

基于网络监控器的专用片上网络动态容错路由

葛 芬¹, 吴 宁¹, 秦小麟², 张 颖¹, 周 芳¹

(1. 南京航空航天大学电子信息工程学院, 江苏南京 210016; 2. 南京航空航天大学计算机科学与技术学院, 江苏南京 210016)

摘 要: 针对专用片上网络(Network on Chip, NoC)全局通信事务管理和可靠性设计问题, 提出片上网络监控器的概念, 用于获取全局网络实时状态信息及执行路径分配算法, 基于此提出一种动态路由机制 DyRS-NM. 该机制能检测和定位 NoC 中的拥塞和故障链路, 并能区分瞬时和永久性链路故障, 采用重传方式避免瞬时故障, 通过重新路由计算绕开拥塞和永久性故障. 设计实现了 RTL 级网络监控器和与之通信的容错路由器模块, 并将 MPEG4 解码器应用映射至基于网络监控器的 4×4 Mesh 结构 NoC 体系结构中, 验证了系统性能以及面积功耗开销. 相比静态 XY 路由和容错动态路由 FADR, DyRS-NM 机制在可接受的开销代价下获得了更优的性能.

关键词: 片上网络; 动态路由; 容错; 拥塞

中图分类号: TN47

文献标识码: A

文章编号: 0372-2112 (2013) 11-2135-09

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2013.11.004

Network Monitor Based Dynamic Fault-Tolerant Routing for Application-Specific Network on Chip

GE Fen¹, WU Ning¹, QIN Xiao-lin², ZHANG Ying¹, ZHOU Fang¹

(1. College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China;

2. College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China)

Abstract: Aiming at the problem of global communication management and reliability design in application-specific Network on Chip (NoC), the concept of network monitor is introduced, which monitors overall network real-time conditions and implements path allocation algorithm. Then a novel dynamic routing scheme based on Network Monitor called DyRS-NM is presented. The proposed scheme has the ability to discover and deal with both congestion and permanent faults and distinguish them from transient faults. DyRS-NM can avoid transient faults by using retransmission scheme, and also can detour congested and permanently faulted links by recalculating routing paths. The RTL-level circuits design of the network monitor and fault-tolerant router are realized, and experimental results with a MPEG4 decoder application mapped onto network monitor based 4×4 mesh NoC architecture verify the system performance and cost of area and power consumption. Compared to both static XY routing and fault-aware dynamic routing FADR, significant performance improvements can be achieved by using the DyRS-NM scheme with acceptable additional cost.

Key words: network on chip; dynamic routing; fault tolerance; congestion

1 引言

片上网络(Network on Chip, NoC)是复杂片上系统设计全局互连与通信问题的有效解决方案, 决定其性能的一个关键因素是所采用的路由器及路由机制的设计^[1]. 随着系统芯片面积和特征尺寸的不断减小, 芯片上可能出现的故障变得不可忽视. 因而, 采用容错策略以便在出现通信故障时继续有效转发数据包, 成为 NoC 路由机制设计需考虑的一个重要问题.

已有一些研究工作提出了许多不同的 NoC 路由机制来避免网络拥塞, 以此提高网络性能, 如, DyAD^[2]、DyXY^[3]、RCA^[4]、DAR^[5]等. 然而, 这些方法将拥塞估计限定在相邻路由器^[2,3]、网络象限图^[4,5]或源-目的节点传输路径^[6]中, 不能有效反映全局的网络状况, 且均没有解决容错问题.

NoC 通信故障分为瞬时故障和永久性故障两类. 基于随机通信的容错路由^[7]以及基于检错纠错码的重传机制^[8]可用于解决瞬时故障. 文献^[9]介绍了多种用于

绕开永久性链路故障的确定性路由算法,但这些容错路由算法均为静态路由,当系统运行过程中出现新的物理故障时,便不能绕过新的故障节点.

文献[10]提出了一种避开链路故障的动态路由机制.文献[11]采用了一种保证 QoS 的最短路径容错路由机制 MinFT.文献[12]介绍了一种低能耗基于转弯模型的容错机制 NARCO.文献[13]提出了一种容错动态路由 FADR,该算法不仅可以定位和绕开故障链路,且可根据相邻节点的负载状态因子选择路由方向,以避免拥塞.文献[14]提出了一种基于故障区构建的可重构路由算法.然而,上述这些动态容错路由算法亦仅考虑相邻节点^[10~13]或所构造区域^[14]中的故障状态,而忽略了由局部故障可能会导致的全局网络拥塞以及数据包并发传输时的相关性.文献[15~17]提出了一些适用于片上网络或并行计算机网络的基于全局故障信息的容错路由算法,如,多遍维序路由^[15,16]、基于段的路由算法^[17].然而,上述路由机制中全局故障信息的获取依赖于系统配置阶段,当系统运行过程中出现新的故障时需要返回系统配置阶段,即属于静态配置,且没有突出片上网络设计的专用性特点.

为解决上述问题,本文从专用 NoC 设计中全局通信事务管理和可靠性设计的角度,提出片上网络监控器的概念,基于此提出一种能避免网络拥塞且具有容错能力的动态路由机制 DyRS-NM (Dynamic Routing Scheme based on Network Monitor).片上网络监控器用于实时监测全局网络状态(拥塞和故障),并基于网络实时状态确定应用任务所有通信踪迹的最优路由路径,以达到动态容错的目的.

本文以典型的 Mesh 结构 NoC 为实例,提出基于网络监控器的 NoC 体系结构构建方法及其关键组件的电路设计方法,并通过 MPEG4 解码器应用分析与比较了 DyRS-NM 的性能及开销.

2 基于网络监控器的 NoC 动态路由机制

2.1 基于网络监控器的 NoC 体系结构

DyRS-NM 机制依赖于在 NoC 中增加一个全局网络监控器.我们选用 4×4 Mesh 结构 NoC 为实例,构建基于网络监控器的 NoC 体系结构如图 1 所示.

从图中可以看出,区别于典型的 Mesh NoC 结构,每个路由器 R 除了与四个相邻路由器和一个本地 IP 核相连外,还通过专用总线连接到一个网络监控器.该专用总线用于传输网络状态和路由路径信息,可采用三模冗余 (Triple Module Redundancy, TMR) 技术,即使用 3 根线和一个投票表决器电路以保证信息传输的可靠性,如图 1 所示,输出数据状态 f 由三个输入数据状态 a 、 b 、 c 决定.

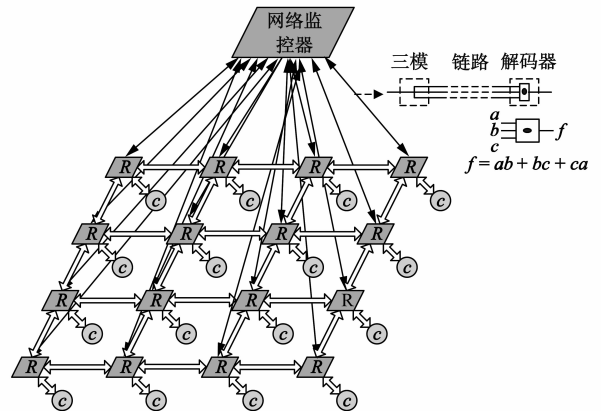


图1 基于网络监控器的NoC体系结构

2.2 动态路由机制 DyRS-NM

基于上述 NoC 体系结构, DyRS-NM 算法流程如图 2 所示,包含以下四个步骤.

(1)初始化每个路由器的路由表.

当专用 NoC 系统构建好后,系统拓扑结构、由映射算法生成的 IP 核在 NoC 中布局以及应用的通信踪迹都

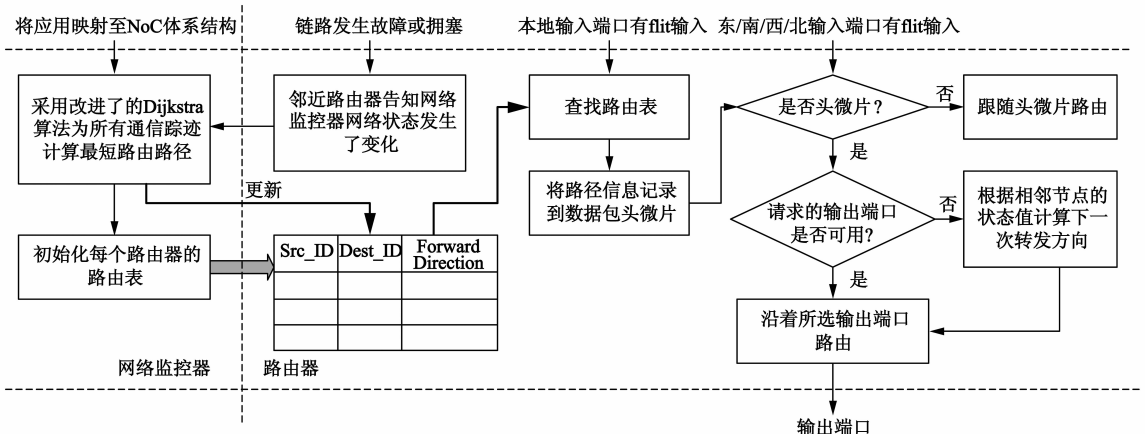


图2 DyRS-NM机制的算法流程图

已确定^[1]. 基于这些信息以及通信约束(带宽、延时), 网络监控器采用改进了的 Dijkstra 算法为每条通信踪迹计算最短路由路径. 一旦 NoC 应用系统开始工作, 网络监控器首先向所有路由器广播路由路径信息. 各个路由器仅接收源节点是自己本地 IP 核的通信路径, 并将路径保存到路由表中.

(2) 在正常数据包传输时, 源节点所连接的路由器查询路由表, 并将路径信息记录到数据包的头微片中, 以源路由方式传输.

(3) 在传输过程中, 当链路发生永久性故障或者拥塞时, 与链路相连的各个路由器同时通过专用总线告知网络监控器网络状态发生了变化.

(4) 网络监控器接收到全局网络实时状态信息后, 重新为所有通信踪迹计算最优可替换的路径, 避开故障节点, 且使得网络链路负载均衡, 避免拥塞, 并将新的路径信息发送给所有路由器.

为了降低传输延时, 当网络状态发生变化时, 路由器不需要等到网络监控器更新完毕路由表后才转发数据包, 而是继续沿着先前保存的路由路径发送数据. 在这种情况下, 数据包就有可能会遇到故障或拥塞的链路而不能继续前进(请求的输出端口不可用). 为了解决这个问题, 我们基于文献[3]提出的动态路由算法 DyXY, 在相邻节点拥塞状态反馈中增加故障状态信息, 据此计算数据包下一次转发方向. 这就意味着在步骤(2)中, 当数据报文离开源节点后, 如果由于一些突发原因(拥塞或者故障), 数据包不能被转发到下一跳, 路由器便会检查沿着目的节点方向相邻节点的状态值, 将数据包发送到状态值最小的相邻路由器(转发端口没有发生故障或者拥塞). 在下一个传输周期, 源节点就可按网络监控器更新的最优路由路径, 重新组包发往目的节点. 如此, 达到了动态容错的目的, 且所有通信可以在不降低服务质量前提下继续进行.

2.3 基于 Dijkstra 算法的 NoC 路径分配

NoC 路径分配问题的形式化描述为: 给定核通信图 CCG(C, A)和 NoC 体系结构图 NAG(R, P), 在 NoC 映射空间找到一个确定性、无死锁的最短路由路径集合, 满足核间通信踪迹的带宽、延时等约束, 使得 NoC 系统数据通信功耗最小, 同时保证最大链路负载最小^[18].

为了确保为每条通信踪迹找到的路径是最短路由, 可将 NoC 路径分配的范围限定在象限图中^[18]. 在 2D Mesh 结构 NoC 中, 根据源节点与目的节点的相对位置, 可划分为 8 种类型的象限图, 如图 3 所示, 分别为 ES、SW、WN、NE、WE、EW、NS、SN.

文献[18]提出了一种采用遗传算法解决 NoC 路径分配问题的方法, 该类采用启发式算法的方法一般用于 NoC 系统级设计阶段. 为降低硬件实现的复杂性, 本

文选用实现简单的经典最短路径算法 Dijkstra, 并做了相应改进, 以实现容错路由, 算法步骤描述如下.

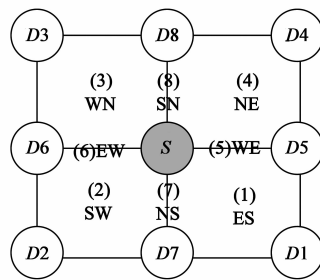


图3 2D Mesh结构 NoC的象限图

(1) 将 NAG 图描述的通信踪迹按通信带宽需求降序排列, 并由源/目的节点对生成象限图. 其中, 若生成的象限图为 WE、EW、NS 或 SN, 且其中存在链路故障或拥塞, 则将源节点和目的节点所在行或列各向外扩展一个节点, 生成新的象限图.

(2) 输入象限图 w 及其邻接表, 源节点 r 和目的节点 t .

(3) 初始化 w 中所有节点的距离, 距离 $d(i)$ 表示从源节点 r 到节点 i 路径上所占用链路带宽之和. 令 r 的距离 $d(r) = 0$; 对于 $v \neq r$, $d(v) = \infty$.

(4) 设 w 的节点集合为 T , 以 $u = r$ 开始搜索.

(5) 计算 u 的后继节点集合 $N(u)$, 对于每一个 $v \in N(u)$, 如果 $v \in T$ 且 $d(v) > d(u) + w(uv)$, 表示找到 v 的更短路径, 则更新 v 的距离 $d(v) = d(u) + w(uv)$. 其中, u, v 边上的权重 $w(uv)$ 为 CCG 中给出的通信踪迹带宽需求. 接着从 T 中删除当前 u 指向的节点, 则 $T = T - u$, 同时令 u 为 T 中另一个距离最小的节点.

(6) 重复步骤(5), 直到搜索到目的节点 t , 输出距离 $d(t)$.

在迭代过程中, 若由于链路故障或拥塞, 从当前节点 u 到其后继节点集合 $N(u)$ 中可访问的节点均无法找到一条更短的路径, 则沿之前建立的最短路径回溯到当前节点 u 的前一个节点, 并将当前节点 u 从 T 中删除, 接着重复执行步骤(5), 直到能找到下一个符合要求的节点 u . 如此, 最终生成的路径便会绕开故障或拥塞链路, 同时保证距离 $d(t)$ 即分配路径占用的链路带宽最小, 使得链路负载均衡.

如图 4 中源节点 $S1$ 与目的节点 $D1$ 生成象限图 SN, 其中存在故障或拥塞的链路(用“ \times ”表示), 则将源/目的节点所在列向外扩展一个节点, 生成新的象限图 WE + WN, 虚线箭头表示生成的路由路径. 在为源节点 $S2$ 与目的节点 $D2$ 分配路径时, 沿北向遇到节点 E 时无法继续往前, 则回溯至节点 N 绕过故障或拥塞的链路.

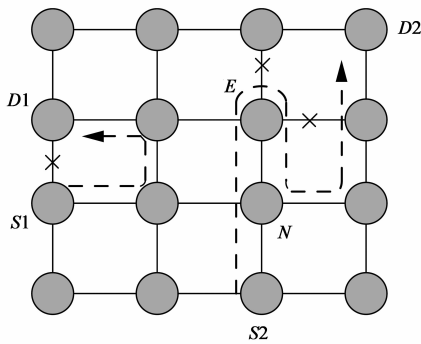


图4 两对节点间的路径分配

2.4 无死锁分析

文献[19]提出了一种在 n 维网格中避免死锁的方法. 该方法将网络分成 2^n 个虚拟网络, 每个虚拟网络中每个节点有 n 个单向的虚通道, 由节点和虚通道组成的网络即称为虚拟网络. 对应于不同虚拟网络的通道集是不相交的, 每个数据报文根据自己的目的地注入不同的虚拟网络并在该网络中路由, 直到到达自己的目的地, 如此, 使得相应的路由函数在通道间不存在环相关.

本文针对二维 Mesh 结构研究路由算法, 因此需要将网络分成 4 个虚拟网络, 并为每个路由节点各输入端口设置 2 条虚通道. 论文所采用改进了的 Dijkstra 算法将生成的路由限定在确定的象限图中, 根据象限图的定义, 象限图 ES (含 NS)、SW (含 EW)、WN (含 SN)、NE (含 WE) 就分别对应于二维 Mesh 可划分成的 4 个虚拟网络. 报文根据其源和目的节点相对位置确定象限图

类型, 即为选择注入某一个虚拟网络, 然后在给定的虚拟网络中路由, 所有属于最小路径的通道都可用于路由. 但是, 在一个虚拟网络内, 报文在每一维上只能沿一个方向移动, 且不能被传送到另一个虚拟网络. 如在图 3 中, 以南向为 X 维正方向, 东向为 Y 维正方向, 注入 $X + Y +$ 虚拟网络的报文只能沿 X 维和 Y 维的正方向移动, 注入 $X + Y -$ 虚拟网络的报文只能沿 X 维正方向和 Y 维的负方向移动, 依次类推. 显然, 通道间没有环相关, 从而避免了死锁.

3 基于动态路由机制的 NoC 结构设计

3.1 路由器设计

为实现 DyRS-NM 机制, 本文提出适用于其的路由器结构如图 5 所示, 每个路由器包含输入输出通道、输入缓冲区、重传缓冲区、输入解码器、输出编码器、端口控制器、状态控制器、路由模块、仲裁模块、通道分配模块、交换开关以及路由表和链路状态表.

输入解码器用于对输入通道的数据微片进行解码, 并验证其是否正确. 如果检测到微片中有错误, 则认为传输过程中产生了瞬时故障, 端口控制器会发送 Nack 信号要求发送者(上级路由器)重新传输该微片, 并将接收的数据丢弃; 若接收的数据微片没有错误, 则存入输入缓冲区, 分配相应的虚通道.

除了本地输入端口控制器, 其余四个方向的端口控制器会记录数据微片重传的请求次数和监控本方向输入缓冲区的占用率. 如果重传次数超过设定的门限值, 端口控制器将会发送状态信号 (Status Value = "10")

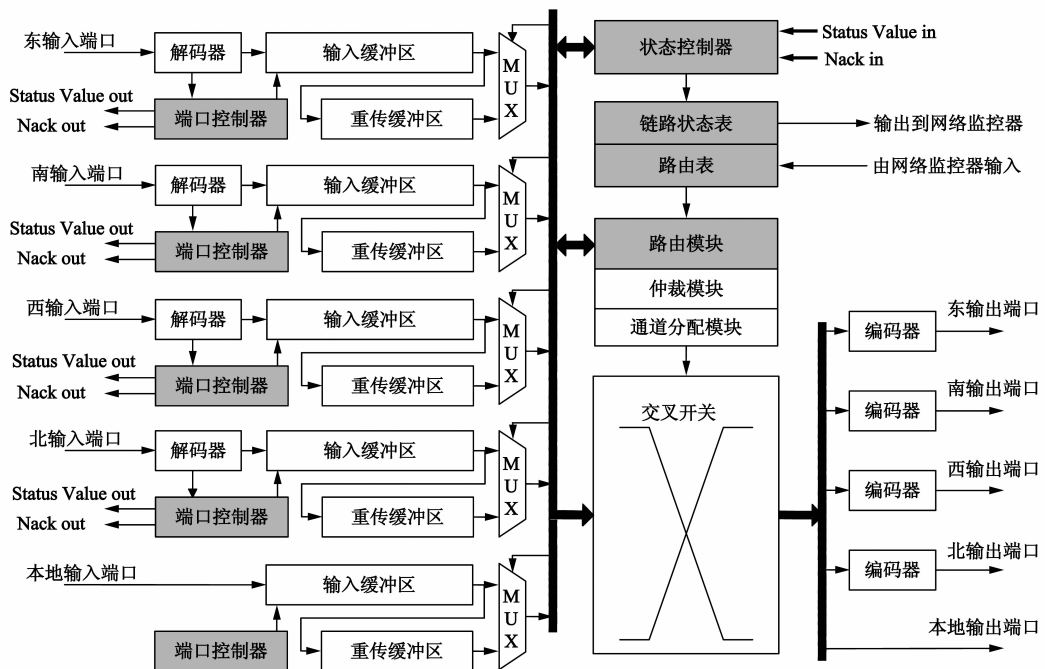


图5 基于动态路由机制的路由器结构

到上级路由器,告知其互连链路发送了永久性故障;如果缓冲区占用率达到设定的门限,端口控制器会发送状态值“01”,告知上级路由器其互连链路发送了拥塞.此外,如果状态值为“00”,表示当前互连链路不存在拥塞和故障,上级路由器需要尽可能快地将数据发送出去,以降低延时.

路由模块对存入输入缓冲区的包头微片进行解析,确定转发方向,选择输出端口.定义数据包头微片长度为 32bit,格式如图 6 所示.字段 Dest_ID 和 Src_ID 分别表示目的节点地址和源节点地址, Routing_path 记录路由路径信息, Rt_flag 为路由类型标志, Packet_length 为数据包长度, Hops 表示路由跳数.路由模块的电路框图如图 7(a)所示,各信号含义及处理流程如图 7(b)所示.

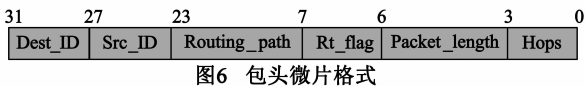
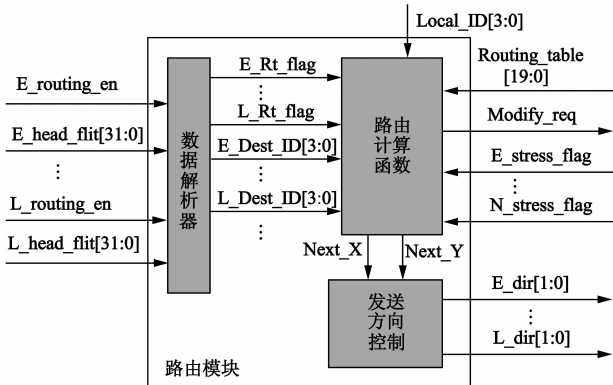


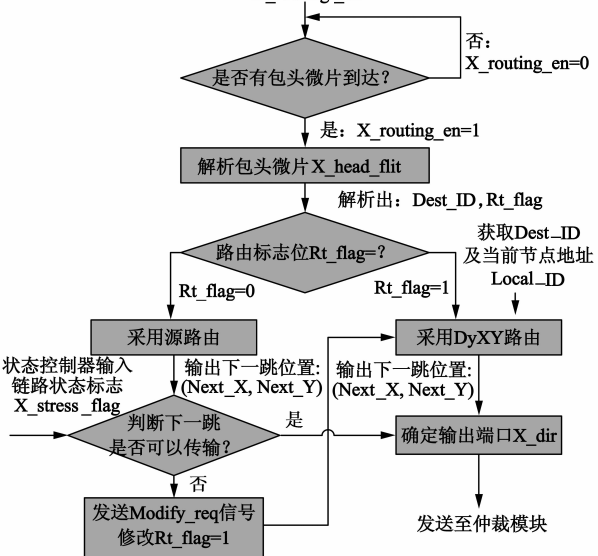
图 6 包头微片格式



(a) 电路框图

端口控制器输入信号

X_routing_en



(b) 路由算法处理流程

图 7 路由模块电路框图及算法处理流程

如果有多个输入端口请求相同的输出端口,仲裁模块根据各方向输入缓冲区的拥塞程度进行仲裁,拥塞程度较大(缓冲区占用率高)的输入端口具有更高的优先级,这样可以避免繁忙端口的传输延时.通道分配模块用于在当前被选择的输入输出端口通道之间建立交叉开关连接.当输入缓冲区中的数据微片被选中发送时,端口控制器会同时将其存入该端口的重传缓冲区.数据微片经输出端口发送到下一级路由器之前经过输出编码器模块进行编码,本文设计采用奇偶校验码的编码方式.

状态控制器用于处理链路状态值 Status Value 和 Nack 信号,它们由下一级路由器四个输入端口控制器反馈.如果接收到 Nack 信号,状态控制器控制 MUX 停止从选中的输入端口发送数据,而转而发送该端口重传缓冲区中的数据.如果状态控制器检测到下一级路由器反馈的状态值不是“00”时,这意味着相应的互连链路发送了故障或者拥塞,那么状态控制器会将该状态信息存入链路状态表,同时请求网络监控器从链路状态表中读取数据.路由表用于保存从网络监控器发送来的路由路径信息.

3.2 网络监控器设计

基于 DyRS-NM 的算法流程,本文提出了一种网络监控器的结构,如图 8 所示,主要由网络状态采集模块、路径分配算法模块、路由路径发送模块、通信任务表、链路状态表、全局路由表以及内建自测试模块组成.

网络状态采集模块与每个路由器相连,用于收集整个网络的实时链路状态信息.当接收到任何一个路由器的请求信号 Req 后,采集模块开始读取相应数据线 Link_state 上链路状态信息,存入链路状态表中,并发送 Ack 信号到相应路由器,并置位触发信号 routing_trigger,将每一条链路状态信息输出到路径分配算法模块.

路径分配算法模块根据当前网络状态为给定应用任务中的每一条通信踪迹计算路由路径 new_route,计算完毕后置位触发信号 send_trigger,请求路径发送模块准备发送路由路径.该模块设计的核心是实现路径分配算法,如,本文提出的改进了的 Dijkstra 算法,亦可替换为其它合适的路径算法,为此可通过基于 AHB-Lite 总线扩展 ARM Cortex-M0 处理器核及数据程序存储器的方式,由处理器核运行路径分配算法,提高计算效率和可扩展性.

路由路径发送模块将路径分配算法模块计算得到的当前结果 new_route 与之前保存在全局路由表中的路径信息 old_route 进行比较,仅将变化了的路径数据发送给相应路由器,并将其存入全局路由表,以提高路由路

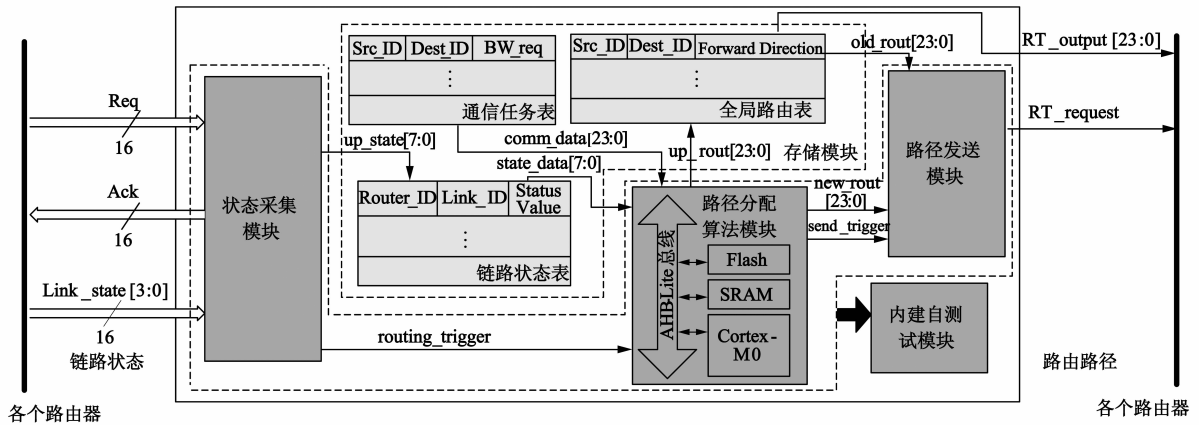


图8 基于动态路由机制的网络监控器结构

径信息更新效率. 发送时向相应路由器发送请求信号 RT_request, 同时输出路由信息至数据线 RT_output.

存储模块中的通信任务表用于保存每一条通信踪迹的源节点和目的节点编号以及通信带宽需求信息, 其中分别用 4bit 数据表示源节点编号 Src_ID 和目的节点编号 Dest_ID, 用 16bit 数据表示通信踪迹的带宽需求 BW_req. 4×4 Mesh 拓扑中 16 个网络节点的 ID 可表示为 0000 ~ 1111. 链路状态表使用 4bit 数据 Router_ID 标识路由器, 2bit 数据 Link_ID 标识与路由器相连的链路 (“00”、“01”、“10”、“11”分别表示路由器东南西北四个方向的链路), 2bit 数据 Status Value 描述链路状态 (“00”、“01”、“1X”分别表示链路状态为空闲、拥塞和故障). 全局路由表中的每条路由路径均按源节点编号、目的节点编号和每一跳转发方向 (“00”、“01”、“10”、“11”分别表示东南西北四个转发方向) 的格式保存. 其中, 在 4×4 Mesh 网络中最短路由路径的最大跳数为 6, 考虑到为了避开拥塞或故障节点需要采用绕道路由, 定义路由路径的转发方向 Forward Direction 为 16bit.

此外, 网络监控器增加内建自测试模块, 用于验证网络监控器本身的可靠性. 内建自测试模块由测试生成器、响应分析器以及测试控制器组成, 采用有限状态机实现.

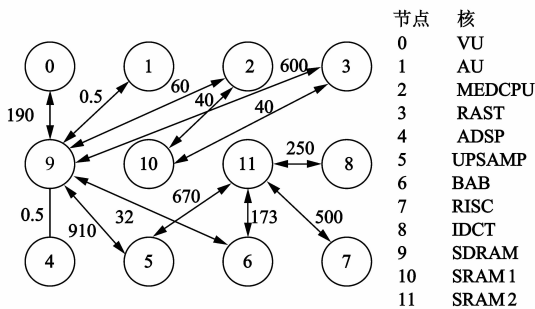
4 仿真结果与分析

4.1 路由机制的性能

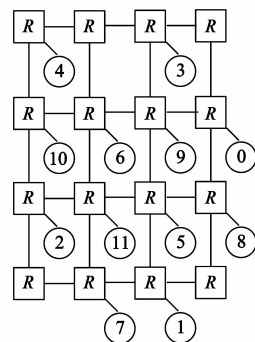
首先在 Synopsys VCS 环境下对构建的 RTL 级 NoC 模型进行仿真, 设置数据包长度固定为 8flits, 每个 flit 长度为 32bits, 输入端口 FIFO 缓冲区大小为 10flits, 拥塞门限为 FIFO 总容量的 60%, 重传次数门限值为 8 次, 工作频率设为 1GHz. 选择将 MPEG4 解码器应用映射至 NoC 体系结构中, 其通信任务图如图 9(a) 所示, 优化映射结果如图 9(b) 所示, 如此便确定了功能 IP 核的布局及核间通信踪迹, 据此初始化网络监控器中的通信任务表. 我们通过两组实验来验证动态路由机制 DyRS-NM 的性能.

第 1 组实验验证系统对于瞬时链路故障的容错能力, 将基于 DyRS-NM 机制的 NoC 系统与非容错的普通路由器构建的 NoC 系统进行比较. 为模拟瞬时故障的产生, 在路由器之间的通信链路上引入一定的误码率 (Error Probability, ER). 仿真运行 2×10^4 个时钟周期后, 不同误码率下网络的收包率和平均延时如图 10 所示.

从图中可以看出, 随着 ER 的增大, 基于 DyRS-NM 机制的网络收包率保持不变, 而非容错路由器组成的

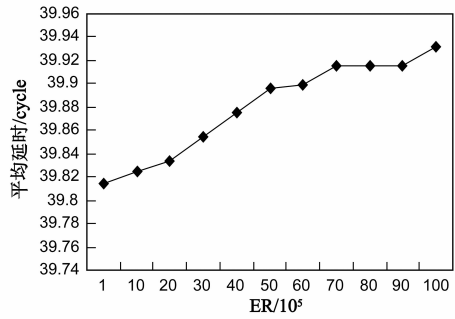
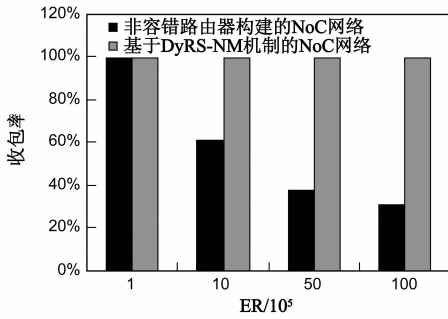


(a) MPEG4解码器的核通信图



(b) 优化映射

图9 基于NoC结构的MPEG4解码器应用



(a) 不同误码率下网络的收包率比较

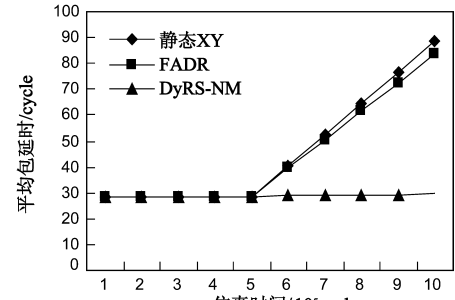
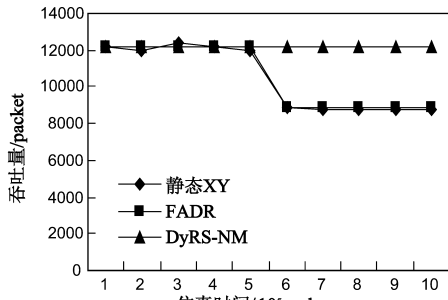
(b) 随误码率增大基于DyRS-NM机制的网络平均延时变化

图10 不同误码率下网络的性能

网络收包率急剧下降.在基于 DyRS-NM 机制的网络中,由于发生瞬时故障后需要数据重传,平均延时会逐渐增加,但增加幅度很小.

在 NoC 中广泛采用的静态 XY 路由算法进行比较.仿真运行 5×10^5 个时钟周期后,分别在三种网络中随机引入一个永久性链路故障,网络的吞吐量和平均包延时如图 11 所示.

第 2 组实验评估出现永久性链路故障下 DyRS-NM 的性能,将其与文献[13]中的容错动态路由 FADR 以及



(a) 网络吞吐量变化

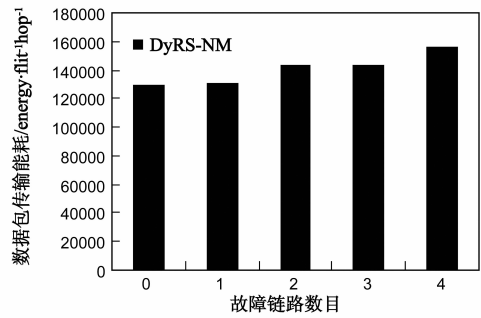
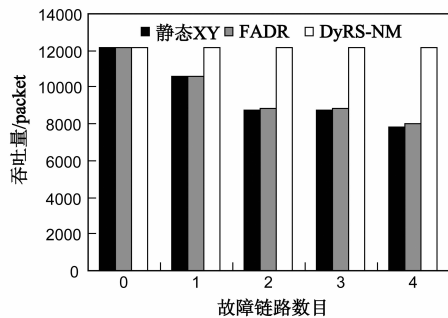
(b) 网络平均包延时变化

图11 发生链路故障后网络吞吐量和平均包延时的变化

从图中可以看出,相比静态 XY 路由和 FADR 路由,DyRS-NM 机制获得了最佳的性能.这是因为,静态 XY 路由无法绕开任何故障链路,导致数据包丢失,从而造成了网络吞吐量明显下降,平均包延时大幅增加.而由于基于网络监控器的全局通信管理,在采用 DyRS-NM 机制的网络中所有数据包会被重传,所以网络吞吐量基本不变化,平均包延时因部分绕道路由而稍许增加.对于 FADR 路由,在有些故障情况下,其可以为每条通信踪迹找到新的可替换的路由路径,绕开故障链路;

但是若源节点和目的节点处于同一维,而该维方向的链路又恰好发生了永久性故障,则 FADR 路由亦无法绕开故障链路正常传输数据,与静态 XY 路由类似,会造成网络性能的降低.

在引入多个链路故障下,三种网络的吞吐量和通信能耗比较如图 12 所示.从图中可以看出,采用 DyRS-NM 机制的网络在多个故障下吞吐量基本保持不变,具有较强的容错能力.但是随着故障链路数目的增加,数据包传输的能耗会增加.

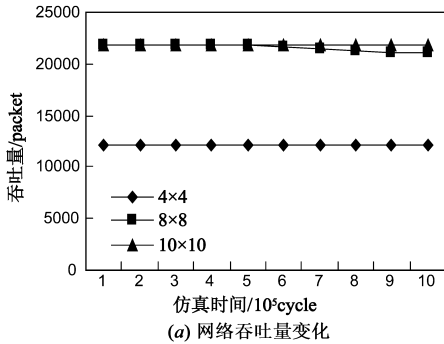


(a) 网络吞吐量变化

(b) 网络通信能耗变化

图12 随着故障链路数目增多网络的吞吐量和通信能耗的变化

为验证路由机制的可扩展性,由 TGFF 软件包^[20]产生一个 48 个节点的随机通信任务图,将其分别映射至 8×8 与 10×10 Mesh 结构 NoC 中,并在仿真运行 5×10^5



个时钟周期后分别引入 6 条和 10 条链路故障,结合图 11,采用 DyRS-NM 机制的不同规模网络下吞吐量和平均包延时如图 13 所示.

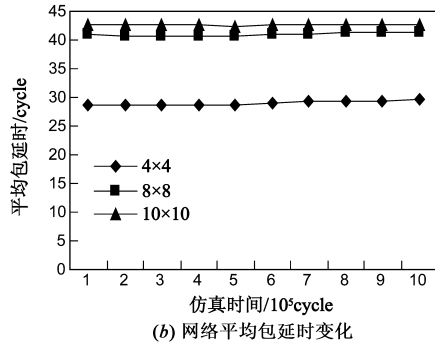


图 13 不同规模网络下发生链路故障后吞吐量和平均包延时的变化

从图中可以看出,在大规模网络中,发生故障后,采用 DyRS-NM 机制仍能保证吞吐量和平均包延时基本不变.

4.2 面积功耗开销

采用 SMIC 0.18 μm 工艺库在 Synopsys Design Compiler 中对基于 DyRS-NM 机制的 NoC 进行了综合,用以评估网络的面积功耗开销.

支持静态 XY、FADR 和 DyRS-NM 三种路由机制的路由器面积和功耗开销如表 1 所示.

表 1 支持三种路由机制的路由器面积和功耗

路由机制	面积(mm^2)	功耗(mW)
静态 XY 路由	0.1983	63.720
FADR 路由	0.1987	77.082
DyRS-NM	0.2095	83.963

与一般不具有容错功能的路由器相比,我们所提出的路由器结构主要增加了重传机制设计以及与网络监控器接口的模块.为了辨别链路的瞬时故障和永久性故障以评估永久性故障情况下三种路由机制的性能,我们在三种路由器中都设计了重传机制,当重传次数超过门限值时,就认为链路发生了永久性故障.因此,相比支持静态 XY 路由、FADR 路由的路由器,支持 DyRS-NM 机制的路由器除了路由计算模块不一致外,主要增加了与网络监控器接口模块,因而在面积和功耗开销上都有所增加.重传机制和与网络监控器接口模块在基于 DyRS-NM 机制的路由器设计中的面积功耗开销比重如表 2 所示.

其中,用于解决瞬时故障的点到点检错重传机制占总面积 18.5%,功耗占总功耗的 31.8%,该部分设计包括每个端口的编码器、解码器和重传缓冲区.与网络监控器的接口部分主要包括发送链路状态信息模块和接收并存储监控器发送来的路由路径信息模块,面积

表 2 基于 DyRS-NM 机制的路由器中增加模块的面积和功耗开销比重

	面积(mm^2)	比例	功耗(mW)	比例
路由器	0.2095	100%	83.963	100%
点到点检错重传机制	0.0387	18.5%	26.68	31.8%
与网络监控器的接口	0.0073	3.5%	6.88	8.2%

和功耗分别占整个路由器的 3.5% 和 8.2%.

相比一般的 NoC 结构,基于 DyRS-NM 机制的 NoC 结构中还需增加一个网络监控器模块.该模块中,C 语言实现的路径分配算法将固化在存储器中,由基于 AHB-Lite 总线扩展的 ARM Cortex-M0 处理器核运行,该处理器核的开销是 12K 门,路径分配算法编译成二进制代码的大小为 25KB.网络监控器中其余各子模块综合后的面积为 0.085mm^2 ,功耗为 14.729mW.然而,为实现 DyRS-NM 机制,整个 NoC 系统只需要一个网络监控器,其开销代价是可以接受的.

此外,为实现 DyRS-NM 机制,每个路由器还需要额外的连线与网络监控器相连,用于传输整个网络的状态和路由路径信息.但是由于 NoC 具有充足的连线资源,该部分开销也认为是可以接受的.

5 结论

本文提出了片上网络监控器的概念,用于获取全局网络状态信息及执行路径分配算法.基于网络监控器,提出了一种动态容错路由机制 DyRS-NM,采用重传方式避免瞬时故障,通过重新路由计算绕开拥塞和永久性故障链路.实验结果表明,相比静态 XY 路由和容错动态路由 FADR,DyRS-NM 机制在可接受的开销代价下获得了更优的性能.

参考文献

[1] 杨盛光,李丽,高明伦,等.面向能耗和延时的 NoC 映射

- 方法[J]. 电子学报, 2008, 36(5): 937 - 942.
- Yang Sheng-guang, Li Li, Gao Ming-lun, et al. An energy and delay-aware mapping method of NoC [J]. Acta Electronica Sinica, 2008, 36(5): 937 - 942. (in Chinese)
- [2] Hu J, Marculescu R. DyAD-smart routing for networks-on-chip [A]. Proc. DAC[C]. San Diego, CA: ACM Press, 2004. 260 - 233.
- [3] Li M, Zeng Q A, Jone W B. DyXY-a proximity congestion-aware deadlock-free dynamic routing method for network on chip [A]. Proc. DAC [C]. San Francisco, CA: ACM Press, 2006. 849 - 852.
- [4] Gratz P, Grot B, Keckler S W. Regional congestion awareness for load balance in networks-on-chip [A]. Proc. HPCA[C]. Salt Lake City, UT: IEEE, 2008. 203 - 214.
- [5] Ramanujam R S, Lin B. Destination-based adaptive routing on 2D mesh networks [A]. Proc. ANCS [C]. San Diego, CA : IEEE, 2010. 1 - 12.
- [6] Tedesco L, Clermidy F, Moraes F. A monitoring and adaptive routing mechanism for QoS traffic on mesh NoC architectures [A]. Proc. CODES + ISSS [C]. Grenoble, France: ACM Press, 2009. 109 - 118.
- [7] Dumitras T, Kerner S, Marculescu R. Towards on-chip fault-tolerant communication [A]. Proc. ASP-DAC [C]. Kitakyushu, Japan: IEEE, 2003. 225 - 232.
- [8] Murali S, Theocharides T, Vijaykrishnan N, et al. Analysis of error recovery schemes for networks on chips [J]. IEEE Design&Test of Computers, 2005, 22(5): 434 - 442.
- [9] Greenfield D, Banerjee A, Lee J G, et al. Implications of rent's rule for noc design and its fault-tolerance [A]. Proc. NOCS [C]. Princeton, NJ: IEEE, 2007. 283 - 294.
- [10] Ali M, Welzl M, Hessler S. A fault tolerant mechanism for handling permanent and transient failures in a network on chip [A]. Proc. ITNG [C]. Las Vegas, NV: IEEE, 2007. 1027 - 1032.
- [11] Rameshan N, Laxmi V, Gaur M S, et al. Minimal path, fault tolerant, QoS aware routing with node and link failures in 2-D mesh NoC [A]. Proc. DFT [C]. Kyoto, Japan: IEEE, 2010. 60 - 66.
- [12] Zou Y, Pasricha S. NARCO: neighbor aware turn model-based fault tolerant routing for NoCs [J]. IEEE Embedded Systems Letters, 2010, 3(2): 85 - 89.
- [13] Hosseini A, Ragheb T, Massoud Y. A fault-aware dynamic routing algorithm for on-chip networks [A]. Proc. ISCAS [C]. Seattle, WA: IEEE, 2008. 2653 - 2656.
- [14] 付斌章, 韩银和, 李华维, 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.
- Fu Bin-zhang, Han Yin-he, Li Hua-wei, 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)
- [15] Ho C, Stockmeyer L. A new approach to fault-tolerant wormhole routing for mesh-connected parallel computers [J]. IEEE Transactions on Computers, 2004, 53(4): 427 - 439.
- [16] Fu B, Han Y, Li H, et al. A new multiple-round dimension-order routing for networks-on-chip [J]. IEICE Transactions on Information and Systems, 2011, E94-D(4): 809 - 821.
- [17] Mejia A, Flich J, Duato J, et al. Segment based routing: an efficient fault-tolerant routing algorithm for meshes and tori [A]. Proc. IPDPS [C]. Rhodes Island, Greece: IEEE, 2006. 84 - 93.
- [18] Ge Fen, Wu Ning. Genetic algorithm based mapping and routing approach for network on chip architectures [J]. Chinese Journal of Electronics, 2010, 19(1): 91 - 96.
- [19] Jesshope C R, Miller P R, Yantchev J T. High performance communications in processor network [A]. Proc. ISCA [C]. Jerusalem, Israel: IEEE, 1989. 150 - 157.
- [20] Dick R P, Rhodes D L, Wolf W. TGFF: task graphs for free [A]. Proc. CODES [C]. Seattle, Washington: IEEE, 1998. 97 - 101.

作者简介



葛 芬(通信作者) 女, 1981 年生于江苏丹阳, 博士, 南京航空航天大学副教授. 主要研究方向为片上网络、专用数字集成电路与嵌入式系统设计.

E-mail: gefen@nuaa.edu.cn



吴 宁 女, 1956 年生于安徽淮南, 硕士, 南京航空航天大学教授, 博士生导师. 主要研究方向数字系统理论与技术、电子系统集成与专用集成电路设计.

E-mail: wunee@nuaa.edu.cn