

片上网络通信性能分析建模与缓存分配优化算法

王 坚 李玉柏 蒋勇男

(电子科技大学通信与信息工程学院 成都 610054)

摘 要: 该文建立了一种面向应用设计的片上网络的性能分析模型,并在此基础上提出了片上缓存优化策略和分配算法。在硬件实现平台上的仿真表明,该文建立的片上网络分析模型能很好地分析片上网络通信时延和路由节点各方向的阻塞概率,以此进行片上网络的缓存资源优化,能在同等缓存资源的情况下降低数据通过网络的平均时延,使片上网络的性能得到改善。

关键词: 片上网络;分析模型;缓存优化;分配算法;性能优化

中图分类号: TN919

文献标识码: A

文章编号: 1009-5896(2009)05-1059-04

Communication Performance Analytical Model and Buffer Allocation Optimizing Algorithm for Network-on-Chip

Wang Jian Li Yu-bai Jiang Yong-nan

(University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: For application aimed NoC design, this paper proposes an analytical model of communication performance and designs a buffer optimizing strategy and allocation algorithm. The hardware simulation results show that the model can analyze the average delay of NoC and the blocking probability of each port of routers, and the algorithm can reduce the average delay of NoC using the same amount of resources, which improves the NoC performance.

Key words: Network-on-Chip(NoC); Analytical model; Buffer optimizing; Allocation algorithm; Performance optimizing

1 引言

随着深亚微米技术的发展,越来越多的处理器核被集成在一块芯片上,传统的总线渐渐不能满足片上数据的通信要求。为了解决这个问题,片上网络(Network on Chip, NoC)被提了出来^[1-4]。在片上网络体系结构中,整个芯片被分成多个互连的模块,模块之间通过片上路由节点构成的网络连接在一起。这样,片上通信不再是由总线来传输,而是将数据封装成包的形式通过网络来传输。

同计算机网络相比,片上网络对资源的使用有很高的要求,因为它直接影响到片上网络的面积和功耗。因此,片上网络的设计要求尽可能地提高资源利用率。而在经典的片上网络路由节点结构中,缓存大小对 NoC 占用资源的多少起着决定性的作用^[5,6]。在以往的片上网络路由节点设计中,人们常常采取统一的缓存长度^[7-9],但由于片上网络大多是针对特定应用设计,网络的通信量并不是理想的均匀模式,路由节点每个方向上的负载是不均衡的。这样,统一长度的缓存分配并不能充分利用有限的资源。为了解决这个问题,本文建立了一个新型的片上网络性能分析模型以用来快速分

析 NoC 中数据通信的瓶颈,并在此基础上提出了一种有效的缓存分配算法。算法能在给出总共可供使用的缓存大小的约束下,结合分析模型中得到的通信瓶颈,得出适用于某种特定片上网络应用的缓存分配方案,使网络的性能得到改善。

本文的安排如下:第 2 节建立片上网络通信瓶颈的分析模型;第 3 节提出片上网络缓存分配算法;第 4 节给出仿真结果;最后是结束语。

2 片上网络性能分析模型

针对 mesh 结构的网络,其典型的路由节点体系结构如图 1 所示。

图 1 中,AD 为包头解析模块,Channel SEL 为链路选择模块,Arbiter 是冲突仲裁模块,Crossbar 是数据交换模块,缓存采用输入 FIFO 形式。假定每个路由节点对数据包进行读写 FIFO、包头解析、路由选择以及路由仲裁等操作的时间都相等,并且路由节点的本地 FIFO 为无穷长^[10,11];数据交换采取虚拟直通^[12]的方式,一个包在无阻塞的情况下从路由节点发送第 1 个切片起到下一级路由节点接收完最后一个切片止所需时间与包长成比例。

为了方便建模,首先对一些参数进行如下的定义: $R_{x,y}$ 为位于 (x, y) 处的路由节点,dir 为路由节点相应的方向,可

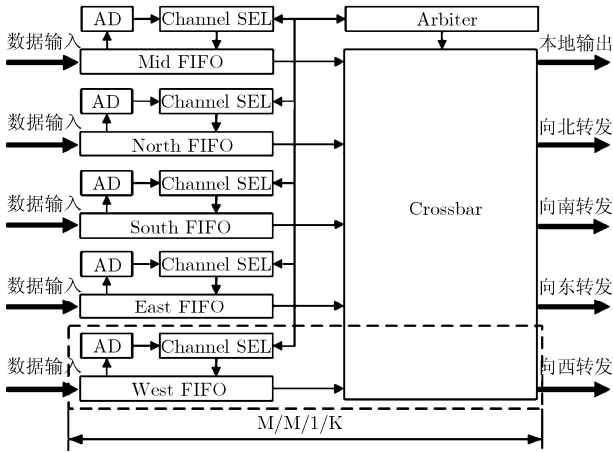


图 1 典型的 NOC 路由节点体系结构

以为 E (east)、 S (south)、 W (west)、 N (north)和 L (Local)， $FIFO_{x,y,dir}$ 为 $R_{x,y}$ 在 dir 方向的 FIFO， $\lambda_{x,y,dir}$ 为 $FIFO_{x,y,dir}$ 处的数据插入率， $\mu_{x,y,dir}$ 为 $FIFO_{x,y,dir}$ 的服务强度， $\rho_{x,y,dir}$ 为 $FIFO_{x,y,dir}$ 的利用率， $K_{x,y,dir}$ 为 $FIFO_{x,y,dir}$ 可容纳的包的个数， $P_{x,y,x',y',dir'}$ 为 $FIFO_{x,y,dir}$ 的数据通过 $FIFO_{x',y',dir'}$ 的概率， $P_{x,y,dir,dir'}$ 为 $FIFO_{x,y,dir}$ 的数据向 dir' 方向转发的概率， $P_{x,y,full,dir}$ 为 $FIFO_{x,y,dir}$ 为满的概率， $S_{x,y,dir}$ 为 $FIFO_{x,y,dir}$ 中的数据平均服务时间， S_L 为数据包从进入路由节点起到准备好接受下一级路由节点服务止所需时间， $S_{x,y,dir,Next}$ 为 $FIFO_{x,y,dir}$ 中的数据接受下级路由节点所需的平均服务时间， $\mu_{x,y,dir,dir'}$ 为 $FIFO_{x,y,dir}$ 中的数据接受 dir' 方向的服务时的平均服务率， M 为数据包的长度， $Total$ 为总的缓存大小。

把图 1 中虚线框内的部分看成一个 M/M/1/K 排队模型^[13]，根据 M/M/1/K 排队模型公式，有

$$P_{x,y,full,dir} = \rho^K \times \frac{1-\rho}{1-\rho^{K+1}} \quad (1)$$

其中

$$\rho = \frac{\lambda_{x,y,dir}}{\mu_{x,y,dir}}, \mu_{x,y,dir} = \frac{1}{S_{x,y,dir}} \quad (2)$$

因此，只要计算出 $FIFO_{x,y,dir}$ 的数据插入率 $\lambda_{x,y,dir}$ 和平均服务时间 $S_{x,y,dir}$ ，就能找出最容易阻塞的 FIFO，并把它作为通信瓶颈进行优化。

现在，选取虚线框内的部分作为分析的对象，分别针对数据插入率与数据服务率进行建模：

(1)数据插入率的计算 位于 (x,y) 处 dir 方向上的 $FIFO_{x,y,dir}$ 的数据插入率用式(3)来计算：

$$\lambda_{x',y',dir'} = \sum_{x,y} \lambda_{x,y,L} \times P_{x,y,x',y',dir'} \quad (3)$$

(2)平均服务率的计算 在片上网络中，将数据包通过路由节点转发所需的时间看成为接受本地路由节点的服务时间和接受下级路由节点的服务时间两个部分之和。

$$S_{x,y,dir} = S_L + S_{x,y,dir,Next} \quad (4)$$

本地路由节点服务时间 S_L 是指数据包从进入路由节点

起到准备好接受下一级路由节点服务止所需时间，包括 FIFO 的读写延迟 S_{FIFO} ，包头解析 S_{AD} ，链路选择 S_{SEL} ，路由仲裁 S_{ARB} 以及包通过 Crossbar 的延迟 S_{CRO} 和包在两个路由节点之间传播的延迟 S_{Link} ，即

$$S_L = S_{FIFO} + S_{AD} + S_{SEL} + S_{ARB} + S_{CRO} + S_{Link} \quad (5)$$

本地路由节点服务时间通常在路由节点设计好以后可以很容易得出，但是需要注意的是，在一些其它的网络性能分析模型中，通常假定数据到达后插入缓存并准备好接受服务这个过程是瞬间完成的，即 $S_{FIFO} = 0$ 。但是，在片上网络中，从数据包到达 FIFO 端口起到数据包可以进行包头解析起，需要经过 4 个时钟周期，而数据包进行包头解析，链路选择，路由仲裁等处理也就只需要几个时钟周期而已。因此，在片上网络建模中，需要考虑 S_{FIFO} 对模型带来的影响。

下级路由节点的服务时间是指数据包包头准备好向下级路由节点转发起到将数据包包尾发送到下级路由节点为止的时间。为了计算出数据包接受下级路由节点服务的平均服务时间，先将图 1 抽象成为图 2 所示：

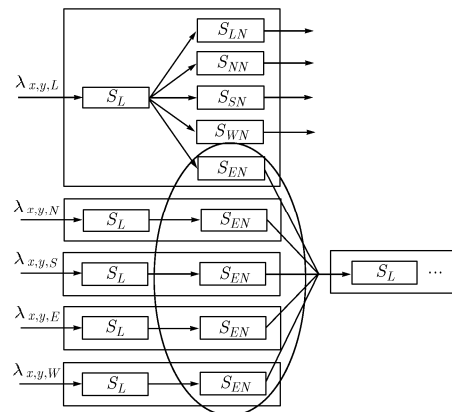


图 2 路由节点排队模型示意图

图 2 将图 1 的虚线方框部分等效成两个级连的部分：第 1 个部分模拟数据包在接受本地路由节点的服务时间 S_L ；第 2 个部分模拟数据接受下级路由节点服务时所需的时间。为了便于分析，把往相同方向传送的数据包抽象出来构成并联的子队列(子队列下标第 1 个字母表示转发方向，第 2 个字母是 Next 的简写，表示接受下级路由节点的服务)，每个子队列的服务率取决于相应的下一级 FIFO 能提供给它的服务率，第 2 个部分的总服务率可由子队列的服务率以及数据选择子队列的概率求出。

$$S_{x,y,dir,Next} = \frac{1}{\sum_{dir'} \mu_{x,y,dir,dir'} \times P_{x,y,dir,dir'}} \quad (6)$$

从式(6)可以看出，只要求出了数据向特定方向转发时能得到服务率 $\mu_{x,y,dir,dir'}$ ，就很容易得到第 2 部分的平均服务率和平均服务时间($P_{x,y,dir,dir'}$ 在路由算法确定下来以后就已经为确定值)。下面对 $\mu_{x,y,dir,dir'}$ 进行求解。

首先, 将一个路由节点中各个 FIFO 向相同方向发送的子队列提取出来构成一个整体(如图 2 中椭圆部分), 则该部分的数据插入率:

$$\lambda = \lambda_{x,y,E} \times P_{x,y,E,E} + \lambda_{x,y,S} \times P_{x,y,S,E} + \lambda_{x,y,W} \times P_{x,y,W,E} + \lambda_{x,y,N} \times P_{x,y,N,E} + \lambda_{x,y,LO} \times P_{x,y,LO,E} \quad (7)$$

将椭圆部分能得到的服务时间用下式表示:

$$S = (1 - P_{x+1,y,full,W}) \times M + P_{x+1,y,full,W} \times (S_{x+1,y,W} + M) \quad (8)$$

所以, 其服务率为

$$\mu = \frac{1}{(1 - P_{x+1,y,full,W}) \times M + P_{x+1,y,full,W} \times (S_{x+1,y,W} + M)} \quad (9)$$

根据 M/M/1 排队模型公式, 数据在图 2 的椭圆部分的平均等待时间:

$$w1 = \frac{1}{\mu - \lambda} \quad (10)$$

对椭圆中的任意一个排队队列应用 M/M/1 排队模型公式(以本地 FIFO 在 E 方向的子队列为例), 有

$$w2 = \frac{1}{\mu_{x,y,L,E} - \lambda_{x,y,L} \times P_{x,y,L,E}} \quad (11)$$

假定路由节点的调度算法足够公平, 使得每个子队列中的数据包等待时间相同, 在这种情况下, 即有 $w1 = w2$ 。联立式(7), 式(9)-式(11), 可得本地 FIFO 在 E 方向上子队列的服务率:

$$\mu_{x,y,LO,E} = \frac{1}{(1 - P_{x+1,y,full,W}) \times M + P_{x+1,y,full,W} \times (S_{x+1,y,W} + M) + \lambda_{x,y,S} \times P_{x,y,S,E} + \lambda_{x,y,W} \times P_{x,y,W,E} + \lambda_{x,y,N} \times P_{x,y,N,E} + \lambda_{x,y,E} \times P_{x,y,E,E}} \quad (12)$$

同理, 可以求出其它几个方向上的子队列的服务率 $\mu_{x,y,W,E}$, $\mu_{x,y,S,E}$, $\mu_{x,y,N,E}$ 以及 $\mu_{x,y,E,E}$ 。

利用式(6), 可以将第 2 部分所能得到的平均服务时间 $S_{x,y,dir,next}$ 表示出来, 再利用式(4), 就可以得到整个队列所能得到的平均服务时间 $S_{x,y,dir}$ 的表达式。对每个 FIFO 建立这样一个等式, 就可以得到一个等式组。结合式(1)求解等式组, 就能将每个 FIFO 的阻塞概率都用相应的 $S_{x,y,dir,L}$ 以及该 FIFO 的长度 $K_{x,y,dir}$ 表示出来, 这样就能得到在当前配置下每个 FIFO 的阻塞概率。

3 片上网络缓存分配算法

按照上述方法, 可以找出网络中最容易阻塞的 FIFO, 将该 FIFO 看出数据在网络中通信的瓶颈, 用以下算法对其进行优化。该算法如下:

第 1 步 输入系统的配置参数, 并将所有缓存资源均匀分配到每个 FIFO 上, 并通过仿真得到当前配置下数据在网络中的平均延迟 Delay。

第 2 步 根据输入参数对系统进行分析, 得到每个 FIFO 阻塞概率的表达式。

第 3 步 找出阻塞概率最大的 FIFO_MAX 以及阻塞概率最小的 FIFO_MIN。

第 4 步 更改系统配置, 令 FIFO_MAX 的深度 $Depth_X = Depth_X + 1$, FIFO_MIN 的深度 $Depth_Y = Depth_Y - 1$, 并通过仿真得到当前配置下数据在网络中的平均延迟 Delay_new。

第 5 步 如果 Delay_new 小于或等于 Delay, 则该配置生效, 并跳回第二步。如果 Delay_new 大于 Delay, 则更改配置前的系统配置为最佳值, 所以将 FIFO_MAX 与 FIFO_MIN 的深度改回原来的值, 即 $Depth_X = Depth_X - 1$, $Depth_Y = Depth_Y + 1$ 。

4 仿真结果与分析

为了验证分析模型与算法的有效性, 在 4×4 的 mesh 拓扑上, 采取 XY 维序路由, 分别对均匀通信模式以及热点通信模式, 在 Total = 96 的情况进行仿真, 并将均匀的缓存配置与算法优化后的缓存配置的网络性能进行比较。并同文献[14]中的优化方法进行了比较。在表 1 与表 2 中, UNoC 指均匀配置缓存的 NoC, CNoC 指按照文献[14]的方法进行缓存配置优化后的的 NoC, SNoC 指按照本文的方法进行缓存配置优化后的的 NoC。数据以 poisson 分布注入网络, 其插入率用参数 λ (packets/node/clock) 进行调整, 仿真时, 数据插入率选取在网络饱和点之后。

表 1 给出了均匀流量模式下数据通过网络平均延迟的比较, 从表 1 可以看出, 在总缓存为 $96P$ (P 为一个包的大小), 采取均匀流量模式的情况下, 文献[14]中的方法与本文的方法均无法对网络性能进行优化。这是因为均匀流量模式的网络中各个 FIFO 负载比较均衡, 而各个 FIFO 的容量又较少(每个 FIFO 只有 $2P$), 在这种情况下, 将任何 FIFO 缓存减少 $1P$ 配置到其他地方去都会使得该 FIFO 成为网络中数据通信的瓶颈, 导致网络的整体性能降低。

表 2 给出了热点模式下数据通过网络平均延迟的比

表 1 均匀流量模式下数据平均延迟的比较

数据插入率	UNoC(clock)	CNoC(clock)	SNoC(clock)
0.05	1399.01	1399.01	1399.01
0.06	2767.36	2767.36	2767.36
0.07	4364.17	4364.17	4364.17
0.08	5215.66	5215.66	5215.66

表 2 热点模式下数据平均延迟的比较

数据插入率	UNoC(clock)	CNoC(clock)	SNoC(clock)
0.05	2571.58	2571.58	2395.86
0.06	3476.71	3476.71	3141.05
0.07	4926.83	4796.54	4683.15
0.08	6098.26	5990.55	5705.41

较, 观察经过缓存分配后的网络 SNoC 可以发现, 与 UNoC 相比, 网络性能得到提高, 数据的平均延迟降低了 10% 左右。并且同文献[14]中的 CNoC 相比, SNoC 在数据插入率较小时也能给出优化后的网络配置(在数据插入率为 0.05 的测试点, CNoC 并不能分析出网络瓶颈并进行优化, 而 SNoC 则能分析出网络瓶颈, 优化后使得网络中数据的平均延迟从 2571.58clk 降低到 2395.86clk)。

5 结束语

本文针对片上网络的特点建立了一种 NoC 数据通信瓶颈的分析模型并提出了一种缓存分配算法。只要给出特定应用的通信特点和总共可用的缓存空间, 该模型可以检测出系统中的通信瓶颈, 并利用算法计算出 NoC 中每个路由节点各个方向上的缓存深度。我们在 4×4 的 mesh 网络中, 对均匀通信模式以及热点通信模式, 在 Total = 96 的情况进行仿真, 并将均匀的缓存配置与算法优化后的缓存配置的网络性能进行比较。并同文献[14]中的优化方法进行了比较。结果表明: 本文的方法能在不增加总的缓存空间的情况下使 NoC 的性能得到改善, 并且比文献[14]中的方法有一定的优势。

参考文献

- [1] Benini L and De Micheli G. Networks on chips: A new SoC paradigm [J]. *Computer*, 2002, 35(1): 70-78.
- [2] Murali S, Meloni P, and Angiolini F, *et al.* Designing application-specific networks on chips with floorplan information [C]. ICCAD '06. IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, Nov. 2006: 355-362.
- [3] Jeang Yuan-Long, Hung Chung-Wei, and Chiang Chuen-Muh. A methodology based on maximal-profit spanning tree for designing application specific networks on chip [C]. ICICIC '06. First International Conference on Innovative Computing, Information and Control, Beijing, Aug 2006, 2: 18-21.
- [4] Andreas H, Maarten W, and Arno M, *et al.* Applying dataflow analysis to dimension buffers for guaranteed performance in networks on chip [C]. NoCS 2008. Second ACM/IEEE International Symposium on Networks-on-Chip, Newcastle University, UK, 7-10 April 2008: 211-212.
- [5] Hu J and Marculescu R. DyAD—smart routing for networks-on-chip [C]. Proc. DAC, San Diego, Jun 2004: 260-263.
- [6] Saastamoinen I and Alho J N M. Buffer implementation for proteo networks-on-chip [C]. Proc Int Symp Circuits and Syst, 2003: 113-116.
- [7] Dally W J and Towles B. Route packets, not wires: On-chip interconnection networks [C]. Proceedings of DAC, Las Vegas, Nevada, USA, June 18-22, 2001: 683-689.
- [8] Guerrier P and Greiner A. A generic architecture for on chip packet-switched interconnections [C]. Proceedings of DATE, Paris, France, March 27-30, 2000: 250-256.
- [9] Wu Ning, Ge Fen, and Wang Qi. Simulation and performance analysis of network on chip architectures using OPNET [C]. ASICON '07. 7th International Conference on ASIC, Guilin, China, 22-25 Oct 2007: 1285-1288.
- [10] Adve V S and Vernon M K. Performance analysis of mesh interconnection networks with deterministic routing [J]. *IEEE Trans. on Parallel Distrib. Syst.*, 1994, 5(3): 225-246.
- [11] Dally W J. Performance analysis of k-ary n-cube interconnection networks [J]. *Computer*, 1990, 39(6): 775-785.
- [12] Kermani P and Kleinrock L. Virtual cut-through: A new computer communication switching technique [J]. *Computer Networks*, 1979, 3: 267-289.
- [13] Kleinrock L. Queueing Systems [M]. Vol.I: Theory. Wiley Interscience, New York, 1975: 17-85.
- [14] Hu Jingcao, Ogras U Y, and Marculescu R. Application-specific buffer space allocation for networks-on-chip router design [J]. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2006, 25(12): 2919-2933.

王 坚: 男, 1982年生, 博士生, 研究方向为片上网络性能优化。

李玉柏: 男, 1965年生, 教授, 主要研究方向为片上网络、高速信号处理、模式识别。

蒋勇男: 男, 1982年生, 硕士生, 研究方向为片上网络路由算法。