

基于虚通道故障粒度划分的 3D NoC 容错路由器设计

欧阳一鸣¹ 张一栋¹ 梁华国² 黄正峰² 常 郝¹

¹(合肥工业大学计算机与信息学院 合肥 230009)

²(合肥工业大学电子科学与应用物理学院 合肥 230009)

(oyymbox@163.com)

Design of Fault-Tolerant Router for 3D NoC Based on Virtual Channel Fault Granularity Partition

Ouyang Yiming¹, Zhang Yidong¹, Liang Huaguo², Huang Zhengfeng², and Chang Hao¹

¹(School of Computer and Information, Hefei University of Technology, Hefei 230009)

²(School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009)

Abstract Routers become subject to physical manufacture defects and running-time vulnerability in the deep submicron technology, which results in virtual channel permanent faults. The faults affect the performance and functionality of systems and result in communication malfunctions. In order to tolerate virtual channel faults effectively, and to ensure system performance and efficient usage of available resources, the type of failure is subdivided into coarse-grained fault and fine-grained fault, and then we propose the SVS router (single virtual channel sharing router) architecture to achieve a single virtual channel sharing between ports in the same group, which contains two ports in the router. Coarse-grained faults are tolerated by using adjacent ports' shared virtual channel in the same group. According to the information of Slot State Table, fine-grained faults are tolerated by configuring read/write pointer value to skip fault buffer slots. Also, in the absence of coarse-grained fault condition, shared virtual channel can be used for load balancing and fault tolerance of calculation module. Experimental results demonstrate significant reduction in average packet latency, and improvement in throughput under three different fault conditions compared with other existing virtual channel architectures. It shows that this scheme effectively improves system reliability, ensures system performance and makes full use of the available resources.

Key words virtual channel faults; coarse-grained faults; fine-grained faults; shared virtual channel; reliability

摘 要 深亚微米工艺下,路由器受制于制造缺陷及运行时的脆弱性,易发生虚通道(virtual channel, VC)永久性故障,从而引起通信故障,影响系统功能和性能.为了能够有效地容忍虚通道故障、保证系统性能及充分利用可用资源,将虚通道故障类型细分为粗粒度故障和细粒度故障,提出 SVS(single VC sharing)路由器架构,通过将路由器端口两两分组,组内端口间实现单虚通道共享.当发生虚通道粗粒

收稿日期:2013-08-09;修回日期:2013-12-19

基金项目:国家自然科学基金项目(61274036,61106038);安徽高校省级自然科学研究重点项目(KJ2010A269);安徽省科技攻关项目(11010202190)

通信作者:张一栋(zyd19891107@126.com)

度故障时,使用组内相邻端口共享虚通道容错.当发生细粒度故障时,根据 Slot State Table 信息配置虚通道读/写指针的值,从而跳过故障 Buffer 槽实现容错.在无粗粒度故障情况下,共享虚通道还可用于负载均衡及容忍路由计算模块故障.实验结果表明:较其他已有的虚通道路由器,SVS 路由器在 3 种不同的故障情况下均较大地降低了延时,提高了吞吐量.这表明 SVS 路由器可有效提高系统可靠性,保证了系统性能,充分利用了可用资源.

关键词 虚通道故障;粗粒度故障;细粒度故障;虚通道共享;可靠性

中图法分类号 TP302

随着电子元件特征尺寸的持续缩小,单个芯片上集成的 IP 核数目越来越多.为了满足片上多核及众核系统对高性能的需求,研究者们将片上网络^[1-4](network-on-chip, NoC)与三维封装技术相结合,提出了新的互连架构——三维片上网络^[5-6](three-dimension network-on-chip, 3D NoC).它通过用于层间通信的硅通孔(through silicon via, TSV)将多个晶片在垂直方向进行堆叠.有效地缩短了全局互连线长度,从而减少了延时,降低了功耗,提高了系统性能.

在 3D NoC 中路由器作为通信资源,负责数据的存储转发,确保其正常工作是保证系统性能的根本.但是由于制造缺陷、电路老化、工艺不稳定性和电子迁移等原因会使路由器中元件出现永久性故障^[7].文献[8]指出,路由器中故障数与元件面积成正比例关系,所含晶体管越多的元件越容易出现故障.实验表明,在 45 nm 工艺下,对于 5 端口的基准路由器,如果 FIFO 大小为 32-Flit,那么 FIFO 面积占路由器总面积的 94%.FIFO 大小为 8-Flit 时则占 80%.以上数据充分说明,在路由器面积中 FIFO 面积占绝大部分,因此 FIFO 模块包含更多的晶体管,也就更容易出现故障.因此对路由器 FIFO 容错至关重要.

面对上述问题国内外研究者纷纷展开了深入研究.文献[9]提出了路由器中通过内建自测试(built-in self test, BIST)和自诊断的方法.针对硬件故障的容错,其思路主要分为两大类:1)通过容错路由算法绕过故障节点.该方法能在不增加额外硬件开销的情况下实现容错,提高系统的可靠性.文献[10]提出了一种基于拓扑划分的快速映射方法,有效地降低网络中心流量.文献[11]提出了一种路由算法,能够有效地解决垂直链路故障问题,提高了系统可靠性,保证了系统性能.文献[12]提出了一种可重构的路由算法,该算法按照故障区域网络边界的相对位置对故障区分类,从而根据不同故障区类型定义具

体路由器状态更新策略以实现容错.2)通过更改路由器结构容错.该方法在能够保证性能的情况下提高系统的可靠性,但是可能会带来较大的面积和功耗开销.如文献[13-14]提出了部分虚通道共享(partial VC sharing, PVS)路由器结构,通过部分端口虚通道共享进行容错,能够对 FIFO 及路由计算等模块容错,同时也能在一定程度上平衡负载,但是该方法对 FIFO 故障粒度划分不够细致,造成了一定资源浪费.文献[8]提出了跳过 FIFO 中故障 Buffer 槽容错方案,该方法能够充分利用可用资源,实现了容错功能.但是一旦 FIFO 中 Buffer 槽故障数较多,剩余可用 Buffer 槽数量小于为满足相邻路由器间的信号往返延时所需最小数目时,依然保留 FIFO 功能显然不合理.

因此本文把虚通道故障类型分为粗粒度故障和细粒度故障,并提出了 SVS(single VC sharing)路由器,在组内端口间实现单虚通道(virtual channel, VC)共享.通过故障分析模块区分故障类型,发生虚通道粗粒度故障时,采用共享 VC 容错.发生虚通道细粒度故障时,通过控制读/写指针,跳过故障 Buffer 槽方式容错. SVS 路由器在实现上述容 VC 故障同时,还能对路由计算模块容错及在一定程度上平衡负载.通过本文方法能够有效地提高系统可靠性,保证系统性能,实现对资源的充分利用.

1 NoC 中虚通道路由器

虚通道路由器^[15]作为应用于 NoC 中的路由器基本架构被国内外学者广泛研究.虚通道路由器克服了虫孔路由器所面临的两大问题:1)由于端口竞争导致的通信效率低问题;2)由于资源竞争导致的死锁问题.其一般结构如图 1 所示,它有 P 个输入/输出端口,每个输入端口由 n 个输入缓冲队列(input buffer, IB)组成,每个输入缓冲队列对应一个虚通道.

输入缓冲模块 IB、路由计算模块(routing computing, RC)、交叉开关模块(crossbar)、虚通道分配器(virtual channel allocator, VA)和交叉开关分配器(switch allocator, SA)相互间协调工作,从而实现路由器存储转发数据的功能。

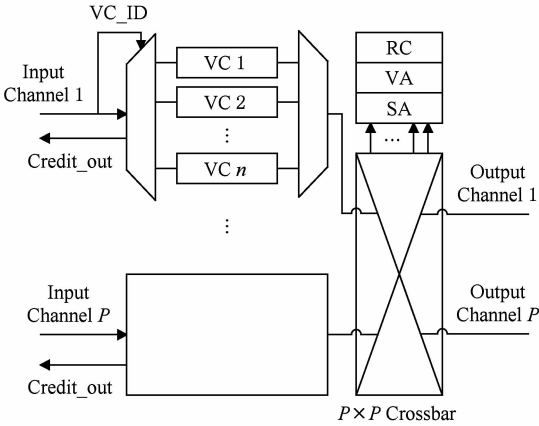


Fig. 1 General structure of the virtual channel router.
图 1 虚通道路由器的一般结构

在虚通道路由器中 n 个 VC 分时复用同一物理链路,每个 VC 对应一个 k -Flit 大小 Buffer,并由一个专属的 VC State Table 来控制数据的写入和读出.VC State Table 包含以下几项:VC 项为此 VC 的编号;WP 为写指针,指向一空的 Buffer 槽,作为多路选择器控制信号,从而写入数据;RP 为读指针,指向下一个要读出的数据,作为数据分配器的控制信号,从而读出数据;OP 为经过 RC 计算所确定的输出端口号;OVC 为经 VC 分配后所确定的输出 VC 号;CR 标记下游路由器可用存储空间大小;Status 域标记目前 VC 所处的阶段.具体结构如图 2 所示:

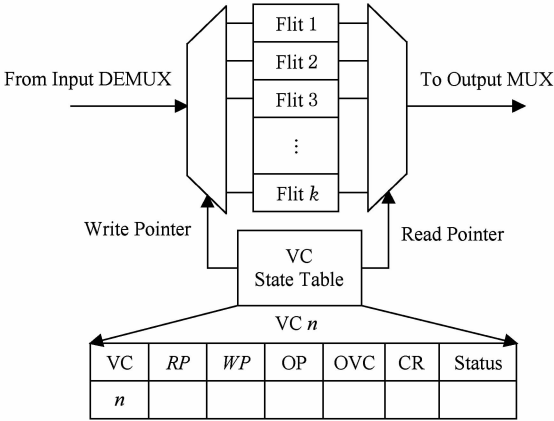


Fig. 2 Basic structure of the VC.
图 2 VC 的基本结构

2 故障模型

针对国内外关于路由器 VC 故障的研究现状,本文提出对路由器 VC 故障更为细致的划分,并假设路由器 VC 应有的最小空间为 N 个 Flit, N 大小由相邻路由器间信号往返延时决定。

1) 虚通道粗粒度故障(coarse-grained fault of VC):是指若虚通道中对于 K 个单 Flit 大小的 Buffer,其中有 M 个发生了故障,且 $K - M < N$,则弃用整个 VC。

由此可知虚通道粗粒度故障发生的概率为

$$P_{CF} = 1 - \left(\sum_{i=N}^K \binom{K}{i} (1-p)^i p^{K-i} \right), \quad (1)$$

其中, P_{CF} 为 VC 发生粗粒度故障概率, p 为单个 Flit 大小的 Buffer 出现故障的概率。

2) 虚通道细粒度故障(fine-grained fault of VC):是指若虚通道中对于 K 个单 Flit 大小的 Buffer,其中有 M 个发生了故障,且 $M \leq K - N$,则保留可用 Buffer。

由此可知虚通道细粒度故障发生的概率为

$$P_{FF} = \sum_{i=1}^{K-N} \binom{K}{i} p^i (1-p), \quad (2)$$

其中, P_{FF} 为 VC 发生细粒度故障概率, p 为单个 Flit 大小的 Buffer 出现故障的概率。

3 SVS 路由器容错原理

本文综合考虑路由器输入 FIFO 发生粗粒度故障和细粒度故障情况,提出了 SVS 路由器设计方法.通过在东西、南北、上下 3 组内分别实现端口间单虚通道共享,实现对虚通道粗粒度故障及路由计算逻辑故障容错,并且在一定程度上平衡端口间通信负载.对于发生虚通道细粒度故障,借鉴文献[8]提出的通过重配置读/写指针来跳过故障 Buffer 槽的方案容错。

3.1 SVS 路由器的结构

SVS 路由器将传统 3D NoC 中路由器的 7 个端口分为 4 组:东西、南北、上下、本地.对于前 3 组,分别实现组内端口间单虚通道共享.其中东、西端口间单虚通道共享结构如图 3 所示。

通过在东输入端口 VC n 与西输入端口 VC 1 之前分别增加一组数据分配器和多路选择器,从而实现这两个 VC 间的相互共享.其具体的控制逻辑

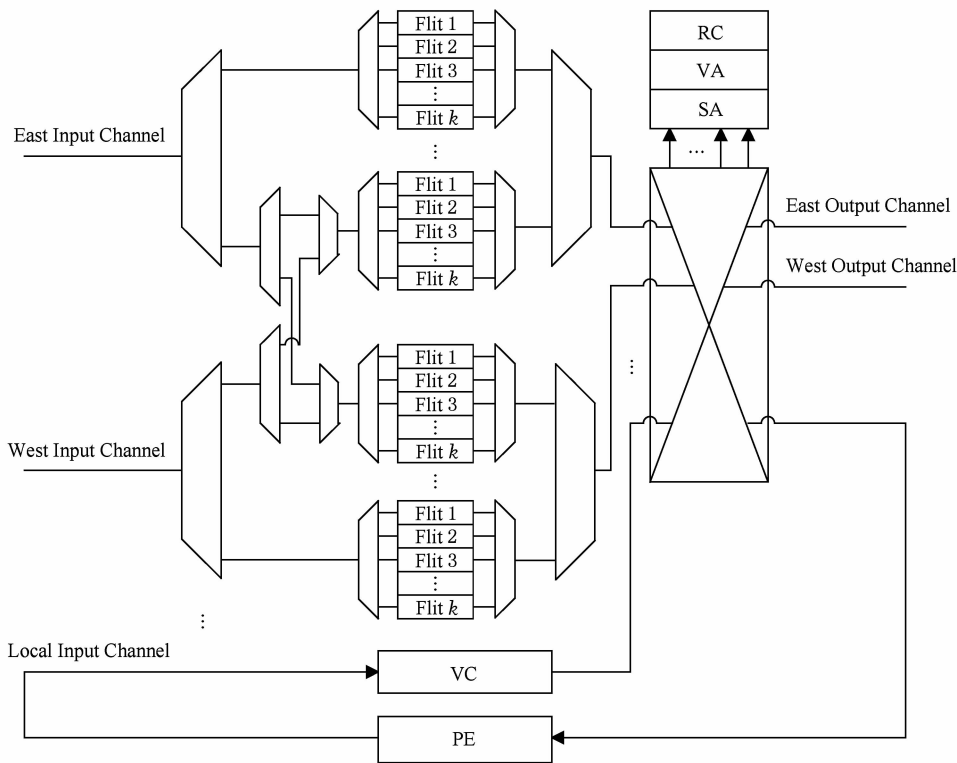


Fig. 3 Single virtual channel sharing router.
图3 单虚通道共享的容错路由器

和容错策略如图 4 所示. SVS 路由器为实现相邻端口间单虚通道共享所需的控制逻辑主要由故障分析器(Fault Analyzer)模块和缓冲区分配器(Buffer Allocator)模块构成. 系统工作在两种不同的模式下,分别为测试模式和工作模式. 在进入工作模式之前,首先工作在测试模式下,由系统向 BIST 模块发出启动信号,启动 BIST 机制对路由器 VC 进行故障检测. 当测试结束后,BIST 模块将测试结果传送给 Fault Analyzer 模块,由此模块对相应的故障情况进行分析,判断故障模型,从而启动不同的容错方案.

1) Fault Analyzer: 系统在结束测试后,Fault Analyzer 模块会接收到来自 BIST 发送的 VC 测试结果,然后根据此信息判断东、西两端口各个 VC 中故障槽的数目,从而判定故障类型. 如果分析结果为某个 VC 发生了细粒度故障,则 Fault Analyzer 模块会使其与该虚通道 VC State Table 相连的 x_slot_F (其中 $x \in \{E, W\}$) 信号有效(x_slot_F 为 n 位信号线,每一位对应相应的 VC,若某个 VC 出现细粒度故障,则相应位被置 1,否则为 0),从而启动虚通道细粒度故障容错方案. 如果分析结果为某个 VC 发生了粗粒度故障,则 Fault Analyzer 模块向上游路由器及 Buffer Allocator 模块发送 x_VC_F (其中 $x \in \{E, W\}$) 有效信号(x_VC_F 为 n 位信号线,每

一位对应相应的 VC,若某个 VC 出现粗粒度故障,则相应位被置 1,否则为 0),告知上游路由器此 VC 发生粗粒度故障,并通过 Buffer Allocator 模块配置路由器进行容错.

2) Buffer Allocator: 此模块的输入信号为 x_VC_F (其中 $x \in \{E, W\}$) 及来自于东、西输入端口数据包的 SVC 域和 VC_ID 域信息. 其中 x_VC_F 表示 x 端口发生了虚通道粗粒度故障,为 n 位信号线,每一位表示相应 VC 是否故障;SVC 域表示此数据包是否将被送入相邻端口的共享虚通道,如果是则为 1,否则为 0;VC_ID 表示数据包将被送往的 VC 的编号. Buffer Allocator 根据 x_VC_F 和 SVC 域来配置路由器,使数据包存入理想的 VC 中. 具体情况如下:

① x_VC_F 无效,数据包 SVC 域被置 0, Buffer Allocator 根据数据包 VC_ID 域信息配置一组 x_sel_y (其中 $x \in \{E, W\}, y \in \{1, 2, 3\}$) 信号,使数据包进入所分配到的 VC.

② x_VC_F 有效,但数据包所申请的 VC 无故障,那么数据包 SVC 域被置 0. 此时 Buffer Allocator 根据数据包 VC_ID 域信息配置一组 x_sel_y 信号,使数据包进入所分配到的 VC.

③ x_VC_F 有效,且数据包申请的 VC 恰为故

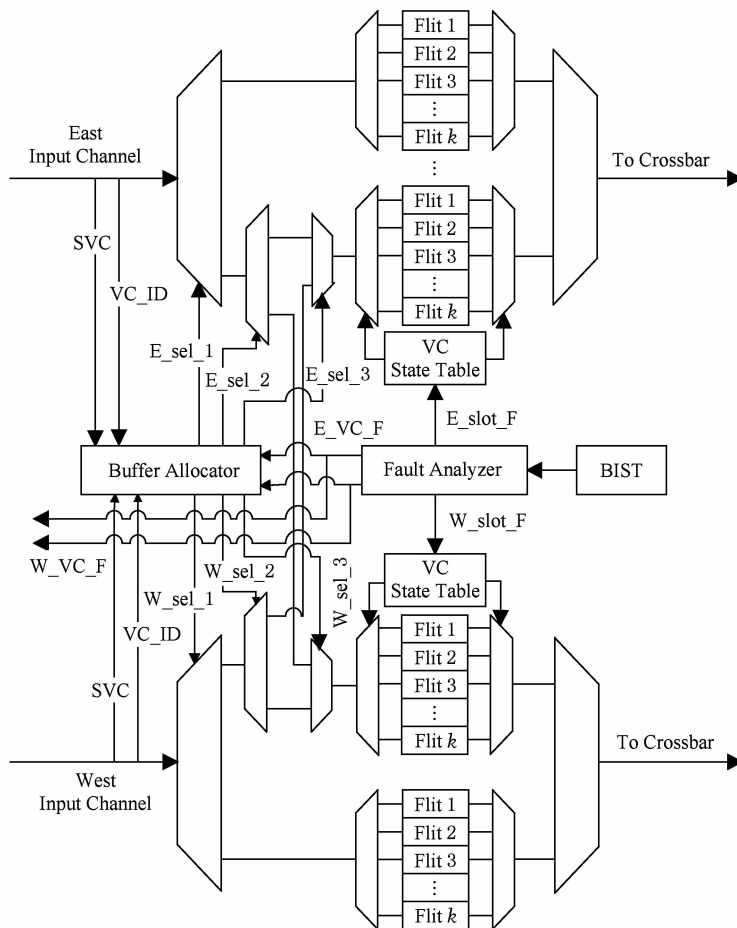


Fig. 4 Control logic of SVS router.

图 4 SVS 路由器控制逻辑

障 VC, 那么数据包 SVC 域被置 1, 此时 Buffer Allocator 不再理会数据包 VC_ID 域的信息, 而是通过配置一组 x_sel_y 信号分配相邻端口共享 VC 给它。

3.2 VC 故障容错方案

在片上网络路由器中 FIFO 面积占据了总面积的绝大部分, 因此其所含晶体管数目越大也就越易出现故障。本文在研究了路由器 VC 故障对系统性能的影响及如何提高 FIFO 资源利用率的基础上, 根据故障程度不同, 分为虚通道粗粒度故障和细粒度故障, 并相应提出不同的容错方案。

1) 虚通道的粗粒度故障。路由器中某个 VC 一旦出现粗粒度故障, 就意味着可用的 Buffer 槽数不能满足相邻路由器间信号往返延时, 系统将弃用整个 VC。针对此问题, 本文采用单虚通道共享路由器中的共享 VC 来容错。以东端口 VC 故障为例, 当路由器东端口某个 VC 出现故障, 系统通过 BIST 模块检测出故障, 并将结果传送给 Fault Analyzer 模块进行故障类型分析, 在确定了故障类型后, Fault

Analyzer 模块驱动相应信号有效从而启动不同的容错方案。本例中 Fault Analyzer 模块会使 E_VC_F 信号相应位有效, 此信号将传向上游路由器及 Buffer Allocator 模块。上游路由器在收到 E_VC_F 信号后, 将对所有请求故障 VC 的数据包的 SVC 域进行更新, 将其置 1。路由器根据 W 端口共享 VC 的状态决定是否响应数据包的请求, 若共享 VC 空则响应请求, 数据包将向本级路由器传输, 否则数据包暂存上游路由器, 等待共享 VC 空闲。当数据包送往本级路由器时, Buffer Allocator 将截取 SVC 和 VC_ID 域信息, 并据此配置路由器。本例中, SVC 为 1, VC_ID 为故障 VC 的编号, 故将 E_sel_1, E_sel_2 置 1, 将 W_sel_3 置 0。这样数据将被存入 W 端口共享 VC 中, 实现了虚通道粗粒度故障容错。

2) 虚通道的细粒度故障。VC 若只是发生细粒度故障, 则弃之不用, 通过共享 VC 容错的方案显然不可取, 因为这必将造成系统中有限且宝贵资源的极大浪费。针对此问题, 本文借鉴文献[8]提出的通过重配置读/写指针跳过故障 Buffer 槽的方法容错,

并提出了低开销的实现方案. 本方案中每个 VC 都维护一个 Slot State Table, 以表征某时刻 VC 中 Buffer 槽的故障及空闲状态. 表中含有 3 项, 其中 Slot 项表示 VC 中 Buffer 槽的编号, 为 Slot 1→Slot k ; State 项则用一位二进制数表示 Buffer 槽空闲状态, 如果槽中有数据则为 1, 空闲为 0; Fault 项表示 Buffer 槽是否故障, 故障为 1, 否则为 0. 对于良好的 Buffer 槽当数据写入时, 系统会将其状态更新为 1, 一旦数据被读出, 系统则将其状态恢复为 0. VC 中 Buffer 槽以循环队列的方式组织先进先出的读写数据. 其中 RP 始终指向的是一个存有数据的槽, 当读出当前槽中数据后, 指针将根据 Fault 项信息指向最近一个无故障的 Buffer 槽, 如果此时恰好 $RP = WP$, 则表示整个 VC 为空. WP 始终指向的是下次

能写入数据的槽, 当向 WP 指向的槽中写入数据后, 指针将根据 Fault 项信息指向最近一个无故障且为空的槽, 如果 $WP \% k + 1 = RP$ 表示整个 VC 为满. 图 5 表示 VC 中 Slot 4 故障时的容错情况. 当 Slot 2 中数据被读出后, 系统根据 Slot State Table 信息更新 VC State Table, 查表后发现 Slot 3 Fault 域为 0 且 State 域为 1, 则 RP 指向 Slot 3 并可读出其中数据. 数据被读出后, 系统查 Slot 4 Fault 域发现其值为 1, 这时 RP 将跳过故障槽指向 Slot 5. 指针 WP 的更新也是如此, 如果指向的槽发生故障, WP 将跳过该故障槽指向下一个 Fault 和 State 域都为 0 的槽. 根据 State 和 Fault 域信息, 系统控制器控制着 RP 和 WP 的动态更新, 以跳过故障槽的方式实现容错.

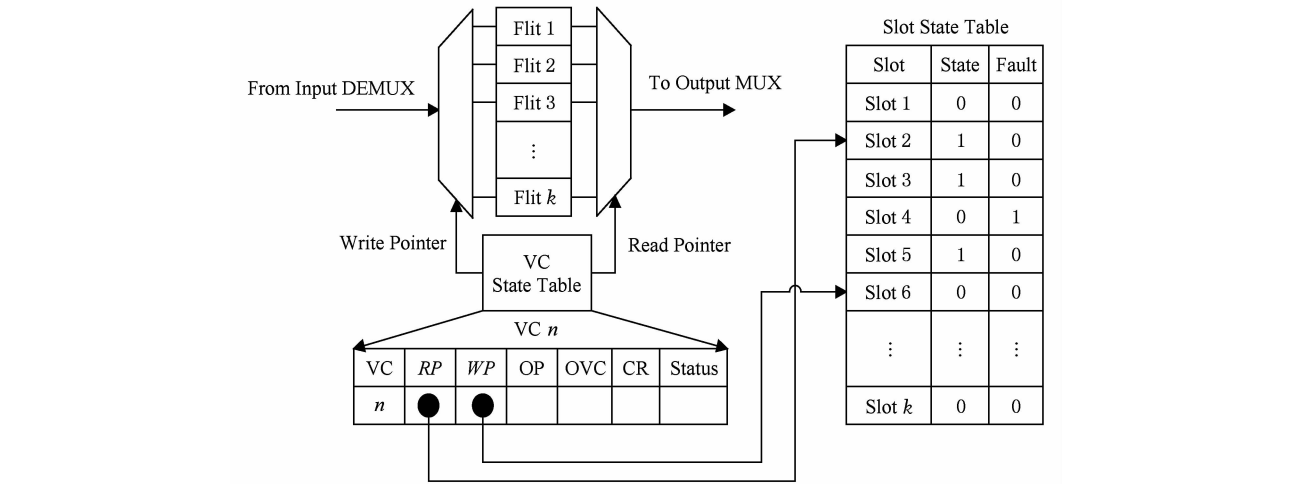


Fig. 5 Fault tolerance of Slot 4.

图 5 Slot 4 故障容错

3.3 SVS 路由器共享方式的额外优势

本文所提出的 SVS 路由器通过端口间单虚通道共享的方式, 不仅能够容虚通道粗粒度故障, 而且 VC 共享的方式还具有以下额外优势:

- 1) 若某个端口路由计算模块(RC)出现故障, 那么数据即使存入当前端口, 由于无法进行路由计算也无法传出去. 此时将数据传入相邻端口的共享 VC, 利用相邻端口 RC 模块进行路由计算, 从而实现了对端口 RC 模块的容错.
- 2) 通过共享 VC 可以在一定程度上缓解端口的拥塞. 如某时刻东数据端口数据流量很大, 没有足够的缓存空间, 而这时恰好西端口数据流量小, 有足够的缓存空间空闲, 那么数据可以通过西端口的共享 VC 传输, 从而平衡了网络负载, 这要一定程度上缓解了拥塞.

本文将 VC 故障类型合理地划分为粗粒度故障和细粒度故障, 并通过不同方案对两种故障进行容错, 提高了可靠性, 保证了系统性能, 充分利用了系统资源.

4 实 验

本文采用的仿真工具是在 Booksim^[16]基础上扩展的 3D NoC 仿真器. 在此平台上搭建 $4 \times 4 \times 3$ 的 3D Mesh 结构的 NoC. 在不同的故障分布情况下, 比较不同容错方案的性能及不同方案对 Buffer 资源利用率. 方案 1: 采用传统路由器. 方案 2: 采用文献[13]提出的 PVS 路由器容错方法. 方案 3: 本文提出的 SVS 路由器容错方法. 在配置环境时, 考虑以下 3 种故障分布: 1) Buffer 故障分布使网络中

只存在虚通道粗粒度故障;2)Buffer 故障分布使网络中只存在虚通道细粒度故障;3)Buffer 故障分布使网络两种类型故障并存。

1) 系统性能

实验分别配置 3 种不同的故障分布环境,采用均匀随机模式,对 3 种不同方案的平均延时和吞吐率作了模拟分析.平均延时和吞吐率是衡量系统性能的重要指标.其中平均延时反映了数据包从进入网络到其尾部离开网络的这段时间差的平均值.吞吐率反映了单位时间网络接收或者发送的消息量,网络达到饱和时,该指标的值越大说明网络性能越好.图 6、图 7 分别为网络中只存在虚通道粗粒度故

障时的网络平均延时和吞吐率的示意图。

图 6、图 7 中的曲线走向表明,在当前故障情况下,采用传统路由器方案延时急剧增大,吞吐率较低.在注入率为 0.12 时便出现严重的网络拥塞,系统达到饱和.而本文提出的 SVS 路由器及文献[13]提出的 PVS 路由器性能基本相当,且优于方案 1,其中注入率达到 0.22 时,延时才急剧增加,网络才出现饱和.这说明两种方案都有效实现了容错,且效果相差不大,这是因为在出现虚通道粗粒度故障时,两种方案都采用了共享 VC 方式容错.图 8、图 9 为网络只存在虚通道细粒度故障时的网络平均延时和吞吐率。

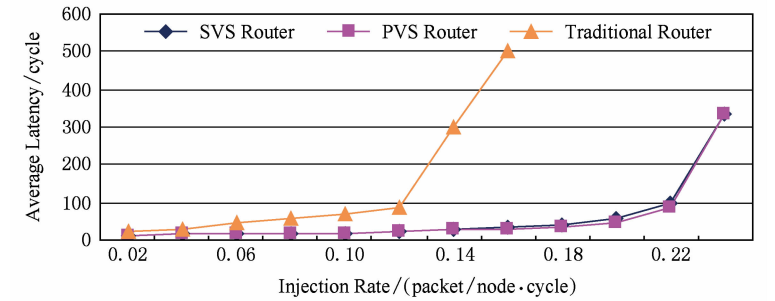


Fig. 6 Average latency under coarse-grained fault of VC.

图 6 虚通道粗粒度故障下平均延时

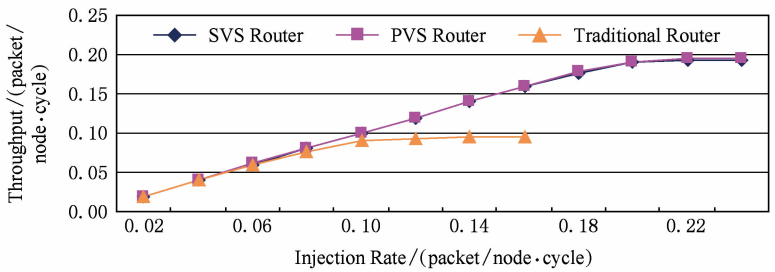


Fig. 7 Throughput under coarse-grained fault of VC.

图 7 虚通道粗粒度故障下吞吐率

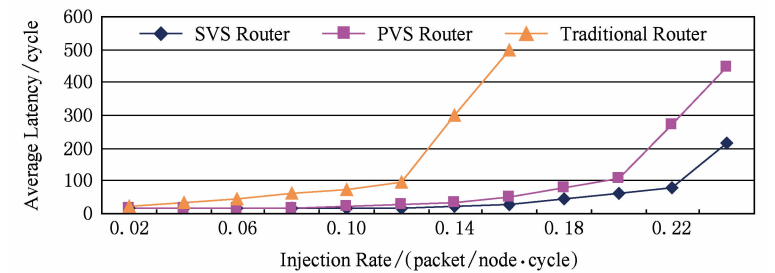


Fig. 8 Average latency under fine-grained fault of VC.

图 8 虚通道细粒度故障下平均延时

图 8、图 9 中的曲线走向表明,本文方案在延时和吞吐率方面都优于其他两种方案.当网络注入率为 0.12 时,本文方案较文献[13]延时约降低了 27.5%,注入率为 0.2 时延时约降低了 42.8%.曲线图表明,在网络注入率为 0.2 时,文献[13]延时急剧增加,说明网络出现拥塞,且达到了饱和点,而本

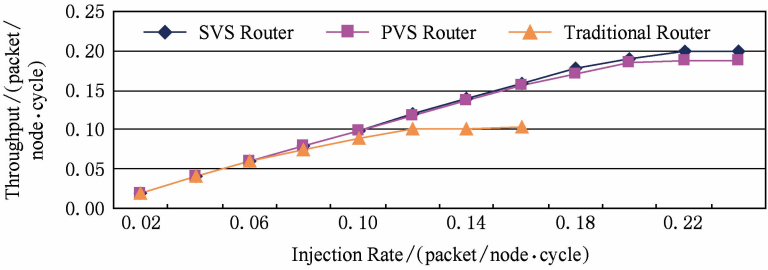


Fig. 9 Throughput under fine-grained fault.
图9 虚通道细粒度故障下吞吐率

文方案直到 0.22 时才出现此种情况. 这是因为对于虚通道细粒度故障本文采用的是通过控制读/写指针方式容错, 控制逻辑较文献[13]简单, 且能充分利用故障 VC 中剩余完好的 Buffer 槽, 因此注入率越

大对资源需求越大, 文献[13]方案也就越容易出现拥塞, 越早地达到饱和. 图 10、图 11 为网络中两种类型故障并存时的网络的平均延时和吞吐率.

图 10、图 11 中的曲线走向表明, 本文方案在延

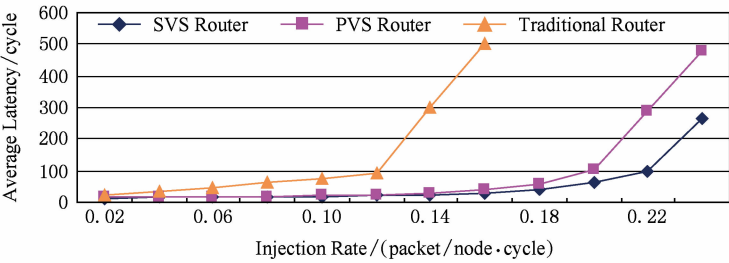


Fig. 10 Average latency under mixed failure.
图10 混合故障下平均延时

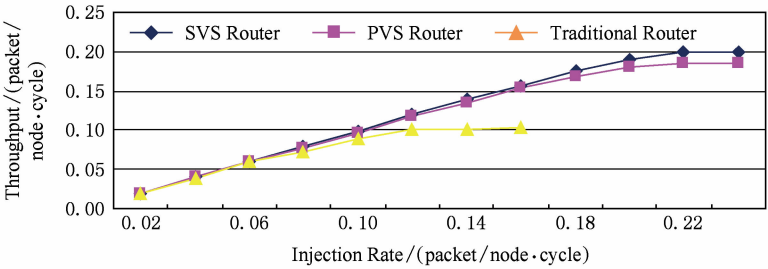


Fig. 11 Throughput under mixed failure.
图11 混合故障下吞吐率

时和吞吐率方面也优于其他两种方案. 当注入率为 0.12 时, 本文方案较文献[13]延时约降低 16.7%, 注入率为 0.2 时约降低了 40.1%. 文献[13]注入率约为 0.19 时便达到了饱和点, 本文方案直到 0.21 时才出现饱和.

2) 面积开销

本文使用 Orion2.0 实验工具在 65 nm 的工艺下, 对 3 种方案路由器的面积和功耗进行仿真. 路由器每个端口有 4 个 VC, 每个 VC 深度为 8-Flit. 实验结果如表 1 所示.

数据表明, 本文提出的 SVS 路由器面积、功耗较传统路由器约增加 8%, 且仅比 PVS 路由器面积、

功耗略高.

实验表明, 本文通过对 VC 故障类型细分, 提出 SVS 路由器容错架构. 此方案在较小的面积、功耗开销情况下, 实现了对 VC 不同类型故障的容错, 降低了延时, 增大了吞吐率, 实现了 Buffer 资源利用率最大化.

Table 1 Area and Power Comparison
表1 面积和功耗的比较

Router	Area/ μm^2	Power Consumption/mW
Traditional Router	236 703	68.71
PVS Router	243 803	72.56
SVS Router	255 639	74.33

5 总 结

随着集成电路工艺水平的提高,可靠性成为 3D NoC 研究重点. 本文通过对 VC 故障类型的研究提出了可行的容错方案. 具有以下 4 点贡献:1)根据故障的分布情况,将 VC 故障分为粗粒度故障和细粒度故障,并提出不同容错方案;2)通过 SVS 路由器中的共享 VC 容虚通道粗粒度故障;3)通过 Slot State Table 的信息动态配置读/写指针容虚通道细粒度故障;4)充分利用发生细粒度故障 VC 中可用的 Buffer 资源. 实验表明:在面积、功耗少量增加的情况下,本文延时较文献[13]降低了约 16%~43%,吞吐率也具有优势,且当网络中发生虚通道细粒度故障较多时尤为明显.

参 考 文 献

[1] Dally W J, Towles B. Route packets not wires: On-chip interconnection networks [C] //Proc of the 38th Int Conf on Design Automation. New York: ACM, 2001: 684-689

[2] Benini L, Micheli G D. Networks-on-Chips: A new SoC paradigm [J]. Computer, 2002, 35(1): 70-78

[3] Gueerrier P, Greiner A. A generic architecture for on-chip packet-switched interconnections [C] //Proc of Int Conf on Design, Automation and Test in Europe (DATE'00). New York: ACM, 2000: 250-253

[4] Zhu Honglei, Peng Yuanxi, Yin Yaming, et al. A NoC router with dynamically allocated virtual-output-queueing [J]. Journal of Computer Research and Development, 2012, 49(1): 183-192 (in Chinese)
(朱红雷, 彭元喜, 尹亚明, 等. 一种动态分配虚拟输出队列结构的片上路由器[J]. 计算机研究与发展, 2012, 49(1): 183-192)

[5] Liu Cheng, Zhang Lei, Han Yinhe, et al. Vertical interconnects squeezing in symmetric 3D Mesh Network-on-Chip [C] //Proc of the 16th Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2011: 357-362

[6] Owens J D, Dally W J, et al. Research challenges for on-chip interconnection networks [J]. IEEE Micro, 2007, 27(5): 96-108

[7] Ouyang Yiming, Ni Jinzhao, Liang Huaguo. Design of network-on-chip groupware based on hamming code and built-in self-test [J]. Journal of Applied Sciences, 2010, 28(5): 501-526 (in Chinese)
(欧阳一鸣, 倪晋照, 梁华国. 应用海明码和内建自测试技术设计片上网络组建[J]. 应用科学学报, 2010, 28(5): 501-526)

[8] DeOrio A, Fick D, et al. A reliable routing architecture and algorithm for NoCs [J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(5): 726-739

[9] Ouyang Yiming, Cheng Lili, Liang Huaguo. A new test data compression technique based on static relativity of variable length data block [J]. Acta Electronica Sinica, 2008, 36(12): 298-302 (in Chinese)
(欧阳一鸣, 成丽丽, 梁华国. 一种基于变长数据块相关性统计的测试数据压缩和解压方法[J]. 电子学报, 2008, 36(12): 298-302)

[10] Deng Zhi, Gu Huaxi, Yang Yintang, et al. A fast topology partition based mapping algorithm for NoC [J]. Journal of Electronics & Information Technology, 2011, 33(12): 3028-3034 (in Chinese)
(邓植, 顾华玺, 杨银堂, 等. 基于拓扑划分的片上网络快速映射算法[J]. 电子与信息学报, 2011, 33(12): 3028-3034)

[11] Rahmani A M, Latif K, Liljeberg P, et al. A stacked Mesh 3D NoC architecture enabling congestion-aware and reliable inter-layer communication [C] //Proc of the 19th Int Euromicro Conf. Piscataway, NJ: IEEE, 2011: 423-430

[12] Fu Binzhang, Han Yinhe, Li Huawei, et al. Building resilient NoC with a reconfigurable routing algorithm [J]. Journal of Computer-Aided Design & Computer Graphics, 2011, 23(3): 448-455 (in Chinese)
(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)

[13] Latif K, et al. A novel topology-independent router architecture to enhance reliability and performance of Networks-on-Chip [C] //Proc of the 26th IEEE Int Symp on Defect and Fault Tolerance in VLSI and Nanotechnology Systems. New York: IEEE, 2011: 454-462

[14] Latif K, Rahmani A M, Liang Guang, et al. PVS-NoC: Partial virtual channel sharing NoC architecture [C] //Proc of the 19th Int Euromicro Conf. Piscataway, NJ: IEEE, 2011: 470-477

[15] Park D, Eachempati S, Das R, et al. MIRA: A multi-layered on-chip interconnect router architecture [J]. Journal of ACM SIGARCH Computer Architecture News, 2008, 36(3): 251-261

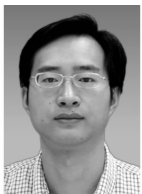
[16] Becker D, James C. BookSim interconnection network simulator [OL]. (2010-09-11)[2012-06-05]. <https://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/BookSim>



Ouyang Yiming, born in 1963. PhD and associate professor at Hefei University of Technology. Senior member of China Computer Federation. His main research interests include synthesis and testing of embedded systems, fault tolerance of network-on-chip, and design automation of digital systems (oyymbox@163.com).



Zhang Yidong, born in 1989. Master candidate at Hefei University of Technology. His main research interests include NoC architecture and VLSI design.



Huang Zhengfeng, born in 1978. PhD and associate professor at Hefei University of Technology. His research interests include synthesis and testing of embedded systems, design automation of digital systems (hanson.hfut@gmail.com).



Liang Huaguo, born in 1959. Professor and PhD supervisor at Hefei University of Technology. Senior member of China Computer Federation. His main research interests include synthesis and testing of embedded systems, built-in self test, ATPG algorithms, and distributed control (huagulg@hfut.edu.cn).



Chang Hao, born in 1983. PhD candidate in the School of Computer and Information, Hefei University of Technology. His research interests include the testing of 3D IC, built-in self test, and fault tolerance (007changhao@163.com).

《计算机研究与发展》征订启事

《计算机研究与发展》(Journal of Computer Research and Development)是中国科学院计算技术研究所和中国计算机学会联合主办、科学出版社出版的学术性刊物,中国计算机学会会刊. 主要刊登计算机科学技术领域高水平的学术论文、最新科研成果和重大应用成果. 读者对象为从事计算机研究与开发的研究人员、工程技术人员、各大专院校计算机相关专业的师生以及高新企业研发人员等.

《计算机研究与发展》于 1958 年创刊,是我国第一个计算机刊物,现已成为我国计算机领域权威性的学术期刊之一. 并历次被评为我国计算机类核心期刊,多次被评为“中国百种杰出学术期刊”. 此外,还被《中国学术期刊文摘》、《中国科学引文索引》、“中国科学引文数据库”、“中国科技论文统计源数据库”、美国工程索引(Ei)检索系统、日本《科学技术文献速报》、俄罗斯《文摘杂志》、英国《科学文摘》(SA)等国内外重要检索机构收录.

国内邮发代号:2-654;国外发行代号:M603

国内统一连续出版物号:CN11-1777/TP

国际标准连续出版物号:ISSN1000-1239

联系方式:

100190 北京中关村科学院南路 6 号《计算机研究与发展》编辑部

电话: +86(10)62620696(兼传真);+86(10)62600350

Email:crad@ict.ac.cn

http://crad.ict.ac.cn