根据其类型进入相应的发射队列,同时也会进入重定序缓冲(ReOrder Buffer, ROB);指令发射是乱序进行的,只要发射队列中指令的操作数准备就绪并且功能单元空闲,指令就可以发射到功能单元执行;指令提交顺序进行,提交时会把重命名寄存器文件(Rename Register File, RRF)中的结果写回到体系结构寄存器文件(Architectural Register File, ARF)。

XMC 使用全局历史缓冲(Global History Buffer, GHB)预测分支方向, GHB 中每一项都是一个 2 位饱和计数器,用于指明分支跳转方向;过程返回指令由返回栈缓冲(Return Stack Buffer,

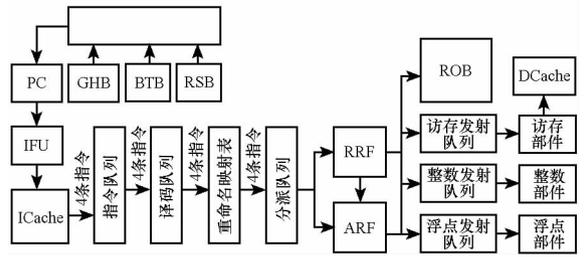


图 1 XMC 处理器核微体系结构

Fig. 1 Micro-architecture of XMC

RSB)提供跳转地址,其他分支指令由分支目标缓冲(Branch Target Buffer, BTB)提供跳转地址。

XMC 在进行寄存器重命名时,整数寄存器和浮点寄存器共享同一个重命名寄存器文件。重命名寄存器文件共有 64 项,每项 64 位。

发射队列采用分布式结构,即每种功能部件拥有独立的发射队列。XMC 处理器核中共有 3 种发射队列:整数发射队列(int issueQ),浮点发射队列(FP issueQ)和访存发射队列(LS issueQ)。每种发射队列负责向相应的功能部件提供指令,其中分支指令在整数部件执行。

2 使用计数器的性能分析方法

受限于硬件实现代价,处理器通常只实现少量硬件性能计数器,所统计的事件也非常有限。为了利用性能计数器提供的信息进行微体系结构层面的性能分析和优化,需要进一步统计微体系结构实现有关的参数。

图 2 给出了基于计数器的性能分析方法的示意图。XMC 的性能监测单元实现了多个性能计数器,可以分时统计多个事件,通过统计事件可以计算出每拍完成的指令数(Instructions Per Cycle, IPC)、分支预测失效率、核内各级 Cache 失效率和 TLB 失效率等,但是这些事件还不足以分析微体系结构的性能受限因素。为此,在 XMC 处理器核外实现了一个专用性能监测单元,进行微体系结构优化需要的任何事件(处理器核内部硬件信号)都可以送到该单元进行统计。数据收集单元收集性能监测单元统计的各类事件,数据收集可以在整个性能统计过程结束后进行,也可以按照一定时间间隔定期进行。收集到的数据送给结果分析器进行分析,寻找处理器核运行应用程序时的性能受限因素。

专用性能监测单元和数据收集单元都用可综合代码进行实现,同 XMC 的寄存器传输级(Register Transfer Level, RTL)代码构成了硬件实现部分,可运行在 NC-Verilog/VCS 模拟器上,也

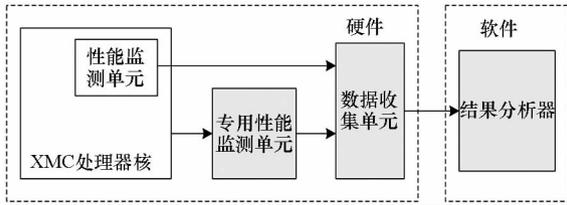


图 2 使用计数器的性能分析方法示意图

Fig. 2 Performance analysis method using counters

可运行在硬件仿真器或现场可编程门阵列(Field-Programmable Gate Array, FPGA)原型系统上。本文面向 SPEC CPU2000 测试程序进行分析,考虑到模拟速度,选择 FPGA 原型系统进行模拟并收集数据^[13],收集的数据通过通用异步收发器(Universal Asynchronous Receiver Transmitter, UART)串口输出送给性能分析器。

在一个应用程序运行结束后,性能监测单元中计数器的值表示这个应用程序执行期间计数器统计的事件发生的次数。如果要继续统计另一个应用程序运行时相应事件发生的次数,那么计数器首先应该被清 0。为了能够通过软件对专用性能监测单元中的性能计数器进行清 0 操作,借用了 XMC 处理器核的性能监测单元中第 0 号性能计数器的有关控制信号,如图 3 所示。对性能监测单元中第 0 号性能计数器的清 0 操作可以同时将所有性能计数器清 0。

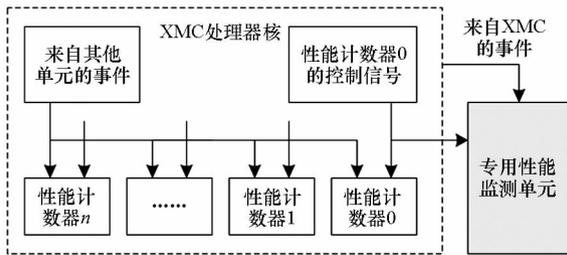


图 3 专用性能计数器的控制

Fig. 3 Controls for specialized performance counters

3 性能受限因素分析

XMC 的机器宽度为 4,理想的 IPC 可以达到 4。但是在 FPGA 原型系统上得到的 SPEC CPU2000 测试程序的 IPC 平均只有 1.23。为了获得微体系结构实现上的性能受限因素,从重命名和分派两处作为切入点进行分析。图 4 给出了 SPEC CPU2000 测试程序平均每拍重命名的指令数(renaming IPC)和平均每拍分派的指令数(dispatching IPC)。

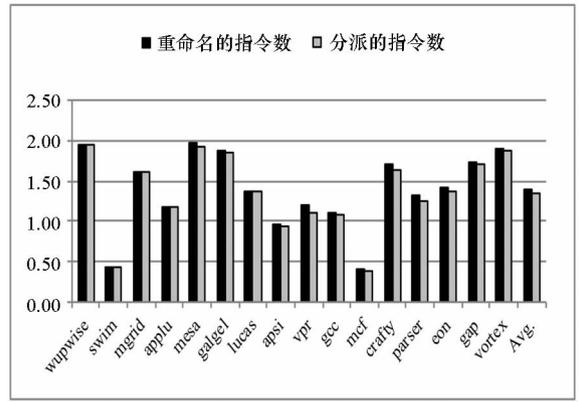


图 4 SPEC CPU2000 平均每拍重命名和分派的指令数

Fig. 4 Renaming IPC and dispatching IPC of SPEC CPU2000

重命名宽度和分派宽度都是 4,但是实际上平均每拍重命名和分派的指令数分别为 1.38 和 1.35,最大时也没有超过 2,是什么导致指令的重命名和分派不能进行呢?

首先分析重命名阻塞原因。为此,统计了如下事件:译码队列无指令,重命名寄存器缺乏和分派队列无空闲项。图 5 给出了 SPEC CPU2000 测试程序重命名阻塞的原因分布。

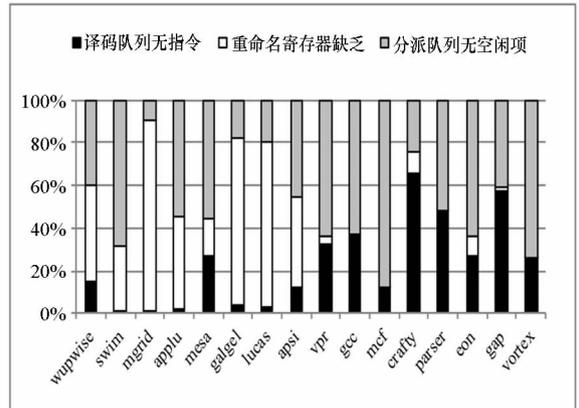


图 5 重命名阻塞的原因分布

Fig. 5 Reasons for renaming stalls

可以看出,对于浮点测试程序(图 5 中左边 8 个程序),重命名寄存器缺乏和分派队列没有空闲项是主要原因,译码队列无指令占的比例很小;而对于整数测试程序(图 5 中右边 8 个程序),重命名阻塞原因则有所不同,重命名寄存器相对并不是受限因素,而译码队列没有指令占的比例较大,分派队列没有空闲项同样也是主要原因之一。

译码队列没有指令是因为流水线前端指令供应不足。为此,统计了对取指有影响的一级指令 Cache 失效率、分支比例(可预测分支指令在所有指令中所占的比例)以及分支误预测率,如图 6、

图 7、图 8 所示。

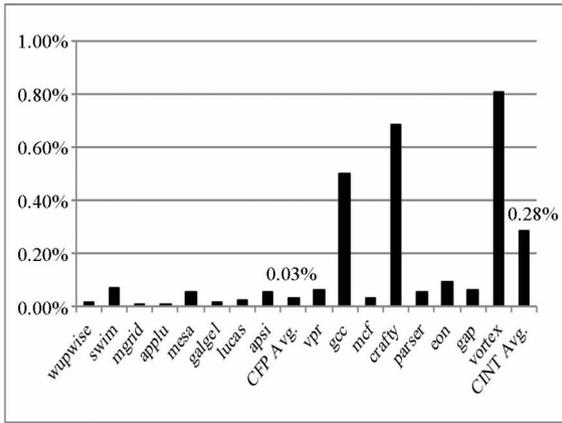


图 6 一级指令 Cache 失效率
Fig. 6 L1 ICache miss rate

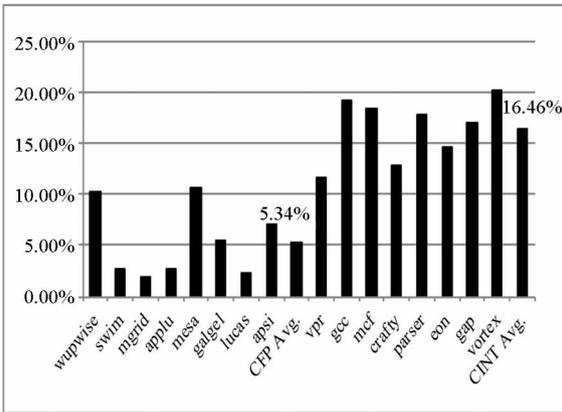


图 7 分支比例
Fig. 7 Proportion of predictable branches to all instructions

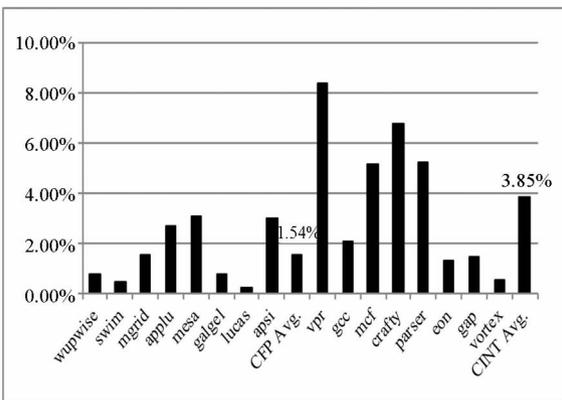


图 8 分支误预测率
Fig. 8 Branch misprediction rate

可以看出,相对于浮点测试程序,整数测试程序在这三个指标上都相对较大,尤其是分支比例和分支误预测率方面,整数测试程序要高出很多。改善分支预测结构(比如提高分支预测准确率,降低分支预测延迟)对整数测试程序性能的提高

会更有帮助。一级指令 Cache 失效率非常低,最大时也不到 1%,优化空间不大。

分派队列没有空闲项接收重命名的指令是因为分派发生了阻塞。从第一章描述可知,分派队列没有指令、发射队列没有空闲项、ROB 没有空闲项都会导致分派阻塞。为此,对这些事件进行了统计。图 9 给出了分派阻塞的原因分布。

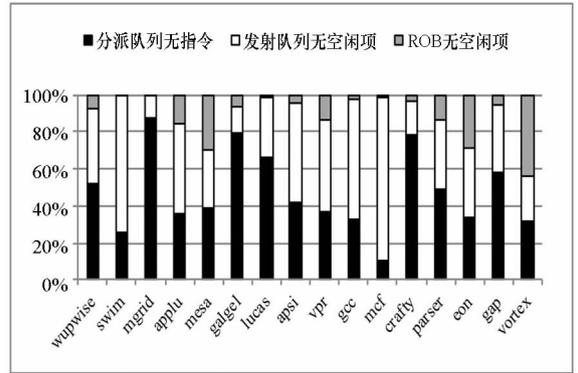


图 9 分派阻塞的原因分布
Fig. 9 Reasons for dispatching stalls

分派队列没有指令是因为重命名阻塞导致的,前面已经分析了原因。当有指令可以分派时,ROB 没有空闲项对个别程序比较明显,比如 vortex,但总的来说并不突出,而发射队列没有空闲项成了主要原因。在 XMC 处理器核的设计中,发射队列采用的是分布式结构,这样可以降低设计的复杂度,但是从利用率的角度考虑并不好。

通过以上分析,发现 SPEC CPU2000 测试程序在 XMC 处理器核上运行时的性能受限因素如下:

- 1)对于浮点测试程序,重命名寄存器个数不足导致重命名阻塞严重;
- 2)优化分支预测结构对整数测试程序的性能提升更有帮助;
- 3)无论整数还是浮点测试程序,发射队列项数不足导致分派阻塞严重。

4 XMC 处理器核的性能优化

依据上述分析结果,从以下几个方面优化 XMC 处理器核的微体系结构。

分支预测结构方面,使用更加先进的分支预测算法来提高预测精度。当前比较流行并被广泛采用的预测算法有 Tournament^[11, 14]和 TAGE^[15-16],本文选择了 TAGE 预测算法进行分支方向预测。首先在软件模拟器上对 TAGE 算法和 XMC 处理器核之前采用的分支预测算法进行了比较,TAGE 算法确实具有更高的预测精度,提

高了近 2%。TAGE 算法的 RTL 实现工作正在进行中。

针对重命名寄存器个数不足问题,修改 RTL 代码,并通过参数化使得重命名寄存器的个数可以在 64,96 和 128 之间进行选择,以从性能和实现代价方面综合评价。

图 10 给出了 128 个重命名寄存器下,SPEC CPU2000 测试程序相对于 64 个重命名寄存器时的性能提升百分比。

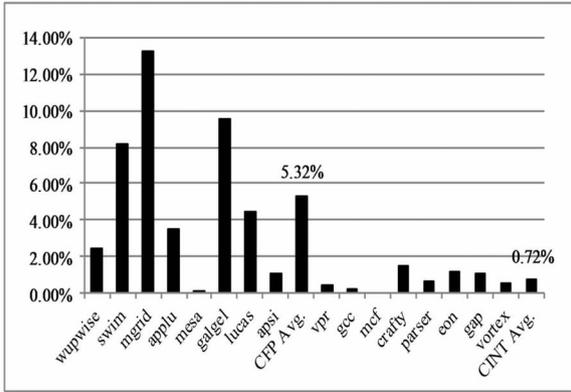


图 10 128 个重命名寄存器和 64 个重命名寄存器的性能比较

Fig. 10 Performance with 128 rename registers compared to that with 64 rename registers

如同第 3 节分析的一样,重命名寄存器个数不足导致浮点测试程序的重命名阻塞严重,增加重命名寄存器有效提高了浮点测试程序的性能,平均达到了 5.32%,最高达到了 13.21%。对于整数测试程序,性能提升并不显著,平均只有 0.72%。

此外,对 96 和 128 个重命名寄存器下 SPEC CPU2000 的性能进行了对比,图 11 给出了 96 个重命名寄存器下,SPEC CPU2000 测试程序相对于 128 个重命名寄存器时的性能降低百分比。对于整数测试程序,影响非常小,平均降低了 0.09%。对于浮点测试程序,重命名寄存器个数从 128 减少到 96,平均性能不但没有降低,反而增加了 0.33%,这主要是由于 apsi 程序的性能不降反增导致的^①。如果去除 apsi 的结果,浮点测试程序的性能平均降低了 1.02%,降低最多的程序是 wupwise,达到了 3.39%。

96 个重命名寄存器相比 128 个重命名寄存器,性能虽然会有所下降,但是实现代价也会相对变小。对 XMC 处理器核进行了综合,相对于 64 个重命名寄存器,96 个重命名寄存器时的 XMC 处理器核的面积增加了 4.70%,128 个重命名寄

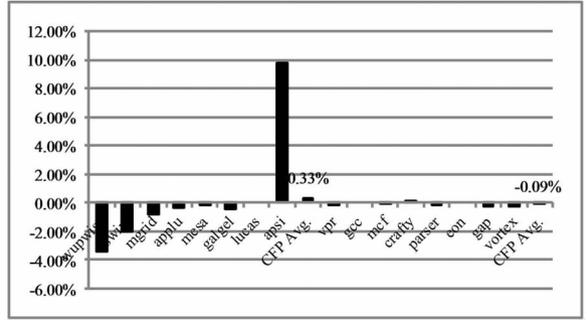


图 11 96 个重命名寄存器和 128 个重命名寄存器的性能比较

Fig. 11 Performance with 96 rename registers compared to that with 128 rename registers

存器时的 XMC 处理器核的面积增加了 7.84%。可见,实现 96 个重命名寄存器时,XMC 处理器核的性能代价比较高。

最后,针对发射队列项数不足问题,分别增加了整数发射队列和浮点发射队列的项数,但是发现单纯增加发射队列项数并不能带来明显的性能提升。这暴露了目前基于计数器分析微体系结构级性能受限因素方法的不足,即提取的事件相对比较独立,未充分将处理器核内各流水线级的事件有效关联起来,还有待进一步完善。

5 结论

处理器微体系结构日益复杂,性能分析成为处理器研制过程中的重要环节。本文提出了一种基于计数器的性能分析方法,该方法在已有的处理器核硬件实现代码外部添加一个专用的性能监测单元,用于收集微体系结构分析需要的各种事件,并把统计的事件送给结果分析器进行分析,获得微体系结构实现的性能受限因素。

以处理器核 XMC 为基础,面向 SPEC CPU2000 测试程序,基于 FPGA 原型系统仿真平台,以流水线中重命名和分派两处作为切入点,采用基于计数器的性能分析方法寻找重命名阻塞和分派阻塞的原因,可以发现:

- 1)对于浮点测试程序,重命名寄存器个数不足导致重命名阻塞严重;
- 2)优化分支预测结构对整数测试程序的性能提升更有帮助;
- 3)无论整数还是浮点测试程序,发射队列项

① 重命名寄存器个数增加是有可能导致性能不增反降。这是因为,不该执行的指令(异常或分支误预测被清除的指令)可能因为重命名寄存器个数的增加而有机会进入流水线,增加了清除代价。

数不足导致分派阻塞严重。

根据以上发现,采取了相应的优化措施。当把重命名寄存器个数从64个增加到128个时,浮点测试程序的性能平均提高了5.32%,最高达到了13.21%。但是增加发射队列项数并没有带来预期的性能改善,这暴露了目前的方法还存在不足,需要进一步完善。

参考文献 (References)

- [1] Moudgill M, Wellman J D, Moreno J H. Environment for PowerPC micro-architecture exploration [J]. IEEE Micro, 1999, 19(3): 15-25.
- [2] Singhal R, Venkatraman K S, Cohn E R, et al. Performance analysis and validation of the Intel Pentium 4 processor on 90nm technology[J]. Intel Technology Journal, 2004, 8(1): 39.
- [3] 张福新,章隆兵,胡伟武. 基于 SimpleScalar 的龙芯 CPU 模拟器 Sim-Godson [J]. 计算机学报, 2007, 30(1): 68-73.
ZHANG Fuxin, ZHANG Longbing, HU Weiwu. Sim-Godson; a godson processor simulator based on SimpleScalar [J]. Chinese Journal of Computers, 2007, 30(1): 68-73. (in Chinese)
- [4] Austin T, Larson E, Ernst D. SimpleScalar: an infrastructure for computer system modeling[J]. Computer, 2002, 35(2): 59-67.
- [5] Martin M M K, Sorin D J, Beckmann B M, et al. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset[J]. ACM SIGARCH Computer Architecture News, 2005, 33(4): 92-99.
- [6] Binkert N L, Dreslinski R G, Hsu L R, et al. The M5 simulator: modeling networked systems [J]. IEEE Micro, 2006, 26(4): 52-60.
- [7] Binkert N L, Beckmann B, Black G, et al. The GEM5 simulator[J]. ACM SIGARCH Computer Architecture News, 2011, 39(2): 1-7.
- [8] Wunderlich R E, Wenisch T F, Falsafi B, et al. SMARTS: accelerating micro-architecture simulation via rigorous statistical sampling [J]. Computer Architecture NEWS, 2003, 31(2): 84-95.
- [9] 朱艳玲,朱怡安,王云岚. 基于硬件性能计数器的编译器性能测试与分析[J]. 微电子学与计算机, 2008, 25(3): 192-196.
ZHU Yanling, ZHU Yian, WANG Yunlan. Compiler performance test and analysis based on hardware performance counters[J]. Microelectronics & Computer, 2008, 25(3): 192-196. (in Chinese)
- [10] 车永刚,王正华,李晓梅. 一个基于硬件计数器的程序性能测试与分析工具[J]. 计算机科学, 2004, 31(1): 170-174.
CHE Yonggang, WANG Zhenghua, LI Xiaomei. A hardware counter based tool for application's performance measurement and analysis[J]. Computer Science, 2004, 31(1): 170-174. (in Chinese)
- [11] Kalla R, Sinharoy B, Starke W J, et al. Power7: IBM's next-generation server processor[J]. IEEE Micro, 2010, 30(2): 7-15.
- [12] Sprunt B. Pentium 4 performance-monitoring features [J]. IEEE Micro, 2002, 22(4): 72-82.
- [13] Huang L B, Wang Y W, Dou Q, et al. Fast FPGA system for micro-architecture optimization on synthesizable modern processor design [C]//Proceedings of 25th International Conference on Field-programmable Logic and Applications (FPL), 2015.
- [14] Evers M, Chang P Y, Patt Y N. Using hybrid branch predictors to improve branch prediction accuracy in the presence of context switches [C]//Proceedings of the 23rd Annual International Symposium on Computer Architecture, 1996.
- [15] Seznec A. A new case for the tage branch predictor [C]//Proceedings of the 44th Annual IEEE/ACM International Symposium on Micro-architecture, 2011.
- [16] Seznec A. TAGE-SC-L branch predictors [C]//Proceedings of the 4th Championship Branch Prediction, 2014.