

一种面向 Microsoft Windows 的 AMBA 设备 PCI 虚拟化机制

庞九凤, 陆俊林, 李 皓, 佟 冬, 程 旭

(北京大学微处理器研发中心, 北京 100871)

摘 要: PKUnity-86 是基于 x86 处理器和 AMBA 开放总线架构的系统芯片, 既能快速集成各种标准 IP 核, 又支持 Microsoft Windows 和 x86 应用程序. 如何使 AMBA 设备支持工业标准的 PCI 设备枚举和访问, 是支持 Microsoft Windows 的首要问题. 本文提出一种硬件 VPCI 机制, 通过为 AMBA 设备实现虚拟的 PCI 配置空间, 对配置、存储和 I/O 交易进行转发, Microsoft Windows 就能无差别地识别位于 PCI 总线的标准 PCI 设备和位于 AMBA 总线上的 AMBA 设备, 复用标准 PCI 设备驱动程序以访问相应的 AMBA 设备. 在 Bochs 模拟器上, 基于该 PCI 虚拟化机制的 PKUnity-86 系统芯片成功启动 Microsoft Windows 98 和 Windows XP 和运行典型的 x86 应用程序. VPCI 的硬件开销仅为标准 PCI 的 13.7%, 性能优于标准 PCI; 读和写 PCI 配置空间的性能分别是 AMD VSA 的 7.31 和 6.73 倍.

关键词: 系统芯片; 微软视窗操作系统; 设备虚拟化; 配置空间

中图分类号: TP303 **文献标识码:** A **文章编号:** 0372-2112 (2011) 05-1013-07

A PCI Virtualization Mechanism for AMBA Devices in Microsoft Windows

PANG Jiu-feng, LU Jun-lin, LI Hao, TONG Dong, CHENG Xu

(Microprocessor Research and Development Center, Peking University, Beijing 100871, China)

Abstract: PKUnity-86 is a system-on-chip that integrates AMBA system bus architecture with a x86 processor. A tremendous amount of standard intellectual properties from embedded world can be leveraged, as well as Microsoft Windows and x86 applications. How to identify and access AMBA devices is principal for Microsoft Windows compatibility. The paper implements a centralized PCI virtualization mechanism called VPCI, which routes configuration memory and I/O transactions to make Microsoft Window identify PCI devices on PCI bus and AMBA devices on AMBA. Standard Windows PCI device drivers can be reused directly. In Bochs simulator PKUnity-86 system-on-chip that adapts this VPCI mechanism boots up Microsoft Windows 98 and Window XP, and run typical x86 applications. The VPCI achieves better performance than standard PCI with only 13.7% hardware cost. The performance of read and write configuration space is 7.31 and 6.73 times that of AMD VSA.

Key words: system-on-chip; Microsoft Windows; device virtualization; configuration space

1 引言

移动嵌入式设备的市场需求日益增加, 如智能手机、超便携个人计算机、移动互联网设备等, 它要求其芯片具备低功耗、低成本、小体积等特点, 这促使采用系统芯片(System-on-Chip, 简称 SoC)设计, 例如 NVIDIA 公司基于 ARM-11 的 Tegra 和 Intel 公司基于 Pentium M 的 EP80579 芯片(代号: Tolapai). 越来越多的外围设备将被集成到芯片内部的高速总线上, SoC 设计已成为嵌入式领域的趋势^[1,2].

嵌入式系统芯片分为两大阵营: ARM 和 x86. 基于

ARM 处理器的嵌入式系统芯片在智能手机领域占据了高达 99% 的市场份额^[3], 该类芯片一般采用开放的片上总线协议, 以及基于平台和 IP 核复用的设计方法^[4]. 影响较大的片上总线包括 IBM CoreConnect^[5]、ARM AMBA^[6]、Silicore Wishbone^[7]. 该类芯片可以快速集成各种标准 IP 核, 减少芯片的上市时间和设计难度, 但是不同芯片厂商的不同实现方式导致在第三方软件存在兼容性问题, 例如 Microsoft Windows 操作系统及其 x86 应用程序, 符合 PCI 总线标准的 WDM (Windows Driver Model) 类型的设备驱动程序. 随着硬件性能的提升, 桌面系统和嵌入式系统的界限越来越模糊, 在嵌入式系统上运行

桌面操作系统已成为可能.基于 x86 处理器的 SoC 以其平台无关性、支持 Microsoft Windows 操作系统和丰富的 x86 应用软件等优势开始进入嵌入式领域,这大大地减少了应用软件和设备驱动程序的开发工作,但这类单芯片解决方案仍沿袭了传统的基于芯片组的南北桥 Wintel 架构,技术细节是少数公司所特有的,无法充分利用嵌入式领域中现有的丰富的标准 IP 核这一优势.

PKUnity-86^[8,9]是一款基于 x86 处理器和 AMBA 开放总线架构的系统芯片,支持运行 Microsoft DOS、Windows 98、Windows XP 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软件优势.本文的研究内容正是基于这一工作而开展.

操作系统的-一个重要工作是提供对设备资源的管理^[10].作为 Microsoft Windows 的组件之一,PnP 管理器 (Plug-and-Play Manager)负责识别计算机系统-中的设备,收集设备的资源需求并分配系统资源,根据设备标识加载相应的驱动程序并启动设备的功能.这一过程是通过 PCI 总线的配置交易完成的.在 PKUnity-86 SoC,实现系统功能的设备位于芯片内部的 AMBA 总线上,这些设备没有实现 PCI 总线控制器和 PCI 配置空间,而且不同于 PCI 总线的分布地址译码机制,AMBA 总线采用集中地址译码,AMBA 设备如何被 Microsoft Windows 所识别、初始化和访问是系统设计中亟待解决的首要问题.

本文提出了一种面向 Microsoft Windows 的 AMBA 设备 PCI 虚拟化机制.VPCI 为每个 AMBA 设备实现一个虚拟 PCI 配置空间.所有访问标准 PCI 设备的配置交易被转发到 Host/PCI 桥,由其负责访问标准 PCI 设备内部的标准 PCI 配置空间,所有访问 AMBA 设备的配置交易被转发到 VPCI,由其负责访问该 AMBA 设备对应的虚拟 PCI 配置空间,这使得不必对集成的设备 IP 核做任何修改,Microsoft Windows 就能无差别地识别、初始化和访问位于 PCI 总线的标准 PCI 设备和位于片上总线的非 PCI 设备(即位于芯片上 AMBA 总线上的 AMBA 设备).该 PCI 虚拟化方法支持直接复用标准 PCI 设备的 Windows 驱动程序以访问相应的 AMBA 设备.和其他方法相比,本文提出的 PCI 虚拟化机制具有性能较高、软件兼容和硬件改变较小等优势,它并不局限于 AMBA 总线,适用于多种具有集中地址译码特性的片上总线.

2 背景介绍

2.1 Microsoft Windows 设备管理

PnP 管理器从系统主总线(大多数情况下是 PCI 总线)开始进行设备的枚举过程^[11].系统主总线的驱动程

序列举其总线上的设备,通过 PCI 配置周期来访问设备的 PCI 配置空间,这是一组大小为 256 字节的配置寄存器,提供了标准的系统配置机制,每个 PCI 设备都必须在其内部实现 PCI 配置空间.设备 ID 和厂商 ID 等设备信息被 PnP 管理器用来创建设备树(Device Tree),系统中的每个设备对应设备树中的一个节点.通过 PCI 配置周期收集每个设备的硬件资源需求情况,包括存储地址空间、I/O 地址空间、中断等,分配系统的资源,建立无冲突的硬件资源映射.INF 文件记录了设备 ID 信息,指定设备驱动程序的位置,寻找并安装设备驱动程序,INF 文件可以由厂商提供的,也可能是随 Microsoft Windows 一起发行的.系统主总线即 PCI 总线可能连接了 USB、ISA 和 SCSI 总线,这一递归的设备列举过程、驱动程序的加载不断进行,直到当前系统上所有的设备都已经被检测到、配置完成和正常启用为止.

通过工业标准的 PCI 设备枚举机制,Microsoft Windows 识别和访问标准 PCI 设备,本文将其作为第一个对比方案:PCI.

2.2 AMD VSA 机制

Geode GX 处理器的片上通信结构 GeodeLink 采用集中地址译码,而 PCI 总线是分布式地址译码,而且 GeodeLink 设备没有 PCI 配置空间.

为了使片上集成的 GeodeLink 设备被 Microsoft Windows 识别和访问,AMD 公司提出虚拟系统体系结构(Virtual System Architecture:简称 VSA)技术^[12].通过软件为每个 GeodeLink 设备提供了一个虚拟的 PCI 配置空间,在系统初始化时,这些虚拟 PCI 配置空间被加载到内存中^[13].当地址捕获电路监测到访问虚拟 PCI 配置空间的配置交易时产生一个使能信号,并通知中断产生电路发出同步系统管理中断(Synchronous System Management Interrupt,简称 SSMI),处理器接收到 SSMI 请求信号后,进入系统管理模式,通过中断处理程序访问位于内存中相应存储区域的虚拟 PCI 配置空间,配置交易的后续处理过程均和标准 PCI 方法类似,不同的是并非访问设备内部通过硬件实现的 PCI 配置空间,而是访问

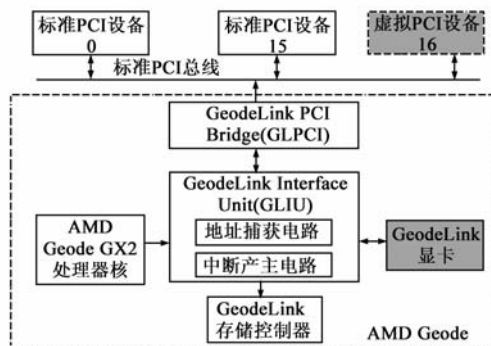


图1 AMD Geode结构示意图和VSA机制

位于内存的虚拟 PCI 配置空间,具有 GeodeLink 接口的显卡被虚拟化为位于标准 PCI 总线上的虚拟 PCI 显卡,如图 1 所示。

通过 VSA 的软件机制,将 GeodeLink 设备虚拟化为 PCI 设备,本文将其作为第二个对比方案:VSA。

3 PKUnity-86 的设备 PCI 虚拟化

本节介绍 PKUnity-86 SoC 中 AMBA 设备的 PCI 虚拟化,介绍 Microsoft Windows 识别和配置 AMBA 设备,以及使用标准 PCI 设备驱动程序访问 AMBA 设备的流程。

3.1 对比分析

PKUnity-86 SoC 是基于 x86 处理器和 AMBA 开放总线协议架构的,如图 2 所示。

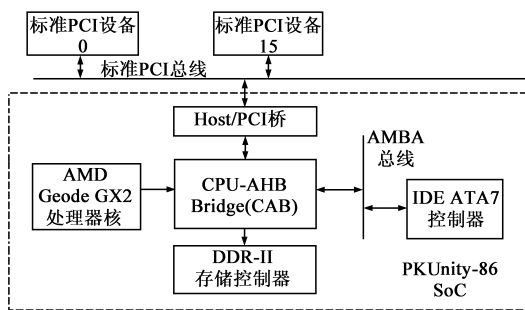


图2 PKUnity-86 SoC结构示意图

AMD Geode 的 GLPCI 和 PKUnity-86 的 Host/PCI 桥在系统中的位置是相同的。AMD Geode 中位于 PCI 总线上的设备内部实现了 PCI 配置空间,不必作任何改动,PKUnity-86 中 PCI 总线上的设备与之类似。GLIU 负责处理器与存储控制器、外部设备的通信,这和 PKUnity-86 中的 CAB 和 AHB 的功能类似。与 GeodeLink 设备类似,AHB 设备只有被虚拟化为 PCI 设备,才能被 Microsoft Windows 正常识别和访问,不同的是 GeodeLink 设备的虚拟 PCI 配置空间和 PCI 配置周期由软件来实现和处理,而本文提出的 AHB 设备 PCI 虚拟化是一种硬件实现机制。

基于以上分析,PKUnity-86 SoC 的 AMBA 设备是否被虚拟化为 PCI 设备,应遵循以下准则:PCI 总线上的设备不需要进行 PCI 虚拟化;具有驱动程序的设备,应被虚拟化为 PCI 设备,使得 Microsoft Windows 能发现该设备,并加载相应的设备驱动程序来访问该设备;在使用过程中产生中断的设备,应被虚拟化为 PCI 设备,使得 Microsoft Windows 为该设备分配中断号,并加载相应的中断服务程序处理该设备发出的中断。

PCI 配置空间的一些寄存器是与设备的功能相关的^[14],由于 AMBA 设备不具有 PCI 设备的某些特性,为减小硬件代价,我们为每个待虚拟化的 AMBA 设备实现少量必须的寄存器,包括设备识别寄存器(厂商标

识、设备标识、版本号、型号和头部型号),基地址寄存器和命令寄存器。实验表明,这个仅占标准 PCI 配置空间(256 字节)的 13.7% 的寄存器子集就能支持 AMBA 设备被 Microsoft Windows 识别和访问。

3.2 结构设计

在 PKUnity-86 系统中有两大类设备,一类是位于芯片外 PCI 总线的标准 PCI 设备,另一类是位于芯片内 AMBA 总线的 AMBA 设备。通过符合工业标准的 PCI 主总线驱动程序的遍历枚举,标准 PCI 设备可以被上层软件所识别。为了使 AMBA 设备也可以像标准 PCI 设备一样被上层软件正常地识别,我们在控制 PCI 设备访问的 Host/PCI 桥和控制 AMBA 设备访问的 AMBA 总线之间,设计了 CAB(CPU-AMBA Bridge)。

作为 CAB 的一个子部件,VPCI(Virtual PCI)模块为每个待虚拟化 AMBA 设备实现了一个虚拟 PCI 配置空间,它是能支持 AMBA 设备识别和访问的最小子集。AMBA 设备和标准 PCI 设备是统一编号且不冲突的。在上层软件看来,AMBA 设备是位于标准 PCI 总线 0 上的 16-31 号 PCI 设备,标准 PCI 设备是位于标准 PCI 总线 0 上的 0-15 号 PCI 设备以及其他标准 PCI 总线上等。

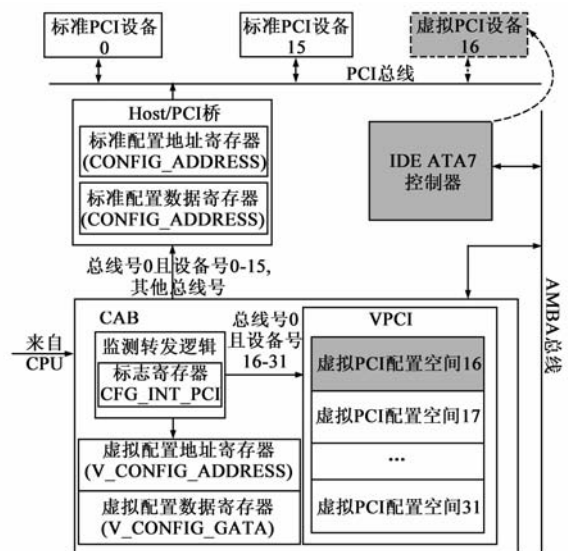


图3 PKUnity-86 SoC AMBA设备PCI虚拟化示意图

PKUnity-86 SoC 中的 VPCI 最多支持将 16 个 AMBA 设备虚拟化为 PCI 设备,其编号为 16 ~ 31,与标准 PCI 总线上 0 ~ 15 号 PCI 设备(编号 0 ~ 15)一起被 Microsoft Windows 识别为 0 号 PCI 总线上的 32 个 PCI 设备,这已能满足目前的大多数应用需求。对于这 16 个 AMBA 设备,VPCI 提供了一个用户可见的 16 位配置寄存器,每一位控制一个 AMBA 设备,取值为 1'b1 时表明对应的 AMBA 设备被虚拟为 PCI 设备,取值为 1'b0 时表明对应的 AMBA 设备不被虚拟化为 PCI 设备。该 16 位配置寄存器在系统初始化阶段由 BIOS 写入相应的值,由用户

决定将哪些 AMBA 设备虚拟化为 PCI 设备,在实际的系统应用中,被虚拟化为 PCI 设备的 AMBA 设备包括 IDE ATA7 硬盘控制器、MAC 网卡控制器、USB 控制器等.对于未来 SoC 上集成的更多的 AMBA 设备,VPCI 机制也能方便地进行扩展:将更多的 AMBA 设备虚拟化为位于 1 号、2 号等 PCI 总线上的 PCI 设备,每条 PCI 总线上可挂载 32 个虚拟或标准 PCI 设备,1 号 PCI 总线通过 PCI/PCI 桥与 0 号 PCI 总线连接并通信,2 号 PCI 总线通过 PCI/PCI 桥与 1 号 PCI 总线连接并通信,依次类推.

CAB 负责监测和转发来自处理器的访问设备的交易,包括配置交易、存储交易和 I/O 交易.对于配置交易,CAB 将所有总线号 0 设备号 16-31 的配置交易分发到 VPCI 部件,将所有总线号 0 设备号 0-15 和其他 PCI 总线的配置交易分发到 Host/PCI 桥. CAB 将所有访问虚拟 PCI 设备的存储交易和 I/O 交易转发给 AMBA 总线,将所有访问标准 PCI 设备的存储交易和 I/O 交易转发给 Host/PCI 桥,由其发给 PCI 总线.

配置端口是上层软件访问配置空间的唯一途径,访问配置空间的配置交易通过对配置端口的 I/O 交易来完成.虚拟配置地址寄存器(V_CONFIG_ADDRESS)和虚拟配置数据寄存器(V_CONFIG_DATA)分别对应从 CF8h 和 CFCh 开始的四个字节 I/O 端口,在功能上,它们分别和 Host/PCI 桥中的标准配置地址寄存器(CONFIG_ADDRESS)和标准配置数据寄存器(CONFIG_DATA)完全相同.

3.3 AMBA 设备的识别

地址阶段交易通过写配置地址寄存器,指明要访问的设备配置空间的总线号、设备号和配置寄存器偏移等地址信息;数据阶段交易通过读或写配置数据寄存器,来读或写地址阶段所指明的配置寄存器.

图 4 是上层软件访问 PCI 设备配置空间的基本工作流程.如果地址阶段交易的总线号和设备号是分配给虚拟 PCI 设备的,则 CAB 将标志寄存器 CFG_INT_PCI 设置为 1'b1,此值表明该配置交易是访问虚拟 PCI 设备的,捕获该配置交易的地址阶段交易以及后续的数据阶段交易,并转发给 VPCI 访问虚拟 PCI 配置空间. VPCI 将地址阶段的写数据写入虚拟配置地址寄存器中;数据阶段的读交易被 VPCI 响应,访问相应设备的虚拟 PCI 配置空间的相应配置寄存器,读取的数据被放入虚拟配置数据寄存器,并通过 CAB 返回给处理器;数据阶段的写交易被 VPCI 响应,写数据被放入虚拟配置数据寄存器,并写入相应设备的虚拟 PCI 配置空间的相应配置寄存器.如果地址阶段交易的总线号和设备号是分配给标准 PCI 设备的,则 CAB 将标志寄存器 CFG_INT_PCI 设置为 1'b0,此值表明该配置交易是访问标准

PCI 设备的,将该配置交易的地址阶段交易以及后续的数据阶段交易转发给 Host/PCI 桥,由 Host/PCI 桥访问相应的标准 PCI 设备的标准 PCI 配置空间.

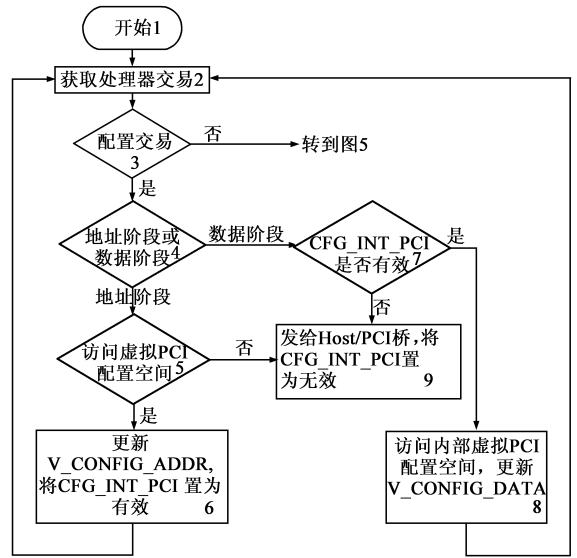


图4 虚拟PCI设备的配置交易处理流程

3.4 AMBA 设备的访问

PCI 总线支持三种地址空间:配置地址空间、I/O 地址空间和存储地址空间,而 AMBA 总线只支持存储地址空间,我们将 AMBA 设备虚拟化为 PCI 设备,就必须解决 PCI 总线和 AMBA 总线地址空间不一致的问题,使得 AMBA 设备支持上层软件对 PCI 设备的所有可能的访问方式.

如上章节所述,配置地址空间是独立于 I/O 和存储地址空间的,VPCI 和 CAB 共同支持上层软件对虚拟 PCI 设备(即 AMBA 设备)的配置空间的访问,完成设备的识别和初始化,以及设备驱动程序的加载.上层软件访问虚拟 PCI 设备的存储地址空间和 I/O 地址空间的基本工作流程如图 5 所示.

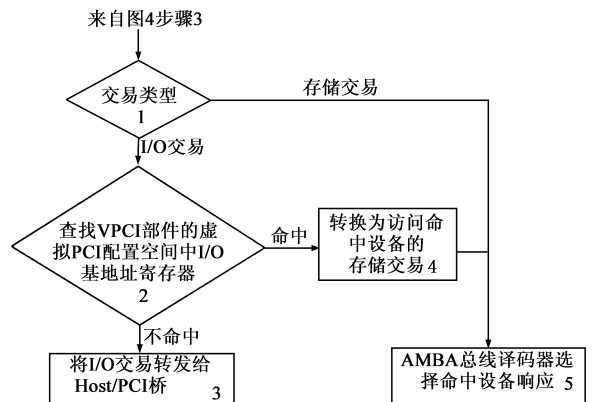


图5 虚拟PCI设备的存储交易和I/O交易处理流程

为使 AMBA 设备支持 I/O 交易,我们在系统 4GB 存储地址空间中保留一段连续的存储区域,任何一个 I/O

交易都被转换成对该存储区域的相应地址的存储交易。中断向量表、BIOS、Legacy VGA 存储和系统 RAM 位于最低 2GB,次高 1GB 是标准 PCI 设备的存储地址空间,最高 1GB 中 C000_0000h 到 COFF_FFFFh 的 16MB 是用于 I/O 地址空间映射的。对每个具有 I/O 功能的 AMBA 设备,我们在该段存储区域内分配 1MB 的连续空间。

4 实验

本节首先对本文提出的 VPCI 虚拟化机制进行功能验证,然后评估该机制的性能、硬件代价和软件开销,并与 AMD VSA 方法和标准 PCI 方法进行对比和分析。

4.1 功能验证

指令级全系统模拟器 Bochs^[15]模拟了一个比较完整的 PC 系统,在 Bochs 模拟器中验证本文 PCI 虚拟化机制的功能正确性,实验环境如图 6 所示。

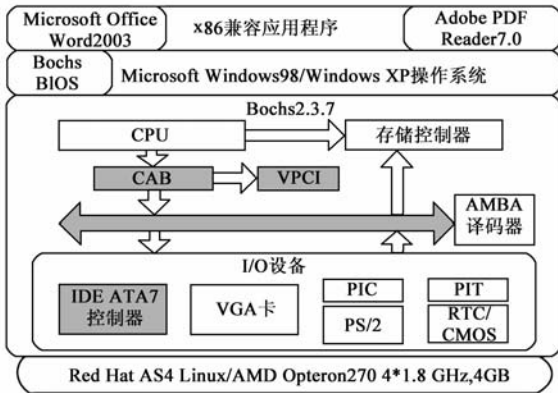


图6 VPCI虚拟化机制的功能验证

在 Bochs 模拟器中实现了 CAB 和 VPCI 行为模型,如图 6 灰色部分所示,这是 PKUsim-86 模拟器^[16]的重要组成部分。VPCI 为 IDE ATA7 控制器实现了一个虚拟 PCI 配置空间,CAB 监测访问 IDE ATA7 控制器的配置交易,并转发给 VPCI,由其控制对虚拟 PCI 配置空间的访问。CAB 监测访问 IDE ATA7 控制器的存储交易和 I/O 交易,对于存储交易,直接转发向 AMBA 总线,由 AMBA 总线译码器选择该设备进行响应,对于 I/O 交易,CAB 先转换成该设备所对应存储空间的存储交易,后续的处理与存储交易类似。

实验表明,基于本文 PCI 虚拟化机制的系统芯片可以启动 Microsoft Windows 98 和 Windows XP 操作系统,支持 x86 兼容应用程序;位于 AMBA 的 IDE ATA7 控制器和标准 PCI 设备一样,可以被正确地识别和访问:

- 在 Windows 设备管理器中,可以查看到该设备的资源类型和配置、以及驱动程序等详细信息
- 在 Windows 注册表中,可以看到当前系统检测到的所有硬件的设备 ID 等信息,包括 IDE ATA7 控制器。

4.2 结果分析

AMD GX DB533 开发板^[17]集成了 x86 处理器芯片,支持 AMD VSA 的软件 PCI 虚拟化机制。我们以该开发板为实验平台,在 BIOS 加入发起 PCI 配置交易的测试代码,通过标准 PCI 总线的配置交易访问 PCI 显卡的配置空间,通过 VSA 软件方法访问虚拟 PCI IDE 控制器的配置空间,对比 PCI 和 VSA 的访问配置空间的性能。

PKUnity-86 FPGA 板具有两颗 Xilinx Virtex-4 XC4LX200 芯片,分别实现 x86 处理器芯片和 PKUnity-86 SoC 设备,该 SoC 设备包括 CAB 部件,实现了本文的 VPCI 硬件 PCI 虚拟化机制。基于该 FPGA 原型,通过标准 PCI 总线的配置交易访问 PCI 显卡的配置空间,通过 VPCI 机制访问虚拟 PCI IDE 控制器的配置空间,对比 PCI 和 VPCI 的访问配置空间的性能。

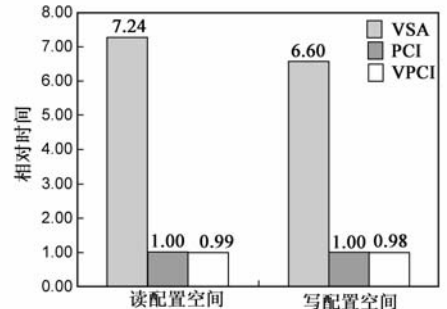


图7 VSA、标准PCI和VPCI读写配置空间的相对时间

三种方法的相对性能如图 7 所示。对于读和写配置空间,VPCI 所需要的时间是标准 PCI 方法的 0.99 和 0.98 倍,前者略好,这是在预期内的。与标准 PCI 方法类似,VPCI 也是采用硬件来捕捉配置交易,并访问硬件实现的虚拟 PCI 配置空间,区别在于 VPCI 并不在 PCI 总线上发起配置交易,而是直接访问芯片上的寄存器,寄存器的访问只需要几个 CPU 周期,而根据 PCI 总线协议,位于芯片外标准 PCI 设备内部配置寄存器的访问需要更长的时间。

图 7 表明 VPCI 的性能优于 VSA 方法。对于读和写配置空间,VSA 所需要的时间是 VPCI 的 7.31 和 6.73 倍。这是因为 VSA 通过软件实现虚拟 PCI 配置空间,在系统初始化时加载到内存中,配置空间的访问触发 SSMI 中断、保存执行现场、执行 SSMI 中断处理程序、访问内存中的虚拟 PCI 配置空间以及恢复执行现场等一系列过程将耗费大量处理器时间。VPCI 机制对虚拟 PCI 配置空间的读写不需要发起总线交易,通过读写寄存器完成,只需要几个处理器周期,远远低于总线交易和内存访问交易,VPCI 机制的读写性能不会构成 SoC 性能瓶颈。随着 SoC 上集成的 IP 核越来越多,其主要性能瓶颈之一将在于 IP 核之间的互联结构和通信^[18]。

在硬件代价上,VSA 所需要的硬件支持最少,仅需实现相关逻辑捕捉的配置交易,实现少量寄存器来记

录系统资源分配信息来进行地址译码;VPCI的硬件代价仅为标准PCI方法的13.7%。对于VPCI方法,所有设备的虚拟配置空间集中地位于一个硬件模块中,只需实现若干设备识别和访问必须的配置寄存器。而对于标准PCI方法,每个PCI设备内部都实现了全部的256字节PCI配置空间。即使集成更多的IP核,其配置空间所需的硬件开销相对于SoC中的大量功能部件来说几乎可以忽略不计。

在软件开销上,对于VPCI和标准PCI是最小的,在上层软件看来,这两种方法均支持工业标准的PCI设备枚举和访问,可以复用标准PCI设备驱动程序;而VSA的软件开销最大,通过软件实现虚拟PCI配置空间,实现独立的中断处理程序。

5 总结

本文提出了一种在Microsoft Windows下将AMBA设备虚拟化为PCI设备的方法。它并不局限于AMBA,适用于多种片上总线,如CoreConnect和Wishbone。CAB根据总线号和设备号,将访问标准PCI设备的配置交易转发给Host/PCI桥,将访问AMBA设备的配置交易转发给VPCI,由其访问内部的虚拟PCI配置空间,将AMBA设备虚拟化为在上层软件看来无差别的PCI设备。CAB将I/O交易转换为存储交易,使得上层软件使用访问标准PCI设备的方法来访问AMBA设备。

在Bochs模拟器,PKUnity-86 SoC成功运行Microsoft Windows 98和Windows XP,以及x86兼容应用程序,证明了本文PCI虚拟化机制的正确可行性。在AMD GX DB533开发板和PKUnity-86 FPGA上的实验表明,和标准PCI方法相比,VPCI方法具有较小的硬件开销和较好的访问性能;VPCI读写配置空间的性能分别是AMD VSA的7.31和6.73倍;它不需要额外的软件开发,支持复用标准的PCI设备驱动程序,使得PKUnity-86 SoC AMBA设备可以被Microsoft Windows正确地识别和访问。

参考文献

- [1] 黄凯,殷燎,林锋毅,葛海通,严晓浪.一种多处理器原型及其系统芯片设计方法[J].电子学报,2009,37(2):305-311.
Huang Kai, Yin Liao, Lin Feng-yi, Ge Hai-tong, Yan Xiaolang. A multiprocessor prototype and its SoC design methodology[J]. Acta Electronica Sinica, 2009, 37(2): 305-311. (in Chinese)
- [2] 王蕾,陆洪毅,王进,戴葵,王志英.一种面向嵌入式应用的片上系统:腾跃-1[J].电子学报,2005,33(11):2036-2039.
Wang Lei, Lu Hong-yi, Wang Jin, Dai Kui, Wang Zhi-ying. A high performance embedded SoC: Teng Yue-1[J]. Acta Electronica Sinica, 2005, 33(11): 2036-2039. (in Chinese).
- [3] Brad Smith. ARM and Intel battle over mobile chips future[J]. IEEE Computers, 2008, 41(5): 15-18.
- [4] 熊志辉,李思昆,陈吉华,王海力,边计年.一种基于层次平台的SoC系统设计方法[J].电子学报,2004,32(11):1815-1819.
Xiong Zhi-hui, Li Si-kun, Chen Ji-hua, Wang Hai-li, Bian Ji-nian. Hierarchical platform-based SoC system design method[J]. Acta Electronica Sinica, 2004, 32(11): 1815-1819. (in Chinese)
- [5] IBM Corporation. CoreConnect bus architecture [EB/OL]. <http://www.ibm.com>, 1999.
- [6] ARM Limited. AMBA™ specification 2.0[Z]. <http://www.arm.com>, 1999.
- [7] Silicore Corporation. WISHBONE System-on-Chip interconnection architecture for portable IP cores [EB/OL]. <http://opencores.org/opencores/wishbone>, 2002.
- [8] 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.
- [9] 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),2008,45(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)
- [10] 顾明,赵曦滨,郭陟,孙家广.现代操作系统的思考[J].电子学报,2002,30(S1):1913-1916.
Gu Ming, Zhao Xi-bing, Guo Zhi, Sun Jia-guang. The overview on modern operating system [J]. Acta Electronica Sinica, 2002, 30(S1): 1913-1916. (in Chinese)
- [11] Mark E. Russinovich, David Solomon. Microsoft Windows Internals: Microsoft Windows Server 2003, Windows XP and Windows 2000 [M]. U. S. Washington: Microsoft Press, 2005.
- [12] AMD. AMD Geode™ GX processor based systems VSA2 programmer's guide [Z]. 2006.
- [13] AMD. AMD Geode™ GX and LX processor based systems virtualized PCI configuration space [Z]. 2006.
- [14] PCI Special Interest Group. PCI local bus specification revision 2.2 [Z]. 1998.
- [15] Lawton K, et al. The Bochs IA-32 emulator project [EB/OL]. <http://bochs.sourceforge.net>, 2008-06-08/2009-12-03.
- [16] 庞九凤,佟冬,李皓,何浪,程旭.面向基于x86处理器和AMBA的系统芯片的全系统模拟器PKUsim-86[J].电子

学报,2010.

Pang Jiu-feng, Tong Dong, Li Hao, He Lang, Cheng Xu. PKUsim-86: A cycle level full system simulator for x86 processor and AMBA-based SoC[J]. Acta Electronica Sinica, 2010. (in Chinese)

[17] AMD. AMD Geode™ GX DB533 development board product brief[Z]. 2006.

[18] 陆俊林. 系统芯片中片上通信结构的研究与实践[D]. 北京:北京大学微处理器研发中心,2009.

作者简介



庞九凤 女,1985年1月出生于河南省濮阳市.博士研究生,主要研究领域为软硬件协同设计、计算机模拟性能评估和系统芯片设计.

E-mail: pangjiufeng@mprc.pku.edu.cn



陆俊林(通信作者) 男,1980年1月出生于浙江省杭州市.博士后,主要研究领域为软硬件协同设计、系统芯片设计和片上通信结构.

E-mail: lujunlin@mprc.pku.edu.cn