

一种带有均衡策略的无虚通道容错路由算法

李 娇^{1,2}, 刘 鹏¹, 王良华¹, 崇云锋¹, 徐海鹏¹

(1 上海大学 微电子研究与开发中心, 上海 200072;

2 上海大学 新型显示技术及应用集成教育部重点实验室, 上海 200072)

摘 要: 在借鉴已有的容错路由算法基础上, 对故障节点进行扩展, 解除部分节点转向限制、实施一定的均衡策略, 缓解了上述问题。在 9×9 的 2D mesh 网络中的仿真结果表明: 与未改进的算法相比, 算法在降低数据延迟和过热节点数量等方面有着明显的优势, 在最优情况下能降低 12.7% 的数据延迟和减少 18.5% 的过热节点。

关键词: 虚通道; 容错; 负载均衡; 过热

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2016)11-0157-05

A Fault-Tolerant Routing Algorithm With Load-Balancing Strategy for 2D-Mesh Without Virtual Channels

LI Jiao^{1,2}, LIU Peng¹, WANG Liang-hua¹, CHONG Yun-feng, XU Hai-peng

(1 Microelectronics R&D Center, Shanghai University, Shanghai 200072, China; 2 Key Laboratory of Advanced Display and System Application (Shanghai University), Shanghai 200072, China)

Abstract: A fault-tolerant routing algorithm with load-balancing for 2D mesh without virtual channels is proposed based on current routing algorithms. The novel algorithm uses a modified fault model and corresponding dissemination mechanism of fault information to remove part of turns limitation and implement load-balance strategy. The simulation results demonstrate that the proposed algorithm outperforms in average package delay and hotpots compared to previous algorithms in the 9×9 2D mesh NoC. In the best case, it reduces 12.7% average delay and 18.5% hotpots.

Key words: virtual channel; fault-tolerance; load-balance; overheating

1 引言

随着半导体技术的进步, 利用多核 SoC(System on Chip)提升单芯片的计算性能已成为一个重要的发展趋势。在多核 SoC 中, 各个 IP 核之间的通信需求迅速增长, 伴随着芯片特征尺寸的持续下降、芯片密集度和复杂度的不断增大, 芯片对制造工艺偏差、外界干扰和元器件老化变得越来越敏感, 致使发生故障的可能性变大, 而且芯片在使用过程中产生的热量会加剧故障发生的可能性。这些故障可能会使 NoC 性能急剧下降甚至直接瘫痪。因此, 为了确保芯片工作的可靠性, 必须要对芯片进行容错设计。

路由算法对 NoC^[1] 的可靠性有着至关重要的影响, 一个高效的容错路由算法能在 NoC 发生故障时, 能在很大程度上降低 NoC 的性能损失。当前容错路由算法按照是否使用虚通道可分为基于虚通道的路由算法^[2-3]和无虚通路路由算法^[4-6]。使用虚通道虽然可以解决死锁问题, 并在一定程度上提升网络的吞吐量, 但是会导致 NoC 硬件开销增大、硬件复杂度变大、带宽下降^[7]。

上述的算法还有一个共同的问题: 它们把工作都集中在容错上, 而忽略了负载的均衡和功耗的分布, 导致数据包在收发过程中平均延迟变大、网络中出现局部过热和路径拥塞等问题, 严重影响了 NoC

的性能. 因此本文提出一种带有均衡策略的容错路由算法, 利用该算法可以把数据负载分配到各条路径上, 使得空闲的带宽资源能被充分的利用, 缓解了局部过热和路径拥塞等问题、提升了 NoC 的性能.

2 带有均衡策略的容错算法

2.1 NoC 故障模型

当前大部分基于无虚通道的容错路由算法使用的故障模型都只适用于基于维序路由扩展的容错路由算法. 这导致算法在非故障区也要执行维序路由, 而不能充分利用网络中的空闲路径来缓解网络中局部拥塞和过热的问题. 针对这一问题本文将对第二节中 Wu 的故障模型^[5]进行改进, 形成适用于本文算法的故障模型.

为了简化算法的复杂度、容忍更多的故障节点, Wu 定义了安全节点、危险节点和边界节点以形成故障区和故障边界. 改进的故障模型在 Wu 的基础之上添加一类缓冲节点来解除部分转向限制, 把数据负载分配到各条路径上.

其中, 危险节点是由网络中的无故障节点转化而来. 它的转化方案如规则一所示.

规则一: 所有的无故障节点在初始状态下都是安全节点, 一个安全节点只要满足下面两个条件中的其中一个, 则变为危险节点.

(1) 在 X 轴和 Y 轴的相邻位置上至少各有一个危险节点或故障节点, 则该安全节点变为危险节点.

(2) 在 X 轴的相邻位置上有一个危险节点或者故障节点, 且在 X 轴的另外一个方向上的相邻点的 Y 轴方向上有相邻的危险节点或者故障节点, 则该安全节点变为危险节点.

重复上述过程直到不再生成新的危险节点. 最终把危险节点和故障节点统称为禁用节点, 然后由这些禁用节点形成故障区.

由于文中带有均衡策略的容错算法是奇偶转向模型扩展而形成, 为了给东向、西向路由的数据包提供足够多的转向许可, 绕过所遇到的故障区, 在故障区的东、西两个方向上会有两条边界. 边界是与故障区相邻的安全节点转化而成, 具体的生成方案如规则二所示.

规则二: 故障区形成之后, 网络中只有禁用节点和安全节点. 一个安全节点满足以下其中一种条件时, 则变为边界节点.

(1) 在 Y 轴方向上与禁用节点相邻的安全节点变为边界节点.

(2) 在 X 轴方向上与禁用节点相距两跳之内的安全节点变成边界节点.

通过顺序执行 1、2 过程生成边界节点, 并由这些边界节点相互连接形成故障边界.

缓冲节点则是由故障区南、北方向上的安全节点转变而成, 根据缓冲节点的位置, 可把其分为南边界缓冲节点和北边界缓冲节点. 其转化方案如规则三所示.

规则三: 在边界形成之后网络中会出现禁用节点、边界节点和安全节点, 其中一些与边界节点相邻的安全节点满足以下任一条件时, 将转化为缓冲节点.

(1) 在 Y 轴负方向上与边界节点或者南边界缓冲节点相邻的安全节点变为南边界缓冲节点.

(2) 在 Y 轴正方向上与边界节点或者北边界缓冲节点相邻的安全节点变为北边界缓冲节点.

重复上述过程直到不再变化. 缓冲节点离故障区较近, 是否能施行均衡策略还需要对处于缓冲节点中的数据路由路径进行预判.

经过上述过程生成禁用节点、边界节点、缓冲节点后, 剩余的安全节点离故障区比较远, 受到的转向限制较少, 在文中又把它称之为“自由节点”.

图 1 是一个典型的故障模型图. 在图中黑色的节点代表故障节点, 在初始状态下没有符合危险节点生成条件 1 的节点, 但有一个符合危险节点生成条件 2 的节点(如图 1 中标有数字 2 的节点), 当这个标有数字 2 的节点变成危险节点后, 网络中又会出现两个符合危险节点生成条件 1 的节点(如图 1 中标有数字 2 的两个节点). 当网络中不再有新的危险节点产生后, 这些禁用节点会形成相应的故障区(黑色虚线内的区域). 故障区形成之后, 再根据边界节点生成规则和缓冲节点生成规则生成边界节点(带有斜线阴影的节点)和缓冲节点(带有网格状的节点), 最终剩下的安全节点称为自由节点(带有白色背景的节点).

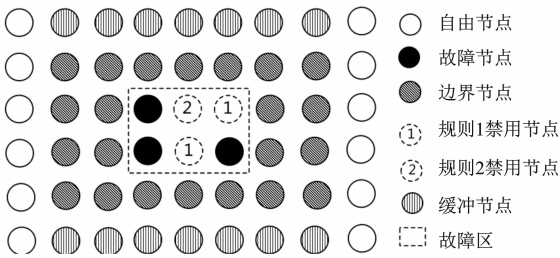


图 1 故障模型图

本文所提出的算法就是基于数据包所处节点的类型,然后对它们施行不同的路由策略,以提高网络资源的利用率。

2.2 算法基本思想

基于上述 NoC 故障模型,均衡容错路由算法将在不同的节点施行不同的路由策略,以实现容错路由的同时对负载进行均衡,以提高 NoC 芯片的性能,具体步骤如下。

- (1)判断当前节点是否为目的节点,如果为目的节点,则将数据包吸收,否则转向步骤2.
- (2)判断节点的类型和路由策略的执行. 这里特别需要注意的是缓冲节点,当缓冲节点中的数据需要绕过故障区才能到达目的节点时,则施行故障绕行策略,反之,则施行均衡策略. 边界节点和自由节点分别执行故障绕行策略和均衡策略(算法流程如图 2 所示).

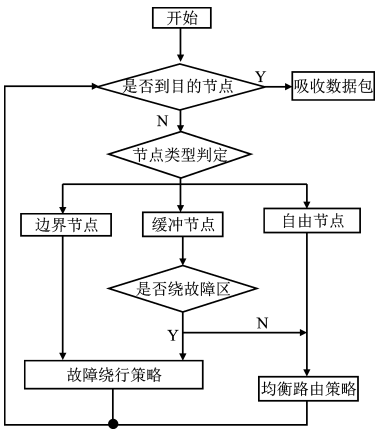


图 2 算法流程图

从算法流程中可以看出故障绕行策略和均衡路由策略决定了数据的路由路径和容错能力,而这些指标与算法的性能有着直接的关系。

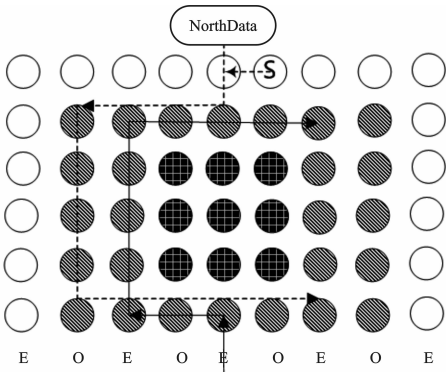
2.3 故障绕行策略

在实施故障绕行策略时,网络中的节点只按禁用节点、边界节点和非边界节点进行区分,非边界节点包含缓冲节点和自由节点. 由于禁用节点中不存在数据路由任务,在故障绕行策略中只考虑边界节点和非边界节点. 本文使用的故障绕行方案将借鉴 Wu 的绕行策略^[5],按照当前所处的边界类型施用不同的绕行规则,以确保在网络不发生死锁前提下,绕过所遇到的故障节点达到目的节点。

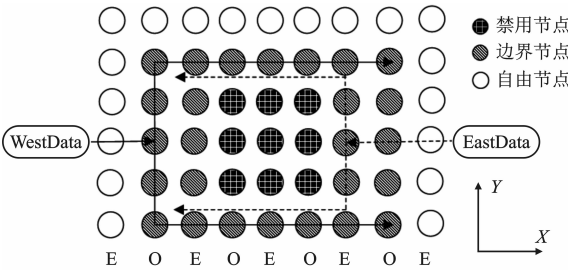
在叙述绕行策略之前,先对部分字母符号做以下定义:dx 为数据包的节点横坐标与当前节点横坐标的差值,dy 为数据包的节点纵坐标与当前节点纵坐标的差值,S 为源节点,E 代表偶数列,O

代表奇数列. 算法将根据节点所处的位置,选择不同的绕行路径,其规则如下。

- (1)如果当前节点为源节点,处于奇数列,且 dy 不为 0,数据要向西发送到相邻的偶数列节点(如图 3(a)中标有字母 S 的源节点). 反之,转向步骤 2.
- (2)如果当前节点为非边界节点,dy 不为 0,则优先进行 Y 方向路由. 在 dy 等于 0 且 dx 不为 0 时,进行 X 方向路由. 反之,当前节点为边界节点转步骤 3.
- (3)对于边界节点,如果数据包的路由方向与边界垂直,则根据边界节点的位置执行不同的绕行策略. 数据包处于南、北边界时,只有当 dy 为 0 且 dx 大于 0 时向 X 轴正方向路由,其他情况一律沿 X 轴负方向绕行数据包,其绕行方式如图 3(a)中 NorthData 和 SouthData 的绕行路径. 向东路由的数据包 EastData 到达西边界的奇数列时开始故障绕行,当目的节点在当前节点的上方时,沿顺时针方向绕行故障区,反之,在当前节点下方时,沿逆时针路由. 向西路由的数据 WestData 使用的故障绕行方式与 EastData 类似,具体如图 3(b)中 WestData 和 EastData 的绕行路径. 如果当前数据包的路由方向不与边界垂直且不需要绕行故障区就能到达目的节点时,则使用均衡策略进行路由,反之,则按照原有的绕行方向继续沿故障区路由。



(a)南/北向数据故障绕行图



(b)东/西向数据故障绕行图

图 3 故障绕行图

2.4 均衡策略

均衡策略是在遵循奇偶转向模型^[9]和最短路径的前提下,对存在多个输出端口的数据进行负载均衡.为了施行均衡策略,在每个路由节点中会有一个 4 bit 的寄存器用于决定均衡模式下的数据的输出端口.在最短路径的限定下,数据包有多个输出端口的必要条件是:dx、dy 都不为 0.因此,本文算法做如下规定:(1)dx、dy 都大于 0 时,均衡模式下的输出端口由 bit_0 决定;(2)dx 大于 0,而 dy 小于 0 时,均衡模式下的输出端口由 bit_1 决定;(3)dx 小于 0,而 dy 大于 0 时,均衡模式下的输出端口由 bit_2 决定;(4)dx、dy 都小于 0 时,均衡模式下的输出端口由 bit_3 决定.在均衡模式下 bit_x(x=0、1、2、3)的值对应的输出端口如表 1 所示.

表 1 均衡输出端口

dx、dy bit_x	0	1
dx>0&.dy>0	向北路由	向东路由
dx>0&.dy<0	向南路由	向东路由
dx<0&.dy>0	向北路由	向西路由
dx<0&.dy<0	向南路由	向西路由

数据包确定输出端口后会把相应的 bit_x 位取反,在下一个数据到达这个节点时能把它分配到另外一条路径上,以实现负载的均衡和空余的带宽资源的充分利用,提升 NoC 的性能.

2.5 无死锁路由算法

本文算法使用了故障绕行策略和均衡策略,只要证明这两种策略在 NoC 中混合使用时,网络中不会发生死锁,那么本文中的算法就是无死锁的.

故障绕行策略和均衡策略都是基于奇偶转向模型进行扩展的,而奇偶转向模型是无死锁的,现在只需要证明这两种路由策略都是遵循奇偶转向模型就能证明本文所提出的算法是无死锁的.其证明过程如下.

(1)显然,本文中使用的均衡路由策略是在奇偶转向模型所允许的转向下对数据流量进行均衡的,所以均衡策略是无死锁的.

(2)故障绕行策略也遵循奇偶转向模型.图 3(a)中,NorthData 和 SouthData 在对故障进行绕行时,可能会发生奇偶转向模型在奇数列禁止 SW 和 NW 转向,但是故障绕行策略步骤 1 中规定数据包首先要确保处于偶数列后才能进行 Y 轴方向路由.因此,图 3(a)中 NorthData 和 SouthData 产生的

SW 和 NW 转向只发生在偶数列,没有违反奇偶转向规则.与之相类似,在图 3(b)中,EastData 故障绕行时产生的 WS、WN 转向和 WestData 故障绕行时产生的 NW、SW 转向分别发生在奇数列和偶数列,也没有违反奇偶转向模型.所以,综合分析可知故障绕行策略和均衡策略都是遵循奇偶转向模型,本文所提出的路由算法是无死锁的.

3 实验分析

3.1 仿真器设置

本文将采用 NIRGAM^[10]仿真器对算法的性能进行仿真实验. NIRGAM 是一种基于 System C 的 NoC 专用仿真器,它具有代码开源、扩展性好等特点,并能对 NoC 的延迟和功耗进行的仿真.仿真器的主要参数设置如表 2 所示,仿真器将在更贴近芯片实际工作状态下的 hotspot 负载模式对 Wu 的算法和本文算法进行仿真.通过对比两种算法在平均延迟和过热节点数量上的差距以证明本文算法的优越性.在 hotspot 模式下将会随机地选取 10% 的节点作为通信热点,这些通信热点的通信量将比普通节点的通信量高出 40%.

表 2 仿真器参数设置

网络参数	取值
拓扑	9×9 2D Mesh
交换机制	虫洞交换机制
流量模式	hotspot
Flit	2 bytes
数据长度	8 filts
链路缓存	4 filts
故障分布	随机
禁用节点比例	0,10%,20%
比较算法	Wu 的算法、本文算法

3.2 实验结果

为评估网络的功耗分布,本文规定:禁用节点比例为 20%,负载率为 100%时,NoC 节点的平均功耗作为过热节点的判断阈值(T).只有当节点的功耗值大于 T 时,才判定该节点为过热节点.通过仿真器得出算法的数据包平均延迟和过热节点数量如图 4、5 所示.

图 4 为禁用节点比例在 0%,10%,20%条件下,Wu 的算法和本文算法的数据包平均延迟曲线图,其中纵轴为每个数据包的平均延迟时间,横轴为负载率或称为最大带宽占用率.在 NoC 负载率较低时,Wu 的算法与本文算法的数据延迟基本上相接近;随着负载率的逐渐增大本文算法的优势也逐渐

显现出来,最优情况下能降低 12.7%的数据延迟;但随着故障比例的增加,能施行均衡策略的节点数量也在不断减少,因此,在负载率为 100%时,两种算法的延迟差也随着故障比例的增加而逐渐变小,最终这两种算法的数据延迟将随着故障比例增加而趋于一致.

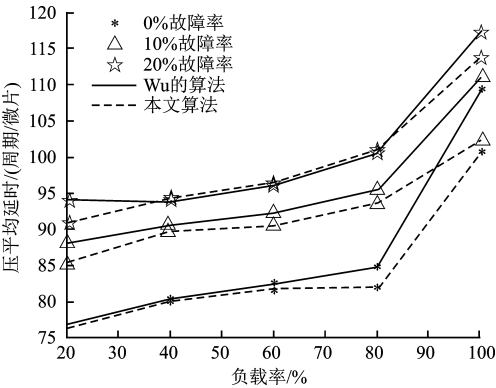


图 4 平均延迟

图 5 为禁用节点比例在 0%,10%,20%条件下 Wu 的算法和本文算法的过热节点数量曲线图. 过热节点的数量随着负载率的增加而增加;当故障率较低时,本文算法所产生过热节点的数量远小于 Wu 的算法,在最优时能减少 18.5%的过热节点;随着故障比例的增加,能施行均衡策略的通信节点也在不断减少,本文算法的优势将逐渐减小,但仍然优于 Wu 的算法.

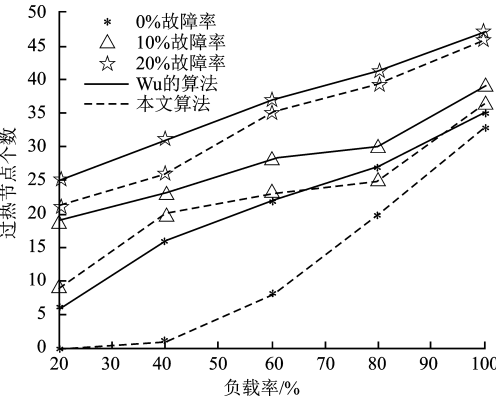


图 5 过热节点

4 结束语

本文针对当前容错路由算法存在的负载不均的问题,借鉴 Wu 的故障绕行策略提出了一种带有负载均衡的 NoC 容错路由算. 该算法能将数据负载分配到其他空闲的通信节点上,在保证网络不发生死锁的同时减少了过热通信节点的数量、降低了数据

包的平均延迟. 实验结果表明在故障率适中、负载较大的情况下本文算法能取得良好的效果 .

参考文献:

[1] Benini L, De Micheli G. Networks on chips: a new SoC paradigm[J]. Computer, 2002, 35(1):70-78.

[2] Charif A, Zergainoh N-E, Nicolaidis M. MUGEN: A high performance fault tolerant routing algorithm for unreliable networks-on-chip [C] //On-Line Testing Symposium (IOLTS). Halkidiki: IEEE, 2015:71-76

[3] Kumar M, Laxmi V, Gaur M S, et al. Fault tolerant and highly adaptive routing for 2D NoCs[C]//Defect and fault in VLSI and Nanotechnology Systems (DFT). Amsterdam:IEEE, 2014:104-109

[4] Christopher C J, Ni L M. Fault tolerant wormhole routing in meshes without virtual channels[J]. IEEE transactions on parallel and distributed systems, 1996, 7(6):620-636.

[5] Jie Wu. A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model[J]. IEEE Transactions on computers, 2003, 52(9): 1154-1169

[6] Binzhang Fu, Yinhe Han, Huawei Li, et al. ZoneDefense: a fault tolerant routing for 2D meshes without virtual channels[J]. IEEE Transactions on very large scale integration system, 2014, 22(1):113-126

[7] Christopher C J, Ni L M. The turn model for adaptive routing[C]//Computer Architecture. Los Alamitos: IEEE, 1992: 278-287.

[8] Fick D, Deorio A, Chen G, et al. A highly resilient routing algorithm for fault tolerant NoCs [C]//Design, Automation and Test Europe Conference & Exhibition. Nice France: IEEE, 2009:21-26

[9] Ge-Ming Chiu. The odd-even turn model for adaptive routing[J]. IEEE transactions on parallel and distributed systems, 2000, 11(7):729-738

[10] Lavina J. A simulator for NoC interconnect routing and application modeling [D]. UK: University of Southampton, 2007.

作者简介:

李 娇 女,(1975-),博士,讲师. 研究方向为高性能数字电路设计、集成电路可测试性设计.

刘 鹏(通讯作者) 男,(1991-),硕士研究生. 研究方向为片上网络可靠性设计. E-mail:15026918671@163.com