DOI:10.11784/tdxbz202504017

基于端口粒度故障定级管理的 NoC 容错架构设计

史再峰[1]，刘鑫涛[1,2]，张熙宇[1,2]，罗　韬[3]
（1. 天津大学微电子学院，天津 300072；2. 飞腾信息技术有限公司，长沙 410073；
3. 天津大学智能与计算学部，天津 300072）

摘　要：随着半导体工艺持续演进至深亚微米节点，片上网络关键组件在高密度集成环境下，面临日益严峻的物理缺陷与电噪声干扰，故障发生概率显著上升. 现有容错机制在处理多类型并发故障时，常因故障模式识别精度不足，导致资源利用率偏低与通信性能下降. 针对上述问题，本文提出一种基于端口粒度的故障定级管理容错架构，设计三级协同故障管控机制. 首先，引入信用返还标识确认机制，实现链路层端口的亚周期级实时故障检测，能够精确定位并快速上报报文丢失与数据校验错误，显著提高故障检测准确率与响应速度；其次，设计轻量化备份缓冲区并融合优先级调度策略，支持故障报文的快速跨步重传，有效降低重传延迟与带宽开销；最后，基于故障状态机模型动态评估端口故障等级，实现通信资源自适应调度，进一步提升系统整体资源利用率. 此外，该架构集成了协同容错路由算法，可快速识别瞬态故障并实现端口重启，同时对永久性故障端口实施智能隔离与动态路径绕行，从而减少冗余重传并降低容错操作带来的带宽损失. 实验结果表明，在多类故障并发的合成流量场景下，本文所提架构的饱和吞吐率较 FT-E2E 容错方案最高可提升 41.6%，较 EsyTest 容错方案最高可提升 26.2%，实现了系统可靠性与通信性能的协同优化.

关键词：片上网络；容错架构；故障定级；可靠通信
中图分类号：TN431.2　　**文献标志码**：A　　**文章编号**：0493-2137（2026）05-0496-11

Design of NoC Fault-Tolerant Architecture Based on Port-Granularity Fault Grading Management

Shi Zaifeng[1]，Liu Xintao[1,2]，Zhang Xiyu[1,2]，Luo Tao[3]
（1. School of Microelectronics，Tianjin University，Tianjin 300072，China；
2. Phytium Technology Co.，Ltd.，Changsha 410073，China；
3. College of Intelligence and Computing，Tianjin University，Tianjin 300072，China）

Abstract：As semiconductor technology continues to advance into deep sub-micron nodes，key components of network-on-chip（NoC）face increasingly severe physical defects and electrical noise in highly integrated environments，leading to a significant rise in the probability of failures. Existing fault-tolerant mechanisms often struggle with multiple concurrent fault types due to insufficient fault-pattern recognition，resulting in low resource utilization and degraded communication performance. To address these challenges，this paper proposes a port-granularity fault grading management architecture，which features a three-level cooperative fault-control mechanism. First，a credit-return identification and acknowledgment mechanism is introduced to achieve sub-cycle real-time fault detection at the link-layer port，enabling accurate localization and timely reporting of packet loss and data-corruption errors，thereby significantly improving fault-detection accuracy and response speed. Second，a lightweight backup buffer combined with a priority scheduling strategy supports fast hop-by-hop retransmission of faulty packets，effectively reducing retransmission latency and bandwidth overhead. Third，a fault-state machine model dynamically assesses port-fault

收稿日期：2025-04-14；**修回日期**：2025-08-06.
作者简介：史再峰（1977—　　），男，博士，教授.
通信作者：史再峰，shizaifeng@tju.edu.cn.
基金项目：天津市科技计划资助项目（22JCYBJC00140）.
Supported by the Science and Technology Plan Program of Tianjin，China（No. 22JCYBJC00140）.

levels and enables adaptive scheduling of communication resources，further enhancing overall system resource utilization. Additionally，the architecture integrates a cooperative fault-tolerant routing algorithm capable of rapid identification and recovery from transient faults，along with intelligent isolation and dynamic rerouting for permanent faults，which reduces redundant retransmissions and minimizes the bandwidth loss introduced by fault-tolerant operations. Experimental results show that，under synthetic traffic scenarios with multiple concurrent fault types，the proposed architecture achieves up to 41.6% higher saturation throughput compared to the FT-E2E fault-tolerant scheme and up to 26.2% higher than the EsyTest scheme，thereby achieving a synergistic optimization of system reliability and communication performance.

Keywords：network-on-chip（NoC）；fault-tolerant architecture；fault grading；reliable communication

随着核心集成度持续提升，传统总线架构在通信效率与可扩展性方面的局限性日益凸显[1]. 片上网络（network-on-chip，NoC）凭借其拓扑灵活性与高效的并行通信能力，已成为多核系统级芯片（multi-processor system-on-chip，MPSoC）中的主流互连方案[2]. 在自动驾驶、医疗电子等关键应用场景中，NoC 的可靠性直接影响系统功能安全. 然而，工艺尺寸持续缩减导致制造缺陷、器件老化效应以及信号完整性恶化，显著增加了 NoC 的故障风险[3]. 同时，互连结构日益复杂也扩大了单点故障的传播范围，进一步威胁系统稳定性.

NoC 中的故障主要可分为瞬态故障、永久性故障和间歇性故障 3 类. 可靠性保障需针对不同故障特性制定差异化的容错机制[4]，瞬态故障常由粒子轰击等环境因素引发，其纳秒级扰动易造成数据比特翻转，引发报文（packet）损坏或路由状态异常. 文献[5]提出基于改进型汉明码的并行检测与纠错机制，提升了长报文场景下的容错处理效率. 文献[6]通过纠错码与检错码协同设计，将错误检测与校正嵌入同一流水线阶段，优化了处理时延. 然而，此类方案的纠错能力受限于编码冗余度，面对多比特错误仍依赖端到端重传，系统延迟随故障频次线性上升. 文献[7]的泛洪式传输和文献[8]的随机行走路由通过多路径冗余规避故障区域，虽能提高系统鲁棒性，却显著增加了带宽开销与能耗负担.

永久性故障通常由热载流子注入[9]和电迁移效应[10]等物理劣化机制引发，具有不可逆性. 针对此类故障，文献[11-12]提出基于内建自测试（built-in self-test，BIST）的方法，通过伪随机测试向量实现链路的周期性检测，但测试过程对硬件资源和通信时隙的占用成为关键瓶颈. 文献[13-15]通过备用路由器和链路进行容错替代，但在无故障情况下相关资源处于闲置状态，利用率较低. 文献[16]引入强化学习方法训练智能代理实现路径规避，增强了策略智能性. 文献[17]则结合了确定性路由与 Tarjan 算法，构建可调路径以实现容错路由. 针对 2.5-D NoC，文献[18]提出 ReD 算法，基于虚拟网络分离策略动态平衡垂直链路负载，并借助哈密顿路径传播故障状态信息，提升网络路由可达性. 上述方案在增强永久性故障容错能力方面各具优势，但仍存在资源利用效率低、实现复杂度高和系统开销显著等问题.

间歇性故障多源于栅氧缺陷等工艺波动，表现为无规律触发和非持续性异常，兼具瞬态故障的可恢复性与永久性故障的可复现特征. 传统 BIST 方法难以准确识别其触发模式，易导致误判或漏检[19]. 文献[20]提出基于双层子网复制的冗余传输机制以增强容错能力，尽管在可靠性方面有所提升，但也引入了额外的硬件与带宽开销. 文献[21]则设计了基于虚通道备份的间歇性故障处理机制，在发生故障时为已通过故障链路的报文插入伪尾分片（flit），而未通过部分则重新注入头 flit，以实现路径重构和资源释放. 但该方案在面对高频、短周期的间歇性故障时，响应延迟较高，容错效率有限.

随着工艺节点的持续演进，不同类型故障之间的耦合效应日益显著：瞬态故障可能演化为间歇性故障，而长期未修复的间歇性故障也可能发展成永久性损伤. 这种动态演化特性对容错设计提出了更高要求：一方面，需具备精确的故障识别能力，避免因对短时扰动的过度响应造成资源浪费；另一方面，必须能够及时识别并隔离永久性故障，保障系统通信的稳定性与可靠性. 然而，现有容错方案在应对瞬态、间歇性和永久性故障交织出现的复杂故障场景时，仍存在明显的局限性，难以满足实时检测、精准分类和差异化处理的综合需求.

为解决上述问题，本文提出一种基于端口粒度的故障定级管理容错架构（port-granularity fault grading management-based NoC，PFGM-NoC）. 该架构引入增强型信用流控与标识确认机制，实现多类故障的实时监测与精确定位，快速上报异常端口并触发重传，提升通信可靠性与数据完整性. 在硬件层面，PFGM-

NoC 为每个端口配置轻量化备份缓冲区队列. 故障发生时控制器可即时切换至备份队列,并通过高优先级跳步重传机制完成故障数据的快速转发,显著降低重传时延. 此外,该架构集成故障状态机,对输出端口的故障等级进行动态评估:对于瞬态故障,可快速重启恢复通信;而针对持续性或永久性故障,则通过智能隔离机制防止故障扩散,保障系统在复杂故障场景下的高效稳定运行.

1 容错架构和算法

1.1 PFGM-NoC 路由器架构

基准 NoC 路由器的微架构主要由输入缓冲区、路由计算单元、虚拟通道分配器、交换仲裁器以及物理交叉开关等构成[22]. 输入缓冲区用于暂存来自不同方向的报文,以缓解端口速率不匹配引发的拥塞,降低数据丢失风险. 路由计算单元根据报文目的地址及网络拓扑状态,动态决策下一跳方向,实现故障链路规避. 虚拟通道分配器将报文映射至不同虚拟通道,避免数据流之间的干扰,有效抑制死锁和队首阻塞现象. 交叉开关(包括交换分配器与物理交换矩阵)负责动态构建输入端口与输出端口之间的传输路径,实现高吞吐量的数据转发.

图 1 展示了 PFGM-NoC 路由器的改进微架构. 本文依据功能特性,将 NoC 故障类型划分为控制模块故障与数据传输通道故障,并分别设计相应的容错机制. 控制模块涵盖路由计算单元、虚拟通道分配器和交叉开关分配器,其故障可能引发路由计算错误或调度异常. 鉴于控制模块的硬件开销相对较小,本文采用三模冗余(triple modular redundancy, TMR)[23]策略部署 3 个并行工作的同构模块,并通过多数表决机制屏蔽单个模块的瞬态故障或永久失效,从而提升控制路径的可靠性.

数据传输通道主要由输入和输出链路、输入缓冲区、交叉开关及其物理互连结构等关键组件构成. 上述组件一旦发生故障,可能导致报文丢失或数据完整性受损,严重影响通信质量. 为增强数据通道的容错能力,本文在基准路由器架构中引入循环冗余校验(cyclic redundancy check, CRC)编解码器、信用验证单元、故障状态机以及备份缓冲区等容错组件. CRC 编解码器与信用验证单元协同运行,实现对报文软错误及丢包事件的实时监测. 备份缓冲区用于在数据确认前暂存待确认报文,以便在发生故障时立即触发重传机制. 故障状态机根据检测反馈动态评估故障等级,并据此控制路径隔离与恢复流程,实现链路资源的自动重启. 针对本地链路与交叉开关等难以通过路由绕行实现容错的结构性瓶颈,本文引入硬件冗余设计. 通过旁路切换机制实现故障组件的快速隔离与备用通道的动态启用,从而确保关键通信路径的持续可达性.

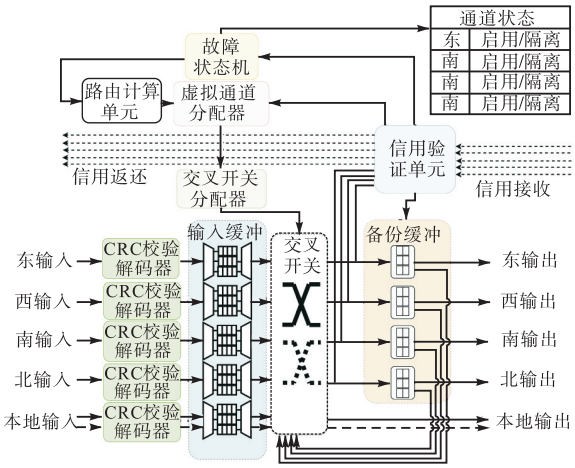

图 1 PFGM-NoC 路由器微架构

Fig.1 Microarchitecture of PFGM-NoC router

1.2 故障检测与报文重传

1.2.1 协同信用反压流控的故障检测

上、下游路由节点之间的通信采用基于信用反压的流量控制机制,其工作流程如图 2 所示. 该机制通过在上游节点维护一个信用计数器,用以实时反映下游节点输入缓冲区的可用资源状态. 信用计数器的初始值设定为下游缓冲区的总存储单元数量. 当本地输入队列的队首报文在交叉开关仲裁中获得传输权限,且当前信用值大于零时,节点置位“发送就绪”信号以启动数据传输,并相应地将信用值递减. 若信用值减至零,表明下游缓冲区资源已被占满,传输过程将被暂时阻断. 随着下游节点释放缓冲区资源,其通过返还信用信号通知上游节点,更新对应的信用值. 这一过程构建了闭合的流控反馈回路,确保数据传输的有序稳定.

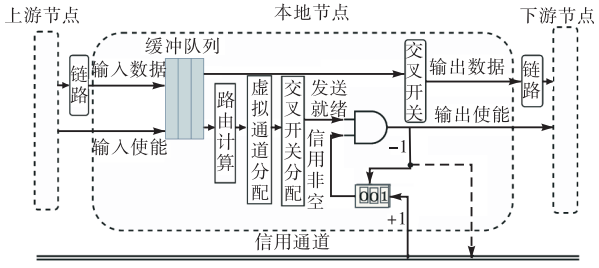

图 2 基于信用反压的流量控制

Fig.2 Credit-based backpressure flow control

为提升报文丢失检测与数据完整性校验能力,本

文在传统信用流控机制基础上扩展引入“校验信用信号”路径. 报文在进入 NoC 前经由 CRC 编码器完成冗余编码；中间节点接收到报文后，路由器根据目标地址计算其下一跳方向及对应的目标虚拟通道，并在交叉开关调度阶段将其转发至下游节点. 下游路由器接收报文后，将其写入指定输入缓冲区队列，随后依次执行路由计算、虚拟通道分配和交叉开关分配操作. 在进入交换调度前，通过 CRC 解码器对报文数据执行完整性校验，并返还校验信用信号至上游节点. 该信号包含 3 个域：虚拟通道编号（VC_ID）、报文唯一标识符（PKT_ID）和校验结果标志（CRC_FLAG），用于辅助上游节点判断报文是否已被成功接收以及数据是否完整.

为实现故障检测灵敏度的可控调节，本文在各虚拟通道路径中引入了校验信用返还延时计数器与状态寄存器. 延时计数器初始值为 0，每个时钟周期自增 1，用于跟踪报文从发送到收到校验反馈的时延. 其阈值根据系统运行负载及容错策略动态设定，以在检测灵敏度与误判控制之间实现平衡. 状态寄存器 S 为 2 bit 宽，初始值为 00，用于跟踪当前虚拟通道最近一次转发报文后的校验反馈状态.

当节点接收到来自下游的校验信用信号时，将依据其 VC_ID 字段暂停对应虚通道首个已发送报文的延时计数器，并更新其状态寄存器：若 PKT_ID 不匹配，S 更新为 10；否则若 CRC_FLAG 异常，S 更新为 01；若两者均正确，S 更新为 11. 通过监控各虚通道的延时计数器数值及状态寄存器内容，可有效评估当前报文是否发生传输异常，并进一步识别故障类型. 故障判别逻辑如图 3 所示.

（1）$S = 11$ 且延时计数器未超过阈值：表示当前报文被成功接收并通过校验，判定为正常通信.

（2）$S = 10$ 且延时计数器未超过阈值：表示当前报文未收到校验信用反馈，但其后续报文已成功确认，判定为个别报文丢失，通道功能恢复.

（3）$S = 01$ 且延时计数器未超过阈值：表示报文已被接收但 CRC 校验失败，判定为数据完整性故障，可能由位翻转等软错误导致.

（4）$S = 00$ 且延时计数器达到阈值：表示当前报文未被确认接收，且对应虚拟通道尚无后续报文校验反馈，判定为报文丢失，通道状态不确定.

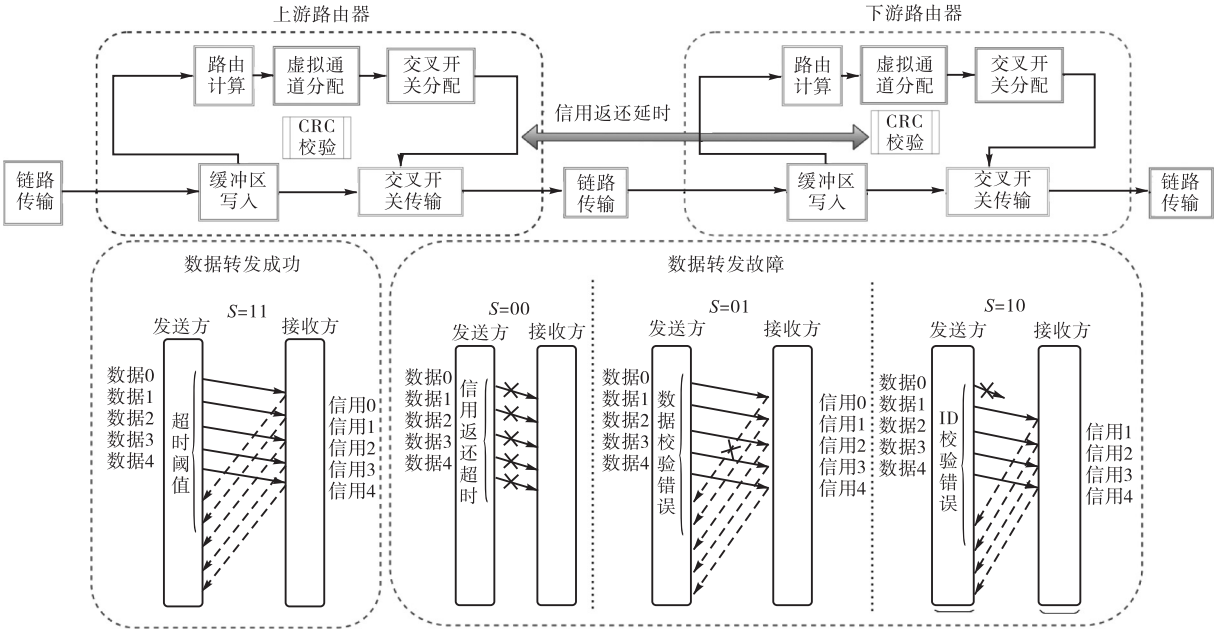

图 3　故障场景检测示例

Fig.3　Fault scenario detection examples

1.2.2　基于备份缓冲区的故障重传

为保障故障场景下的无丢包可靠传输，PFGM-NoC 架构为每个路由器的转发端口（东、西、南、北）配置了轻量级备份缓冲区. 备份缓冲区细分为多个子队列，分别对应不同虚拟通道. 报文通过交叉开关转发时，其副本同步写入当前路由器对应输出方向的备份缓冲区队尾. 仅当收到下游节点返回的校验确认，确认报文成功接收且通过 CRC 校验后，该副本方可从本地备份缓冲区安全删除. 同时，下游节点在转发报文前，也将其写入本地备份缓冲区，实现跳对跳的级联式报文备份.

一旦检测到通信故障，相关备份队列的重传使能信号即被激活. 此时，备份缓冲区通过 2 选 1 多路复用器接入交叉开关输入端，用于调度并重传故障报

文. 若重传使能未激活,多路复用器则维持输入缓冲区与交叉开关之间的常规连接,实现正常数据转发. 为确保故障报文的优先重传,交叉开关仲裁机制在调度策略中赋予其更高的服务质量等级,从而保障所需带宽资源,降低故障对整体性能的影响,并显著缩短网络恢复延迟. 图 4 展示了基于备份缓冲机制的故障重传逻辑.

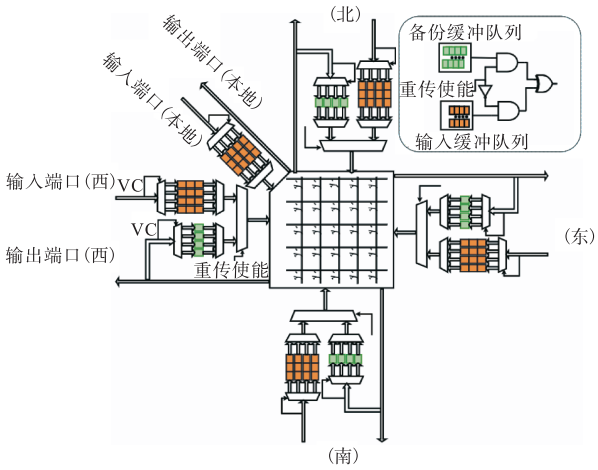

图 4 备份缓冲区的报文重传

Fig.4 Backup buffer-based packet retransmission

该机制确保数据副本在下游节点确认报文无误之前得以完整保留,实现上一跳节点快速响应故障重传,无需通知源节点进行重发. 备份缓冲区专门用于临时存储尚未获得校验确认的报文. 在网络正常运行状态下,各路由节点间的转发速率基本匹配. 因此,写入备份缓冲区与确认删除的速率相对平衡,报文在备份缓冲区中的驻留时间较短. 基于该特性,备份缓冲区可采用轻量级队列结构实现,既保障容错能力,又避免过度占用缓存资源. 当备份缓冲区写满时,通常表明对应通道存在严重的通信故障或拥塞状况. 此时,系统将主动施加反压,限制该端口的报文继续转发. 这一机制为链路恢复与拥塞缓解提供了必要的缓冲空间,防止异常扩散,从而维护网络整体的稳定性与可靠性.

1.3 基于状态机的故障定级管理

故障状态机(fault state machine,FSM)作为本架构中的关键控制单元,负责管理各输出通道的状态,并动态调整路由策略. 该模块内部维护 3 组核心寄存器:启用状态寄存器 E_i,用于指示第 i 个输出通道的可用性;故障等级寄存器 L_i,用于量化通道故障的严重程度并确定隔离时长;故障恢复计数器 C_i,负责记录隔离状态下的周期计数值.

在系统上电后的初始化阶段,各寄存器的值设定如下:$E_i=1$,表示所有输出通道默认处于启用状态;$L_i=0\text{x}0001$,对应最低故障等级;$C_i=0$,表示无初始隔离等待时间. 当信用验证单元检测到通道 i 发生故障时,将向 FSM 发送故障检测脉冲 P_d,此时寄存器 E_i 被置为 0,通道随即进入隔离状态. 通道恢复至启用状态存在两种情形:其一,在隔离期间接收到有效的校验信用,表明故障为瞬态或间歇性,此时信用校验单元向故障状态机发送故障恢复脉冲 P_r,触发寄存器 E_i 置为 1,通道随即恢复启用;其二,当计数器 C_i 的值达到当前故障等级寄存器 L_i 所设定的阈值,表明隔离周期结束,寄存器 E_i 同样被置为 1,以恢复该通道的启用状态.

若启用状态下再次检测到故障,则表明该通道可能存在持续性故障. 此时,故障等级寄存器 L_i 左移 1 位,以提升故障等级并使隔离周期呈指数级增长. 该策略在兼顾瞬态故障快速恢复与永久性故障有效隔离方面展现出良好适应性,既可降低因误判引发的激进隔离和带宽资源浪费,又能有效避免对不可用通道的重复访问所导致的网络拥塞风险. 当接收到故障恢复脉冲 P_r 后,寄存器 L_i 被重置为初始值 0x0001. 在隔离状态下,计数器 C_i 于每个时钟周期自增;若接收到 P_r 或者其计数达到 L_i 所设定的阈值,通道即自动恢复至启用状态,且 C_i 同步清零. E_i、L_i 与 C_i 的状态演化可形式化定义为

$$E_i'=\begin{cases}0 & P_\text{d}\wedge E_i\\ 0 & (P_\text{r}\vee(C_i=L_i))\wedge\overline{E_i}\\ E_i & \text{其他}\end{cases}\tag{1}$$

$$L_i'=\begin{cases}L_i & P_\text{d}\wedge E_i\\ 1 & P_\text{r}\\ L_i & \text{其他}\end{cases}\tag{2}$$

$$C_i'=\begin{cases}0 & P_i\vee(C_i=L_i)\\ C_i+1 & E_i\wedge\overline{(P_\text{r}\vee(C_i=L_i))}\\ C_i & \text{其他}\end{cases}\tag{3}$$

式中:E_i 和 E_i'分别表示第 i 个输出通道在当前与下一时钟周期的启用状态;L_i 和 L_i' 表示对应通道在当前与下一时钟周期的故障等级;C_i 和 C_i' 则分别表示当前与下一周期的隔离计数器数值.

通过上述机制,FSM 在端口粒度上实现了故障等级的动态管理,支持对各输出通道的灵活隔离与启用控制. 系统根据故障严重程度自适应调整隔离周期,避免因反复访问故障通道而引发的网络瓶颈,同时也能在故障解除后及时释放资源,恢复通道可用性. 通道状态由寄存器 E_i 指示,并同步传递至路由计算模块,后者据此调整路由策略. 当某输出通道处于

隔离状态时，路由逻辑自动屏蔽该方向，并选择其余可用通道进行绕行，从而保障数据传输的连续性，维持网络整体的高效运行.

1.4　协同容错路由算法

在 PFGM-NoC 架构中，完成通道检测后，故障状态机将主动隔离故障通道，以防止故障扩散. 为维持网络连通性，报文需绕行至可用路径继续转发，但此过程可能引发路径拥塞甚至死锁. 为此，本文提出协同容错自适应路由（collaborative fault-tolerant adaptive routing，CFAR）算法，以在故障环境下实现安全稳定的路径重构.

PFGM-NoC 架构中每个端口配置多个虚拟通道，包括至少一个逃逸虚拟通道（escape virtual channel，EVC）和多个非逃逸虚拟通道（non-escape virtual channel，NEVC）. 该设计可在故障发生时，利用 EVC 打破路径循环依赖，保障网络连通性. CFAR 的具体路由选择算法伪代码如下.

算法 1：CFAR 路由选择算法

输入　C_{p}：当前报文的输入端口；
　　C_{vc}：当前报文所处的虚拟通道；
　　端口状态表：各输出端口的启用/禁用状态

输出　O_{s}：可选输出路径集合，集合中的每一项为三元组形式（端口，虚拟通道，优先级）

//定义：X_{p}：按 *XY* 维序优先的输出端口；
　　Y_{p}：按 YX 维序优先的输出端口；
　　R_{p}：本地处理单元端口；
　　O_{p}：其他启用状态端口；

初始化 O_{s} 为空集合；
若 $C_{\mathrm{vc}}\in$EVC，则
　若 X_{p} 启用：
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(X_{\mathrm{p}},\mathrm{EVC},\text{高})\}$；
　否则：
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(R_{\mathrm{p}},\mathrm{EVC},\text{高})\}$；
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(R_{\mathrm{p}},\mathrm{NEVC},\text{高})\}$；
结束；
否则若 $C_{\mathrm{vc}}\in$NEVC，则
　若 X_{p} 启用且 $X_{\mathrm{p}}\neq C_p$，则
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(X_{\mathrm{p}},\mathrm{EVC},\text{中})\}$；
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(X_{\mathrm{p}},\mathrm{NEVC},\text{高})\}$；
　结束；
　若 Y_{p} 启用且 $Y_{\mathrm{p}}\neq C_p$，则
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(Y_{\mathrm{p}},\mathrm{EVC},\text{中})\}$；
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(Y_{\mathrm{p}},\mathrm{NEVC},\text{高})\}$；
　结束；
　若 O_{p} 启用且 $O_{\mathrm{p}}\neq C_{\mathrm{p}}$，则
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(O_{\mathrm{p}},\mathrm{EVC},\text{低})\}$；
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(O_{\mathrm{p}},\mathrm{NEVC},\text{低})\}$；
　结束；
结束；
若 O_{s} 为空集合，则
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(R_{\mathrm{p}},\mathrm{EVC},\text{高})\}$；
　　$O_{\mathrm{s}}\leftarrow O_{\mathrm{s}}\cup\{(R_{\mathrm{p}},\mathrm{NEVC},\text{高})\}$；
结束.

算法的具体步骤如下.

步骤 1　当报文处于 EVC 中时，其路由严格遵循 *XY* 维序规则，确保数据始终沿无环路径前向传输至目标节点. 若目标端口按 *XY* 维序可达并处于启用状态，报文将经由该端口转发至下一节点的 EVC 缓冲区. 若该端口被隔离，则报文通过本地端口排出，并由当前节点重新注入网络进行绕行.

步骤 2　当报文处于 NEVC 中时，路由器优先选择最短路径方向的可用端口转发报文. 此时，NEVC 的优先级高于 EVC，以充分利用高带宽资源. 当所有最短路径端口均不可用时，CFAR 会选择可用的非最短路径进行绕行. 此时，所有虚拟通道仍可被选用，但其调度优先级会进一步降低.

为避免路径依赖引发死锁，CFAR 明确禁止报文沿原输入方向回退. 例如，若报文自北方向注入，则在其转发过程中，北向端口不可作为合法输出路径，从而确保路径始终朝前推进. 通过将 EVC 的调度优先级设置低于 NEVC，可以避免其在非死锁情形下被频繁抢占，从而保障 EVC 在死锁解除时发挥关键作用. 若因虚拟通道限制或回退禁止导致无端口可供选择，报文将通过本地端口排出后重新注入网络. 这一机制在消除转发依赖环路的基础上恢复传输，兼顾了死锁规避与连通性维持.

在局部高密度故障场景中，多个相邻节点间通道同步失效，导致网络可用路径明显减少，路径选择复杂度显著提升. CFAR 算法通过灵活的路径调度策略与动态优先级调整机制，有效应对此类挑战. 图 5 展示了算法在不同故障场景下的容错路径选择策略. 在图 5(a)的单通道故障场景中，算法将优先选择其余可用的最短路径端口进行报文转发，确保路径最优性，有助于降低传输延迟与能耗，实现高效的局部容错. 在图 5(b)的双通道故障场景中，所有最短路径端口均被隔离，算法启用非最短路径绕行机制，引导转发报文避开故障区域. 在图 5(c)的三通道故障场景下，所有前向端口均被隔离，且因禁止回退，原输入方向不可用. 此时，算法将报文从本地端口排出并于稍后重新注入，完成避让式重路由. 借助 CFAR 算法的路径约束、优先级调度与自适应绕行策略，

PFGM-NoC 架构能够在多类型混合及局部高密度故障场景下实现高效的路径重构，保障网络连通性与通信可靠性.

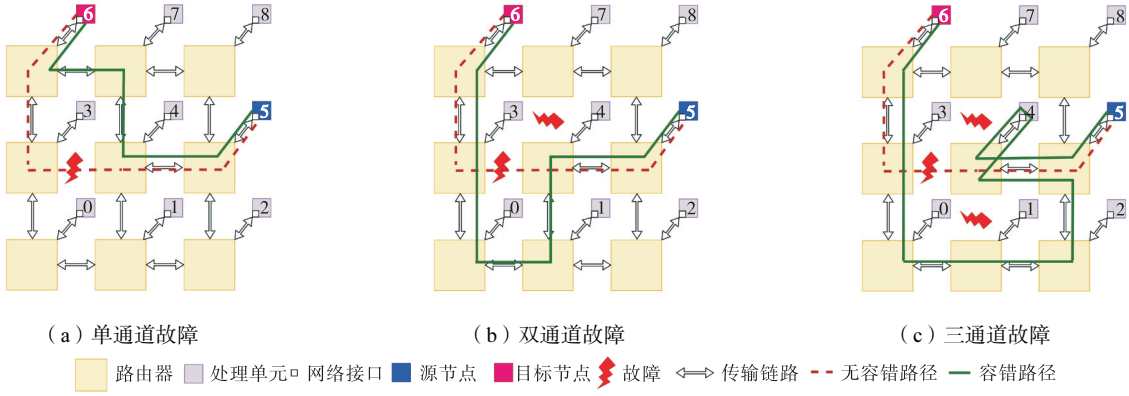

图 5 不同故障场景下的容错路径选择策略

Fig.5 Fault-tolerant path selection strategies in different fault scenarios

2 实验结果与分析

2.1 实验环境配置

本文基于扩展的 Booksim 2.0 模拟器对所提 PFGM-NoC 架构进行了性能评估，具体仿真参数见表 1. 实验中每个端口配置两个虚拟通道，其中一条作为逃逸通道. 虚拟通道及交叉开关仲裁均采用基于优先级的调度机制，以强化对高优先级报文的服务能力，实现差异化的服务质量管理.

表 1 仿真实验参数

Tab.1 Simulation experiment parameters

参数	设置
拓扑结构	8×8 二维网格
流量控制	基于信用的虫洞流控
虚拟通道数	2
虚拟通道缓冲区大小	8 个 flit
消息大小	10 个 flit
虚拟通道/交换仲裁器	优先级选择分配器
路由器流水线阶段数	5 阶段
流量模式	均匀,热点,邻近

报文传输延迟定义为首个 flit 注入网络起至尾部 flit 被目标节点接收所经历的时间. 网络饱和吞吐率在各节点持续发起报文传输请求的条件下测量，计算所有节点在单个时钟周期内接收的 flit 数量平均值. 为保证结果的稳定性与可靠性，仿真设置包含 30 000 个时钟周期的预热阶段，随后采集连续 100 000 个周期的稳态性能数据.

为评估 PFGM-NoC 架构的性能，选取以下 3 种具有代表性的容错机制作为对比方案.

(1) FT-E2E：文献[24]提出的容错机制，通过协议级在线诊断实现故障链路检测与隔离，采用端到端重传完成故障报文恢复.

(2) EsyTest：文献[25]提出的容错机制，基于周期性自检对网络通道进行定期扫描，同样通过端到端重传实现故障数据恢复.

(3) Inject-Fault：未引入路由调整策略，故障报文依赖源节点的超时重传机制重新注入网络，用以评估缺乏主动容错支持时的性能下限.

2.2 性能分析

本文通过多组对比实验，采用平均报文传输延迟与饱和吞吐率作为核心性能指标，系统评估网络在不同故障条件下的实时响应能力与最大负载承受能力. 针对所提出的 PFGM-NoC 架构，本文将其容错机制拆解为 3 个关键步骤，逐步实现并验证各子机制之间的协同增效作用.

步骤 1(PFGM-1) 实时故障检测与定位. 在该阶段，所提出架构引入了信用返还标识确认机制，可实现亚时钟周期级延迟的多类型故障感知，显著提升系统的故障响应效率，为后续的资源调度与路径重构提供支撑.

步骤 2(PFGM-2) 快速故障报文重传. 该阶段融合了轻量化备份缓冲区机制与优先级调度策略，支持对故障报文进行跳步式快速重传，有效降低了因故障引发的重传时延，提升系统在高故障率环境下的通信稳定性与服务质量保障能力.

步骤 3(PFGM-NoC) 动态故障等级评估与管理. 该阶段引入基于故障状态机的动态等级评估机制，实现端口故障的演化跟踪，支持通道的自适应隔离与重启控制，有效提升了系统资源利用率，减少了容错操作对有效带宽及整体性能的影响.

图 6 展示了不同流量模式及故障注入率条件下，各方案的平均报文传输延迟对比. 实验结果显示，在

所有测试场景中，PFGM-NoC 架构均表现出显著优势. 其中，Inject-Fault 方案缺乏动态故障适应机制，仅依赖端到端重传，导致延迟性能最差. FT-E2E 方案在检测同一通道多次故障后将其标记为永久性故障，难以有效应对间歇性故障场景，限制了网络资源的动态复用能力. EsyTest 方案依赖周期性自检，存在故障感知滞后，且需从源节点重新注入故障报文，导致重传路径延长，进一步加剧了网络传输延迟. 相比之下，PFGM-NoC 架构通过实时故障诊断和动态隔离机制，显著减少了非必要重传次数，大幅提升了网络可用通信资源的利用效率. 同时，对重传报文实施高优先级调度，有效降低了故障对整体传输延迟的影响，使其在高故障率环境下表现出更卓越的传输时延性能.

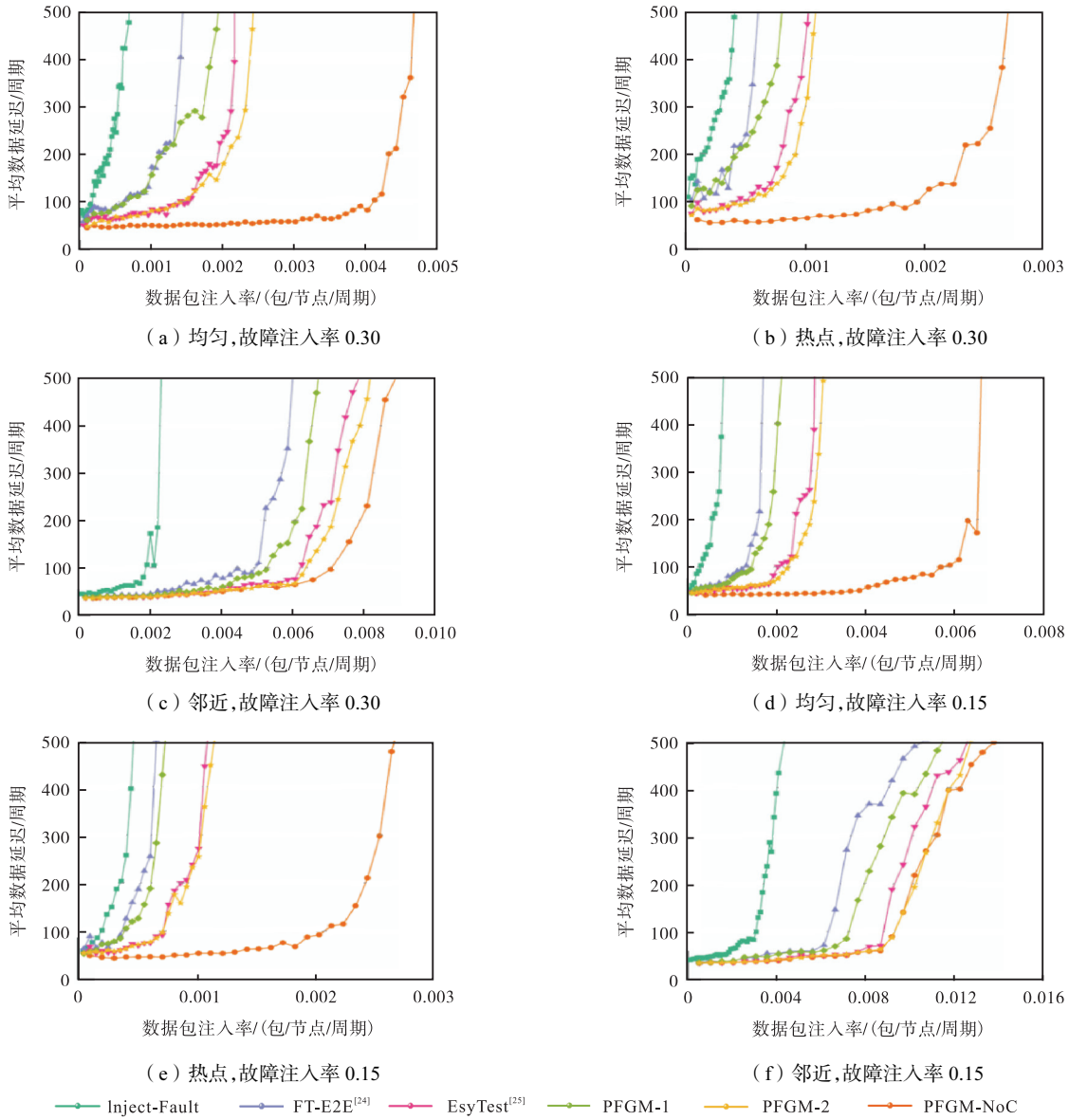

图 6　不同故障注入率与流量模式下的平均延迟对比

Fig.6　Average latency comparison under different fault injection rates and traffic patterns

表 2 展示了在不同故障注入率及流量模式下的饱和吞吐率比较结果. 实验表明，PFGM-NoC 架构在各类测试流量模式下均显著提升了网络的饱和吞吐率. 在均匀流量模式中，当故障注入率为 0.15 时，PFGM-NoC 的饱和吞吐率较 FT-E2E 和 EsyTest 分别提升了 19.3% 和 16.6%；故障注入率增加至 0.30 时，提升幅度进一步增至 21.8% 和 14.3%. 在热点流量模式下，当故障注入率为 0.15 时，PFGM-NoC 的饱和吞吐率分别较 FT-E2E 和 EsyTest 提高了 35.5% 和 23.0%；故障注入率达到 0.30 时，相应提升幅度分别达到 41.6% 和 26.2%. 在邻近流量模式中，故障注入率为 0.15 时，PFGM-NoC 较 FT-E2E 和 EsyTest 的饱和吞吐率提升幅度分别为 1.1% 和 0.5%；当故障注入率增至 0.30 时，提升幅度分别扩大至 2.7% 和

1.8%. 上述结果表明, PFGM-NoC 架构在多种流量模式及故障条件下均表现出显著的性能优势.

表 2 不同故障注入率与流量模式下的饱和吞吐率对比

Tab.2 Comparison of saturated throughput under different fault injection rates and traffic patterns 分片/周期

注入率	流量模式	Inject-Fault	FT-E2E[24]	EsyTest[25]	PFGM-1	PFGM-2	PFGM-NoC
0.15	均匀	0.067 26	0.126 70	0.129 68	0.127 10	0.132 53	0.151 17
	邻近	0.530 54	0.626 24	0.629 90	0.626 19	0.630 11	0.633 17
	热点	0.057 45	0.095 60	0.105 35	0.098 90	0.112 16	0.129 60
0.30	均匀	0.061 22	0.107 71	0.114 70	0.109 74	0.119 65	0.131 17
	邻近	0.504 56	0.587 15	0.592 49	0.589 11	0.598 96	0.602 99
	热点	0.053 45	0.082 61	0.092 69	0.087 62	0.095 71	0.116 99

在热点流量模式下, 由于网络流量分布不均, 各容错机制的吞吐率均低于均匀流量和邻近流量模式. 邻近流量模式中, 由于报文跳数较少, 有效降低了传输延迟并提升了吞吐性能, 因而所有容错机制均表现较好. 该模式下, 节点仅向相邻节点转发报文, FT-E2E 和 EsyTest 方案的故障报文重传均从上一跳开始, 导致 PFGM-NoC 在跳对跳重传方面的优势有所减弱. 然而, 凭借优先级调度与动态资源管理机制, 该架构依然保持较高的网络资源利用率, 性能优于其他两种方案.

通过分阶段实施并分析 PFGM-NoC 架构的 3 项优化措施, 量化了各阶段对网络容错性能的提升效果. 在均匀流量模式下, 故障注入率为 0.15 时, 第 1 阶段引入信用返还标识确认机制, 实现快速故障检测, 使饱和吞吐率较 Inject-Fault 方案提升 89.0%; 第 2 阶段通过备份缓冲区设计与优先级调度, 进一步降低重传延迟, 饱和吞吐率提升至 97.3%; 第 3 阶段引入故障等级动态评估与管理, 及时识别并恢复间歇性及瞬态故障通道, 饱和吞吐率提升至 124.8%. 当故障注入率提升至 0.30 时, 3 个阶段优化分别使饱和吞吐率较 Inject-Fault 方案提升 79.3%、95.5% 和 114.3%.

上述结果表明, PFGM-NoC 架构在多类型故障条件下显著提升了网络容错能力, 有效提高了饱和吞吐率的同时, 保持了较低的传输延迟, 并在高故障注入率下表现出良好的适应性和稳定性.

2.3 硬件实现评估

为评估各容错方案在功耗、复杂度及硬件资源开销方面的表现, 本文基于 Verilog HDL 实现各方案, 并在 Vivado 平台及 Xilinx XC7A200T FPGA 上进行综合分析. 测试统一采用 128 bit 报文宽度和 100 MHz 时钟频率, 评估指标涵盖关键路径延迟、功耗及资源占用, 具体对比如表 3 所示.

在本文采用的流水线化路由器中, 关键路径延迟主要受各流水段固有延迟限制. 综合评估结果表明, 引入容错机制后对关键路径时序影响较小, 未产生新的瓶颈. 然而, 随着故障检测与容错绕行逻辑的实现, 路由器的功耗和资源开销有所上升. 其中, Inject-Fault 方案采用简单的超时重传机制实现被动容错, 未引入主动检测或绕行逻辑, 在功耗和资源利用上较为经济, 适用于故障率低且以瞬态故障为主的场景. 但在高故障密度或复杂故障条件下, 其固定路径重传策略缺乏动态适应能力, 易导致报文路径不可达且持续重传, 造成无效通信和带宽浪费, 显著影响系统通信可靠性.

表 3 关键路径延迟、功耗与资源开销对比

Tab.3 Comparison of critical path delay, power consumption and resource overhead

容错方案	关键路径/ns	功耗/W	查找表	寄存器
Inject-Fault	3.711	0.665	3 621	1 100
FT-E2E[24]	3.718	0.697	3 689	1 155
EsyTest[25]	3.725	0.722	3 825	1 228
PFGM-NoC	3.718	0.764	4 145	1 340

相比之下, FT-E2E 和 EsyTest 方案分别通过自定义协议的在线诊断与周期性 BIST 检测实现故障检测与隔离, 能够覆盖大部分静态故障, 但对动态故障的响应存在滞后, 限制了系统的实时性及资源复用效率. PFGM-NoC 的功耗及资源消耗略高于对比方案, 主要因其实现了更细粒度的故障检测和更完善的容错机制, 包括端口备份缓冲区、信用返还确认及故障状态机的动态维护等功能逻辑. 上述设计显著提升了 NoC 在混合型及高动态故障环境下的适应性和网络可达性, 保障了报文的高可靠传输.

此外, PFGM-NoC 通过对故障报文实施高优先级调度, 有效缓解了由局部故障引发的网络拥塞, 提升了关键数据传输的服务质量. 综上所述, 尽管本文提出的容错架构在功耗与资源开销方面有所增加, 但综合其在复杂环境中所展现的优越容错性能与系统鲁棒性, 该部分开销在工程实践中具有充分的合理性与较高的性价比.

3 结　语

本文针对深亚微米工艺节点下 NoC 面临的可靠性退化问题，以及现有容错机制在故障类型识别精度与差异化处理方面的局限性，提出了一种基于端口粒度故障定级管理的新型容错架构 PFGM-NoC. 该架构引入信用返还标识确认机制，实现对校验错误、报文丢失等多类型故障的高精度实时监测与定位；结合轻量级端口备份缓冲区设计，构建具备优先级调度机制的重传通道，显著增强了故障场景下的通信服务质量. 进一步，引入基于端口故障状态机的动态分级管理策略，实现对瞬态故障通道的快速自恢复以及对永久性故障通道的智能隔离，显著提升了网络资源的利用效率与调度灵活性. 在此基础上，结合所提出的协同容错路由算法，PFGM-NoC 可依据实时感知的故障拓扑自适应重构最优绕行传输路径，保障多故障并发场景下高效且无死锁的数据路由. 仿真实验结果验证了所提架构在多种故障模式与典型负载条件下，能够在硬件资源开销可控的前提下，显著提升网络吞吐率并有效降低平均传输时延. 综上所述，PFGM-NoC 架构在可靠性与性能之间实现了良好的权衡，为片上互连系统提供了一种灵活可行的容错解决方案.

参考文献：

［1］ Zhang J J，Chen J Y. Research on efficient routing algorithm for mesh on-chip networks based on transpose traffic pattern[C]//Proceedings of 2024 4th International Conference on Electronic Information Engineering and Computer Communication. Wuhan，China，2024：762-765.

［2］ Bhargavi M B，Rokkam S S，Parameswaran S，et al. Automated design and configuration of RISC-V based NoC-MPSoC framework on FPGA[C]//Proceedings of the 2024 28th International Symposium on VLSI Design and Test. Tamil Nadu，India，2024：1-6.

［3］ Khalil K，Kumar A，Bayoumi M. Dynamic fault tolerance approach for network-on-chip architecture[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems，2024，14(3)：384-394.

［4］ Khalil K，Mohaidat T，Sherif A，et al. Hierarchical fault-tolerant NoC architecture for reliable communication[C]//Proceedings of the 2024 IEEE 17th International Symposium on Embedded Multicore/Many-Core Systems-on-Chip. Kuala Lumpur，Malaysia，2024：354-360.

［5］ Kanakala S，Ashok Kumar K，Dananjayan P. High reliability NoC switch using modified Hamming code with transient faults[C]//Proceedings of the 2018 IEEE International Conference on System，Computation，Automation and Networking. Pondicherry，India，2018：1-5.

［6］ Huang L T，Yuan C K，Wang J S，et al. ECDR²：Error corrector and detector relocation router for network-on-chip[J]. IEEE Transactions on Computers，2021，70(4)：606-613.

［7］ Dumitras T，Marculescu R. On-chip stochastic communication[C]//Proceedings of the Design，Automation and Test in Europe Conference and Exhibition. Munich，Germany，2003：790-795.

［8］ Pirretti M，Link G M，Brooks R R，et al. Fault tolerant algorithms for network-on-chip interconnect[C]//Proceedings of the IEEE Computer Society Annual Symposium on VLSI. Louisiana，USA，2004：46-51.

［9］ Groeseneken G V. Hot carrier degradation and ESD in submicrometer CMOS technologies：How do they interact[J]. IEEE Transactions on Device and Materials Reliability，2001，1(1)：23-32.

［10］ Mahapatra S，Bharath Kumar P，Dalei T R，et al. Mechanism of negative bias temperature instability in CMOS devices：Degradation，recovery and impact of nitrogen[C]//Proceedings of the IEEE International Electron Devices Meeting. San Francisco，USA，2004：105-108.

［11］ Mohammed S W，Afroz F. Power optimized 7-port router design with BIST capability for 3D NoC architecture[J]. Journal of Electrical and Electronics Engineering，2017，10(1)：91-94.

［12］ Aghaei B，Khademzadeh A，Reshadi M，et al. A new BIST-based test approach with the fault location capability for communication channels in network-on-chip[J]. IEEE Transactions on Electron Devices，2017，33(4)：501-513.

［13］ Chang Y C，Chiu C T，Lin S Y，et al. On the design and analysis of fault tolerant NoC architecture using spare routers[C]//Proceedings of the 16th Asia and South Pacific Design Automation Conference. Yokohama，Japan，2011：431-436.

［14］ Ren Y，Liu L B，Yin S Y，et al. A VLSI architecture for enhancing the fault tolerance of NoC using quad-

spare mesh topology and dynamic reconfiguration[C]// Proceedings of the 2013 IEEE International Symposium on Circuits and Systems. Beijing, China, 2013: 1793-1796.

[15] Chatterjee N, Chattopadhyay S, Manna K. A spare router based reliable network-on-chip design[C]//Proceedings of the 2014 IEEE International Symposium on Circuits and Systems. Melbourne, Australia, 2014: 1957-1960.

[16] Jagadheesh S, Bhanu P V, Soumya J, et al. Reinforcement learning based fault-tolerant routing algorithm for mesh based NoC and its FPGA implementation[J]. IEEE Access, 2022, 10: 724-737.

[17] Chen Z S, Zhang Y, Peng Z B, et al. A deterministic-path routing algorithm for tolerating many faults on wafer-level NoC[C]//Proceedings of the 2019 Design, Automation Test in Europe Conference Exhibition. Florence, Italy, 2019: 1337-1342.

[18] Taheri E, Pasricha S, Nikdast M. ReD: A reliable and deadlock-free routing for 2.5-D chiplet-based interposer networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, 43(12): 4599-4612.

[19] Frantz A P, Kastensmidt F L, Carro L, et al. Evaluation of SEU and crosstalk effects in network-on-chip switches[C]//Proceedings of the 19th Annual Symposium on Integrated Circuits and Systems Design. New York, USA, 2006: 202-207.

[20] Feng C C, Lu Z H, Jantsch A, et al. Addressing transient and permanent faults in NoC with efficient fault-tolerant deflection router[J]. IEEE Transactions on Very Large Scale Integration Systems, 2013, 21(6): 1053-1066.

[21] 欧阳一鸣, 孙成龙, 李建华, 等. 针对瞬时故障和间歇性故障的 NoC 链路容错方法[J]. 计算机研究与发展, 2017, 54(5): 1109-1120.
Ouyang Yiming, Sun Chenglong, Li Jianhua, et al. Addressing transient and intermittent link faults in NoC with fault-tolerant method[J]. Journal of Computer Research and Development, 2017, 54(5): 1109-1120(in Chinese).

[22] Veera B E, Surya T, Sheik A I, et al. Robust error resilience network on-chip router architecture[C]//Proceedings of the 2024 International Conference on Computing, Semiconductor, Mechatronics, Intelligent Systems and Communications. Mangalore, India, 2024: 1-5.

[23] Xu H Z, Zhang B L, Pan C, et al. Energy-efficient triple modular redundancy scheduling on heterogeneous multi-core real-time systems[J]. Journal of Parallel and Distributed Computing, 2024, 191: 104915.

[24] Schley G, Batzolis N, Radetzki M. Fault localizing end-to-end flow control protocol for networks-on-chip[C]//Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. Washington, USA, 2013: 454-461.

[25] Wang J S, Ebrahimi M, Huang L T, et al. Efficient design-for-test approach for networks-on-chip[J]. IEEE Transactions on Computers, 2019, 68(2): 198-213.

(责任编辑:孙立华)