

基于网络演算的片上网络重组缓存上界的预测方法

杜高明, 宋 平, 宋宇鲲, 张多利, 李 苗

(合肥工业大学 微电子设计研究所, 安徽 合肥 230009)

摘 要: 多路径路由片上网络(Network on Chip, NoC)中, 业务流之间的冲突导致数据包乱序到达目的节点, 在目的节点处的重组缓存通过暂存乱序到达的数据包以保证数据包按照发包的先后顺序输出. 传统方法通过全遍历所有路径得到重组缓存的最大值, 即重组缓存上界, 对于较大规模的网络, 全遍历耗时长且效率低. 对此提出一种得到重组缓存上界的预测方法: 通过冲突矩阵, 推导出每条目标子业务流的冲突系数, 进而选出冲突系数最大的无交叉路径组, 通过路径组来预测重组缓存上界. 相比于全遍历方法, 该预测方法可以缩短分析时间, 简化算法复杂度, 提高效率, 从而有效减少重组缓存区面积, 降低片上网络设计成本.

关键词: 多路径最短路由; 片上网络; 重组缓存; 预测方法

中图分类号: TN401

文献标识码: A

文章编号: 1000-7180(2016)04-0006-04

A Prediction Method for Upper Bound of Reorder Buffer of NoCs Based on Network Calculus

DU Gao-ming, SONG Ping, SONG Yu-kun, ZHANG Duo-li, LI Miao

(Institute of VLSI Design, Hefei University of Technology, Hefei 230009, China)

Abstract: In multi-path routing NoCs, the problem of packets disorder will occur due to the conflicts among flows. Reorder buffer at target node can keep the order of sending the packets by storing the packets temporarily. Traditional way is traversing all the paths to get the maximal value of reorder buffer, which is the upper bound of reorder buffer. As to a network of larger scale, traversal way is a time-consuming and low efficiency process. An prediction method to calculate the upper bound of reorder buffer was proposed: derive conflict coefficient of each target sub-flow from conflict matrix, and select the non-cross path pairs which have maximal conflict coefficient, predict the upper bound of reorder buffer through these path pairs. Compared to traditional way, this prediction method can shorten analysis time, simplify algorithm complexity, improve efficiency, and thus effectively reduce reorder buffer area and the design cost of NoC.

Key words: multi-path routing; network on chip; reorder buffer; network calculus

1 引言

在 NoC 设计中, 相比于单路径路由, 多路径路由在容错、路由可靠性、服务质量(Quality of Service, QoS)路由等方面有很多优势. 然而, 在多路径路由中, 由于每条路径上的冲突情况不同, 就会导致数据包后发先到. 在目的节点处设置重组缓存区存储后发先到的数据包, 保证数据的有序传输.

国内外对 NoC 的研究日臻成熟. 如 Kwon 等人^[1]提出了通过在网络表面设置一个重组缓存区来实现乱序包重组. 然后这种方法面积开销大, 利用率低. 随后 W.-C. Kwon 等人^[2]通过把重组缓存区从网络表面移动到片内路由来提高利用率和面积开销. 虽然这提升了资源利用率, 但会引起 NoC 尺寸增大而使性能降低. Ebrahimi M 等人^[3]提出了一个新型的动态的缓存分配管理. 实现缓存的共享. 上述

工作描述了重组缓存区对于 NoC 中的并行通信是重要和实际的,尤其在提高内存效率方面。然而,它们大多缺乏重组缓存上界的分析。Du 等人^[4]提出了一种得到 NoC 中多路径最短路由最差性能下延迟上界的方法,并且提出了冲突矩阵,但是并没有对重组缓存上界进行分析。

文献^[5]提出了基于网络演算计算最坏情形下的重组缓存上界的方法。但是该项工作仅初步探讨重组缓存上界问题,整个过程计算量大,耗时长,效率也会十分低。因此,如何通过有效的冲突预测来减少计算量,提高计算效率,对拓展重组缓存上界计算模型的适应性具有重要意义。

针对上述问题,本文提出基于网络演算进行冲突分析从而预测重组缓存上界的方法,通过冲突分析预测评估缓存的最大面积,而不需要遍历所有情况,有效缩短分析时间,简化算法复杂度,提高效率,从而有效减少重组缓存区面积,降低片上网络设计成本。尤其是对于大规模网络,其优势更加突出。

2 基本原理

本文提出一种基于网络演算的 NoC 中多路径最短路由重组缓存上界的预测方法,通过冲突矩阵,推导出每条目标子业务流的冲突系数,进而选出冲突系数最大的无交叉路径组,其中无交叉路径组指的是目标业务流拆分的两条目标子流不共享除起始节点和目的节点外的任一相同路由节点的路径组,通过设置约束条件 $A=4(N-1)$, $N \geq 2$ 进行挑选。根据各路径组的重组缓存值来预测重组缓存的最大值,即重组缓存上界。

图1给出了 NoC 的基本机制。其中冲突流的拆分矩阵定义为每条冲突流在每个节点均匀拆分,得到的拆分流量构成的矩阵。

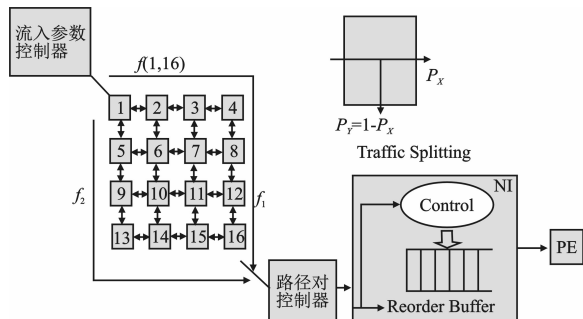


图1 NoC 基本机制

假设网路中共有 M 条冲突流,则节点在其方向

上的冲突系数定义为 $C = \sum_{k=1}^M (\epsilon r + \eta b)_k p_{k,i,d}$ (其中 k 代表冲突流, i 代表路由节点, d 代表路由节点的方向),即为每条冲突流的参数 $(\epsilon r + \eta b)$ 乘以相应冲突流的节点拆分流量 p 的总和, r 和 b 分别为表示业务流特性的平均转发速率和突发度, ϵ 和 η 分别为平均转发速率和突发度对冲突的贡献指数, p 为冲突流在每个节点的拆分流量。

把每条目标子流经过的节点在其方向上的冲突系数加和 $C_{\text{sub-tag}} = \sum C_{i,d}$ (其中 i 代表目标子流经过的节点, d 代表节点的方向) 为每条目标子流的冲突系数。

冲突矩阵为 $N^2 \times 4$ 矩阵,行代表路由节点,范围为 $\in (1, N^2)$,列代表东西南北四个方向,每个元素代表路由节点在其方向上的冲突系数,其中北和西方向均为0(对于最短路由,只有两个方向)。每一组无交叉路径组中路径冲突系数大的作为该组的冲突系数 $C_{(m,n)} = \max(C_{\text{sub-tag-m}}, C_{\text{sub-tag-n}})$ (其中 m 代表路径组的第一条子流, n 代表路径组的第二条子流)。路径组中重组缓存上界与延迟上界大的趋势一致,而子流冲突系数与子流延迟的趋势一致,因而可以用路径组中冲突系数大的来表征重组缓存上界。

3 预测机制

3.1 挑选过程

NoC 中目标业务流定义为所需要研究的对象,冲突业务流定义为对目标业务流造成冲突的业务流,子业务流为全拆分情况下所产生的所有子流,全拆分指在每个路由节点都进行拆分。挑选过程为通过冲突矩阵,推导出每条目标子业务流的冲突系数,进而选出无交叉路径组冲突系数最大值对应的路径组的过程。

首先根据冲突流的拆分情况计算每条冲突流在其对应节点的拆分矩阵,把每个节点的拆分流量进行线性加和得到节点的冲突系数,即为目标流的冲突矩阵;其次,对目标流进行全拆分,得到所有目标子流的冲突系数;进而得到无交叉路径组的冲突系数;最后,挑选出无交叉路径组冲突系数的最大值对应的路径组为预测路径组。

3.2 预测过程

根据网络演算计算出路径组对应的重组缓存的大小;挑选出重组缓存上界的过程称为预测过程。

基于网络演算理论,建立多路径路由 NoC 等价服务曲线的分析模型,推导出各条子流的延迟上界,

进而建立位于路径末端重组缓存的分析模型,计算出所有预测路径组对应的重组缓存大小;基于 code-blocks 平台,从中挑选出重组缓存最大值即为重组缓存上界。

3.3 算法分析

对上述的冲突预测重组缓存上界进行总结,得到如下算法:第 1 行到第 6 行为求解无交叉路径组;第 7 行到第 19 行为计算每对无交叉路径组对应的冲突系数;第 20 行到第 26 行为计算所有无交叉路径组冲突系数的最大值对于的路径组;最后 2 行为从预测得到的路径组对应的重组缓存中挑选出重组缓存上界。

算法 冲突系数预测重组缓存上界

已知: Noc 中网络规模为 $N \times N$, 目标子流数 A 及每条目标子流的冲突系数

求: 得出无交叉路长组冲突系数最大值对应的路径范围 pathMax.

计算重组缓存上界

```

1. for all ( $k=1; k \leq C_A^i; i++$ ) do
2.    $M \leftarrow 0$  // 从所有目标子流中选择两条子流
3.   if Paramcter ==  $4(N-1)$  then
4.      $M \leftarrow M+1$  // 挑选出无交叉路径组.
      $M$  记录无交叉路径组的个数
5.   end if
6. end for
7. path.txt // 把所有无交叉路径相存入 path.txt
8. coefficient.txt // 把各目标子流的冲突系数存入 coefficient.txt
9.  $a[M][2] \leftarrow$  path.txt // 把 path.txt 读入到数组  $a$  中, 第一列代表第一条子流的序号, 第二列代表第二条子流的序号
10.  $b[A][2] \leftarrow$  coefficient.txt 把 coefficient.txt 读入到数组  $b$  中, 第一列代表目标子流序号, 第二列代表目标子流对应的冲突系数
11. for all ( $i=0, i < M, i++$ ) do
12.   index1  $\leftarrow a[i][0]-1$  // index1 代表第一条子流的序号
13.   index2  $\leftarrow a[i][0]-1$  // index2 代表第一条子流的序号
14.   if  $b(\text{index1})[1] > b(\text{index2})[1]$  then
15.      $\max \leftarrow b(\text{index1})[1]$  取两条子流中冲突系数大的为读路径的冲突系数
16.   else
17.      $\max \leftarrow b(\text{index2})[1]$ 
18.   end if
19.    $a[M][3]$  // 把各无交叉路径的剖析突系数存入到数组  $a$  的第 3 列
20.   coefficientt Max  $\leftarrow a[0][2]$ 
21.   if  $a[i][2] > \text{coefficientt. Max}$  then
22.     coefficientt. Max  $\leftarrow a[i][2]$  // 求所有无交叉路径组冲

```

突系数最大值

```

23.    $t \leftarrow i // t$  记录冲突系数最大值在数组  $a$  中的位置
24.   end if
25.   path. Max  $\leftarrow a[t][0]a[t][1]$ 
26. end for
27.  $S \leftarrow$  path. Max // 根据预测得判到的路径范围计算对应的重组缓存
28.  $S \leftarrow \text{mar}(S)$  // 得到重组缓存上界

```

网络规模为 $N \times N$, 无交叉路径组数目为 M , 第 1 行的时间复杂度为 $O(A * (A+1)/2)$, 第 12 ~ 18 行的时间复杂度为 $O(2M)$, 第 19 ~ 23 行的时间复杂度为 $O(3M)$, 因此整个算法的时间复杂度为 $O(A^2)$.

4 实验分析

为了进一步验证优化方法的可行性和准确性, 我们进行了实验. 实验主要分为两个方面: 对于不同的冲突流数和不同的网络规模进行实验, 研究这两个因素对于冲突系数预测重组缓存上界的影响。

4.1 固定冲突流数目, 对不同的网络规模进行实验

网络规模由 3×3 增大到 6×6 条, 设目标流 f 输入到达曲线 $\alpha^f = rt + b = 0.08t + 1$; 冲突流 f_3 输入到达曲线为 $\alpha_3^{f_3} = r_3t + b_3 = 0.08t + 10$; 冲突流 f_4 的输入到达曲线 $\alpha_4^{f_4} = r_4t + b_4 = 0.08t + 20$; 冲突流 f_5 的输入到 $\alpha_5^{f_5} = r_5t + b_5 = 0.08t + 30$; 冲突流 f_6 的输入到 $\alpha_6^{f_6} = r_6t + b_6 = 0.08t + 40$. 设置服务曲线为 $\beta(t) = 0.33(t-3)^+$, 如表 1 所示。

表 1 不同网络规模中业务流的信息及运行时间

网络规模	3 * 3	4 * 4	5 * 5	6 * 6
P	8	12	16	20
M_t	6	20	70	252
$M_{\text{non-cro}}$	5	40	393	4 296
T_{all}/s	3	60	1 800	16 200
T_{max}	2	8	24	82
T_{run}/s	1	4	20	60

其中, P : 无交叉参数, $p = 4(N-1)$; M_t : 目标子流数目; $M_{\text{non-cro}}$: 无交叉路径组; T_{all} : 全遍历时间; T_{max} : 冲突系数最大路径数; T_{run} : 运行时间。

图 2 所示为 4×4 网络规模的实验结果; 图 3 所示为 6×6 网络规模的实验结果. 横坐标表示不同网络规模中无交叉路径组的序号; 纵坐标表示两组曲线的取值(为让图示更易观察, 对冲突系数进行了比例调整, 并不影响整个冲突预测结果). 3×3 规模中全遍历需要从 5 个重组缓存值中得到重组缓存上界, 而

冲突预测只需从2个冲突系数最大路径范围中预测出重组缓存上界,计算效率提高了2.5倍,运行时间缩短了2s;6×6规模中全遍历需从4296个重组缓存值中得到重组缓存上界,而冲突预测只需从82个冲突系数最大路径组中预测出重组缓存上界,计算效率提高了52倍,运行时间缩短了16140.实验表明随着网络规模的增大,优化效果更明显.

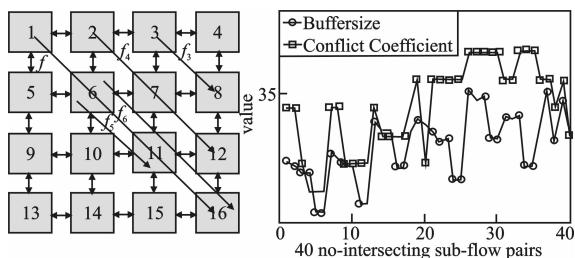


图2 4×4网络规模的实验结果

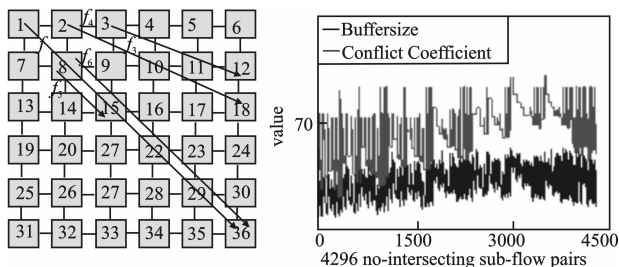


图3 6×6网络规模的实验结果

4.2 4×4网络中,对不同冲突流数进行实验

如图所示,冲突流条数由3条增加到5条,图5为3条冲突流的实验结果;图5为5条冲突流的实验结果.横坐标表示4×4网格中40对无交叉路径组的序号;纵坐标表示两组曲线的取值;正方形曲线代表40对无交叉路径组对应的冲突系数,圆形曲线代表全遍历得到的40对无交叉路径组对应的重组缓存大小.在图3中全遍历需从40个重组缓存值中得到重组缓存上界,而冲突预测只需从8个冲突系数最大路径范围中预测出重组缓存上界(图4同之),计算效率提高了5倍.实验结果表明无交叉路径组的冲突系数与重组缓存大小的趋势基本一致,证明可用冲突系数来预测重组缓存上界.且随着冲突流数目的增多,冲突系数最大路径范围数量无变化,都能很好的表征重组缓存.

5 结束语

本文提出一种Noc中多路径最短路由重组缓存上界的预测方法,目的在于避免传统全遍历方法求解重组缓存上界耗时且效率低等缺点.在该预测方法的基本原理上,通过挑选过程和预测过程来实

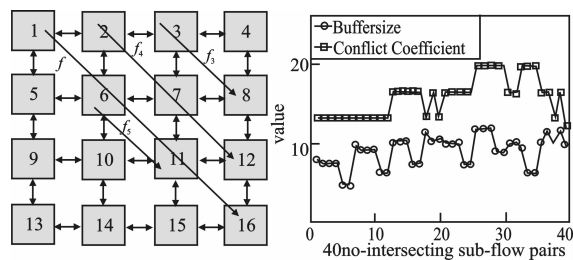


图4 3条冲突流的实验结果

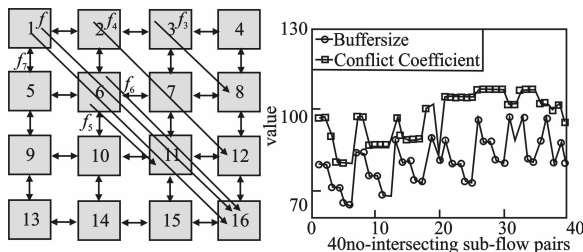


图5 5条冲突流的实验结果

现优化机制,通过实验分析得到固定网络规模,随着冲突流的增多,冲突系数可以更好地表征重组缓存上界;固定冲突流数随着网络规模的增大,优化效果更明显,实验结果很好地支持了整个优化过程.下一步的工作是把二维网络扩展到三维,并通过对更大规模的网络进行实验来验证该优化方法的正确性.

参考文献:

- [1] Kwon W C, Yoo S, Hong S M, et al. A practical approach of memory access parallelization to exploit multiple off-chip DDR memories[C]//Design Automation Conference, 2008, 45th ACM/IEEE. Anaheim; IEEE, 2008; 447-452.
- [2] Kwon W C, Yoo S, Um J, et al. In-network reorder buffer to improve overall NoC performance while resolving the in-order requirement problem[C]//Design, Automation & Test in Europe Conference & Exhibition, 2009. Nice; IEEE, 2009; 1058-1063.
- [3] Ebrahimi M, Daneshmand M, Liljeberg P, et al. A high-performance network interface architecture for NoCs using reorder buffer sharing[C]//Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on. Pisa; IEEE, 2010; 546-550.
- [4] Du G, Zhang C, Lu Z, et al. Worst-case performance analysis of 2-D mesh NoCs using multi-path minimal routing[C]//Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis. New York; ACM, 2012; 123-132.

(下转第15页)

	Name	Value	ps	80.0 ns	160.0 ns	240.0 ns	320.0 ns	400.0 ns	480.0
		960.0							
0	a	B 000C		00011100	X	00001010	X	00001100	X
9	b	B 000C		00000101	X	00001010	X	00000101	X
18	c	B 000C		01100000	X	00110000	X	01010000	X
27	n	B 001C		01000000	X	00100000	X	01000000	X
36	p	B 000C		00000001	X	00000100	X	00000001	X
45	z	B 1101		10111110	X	11011011	X	10111110	X

图 8 $F<0$ 实验数据的仿真结果

5 结束语

本文完成了三操作数前导 1 预测算法纠错模块预编码和预编码的硬件结构设计,通过编写可综合的代码、设计编码逻辑规则,生成子模块从而实现了三操作数的前导 1 预测算法纠错模块的整体结构.使用 VHDL 语言作为设计输入,仿真数据均在 Quartus ii 下进行综合得到.仿真结果表明,设计的三操作数前导 1 预测算法纠错模块可以快捷正确地完成其功能,达到了预期的目标.

参考文献:

[1] Javuer D Bruguera, Tomas Lang. Leading-one prediction with concurrent position correction[J]. IEEE Transactions on Computers, 1999, 48(10): 1083-1097.

[2] 梅小露. 浮点乘加部件中三操作数前导 1 预测算法的设计[J]. 微电子学与计算机, 2005, 22(12): 41-45.

[3] 李星. 前导 1 预测算法的设计与实现[J]. 计算机科学, 2013, 40(4): 31-33.

[4] Vojin G Oklobdzija. An algorithmic and novel design of a leading zero detector circuit ; Comparison with

logic synthesis[J]. IEEE Transactions on VLSI Systems, 1994, 2(1): 124-128.

[5] 张峰. 128 位浮点乘加融合部件的研究与实现[D]. 长沙:国防科学技术大学, 2007.

[6] 罗淑贞, 富坤, 高艳, 等. 基于 FPGA 的三操作数前导 1 预测算法的设计与性能分析[J]. 微电子学与计算机, 2015, 32(5): 16-20.

[7] Ji Rong, Ling Zhiqiang. Comments on "leading-one prediction with concurrent position correction" [J]. IEEE Transactions on Computers, 2009, 58(12): 1726-1727.

[8] P M Farmwald. On the design of high performance digital arithmetic units[D]. Palo Alto: Stanford University, 1981.

[9] N T Quach. Reducing the latency of floating-point arithmetic operations[D]. Palo Alto: Stanford University, 1994: 1-80.

作者简介:

王京京 女, (1990-), 硕士研究生. 研究方向为计算机智能控制、电子设计自动化.

富 坤(通讯作者) 女, (1979-), 博士, 讲师. 研究方向为可重构计算、计算机系统结构. E-mail: fukun@hebut. edu. cn.

(上接第 9 页)

[5] Du G, Li M, Lu Z, et al. An analytical model for worst-case reorder buffer size of multi-path minimal routing NoCs [C] // Networks-on-Chip (NoCS), 2014 Eighth IEEE/ACM International Symposium on. Ferrara: IEEE, 2014: 49-56.

作者简介:

杜高明 男, (1977-), 博士, 副研究员. 研究方向为多核集成电路体系结构、二维/三维片上网络集成体系结构. E-mail: dugaoming@ hfut. edu. cn.