

基于 RapidIO 的多 DSP 互联仿真实现^①



蔡恒雨, 凤维杰, 丁上义, 卢茂辉, 郑启龙

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

通讯作者: 蔡恒雨, E-mail: chy520@mail.ustc.edu.cn

摘要: RapidIO 协议作为数据通信协议之一在嵌入式系统开发中具有重要作用, 适合短距离, 需要多处理单元合作的应用场景, 例如多 DSP 构成的板卡系统. BWDSP 芯片作为一款高性能数字信号处理器, 其丰富的计算资源在雷达信号处理等领域具有重要潜力. 硬件设计开发中, 直接采用已有数据通信协议很难适配具体硬件资源导致最终产品的数据传输性能较低. 因此需要结合具体硬件模型, 进行数据通信交换模型仿真建模设计, 提高数据传输效率. 本文首先介绍了 RapidIO 协议和 BWDSP 体系架构, 然后设计了基于 SystemC 语言的串行 RapidIO 交换模型, 最后设计实现了 BWDSP 虚拟平台. 本文设计的 BWDSP 虚拟平台功能符合实际 RapidIO 协议标准, 对硬件产品开发具有一定指导意义.

关键词: RapidIO; BWDSP; SystemC; 交换模型; 虚拟平台

引用格式: 蔡恒雨, 凤维杰, 丁上义, 卢茂辉, 郑启龙. 基于 RapidIO 的多 DSP 互联仿真实现. 计算机系统应用, 2020, 29(7): 95-102. <http://www.c-s-a.org.cn/1003-3254/7474.html>

Realization of Multi DSP Interconnection Simulation Based on RapidIO

CAI Heng-Yu, FENG Wei-Jie, DING Shang-Yi, LU Mao-Hui, ZHENG Qi-Long

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: RapidIO protocol, as one of the data communication protocols, plays an important role in the development of embedded systems. It is suitable for short-distance application scenarios that require cooperation of multiple processing units, such as the board card system composed of multiple DSPs. As a high-performance digital signal processor, BWDSP chip has great potential in radar signal processing. In the hardware design and development, it is difficult to adapt the specific hardware resources directly using the existing data communication protocol, resulting in the low data transmission performance of the final product. Therefore, it is necessary to combine the specific hardware model to design the simulation model of data communication exchange model, so as to improve the efficiency of data transmission. This study first introduces RapidIO protocol and BWDSP architecture, then designs a serial RapidIO exchange model based on SystemC language, and finally designs and implements BWDSP virtual platform. The functions of BWDSP virtual platform designed in this study are in line with the actual RapidIO protocol standard, which has a certain guiding significance for the development of hardware products.

Key words: RapidIO; BWDSP; SystemC; exchange model; virtual platform

近年来, 高性能的嵌入式系统层出不穷, 例如 DSP, FPGA 和 ARM 等众多硬件系统都得到了快速发展. 伴

随着嵌入式系统的发展, 分布式计算系统以及数据交换协议和交换芯片也进入了快速发展期, 如 RapidIO

① 基金项目: 国家核高基重大专项 (2012ZX01034-001-001)

Foundation item: Core Electronic Devices, High-end Generic Chips, and Basic Software National Science and Technology Major Project (2012ZX01034-001-001)

收稿时间: 2019-11-29; 修改时间: 2019-12-27; 采用时间: 2020-01-02; csa 在线出版时间: 2020-07-03

协议和以太网协议等在各种数据互联系统中发挥着重要的作用。

RapidIO 协议具有传输速率快、基于硬件进行处理和低成本等特点。它主要在嵌入式系统^[1]的内部互联中发挥重要作用,可以实现芯片和芯片、板卡和板卡之间的相互连接,如用于 DSP 之间和 DSP 板卡之间的互联并进行数据通信。IDT 公司生产的交换芯片 CPS1848^[2]支持 RapidIO2.0 标准,有 48 个双向的数据传输通路,支持 1x、2x 或 4x 三种端口模式,每个端口的速率可以设置为不同速率。

随着硬件设计的发展,软件模拟硬件实现的仿真建模技术也得到快速发展,例如时钟精确级的模拟和功能级的模拟等。SystemC^[3]是一种新的系统建模语言,其目的是为系统级设计提供一种单一的语言,通过提供一个公共平台来更好地进行软硬件协同设计和验证。其中集成的 TLM2.0 作为传输模型建模标准,定义了通用数据单元 payload,十分适合进行 RapidIO 交换模型^[4]的功能级模拟。中国电子科技集团第三十八研究所设计研发的 BWDSP 芯片^[5]是一款高性能数字信号处理器,具有丰富的计算资源,可以处理计算密集型任务。故本文基于 SystemC 语言,设计了用于多个 BWDSP 之间互联的 RapidIO 交换模型,并且设计实现了 BWDSP 芯片构成的虚拟板卡平台,使 BWDSP 芯片模型之间可以进行高效的数据传输。

本文的主要工作如下:基于 RapidIO 协议特点和 BWDSP 体系结构特点,设计了基于 SystemC 语言的 RapidIO 交换模型,并且设计了多 BWDSP 芯片模型构成的虚拟板卡平台,对虚拟平台的性能测试结果进行分析。本文第 1 节介绍了 RapidIO 协议和 BWDSP 的体系结构;第 2 节介绍了 BWDSP 虚拟平台的整体设计以及 RapidIO 交换模型的设计,并且分析了 BWDSP 虚拟平台的工作流程和性能指标;第 3 节对 BWDSP 虚拟平台的性能测试结果进行分析;第 4 节是总结和展望。

1 RapidIO 协议与 BWDSP 体系结构

1.1 RapidIO 协议

RapidIO 协议作为一种快速数据交换协议,具有总线频率高、延迟低和可靠性高等优点,它可以为未来高性能的嵌入式系统发展提供数据交换和通信机制。RapidIO 协议包括逻辑层、传输层和物理层 3 层^[6]。其

中逻辑层定义了 RapidIO 组件使用的读写和消息传递语义;传输层定义了 RapidIO 数据包如何通过 RapidIO 结构进行路由;物理层定义了 RapidIO 链路的电路编码和电气特性。RapidIO 架构主要由两种设备组成:端点设备和交换机设备。

RapidIO 协议是一种基于数据包交换的互联体系标准^[7],总体来说定义了请求包和响应包两种类型的包,主要定义了维护包、读写包和消息包等数据包结构,其中消息包又可以分为邮箱包和门铃包。RapidIO 规范中定义了 8 种类型的数据包,其类型如表 1 所示。

表 1 RapidIO 数据包类型

包类型	Ftype(二进制)
请求包	0010
写操作	0101
流写包	0110
维护包	1000
流操作	1001
门铃包	1010
消息包	1011
响应包	1101

RapidIO 的事务是通过数据包的传输来完成的^[8]。在每一个事务中,发起方发出请求并产生相应的事务请求包,事务请求包通过交换设备到达接收方,在事务请求包收到来自接收方的响应后才会开始执行事务。RapidIO 事务包在两个设备间传递时,需要及时进行响应确认,类似采用握手协议实现同步控制。RapidIO 读响应包处理流程如图 1 所示。

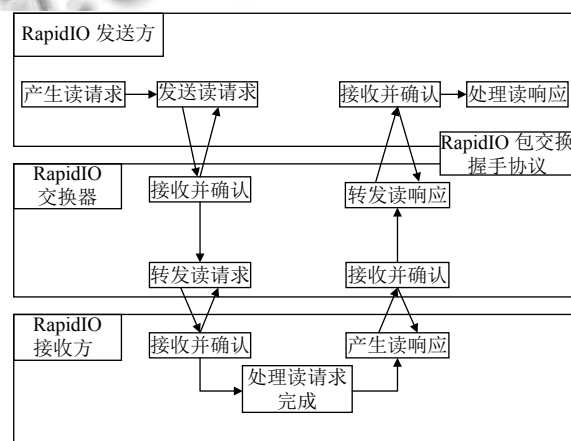


图 1 RapidIO 读响应事务处理流程

1.2 BWDSP 体系结构

本文所采用的 BWDSP 模型是由中国电子科技集

团第三十八所研制的 BWDSP 系列处理器, 可以对雷达信号进行快速处理^[9], 是一种分簇架构设计的 32 位 DSP 处理器. 它同时兼容 16 位和 32 位的定点数据格式, 指令槽能够同时发射 16 条指令, 并使用 SIMD 和 VLIW 架构. BWDSP 芯片的读数据和写数据总线位宽不同, 其中读数据位宽为 512 位, 写数据位宽为 216 位. BWDSP 的最大工作主频为 500 MHz, 指令周期为 2 ns, 同时内部有 11 级的流水线. BWDSP 体系结构中具有 4 个通用执行宏单元 (计算簇)、3 个地址产生器 (U, V, W) 和 3 个内存块 (Block0, Block1, Block2), 每个执行宏间是彼此独立的高速链路口, 且每个执行宏可向其他 3 个执行宏传输数据. 每个计算簇与内存块、地址产生器间都通过总线相互连接. BWDSP 的体系结构图如图 2 所示.

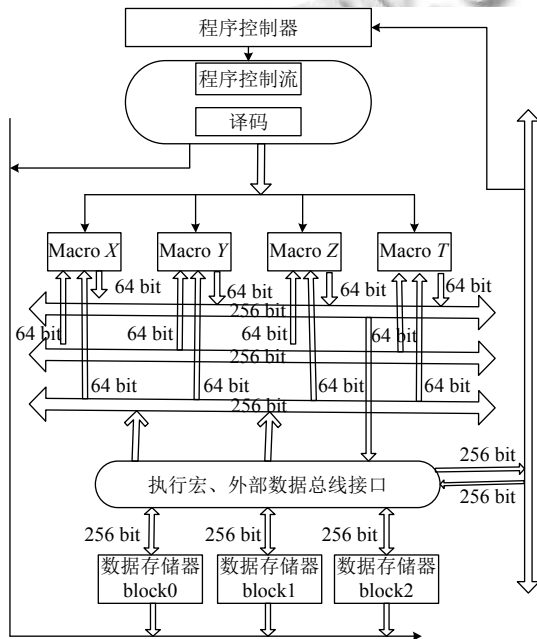


图 2 BWDSP 体系结构

2 BWDSP 虚拟平台设计

BWDSP 板卡系统整体的设计是采取 4 片 BWDSP 模型与 RapidIO 交换机模型的 4 个端口相连接, 将交换机剩余的一个端口互联其他交换机模型或者外部存储模型, 如图 3 所示. 外部存储器模型可以是互联 DDR 等快速外设, 实现 DSP 对存储器中信息的快速读写. 外部存储器的地址空间映射到 DSP 地址空间中的某一块区域^[10], 然后通过地址转换映射机制实现对外设中信息的读写, 通常使用 DMA 方式进行读写.

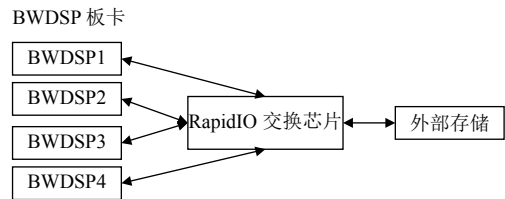


图 3 BWDSP 板卡结构图

2.1 BWDSP 虚拟平台数据流动

BWDSP 板卡系统中, 4 个 BWDSP 模型可以通过 switch 模型相互发送数据包, 数据包经过 switch 模型解析后会发送到正确的接收方 BWDSP 模型中. 由于在 switch 模型的各个端口设计了缓冲区, 各 BWDSP 模型同时发送数据时不会产生冲突阻塞问题. 当系统中数据流量在一定范围内, switch 模型中不会发生缓存区溢出问题, 可以保持高效的数据转发速率^[11].

在 BWDSP 虚拟平台中, RapidIO 数据包是嵌入在 TLM2.0 标准定义的 payload 中进行传输的. RapidIO 数据包在虚拟平台中的流动过程主要有如下几步:

- (1) BWDSP 模型根据事务要求生成相应的 RapidIO 数据包.
- (2) RapidIO 数据包会被包装进 payload 中的数据属性中, 并在 payload 的扩展字段设置相应命令.
- (3) Payload 通过 BWDSP 模型的 tlm_socket 接口发出, 然后等待响应信息.
- (4) RapidIO 交换模型收到 payload, 并将其存入内部的 buffer 缓冲区中, 采取存储转发模式, 同时向发送方响应交换模型收到该数据包.
- (5) RapidIO 交换模型从 buffer 中取出一个 payload, 并从中解析出 RapidIO 数据包部分. 根据 RapidIO 数据包的类型和目标 ID 进行处理或转发.
- (6) 如果该数据包有 RapidIO 模型进行处理, 则不需要转发操作. 否则, RapidIO 交换模型根据目标 ID 查找路由表, 定位到某个出站接口.
- (7) RapidIO 交换模型从选择的出站接口转发该 payload, 并等待下一个设备的响应确定是否从 buffer 缓冲区删除该 payload.
- (8) BWDSP 模型接收到该 payload 后, 解析出其中的 RapidIO 数据包, 并根据事务要求进行处理, 再根据 RapidIO 数据包中的源设备 ID 进行响应, 发送一个响应数据包给发送方.

2.2 BWDSP 虚拟平台功能特性

BWDSP 芯片最大工作主频 500 MHz, 指令周期 2 ns, 读总线位宽 512 bit, 写总线位宽 256 bit, 理论上读数据速率可以达到 200 Gbs, 写数据速率可以达到 100 Gbs, 完全可以处理 RapidIO 标准所定义的数据传输速率. 故 BWDSP 虚拟平台中 RapidIO 交换模型的数据处理速率应该达到 RapidIO 规范标准, 数据传输速率达到 1.25 Gbs、2.5 Gbs 和 3.125 Gbs, 各个 BWDSP 模型之间可以进行快速通信.

BWDSP 模型发送和接收的 RapidIO 数据包长度为 len 字节, 时钟频率为 bw_fre , 总线的位宽 $bandwidth$ 位. RapidIO 交换模型的时钟频率 rio_fre , 传输速率为 rio_speed , 该模型每个时钟周期可以处理一个完整的 RapidIO 数据包. 理论上当 BWDSP 模型与 RapidIO 交换模型协同工作时, rio_fre 和 bw_fre 应该满足式 (1), rio_speed 应该满足式 (2).

$$rio_fre = \frac{bw_fre \times bandwidth}{len \times 8} \quad (1)$$

$$rio_speed = rio_fre \times len \times 8 \quad (2)$$

2.3 RapidIO 交换模型设计

本文设计的 BWDSP 虚拟板卡中也含有 4 个 BWDSP 模型和一个 RapidIO 交换模型, RapidIO 交换模型除了用于板内通信的接口以外, 还有一个离开板卡的接口来和外部板卡之间进行通信. 根据上述原则, 可设计如图 4 所示的 switch 功能模块, 一共有 5 个接口, 每个接口都有一个路由查询表和相应的入站/出站模块. 其中, 0 号接口负责外部板卡和存储设备的通信, 第 1、2、3 和 4 号接口分别与板卡内部的 4 个 BWDSP 芯片模型相连.

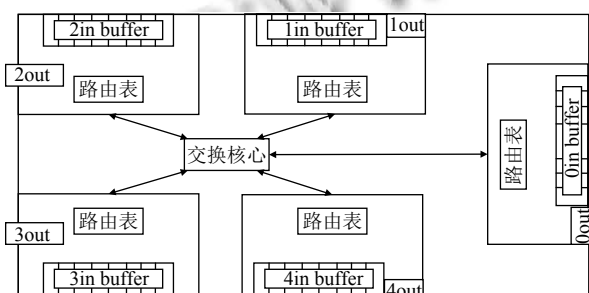


图 4 RapidIO 交换模型结构图

RapidIO 交换模型的工作方式为存储转发模式, switch 的设计中为每个端口保留一个缓冲队列, 设计

为 8 个缓冲单元, 每个 switch 模型有 5 个端口, 按每个缓存单元为 RapidIO 数据包最大长度 276 字节计算, 为 $276 \times 8 \times 5$ 字节. 该类初始化时通过参数设置来初始化其各个端口, 并且初始化各端口相连的 buffer 缓冲区. 同时设计路由表基类 `Route_table` 类作为 switch 类的模板参数, 在 switch 模型初始化的同时实例化特定的路由表, 使 switch 类的设计具有一定的扩展性, 可以兼容新的路由表结构设计.

路由表的基本工作原理如下: 从输入分组中提取目的地 ID 字段并将其与可编程路由表中的值进行比较并转发. 本系统的 RapidIO 交换机为每个交换机端口设计了一个单独的路由表, 因此每个接口都可以不同地路由数据包, 根据数据包的目标 ID 字段中的值与路由表中编程的值之间的比较, 将传入数据包路由到其目标接口. 数据包可以路由到以下目的地之一: 单播到某一接口, 终端设备或者 switch 设备; 多播到多个接口^[12].

2.4 BWDSP 虚拟平台初始化

BWDSP 虚拟平台初始化时, 首先需要进行 RapidIO 系统的初始化, 该阶段有一个初始化枚举探测阶段, 用于探测整个系统的拓扑结构并且分配终端设备 ID. 该功能主要由系统中一个或多个终端设备加载系统初始化驱动程序代码, 作为主设备开始执行系统探测功能, 本设计中将初始化设备 ID 分配为 0 的终端节点当作为主设备开始执行系统枚举探测. 系统中各 switch 路由表的初始化有静态配置和动态配置方式^[13]. 静态配置通常是通过系统上电初始化 I2C 等方式进行配置, 动态配置则是在系统枚举探测时, 分配各个设备 ID 的同时将路由信息写入 RapidIO 交换模型的路由表中, 同时设置默认路由端口信息.

RapidIO 的系统初始化探测算法采用类似于深度优先的图搜索算法^[14,15], 在搜索过程中会形成一颗搜索树, 该搜索树即可代表整个系统设备互联的拓扑结构图. RapidIO 系统初始化探测算法如算法 1 所示, 其中 `rioEnumerateNeighbor` 函数实现深度优先搜索部分.

算法 1. RapidIO 系统初始化探测算法

输入: BWDSP 芯片和 RapidIO 交换模型组成的 BWDSP 虚拟平台系统
输出: BWDSP 虚拟平台系统拓扑结构和各交换模型路由表信息

1) 初始化时 `global_id=0, hop_count=0`, 所有终端设备的 `base_id` 设为 `0xff`

选择某片 BWDSP 模型作为主设备, 分配 `base_id` 为 `0x00`

由该主设备执行 `rioEnumerateNeighbor(hop_count)` 算法

2) `rioEnumerateNeighbor(hop_count)` 算法流程:

发送跳数为 `hop_count` 的初始化维护数据包, 然后等待响应信息 (调用 `read_information()` 函数实现)

if 收到终端节点的响应, 且该终端节点未被锁定

`global_id++`

分配 `global_id` 为该终端设备的 `base_id` (调用 `write_information()` 函数实现)

算法返回

else if 收到终端节点的响应, 且该终端节点已经被锁定

则该终端设备已经分配 `base_id`, 放弃分配

算法返回

else if 收到 switch 节点发回的响应信息

提取响应信息数据负载中 switch 端口数总数 `port_total` 和与该设备相连的 switch 端口信息 `port_number`

通过一个 `for` 循环将所有已检测到的设备 `id` 即 `0-global_id` 和 `port_number` 作为路由信息写入该响应 switch (调用 `write_information()` 函数实现)

while(`port_total-1`) //将 `port_number` 端口去掉, 防止重复检测

记录此时检测的 switch 端口号为 `port_numvar`

递归执行 `rioEnumerateNeighbor(hop_count+1)` 算法, 并等待算法返回结果

将新检测到的所有设备 `id` 信息和 `port_numvar` 作为路由信息写入该 switch 中

算法返回

假设该系统中每个 BWDSP 板卡结构都如图 3 所示, RapidIO 交换模型用于互联各 BWDSP 板卡, 系统中有 N 个 BWDSP 芯片和 M 个 RapidIO 交换模型, 其中 N 为 4 的整数倍, N 和 M 的关系近似为 $N=3\times M$. 采用单主机进行系统初始化, 则主机到各终端设备路径长度近似为 $\log_4 N$. 需要读取一次设备信息, 并且写一次设备 ID 信息到终端设备中, 同时写一次路由信息到相应的 RapidIO 交换模型中, 故时间上限为 $3\times(N-1)\times\log_4 N$, 时间复杂度为 $O(N\times\log_4 N)$. 每次发送数据包都是由主机进行发送的, 所以需要的存储量为每个 RapidIO 交换模型保留一个 RapidIO 数据包.

2.5 BWDSP 虚拟平台工作流程

图 5 是 BWDSP 虚拟平台运行流程图, 从系统启动时 RapidIO 系统初始化探测枚举阶段开始, 到系统正常工作时 BWDSP 模型之间通过 switch 模型进行相互通信. 图 5 是以 RapidIO 交换模型的工作视角为主体进行展示的, 故其工作流程是对 RapidIO 数据包的解析和转发, 同时对寄存器读写和路由表维护信息进行简单展示.

BWDSP 虚拟平台启动后^[16], 首先会初始化终端设备和交换机设备中的各个寄存器, 并且选择其中一个终端设备将其 `base_id` 设为 0, 使其成为主设备. 然后,

由主设备开始进行系统初始化阶段的探测枚举阶段, 该阶段主要是探测并获得整个系统的网络拓扑结构, 并且给各 switch 中的路由表动态配置路由信息. 当整个系统初始化阶段结束以后, 各个 BWDSP 模型之间可以相互发送信息, 通过 RapidIO 交换模型进行数据的交互, 协同完成某项工作. 系统初始化结束之后, 主设备拥有整个系统的拓扑结构信息, 可以提取该信息并保存.

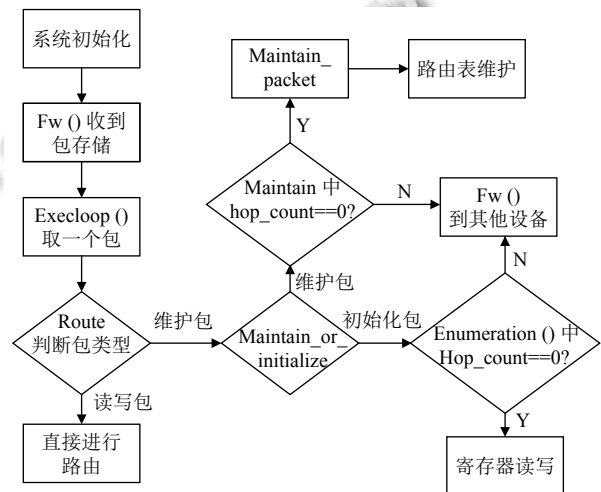


图 5 BWDSP 虚拟平台运行过程

3 BWDSP 虚拟平台实验和分析

3.1 系统传输速率分析

对普通数据读写包的发送进行测试, 将数据发送设备和交换机设备的频率都设置为 2 MHz, 数据接收 BWDSP 保持在正常的 500 MHz 工作频率下. 根据上文式 (1) 和式 (2), 当 RapidIO 数据包长度取 276 字节时, RapidIO 交换模型在 1 s 内转发处理的数据字节数为 $(256+20)\times 2$ MB, 速率近似为 4.416 Gbs. 然后又测试了发送不同 RapidIO 数据包长度时, 系统整体的传输速率, 如数据字节数为 $(64+20)\times 2$ MB 时, switch 模型的速率近似为 1.344 Gbs. 实验结果表明, 该 BWDSP 虚拟平台中交换模型的数据传输速率符合 RapidIO 规范标准中的传输速率. 表 2 是在不同数据包长度下, RapidIO 交换模型的传输速率测试结果.

表 2 中第 1 条和第 2 条测试数据中数据包长度字节为 36 B 和 52 B 时, 发送设备采用固定的发送频率进行数据发送. 此时交换设备和接收设备无法充分利

用信道和计算资源,数据传输速率分别是 0.576 Gbs 和 0.832 Gbs.

表 2 数据传输速率表

数据包长度字节 (B)	传输速率 (Gbs)
36	0.576
52	0.832
84	1.344
146	2.33
276	4.416

3.2 系统初始化阶段分析

RapidIO 系统初始化阶段主要是进行系统的探测枚举,即枚举整个系统中的所有设备,并分配终端设备 base_id 和配置各交换机路由表.本文在设计时对 RapidIO 初始化阶段数据包采取直通转发,这是为了加快系统初始化,故我们采取对传输数据包的长度和个数进行统计,并近似算出系统初始化阶段时间,与理论时间进行对比分析.设 BWDSP 虚拟平台中 BWDSP 模型数为 bw_num 个,switch 模型数为 sw_num 个,初始化阶段发送数据包数为 pkt_num 个.则系统初始化时间下限为 t_lower ,上限为 t_high ,下限和上限分别符合式 (3) 和式 (4).

$$t_lower = pkt_num \times \frac{1}{rio_fre} \quad (3)$$

$$t_high = (pkt_num + sw_num) \times \frac{1}{rio_fre} \quad (4)$$

系统初始化枚举探测阶段执行完成时,通过打印输出来检查各 switch 模型的路由表,与系统理论拓扑结构保持一致,说明系统初始化阶段枚举探测算法正常工作.表 3 所示是在不同系统拓扑结构下,BWDSP 和交换机模型的频率都设置为 2 MHz 时,BWDSP 系统初始化时发送的数据包个数.

表 3 初始化数据包数

BWDSP 模型数	switch 模型数	初始数据包数
6	3	109
11	3	185
17	5	343
20	6	434
23	7	527
23	7	533

表 3 中 BWDSP 模型数为 20 和 switch 模型数为 6 进行分析,系统发送的 RapidIO 数据包的数量为 434,系统时钟频率为 2 MHz,发送数据包时间为 217 us.对

于最后一个发送数据包被接收时,理论上所经过的 switch 模型个数少于 BWDSP 模型数,故系统初始化阶段需要时间下限为 217 us,上限为 227 us.当系统拓扑结构更大时,初始化阶段发送的数据包数量也会相应增加,上述时间计算公式依然适用.

表 3 中第 5 条和第 6 条实验数据具有相同 BWDSP 模型数和 switch 模型数,但是系统拓扑结构互联方式不同,导致系统初始化时需要的数据包数量也不相同.在两种系统拓扑结构下,RapidIO 初始化功能都可以正确执行,设计的 BWDSP 虚拟平台具有一定的可扩展性.

3.3 系统功能正确性分析

中国电子科技集团第三十八研究所设计了用于软硬件模型开发和调试的 CES 集成开发环境,本文设计的 BWDSP 虚拟板卡平台使用三十八所的 CES 集成开发环境进行调试.BWDSP 主设备模型会向 RapidIO 交换模型相连的存储模型中指定位置写入数据,该数据包会通过 switch 模型进行转发,然后检查该位置数据是否写入正确,以此验证 RapidIO 交换模型能否对数据包进行正确转发.实验中 BWDSP 模型向存储位置为 0x012003001 位置写入 0x80200401,然后逐个位置写入加 1 后的值,测试结果如图 6 所示.该测试结果表明该 RapidIO 交换模型可以正常工作,将 RapidIO 数据包转发到正确设备和地址并进行处理.

图 6 数据传输测试结果

RapidIO 协议中其他的数据包类型如维护数据包等,都进行了相应的数据通信测试^[17],并且测试结果正确,符合 RapidIO 规范标准.如路由信息维护数据包的测试结果,成功将 switch 模型中的路由信息修改为维护数据包中数据负载部分,正确实现了 switch 模型的路由表信息动态维护功能.

3.4 系统延迟性分析

BWDSP 虚拟平台正常工作时,即各个 BWDSP 模型之间进行数据传输时,通过实验测量数据传输延迟时间,对 BWDSP 虚拟平台传输性能进行分析.

表 4 中的数据包长度为 276 字节,传播时延信息是发送方结束发送时刻到接收方开始接收时刻之间的

时间,单位为 us. 各个 BWDSP 芯片同时通过 RapidIO 交换模型进行数据发送,并且是随机发往其他设备,故传播时延并非是单个数据包传播时延乘以包的数量. 其中第 5 条和第 6 条实验数据的传播时延也说明了系统的拓扑结构会影响 BWDSP 芯片之间的数据传输. 结合终端设备频率和读写位宽信息,可以得知数据发送时间和接收时间为 20 us. 完整的数据传输时间 tm 如式 (5) 所示,其中 tp 是传播时间, ts 是发送数据时间, tr 是接收数据时间.

$$tm = tp + ts + tr \quad (5)$$

表 4 数据传输时延表

BWDSP 模型数	switch 模型数	数据包数	传播时延 (us)
6	3	96	152
11	3	176	254
17	5	272	334
20	6	320	402
23	7	368	526
23	7	368	580

当系统中数据流量^[18]过大时,测试 RapidIO 交换模型发生缓存数据溢出时的系统行为. 该 RapidIO 交换模型检测到缓存溢出情况,立刻向数据的发送方进行缓存已满的响应,并减缓数据发送方的数据发送频率,调整系统的拥塞程度.

3.5 实验总结

Tundra 公司开发的 RapidIO 交换芯片 Tsi578 应用广泛,符合 1.3 版本的串行 RapidIO 互联规范,串行数据传输速率可以达到 1.25 Gbs. 而本文所设计的 RapidIO 交换模型除了正常工作的 5 端口模式外,支持扩展端口数,可以灵活配置形成复杂的系统拓扑结构. 采用 RapidIO 最大数据包长度 276 字节时,数据传输速率可以达到 4.416 Gbs.

BWDSP 芯片计算峰值性能为 12.8 GFLOPS,则 4 个芯片构成的 BWDSP 板卡系统在忽略系统通信消耗的情况下,理论计算性能可以达到 51.2 GFLOPS,具有进行深度学习应用开发的潜力.

综合上述实验结果及分析,该 BWDSP 虚拟平台成功实现了多 BWDSP 模型之间的数据通信,且数据传输速率符合 RapidIO 标准规定.

4 总结与展望

为了使 RapidIO 协议更好的适用于具体硬件产品,

提高 BWDSP 模型之间相互通信的能力. 本文以国产 BWDSP 芯片模型为基础,基于 SystemC 建模语言设计了 RapidIO 交换模型,并构建了 BWDSP 虚拟平台. 实验结果表明该串行 RapidIO 的功能级模拟符合实际 RapidIO 规范标准,设计的 BWDSP 虚拟平台数据传输速率达到预先设计的性能指标.

本文 BWDSP 虚拟平台中的交换模型主要实现串行 RapidIO 功能级的模拟,在 switch 交换模型内部的调度算法模块中,目前只实现了按照时间和优先级的轮询调度方式,考虑为 BWDSP 虚拟平台提供更加复杂的数据包调度方式以适应多种数据通信应用场景. 未来针对更加复杂的系统互联拓扑结构,需要设计更加灵活的动态路由算法,优化数据传输性能.

参考文献

- 胡阳. 基于 PowerPC 的嵌入式无线通信控制单元设计[硕士学位论文]. 北京: 北京邮电大学, 2015.
- Ma YK. Design on SRIO bus switch module based on CPS1848. *Radio Engineering*, 2014, 44(10): 73–76.
- 李挥, 陈曦. *SystemC 电子系统级设计*. 北京: 科学出版社, 2010.
- Yang CX. Design and implementation of switch unit in communication system based on RapidIO bus. *Computer & Digital Engineering*, 2014, 42(6): 1003–1010.
- 邓文齐. 基于 BWDSP 的众核深度学习加速器的研究[硕士学位论文]. 合肥: 中国科学技术大学, 2018.
- Gao YG, Wang H, Ni M. Design of serial RapidIO-PCI bridge based on FPGA. *Computer Engineering*, 2009, 35(2): 233–235.
- Wu CR, Cen F, Cai HZ. A high-performance heterogeneous embedded signal processing system based on serial RapidIO interconnection. *Proceedings of the IEEE International Conference on Computer Science and Information Technology*. Chengdu, China. 2010. 611–614, doi: 10.1109/ICCSIT.2010.5564987.
- Liu J, Tian Y, Wei H, et al. Design and implementation of serial RapidIO based on DSP and FPGA. *Advanced Materials Research*, 2014, 971–973: 1581–1585. [doi: 10.4028/www.scientific.net/AMR.971-973.1581]
- 王凤驰, 陈常青, 李正东. 基于“魂芯一号”的雷达信号处理机边扫设计. *雷达科学与技术*, 2014, 12(6): 645–648. [doi: 10.3969/j.issn.1672-2337.2014.06.015]
- Huang L, Liu FY. High-speed interconnection network based on RapidIO and memory mapping. *Computer Engineering*,

- 2008, 34(14): 116–117, 120.
- 11 Shah SA, Khanvilkar S, Khokhar A. RapidIO traffic management and flow arbitration protocol. *IEEE Communications Magazine*, 2006, 44(7): 45–52. [doi: [10.1109/MCOM.2006.1668380](https://doi.org/10.1109/MCOM.2006.1668380)]
 - 12 Deng B, Zhang YS. Research on RapidIO multicasting configuration and transfer performance. *Aeronautical Computing Technique*, 2016, 46(3): 111–115.
 - 13 Huang ZP, Wu G, Cui WL. Technology of RapidIO concurrent static routing enum. *Modern Radar*, 2017, 39(9): 16–19. [doi: [10.16592/j.cnki.1004-7859.2017.09.004](https://doi.org/10.16592/j.cnki.1004-7859.2017.09.004)]
 - 14 刘萍, 冯桂莲. 图的深度优先搜索遍历算法分析及其在应用. *青海师范大学学报(自然科学版)*, 2007, (3): 41–44.
 - 15 殷从月, 张兴明, 魏帅, 等. RapidIO 网络中一种基于最小隔离块的流量驱动机制. *计算机工程*, 2018, 44(9): 118–123, 135. [doi: [10.19678/j.issn.1000-3428.0049197](https://doi.org/10.19678/j.issn.1000-3428.0049197)]
 - 16 Huang ZZ, Chai XL, Li X, *et al.* Driver design of PCI-RapidIO bridge based on vxworks. *Computer Engineering*, 2010, 36(4): 239–240, 243.
 - 17 Rao K. Test and analysis of serial RapidIO system based on DSP and FPGA. *Information and Electronic Engineering*, 2012, 10(6): 680–684.
 - 18 Zhang Y, Wang Y, Zhang P. Research on physical layer traffic management schemes in serial RapidIO interconnect. *The Journal of China Universities of Posts and Telecommunications*, 2011, 18(1): 64–69. [doi: [10.1016/s1005-8885\(10\)60029-5](https://doi.org/10.1016/s1005-8885(10)60029-5)]