

基于故障节点再利用的细粒度 NoC 容错路由算法^{*}

陈庆强, 罗兴国, 张帆, 刘亮亮

(国家数字交换系统工程技术研究中心, 郑州 450002)

摘要: 针对传统 NoC 容错算法中容错粒度过粗造成资源浪费的问题, 提出了一种细粒度的自适应容错路由算法, 对带有部分故障的节点重新利用。算法将各种故障映射为一种功能故障模型, 结合新提出的路由端口优先级策略和嵌入的奇偶转向模型, 实现数据包的无死锁容错路由。实验表明, 随着负载和故障数目的增加, 该算法具有更优越的容错性能, 证明了算法的有效性。

关键词: 片上网络(NoC); 容错; 故障模型; 优先级; 细粒度; 奇偶转向模型

中图分类号: TP302; TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)07-2586-03

doi:10.3969/j.issn.1001-3695.2012.07.050

Fine-grained fault-tolerance routing algorithm of NoC based on reuse of partly defective switches

CHEN Qing-qiang, LUO Xing-guo, ZHANG Fan, LIU Liang-liang

(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: While previous work on fault-tolerance assumed switches to be either available or fully out of service, which resulted in the waste of resource. This paper proposed a new fine-grained adaptive routing algorithm, which reused the remaining functionality of partly defective switches. The algorithm matched all kinds of faults into a functional-fault model and routed the packets combined with the new priority strategy of router ports and odd-even turn model. Results obtained show that the algorithm performs better than other routing algorithms in fault-tolerance with the increasing of traffic load and fault numbers.

Key words: network on chip(NoC); fault tolerance; fault model; priority; finely granularity; odd-even turn model

随着片上资源的不断增加, 数据通信需求迅猛增长, 传统以处理速度为中心的总线互连结构已不能满足芯片的通信需求, 人们开始转向设计以通信为中心的互连结构^[1]。为此, 研究界提出在多核之间采用分组的片上通信方式, 即片上网络(network on chip, NoC)^[2]。NoC 比传统的片上总线更有效地利用了通信资源, 解决了人们对芯片通信方面的一系列要求。然而随着芯片特征尺寸和工作电压的不断减小, 集成度和工作频率的不断提高, 片上通信受到越来越多的干扰, 其故障发生率也越来越高。为了确保芯片的产量以及工作的可靠性, 必须对大量的故障进行容错处理^[3]。

传统的计算机网络中, 故障容错采用的是校正和重传的机制。然而在 NoC 中, 受到功耗与面积因素的制约, 传统的容错技术无法直接移植到 NoC 中^[4]。目前 NoC 容错路由主要有基于随机通信的路由容错算法和自适应路由容错算法^[5~11]。随机通信机制采用的是洪泛数据包的方法, 导致网络中存在大量冗余的数据包, 且功耗较大, 影响芯片性能^[5,6]。自适应路由容错算法, 通过一定方法获取故障信息, 然后绕开故障节点或链路, 达到容错目的, 所需开销较小, 且容易实现。文献[7]提出了一种基于奇偶转向模型的自适应容错路由算法, 将失效路由器包围在不相邻且不在边界上的矩形故障区内; 文献[8]提出的 DyAD 算法, 根据周围节点情况来选择采用确定路由还是

转向模型路由; 文献[9]提出的 NoC-LS 算法中, 节点将路由消息全网络发送, 每个节点都有一张网络的状态图并根据网络状态进行路由计算。

上述算法的容错粒度均在节点以上层次, 一个元件的单故障就会导致整个节点报废, 甚至要将无故障节点一起划入故障区域。但实际上, 故障节点往往是部分故障, 仍有一定的剩余功能可以利用^[12]。本文在改进的功能故障模型及其记录方式基础上, 结合路由端口的优先级策略和奇偶转向模型, 提出了一种细粒度的自适应容错路由算法(fine-grained fault-tolerance adaptive routing algorithm, FFAR)。

1 NoC 架构及故障模型

1.1 NoC 架构

图1所示是一个4×4的2D Mesh NoC结构。在结构中, 每一个节点最多与周围的四个节点相连, 并通过一个网络接口(network interface, NI)与本地资源相连。本地资源可以是一个处理器核、内存、用户自定义硬件模块或其他IP核。工作时, 本地资源将要传输的内容发给NI, NI将数据进行打包交给与之相连的节点, 然后节点根据一定的路由算法将数据包转发到目的节点, 目的节点NI对数据包进行解包, 最终送达目的节点

收稿日期: 2011-09-30; **修回日期:** 2011-11-20 **基金项目:** 国家“863”计划资助项目(2009AA012201); 上海市科委重大科技攻关项目(08dz501600)

作者简介: 陈庆强(1987-), 男, 山东泰安人, 硕士研究生, 主要研究方向为片上网络容错路由(cqq0919@163.com); 罗兴国(1951-), 男, 重庆人, 教授, 博导, 主要研究方向为计算机网络体系结构; 张帆(1980-), 男, 安徽芜湖人, 博士研究生, 主要研究方向为高性能计算机体系结构; 刘亮亮(1987-), 男, 山东邹城人, 硕士研究生, 主要研究方向为片上网络微体系结构。

的本地资源,完成一次通信。

1.2 功能故障模型

文献[12]在 NoC 测试中提出了一种对 switch 内部故障具有高覆盖率的功能故障模型,文献[13]对上述故障模型作了进一步的定义,细分为四种直通故障和八种转弯故障。结合以上两种故障模型,本文将 NoC 中的各种故障统一映射为一种功能故障模型,具体描述如下:

- 一个节点端口若不能正常地将一个数据包转移到节点另一侧端口,即视为该端口到另一侧端口之间的通道发生故障;
- 两个节点之间的链接故障映射为链接两端节点对应端口与节点其他三个端口之间的通道均发生故障;
- 节点彻底报废映射为与该节点相连接的四个相邻节点的对端口与节点其他三个端口之间的通道发生故障。

根据上述功能故障模型,故障诊断的结果记录在节点内部的寄存器。故障的记录方式采用功能故障矩阵的形式,如图2所示。纵轴为节点输入端口,横轴为节点输出端口,均分别为 N、E、S、W 四个方向。矩阵中各元素用 0 或 1 来表示相对应的输入/输出端口之间的通道是否存在故障,其中,0 表示对应通道无故障,1 表示对应通道存在故障。如图3中的故障节点 IP_{P1} ,其内部诊断出 W 端口与 S 端口之间的通道发生故障,使数据包无法在两个端口之间相互转发,则其功能故障矩阵为图2(a)所示;故障节点 IP_{P2} 中,其 E 端口侧链路发生故障,使各端口与 E 端口之间均无法进行数据包的转发,则其功能故障矩阵如图2(b)所示。

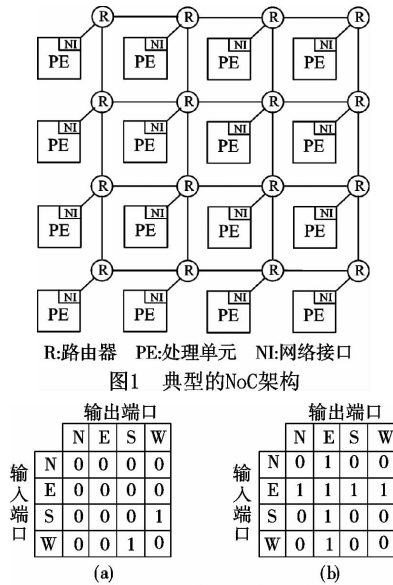


图2 节点功能故障矩阵

2 基于故障节点再利用的容错路由算法

2.1 算法基本思想

在以往的 NoC 故障容错研究中^[7-11],基本都是将节点简单地划分为正常工作 and 彻底报废两类,然后采取各种路由算法绕开故障节点(甚至是故障区域)进行容错,其容错粒度处于节点层次。因此,传统的容错算法一定程度上造成了资源和功耗的浪费。本文算法的思想就是将 NoC 故障容错的粒度进一步降低,对故障节点的剩余功能重新加以利用,提高芯片的利用效率。提出的路由算法将 NoC 各种故障映射为一种功能故障模型,根据节点坐标的相对位置决定节点内部端口的优先路

由方向,并结合节点的功能故障矩阵进行判别,从而实现故障节点内部的绕道路由,同时将奇偶转向模型嵌入到故障矩阵中,达到无死锁容错路由的目的,无须绕开整个故障节点或故障区域。

2.2 端口路由方向优先级策略

数据包在进入节点之后,一般会有四个端口选择转发。按照端口选择的优先顺序定义了四个等级,由高到低分别为 P_1 、 P_2 、 P_3 和 P_4 级。任何情况下,数据包输入端口为最低级 P_4 级端口。在剩余的三个端口中,坐标位置上方向更靠近目的节点的两个(或一个,零个)端口为最高级 P_1 级端口;与更靠近目的节点的方向相垂直方向上的一个(或两个)端口为次高级 P_2 级端口,若 P_1 级端口和 P_2 级端口均只有一个,则剩余第三个端口即为次低级 P_3 级端口。

当前节点与目的节点均不在同一水平或垂直方向上时,如当前节点 IP_c 坐标为(1,1),目的节点 IP_{D1} 坐标为(2,3),有:

- 数据包由 N 端口进入,则端口优先级排序为 $E > S > W > N$;
- 数据包由 E 端口进入,则端口优先级排序为 $N > W > S > E$;
- 数据包由 S 端口进入,则端口优先级排序为 N 或 $E > W > S$;
- 数据包由 W 端口进入,则端口优先级排序为 N 或 $E > S > W$ 。

当前节点与目的节点在同一水平或垂直方向上时,如当前节点 IP_c 坐标为(1,1),目的节点 IP_{D2} 坐标为(1,3),有:

- 数据包由 N 端口进入,则端口优先级排序为 E 或 $W > S > N$ (无 P_1 级端口);
- 数据包由 E 端口进入,则端口优先级排序为 $N > W > S > E$;
- 数据包由 S 端口进入,则端口优先级排序为 $N > W$ 或 $E > S$;
- 数据包由 W 端口进入,则端口优先级排序为 $N > E > S > W$ 。

2.3 死锁避免

上文的数据包转发策略具有全适应度,但当网络中的一些数据包由于相互等待形成依赖环,从而导致无法继续向前推进时,产生称之为网络死锁的现象。采用增加虚通道的方式可以避免死锁。但使用虚通道需要额外地引入输入缓冲队列,造成芯片面积的增加,并产生虚通道分配的延迟^[14]。Glass 等人^[15]提出使用转向模型来避免死锁,但其原始的三个部分自适应路由算法都存在着不公平问题,即对不同方向的数据包提供的路由能力不同。为了解决这个问题,Chiu^[16]提出了 odd-even 模型,按照该转向模型的要求,所有在奇数列的路由器禁止 EN 和 ES 转向,而在偶数列的路由器禁止 NW 和 SW 的转向。以往的各种转向模型均用在故障区模型的容错路由中,并以算法的形式实现。在本文中,如图4所示,按照节点所在列的奇偶位置,在其故障矩阵中标注相应的禁止转向,即以嵌入故障模型的形式实现奇偶转向模型。

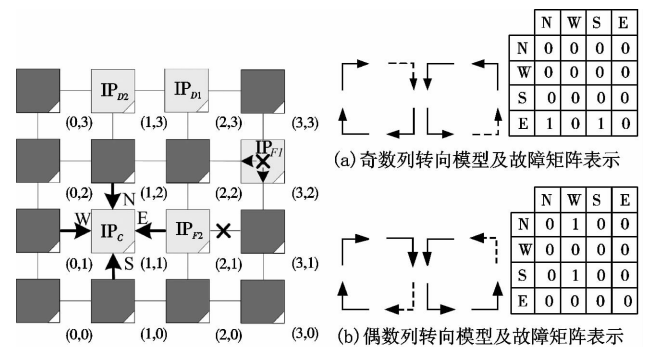


图3 带有故障的NoC 图4 奇偶转向模型及其故障模型嵌入

2.4 算法描述

根据端口路由方向优先级策略结合节点的功能故障矩阵,最终实现了基于故障节点再利用的细粒度无死锁容错路由 FFAR。算法伪代码描述如下:

```

for each input packet( )
1. read the destination address ( $X_D, Y_D$ ) for the packet to be routed.
2. compare ( $X_D, Y_D$ ) with current router address ( $X_C, Y_C$ ).
3. decide the preferred routing port/s to push the packet closer to the destination.

if the current address is the destination address
then send the packet to the local port
else if the status of  $P_1$  port(s) is available
then send the packet through this port(s)
else if the status of  $P_2$  port(s) is available
then send the packet through this port(s)
else if the status of  $P_3$  port is available
then send the packet through this port
else send the packet through the  $P_4$  port
end if

```

每个数据包进入节点进行转发时,FFAR 算法会计算其偏移量,决定优先转发端口,并查找内部功能故障矩阵来判断该优先端口是否存在功能故障。若无故障则数据包由该优先端口转发,若存在故障,则选择下一级端口并判断故障情况,依次下去,直到选择最低级端口(即输入端口)返回上一节点。当同一优先级端口有两个时,本文按照常用的 XY 维序路由进行数据包的转发;若整个网络无任何故障,则该算法等价于 XY 维序路由算法。

只要源节点到目的节点之间存在路由路径,FFAR 算法就可以找到其中一条作为路由路径,因此算法具有较好的自适应性。当数据包到达某个节点而无法继续向其他三个方向路由时,算法会选择输入端口原路返回。对于回退情况,算法始终将输入端口定为最低级的端口路由方向,能够保证数据包回退到上一节点后不会作出相同的路由决策而是选择其他方向,很大程度上避免了活锁情况的发生。同时,由于采用了奇偶转向模型,算法也能解决死锁的问题。此外,也可以专门采取一定的恢复技术来解决活锁与死锁问题。

3 实验仿真

3.1 仿真平台

为了验证本文提出的 FFAR 算法,采用仿真工具 NS-2^[17]进行算法的仿真验证,仿真模型具体参数如表 1 所示,仿真比较各路由算法在网络有无故障时的吞吐率与时延性能。在均匀分布流量模式下每个节点等概率地发送数据包到其他节点,分组之间的时间间隔由数据注入率 λ (flits/node/cycle) 确定。实际中的网络故障数要低于本文设定的值,这里设定较高的故障数是为了突出 FFAR 对带有部分故障的 NoC 的性能的提高。XY 路由算法对 NoC 中的故障部件采取的策略是将所有故障/部分故障的部件丢弃不用。

3.2 仿真结果对比

图 5(a) 为算法在时延方面的仿真结果。在无故障情况下,XY 路由算法凭借其简单高效的优点取得时延性能上的优势;在存在故障的情况下,当负载较小时,FFAR 算法由于使用

了故障节点内部的一些端口,导致一些情况下的时延增加。但在重负载的情况下或故障数量增多时,由于 FFAR 能够使用的节点数目要多于其他路由算法,其时延增加得更缓慢,其时延性能也要好于其他自适应路由算法。

图 5(b) 为算法在吞吐率方面的仿真结果。结果表明,在无故障情况下,XY 路由算法同样由于简单高效的优点,其吞吐性能占据明显优势;但随着故障数目和负载的增加,自适应路由算法表现出其优越的一面。FFAR 算法相对于其他路由算法,在同等数量故障数目和负载的情况下,由于使用更多的节点,因而能够实现更多的数据包传输;并且随着故障数量和负载的增加,FFAR 算法的吞吐率优势更加明显。

表 1 仿真模型具体参数

网络参数	取值
拓扑	6 × 6 2D Mesh
交换机制	虫孔交换机制
虚通道个数	2 个
虚通道缓存大小	16 flits
流量模式	均匀分布 (uniform)
网络故障数	0, 3, 6
故障分布及程度	随机
比较的算法	XY、DyAD ^[8] 、odd-even ^[7] 、FFAR

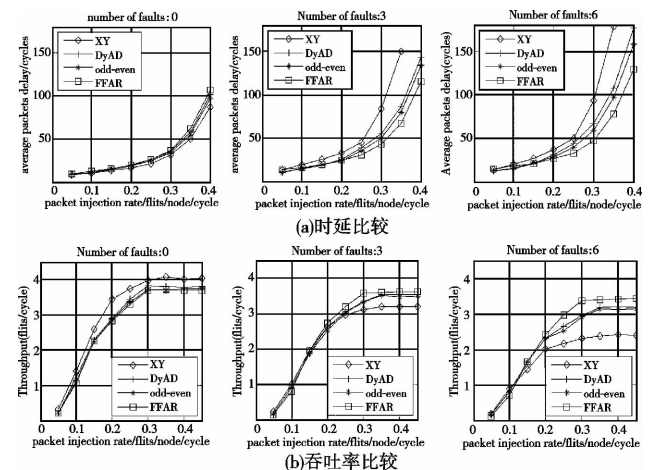


图 5 不同故障数目下各算法时延和吞吐率比较

4 结束语

发生故障的 NoC 仍可通过一定容错手段保证通信的正常进行,而发生故障的 NoC 节点,也仍有一定的剩余功能可以继续使用。本文提出了一种细粒度的自适应容错路由算法,将功能故障矩阵和路由端口优先级策略相结合,实现了故障节点的再利用,并达到了容错的目的。仿真结果表明,相比较于其他容错路由算法,FFAR 算法可以有效提高网络的吞吐率,具有较低的时延,特别是在重负载和故障数目较多的情况下,其性能优势更加明显。同时通过对故障节点再利用,提高了芯片硬件资源和功耗的使用效率。新的功能故障模型的故障检测要比检测整个路由器复杂很多,同时算法也使得容错路由器的设计变得更加复杂,这也是今后面临的挑战。

参考文献:

- [1] AGARWAL A, ISKANDER C, SHANKAR R. Survey of network on chip (NoC) architectures & contributions [J]. *Engineering, Computing and Architecture*, 2009, 3(1): 12-19.
- [2] BENINII L, MICHELI G D. Network on chip: a new SoC paradigm [J]. *IEEE Computer*, 2002, 35(1): 70-78. (下转第 2617 页)

(上接第2588页)

- [3] WU M S, LEE L C. Using a periodic square wave test signal to detect crosstalk faults [J]. *IEEE Design & Test of Computers*, 2005, 22 (2): 160-169.
- [4] DUMITRAS T, KEMER S, MARCULESCU R. Towards on-chip fault-tolerant communication [C]//Proc of Asia and South Pacific Design Automation Conference. New York:ACM Press, 2003: 225-232.
- [5] DUMITRAS T, MARCULESCU R. On-chip stochastic communication [C]//Proc of Design, Automation and Test in Europe Conference and Exhibition. Washington DC:IEEE Computer Society, 2003: 790-795.
- [6] PIRRETTI M, LINK G, BROOKS R, *et al.* Fault tolerant algorithms for network-on-chip interconnect [C]//Proc of IEEE Computer Society Annual Symposium on VLSI. 2004: 46-51.
- [7] WU Jie. A fault tolerant and deadlock free routing protocol in 2D meshes based on odd-even turn model [J]. *IEEE Trans on Computers*, 2003, 52(9): 1154-1169.
- [8] HU Jing-cai, MARCULESCU R. DyAD: smart routing for networks-on-chip [C]//Proc of the 41st Design Automation Conference. New York:ACM Press, 2004: 260-263.
- [9] ALI M, WEIZL M, HELLEBRAND S. A dynamic routing mechanism for network-on-chip [C]//Proc of the 23rd NORCHIP Conference. 2005: 70-73.
- [10] FENG Chao-chao, LU Zhong-hai, AXEL J, *et al.* FoN: fault-on-neighbor aware routing algorithm network-on-chip [C]//Proc of the 23rd IEEE International SoC Conference. 2010: 441-446.
- [11] 付斌章, 韩银和, 李华伟, 等. 面向高可靠片上网络通信的可重构路由算法 [J]. *计算机辅助设计与图形学学报*, 2011, 23(3): 448-455.
- [12] RAIK J, GOVIND V, UBAR R. An external test approach for network-on-a-chip switches [C]//Proc of the 15th Asian Test Symposium. Washington DC:IEEE Computer Society, 2006: 437-442.
- [13] 郑焱, 王红, 杨士元. 基于功能故障模型的 NoC 容错路由 [C]//第六届中国测试学术会议论文集. 2010: 204-208.
- [14] DUATO J. A new theory of deadlock-free adaptive routing in worm-hole networks [J]. *IEEE Trans on Parallel and Distributed Systems*, 1993, 4(12): 1320-1331.
- [15] GLASS C, NI L. The turn model for adaptive routing [C]//Proc of the 19th Annual International Symposium on Computer Architecture. New York:ACM Press, 1992: 278-287.
- [16] CHIU G M. The odd-even turn model for adaptive routing [J]. *IEEE Trans on Parallel and Distributed Systems*, 2000, 11(7): 729-738.
- [17] 杨智峰, 田泽. 用 NS2 评估片上网络体系结构的性能 [J]. *计算机工程与应用*, 2010, 46(18): 74-76.