

# 片上网络分组混合并行仲裁器的设计

周文强<sup>1</sup>, 张金艺<sup>1,2,3</sup>, 周 多<sup>2</sup>, 刘 江<sup>1</sup>

(1 上海大学 微电子研究与开发中心, 上海 200072;

2 上海大学 特种光纤与光接入网省部共建教育部重点实验室, 上海 200072;

3 上海大学 教育部新型显示与系统应用重点实验室, 上海 200072)

**摘 要:** 提出了适用于二维或三维片上网络的分组混合并行仲裁策略, 该策略对仲裁输入请求个数进行分组处理并实现了并行计算, 同时结合了 matrix 和 round robin 两种仲裁策略各自的优势. 基于此策略, 提出了 2 种分组混合并行仲裁器, 有效地改善了片上网络仲裁器在延时、最大工作频率、占用芯片资源等方面的性能指标.

**关键词:** 分组混合; 片上网络; 并行计算; 参数化设计; Verilog HDL

**中图分类号:** TN492

**文献标识码:** A

**文章编号:** 1000-7180(2015)03-0104-05

## Design of Grouped and Mixed Parallel Arbiter for NoC

ZHOU Wen-qiang<sup>1</sup>, ZHANG Jin-yi<sup>1,2,3</sup>, ZHOU Duo<sup>2</sup>, LIU Jiang<sup>1</sup>

(1 Microelectronic Research and Development Center, Shanghai University,

Shanghai 200072, China; 2 Key Laboratory of Special Fiber Optics and Optical Access Networks,

Ministry of Education, Shanghai University, Shanghai 200072, China; 3 Key Laboratory of Advanced

Displays and System Application, Ministry of Education, Shanghai University, Shanghai 200072, China

**Abstract:** A grouped and mixed parallel arbitration strategy is proposed in this paper which is suitable for two-dimensional or three-dimensional network-on-chip. The strategy groups input requests so as to process with them in parallel computing and assimilates the advantages of matrix and round robin arbitration strategy. Based on the new strategy, two grouped and mixed parallel arbiters are proposed and implemented in Verilog HDL with better performance such as delay, maximum working frequency and chip area.

**Key words:** grouped and mixed; network-on-chip; parallel computing; parametric design; Verilog HDL

### 1 引言

目前, 在片上网络仲裁器的研究中, 主要有固定优先级仲裁器、轮循 (round-robin) 仲裁器、矩阵 (matrix) 仲裁器、基于权重的优先级仲裁器、彩票仲裁器等<sup>[1-2]</sup>. 文献[3]中, 在 round-robin 仲裁器的基础上作了适当改进, 提出了一种具有动态权值的 round-robin 仲裁器, 该仲裁器在仲裁控制程度不同的时候可以实现各个请求的不同优先级仲裁, 且不需要知道通信量情况即可进行仲裁, 但同时仲裁逻辑

也增加了一定的复杂性. 文献[4]中针对 matrix 仲裁器延时较大、占用资源较多的缺点进行了一定的改进, 有效地减少了延时, 但是改进的同时也增加了额外的逻辑, 增加了功耗和面积.

本文针对 matrix 仲裁器最大工作频率高, 但是占用资源较多; round-robin 仲裁器占用资源少, 但是速度受到一定限制, 且这 2 种仲裁器在执行仲裁时都要求对所有输入请求进行统一处理而导致仲裁速度慢的情况, 综合考虑功耗、面积、延时以及仲裁逻辑复杂度等方面, 结合 matrix 仲裁策略和 round-

收稿日期: 2014-05-02; 修回日期: 2014-06-22

基金项目: 国家“八六三”计划项目(2013AA03A1121, 2013AA03A1122); 上海市教委重点学科资助项目基金(J50104)

robin 仲裁策略各自的优势,同时采用并行计算的思想对仲裁输入请求个数进行分组并行化处理,提出了分组混合并行仲裁器 1 和分组混合并行仲裁器 2. 在平均最大工作频率上,分组混合并行仲裁器 1 是 matrix 仲裁器的 1.63 倍,是 round robin 的 2.03 倍;分组混合并行仲裁器 2 是 matrix 的 1.60 倍,是 round robin 的 2.00 倍. 另外,在相同输入请求个数下,提出的 2 种仲裁器与 matrix、round robin 仲裁器相比,在功耗、延时、占用芯片资源等方面均有显著的改善.

## 2 传统片上网络仲裁器

由于片上网络仲裁器在芯片上实现,因此会受到芯片面积、功耗、延时等一系列因素的限制. 目前,片上网络仲裁器使用较多的是 matrix 仲裁器和 round-robin 仲裁器. 图 1 为 3D mesh 片上网络结构及路由节点示意图,从图中可以看到仲裁器模块是片上网络路由器的一个核心模块,它负责在各个输入请求的仲裁,对片上网络路由器和整个片上网络的吞吐量、延时、功耗等都有着重要的影响.

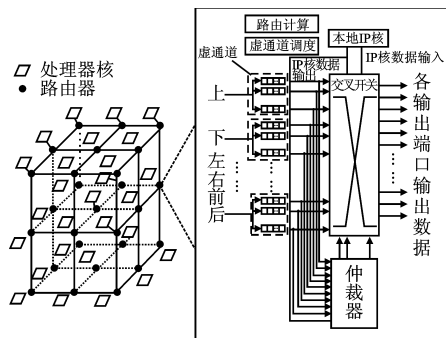


图1 3D Mesh 片上网络结构及路由器节点示意图

### 2.1 片上网络矩阵(matrix)仲裁器

片上网络矩阵(matrix)仲裁器<sup>[4]</sup>主要是将输入的  $p$  个请求之间的优先级关系以一个  $p \times p$  矩阵的形式列出,矩阵中的每个元素  $A_{i,j}$  表明了输入请求  $i$  与输入请求  $j$  之间的优先级关系,例如,输入请求  $i$  的优先级高于输入请求  $j$ ,那么  $A_{i,j}$  为 1,否则为 0. 如果矩阵中第  $i$  行都为 1,则表明输入请求  $i$  的优先级高于其他所有输入请求,也就是输入请求  $i$  具有最高优先级. 仲裁结果将响应具有最高优先级的请求. 简单地讲,矩阵仲裁器主要通过更新优先级矩阵来确定各个输入端口请求的优先级关系,从而实现仲裁. 但是,随着输入请求个数的增加,优先级矩阵扩大,占用资源急剧增多、仲裁速度逐渐变慢.

### 2.2 片上网络轮循(round robin)仲裁器

片上网络轮循(round robin)仲裁器<sup>[5]</sup>是一种优先级可循环改变的仲裁器. 在一个周期中,如果某一个端口的输入请求得到仲裁器响应输出后,在下一个时钟周期内,此端口的优先级变为最低,同时最高优先级位以循环的方式传递给下一位. 例如对于图 1 中的 3D Mesh 片上网络结构来说,假定开始时,上方端口具有最高优先级,且优先级顺序为上-下-左-右-前-后-本地,那么当上方端口得到响应后,在下一周期,上端口优先级最低,且下端口具有最高优先级,此时优先级关系变为下-左-右-前-后-本地-上. 但是,由于轮循仲裁器在执行仲裁时,需要依次检查所有输入是否存在请求,仲裁响应后,还需要对所有输入的优先级关系作出调整,当输入请求数较多时,这将会导致整个处理速度变慢,功耗显著增加.

## 3 分组混合并行仲裁器

针对 matrix 和 round robin 仲裁策略的缺点,本文提出了分组混合并行仲裁策略,即对输入请求数进行分组处理,并实现并行计算,同时混合了 round robin 占用面积小和 matrix 工作速度快的优势. 在该策略下,本文设计了 2 种片上网络分组混合并行仲裁器,且这 2 种仲裁器可以对仲裁输入请求个数任意分组,具有极大的灵活性. 其原理图如图 2 所示.

提出的片上网络仲裁器主要由分组参数设置模块、输入请求检测模块、仲裁逻辑  $A_x (x=1, 2, 3, \dots, m)$ 、仲裁逻辑 B、优先级使能控制器、输出控制器等组成.

需要说明的是,提出的分组混合并行仲裁器采用了分组并行计算的处理机制,在相同仲裁输入请求个数下,与单纯的 matrix 或单纯的 round robin 仲裁器相比将会有效提高速度、减少功耗. 另外由于混合了 round robin 和 matrix 仲裁策略,因此,图 2 中表示的仲裁逻辑  $A_x (x=1, 2, 3, \dots, m)$  和仲裁逻辑 B 有 2 种分配方法:

(1) 仲裁逻辑  $A_x (x=1, 2, 3, \dots, m)$  采用 matrix 策略,仲裁逻辑 B 采用 round robin 策略,则为分组混合并行仲裁器 1;

(2) 仲裁逻辑  $A_x (x=1, 2, 3, \dots, m)$  采用 round robin 策略,仲裁器逻辑 B 采用 matrix 策略时,则为分组混合并行仲裁器 2.

系统主要的工作原理是: ① 一组仲裁输入请求数  $request[n-1:0]$  进入仲裁器后,分组参数设置

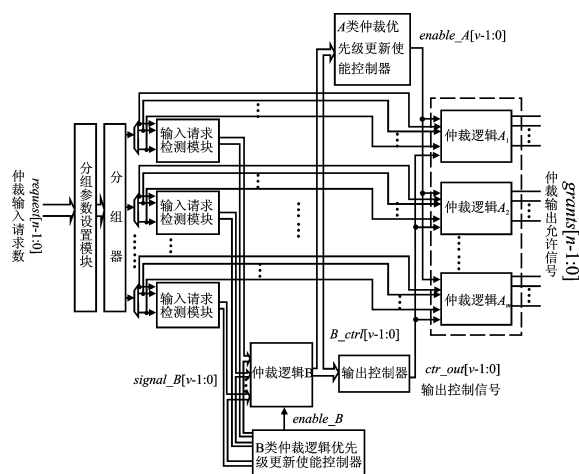


图2 片上网络分组混合并行仲裁器系统原理图

模块对  $request[n-1:0]$  进行组数划分配置及每组请求个数的配置(由于本文采用了参数化设计,划分的组数及每组请求个数等参数可任意配置),参数配置完成后,分组器模块按照配置的参数对仲裁输入请求数  $request[n-1:0]$  进行合理地分组,以便于后面仲裁逻辑  $A_x(x=1,2,3,\dots,m)$  对其进行并行处理;② 为了便于后面检测模块的检测,分组后的每一组请求数被进一步拆分,拆分后的每组请求一方面分别进入各自的仲裁逻辑  $A_x(x=1,2,3,\dots,m)$  进行仲裁处理(注意,这里的各个仲裁逻辑  $A_x(x=1,2,3,\dots,m)$  都是并行仲裁的);另一方面进入输入请求检测模块,由该模块对进一步拆分的请求数进行检测,检测其是否存在输入请求(各组输入请求中如果某一位为 1,则表示该位有请求,即对应输入端口有请求,如果某一位为 0,则表示对应的输入端口没有请求),每个输入请求检测模块的输出值是唯一的(0 或者 1),1 表示该进一步拆分的请求数中有输入请求,0 表示该请求数中无请求。③ 各输入检测模块的输出值  $signal\_B[v-1:0]$  进入仲裁逻辑 B 及 B 类仲裁逻辑优先级更新使能控制器,仲裁逻辑 B 按照现有的优先级关系对输入数进行仲裁,仲裁完毕后,B 类仲裁逻辑优先级更新使能控制器按照输入数  $signal\_B[v-1:0]$  对仲裁逻辑 B 各输入端口的优先级关系作出调整,如果输入数中有 1 出现,则  $enable\_B$  信号对仲裁逻辑 B 优先级进行使能,如果输入数中没有 1 出现,则维持仲裁逻辑 B 各端口优先级关系。④ 仲裁逻辑 B 的输出信号  $B\_ctrl[v-1:0]$  一方面进入输出控制器,另一方面进入 A 类仲裁逻辑优先级更新使能控制器,为了保证一定的仲裁公平性,A 类仲裁逻辑优先级更新使能控制器根

据信号  $enable\_A[v-1:0]$  对仲裁逻辑  $A_x(x=1,2,3,\dots,m)$  各个端口的优先级更新分别进行使能控制。同样为了保证良好的公平性,输出控制器按照信号  $ctrl\_out[v-1:0]$  对各仲裁逻辑  $A_x(x=1,2,3,\dots,m)$  分别进行控制输出,最终输出仲裁允许信号  $grants[n-1:0]$ 。其流程图如图 3 所示。

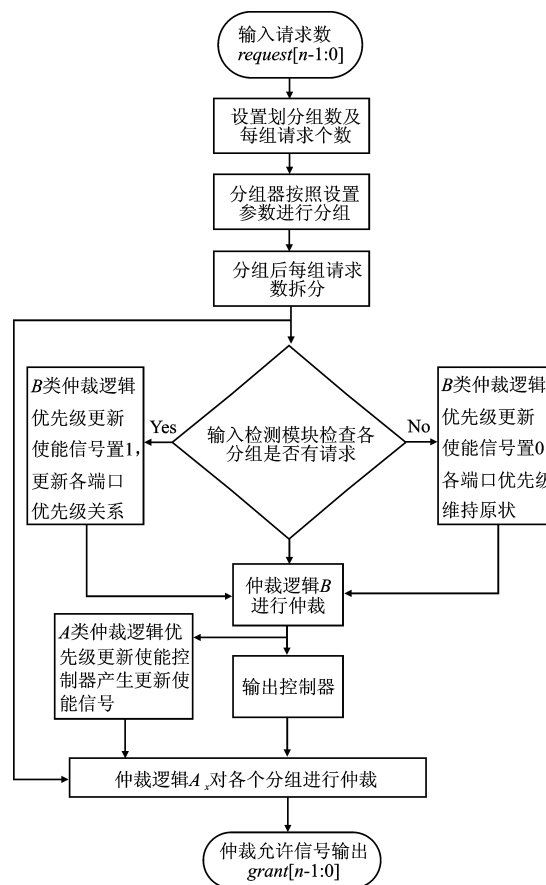


图3 分组混合并行仲裁器 1 和 2 的工作流程图

分组混合并行仲裁器 1 和 2 均采用了分组混合并行仲裁策略,部分结构不同,但工作流程类似,因此,其流程图均可由图 3 表示。

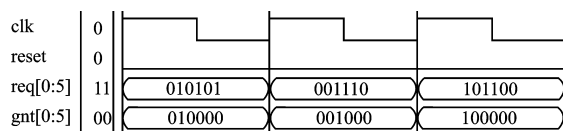
#### 4 实验结果及性能对比

本文在 Xilinx ISE 12.4 环境下采用 Verilog HDL 对提出的片上网络分组混合并行仲裁器进行了实现,同时,为了保证设计的灵活性与应用性,实现时采用了参数化设计。另外,本文对 2 种片上网络分组混合并行仲裁器进行了软件仿真及硬件 FPGA 验证,其中硬件 FPGA 采用的是 Xilinx Virtex-5 FPGA,型号为 XC5VFX70T。

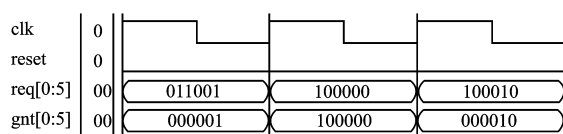
##### 4.1 仿真验证

在 Xilinx ISim 仿真环境中,本文对提出的 2 种

片上网络仲裁器在输入请求数  $req$  为 6 个时进行了仿真验证,图 4(a)所示为分组混合并行仲裁器 1 的仿真结果,图 4(b)所示为分组混合并行仲裁器 2 的仿真结果。仿真图 4 中输入请求  $req$  中 1 表示相应位有请求,0 表示无请求。从仲裁允许信号  $gnt$  可以看到,2 种分组混合仲裁器在有多个输入请求时能有效地仲裁出一个请求,能很好地完成仲裁功能,且能保证一定的公平性。



(a) 分组混合并行仲裁器 1 仿真结果



(b) 分组混合并行仲裁器 2 仿真结果

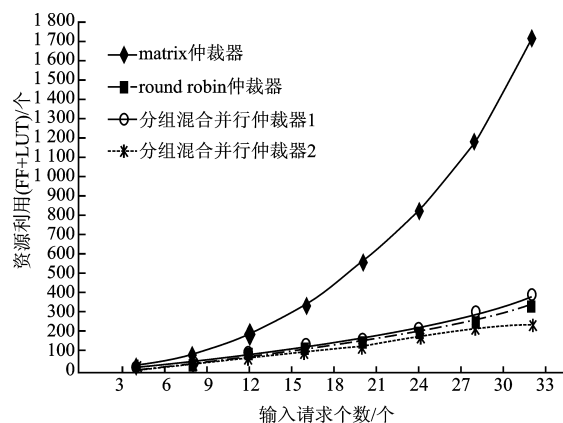
图 4 分组混合并行仲裁器 1 和 2 的仿真结果

#### 4.2 FPGA 上消耗的资源对比

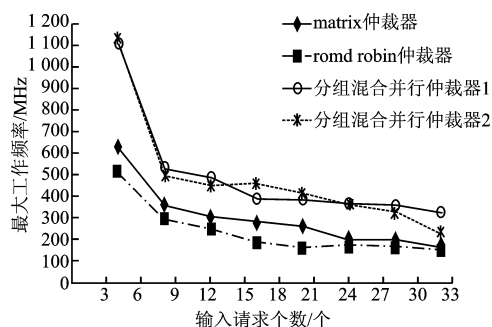
本文采用 XST 综合器对 4 种仲裁器在不同输入请求个数下分别进行综合,相应地得到了在 Virtex-5 FPGA 上占用的资源(LUT+FF)情况,从图 5(a)中可以看出,在相同输入请求个数下,matrix 仲裁器消耗的资源最多,分组混合并行仲裁器 1 和 round robin 仲裁器消耗的资源很接近,分组混合并行仲裁器 2 消耗的资源最少。另外,随着输入请求个数的增加,matrix 仲裁器消耗的资源急剧增加,而其他 3 种仲裁器则保持平稳增加的态势。

#### 4.3 最大工作频率对比

图 5(b)为不同输入请求个数下 4 种仲裁器最



(a) 不同输入请求个数下 FPGA 上的资源开销



(b) 不同输入请求个数下各仲裁器最大工作频率

图 5 不同输入请求个数下 4 种仲裁器在 FPGA 上的资源开销和最大工作频率

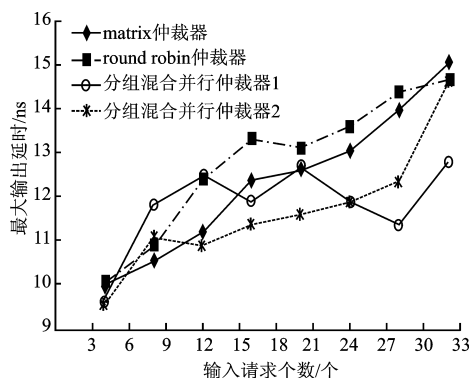
大工作频率对比图。实验数据分析表明,在相同输入请求个数下,分组混合并行仲裁器 1 和分组混合并行仲裁器 2 达到的最大工作频率远远高于 matrix 仲裁器和 round robin 仲裁器所能达到的最大工作频率。

#### 4.4 最大输出延时对比

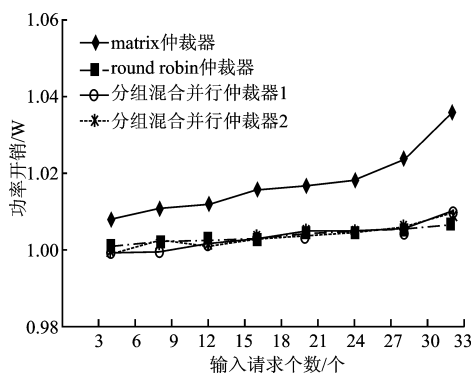
本文对 4 种仲裁器进行了时序分析,得到了 4 种仲裁器在不同输入请求个数下的最大输出延时。图 6(a)为 4 种仲裁器的最大输出延时随输入请求个数变化的曲线图,数据分析结果表明,分组混合并行仲裁器 1 的平均最大输出延时比 matrix 仲裁器平均最大输出延时减少了 4.23%,比 round robin 仲裁器的平均最大输出延时减少了 7.76%。分组混合并行仲裁器 2 的平均最大输出延时比 matrix 仲裁器减少了 5.26%,比 round robin 仲裁器减少了 8.75%。

#### 4.5 功耗开销对比

对 4 种仲裁器在不同输入请求个数下,分别进行功耗分析,得到了相应的功耗开销情况。图 6(b)为 4 种片上网络仲裁器在 FPGA 上实现时功耗(包



(a) 不同输入请求数下各仲裁器最大输出延时变化情况



(b) 不同输入请求个数下各仲裁器功耗开销情况

图 6 不同输入请求个数下 4 种仲裁器  
最大输出延时和功耗开销情况

括静态功耗和动态功耗)随输入请求个数变化的曲线图,实验数据结果表明,2 种分组混合并行仲裁器的平均功耗比 matrix 仲裁器有明显的减少,和 round robin 仲裁器相比则较为接近.实验数据表明 4 种仲裁器的静态功耗占主导地位,但是 4 种仲裁器的静态功耗都在 0.992 W 左右,几乎没有差别.而在动态功耗方面,分组混合并行仲裁器 1 的平均动态功耗比 matrix 仲裁器减少了 55.8%,比 round robin 仲裁器减少了 3.2%;分组混合并行仲裁器 2 的平均动态功耗比 matrix 仲裁器减少了 53.3%,与 round robin 仲裁器的平均动态功耗较为接近.

## 5 结束语

本文结合了 matrix 仲裁策略和 round robin 仲裁策略各自的优势,同时利用并行计算的思想对仲裁输入请求个数进行分组并行处理,提出了 2 种适用于二维或三维片上网络的分组混合并行仲裁器,并采用 Verilog HDL 进行了实现,实验结果分析表明,提出的 2 种片上网络分组混合并行仲裁器能够很好地实现仲裁功能,且能保证一定的公平性.在相

同输入请求个数下,提出的 2 种片上网络仲裁器在 FPGA 资源消耗、最大工作频率、最大输出延时、功耗等方面与 matrix 仲裁器和 round robin 仲裁器相比都有显著的改善.接下去的研究是对分组混合仲裁结构作进一步改善,同时设计分组非混合并行仲裁器.

## 参考文献:

- [1] Sebastien Le Beux, Guy Bois, GABRIELA Nicolescu, et al. Combining mapping and portioning exploration for NoC-based embedded systems[J]. Journal of Systems Architecture, 2010, 56(7):223-232.
- [2] Heisswolf J, Konig R, Becker J. A scalable NoC router design providing QoS support using weighted round robin scheduling[C] // 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications. Leganes: IEEE, 2012:625-632.
- [3] Soko Divanovic, Milutin Radonjic, Gordana Gardasevic, et al. Dynamic weighted round robin in crosspoint queued switch[C] // 21st Telecommunications forum TELFOR 2013. Belgrade:IEEE, 2013:109-112.
- [4] Peter G Sassone, Jeff Rupley II, Edward Brekelbaum, et al. Matrix scheduler reloaded[C] // Proceedings of the 34th annual international symposium on Computer architecture. San Diego: ACM, 2007:335-346.
- [5] Anuprita S Kale, Prof M A. Gaikwad. Design and analysis of on-chip router for network on chip[J]. International Journal of Emerging Technology and Advanced Engineering, 2012, 2(1):143-147.

## 作者简介:

周文强 男,(1988-),硕士研究生,研究方向为 VLSI 设计、NoC 设计. E-mail: zhwq\_work@163.com

张金艺 男,(1965-),教授,博士生导师.研究方向 SOC/NO C 可靠性设计、无线通信类 ASIC 设计.