

# 基于链路状态感知的 NoC 自适应路由机制

张泽奇, 韩国栋, 黄万伟, 郑良泉

(国家数字交换系统工程技术研究中心, 郑州 450002)

**摘要:** 针对片上网络(NoC)链路出现故障时, XY 路由无法保证网络的正常通信问题, 提出一种基于 NoC 链路感知的自适应路由算法, 保证在链路出现故障后通信正常。仿真结果证明了在网络出现故障后, 使用该算法仍能保持节点之间的通信, 并且在一定的数据注入速率下, 网络出现故障前后吞吐量不变。

**关键词:** 片上网络; 路由; 链路状态感知; 自适应算法

## Adaptive Routing Mechanism of NoC Based on Awareness of Link-state

ZHANG Ze-qi, HAN Guo-dong, HUANG Wan-wei, ZHENG Liang-quan

(National Digital Switching System Engineering & Technological Research Center, Zhengzhou 450002)

**【Abstract】** As link failures are in regular Network on Chip(NoC), XY routing can not keep the communication natural. This paper proposes an adaptive routing algorithm based on the awareness of link-state in order to keep the communication well after the link failures. Simulation result of this algorithm shows that the nodes can keep communicating with each other with failures in network. Also, with some date injecting rates, the throughputs can hold the line before or after the failures in the network.

**【Key words】** Network on Chip(NoC); routing; awareness of link-state; adaptive algorithm

### 1 概述

随着集成电路制造技术的不断进步, 成百上千个 IP (Intellectual Property) 将集成到单个系统芯片 (Systems on Chip, SoC) 上<sup>[1]</sup>, IP 盒的增多带来的 SoC 片上通信的问题将成为 SoC 设计中的主要问题<sup>[2]</sup>。受到传统网络的启发, 芯片设计领域的研究人员将传统的计算机网络融入到大规模多处理器系统之中, 提出片上网络(Network on Chip, NoC)作为新的片上通信解决方案<sup>[3-4]</sup>。

NoC 概念的提出以及随着对 NoC 不断的研究, NoC 中存在的一些问题也被提出<sup>[5]</sup>。这些问题的提出是基于 NoC 状态稳定的条件, 没有考虑网络中的链路或路由节点以及链路和路由节点同时出现暂时的或永久的失效的问题。然而, 在大规模集成电路的应用中这样的问题是不可避免的。例如, 色度亮度干扰都会导致通信链路或节点出现暂时的或永久的失效<sup>[6]</sup>。虽然现在有一些针对 NoC 的容错及错误诊断测试, 但却没有对其链路的状态进行分析考虑。本文针对此问题, 对 XY 路由算法进行改进。改进后的路由算法要求每个节点存储与其相邻链路的状态, 在链路出现故障时能够自动计算选路。节点或链路故障及节点和链路故障都会导致通信中断, 因此, 统称其为链路故障; 反之, 称为链路正常。

### 2 NoC 网络 XY 路由算法

NoC 的拓扑结构必须保证每个节点可以发送数据包到其他节点。路由算法决定数据包从源地址开始选择哪一条路径到目的地址。所以, 有效的路由算法对 NoC 网络性能的好坏是至关重要的。路由算法可以按照不同的标准分为不同的几类, 如说源路由(source routing)和分布式路由(distributed routing), 确定路由(deterministic routing)和自适应路由

(adaptive routing)。

确定路由是一种常见的路由, 它的路由路径只与起点地址和终点地址有关, 给定起点和终点地址, 路由路径就被确定了, 与当前网络的状态无关。而在确定路由中, 使用最多的是维序路由(dimension-ordered routing), 因为它有着非常简单易实现的路由逻辑。在维序路由中, 每个数据包一次只在一个维上路由, 当在这个维上到达恰当的坐标之后, 才按由低维到高维的顺序在另外的维上路由。因为数据包是按照严格单调的维数变化顺序在通道内路由, 所以维序路由是没有死锁的。

以网络拓扑结构 4×4 的 Mesh 网络为例, 芯片中每一个路由节点连接着与其相邻的路由节点, 并通过一个网络接口(NI)连接着一个资源。这里的资源可以是一个处理器核、内存、一个用户自定义硬件模块或其他与 NI 接口匹配的 IP 核。

Mesh 网络中的 XY 路由的路由过程是: 先沿 X 方向将数据包送至目的节点所在的列; 再沿 Y 方向将数据包送至目的节点所在的行。即当路由节点开始处理一个数据包的时候, 因为采用的是 XY 路由, 它先会判断 X 方向上是否到达了目的节点所在的列。如果没到达, 再比较目的 X 和当前 X, 如果目的 X 比当前 X 大, 则还应该向东路由; 如果目的 X 比当前 X 小, 则说明还应该向西路由。如果判断出来 X 方向上已经到达了目的节点所在的列, 则再比较目的 Y 和当前 Y。如果相等, 说明该数据包已经到达了目的地, 则收包, 向 IP

**作者简介:** 张泽奇(1983 - ), 男, 硕士研究生, 主研方向: 片上网络, 可重构系统; 韩国栋, 副教授、博士; 黄万伟, 博士研究生; 郑良泉, 硕士研究生

**收稿日期:** 2009-06-20 **E-mail:** zhangzeqi1983@163.com

路由。如果目的 Y 大于当前 Y，则还应该向南方路由；如果目的 Y 小于当前 Y，则说明还应该向北方路由。

具体举例来说，一个从(1, 0)发出的数据包，其目的地是(3,3)，则它采用 XY 路由算法的路由路径是：(1,0)→(2,0)→(3,0)→(3,1)→(3,2)→(3,3)。

### 3 基于链路状态的自适应 XY 路由算法

XY 路由算法的优点是简单，在假设网络没有出现故障的时候具有较低的延迟。但是如果网络出现故障则会出现通信中断的严重后果。因此，需要对现有的 XY 路由算法进行相应的改进。

基于链路故障问题，本文提出自适应链路状态 XY 路由(Adaptive State-Link XY routing, ASL-XY)，使路由由节点能够处理链路中断的问题，并实时对中断的链路进行检测，以确定链路是暂时中断还是永久性中断。

在维序路由中，每个数据包一次只在一个维上路由，当在这个维上到达了目的坐标之后，才按由低维到高维的顺序在另外的维上路由。而 ASL-XY 路由则是当数据包沿某一维(如 X 方向)，从最低维到最高维路由的过程中发现某节点记录相邻 X 链路有中断的时候，即向另一维(Y 方向)传送数据，否则又转回 X 方向。如此循环，直到数据包到达目的地址。同时规定，不允许数据向远离目的节点的方向运动，因此，ASL-XY 路由也是没有死锁的。

下面简述 ASL-XY 路由算法：

(1)初始化时，每个路由节点向与其相邻的路由节点发送一个 HELLO 数据包。

(2)当路由节点接收到这个数据包时，立即向发送方回复一个 ECHO 数据包，并携带自己的统一标识。

(3)当有的路由节点在一定的方向上没有收到 ECHO 数据包时，判断这个方向上链路出现故障，发送节点记录错误状态并按照上述的自适应路由进行选路。此节点在下次有数据包向此方向传送时再发送 HELLO 数据包，检验链路是否恢复通路。

(4)由于每个节点内存储与其相邻链路的状态，所以节点将接收到的数据包向适当的方向传输。

(5)当所有路由节点都正常工作时，ASL-XY 路由与 XY 路由一致。

以 4×4 的 2D Mesh 网络为基础，具体解释 ASL-XY 路由的选路过程。如图 1 所示，一个从(1,0)发出的数据包，其目的地是(3,3)。

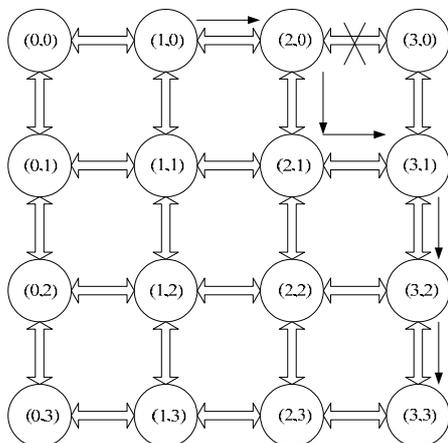
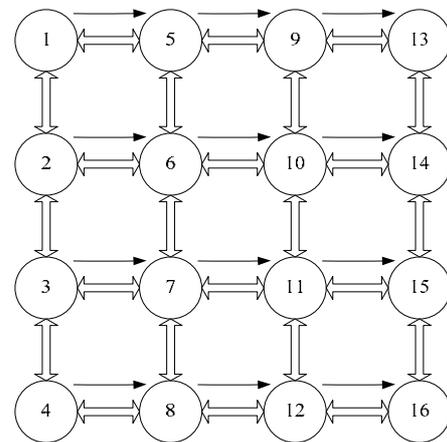


图 1 链路出现故障后的 ASL-XY 路由算法选路

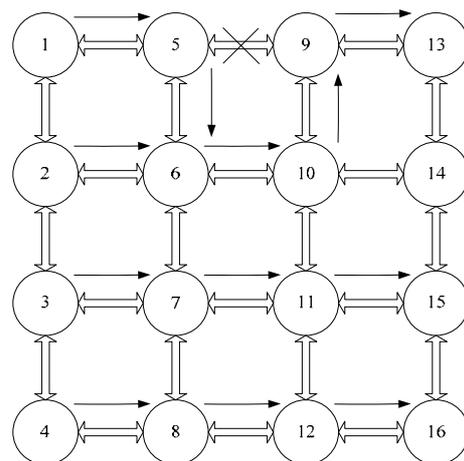
按照 XY 路由的路由过程，它的路径应该是(1,0)→(2,0)→(3,0)→(3,1)→(3,2)→(3,3)。如果节点(2,0)发向节点(3,0)的 HELLO 数据包没有得到回复，则认为这条链路出现故障，数据包不能继续向(3,0)发送。这时数据包被阻塞在了节点(2,0)中。如果采用 ASL-XY 路由，节点(2,0)会向 Y 维方向上的节点(2, 1)发送，数据包就被发往节点(2,1)了。到达(2,1)后，数据包又会先向 X 维方向上的(3,1)发送。就这样，最终的路由路径为(1,0)→(2,0)→(2,1)→(3,1)→(3,2)→(3,3)。从上面的路由路径可以看出，ASL-XY 路由和一般的 XY 路由的差别就在于链路出现故障之后，它可以根据节点中记录的链路状态不按照先 X 维再 Y 维的顺序路由，而是直接向另一个维发送数据包。这样就可以从一定程度上解决链路故障问题。

### 4 算法仿真与结果分析

本算法以 4×4 的 Mesh 拓扑结构为例进行仿真，对网络中的节点用 1~16 进行标注。以一个简单的想法为例，分别从节点 1~节点 4 将包发往节点 13~节点 16，选择的路由如图 2(a)所示。在任何时刻 t，节点 5 与节点 9 之间链路断开，则根据 ASL-XY 路由重新进行选路，路由如图 2(b)所示。



(a)链路状态正常



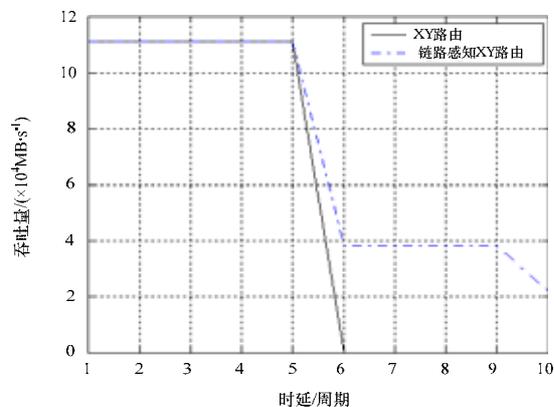
(b)链路出现故障

图 2 多节点注入数据下的自适应链路感知路由算法选路

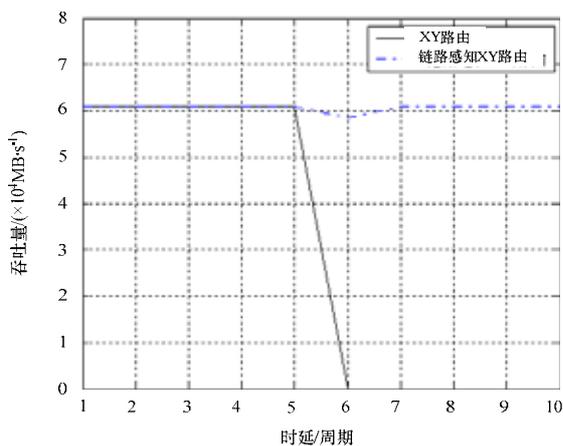
芯片上通信链路带宽的期望范围是 2.2 GB~3.2 GB 之间，包的大小为 64 Byte<sup>[7]</sup>。为了简化仿真的过程，将带宽按比例缩小到 200 MB，包大小为 12 Byte，数据注入节点的速率设为 180 MB/s。然而，在缓存数量为 4 时，只有当数据传输速

率为有效带宽的一半时,才能保证在数据传输过程中不发生丢包<sup>[8]</sup>。尽管这样,在真实的环境中,一条链路很可能被多个节点之间的数据传输所占用,所以最好是选用较低的数据传输速率,这样能够在不丢失数据的前提下更好地利用有效带宽。

图3为在4×4 Mesh拓扑结构下,维序路由和自适应链路状态感知路由的网络吞吐量比较。



(a)数据注入网络速率为 180 MB/s



(b)数据注入网络速率为 80 MB/s

图3 在4×4 Mesh拓扑结构下的网络吞吐量比较

图3(a)展示的是使用ASL-XY路由和XY路由的通信。从图中可以看出,在节点5与节点9之间的链路没有发生故障时,无论是ASL-XY路由还是XY路由,节点13的吞吐量保持不变。当节点5与节点9之间的链路出现故障时,如果使用XY路由,则节点13的吞吐量迅速下降到0,如实线所示;如果使用的是ASL-XY路由,则节点1与节点13之间仍然保持通信,但吞吐量急剧下降,并且下降超过了50%,如虚线所示。这是由于当节点5与节点9之间的链路出现故障时,ASL-XY路由的选路也占用了节点6与节点10之间的链路,这时信道同时被2条数据传输所占用。而仿真设置的数据注入速率为180 MB/s,与带宽基本上是一致的,这时就

有大量的包会被丢弃,吞吐量也急剧下降。因此,将数据注入的速率减小到80 MB/s,即减小到可用带宽的45%左右,如图3(b)所示。在这种条件下,发现当节点5与节点9之间的链路出现故障时,使用ASL-XY路由,吞吐量在略微下降之后马上又恢复到原来的值。这是因为出现故障时节点有一个重新选路的过程,而选路结束后由于数据传输速率不到带宽的一半,因此当2个通信同时占用节点6与节点10之间的链路时,就不会出现丢包,吞吐量在链路发生故障的前后大小也保持一致。

## 5 结束语

本文提出一种基于链路感知的自适应路由算法,用来解决NoC中的链路和节点出现故障后的选路问题。通过仿真验证,对维序路由中的XY路由和链路感知自适应路由进行比较,证明了该自适应算法能够解决链路故障问题,且在一定的数据注入速率下,网络出现故障前后吞吐量不变。

算法的代价是会引起其他链路的瞬时负载增加,解决方法是适当增加网络节点缓存空间并根据缓存空间状态进行合理调度。

## 参考文献

- [1] Benini L, DeMicheli G. Networks on Chips: A New SoC Paradigm[J]. IEEE Computer, 2002, 35(1): 70-78.
- [2] Jerraya A, Tenhunen H, Wolf W. Multiprocessor System-on-chips[M]. IEEE Computer, 2005, 38(7): 36-40.
- [3] Jantsch A, Tenhunen H. Networks on Chip[M]. [S. l.]: Kluwer Academic Publishers, 2003.
- [4] Hemani A, Jantseh A, Kumar S, et al. Network on a Chip: An Architecture for Billion Transistor Era[C]//Proc. of IEEE NorChip Conference. [S. l.]: IEEE Press, 2000: 166-173.
- [5] Ogras U Y, Hu Jingcao, Marculescu R. Key Research Problems in NoC Design: A Holistic Perspective[C]//Proc. of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. New York, NY, USA: ACM Press, 2005: 69-74.
- [6] Wu Ming-Shae, Lee Chung-Len. Using a Periodic Square Wave Test Signal to Detect Cross Talk Faults[J]. IEEE Design and Test of Computers, 2005, 22(2): 160-169.
- [7] Bertozzi D, Benini L. Xpipes: A Network-on-chip Architecture for Gigascale Systems-on-chip[M]. IEEE Circuits and Systems Magazine, 2004(4): 18-31.
- [8] Kumar S, Jantsch A, Soininen J P, et al. A Network on Chip Architecture and Design Methodology[C]//Proc. of IEEE Computer Society Annual Symposium on VLSI. Washington D. C., USA: IEEE Computer Society, 2002: 105-112.

编辑 顾逸斐