第 34 卷 第 1 期 计 算 机 工 程 2008 年 1 月
Vol.34 No.1 Computer Engineering January 2008
·工程应用技术与实现· 文章编号：1000—3428(2008)01—0241—03 文献标识码：A 中图分类号：TP302

多处理器芯片组中 PCI 桥控制器的设计与实现

方志斌[1,2]，孙凝晖[1]，安学军[1]，胡 鹏[1,2]
(1. 中国科学院计算技术研究所计算机系统结构重点实验室，北京 100080；2. 中国科学院研究生院，北京 100039)

摘 要： 分析了多处理器芯片组内通信和 PCI 通信的特点，设计并实现了在切入通信机制下的 PCI 桥控制器，该控制器在 FPGA 布局布线后可以达到 66 MHz，有效隐藏 PCI 协议中的突发传送和读延迟机制给多处理器切入通信带来的性能损耗。该控制器已稳定运行在龙芯多处理器系统中。

关键词： PCI 桥控制器；多处理器；芯片组；切入

Design and Implementation of PCI Bridge Controller in Multi-processor Chipset

FANG Zhi-bin[1,2], SUN Ning-hui[1], AN Xue-jun[1], HU Peng[1,2]
(1. Key Lab of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080;
2. Graduate School, Chinese Academy of Sciences, Beijing 100039)

【Abstract】 This paper analyzes the features of communication in multi-processor chipset and PCI communication. It designs and implements PCI bridge controller in cut-through communication. The frequency of the controller is up to 66 MHz after placing and routing in FPGA, and it can effectively hide the delay of performance brought by burst transfer and delay read in PCI protocol in multi-processor cut-through communication. The PCI controller is used in Godson multi-processor system.

【Key words】 PCI bridge controller; multi-processor; chipset; cut-through

为突破机群内处理器间通信能力受限于I/O总线的局限，本文在新一代机群的研究中将多个龙芯处理器通过芯片组直接互连组成基本节点，以此设计并实现了未来低成本高性能的计算机。该系统实现了全局统一编址，芯片组可连接 4 个处理器，每个处理器有独立的内存控制器直连内存，可以直接访问全部物理内存；芯片组内多处理器间采用切入方式进行通信。多处理器系统需要I/O支持，比如硬盘和显示器等外部设备，目前符合PCI规范的I/O设备种类齐全，可选择范围广，因此，系统采用PCI规范实现I/O总线，要求多处理器芯片组实现PCI控制器。目前开源社区已实现 32 位内部总线的PCI桥控制器[1]，但其实现的通信协议采用了突发传送和读延迟机制，在多处理器间通信的环境下性能会受影响。

1 通信机制

芯片组内的 PCI 桥控制器需要实现与多处理器的通信，以及与 PCI 设备的通信，因此，需要对这 2 种通信机制进行分析，以此作为设计的基础。

1.1 切入通信机制

多处理器间通信采用切入机制来减少交换延迟，其特征是：传输的包规整，包头标记包的固定长度；接收方收到包头信息后不必等待包尾到达就可向下级接收方传递；缓冲区容量大于数据包的大小，阻塞时将整个数据包存储在本地的缓冲区内，流量控制以包为单位[2]。处理器的读操作以分离事务协议完成，即请求方发出读命令后，断开连接，等待应答方准备好数据，之后应答方作为主设备，发起一次新的读响应将读数据返回。

多处理器芯片组内部总线协议采用切入通信机制，地址线与数据线分离，数据线 64 位。内部总线协议分成 Slave 和 Master 两部分，通过交叉开关实现点对点通信。Master 用于传送数据，是总线交互的发起者，通过 CYC 信号有效标志一次传送请求的开始；Slave 用于接收数据，是总线交互的接收者。Slave 看到 CYC 有效后，如果当前缓冲区容量大于或等于 Master 请求传输的数据量，则向 Master 发出 ACK 信号，反之则不发出 ACK 信号。Master 看到 ACK 信号后把数据传送给 Slave，数据是否有效依靠 STB 指示，数据有效期间内 STB 为高电平。CYC 信号有效时，CMD 指示本次传送的固定长度。具体时序如图 1 所示。

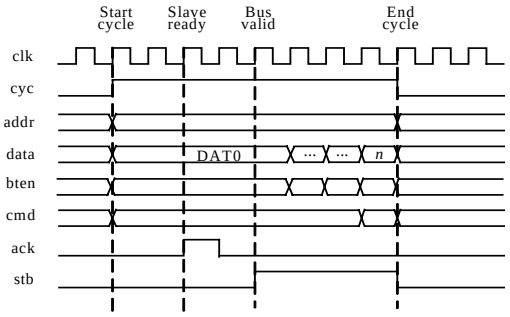

图 1 内部总线交易时序

基金项目： 中国科学院创新课题基金资助项目“新一代机群关键技术研究”(20054010)；中国科学院计算技术研究所创新课题基金资助项目“基于龙芯的 SMP 芯片组及其相关系统的研究”(20046080)
作者简介： 方志斌(1973－)，男，博士研究生，主研方向：计算机体系结构；孙凝晖，研究员、博士；安学军，副研究员、博士；胡 鹏，博士研究生
收稿日期： 2007-03-22 **E-mail：** fzb@ncic.ac.cn

1.2 PCI 通信机制

PCI 协议可参考文献[3-4]，其通信一般采用突发机制，每次传送数据的长度不固定，由总线主设备和目标共同决定；在交易开始时，目标得到起始地址和交易类型，但没有传送长度；主设备准备传送每一个数据项时，会通知目标是否为最后一个数据项，目标通知主设备是否能够继续传送，当最后一个数据项传送后交易即告结束。

PCI 读操作以延迟重试完成，即请求方发出读命令后，若应答方准备好则传送，若未准备好，则断开连接，延迟后继续发出读命令重试。

PCI 的读延迟重传机制在读命令被接收但读响应尚未完成时，主设备依然不断进行读重传申请，同时突发机制会使带来数据包长度不定，这些特点会降低芯片组内多处理器间通信性能。

2 PCI 桥控制器的设计和实现

针对 2 种通信机制的特点，本文不仅设计了各自的状态机以实现对应接口，还设计异步 FIFO 及其控制逻辑以实现 2 种不同通信机制协议间的互连和转换，最后解决了 PCI 地址转换等相关问题。

2.1 PCI 桥控制器组成结构

如图 2 所示，PCI 桥控制器的组成包括 4 个部分：PCI 接口模块，内部总线接口模块，内部缓冲区及其控制逻辑模块和配置空间寄存器。PCI 接口模块包括 PCI 总线接口、PCI 主模块、PCI 目标模块。内部总线接口模块包括内部 Master 模块和内部 Slave 模块，这 2 个模块挂在芯片组内部的 crossbar 接口上。内部缓冲区及其控制逻辑模块包括 7 个 FIFO 模块。

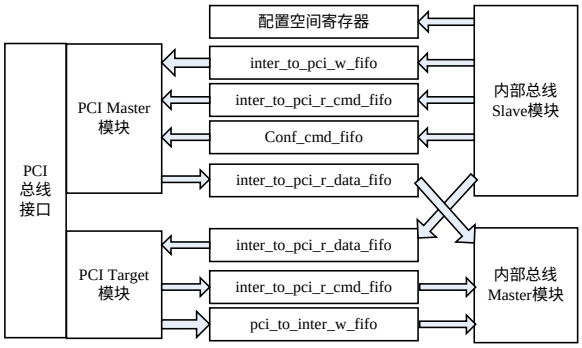

图 2 PCI 桥控制器结构

2.2 PCI 接口设计与实现

PCI 接口实现的功能包括在 PCI 物理总线上进行 I/O 读写、内存读写等交易，作为主桥，它的功能还包括根据内部总线命令产生对外围 I/O 设备的配置读写命令。PCI 桥控制器分主设备和目标设备 2 部分实现。

PCI 主设备在如图 3 所示的有限状态机下工作。无交易时，处于空闲状态 idle；当需要进行一次 PCI 交易时，通过 REQ#请求总线，PCI 仲裁器仲裁，若许可则有效其 GNT#，然后主设备有效其 FRAME#，并进入地址状态 address。地址状态的第 1 个 PCI Clock 产生地址段，主设备发出地址与命令，然后循环进入 address 状态等待，在一个协议约定的周期内若无任何目标响应命中（有效 DEVSEL#），则主设备返回 idle 状态；若目标响应命中但目标中的数据未准备好或主设备只传送一个数据段，则进入 turn-around 状态，若主设备传送多个数据段则进入 data 状态。

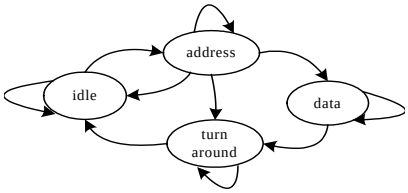

图 3 PCI 主设备与目标设备的有限状态机

在 data 状态，只有主设备的 IRDY#和目标的 TRDY#均有效时才传送数据，如果目标在此过程中无法继续交易则通过 STOP#来中止此次交易。在传送交易过程中，主设备需要 FIFO 的飞行字节来识别是否为最后一次交易，因为写缓冲的数据或读缓冲的空闲位需要 2 个 PCI Clock 才能到达 PCI 接口边界，所以状态机需要读或写缓冲 FIFO 提供当前时钟后 2 个时钟内是否到尽头。主设备通过无效 FRAME#并有效 IRDY#来标识最后一次交易。当主设备进行最后一次交易或目标发出 STOP#时，进入 turn-around 状态，如果是主设备进行最后一次交易，但目标未准备好，循环进入 turn-around 状态直到目标准备好。

主设备与目标设备需要在交易的所有非 idle 状态进行协商握手，在每个非 idle 状态下需要采样 PCI 总线上的状态信号，这种协议使总线频率难以提高，每次交易的传输长度无法预先判定。

目标设备的状态机与主设备的状态机基本相同，当目标被译码命中时进入 address 状态，在此状态下不能直接返回 idle 状态。在非 idle 状态中，目标设备同样需要 FIFO 的飞行字节来标识其读响应数据或接收缓冲是否准备好，若未准备好则有效 STOP#以通知主设备进入 turn-around 状态。在 data 状态，目标设备采样到主设备无效的 FRAME#和有效的 IRDY#，则进入 turn-around 状态。

2.3 内部总线接口设计与实现

内部总线接口实现 64 位内部总线的读与写的功能，包括 Master 和 Slave 两部分。Master 需要进行内部总线数据传送时，首先有效 CYC 进入 address 状态，在 address 状态 Master 发出地址和命令，同时标记此次读写交易的数据长度。一旦 Slave 被译码选中，如果当前缓冲区容量小于 Master 请求传输的数据量，则 Slave 发 RTY 信号使 Master 返回 idle 状态；如果当前缓冲区能容纳 Master 的传输请求，对于写操作，按双方在 address 状态协商好的数据长度进入固定的状态，不再进行双方的握手；对于读操作，Master 主动放弃连接，返回 idle 状态等待，此次交易的 Slave 方将响应数据准备好后，以 Master 的身份发起读响应交易将数据返回，因此，一次读操作包含了 2 次交易，但读操作的 Master 方无须进行重试连接。

在图 4 所示的状态机中，data 状态中每一个 clock 均须传输数据，last 状态只持续一个周期，这 2 个状态下的转换被大大减化，有利于提高频率，并实现虚切入机制的多处理器间通信。内部总线 Slave 的操作按照图 4 所示进行，在 address 状态做存取权限检查，并进行译码和重映射，在译码的基础上决定其所对应的 5 个去向。

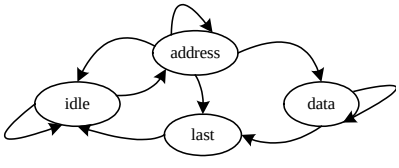

图 4 内部总线 Master 与 Slave 的有限状态机

2.4 内部缓冲区及其控制逻辑

内部缓冲区将 PCI 接口与内部总线接口互连，需要实现从 PCI 接口到内部总线接口和从内部总线接口到 PCI 接口 2 个方向的数据传递通道，每个方向按照读与写性质的不同分成 2 类，因此设计了 4 组 FIFO，写 FIFO 中包含命令与数据，可以容纳多次写事务，读 FIFO 中命令与数据分离，读命令 FIFO 可以容纳 4 次读事务。

4 组 FIFO 需要实现异步 FIFO，因为设计内部总线协议时需要尽可能提高内部总线频率、以降低多处理器间通信延迟，因此，必须考虑到 PCI 接口与内部总线接口工作在不同的时钟域，即对缓冲区的读与写操作在不同时钟域。PCI 接口与内部总线接口的状态机都需要内部缓冲区的包括读写在内的所有状态，因此，需要格雷码将不同时钟域的读写指针转换到同一时钟域以产生内部缓冲区的状态。对读命令缓冲区的读请求与读响应命令也需要进行跨时钟域的传递。

设计内部总线协议采用地址与数据线分离，数据线 64 位以提高通信带宽，因此，所有数据缓冲区必须完成 64 位内部总线数据和 32 位 PCI 总线数据的对齐和转换，这是通过对 FIFO 读、写指针的控制来实现的。对于 64 位转换到 32 位，可以采用 1 次写 2 次读的方式实现；对于 32 位转换到 64 位，采用 2 次写 1 次读的方式实现，同时判断初始 32 位数据是否需要对齐、最后 32 位数据是否需要填充。

从 PCI 接口接收的写数据是按突发方式组织的，没有固定长度，但内部总线接口按切入方式传送，需要固定长度，因此，需要对 PCI 接口到内部总线接口的写缓冲区进行组织并打包传送。

2.5 PCI 总线地址、内存总线地址的转换

PCI 桥控制器连接内部总线接口和 PCI 接口，需要实现数据通道及地址空间的转换。对于来自 CPU 的内存总线地址，需要对其译码并转换为内部寄存器、PCI 内存地址、PCI I/O 地址或 PCI 配置命令。对于来自 PCI 总线的 PCI 地址，需要对其译码并转换为内存总线地址。

3 仿真验证和性能分析

本文的工作使用 Verilog HDL 硬件描述语言实现，在 Xilinx 公司的环境中开发，为验证逻辑的正确性，独立开发了虚拟的 PCI 设备以及内部总线接口设备，在仿真正确的基础上，采用 Xilinx 公司的 Virtex 4 系列 XC4VLX60 芯片在项目组设计的 PCB 上进行验证，该控制器使用了 4 465 个 LUT、5 个 BlockRAM，布局布线后的工作频率可以达到 66 MHZ。

通过仿真以及利用 Xilinx 公司的 Chipscope 调试工具采样到如下结果：PCI 桥控制器在缓冲区为空的条件下，接收到来自内部总线的请求后，经 2 个内部时钟和 2 个 PCI 时钟就可以到达 PCI 接口边界，然后利用 1 个 PCI 时钟作寄存即可输出到 PCI 总线上，反方的周期延迟相同。在空载的条件下，PCI 桥控制器发出对内存的请求后至少等待 20 个内部时钟才能得到读响应数据，如果负载加重，读响应等待周期会成倍增加，PCI 控制器采用分离读，大大降低了在长延迟等待读响应中的 Crossbar 争用冲突，提高了多处理器间通信性能。

为了研究 PCI 接口模块的设计参数在多处理器芯片组条件下的优化，尤其是缓冲区深度对通信性能的影响，本文利用 ModelSim 设计了一个仿真环境，在该环境中实现了符合内部协议的虚设备和符合 PCI 规范的虚设备。实验中测试的缓冲区深度依次选为 512 B, 2 KB, 8 KB，仿真中假设处理器接口需要利用 Crossbar 连续发送若干个数据块，该假设符合芯片组集成多个处理器的通信需求特征，连续发送数据块大小按照 MPI 通信的特征选择 1 KB, 4 KB, 16 KB, 64 KB, 256 KB，每次交易的基本单位在芯片组支持的包长度范围(8 B~256 B)中取中间值 32 B，按照目前实测的 Crossbar 通信协议的延迟，每次交易发送 32 B 需要 12 个周期，即理想状况无限大缓冲区条件下 PCI 接口带宽为 (4/12) × 640= 213 MB/s，以该带宽为基准，测试各种条件下的带宽与基准值的比率如图 5 所示。

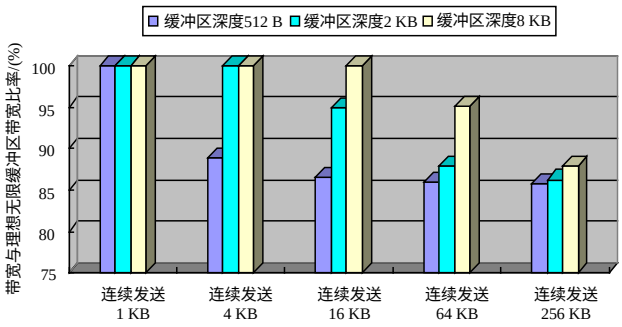

图 5 PCI 接口各种情况下带宽与理想无限缓冲区带宽比率

由图 5 可以得出，芯片组 PCI 接口带宽随缓冲区深度的增加而提高，小块数据传输利用率提高的比率大，大块数据传输利用率提高的比率小。在连续发送 256 KB 数据的情况下，缓冲区深度即使从 512 B 增加到 8 KB，带宽基本保持不变，这表明在传输大块数据时加大缓冲区深度对带宽的提高没有显著作用。PCI 接口带宽比率随连续发送数据的增加而大幅降低，这表明随着芯片组多处理器数目的增多，通信的压力和需求逐步增大，外设接口的带宽反而逐步降低，形成通信性能恶化的连锁反应，因此，需要对各接口的带宽和参数进行更平衡的设计。

目前，本 PCI 桥控制器稳定运行在基于龙芯的多处理器系统中，能够挂接符合 PCI 规范的硬盘卡、显卡和网卡。

4 结束语

本文设计并实现了多处理器间芯片组中的 PCI 桥控制器，该控制器能够降低 PCI 协议中的突发传送和读延迟给多处理器切入通信带来的性能损耗。目前市场上 PCI 设备驱动程序的开发和调试因为有开放源代码的支持，降低了开发难度，加快了开发进度。

参考文献

[1] Dolenc M, Markovic T. PCI IP Core Design Document[Z]. (2002-01-24). http://www.opencores.org/cvsget.cgi/pci/doc/pci_design_docu-ment. pdf.

[2] Culler D, Singh J P. Anoop Gupta. Parallel Computer Architecture：A Hardware/Software Approach[M]. [S. l.]: Morgan Kaufmann Press, 1998-08-01.

[3] Shanley T, Anderson D. PCI 系统结构[M]. 北京: 电子工业出版社, 2000.

[4] PCI Special Interest Group. PCI Local Bus Specification(Rev. 2.3)[Z]. (2001-10-31). http://www.pcisig.com.