计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(4):219−225 [doi: 10.15888/j.cnki.csa.006286]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 FPGA 的拟态服务器设计①

崔冰萌, 倪　明, 凌幸华

(中国电子科技集团公司第三十二研究所, 上海 201800)
通讯作者: 崔冰萌, E-mail: 824458931@qq.com

摘　要: 随着计算机技术的不断发展, 传统架构下的 CPU 处理能力已无法应对日益多样化的计算处理任务, 新型异构计算体系也存在可提升的空间. 分析了以“应用决定结构, 结构决定效能”为理念, 基于多维重构函数化结构与动态多变体运行机制的拟态计算 (Mimicry Computing, MC) 体系架构, 利用 FPGA 硬件可编程、动态可重构和功耗低的特性, 设计了一种基于 FPGA 的拟态计算服务器, 并阐明了该服务器的核心电路设计与关键技术实现.

关键词: 拟态计算; FPGA; 多维重构

引用格式: 崔冰萌,倪明,凌幸华.基于 FPGA 的拟态服务器设计.计算机系统应用,2018,27(4):219–225. http://www.c-s-a.org.cn/1003-3254/6286.html

Design of Mimicry Computing Server with FPGA

CUI Bing-Meng, NI Ming, LING Xing-Hua

(The Thirty-Second Institute of Chinese Electronic Technology Company, Shanghai 201800, China)

Abstract: With the development of computer technology, the processing capability of CPU with traditional architecture cannot address the various computing tasks. There is still room for the performances of heterogeneous computing to be promoted. This study analyzes the idea of “applications make structures, structures make efficiencies”, clarifies the conception of mimicry computing, which is based on multidimensional reconfiguration function and dynamic changeable operation system, and then uses the characteristics of programmable hardware, dynamic reconfiguration, and low power of FPGA to design a mimicry computing server which is based on FPGA, and also utilizes the core circuit and critical technology of this mimicry computing server.

Key words: mimicry computing server; FPGA; multidimensional reconfiguration

1　引言

自第一台电子数字计算机问世以来, 计算机技术发展迅速, 计算机系统对设计灵活性、速度以及并行性方面提出了越来越高的要求. 在传统架构下, 尽管 CPU 编程灵活且在很多领域应用广泛, 但其计算能力的提升只能寄希望于器件设备的更新换代, 且现有器件技术的发展已经无法满足更大数量级的计算效能的需求.

在这样的现状下, 使用不同类型指令集和体系架构的计算单元组成系统的异构计算方式被提出. 异构计算作为一种特殊形式的并行计算方式, 能够根据每个计算子系统的结构特点为其分配不同的计算任务, 在提高服务器计算性能、能效比和计算实时性方面体现出了传统架构所不具备的优势, 因此异构计算技术是解决计算能效问题的重要手段[1]. 目前有关异构计算的研究主要是基于 CPU+GPU 的这种异构计算系统, 它在传统计算机系统中加入 GPU 作为加速部件并配合 CPU 共同承担计算任务, 但 GPU 高能耗、小缓存等特点限制了其能效的提升与应用范围[2]. 而且, 这样的体系结构更多的也只是系统软件程序上实现数据计算的优化, 不能真正实现硬件上更细粒度的设计和定制, 在硬件结构设计和其空间资源利用上也仍有提升

① 收稿时间: 2017-07-14; 修改时间: 2017-08-02; 采用时间: 2017-08-07; csa 在线出版时间: 2018-03-31

的空间. 在此基础上, 拟态计算作为一种基于认知的软硬件结构动态变化的计算架构, 除了部件级的结构组合之外, 还能在硬件层面实现动态重构, 以硬件适应算法、硬件定制和硬件并行的方式实现计算, 让计算效能的进一步提升成为可能.

2 拟态计算

2.1 拟态计算的定义

从体系结构创新入手, 邬江兴院士在 2014 年首次引入了“应用决定结构, 结构决定效能”的思想, 提出了基于多维重构函数化结构与动态多变体运行机制的拟态计算体系[3].

传统计算中的硬件执行体结构基本不变, 仅依靠软件算法的改进获取运算速度和计算效能的等效提高. 拟态计算在应用处理的过程中, 计算、存储、互联等执行结构都是随事务处理过程效能动态可变的. 对一个确定的可计算问题, 在拟态架构中可以由多种功能等价、计算效能不同的硬件结构和软件算法结合来实现. 即拟态计算是一种基于认知的主动重构计算技术 (Proactive Reconfigurable Computing Architecture, PRCA), 它是以实现高效能计算为目的, 在任务处理的全过程中通过感知自变量动态地选择或生成应用问题的最佳计算结构集合[4]. 计算结构的函数化是拟态计算的本质.

图 1 说明了 PRCA 的基本理念, 从图中可以看出, 拟态计算融合了通用计算和专用计算, 在这一点上, 它与现在流行的以 CPU+GPU 为基础的异构计算有着相似之处. 现在异构计算的高速发展已经证明了结构模式的调整对于计算效能的促进作用, 而拟态计算是在此基础上进行的更进一步的拓展. 拟态计算希望做到的是基于认知的主动重构, 即根据不同应用的要求, 通过分析应用的需求认知, 动态重构一个适应它的硬件系统和软件环境. 因此, 拟态计算架构不仅仅是对资源的按需调配, 其系统软件程序和硬件设计均具有编程重构的功能, 能够通过对这些可计算变体的动态选择和使用实现计算效能的最优化.

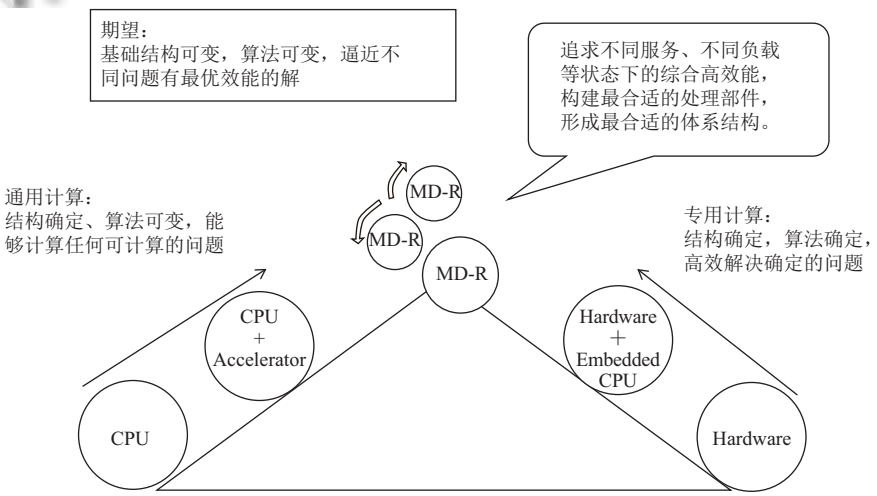


图 1 PRCA 的基本理念

2.2 拟态计算的架构说明

目前常用的用于研究和实现数据计算和分析的主要器件包括 CPU、GPU、DSP 和 FPGA.

(1) CPU

CPU 是传统计算机运行管理和计算的主要设备, 擅长复杂运算, 内部拥有丰富的指令集和计算资源, 相关的算法开发完善, 用户友好度高. 但是对于大量多发简单处理的数据, 运算能力相较并行设备有很大差距.

(2) GPU

GPU 具有超高的运算速度, 擅长图像处理、浮点计算和并行计算. 另外, OpenCL 的提出, 也为 GPU 相应的软件开发提供了便利. 但 GPU 开发也面临着负载均衡、功耗和任务划分等问题.

(3) DSP

DSP 是专用的数字信号处理工具, 对数字信号采集、处理的效率极高. 同时可通过汇编或高级语言进行编程, 实时实现方案[5]. 但受指令集的时钟周期的限制, DSP 适用于系统较低取样速率、低数据率的信号, 不能处理频率太高的信号.

(4) FPGA

FPGA 的集成度很高, 可以完成极其复杂的时序与组合逻辑电路功能. 一般情况下, FPGA 可以反复地编

程、擦除. 在不改变外围电路的情况下, 设计不同片内逻辑就能实现不同的电路功能. 同时新型 FPGA 内嵌 CPU 或 DSP 内核, 支持软硬件协同设计[6]. 但是, FPGA 更适合开发时序逻辑电路, 对于复杂算法的开发难度较大.

从图 2 PRCA 的计算部件架构图可以看出, 拟态计算的思想就是融合了 CPU、GPU、DSP、FPGA、HRCA(混合可重构计算阵列) 和 RIC(可重构互连网络) 等技术, 建立了由通用系统、专用系统和可认知的柔性结构共同构建的拟态计算系统. 其中通用系统指传统的 CPU 通用处理器, 用来处理基本的资源分配、任务调度等问题. 专用系统指 DSP、ASIC 等专用处理器, 用来完成特定计算任务. 可认知的柔性结构指 CPU+GPU、CPU+FPGA、HRCA 等混合异构计算模式, 用来分析应用需求, 并针对不同应用对计算资源的不同需求动态配置系统的软硬件架构. 以上 3 种系统融合来实现拟态计算基于指令、部件、系统级的深度重构, 保证计算效能最大化.

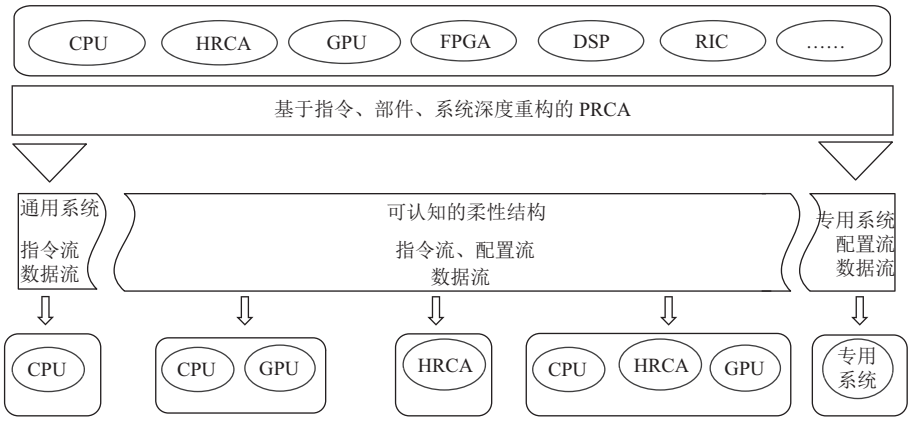


图 2 PRCA 的计算部件

3 拟态计算服务器的设计与实现

3.1 拟态计算服务器的整体框架设计

拟态计算架构基于应用驱动的变结构计算模式, 根据决策系统的指令来实现计算单元间的部件级重构, 以满足应用对计算资源、带宽资源、存储资源等的需求. 其中, 它的初步实现就是对于计算单元的指令级重构. FPGA 作为一种可编程的专用处理器, 其动态可重构技术能够在仅更新配置 FPGA 的一部分资源的同时, 保证内部未改变的部分继续工作. 通过对 FPGA 动态重新配置, 可以使它在不同的时间执行不同的功能. FPGA 的动态可重构技术可以用来实现系统的自诊断, 产生对不同运行环境重新配置的系统能力, 或者对给定的应用实现多用途的硬件[7], 也可以通过远程重构实现产品的升级和维护. 这些都能保证拟态计算服务器在资源调度、应用分析层面的重构需求. 除此之外, FPGA 可在不同逻辑下执行多个线程, 具有较强的并行处理能力[8], 且相较于 GPU 具有更低的功耗, 可实现更高效的计算加速. 因此, 针对拟态计算数字处理、数据计算、网络存储的需求, 本文设计了一种基于 FPGA 的拟态计算服务器.

拟态计算服务器硬件结构图如图 3 所示, 整机由 4 片大规模 XILINX 7 系列 FPGA 板、基于 ATOM 的控制单元、机箱管理模块 (BMC) 以及底板模块所组成. 核心板由 FPGA 7 系列芯片组成, 完成片内及片间的逻辑电路重构. 控制单元由 N455、NM10(桥片)、DDR3 内存条、SATA-SSD、网卡、USB 转 JATG 模块组成, 实现 FPGA 程序下载和远程管理, 另外通过 Open IPMI 与 BMC 的通信, 实现对机箱和电源信号的管理. BMC 模块采用以 Intel 的 Atom 处理器为核心的 CPU 功能模块作为上层控制/管理模块, 完成对硬件平台电源管理、温度监控、低功耗管理、机箱散热管理、SOL、时钟管理等硬件管理功能, 并提供给用户管理相关系统资源的界面. 同时, 底板模块为整个系统提供物理连接通路、时钟、电源, 以及各种 GPIO 控制信号.

拟态计算服务器采用 FPGA 全互联的方式实现高速高带宽数据通信, FPGA 两两之间都要相互配置互联接口. 如果单板的 FPGA 数目过多, 就会造成用于配置的接口数在 FPGA 板的占用比加大, 留给实际处理计算任务的可用资源变少, 同时相应产生的整板功耗也会加大. 因此单纯通过增加每个服务器的 FPGA 个数来提高计算规模, 可能会适得其反. 如图 4 所示, 在实际使用的过程中, 每一台服务器作为一个科学计算单元可以通过交换机与其他服务器进行互联, 组成一个更大的集群, 从而可以进行更大规模的计算任务.

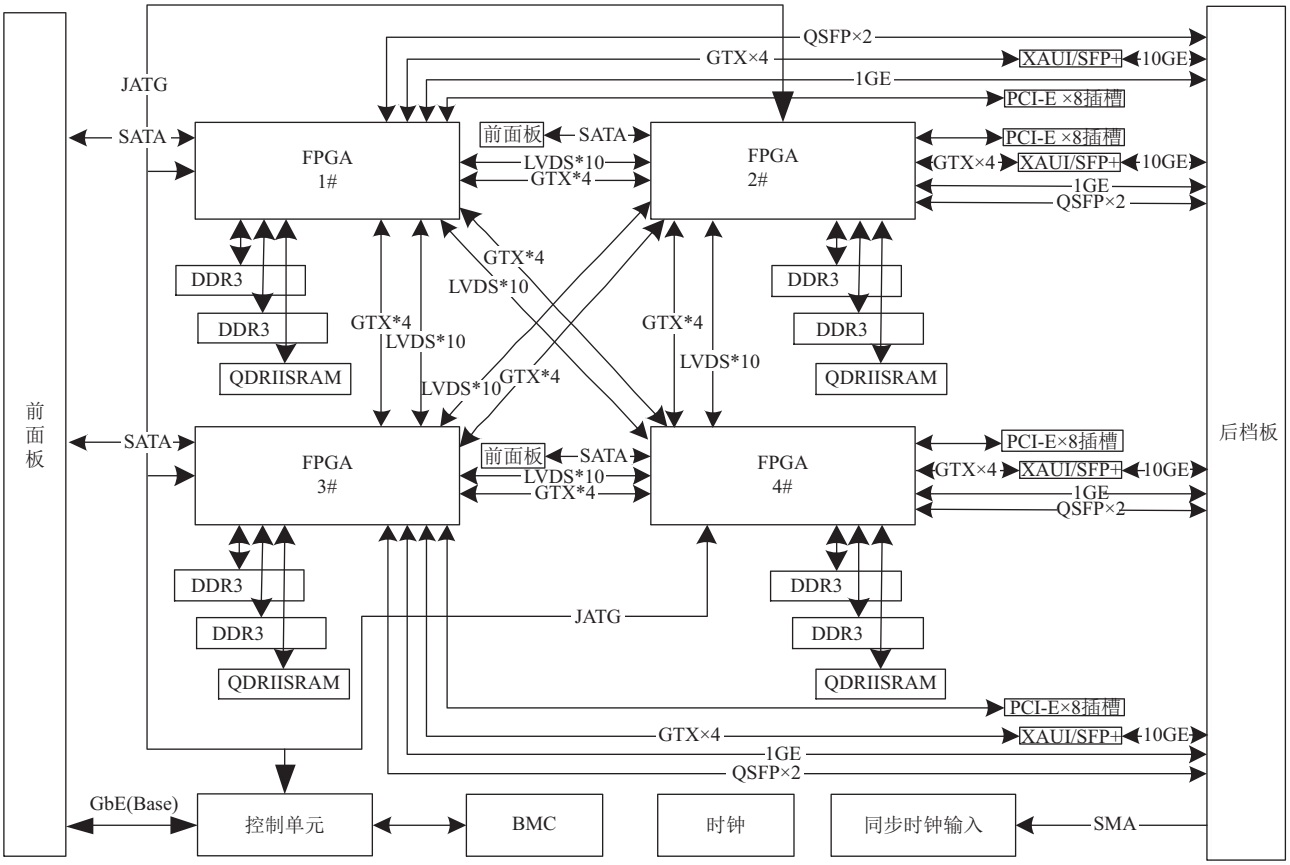


图 3 拟态计算服务器硬件结构图

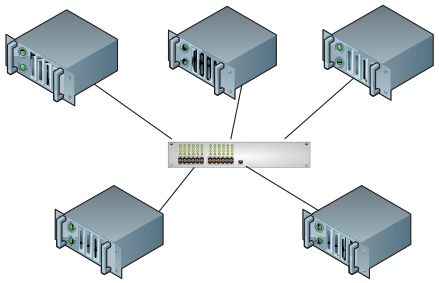


图 4 服务器互联方式

3.2 拟态计算服务器的核心板设计

FPGA 核心板总体框图如图 5 所示, 主芯片选择 Xilinx 公司的 Virtes-7 系列 FPGA 芯片, 其配置接口为 BPI, 配置 FLASH 为 PC28F00AG18FE, 通过并行接口, 实现对 FPGA 的配置.

该核心板通过两个高速连接器与底板相连, 高速连接器分别连接 GTX 信号、LVDS 信号和其他控制信号. 其中 GTX 信号包括: 1 路 SFP+, 2 路 QSFP, 1 路 PCIE, 一路 SATA 和三路用于 FPGA 核心模块间互连的 GTX 信号. LVDS 信号及相关控制信号包括电源及电源控制, SFP+, QSFP, PCIE 控制信号, 串口, I2C 信号等. 板载二路 DDR3 内存, 接口形式为 SODIMM. 板载千兆 PHY 芯片, 型号为 88E1111, 其通过 SGMII 总线与 FPGA 互通, 对外通过高速连接器连接到底板的 RJ45 接口. 板载温度传感器 LM84 及 FPGA 核心温度传感器 LM84, 通过 SMBus 总线互连, 可由底板 BMC 读取温度信息. 7 路时钟芯片 SI534 产生用于各个基于 GTX 应用所需要的时钟, WX501 提供一个单端全局时钟, WX703 提供一个差分全局时钟. 机箱采用 12V 为主供电电压, 通过在电源输入端串联保险丝和 PMOS 管分别实现对系统电源的过流保护和输入开关控制功能.

拟态服务器的核心部分采用这样的四块 7 系列的 FPGA 互联组成. 7 系列的 FPGA 提供超过 2×106 个逻辑单元, 收发器的线路速率达到 28.05 Gb/s, I/O 带宽达到 2.8 Tb/s, 与前几代的产品相比, 功耗下降达 50%, 能够在提升计算速度, 提供更多计算资源的同时, 降低了整版的系统功耗, 保证服务器实际使用时的计算资源需求和通信速度要求. 4 块 FPGA 分别拥有独立的内存通道, 搭载 2 块 DDR3 SDRAM 和一块 4 M 18 bit

QDRII SRAM 存储器, 其中单个 DDR3 支持的通量可达 8 G, 满足 FPGA 的存储需求. 4 个 FPGA 之间采用通过 GTX×4 和 LVDS×10 两套独立互联资源实现相互的全互联通信, 最高带宽可达 37 Gbps. FPGA 之间可以通过调用该高速接口来实现局部可重构网络以及实现对其它 FPGA 资源的访问和扩展.

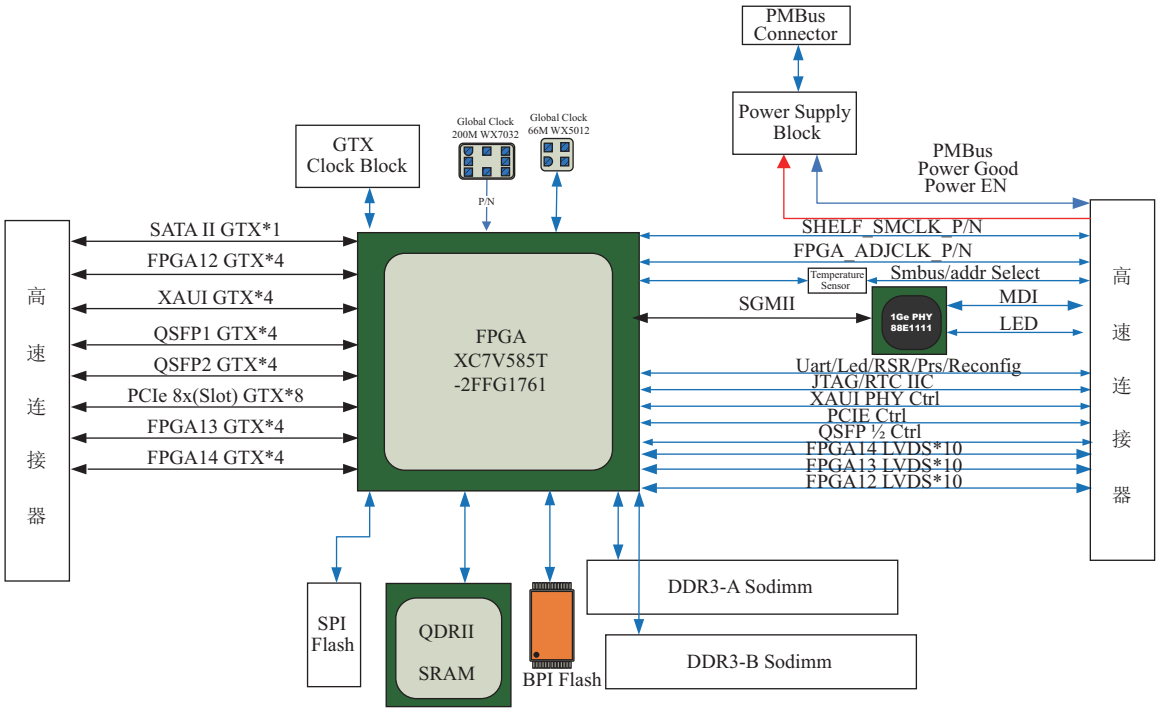


图 5 FPGA 核心板总体框图

3.3 基于 RapidIO 的 FPGA 互联实现

FPGA 互联是该拟态计算服务器的关键技术. 计算的任务分配、数据传送、资源共享等都需要通过 FPGA 互联资源高速处理完成. 在保证 FPGA 之间的通讯数据带宽的同时, 还需要与其他外设进行数据交互. RapidIO 技术主要面向于嵌入式应用, 其协议均有硬件实现, 能够实现高性能的点对点通信. 因此, 本文设计实现了基于 RapidIO 的 FPGA 互联, 并对其进行了仿真.

3.3.1 基于 RapidIO 的 FPGA 互联设计

RapidIO 总线技术是一种高性能、低引脚数、基于数据包交换的互连技术, 具有极低延迟和高带宽等特点, 主要应用于嵌入式系统内部互连, 支持芯片到芯片、板到板间的通讯. 从体系结构上 RapidIO 可以分为三层, 分别是逻辑层、传输层和物理层. 逻辑层界定了协议和包格式; 传输层规定 RapidIO 地址空间和在端点器件间传输包所需的路由信息; 物理层定义包传送机制、电气特性和信息流控制及低级错误管理等方式[9].

RapidIO 的接口模块设计[10]方案如图 6 所示. 设计将 RapidIO 的逻辑 (I/O) 和传输层整合为一个逻辑层模块 (Logic CORE), 用来完成与远端目标进行数据包收发的请求与响应. 另外还包括了物理层 (phy_wrapper) 模块、缓存器, 以及时钟、复位和配置访问等参考设计.

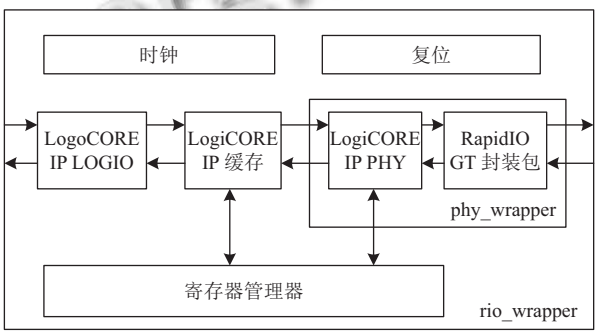


图 6 RapidIO 接口模块设计

逻辑层模块包含 3 类接口: 用户接口、链路接口和保持接口. 用户接口包含发起者响应 (Initiator Response)、目标请求 (Target Request)、发起者请求 (Initiator Request) 和目标响应 (Target Response) 四个端口, 用于与远端端点之间发送和接收请求和响应. 链路接口包含接收和发送两个端口, 用于连接 RapidIO 的物理层或作为缓存应用. 保持接口包含保持

请求/响应和寄存器配置两个端口, 用于将数据读写到任意由用户定义的或连接至物理层的配置寄存器[11].

物理层 (phy_wrapper) 模块接口提供两种配置模式: 单通道 (1x) 和 4 通道 (4x). 在数据带宽要求较低时使用单通道可以使占用的逻辑资源最少, 4 通道物理层模块实现要求的逻辑资源较多, 但可以提供更大的带宽和更高的可靠性, 而且可以通过训练从 4x 降低到 1x, 实现单通道下的操作[12].

3.3.2 基于 RapidIO 的 FPGA 互联的仿真实现

本次设计的 RapidIO 总线互联方式采用 4x 方式传输, 单路传输频率为 3.125 G 波特率, 输入时钟信号频率为 156.25 MHz.

仿真实现的是互联的两个 FPGA 之间基于 RapidIO 的逻辑层用户接口的数据收发. RapidIO 逻辑层模块的用户接口包含四个端口, 分别为: 发起者响应 (Initiator Response) 端口、目标请求 (Target Request) 端口、发起者请求 (Initiator Request) 端口和目标响应 (Target Response) 端口, 即 FPGA 的两方均可以作为发起方发送数据或作为目标方接收数据. 具体数据传输流程图如图 7 所示, 在仿真设计中是以 RapidIO(A) 作为发送方, RapidIO(B) 作为目标接收方进行数据收发的, 因此在图 7 中的 RapidIO(A) 的用户接口只包含发起者接口, RapidIO(B) 只包含目标接口.

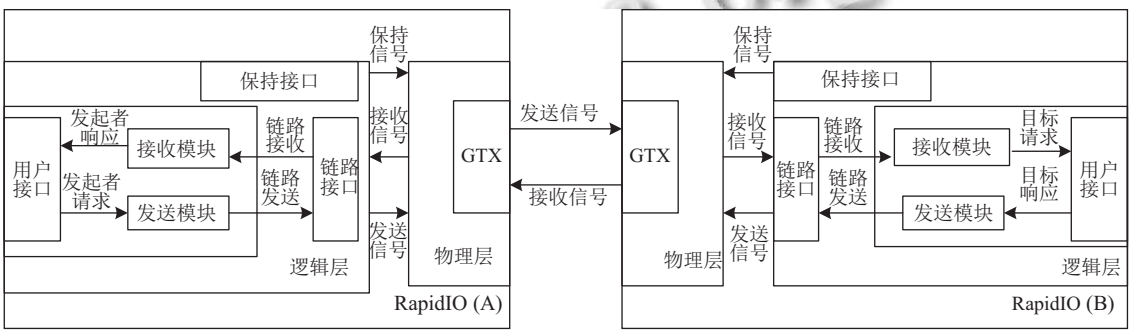


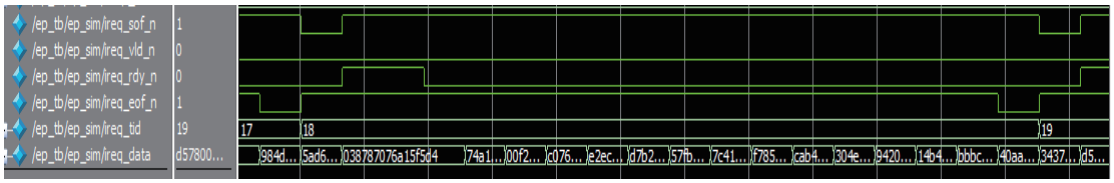
图 7 RapidIO(A) 数据收发流程

由图 7, RapidIO(A) 和 RapidIO(B) 分别代表数据的发起者 (Initiator) 和数据的目标接收者 (Target). 首先由发起者 RapidIO(A) 的用户接口的发起者请求 (Initiator Request) 端口通过发送模块将整合好的数据包发送, 由链路发送接口到达 RapidIO 的物理层, 物理层通过 4x 的差分信号由 GTX 收发器将信号发送到数据目标接收者 RapidIO(B). RapidIO(B) 物理层接收数据包后通过链路接收接口发送至 RapidIO(B) 的接收模块并由 RapidIO(B) 逻辑层的用户接口的目标请求 (Target Request) 端口接收.

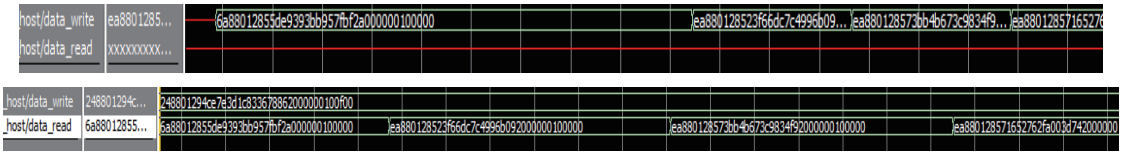
目标接收者 RapidIO(B) 将接收到的数据包通过用户接口的目标响应 (Target Response) 端口将收到的数据包经由 RapidIO(B) 的发送模块返回, 通过物理层和逻辑层链路接口后到达 RapidIO(A) 的逻辑层接收模块, 再由发起者响应 (Initiator Response) 端口输出数据包. 对比发送端的发起者请求 (Initiator Request) 端口和发起者响应 (Initiator Response) 端口的数据包, 即可判断数据包收发的正确性. 具体的仿真图像如图 8 所示.

图 8(a) 以发起者请求 (Initiator Request) 端口发送数据包为例分析控制信号与发送数据包的关系. 首先, 发送端通过 ireq_sof_n 和 ireq_vld_n 端口置位说明帧开始且数据有效. 在下一个时钟周期, ireq_rdy_n 端口置位, 表示端口已准备好接收发送来的数据包. ireq_rdy_n 和 ireq_vld_n 共同构成有效地数据环路, ireq_sof_n 的端口置位不仅用于标记一个新的数据包, 还用于验证端口头文件信息的有效性. 接下来, ireq_sof_n 置为高电平, 开始传输数据. 当 ireq_vld_n 置为高电平时, 暂停数据发送, 此时的数据环路作废. 当 ireq_vld_n 重新置位并伴随着 ireq_rdy_n 的信号置位, 数据继续传输. 当 ireq_eof_n 信号置位时表明数据包发送完成.

图 8(b) 是发起者请求 (Initiator Request) 端口和发起者响应 (Initiator Response) 端口的数据对比, 说明了数据传输的正确性. 由图 8(b), 首先发起者请求端口发送的数据包 data_write, 此时发起者响应端口还没有数据. 经过一段时间后发起者响应 (Initiator Response) 端口的收到之前由发起者请求端口发送数据包, data_read 与 data_write 数据一致, 说明经过了一段时间之后, 发送端的数据成功接收并反馈发起者响应 (Initiator Response) 端口, 即数据收发实现.



(a) 请求数据包控制信号与发送数据仿真



(b) 发起者的请求和响应数据包仿真

图 8 数据收发仿真图像

图 9 以文字的形成呈现了数据的对比情况, 其中 iresp 即为发起者响应 (Initiator Response) 端口. 经对比, 期望接收的数据与实际收到的结果一致, 因此输出 “所有响应接受正确 (All Responses received correctly)” 的标识, 即数据收发成功.

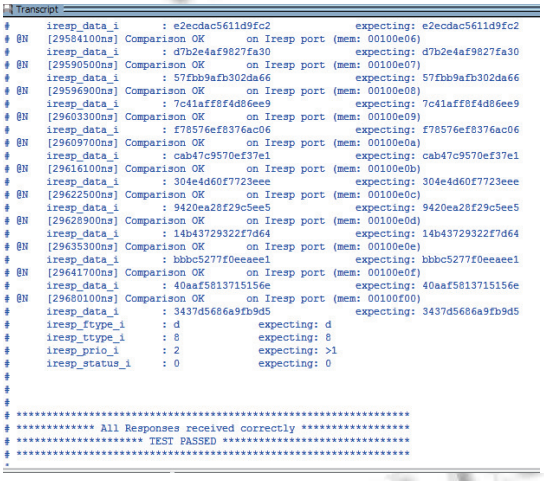


图 9 数据收发情况对比

4 结束语

FPGA 的互联设计仿真只实现了以 RapidIO 总线这一种方式, 在之后的开发中, 也将实现以其他高速总线作为连接方式的互联技术, 从而进一步提升拟态计算服务器的通用性.

在现有的 FPGA 可动态重构特性的基础上, 服务器的设计中将进一步研究 CPU+FPGA 的模式, 这样可以弥补 FPGA 在复杂算法开发难度大的问题, 能够使拟态计算服务器在兼顾复杂算法设计和资源配置管理的同时, 保证其良好的可编程性、高效的计算加速能力和低功耗等的优势. 最终服务器也将与 CPU、GPU 和 DSP 协作, 通过对应用和计算资源情况的合理分配, 发挥各个部件的优势协同工作, 以实现拟态计算基于认知的的部件级和系统级重构.

参考文献

1 Iyer R, Tullsen D. Heterogeneous computing. IEEE Micro, 2015, 35(4): 4–5. [doi: 10.1109/MM.2015.82]

2 Mittal S, Vetter JS. A survey of CPU-GPU heterogeneous computing techniques. ACM Computing Surveys (CSUR), 2015, 47(4): 69.

3 邬江兴. 拟态计算与拟态安全防御的原意和愿景. 电信科学, 2014, 30(7): 2–7.

4 成平广. 基于拟态计算的社会网络划分算法. 计算机科学, 2015, 42(8): 136–137.

5 刘荣. DSP 技术的现状及发展. 电子技术应用, 1999, 25(4): 7–10.

6 EDA 先锋工作室, 王城, 蔡海宁, 等. Altera FPGA/CPLD 设计 (基础篇). 2 版. 北京: 人民邮电出版社, 2011.

7 周盛雨. 基于 FPGA 的动态部分重构系统实现[博士学位论文]. 北京: 中国科学院研究生院 (空间科学与应用研究中心), 2007.

8 Afonso G, Baklouti Z, Duvivier D, *et al*. Heterogeneous CPU/FPGA reconfigurable computing system for avionic test application. Proceedings of the 27th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW). Cambridge, MA, USA. 2013. 260–267.

9 高毅, 刘永强, 梁小虎. 基于串行 RapidIO 协议的包交换模块的设计与实现. 航空计算技术, 2010, 40(3): 123–126.

10 刘东华. Xilinx 系列 FPGA 芯片 IP 核详解. 北京: 电子工业出版, 2013.

11 LogiCORE IP serial RapidIO user guide. 64–66. http://tec.icbuy.com/uploads/2010/5/20/srio_gsg247.pdf

12 LogiCORE IP serial RapidIO product specification. 1–2.