PCIe 总线接口的多处理器数据传输技术

曹 月[1] 张奕然[1] 徐锦涛[1]
CAO Yue ZHAGN Yiran XU Jintao

摘 要 PCIe 总线接口在分布式处理平台中可以满足各任务系统之间的高速率数据传输，已成为多处理器或处理器与外设交互的主要方式，但各任务系统之间的海量数据交换存在传输不稳定问题。针对以上问题，提出了一种基于 PCIe 总线的多处理器数据传输技术的设计方案，设计了以国微电子公司的国产 SM8748 交换芯片为核心、以飞腾八核处理器 FTD2000/8 为根设备的互连系统。系统采用一片 FTD2000/8 处理器作为根节点，另一片 FTD2000/8 处理器、飞腾双核处理器 FT2000AHK 以及其他两个计算节点作为端设备，实现外部数据与内部数据的控制交换。实验测试根节点 FTD2000/8 处理器与端节点之间通过单字节 PIO 与直接存取 DMA 两种方式的数据读写传输带宽，结果表明，设计的 PCIe 总线接口的多处理器数据传输软件能够稳定可靠地实现数据交换。

关键词 PCIe 总线；多处理器；FTD2000/8 处理器；互连系统；PIO；DMA

doi：10.3969/j.issn.1672-9528.2024.04.040

0 引言

现代处理器对数据通信与实时事件快速响应能力要求越来越高，PCIe 总线作为一种主流的高速串行传输总线，在多处理器系统的数据传输方面具有一定的优势。PCI 总线使用共享并行总线架构，所有设备只能轮流占用 PCI 总线，当其他设备长期占用时，将会影响到总线的传输质量。而 PCIe 总线基于点对点的拓扑结构，每个 PCIe 设备都采用单独的链路与主机通信。独立的通道资源保证了端设备独享带宽，提高了设备之间数据的传输速率，实现了多设备同时通信的可能 [1]。PCIe 总线单通道由一对发送差分总线与一对接收差分总线组成，单通道为 x1，设备之间的物理连接可根据实际需要配置为 x1、x2、x4、x8、x16 等并行通道，来满足不同的带宽需求，在链路训练过程中确定 PCIe 总线的链路宽度、链路速率等。由于 PCIe 点对点的特性，一条 PCIe 链路上只能挂接一个设备，这对于现代大规模处理单元的系统架构而言显然不够。PCIe 交换实现了链路扩展功能，允许更多的设备连接到一个 PCIe 端口，为多处理器之间的数据交换提供了一种高效的手段 [2]。

国产飞腾八核处理器 FTD2000/8 的 PCIe 接口采用 PCIe 3.0 规范，支持 RC（RootComplex）和 EP（EndPoint）两种模式，设计了两个 PCIe 单元 PEU（PCIeUnit），每个 PCIe 单元中包含 x16、x1 两个 PCIe 接口，每个 x16 PCIe 接口又可进一步拆分为两个 x8，主要有四种访问传输方式：向内部 PCIe 控制器或者外部 PCIe 设备发起配置空间访问；以 PIO（Programmed I /O）数据传输方式向外部 PCIe 设备发起 Memory/IO 空间访问；以 DMA 数据传输方式进行外部 PCIe 设备与本地 DDR 存储器之间的数据传输；解析执行外部 PCIe 设备对本地 DDR 存储器的访问 [3]。

本文采用 FTD2000/8 处理器，基于国微电子的国产交换芯片 SM8748，设计并实现了一种 PCIe 总线接口的多处理器数据传输驱动软件。

1 硬件架构

某项目某模块承担分布式处理平台中的与各任务系统功能有关的智能计算处理。本文以其为背景，设计并实现了 FTD2000/8 处理器下基于 PCIe 通信的数据传输技术。该模块采用 SM8748 交换扩展 PCIe 总线，形成以 FTD2000/8 处理器为根节点的一主多从的树形拓扑结构。

通用处理器 FTD2000/8 作为系统的根节点 RC，挂接在 SM8748 交换的上游端口。4 个下游端口分别与一个通用计算节点 FTD2000/8、一个 FPGA 节点、一个 DPU 节点和一个 GPU 连接器连接，硬件架构图如图 1 所示。FTD2000/8 处理器能够完成复杂通用计算任务，实现海量数据的计算以及复杂通用计算任务。DPU 节点以 FPGA 为核心进行构建，外部配置 FT2000AHK 处理器，实现数据处理以及两个通用计算节点之间的数据传输功能，支持计算平台高速无损以太网互联。GPU 节点用于传感器中的识别类、任务系统中的决策类等智能计算处理。FPGA 能够部署信号处理 NPU IP 核实现信号处理加速功能。

1. 航空工业计算所 陕西西安 710068

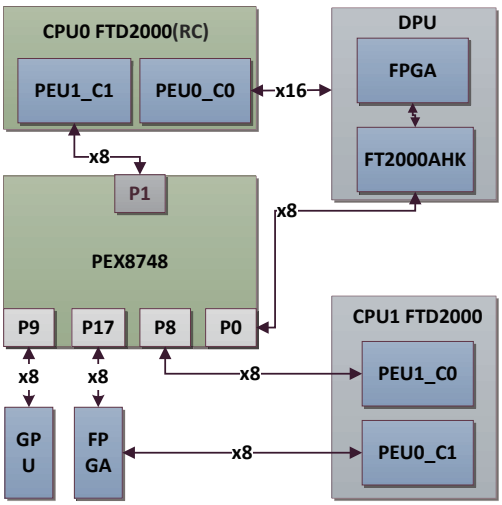

图 1 硬件架构图

2 PCIe 交换设计

SM8748 为深圳国微公司提供的交换芯片，具有 48 个通道，最多 12 个端口，可实现各端口之间的数据交换。在基本模式下，SM8748 交换所在的整个 PCIe 系统表现为树形拓扑结构，仅有一个根节点 RC，RC 管理该 PCIe 系统下的所有设备，为设备分配空间并实现根节点与端节点之间的数据传输。

SM8748 交换通过配置 STRAP_STNx_PORTCFG 管脚决定每个端口的链路宽度，链路宽度可选值为 x16、x8、x4，并可自协商至 x2 和 x1。在 PCIe 通信设计方案中，为交换芯片配置 STRAP_STN[0:2]_PORTCFG[0:1]，管脚为 01（下拉、上拉），形成 6 路 x8 链路宽度的通信模式，实际应用中使用一路 x8 与作为 RC 端的 FTD2000 处理器主桥相连，选用四路 x8 端口分别连接作为 EP 端的 FTD2000/8 处理器、GPU 连接器、DPU 模块以及 FPGA，最后一路 x8 端口未使用。

3 PCIe 驱动程序设计

本文针对某项目某模块的通信需求，主要对系统架构中 CPU0（RC 端）和 CPU1（EP 端）的处理器软件设计进行研究，其他 GPU、FPGA、DPU 节点均是以 CPU0 为根节点的 PCIe 设备，程序设计与之类似。

RC 端与 EP 端的处理器为 FTD2000/8，在此处理器上适配 32 位天脉容器操作系统。PCIe 驱动软件的实现位于核心层。RC 与 EP 双向通信的方式为 PIO 或 DMA，作为 RC 的 FTD2000/8 驱动程序逻辑上可以分为三部分：PCIe 初始化、DMA 驱动以及中断配置，作为 EP 的 FTD2000/8 驱动程序在 RC 端驱动的基础上增加 BAR 属性配置、窗口映射部分。PCIe 初始化主要调用板级中的 API，以查看链路训练状态、映射内存空间、分配 DMA 描述符空间以及绑定中断处理函数 [4]。DMA 驱动包括 DMA 读写两部分，中断配置根据 FTD2000 处理器芯片手册的 MSI-SPI 通路中断设置相关参数。

3.1 PCIe 总线初始化

启动天脉容器操作系统之前，uboot 可以配置处理器为 RC 或 EP 模式及 PCIe 接口的链路速率与宽度，RC 可以扫描主桥下的 PCIe 设备，但是扫描结果无法直接使用。uboot 引导操作系统内核后，启动板级包的 PCIe 驱动程序，查看链路训练状态，挂接 MSI 中断服务函数，最后进行总线设备枚举。

PCIe 总线设备枚举分为两步：深度优先搜索和分配资源。深度优先搜索从 RC 开始，自顶向下遍历每个 PCIe 设备，来获取整个总线拓扑上的设备信息。在搜索过程中，通过配置信息获取该设备是否为桥设备，如果是桥设备则扩展一个总线号，延续深度优先法搜索该总线，直至获取到整个系统的拓扑信息与每个设备的资源信息。分配资源同样从根节点出发，每扫描到一个设备，便从系统的总资源中为其分配。桥片需更新总线号及地址范围，来实现 TLP（transaction layer packet）报文的路由 [5]。PCIe 总线设备枚举流程如图 2 所示。

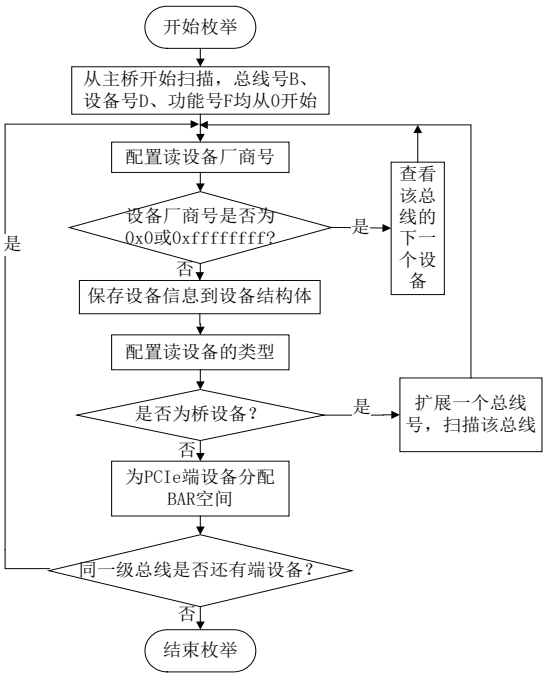

图 2 PCIe 总线设备枚举流程图

3.2 BAR 属性设置

RC 端扫描算法执行之前，EP 端根据需求设置控制器模式及 BAR 属性。本文将 CPU1 处理器（EP 端）的 PEU1_C0 控制器配置为 EP 模式，写 PCIE_BAR_01 寄存器为 0x00000000ff000000，即配置 EP 端的 BAR0 属性为 32 位不可预取的内存空间，BAR 大小为 0x100000（1 MB）。

EP 端 BAR 属性配置完成后，RC 端驱动程序采用深度扫描算法完成端设备与桥设备的地址空间分配，当 RC 主桥扫描到 EP（CPU1）时，发送配置写报文向设备配置空间 BAR 寄存器写全 1，获取 EP 端 BAR 属性，包括 BAR 空间大小、类型、地址宽度等，再分配对应的空间，进行数据交换。

3.3 窗口映射

FTD2000/8 处理器具备 PCIe 设备地址空间的配置与管理能力，其根节点下的外部设备在允许的范围内可以映射到系统内存，通过存储映射方式访问外部设备。窗口映射包括从 PCIe 域到存储器域地址转换（outbound 映射）与从存储器域到 PCIe 域地址的转换（inbound 映射）。FTD2000/8 处理器作为 RC 时，inbound 与 outbound 映射为“直接相等”，即存储器域的地址与 PCIe 总线地址相同，而处理器作为 EP 时这种映射关系失效，软件需要重新建立 [6]。

FTD2000/8 处理器的 inbound 映射使用 inbound 地址转换寄存器组将 PCIe 域地址转换为存储器域地址。inbound 地址转换表共有 8 个 table（table0~table7），每个 table 的 inbound 存储器域地址默认为 0，可通过配置 TRSL_ADDR 寄存器进行修改，完成 inbound 映射后，根节点可通过系统分配的地址对 EP 处理器进行读、写操作。基于这种存储空间映射的方式，对于处理器之间 PIO 或 DMA 方式的通信，软件只负责建立 EP 处理器端的 BAR 空间映射，其他数据传输工作均由 PCIe 硬件完成，带宽利用率较高。

FTD2000/8 处理器集成两个 PCIe 控制单元，可分为 RC 和 EP 模式。当设置为 RC 模式时，负责管理以该控制器为根节点的外部设备；设置为 EP 时，如果 PIO 请求流向该控制器，同时处理器的 outbound 寄存器组已完成存储器域到 PCIe 域地址的映射，则 EP 模式下的处理器能够通过 PIO 方式与其他处理器进行数据传输。EP 模式下的 PCIe 控制器具体配置过程如下。

（1）清零 RC 模式下 PCIe 控制器的 outbound 地址转换表。

（2）设置 EP 模式下 outbound 寄存器组，包括转换的源起始地址、目的起始地址以及翻译的空间大小等。

（3）通过配置 Cx_MEM_BASE_LIMIT 寄存器，设置 EP 模式下 PIO 请求的译码方式，即 PIO 请求流向哪个控制器。

3.4 DMA 驱动

直接存储访问可以降低传输对系统 CPU 资源的占用率，显著提高系统运行效率，达到更高的数据传输速率。FTD2000/8 处理器在 PCIe 控制器内部集成有 PCIe 相关的 DMA 通道，每个 PCIe 控制器中包含 2 个 DMA（direct memory access，DMA）通道（DMAChannel0 和 DMAChannel1）。该 DMA 通道可以实现处理器外部 PCIe 设备与处理器本地 DDR 存储器之间的 DMA 数据传输 [7]。

FTD2000/8 处理器 PCIe 接口通过“DMA 描述符 +DMA 描述符指针”方式来发起 DMA 数据传输，DMA 传输所需的地址和数据大小等信息均由主机端通过 PIO 方式写入 PCIe 空间映射的状态寄存器 BAR（base address register）中，另一 FTD2000/8 处理器等待主机端发起 DMA 操作，具体步骤如下。

（1）设置 PCIe 控制器地址映射窗口，将 FTD2000/8 处理器本地的某段地址空间映射到 PCIe 总线 Memory/IO 空间中的某个地址段（对应于外部 PCIe 设备）。DMA 数据传输一端为外部 PCIe 设备，另一端为本地 DDR 存储器。上述的 PCIe 控制器地址映射窗口的本地地址空间，即为 DMA 控制器访问外部 PCIe 设备的地址空间。

（2）在 DDR 存储器中，存放设置好的 DMA 描述符，DMA 描述符中包含 DMA 源地址、目的地址、传输长度等信息，其格式参见 FTD2000/8 处理器相关文档。

（3）向 PCIe 控制器的 REG_DMA_CH0/1_SP_H、REG_DMA_CH0/1_SP_L 寄存器（DMA 描述符指针）中，写入 DMA 描述符在 DDR 存储器中的存放基址。

（4）设置 PCIe 控制器的 REG_DMA_CH0/1_CTRL 寄存器，设置 DMA 传输方向，发起 DMA 数据传输。

实际应用中，可根据数据传输需要，在 DDR 存储器中存放多个预先设置好的 DMA 描述符，每次通过配置 REG_DMA_CH0/1_SP_H、REG_DMA_CH0/1_SP_L 寄存器来调用不同的 DMA 描述符。

3.5 MSI 中断配置

PIO 与 DMA 数据传输方式均可以采用轮询或中断的方式向处理器上报读写操作状态，轮询方式持续占用 CPU，消耗大量 CPU 的处理时间，效率较低。在程序设计时，采用 MSI-SPI 通路中断的方式向对端发送完成消息。具体设计分为两种：一种是 RC 为发送方，EP 为接收方。另一种是 RC 为接收方 EP 是发送方。在数据传输过程中，前者约定 EP 端 BAR0 的最后 64 kB 空间接收 MSI 中断消息，后者约定 RC 端 outbound 窗口最后 64 kB 空间接收 MSI 中断消息，发送方数据传输完成后，写对端中断消息寄存器来上报读写完成状态 [8]。

利用 BAR 或 outbound 地址空间的 MSI 中断与标准 ARM 体系的 MSI 中断相比，使用起来更加简单方便，而且不需要操作系统提供申请与管理向量号的复杂机制 [9]。配置过程为：接收方使能 MSI 中断控制器，将接收中断消息的基址写入 MSI64 高位与低位地址寄存器，最后使能 MSI-SPI 通路寄存器。发送方将数据通过存储器写 TLP 报文传输至接收方后，在 MSI64 寄存器存放的地址写数，接收方 PCIe 控制器收到 SPI 通路的 MSI 中断，上报至 CPU。

4 系统验证与测试

本节采用 FTD2000/8 处理器，重点进行 DDR 模块与 DMA 控制器模块的 PIO 与 DMA 读写测试，读写流程如图 3、图 4 所示。RC 处理器与 EP 处理器之间 PCIe 总线链路宽度为 x8，支持 PCIe3.0 规范。

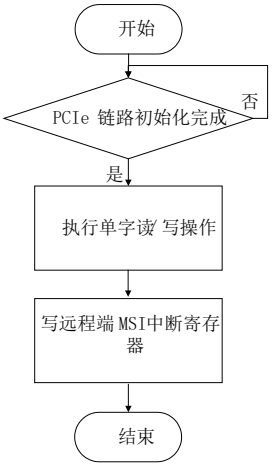

图 3 DMA 读 / 写流程图

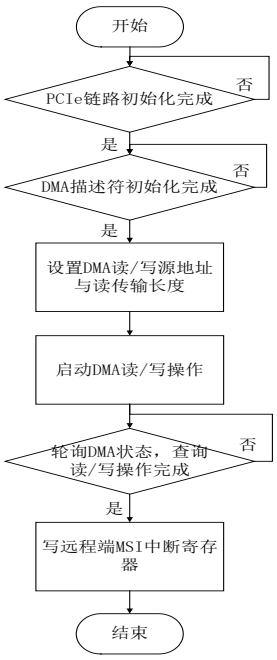

图 4 DMA 读 / 写流程图

通过 FTD2000/8 处理器的一路调试串口显示 PCIe 总线的相关信息。RC 端串口打印 PEU1_C0 与 PEU1_C1 控制器链路状态为 0x10（L0 态），代表链路训练成功，配置头类型 type1，表示该处理器为 RC 设备。EP 端串口打印 PEU0_C1 与 PEU1_C0 控制器链路状态为 0x10（L0 态），代表链路训练成功，PEU0_C1 控制器配置头类型为 type1，表示其为 RC 设备。PEU1_C0 控制器配置头类型为 type0，表示其为 EP 设备。

测试程序包含 RC 处理器与 EP 处理器采用 PIO 或 DMA 方式的读写，发送端与接收端初始化 PCIe 总线、初始化 SPI 通路的 MSI 中断。写远程端设备内存时，本地端将数据发送至对端，数据发送完后写 MSI 中断寄存器，通知远程端接收；读远程端设备内存时，本地端读取并写远程端 MSI 中断寄存器，通知远程端已读取完成。

实验结果分为两组，分别为主机端通过 PIO 方式写远程端内存、主机端通过 DMA 方式写远程端内存 [10]。读写数据量大小为 32 kB、64 kB、128 kB、256 kB、512 kB、1 MB，实验结果如表 1 所示。

表 1 PIO 与 DMA 传输速率表

数据量大小 /kB	PIO 读 /(MB·s^{-1})	DMA 读 /(MB·s^{-1})	PIO 写 /(MB·s^{-1})	DMA 写 /(MB·s^{-1})
32	1.12	2 695.33	6.35	3 124.99
64	1.12	3 509.30	6.35	4 184.10
128	1.12	4 430.20	6.35	5 196.13
256	1.12	4 523.19	6.35	5 706.13
512	1.12	5 003.14	6.35	6 016.54
1024	1.12	5 202.47	6.35	6 241.06

经实验测试，DMA 读写速度随数据量增大而提升，当数据量为 1 MB 时，DMA 读写速度分别为 5 202.47 MB/s、6241.06 MB/s，是理论速度 8 GB/s 的约 65% 与 30%。PIO 读写速度较 DMA 传输效率低，读速度为 1.12 MB/s，写速度 6.35MB/s。

5 结束语

本文基于 FTD2000/8 处理器实现了 PCIe 总线的深度扫描算法、窗口映射配置以及 PIO 与 DMA 读写机制。针对不同的数据量大小，测试了 PIO 与 DMA 读写的传输带宽。测试结果表明，PCIe 总线接口的多处理器数据传输技术能够满足嵌入式数据处理的实际需求。

参考文献：

[1] 王齐 . PCI Express 体系结构导读 [M]. 北京 : 机械工业出版社 , 2010.

[2] 毕城 , 元永红 . 基于 PCIe 总线的多处理器数据交换技术 [J]. 电子科技 ,2017,30(7):11-15.

[3] 杨佳丽 . 基于 PCIe 总线的 FPGA 与 PC 间数据传输系统设计 [J]. 微型电脑应用 ,2022,38(4):34-36.

[4] TANWAR P K, THAKUR O P, BHIMANI K, et al. Zynq SoC Based High Speed Data Transfer Using PCIe: A Device Driver Based Approach[C]//2017 14th IEEE India Council International Conference (INDICON).Piscataway:IEEE, 2017: 1-6.

[5] 张健 , 李跃鹏 , 刘威鹏 , 等 . 基于 VxWorks 的 PCIe 多路传输系统驱动设计 [J]. 电工技术 ,2023(17):173-175+179.

[6] 刘肖婷 . 基于 DWC PCIe Core 的数据传输系统设计 [J]. 铁路通信信号工程技术 ,2024,21(1):26-29+46.

[7] 刘佳宁 , 单伟 , 刘金鹏 .PCIe 总线 DMA 高速传输系统的设计与实现 [J]. 电子技术应用 ,2023,49(12):85-89.

[8] 唐雷雷 , 贺占庄 .PCI Express 总线中消息中断的研究 [J]. 微电子学与计算机 ,2013,30(7):137-140+144.

[9] 刘佳 . 基于 WinDriver 进行 PCIe 驱动开发和数据交互实现 [C]// 天津市电子学会 . 第三十六届中国（天津）2022IT、网络、信息技术、电子、仪器仪表创新学术会议论文集 . 天津：天津市电子学会 , 2022:51-54.

[10] MARKUSSEN J, KRISTIANSEN L B, STENSLAND H K, et al. Flexible device sharing in PCIe clusters using device lending[C]//Workshop Proceedings of the 47th International Conference on Parallel Processing. New York:ACM,2018: 1-10.

【作者简介】

曹月（1996—），女，陕西渭南人，硕士研究生，助理工程师，研究方向：嵌入式软件开发。

张奕然（1997—），女，湖北十堰人，硕士研究生，助理工程师，研究方向：网络信息安全。

徐锦涛（1991—），男，陕西宝鸡人，硕士研究生，助理工程师，研究方向：嵌入式软件研究。

（收稿日期：2024-03-05）