

基于两级拆分的多播片上网络路由器研究

杜高明, 吴树明, 李向阳, 宋宇鲲, 张多利, 欧阳一鸣
(合肥工业大学 电子科学与应用物理学院, 安徽 合肥 230009)

摘 要: 本文在层次化网络结构多播路由算法的基础上, 提出一种改进的多播数据包拆分机制, 并设计实现出多播片上网络. 该机制针对目的节点为上行路径上以及上行路径相邻节点的情况, 采用空闲拆分策略, 在不引入新的死锁情况下, 有效降低了多播数据包的整体延迟. 实验分析表明, 该拆分机制在目的节点在上行路径及其邻域数量多时表现更加优越. 相对于不拆分传输方式, 传输延迟最大降低了 50.3%.

关键词: 片上网络; 多播路由; 两级拆分; 延迟

中图分类号: TN47

文献标识码: A

文章编号: 1000-7180(2018)05-0035-06

Research on Two-level Packet Split Multicast NoC Router

DU Gao-ming, WU Shu-ming, LI Xiang-yang, SONG Yu-kun,
ZHANG Duo-li, OUYANG Yi-ming

(School of Electronic Science & Applied Physics, Hefei University of Technology, Hefei 230009, China)

Abstract: This paper proposes an improved multicast packet split mechanism based on the multicast routing algorithm of hierarchical network structure; this paper designs and implements the multicast network. This mechanism effectively reduces the overall delay of multicast packets without introducing new deadlocks when the destination node is the upstream node and the neighboring nodes in the upstream path. The experimental results show that the proposed method is superior to the destination node in the upstream path and its neighborhood. The transmission delay is reduced by 50.3%, as opposed to no split transmission.

Key words: NoC; multicast routing; two level splitting; delay

1 引言

随着 NoC 的不断发展, 片上网络上核与核之间的通讯关系有了进一步的需求^[1]. 传统的片上网络设计只考虑一个源节点向一个目的节点发送数据的单播通信模式^[2]. 不少应用实例要求 NoC 须支持源节点与多个目的节点同时通讯的模式^[3], 单播模式中向多个目的节点发送数据时, 采用源节点依次向每个目的节点发送数据的方法, 但是这样会使网络中的流通数据增多, 加大拥塞^[4]. 相比多次单播的通信方式, 采用多播能够将网络的通信性能提高 90%, 功耗降低 53%^[5]. 文献[6]提出一种基于层次

化网络结构 (Hierarchical Network Architecture, HNA) 的混合多播 (Hybrid Multicast, HM) 路由算法, 这种实现方式相比传统的二维网络的多播通信的性能得到很大提高, 延迟有所降低, 但是存在局限性. 也会造成“绕道传输”, 增加传输延迟.

针对上述问题, 本文提出了一种基于两级拆分多播路由 NoC. 针对上述两个问题采用的两级拆分机制, 降低了网络的整体延迟, 提高数据传输效率.

2 两级拆分机制

如图 1 所示为传统的基于 HNA 的混合多播路由 NoC. 所示层次化片上网络结构是由 6×6 的底层

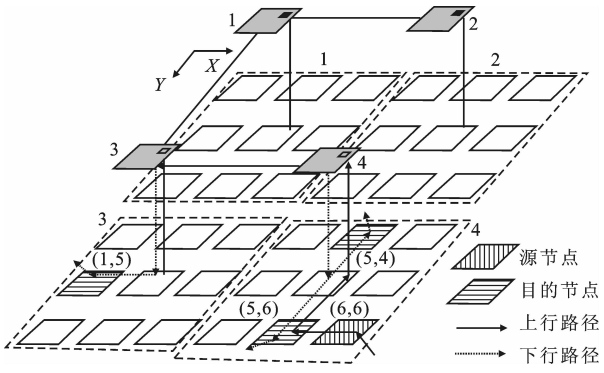


图1 层次化网络结构

二维网络和 2×2 的上层网络所构成的双层片上网络,底层二维网络分为 4 个区,每个区由 3×3 的子网络组成. 多播数据包的传输先沿上行路径到达上层网络,再由上层网络下发到各个区域. 例如,源节点(6, 6)要向目的节点(5,6)、(5,4)和(1,5)进行多播通信. 多播数据包先依照上行路径传输到上层网络 4 号路由器,将要到达(5,6)和(5,4)节点的数据按照下行路径直接传到底层网络 4 号区域,发送到目的节点,而要到达(1,5)的数据则是先从 4 号路由器转发到 3 号路由器,再传到底层网络 3 号区域.

基于 HNA 的混合多播路由算法的一个明显不足之处在于多播数据包必须先传输到上层网络,这会增加当目的节点为上行路径上和上行路径近邻节点的延迟. 如图 1 所示, (5,6)节点为上行路径上的目的节点,同时(5,4)节点在上行路径的相邻节点处. 如果多播数据包在上行的过程中,能够将这两种节点的数据从原多播包中拆分出来,直接传输到目的节点而不需要上传到上层网络,则可降低片上网络的延迟.

如图 2 所示为多播数据包在底层网络传输的上行通道与下行通道. 在原有的 HM 算法中,数据上行时,按规定路径经过 R_4 节点到达上层网络. 数据下行时,经过 R_4 到达下层网络,按规定路径拆分数据包并传输到目的节点. 在数据包上行传输的过程中引入树状多播的思想,把上行路径看作是树状数据包,多播数据包在上行过程中,对于合适的叶子节点将拆分出来直接从底层网络传输.

从图 2 可以看出,区域内的上行路径与下行路径之间没有重合的部分,所以将上行路径上的目的节点从多播数据包中拆分出来,不会与下行的数据包发生死锁情况. 针对目的节点在上行路径相邻目的节点上的情况,拆分后与下行路径会有重合部分,

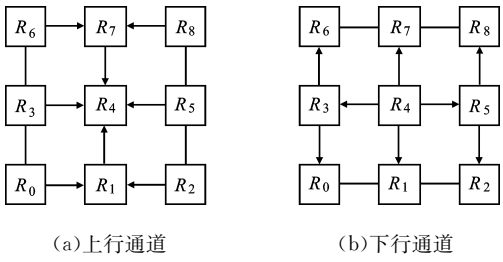


图2 上行与下行通道

进而产生死锁.

如图 3 所示,对于路由器 R_1 ,从左右两侧输入到 R_1 的多播数据包有可能会因同时请求北方向和本地方向的输出端口,进而产生死锁. 原有的 HM 算法为了避免拆分数数据包时网络中传输的数据发生死锁,规定了所有的多播数据包都必须上传到上层网络,再由上层网络下发到各个区域. 不考虑拥塞的前提下,当数据包在底层网络传输的跳数比从上层网络传输的跳数小很多时,则选择在底层网络传输的延迟会较小. 针对 HM 算法的不足,本文在不引入新的死锁情况下且兼顾整个多播数据包的延迟,加入了以上行路径上的目的节点和上行路径相邻的目的节点为拆分对象的多播数据包两级拆分机制.

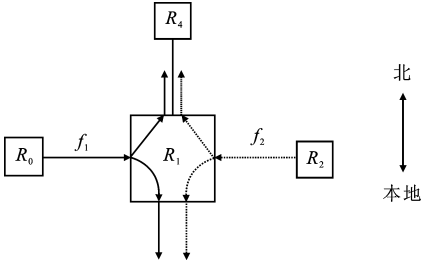


图3 拆分数据间的死锁

针对上述数据包拆分后的死锁问题,采用不等待拆分传输策略. 在多播数据包上行的路径中,对于目的节点在其上行路径上的情况,若目的节点的本地输出端口空闲则拆分传输;若被占用,则按原有路径传输到下一个节点. 对于目的节点在其上行路径相邻节点的情况,若对应方向输出通道空闲,则将目的节点的数据拆分出来,单独传输;若对应输出通道处于占用的状态,则不拆分,按照原先的多播路径传输到下一节点. 如图 4 所示,实线为原数据包的传输路径,虚线为可拆分情况下的拆分传输路径.

3 两级拆分机制算法实现

3.1 数据包结构设计

改进的多播路由算法(Improved Hybrid Multi-

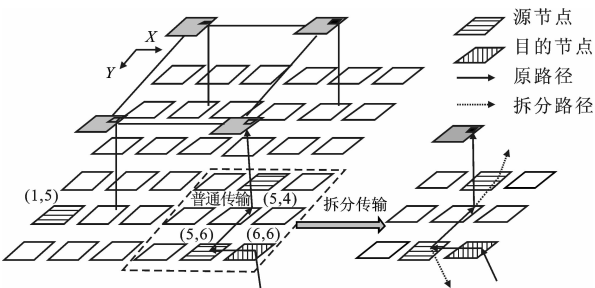


图4 拆分传输示例

cast, IHM)重新定义了网络数据包的传输协议. 协议将数据包中的微片分为头微片、体微片和尾微片. 单播数据包与多播数据包体微片和尾微片采用相同的编码方式,但其头微片则采用不同的编码. 这是因为单播数据包只在底层网络按照XY维序路由的方式从源节点传输到目的节点,而多播数据包则需要先上行到区域的上层路由器,再从上层网络分发到底层网络,从而到达目的节点.

协议的具体内容如图5所示. 为了便于拆分多播数据包,每一个目的节点都对应一个多播头微片. 路由器根据vcid字段将单播和多播数据包分别存储在不同的缓存中. 标志位M用来区分该数据包是单播数据包还是多播数据包. 标志位N用来区别数据包在上层网络传输方向(由北向南或由南向北). 标志位U用来确定数据包当前是处在上行状态还是下行状态. 多播数据包在传输过程中头微片中的U标志在从上层网络进入到底层网络时,会从1变为0,以此来判断该数据包往区域中间路由器发送还是分发到区域的各个目的节点. up_addr字段表示目的节点所对应上层路由器的编号,作为数据包在上层网络传输的目的节点,采用XY二维编码方式. dst_node和src_node字段分别是数据包在底层网络的目的节点和源节点编号,采用XY二维编码方式.

多播头微片								
vcid (2 bit)	flit_type (2 bit)	M	N	U	8 bit	up_addr (8 bit)	dst_node (8 bit)	src_node (8 bit)
单播头微片								
vcid (2 bit)	flit_type (2 bit)	M	18 bit			dst_node (8 bit)	src_node (8 bit)	
体/尾微片								
vcid (2 bit)	flit_type (2 bit)	payload (35 bit)						

图5 IHM 算法中的微片格式

多播数据包的两级拆分机制是在多播数据包上

行阶段,通过对比dst_node字段和当前节点的坐标值,来确定是否将多播数据包中该目的节点的数据拆分传输.

3.2 硬件实现

整个层次化多播路由器按其路由节点的结构不同分为底层区域非中间路由器、底层区域中间路由器和上层路由器三个部分组成.

根据改进后的IHM算法设计实现出路由器. 本节所设计的路由器在传统的路由器架构上进行改进. 传统的路由器设计一般包括五个方向的输入输出端口,分别是北(North N)、南(South S)、西(West W)、东(East E)、本地(Local L);按照模块划分一个路由器包含五个输入模块;五个路由计算模块;五个仲裁器和一个交叉开关.

其中输入模块是路由器的数据输入端. 输入模块分类存储输入的数据,并根据数据微片的类型向路由计算模块发出译码请求或者直接输出. 路由计算模块是路由算法的具体实现,控制着数据包按照规定的方式进行传输,并向仲裁器模块申请总裁. 仲裁器针对多个输出模块同时请求同一个输出模块的情况,对数据传输请求作仲裁,以保证数据有序正确的输出. 每个输出端口均有一个仲裁器. 交叉开关是路由器的数据输出端,主要是由多个多路选择器构成,连接下一个路由器的输入模块. 其中本文在传统路由器的基础对其路由计算模块进行改进,使数据能按照IHM算法规定的方式进行传输.

3.2.1 路由计算模块

路由计算模块译码数据包中的头微片确定数据包的输入方向,并向仲裁模块发出仲裁请求. 路由计算模块译码尾微片以撤销已申请的链路,并向仲裁模块发起撤销信号. 如图6所示,SR为单播数据包路由计算单元;UR为多播上行数据包路由计算单元,多播数据包的拆分传输策略包含在DR中;DR为多播下行数据包计算单元.

根据头微片中的信息判断该数据包是单播数据包还是多播数据包,上行数据包还是下行数据包,分别选择不同的路由路径,分为三种情况.

(1)单播数据包.

若为单播数据包,则按照传统二维网格结构路由器的XY路由方式计算输出端口.

(2)多播上行数据.

若为多播上行数据包,首先判断头微片中的目

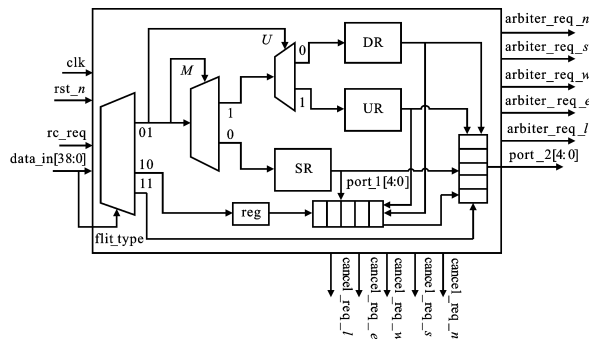


图 6 路由计算模块电路结构图

的节点是否为当前节点的相邻节点,且输出到该相邻节点的输出端口是否空闲.若条件都满足,则从多播数据包中拆分出来直接传输到目的节点.其次判断目的节点是否就是当前节点且本地输出端口是否空闲,若是,则从本地输出端口输出.最后,若之前的条件都不满足,则根据头微片中源节点编号,计算出源节点所处区域,向该区域的中间路由器传输.

(3)多播下行数据包.

若为多播下行数据包,则按照先 X 后 Y 的方式,发送到各个目的节点.

模块中定义了两个 5 位的寄存器,分别是 port_1 和 port_2. port_1 记录当前译码的头微片所请求的输出方向,用作当前头微片输出方向选择的判断条件.由于多播数据包中会含有多个头微片,所以 port_2 记录一个数据包中所有头微片请求的所有输出端口方向,用作体微片和尾微片输出端口选择的判断条件.当传输体微片和尾微片时,寄存器 port_2 的值会赋给 port_1.这两个寄存器在译码尾微片时,都会置零.多播数据包中的头微片可能会存在多个头微片请求同一个输出方向的情况.针对这种情况,五个输出方向的仲裁请求信号 arbiter_req_i ($i=N, E, S, W, L$),对应额外的五个信号 arbiter_req_i_copy ($i=N, E, S, W, L$).当多个头微片重复请求相同的输出端口时,不再拉高 arbiter_req_i,而是拉高 arbiter_req_i_copy 信号不经过仲裁,直接打开交叉开关中对应的输出端口.

3.2.2 网络结构设计.

本文所设计的底层非中间路由器仍采用传统的路由器架构,在传统的路由器上对路由计算模块进行改进.

底层区域中间路由器的整体结构与底层区域非中间路由器类似.相对于区域非中间路由器,中间路

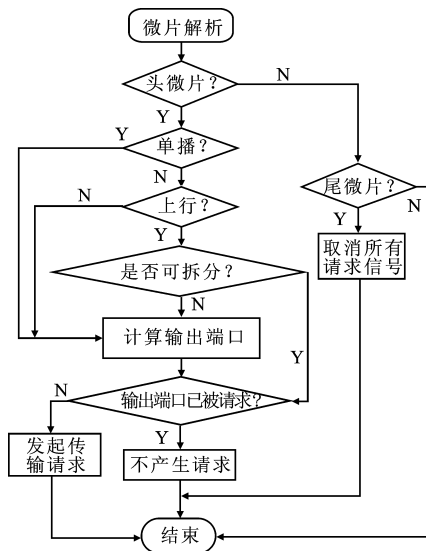


图 7 区域中间路由器路由算法

由器多出一对输入与输出端口与上层路由器连接.多播数据包到达此路由器时,除了拆分出的数据,其余数据直接传输到上层网络,如图 7 所示.区域中间路由器包括六个输入模块、六个路由计算模块和六个仲裁器,分别对应本地(L)、上(U)、东(E)、南(S)、西(W)、北(N),以及一个交叉开关.

U方向的输入模块中有两个虚通道,它们的数据输入信号以及数据有效信号是相互独立.这是为了满足上层路由解决死锁问题的需要.

对于上层 2×2 规模的网络,并不单独以基于路径的多播路由算法设计,而是采用基于 XY 的路由算法,在底层区域非中间路由的基础上修改,完成设计. N、E、S、W 四个方向的输入输出控制机制与区域非中间路由器一致.在非中间路由器的基础上设计下方向的输入模块接收来自底层网络区域中间路由器的数据包.

4 实验与分析

为了进一步优化数据包的传输延迟,在传统的基于 HNA 的混合多播路由 NoC 的基础上,加入两级拆分传输策略.在数据上行过程中,通过将数据包拆分出来的方式,避免了绕道传输.降低了区域内和区域边界目的节点的延迟,优化了网络性能.为了定量分析基于两级拆分的多播路由 NoC 性能,实验通过改变数据包长、数据包数目、上行路径上可拆分的目的节点数和上行路径相邻可拆分的目的节点数四个变量,观察其对网络的延迟影响,对 NoC 的性能

进行分析. 并将其与传统的基于 HNA 的混合多播路由 NoC 进行比较, 分析其优化效果. 实验中, 一个多播数据包的延迟定义为 T_d .

$$T_d = \sum_{i=1}^N T_i / N \quad (1)$$

式中, T_i 为数据包到达第 i 个目的节点的延迟, N 为目的节点数目.

4.1 不同包长情况

对于底层 6×6 的网络, 按照每 3×3 分成一个区域. 那么多播数据包在上行路径上最多有两个可拆分的目的节点. 数据包中的微片数目对于数据的到达延迟有一定的影响. 为了验证延迟与包长的关系, 实验拟保持数据包中头微片的数目不变, 同时控制多播上行路径上目的节点的数目一致, 通过改变数据包中体微片的数目来改变包的长度.

实验设定数据包的头微片数目为 3, 实验以 (1, 1) 为源节点, 目的节点分别是 (2, 1)、(4, 2) 和 (2, 2). 体微片的数目分别为 3、6、9. 对比不同包长情况下的延迟情况. 如图 8(a) 所示, 对于不同包长的情况, 数据包拆分与不拆分相比, 延迟平均降低 20.3%. 对于本文设计的路由器, 延迟随着包长的增加而增加.

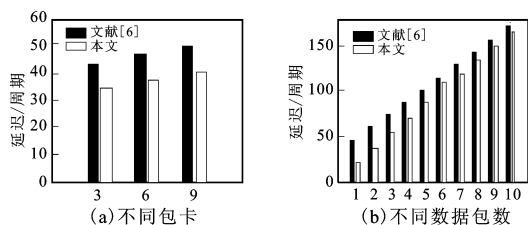


图8 数据包不同情况的延迟

4.2 改变数据包的数目

实验设定源节点为 (3, 4), 目的节点为 (2, 3), (3, 3), (4, 4) 和 (3, 5). 改变源节点到目的节点通讯的数据包总数, 以观察数据包数量对数据延迟的影响. 实验将各个源节点到目的节点的通讯数据包数从 1 增加到 10, 每个数据包包含 3 个体微片. 实验结果如图 8(b) 所示, 数据延迟随着数据包数的增加而增加. 相比于文献[6]的不拆分情况, 延迟最大优化 53.2%, 最小优化 3.43%.

4.3 改变上行路径上可拆分的目的节点数

实验从源节点向目的节点发送一个数据包, 包括三个头微片、三个体微片和一个尾微片. 图 9 中给出底层网络以及源目的节点位置.

上行路径可拆分节点数为 1: 如图 9 中 (a) 所示, 实验以 (1, 1) 为源节点, 目的节点分别是 (2, 1)、(4, 2) 和 (3, 4).

上行路径可拆分节点数为 2: 如图 9 中 (b) 所示, 实验以 (1, 1) 为源节点, 目的节点分别是 (2, 1)、(4, 2) 和 (2, 2).

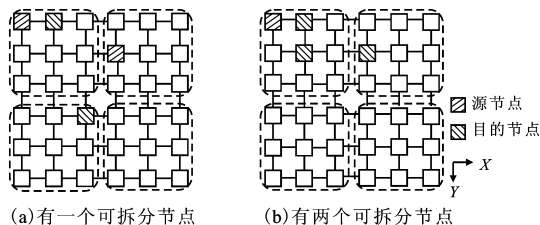


图9 上行路径上拆分节点实验配置图

不同情况下数据包到达目的节点延迟结果图 10(a) 所示. 当可拆分节点只有一个时, 包的平均延迟降低了 15.9%; 当可拆分节点有两个时, 包的平均延迟降低了 22%. 由此可以看出, 路径上拆分传输的目的节点越多, 相比于不拆分传输, 数据包的到达延迟降低得越多.

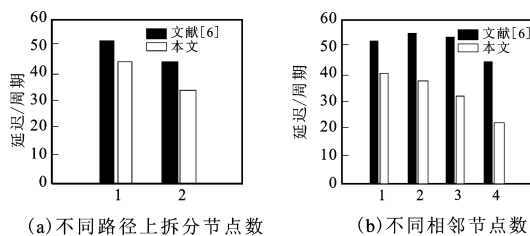


图10 拆分情况不同的延迟

4.4 改变多播上行路径相邻可拆分的目的节点数

网络中可拆分的路径相邻目的节点最多会有 7 个. 实验从源节点向目的节点发送一个数据包, 包括四个头微片、三个体微片和一个尾微片. 图 11 中给出底层网络以及源目的节点位置.

相邻可拆分节点数为 1: 如图 11 中 (a) 所示, 实验设定源节点为 (3, 4), 目的节点为 (2, 2), (3, 3), (2, 6) 和 (2, 1).

相邻可拆分节点数为 2: 如图 11 中 (b) 所示, 实验设定源节点为 (3, 4), 目的节点为 (2, 2), (3, 3), (4, 4) 和 (2, 1).

相邻可拆分节点数为 3: 如图 11 中 (c) 所示, 实验设定源节点为 (3, 4), 目的节点为 (2, 2), (3, 3), (4, 4) 和 (3, 5).

相邻可拆分节点数为 4: 如图 11 中 (d) 所示, 实验设定源节点为 (3, 4), 目的节点为 (2, 3), (3, 3),

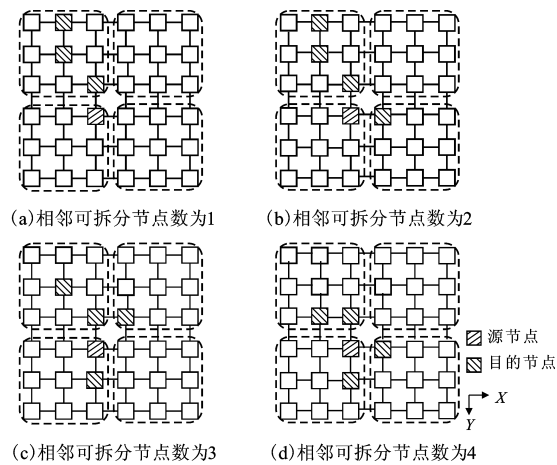


图 11 上行路径相邻拆分节点实验配置图

(4, 4)和(3, 5).

基于上述设定的实验条件,得出的延迟结果如图 10(b)所示.可拆分的相邻节点数从 1 个增加到 4 个所对应的延迟降低结果分别是 22.7%、31.5%、41.4%和 50.3%.可以得出,可拆分的相邻节点数越多,包的平均延迟优化越大.

5 结束语

本文针对已有基于层次网络结构混合的多播路由算法的不足,加入了多播数据包的两级拆分机制.在不引入新的死锁且兼顾多播数据包的整体延迟情况下,加入了数据包的拆分策略.通过改变上行路径上以及上行路径相邻的可拆分节点数,对比拆分与不拆分的数据包到达延迟,证明了拆分传输策略相对于不拆分能够有效降低数据包的到达延迟,且拆分的目的节点越多,到达延迟降低越多.

参考文献:

[1] 李世云,袁超. 片上网络容错路由算法研究[J]. 网络安全技术与应用, 2015,8(3):77-79.

[2] Salamat R, Ebrahimi M, Bagherzadeh N. An adaptive, low restrictive and fault resilient routing algorithm for 3d network-on-chip[C]// Euromicro International Conference on Parallel, Distributed and Network-Based Processing. Turku, Finland, 2015: 392-395.

[3] 陈亦欧,谭芳,凌翔. 基于网络编码的无线 NoC 多播映射[J]. 北京航空航天大学学报, 2016, 42(7):1335-1343.

[4] 张颖,吴宁,葛芬. 基于路径的无死锁 NoC 多播路由器的设计[J]. 微电子学与计算机, 2014,31(11):1-5.

[5] 郑国海. 多播片上网络路由器的设计与实现[D]. 西安:西安电子科技大学, 2014.

[6] 胡哲琨,陈杰. 基于层次化片上网络的混合多播路由算法[J]. 哈尔滨工程大学学报, 2013,33(10):1327-1333.

作者简介:

杜高明 男,(1977-),博士,副教授,硕士生导师. 研究方向为多核体系结构、片上网络体系结构、多核/NoC 性能分析.

吴树明(通讯作者) 男,(1992-),硕士研究生. 研究方向为数字集成电路设计. E-mail:1612295928@qq.com.

李向阳 男,(1991-),硕士. 研究方向为数字集成电路设计.

宋宇鲲 男,(1975-),博士,副教授. 研究方向为面向数据密集与计算机密集应用的 SoC/MPSoC 体系结构与实现.

张多利 男,(1976-),博士,教授. 研究方向为多核处理器体系结构与设计方法、多媒体专用集成电路设计实现等.

欧阳一鸣 男,(1963-),博士,教授. 研究方向为片上网络(NoC)与片上系统(SoC)、嵌入式系统的综合与测试、数字系统设计自动化.