面向基于 x86 处理器和 AMBA 的系统芯片的全系统模拟器 PKUsim-86

庞九凤,佟 冬,李 皓,何 浪,程 旭
(北京大学微处理器研发中心,北京 100871)

摘 要: 基于周期级全系统模拟器对微体系结构进行系统性能评估成为芯片设计必不可少的环节.虽然 x86 处理器是当前商业和科学计算领域最广泛采用的处理器,很少有开源的 x86 模拟器能够满足研究需要.本文面向基于 Geode GX x86 处理器和 AMBA 总线的 PKUnity-86 系统芯片,设计并实现了周期级全系统模拟器 PKUsim-86.它可以启动 Microsoft DOS、Windows 98、Windows XP 等操作系统,运行典型的 x86 应用程序.PKUsim-86 支持功能模拟和性能模拟的在线切换,其指令模拟速度为 0.86MIPS,与真实硬件的对比表明,PKUsim-86 具有较高的相对准确度.

关键词: 全系统模拟; 性能评估; 系统芯片; x86 处理器
中图分类号: TP302.7 **文献标识码:** A **文章编号:** 0372-2112 (2011) 02-0351-07

PKUsim-86: A Cycle Level Full System Simulator for x86 Processor and AMBA-Based SoC

PANG Jiu-feng, TONG Dong, LI Hao, HE Lang, CHENG Xu
(*Microprocessor Research and Development Center, Peking University, Beijing* 100871, *China*)

Abstract: Cycle level full system simulator based performance evaluation has been indispensable to increasingly complex System-on-Chip designs. Although x86 processors have been the most ubiquitous processors in both commercial and science computing world, there is a scarcity of open source simulators that enable academic researchers to experiment with new x86 microprocessor based designs. This paper presents a cycle level full system simulator for PKUnity-86 System-on-Chip platform, which is based on Geode GX x86 processor and AMBA bus architecture. PKUsim-86 can boot up Microsoft DOS, Windows 98, Windows XP and run common x86 applications. On-the-fly switching between the functional simulation and performance simulation mode is supported. PKUsim-86 executes 0.86 million simulated instructions per second on average. Compared with real machines, the relative accuracy of PKUsim-86 is acceptable.

Key words: full system simulation; performance evaluation; system-on-chip; x86 processor

1 引言

通过对硬件建模,研究者可以用参数可控的形式来考察目标应用程序在被模拟系统上执行时的行为与特征.基于周期精确的性能模拟器对微体系结构进行性能分析和设计空间探索,已经逐渐取代早期数学形式化的性能评价方法[1],成为芯片设计流程的必要环节.现有的大多数模拟器是用户级模拟,比如被广泛应用于超标量处理器微体系结构研究的 SimpleScalar 模拟器[2].但是,用户级模拟忽略了中断、系统调用等操作系统代码的执行对系统性能的影响,对 SPEC CINT2000 的评测误差可达 100%[3].因此,全系统模拟将成为今后体系结构研究和设计的趋势[4].

x86 处理器是当前商业和科学计算领域使用最广泛的处理器,但目前很少有开源的面向 x86 处理器的模拟器可用于 x86 处理器的硬件设计和 x86 应用程序的研究.从国际的研究现状来看,现有的 x86 全系统模拟器要么为了保证模拟的详细而降低了实现的灵活性,要么为了灵活性而影响了模拟速度,大都存在修改困难、执行速度慢、对性能模拟的支持有限等问题.仅针对抽象的硬件而非真实的机器建模,无法对模拟器本身存在的误差进行分析和校正,难以获得准确可信的结论.

PKUnity-86[5] 是一款基于 Geode GX x86 处理器和 AMBA 开放总线架构的系统芯片(System-on-Chip, SoC),

收稿日期:2009-12-03;修回日期:2010-07-26
基金项目:国际科技合作基金(No.2008DFB10010)

支持运行 Microsoft DOS、Windows 98 等操作系统及 x86 应用程序.其开放的总线架构使得快速集成各种标准 IP 核成为可能,充分利用了传统嵌入式领域的设计资源优势,其 x86 处理器支持运行 Microsoft Windows 操作系统及其丰富的 x86 应用软件,充分利用了传统个人计算机的软件优势.

本文设计了面向 PKUnity-86 系统芯片的周期级全系统模拟器 PKUsim-86.PKUsim-86 由执行部分、功能模拟和性能模拟等构成,具有以下特征:(1)执行部分与模拟部分分离,具有良好的可重定向性;(2)功能模拟和性能模拟分离,支持在线实时切换,以节省模拟所需的时间;(3)提供灵活的体系结构配置机制;(4)平均每秒模拟执行指令数达到 0.86 百万条,通过与真实 Geode GX x86 处理器的校对,保证了模拟器本身的准确度.

2 背景介绍

2.1 PKUnity-86 系统芯片

PKUnity-86 是基于 Geode GX x86 处理器和 AMBA 开放总线架构的 SoC[6],其核心功能部件如图 1 所示.

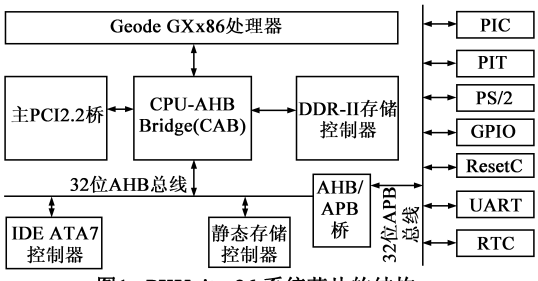

图1 PKUnity-86 系统芯片的结构

Geode GX 处理器是 AMD 公司转让给中国的一款 32 位按序单发射八级流水结构的低功耗 x86 处理器.CAB(CPU-AHB Bridge)负责 x86 处理器的总线接口协议到 AHB 总线协议的转换,以及处理器、存储控制器、主 PCI2.2 桥之间的通讯.位于 32 位 AHB 总线上的高速系统设备硬盘控制器、静态存储控制器.位于 32 位 APB 总线上的低速外围设备包括串行接口、通用输入输出接口、PS/2 接口、复位控制以及传统设备,即可编程中断控制器,可编程中断定时器和实时时钟.AHB 和 APB 总线之间通过 AHB/APB 桥进行通讯.

2.2 相关工作

Pin[7]是 Intel 公司开发的动态插桩系统(Dynamic Instrumentation System),将 C/C++代码插入正在运行的二进制程序的某些位置,用于探视程序的代码覆盖和执行踪迹、转移预测、Cache 失效率、缺陷容忍、内存泄漏等情况,在分析复杂应用程序行为的研究中有着广泛的应用.动态插桩干扰原始程序的运行,运行速度会降低 1/10[8],适用于设计后的硬件机器.

Bochs[9]模拟了完整的基于 Intel 80x86 处理器的个人计算机系统,可以运行 Microsoft DOS、Windows 95/98/NT/2000/XP/2003、GNU/Linux 等操作系统,配置成 386、486、Pentium、Pentium Pro、Pentium II、Pentium III、Pentium4 和 x86-64 等处理器.Bochs 被广泛应用于调试新的操作系统、开发新的设备驱动程序、作为新的 x86 兼容硬件的参考模型.但是,Bochs 是指令级的全系统模拟器,不支持性能评测.

PTLsim[10]模拟 x86 和 x86-64 指令系统体系结构(Instruction Set Architecture,ISA)处理器,对高速缓存、存储系统和输入输出设备都进行了性能建模.它通过集成 Xen 虚拟机,直接在宿主机上运行指令,以实现全系统模拟的支持.PTLsim 是目前唯一开放源代码的 x86 架构周期级全系统模拟器.但是,PTLsim 要求宿主机必须是 64 位,不支持 Microsoft Windows 操作系统及 x86 应用程序,这些局限性使得它不适用于 PKUnity-86 系统.

LLVM(Low Level Virtual Machine)[11]是美国伊利诺斯大学开发的开放源代码编译器架构.它包括 LLVM 虚拟指令集,用于分析、优化、代码生成等工作的集成库,以及建立在以上集成库基础之上的汇编器、链接器和调试器等.PKUsim-86 采用 GCC 编译器,通过解释执行每一条 x86 指令,解释执行有很高的兼容性,但速度较慢,基于 LLVM 的动态编译器具有很低的兼容性,但是速度却较快.这需要在兼容性和模拟速度之间取得合理的权衡.

3 PKUsim-86 的设计与实现

PKUsim-86 的设计目标是修改配置灵活、执行速度快、可重定向.它采用了两种设计原则:执行部分与模拟部分的分离,功能模拟与性能模拟的分离.为了减少非必要的工作量,采用将 SESC[12]的 x86 性能模型融入 Bochs 全系统模拟环境中的设计思路.

3.1 总体框架与设计原则

PKUsim-86 由执行部分、功能模拟和性能模拟等三部分组成,如图 2 所示.

执行部分用于控制模拟器的启动、配置和运行,以及与用户的交互.功能模拟由 Bochs 和 RPT(RePlay Transmogrifier)[13]完成.性能模拟的核心是 SESC.

(1)执行部分与模拟部分的分离

这是 PKUsim-86 的首要设计原则.执行部分与模拟部分的关系为松耦合,通过定义交互接口,将执行部分和模拟部分有机地结合起来,使设计者几乎可以完全复用执行部分,只关心目标硬件系统的模拟,易于支持不同目标硬件系统的模拟,从而实现了可重定向性的支持.除 x86 处理器外,PKUsim-86 已经成功移植到 UniCore32 处理器和 ARM 处理器上.以基于 UniCore32 处理器的 Unity-863 系统为例说明.通过在模拟部分实现 U-

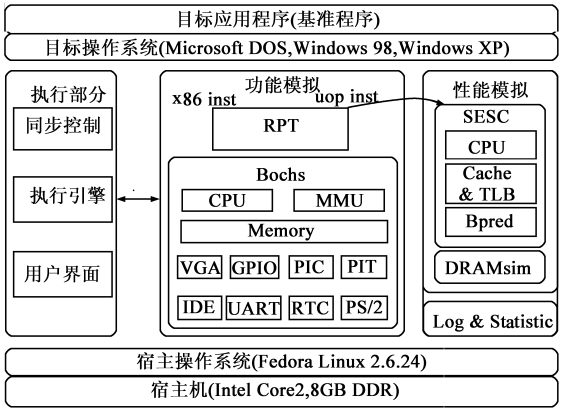

图2　PKUsim-86模拟器的结构

niCore32 处理器、存储系统和输入输出系统的模型,将 Bochs 替换为 UniCore32 的指令级功能模拟器 sim-UniCore32,并模拟 MMU 的虚实地址转换行为和实地址空间,由于 Unity-863 系统的外设与 PC 基本相同,复用标准的外设模型,并实现 Unity-863 系统所特有的时钟控制器与中断控制器.

(2)功能模拟与性能模拟的分离

功能模拟与性能模拟的分离使得在保持一部分不变的情况下另一部分具有更大的灵活性.例如在保证功能模拟不变的情况下,可以支持不同抽象层次的性能模拟,包括交易级、周期级和寄存器传输级等.PKUsim-86 中性能模拟的建模粒度是周期级精确.表 1 是功能模拟与性能模拟的特征.

表 1　功能模拟与性能模拟的特征

项目	功能模拟	性能模拟
输入	二进制程序代码	微指令序列
输出	微指令序列	性能分析结果
模拟任务	启动操作系统,运行应用程序	执行微指令序列,产生各个部件以及总的详细统计信息
抽象层次	指令级	交易级,周期级,寄存器传输级

该设计原则为功能模拟与性能模拟的在线实时切换提供了支持.性能模拟执行应用程序往往耗费大量的时间,在应用程序进入核心循环或研究人员感兴趣的区域之前,使用功能模拟模式,然后通过用户界面的按钮切换到详细的性能模拟模式,统计性能数据.PKUsim-86 为研究者提供了不同模拟粒度的在线切换机制,功能模拟的指令数由研究者决定.在线模拟粒度切换加快系统的启动,从而将有限的时间花费在性能数据的收集.

3.2　功能模拟

处理器的功能模型是计算机模拟的核心,它模拟了指令级的 x86 处理器.处理器模型的抽象类接口可以适用于不同的指令系统体系结构,重定向是指从抽象类接口继承并实现目标硬件系统的处理器.

图 3 是处理器模型的类层次.cpu_stub_c 是抽象类接口,cpu_c 类继承 cpu_stub_c 类并模拟具体处理器,此处模拟的是 x86 处理器,其内嵌成员 pluginMemsys 指向存储系统的抽象接口,cpu_c 类具有一个通用的译码方法 md_init_decoder,负责对 x86 指令进行译码.

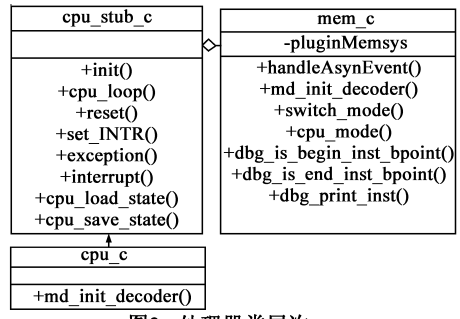

图3　处理器类层次

cpu_loop 方法模拟了 x86 处理器的取指、译码和执行过程,按照图 4 中的步骤进行无限循环直至退出模拟,具体过程如下:

(1)如果发生外部中断,调用 handleAsyncEvent 方法处理外部中断;

(2)取出一条 x86 指令;

(3)对该条 x86 指令进行译码并执行;

(4)更新时钟滴答.系统内部时间的计算是以单条指令的模拟为基本单位的.各个模拟部件均注册自己的时间调度周期和回调函数,在 cpu_c 类的每个执行步后,通过 cpu_loop 方法调整事件调度队列中每一项的剩余时钟滴答数,在剩余滴答数减至 0 时,调用该项注册事件的回调函数.被模拟部件发送中断、检测是否有用户输入等都是通过这种机制来实现的.

存储模型是由物理内存和内存管理单元(Memory Management Unit,MMU)组成的.MMU 是现代计算机系统中的重要组成部分,负责将虚拟地址转换为物理地址.MMU 直接访问内存的页表,当遇到权限受限或 Page Fault 时,产生异常并将控制转给处理器.另外,MMU 还负责维护页表的访问位和脏位,以及分发对内存访问

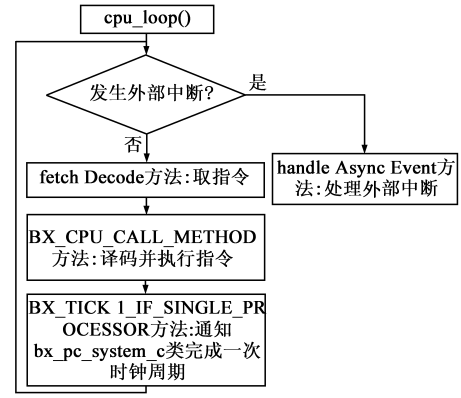

图4　功能模拟的过程

和输入输出设备(Input/Output,I/O)的访问.由于是功能模拟,所以没有实现 TLB 和 Cache,直接访问物理内存来读取和写入数据.

PKUsim-86 复用了 Bochs 中已有的传统个人计算机系统中的输入输出设备模型,包括可编程中断控制器、可编程中断定时器、实时时钟、PS/2 接口的键盘和鼠标、主 PCI 桥、显卡、IDE 控制器等.

RPT 是一个通用的指令译码器,它将特定指令系统体系结构的指令转换为通用的微指令序列.这些微指令序列被性能模型用于详细的模拟.RPT 是微体系结构性能模拟器与 ISA 功能参考模型或 FPGA 仿真器之间的耦合部件.在 PKUsim-86 中,RPT 将功能模拟的每条 x86 指令转换成一组微指令序列,该组微指令序列以指令对象(instruction object)的形式提交给性能模型.

以 x86 ISA 中的指令 POP %eax 为例,如图 5 所示,栈中的 POP 指令被转化为两条微指令,第一条微指令 uLD 将栈顶的数据写入 eax 寄存器中,第二条微指令 uSUB 将栈顶指针 esp 寄存器的值减 4.RPT 将由 uLD 和 uSUB 组成的一组微指令序列发送给性能模型,由其进行详细的性能模拟评测.

$$\text{POP \%eax} \Rightarrow \begin{matrix}\text{uLD \%eax,[\%esp]}\\ \text{uSUB \%esp,4}\end{matrix}$$

图 5 x86 指令到微指令序列的转换示例

3.3 性能模拟

处理器的性能模型通过修改 SESC 而来.原始的 SESC 是一个面向 MIPS 指令系统体系结构的性能模拟器,采用 MINT(MIPS 解释器)作为其功能模拟部分.在本文中,我们用 Bochs 替换 MINT,从而将 SESC 移植到 x86 指令系统体系结构.SESC 对流水线、分支预测、缓存等进行了详细建模.RPT 将指令对象发给 SESC 进行性能模拟,以计算执行所需要的时间.指令对象包含了性能模拟所需的所有信息:指令地址、Load/Store 指令的存储地址、源寄存器、目标寄存器等.

图 6 是 SESC 中部分关键类的相互关系.ExecutionFlow 类控制了 SESC 的整个模拟过程,它通过 executePC()和 exeInst()方法调用仿真模块执行指令,并产生相应的动态指令类,交由性能模型进行模拟.GProcessor 类模拟了处理器核,控制整个流水线的运行,其取指令功能又通过 FetchEngine 类实现,FetchEngine 类接收来自 RPT 的微指令序列.Resource 类是处理器核中各功能单元的基类,在其之上派生出了定点处理单元、浮点处理单元、分支预测单元、访存处理单元、Cache 和 TLB 等类.当 GProcessor 类调用 issue()方法时,动态指令对象将传递给相应的功能单元进行处理.DepWindow 类负责检查流水线中正在处理的动态指令对象之间的相关性,决定各条指令是否可发射或进入下一个流水级.Cluster 类则用来协调多处理器的模拟时各功能单元直接的关系.

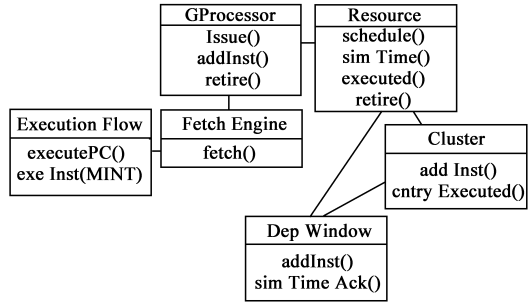

图6 SESC中部分类之间的关系[12]

总线的性能模型对 AMBA 总线进行了周期级建模,如表 2 所示,其总线宽度、频率等特性是在总线初始化时指定的.当 Cache 或 TLB 失效时,处理器的性能模型通过 access 方法发起总线请求,并指定总线请求的类型,例如读写类型、地址和控制信息等.总线的性能模型在每个周期都调用 bus_loop 方法,该方法进一步调用 arbitrate 和 decode 方法分别对总线请求进行仲裁和译码,并选择相应的从设备进行响应.从设备(此处为内存控制器)通过 pluginSlaves 方法注册到总线上,接收并处理总线上的请求.

受到预充电、内存刷新、地址译码等影响,内存访问延迟差别很大,而 SESC 中原有的内存性能模型指定了固定的内存延迟,难以准确地体现内存对于计算机系统的性能影响.因此,PKUsim-86 将 DRAMsim[14] 集成到性能模型中.DRAMsim 将处理器发到总线上的访存请求转换成一系列的 DRAM 命令序列,并将物理地址进一步地转换成内存的通道 ID,体 ID,行 ID 和列 ID 等,用于控制读和写内存数据.由于 PKUnity-86 SoC 采用 DDR2 存储控制器,PKUsim-86 中的 DRAMsim 也被配置成 DDR2,以尽可能接近真实的硬件设计.

表 2 总线的性能模型的特征

类型	描述
init	总线初始化,通过参数确定总线的宽度,频率等
access	处理器访问总线的调用方法,通过参数确定请求
arbitrate	总线仲裁方法,提供总线的仲裁机制
decoder	总线地址译码方法,提供地址译码机制
bus_loop	总线调用方法,处理每个周期总线的行为
pluginSlaves	从设备注册方法

输入输出设备的行为受外界事件的影响,很难进行精确的性能建模.一般来说,程序中平均只有 1% 的指令是 I/O 指令,大多数设备的 I/O 访问很少发生.所以,本文主要关注 PKUnity-86 SoC 对系统性能有关键影响的部件,包括 x86 处理器、总线和存储控制器.

3.4 其他特性

在真实的处理器中,当转移指令预测错误时,错误分支上的指令将进入流水线,直到转移指令的转移方

向确定为止.PKUsim-86 时序模拟的指令来自于功能模拟,且均被正确执行,为了模拟错误分支上的指令对微体系结构状态的影响,PKUsim-86 在流水线中插入若干无关指令,数目由平均转移预测代价决定.

模拟器的调试过程将会占据相当大的工作量,必须从设计之初就考虑提供方便的程序调试接口.PKUsim-86 提供了两种调试手段:GDB 调试插桩和检查点(checkpoint).PKUsim-86 支持利用 GDB 调试插桩交叉调试被模拟系统,可以对被模拟系统进行源代码级调试.在全系统模拟器中,错误往往发生在程序执行很长时间以后,通过加入检查点的支持,可以在很短时间内恢复到错误现场,从而提高调试的效率.

PKUsim-86 的另一大特点是通过 Perl 脚本来控制模拟器的运行并进行性能数据的统计,以及配置文件的修改.由于脚本的解释独立于系统的模拟,脚本的修改无须重新编译模拟器源代码,不会对模拟器的正常运行造成影响.

4　实验

本文实验的宿主机采用四个 2.83 GHz Intel Core2 CPU,内存 8GB DDR2,硬盘 1TB;软件是 Fedora Linux,内核版本 2.6.24.实验包括模拟器的准确度评测,执行速度评测,以及对设计的指导.

4.1　准确度评测

PKUnity Hilon 开发板上集成了 Geode GX x86 处理器芯片.通过在 PKUsim-86 模拟器和 PKUnity Hilon 开发板的对比实验,评估 x86 处理器内部的二级 TLB(L2TLB)和转移目标缓冲器(BTB)、指令 Cache(ICache)、数据 Cache(DCache)等功能单元对系统性能的影响.本实验采用的基准程序是 SPEC CINT2000.

如图 7 所示,在 PKUsim-86 模拟器和 PKUnity Hilon 开发板上的实验表明,x86 处理器内部的功能单元对系统性能的相对影响很接近,例如,关闭 L2TLB,PKUsim-86 的性能下降为原始被模拟系统的 0.950,PKUnity Hilon 开发板的性能下降为原始真实硬件机器的 0.906.相对于绝对准确度(absolute accuracy)而言,模拟器的相对准确度(relative accuracy)对系统的评测更为关键,研究人员可以基于模拟器对各种不同的设计方案进行对比分析,从而选择较优的设计方案.

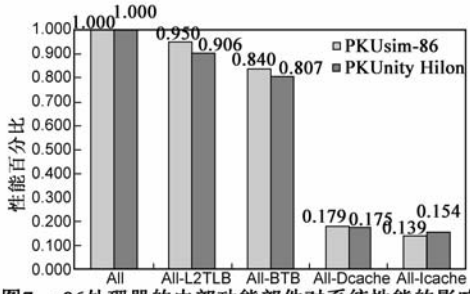

图7　x86处理器的内部功能部件对系统性能的影响

4.2　模拟速度评测

文献[15]指出,典型的指令级功能模拟器每秒平均模拟执行 6 ~ 7 百万条指令(Million Instruction Per Second,MIPS),在真实机器上,平均每秒执行 1000 ~ 2000 百万条指令,功能模拟比本地执行要慢 2 到 3 个数量级.对于周期精确的性能模拟而言,模拟速度比功能模拟慢 1 到 2 个数量级,平均每秒最多模拟执行 0.3 百万条指令,比本地执行要慢 10,000 倍(160h vs. 1min).

表 3 列出了 PKUsim-86 对不同程序的模拟速度.在 PKUsim-86 上启动 Microsoft Windows XP,运行 SPEC CINT2000 基准程序子集,以及 ATTO Disk,Benchmark1.3f,Crystal Disk Mark,Crystal Mark 和 Performance Test 等评测程序.PKUsim-86 每秒模拟执行的周期数最多是 4.465 百万个周期,最少是 2.571 百万个周期,平均每秒模拟执行 3.569 百万个周期.在性能模拟模式下,PKUsim-86 的模拟速度最高可达 1.136 MIPS,最低为 0.583 MIPS,平均模拟速度为 0.86MIPS.

表 3　PKUsim-86 的模拟速度

基准程序	周期数(K)	指令数(K)	执行时间(h)	MCPS(每秒百万个周期)	MIPS(每秒百万条指令)
ATTO Disk	150977679	34205805	11	3.812	0.864
Benchmark1.3f	14945745	3321288	1	4.152	0.923
Crystal DiskMark22	55531300	14701023	6	2.571	0.583
Performance Test	11338689	2975022	1	3.150	0.879
CrystalMark CM09GDI	48225169	12161278	4	3.349	0.845
CrystalMark CM09OGL	755493677	157312671	47	4.465	0.930
CrystalMark CM09D2D	224388119	47476936	15	4.155	1.136
164.gzip	13139458	3197290	1	3.650	0.889
181.mcf	9678855	2431183	1	2.689	0.675
256.bzip2	332448510	7899409	25	3.694	0.878
平均值	/	/	/	3.569	0.860

PKUsim-86 的性能模型采用事件驱动模拟,只在当使某个模块状态改变的事件发生时才去访问该模块,相对于周期驱动模拟而言,节省了计算资源,加快了模拟速度,其模拟速度是一般性能模拟器的 2.9 倍,远远高于 GEMS 模拟器[16],GEMS 模拟器的指令模拟速度仅为 0.12 MIPS.

4.3　对设计的指导

基于 PKUsim-86 模拟器,我们评估了不同的总线配置对性能的影响,如表 4 所示.

表 4 不同的总线配置对性能的影响

配置	总线宽度(位)	读操作返回首数据的延迟(CPU 时钟周期数)	读操作返回后续数据的延迟(CPU 时钟周期数)	平均性能(Instruction Per Cycle,IPC)
1	32	40	4	0.236
2	32	30	3	0.279
3	64	40	4	0.287
4	64	30	3	0.313

处理器频率为 533MHz,DDR2 存储控制器频率 667MHz,总线频率为 266MHz.32 位总线只能同时传输 1 个字(word),当 Cache 失效时,传输第一个字需要 m 个 CPU 周期,传输后续的每个字需要 n 个周期,则完成行填充需要($m + n * (\$LS-1)$)个周期,其中$\$LS$ 是 Cache 的块大小,而若采用 64 位总线,行填充仅需要($m + n * (\$LS-1)/2$)个周期.提高总线带宽(包括增加总线宽度和提高吞吐率)将有效地提高性能,配置 4 比配置 1 的性能提高了 32.6%.相对于提高总线吞吐率而言,增加总线宽度更能带来性能的改善,配置 2 比配置 1 的性能提高 18.2%,而配置 3 比配置 1 的性能提高 21.6%.

通过 PKUsim-86 模拟器得到若干关键微体系结构的统计数据,每 2 百万个周期进行一次采样,图 8 和图 9 显示了程序在运行的不同采样点时分支预测错误率,一级数据 Cache 失效率,以及处理器性能的变化,较高的数据 Cache 失效率引起处理器性能的急剧下降,如第 133 个采样点.正在进行的一项研究是分析 x86 应用程序中引发一级数据 Cache 失效次数最多的重要装载(important load)和关键分支转移(critical branch),进而优化 load/Store 队列和分支转移预测器的设计,类似的研究参见文献[17,18].

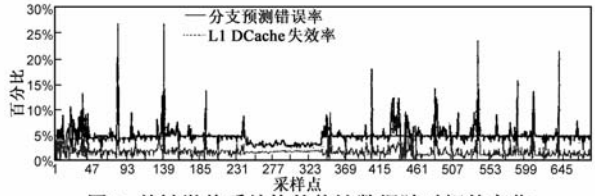

图8 关键微体系结构的统计数据随时间的变化

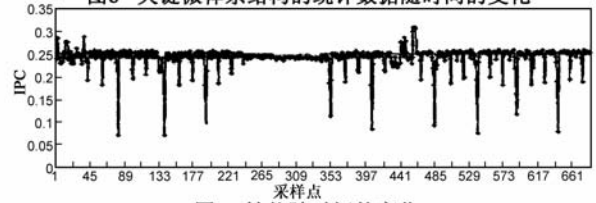

图9 性能随时间的变化

5 总结和未来工作

本文设计并实现了基于 Geode GX x86 处理器和 AMBA 总线的 PKUnity-86 系统芯片的周期级全系统模拟器 PKUsim-86,启动 Microsoft DOS,Windows98、Windows XP 等操作系统和运行典型的 x86 应用程序.PKUsim-86 采用执行部分与模拟部分分离的设计原则,具有较好的可重定向性;支持功能模拟和性能模拟的在线切换,其指令模拟速度可达 0.86MIPS,是一般性能模拟器的 2.9 倍.与真实硬件的对比实验表明,PKUsim-86 具有较高的相对准确度.直接执行是在宿主机上执行每条 x86 指令,其模拟性能还有进一步提升的可能,但直接执行要求目标机器与宿主机具有相同的 ISA,这需要在模拟速度与灵活性之间做到合理的权衡.PKUsim-86 对多处理器模拟的支持也是正在进行一个重要研究工作.

参考文献:

[1] 林闯,李雅娟,王忠民.性能评价形式化方法的现状和发展[J].电子学报,2002,30(12A):1917 - 1922.
Lin Chuang,Li Ya-Juan,Wang Zhong-min.Status and development of formal methods for performance evaluation[J].Acta Electronica Sinica,2002,30(12A):1917 - 1922.(in Chinese)

[2] 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 Fu-xin,Zhang Long-bing,Hu Wei-Wu.Sim-Godson:a Godson processor simulator based on SimpleScalar[J].Chinese Journal of Computers,2007,30(1):68 - 73.(in Chinese)

[3] Joshua J Y,Lilja D J.Simulation of computer architectures:simulators,benchmarks,methodologies and recommendations[J].IEEE Transaction on Computers,2006,55(3):268 - 280.

[4] 赵霞,郭耀,雷志勇,陈向群.基于模拟器的嵌入式操作系统能耗估算与分析[J].电子学报,2008,36(2):209 - 215.
Zhan Xia,Guo Yao,Lei Zhi-yong,Chen Xiang-qun.Estimation and analysis of embedded operating system energy consumption[J].Acta Electronica Sinica,2008,36(2):209 - 215.(in Chinese)

[5] Huang Kan,Lu Jun-lin,Pang Jiu-feng,Zheng Yan-song,Li Hao,Tong Dong,Cheng Xu.FPGA prototyping of an AMBA-based SoC compatible with Microsoft Windows[A].Proceedings of International Symposium on Field Programmable Gate Arrays[C].New York,NY,USA:ACM,2010.13 - 22.

[6] 郑衍松,佟冬,李皓,庞九凤,程旭.MS Windows 兼容的系统芯片硬件核心的分析与实践[J].北京大学学报(自然科学版),2009,45(6):973 - 978.
Zheng Yan-song,Tong Dong,Li Hao,Pang Jiu-feng,Cheng Xu.Analysis and practice of a SoC hardware kernel for MS Windows[J].Acta Scientiarum Naturalium Universitatis Pekinensis,2008,45(6):973 - 978.(in Chinese)

[7] Chi-Keung Lul,et al.Pin:building customized program analysis tools with dynamic instrumentation[A].Proceedings of ACM Conference on Programming Language Design and Implementation[C].New York,NY,USA:ACM,2005.190 - 200.

[8] Lizy Kurian John.Performance Evaluation And Benchmarking[M].Boca Raton,FL,USA:CRC Press,2006.5 - 25.

[9] Lawton K, et al. The Bochs IA-32 emulator project[OL]. http://bochs.sourceforge.net,2008-06-08/2009-12-03.

[10] Matt T Yourst. PTLsim: a cycle accurate full system x86-64 microarchitectural simulator[A]. Proceedings of IEEE International Symposium on Performance Analysis of Systems & Software[C]. San Jose, CA, USA: IEEE, 2007. 23 – 34.

[11] Chris Arthur Lattner. LLVM: An Infrastructure for Multi-stage Optimization[D]. Urbana, Illinois: Universtiy of Illinois at Urbana-Champaign, 2002.

[12] Ortego P M, Sack P. SESC: SuperESCalar simulator[OL]. http://sourceforge.net/projects/sesc/. 2004 – 12 – 10.

[13] Sanjay Patel. ACS Simulator Tools: Replay Transmogrifier [OL]. http://www.crhc.illinois.edu/ACS/tools/, 2003/2009.

[14] David Wang, Brinda Ganesh, et al. DRAMsim: a memory system simulator[J]. SIGARCH Computer Architecture News, 2005, 33(4): 100 – 107.

[15] Ayose F, Paolo F, et al. Combining simulation and virtualization through dynamic sampling[A]. Proceedings of IEEE International Symposium on Performance Analysis of Systems & Software[C]. San Jose, CA, USA: IEEE, 2007. 72 – 83.

[16] Milo M K Marin, et al. Multifacet's general execution driven multiprocessor simulator (GEMS) toolset[J]. ACM SIGARCH Computer Architecture News, 2005, 33(4): 92 – 99.

[17] 谢学军,叶以正,邱善勤,喻明艳.基于马尔可夫模型的数据值预取方案[J].电子学报,2007,35(2):307 – 310.
Xie Xue-jun, Ye Yi-zheng, Qiu Shan-qin, Yu Ming-yan. Data value prefetching method based on markov model[J]. Acta Electronica Sinica, 2007, 35(2): 307 – 310. (in Chinese)

[18] 朱德新,程旭,慎辉. UNICORE 体系结构中动态转移预测机制的研究与设计[J].电子学报,2004,32(8):1351 – 1355.
Zhu De-xin, Cheng Xu, Shen Hui. Dynamic branch prediction research and design for UNICORE architecture[J]. Acta Electronica Sinica, 2004, 32(8): 1351 – 1355. (in Chinese)

作者简介:

庞九凤　女,1985 年 1 月出生于河南濮阳.博士研究生,主要研究领域为软硬件协同设计、计算机模拟和性能评估、系统芯片设计.
E-mail: pangjiufeng@mprc.pku.edu.cn

佟　冬　男,1971 年 10 月生于黑龙江.副教授,主要研究领域为存储系统、片上通信、系统芯片和微处理器系统结构设计、软硬件协同设计.