A Novel Topology-Independent Router Architecture to Enhance Reliability and Performance of Networks-on-Chip

Khalid Latif[1,2], Amir-Mohammad Rahmani[1,2], Ethiopia Nigussie[1], Hannu Tenhunen[1,2]
[1]*Department of Information Technology, University of Turku, Finland*
[2]*Turku Centre for Computer Science (TUCS), Finland*
{khalid.latif, amir.rahmani, ethiopia.nigussie, hannu.tenhunen}@utu.fi

Tiberiu Seceleanu
ABB Corporate Research
Västerås, Sweden
tiberiu.seceleanu@se.abb.com

Abstract

We present the partial virtual-channel sharing (PVS) NoC architecture which reduces the impact of fault on system performance and can also tolerate the faults on routing logic. A fault in one component makes the fault-free connected components out of use and this in turn leads to considerable performance degradation. Improving utilization of resources is a key to either enhance or sustain performance with minimal overheads in case of fault or overloading. In the proposed architecture autonomic virtual-channel buffer sharing is implemented. The runtime allocation of the buffers depends on incoming load and fault occurrence. This technique can be used in any NoC topology and for both 2D and 3D NoCs. The synthesis results for an integrated video conference application demonstrate significant reduction in average packet latency compared to existing VC-based NoC architecture. Extensive quantitative simulation results for synthetic benchmarks are also carried out. Furthermore, the simulation results reveal that the PVS architecture improves the performance significantly under fault free conditions compared to other VC architectures.

Keywords

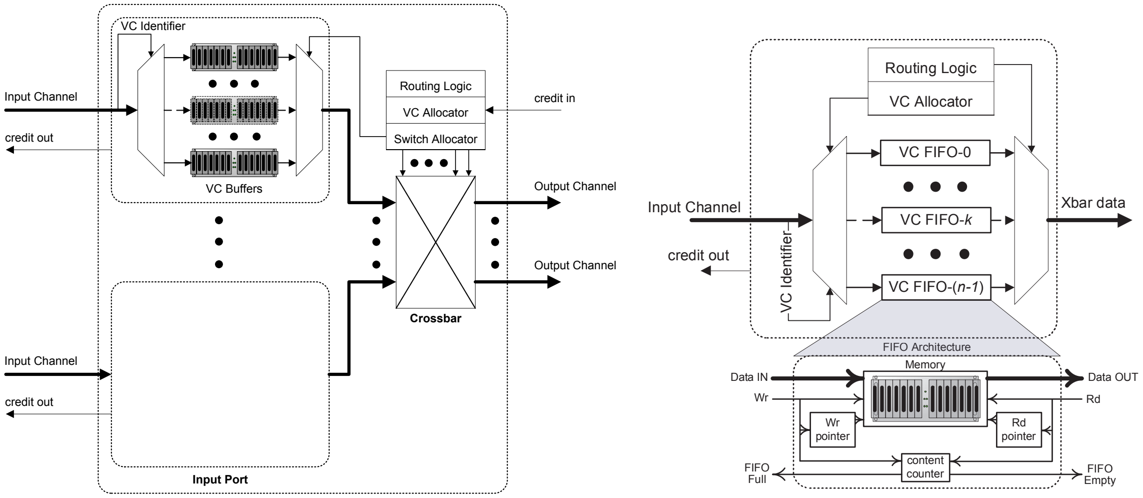
Networks-on-Chip (NoC); Virtual Channel Sharing; Fault Tolerance; Resource Utilization

I. Introduction

The continuous development of semiconductor technology over last few decades have changed our everyday life. Multiprocessing is a promising solution to meet the requirements of near future applications. To get full benefit of parallel processing, a multiprocessor needs efficient on-chip communication architecture [14]. Network-on-Chip (NoC) is a general purpose on-chip communication concept that offers high throughput, which is the basic requirement to deal with complexity of modern systems. A typical NoC based system consists of processing elements (PEs), network interfaces (NIs), routers and channels. The router further contains switch, buffers and routing logic as shown in Figure 1(a). All links in NoC can be simultaneously used for data transmission, which provides a high level of parallelism. It is an attractive solution to replace the conventional communication architectures such as shared buses or point-to-point dedicated links by NoC. NoC provides better scalability than on-chip buses because as more resources are introduced to a system, also more routers and links are introduced to connect them to the network.

Buffers consume the largest fraction of dynamic and leakage power of the NoC node (router + link) [2]. Storing a packet in buffer consumes far more power as compared to its transmission [24]. Thus, increasing the utilization of buffers and reduction in their number and size with efficient autonomic control reduces the area and power consumption. Wormhole flow control [22] has been proposed to reduce the buffer requirements and enhance the system throughput. However, one packet may occupy several intermediate switches at the same time. This introduces the problem of deadlocks and livelocks [14]. To avoid this problem the use of virtual channel is introduced. A typical VC architecture [15] for input port of a router is shown in Figure 1(b). Virtual channel flow control exploits an array of buffers at each input port. By allocating different packets to each of these buffers, flits from multiple packets may be sent in an interleaved manner over a single physical channel. This improves the throughput and reduces the average packet latency by allowing blocked packets to be bypassed.

With the trend of technology and supply voltage scaling and increasing interconnect density, devices are exposed to a large number of noise sources such as capacitive and inductive crosstalk, power supply noise, leakage noise, thermal noise, process variations, charge sharing, soft errors. Due to this, reliability of the manufactured devices is becoming endangered [17][11]. There are a number of fault tolerant solutions available to deal with reliability at different abstraction levels for example routing algorithms [13][25], architectures [16][23] and error control coding schemes [7][20]. Some of the fault tolerant NoC architectures proposed by researchers use intelligent routing algorithms [6][5]. The key problem with this approach is that the fault-free resources which are interconnected with the faulty resource can not be used. This inturn leads to a reduction in system performance. For instance, if there is a link failure in a VC based NoC, the VC buffers connected



(a) Router Architecture [15].

(b) Input Port Architecture.

Figure 1: Conventional Virtual Channel NoC Architecture.

to the failed link can not be used. To reduce the effect of fault on the system performance, such unused resources should be utilized by the system. A well designed network exploits all available resources to sustain performance [1].

In this paper, we propose a novel architecture for NoC routers with autonomic sharing of VC buffers. It enhances the utilization of resources especially in the presence of faults with minimized overhead. The proposed autonomic buffer sharing technique enables the utilization of resources that become inaccessible because of fault and this in turn retain the required performance.

The rest of the paper is organized as follows. In Section II, an overview of the state of the art in NoCs by focusing on fault tolerance requirements is presented. In Section III different fault scenarios for NoC router and how fault on one resource affects the performance and utilization of other resources are discussed. The proposed architecture is presented in Section IV. Section V explains the system performance sustainability under faults. Finally experimental results are presented and conclusions are drawn.

II. Related Work

Faults can be categorized as permanent, intermittent, and transient [4]. Different techniques are required to deal with different kind of faults. A single fault tolerance method is not an optimal solution for all types of faults [11]. Neishabouri et al. [16] propose *Reliability Aware Virtual Channel* (RAVC) architecture for NoC. RAVC enables both dynamic VC allocation and reliability aware sharing among input channels. The proposed architecture allocates more memory to the busy channels and less to the idle channels. RAVC shows significant reduction in average packet latency (APL) for normal system operation at the expense of complex memory control logic. Under faults, if one router node is marked faulty, the approach deals well with the situation to balance the traffic load but if few router resources are faulty, the approach can not utilize the rest of the corresponding router resources.

Concatto et al. [3] present highly reconfigurable fault tolerant NoC router architecture. The proposed architecture can dynamically stop using the faulty flit buffer unit and to sustain the performance, it can borrow the flit buffer units from the neighboring channels, if necessary. The approach sustains the performance well under faults without wasting the whole FIFO and just bypassing the faulty flip-flops. The granularity level of bypassing and borrowing the flit buffer units makes the control logic very complex. Thus, the solution is not area and power efficient.

In typical NoC architectures, a fault in router or network interface (NI) results in an unconnected resource. Lehtonen et al. [12] addresses the fault tolerance in such situations by introducing multiple-NI architectures. The approach improves the system fault tolerance on topology level. The throughput performance can also be enhanced by utilizing multiple routes and reducing the number of communication hops. The solution is not area efficient. The technique is not power efficient for synchronous systems unless power gating is introduced for all NIs. The FIFO buffers use major fraction of silicon area in NI. In case of multiple-NI architectures, all NIs are not use at a time in normal operations. Thus, to reduce the number of buffers (silicon area) and enhance the resource utilization, if each NI shares the buffer with one of the other neighboring router

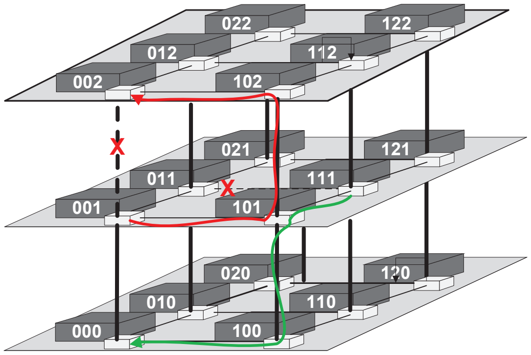


Figure 2: Routing in presence of faulty links.

ports, area overhead can be significantly reduced. It will reduce the static power consumption for synchronous systems. In [26], authors propose the dual connected mesh structure (DCS) like [12] and face the similar problems.

Lotfi-Kamran et al. [13] present a decision making routing algorithm to avoid congestion in 2D NoC architectures. In addition, the proposed dynamic routing approach can tolerate a single link failure. In case of inter-router link failure, apart from the faulty link, rest of the resources like VC buffers and control logic cannot be utilized by the proposed technique. Similar is the issue with other fault tolerant routing techniques. We propose a NoC architecture to deal with faulty conditions by utilizing the available communication resources.

The main motivation of this work is to design a NoC architecture which takes into considerations of the discussed issues. The proposed architecture has a better tradeoff between performance, power consumption and area. It also reduces the impact of fault on performance.

III. Problem Description

As discussed in Section I, a typical NoC system consists of PEs, NIs, routers and communication links. Fault tolerance techniques for PEs are out of the scope of this paper. Approaches like [12][26] can be used to create fault tolerant NIs. Here we address the fault tolerance for routers and communication links by considering different scenarios.

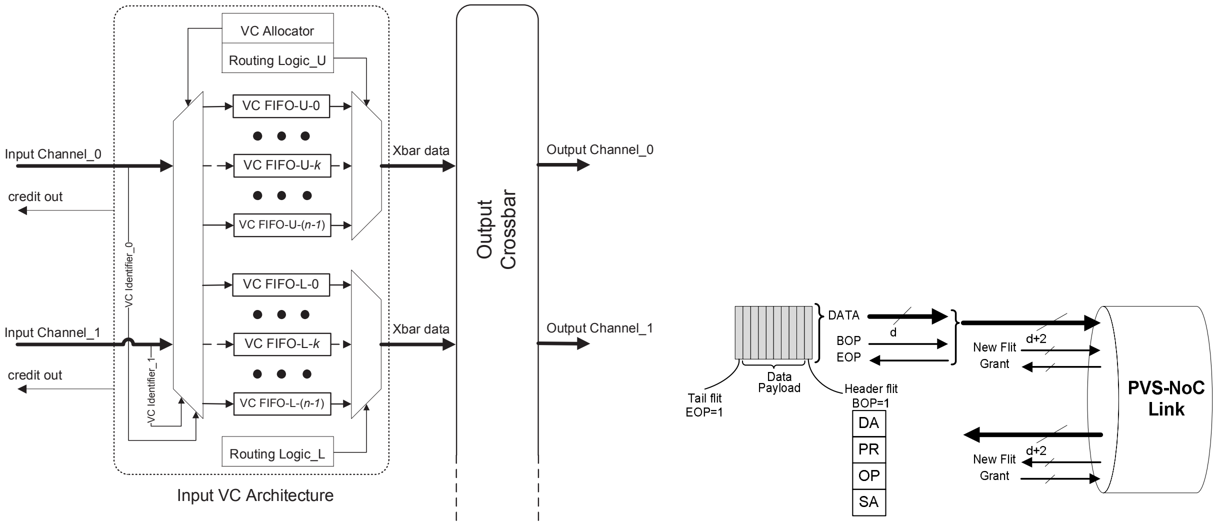
CASE-1: Faulty Links: Consider that a fault occurred on a network inter-router-link due to for example electromigration, strike of alpha particle, driver/receiver circuit failure or any other fault. In typical architectures, the input/output buffers connected to the link cannot be utilized any more. Assume that there are X faulty links in NoC based system and each input port contains V virtual channels with buffer depth d and each flit size or buffer width is f bits. If there is a small VC controller for each virtual channel, the resources which cannot be used by the system after the fault occurs on a physical channel can be estimated by Eq. 1.

$$\text{Resource Overhead} = X \cdot V \cdot d \cdot f_{\text{(Memory Cells)}} + X \cdot V \cdot f_{\text{(wires in Output Crossbar)}} + X_{\text{(VC Control Logic)}} \tag{1}$$

Instead of using power gating technique, it would be better to utilize the resources mentioned in Eq.1 to improve the system performance as well as to reduce the impact of fault on performance degradation. Consider the NoC platform shown in Figure 2. For node '001' as source and node '002' as destination, the route of packet will be just to go up vertically in normal situation. If the communication link between the nodes is broken as shown in Figure 2, the packet will be rerouted via nodes '101' and '102'. Similar is the situation, when node '111' is the source and node '000' is the destination. Now, the resources on the new routes and especially the router at node '101' will be overloaded. The input ports from '111' and '001' will be overloaded due to the faulty links, while other inputs might not be congested.

CASE-2:Deadlock Situation: VCs are used to avoid deadlock by inserting array of buffers at each input port as discussed in section I. By allocating different packets to the VC buffers, multiple packets can be transmitted in interleaved manner over a single channel. Now consider the situation that there are N VCs and out of them, *N-1* VCs have a fault. This fault can occur in any part of FIFO architecture like Rd./Wr. controller, content counter or memory flip-flops, see Figure 1(b). In this case, the architecture will be equivalent to the typical non-VC architecture. Thus, only one packet can be transmitted and deadlock can occur.

CASE-3:Load Management: If a FIFO is faulty due to any of the reasons mentioned in CASE-2, the FIFO cannot be used. In that case, its port becomes overloaded. The number of packets, which can be transmitted in interleaved manner is proportional to the number of VCs. Now consider that the neighboring port is free and thus the corresponding VC buffers are also free to use. The overloaded port cannot utilize the free resources to manage the load. Such faults become the bottleneck



(a) Proposed PVS Architecture.

(b) Data transmission format.

Figure 3: Proposed PVS NoC architecture and data transmission format.

for overall system performance. The load management mechanism is not considered at micro-architecture level, although there are traffic routing algorithms dealing with load balancing as discussed in section II.

CASE-4:Faulty Routing Logic: If a fault occurs in routing logic of VC allocator due to a soft error [11] or any other reason, the physical link and VC buffers cannot be used to route the packets. Then the traffic needs to be re-routed using some fault-tolerant routing algorithm as discussed in section II. The use of the non-minimal route consumes significant amount of power. In this case, not only the system throughput is considerably reduced but also there is unnecessary power consumption for VC buffers and control logic. Congestion on some nodes and power consumption due to non-minimal paths may raise thermal issue as there is vicious circle between heat and power consumption [21]. The scenario discussed in CASE-1 can be considered. If there is a fault on corresponding VC routing logic instead of inter-router links, the set of resources cannot be used.

IV. The Proposed Partial Virtual-channel Sharing (PVS) Approach

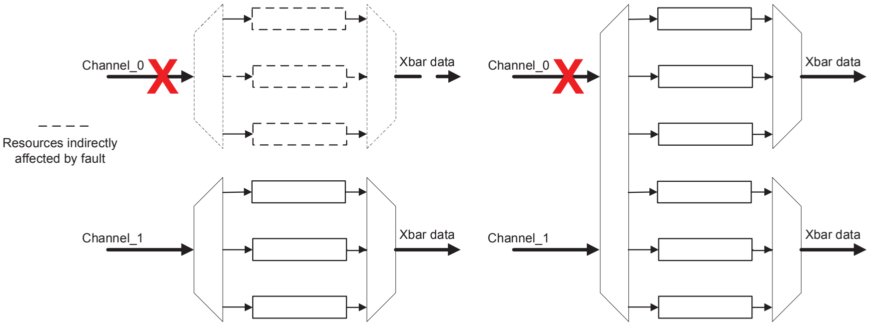
To address the fault tolerance issues discussed in section III, partial virtual-channel sharing NoC (PVS-NoC) architecture has been proposed. Due to sharing, the proposed approach enhances VC utilization as well because free buffers can be utilized by other channels. The improved VC utilization directs to reduce the number of VC buffers and sustain the system performance. VC utilization can be further improved by sharing them among all the input ports. However, full sharing increases the control logic complexity and power consumption due to a larger input crossbar. Thus a tradeoff between resource utilization and power consumption is needed.

Instead of sharing the VC buffers among all the input ports, PVS approach shares the VC buffers among a few input ports according to the communication requirements. With this technique, the buffer utilization is increased and approaches close to the utilization level of fully shared architecture without significant silicon area and power consumption overhead because of reduced input crossbar size. Overall, PVS approach is a tradeoff between system throughput, resource utilization and power consumption.

Data is injected to the network in the form of packets produced by Network Interface (NI). The packet format is shown in Figure 3(b). While receiving a packet, NI also de-packetizes it and delivers the payload to the PE. Header flit carries the source address (SA), the operational code (OP), the priority level (PR) and the destination address (DA). The beginning of packet (BOP) and end of packet (EOP) are the indicators of header and tail flits respectively. The PVS-NOC architecture can be divided as input and output architectures.

A. The Input Controller and Autonomous Buffer Allocation

The contribution of PVS approach stands on the input side. The buffer utilization is enhanced by dynamically allocating the free buffers to the overloaded ports. Input architecture for sharing one or multiple buffers among a group of multiple channels can be extended to any number of input ports according to the topology requirements. Normally, the processing core uses the dedicated buffer for packet injection to the network and does not share it with any other router ports.



(a) Conventional architecture.

(b) PVS architecture.

Figure 4: Effect of fault on resource utilization.

In PVS approach, the input crossbars and control logic are responsible for buffer allocation and receiving the data packets. Input PVS architecture for one group, where two channels share the VC buffer is shown in Figure 3(a). Input architecture uses distributed routing logic but central VC allocation for the group of channels sharing the buffers. Both VC allocator and routing logic operate independently without communicating with other group's control logics. The task of VC allocator is to keep track of free buffers and allocate them to the incoming traffic. After allocation, routing logic computes the route for the packet and selects the port on output crossbar for packet transmission. The distributed nature of the routing logic makes the PVS architecture fault tolerant. If one routing element is faulty, only the corresponding buffers are affected and the rest of the resources in the group can be used to route the packets. Distributed routing logics also reduces the communication overhead and power consumption.

VC allocator works only when the *BOP* or *EOP* signal is received. Once the buffer has been tied to the requesting router, VC allocator goes into the sleep mode to save power. When an *EOP* signal is recieved, VC allocator marks the buffer as free. It does not consume power until the next *BOP* signal is received.

B. The Output Controller and Packet Transmission

The *Output* part is modeled by a typical N×N crossbar with central control logic and deals with packet transmission. 'N' represents the total number of ports including the port for processing element. The crossbar size can be customized according to the topology requirements. A wormhole flow control is used for packet transmission, which makes efficient use of buffer space as small number of flit buffers per VC are required [22].

C. Virtual Channel Selection for Sharing

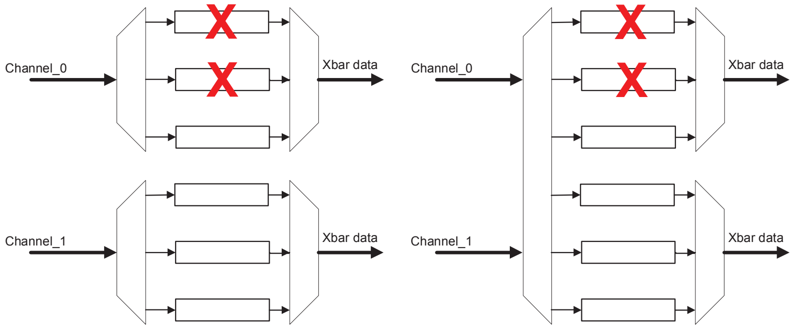
For load balancing, the selection of ports to share the VC buffers should be made on the basis of number of router ports, the number of VCs per port, input bandwidth requirements for each input port and the number of groups (sharing VCs). The input ports are grouped to share the buffers in such a way that the total incoming load for the whole group approaches the value of average bandwidth requirements per VC times the number of VC buffers in the current group. The detailed algorithm is presented in [10].

V. Performance Sustainability under Faults

The main feature of PVS-NoC architecture is to retain the system performance till certain level after the occurrence of fault. In NoC based interconnection platform, the fault can occur in three types of components: physical link, buffer and the controller. Here, the problems mentioned in section III are addressed.

CASE-1: Faulty Links: In case of fault on physical link, buffers and control logic cannot be used by the NoC based system as can be observed in Figure 4(a). If a fault occurs in 'Channel_0', its VC buffers and routing logic can not be used to route a packet. Now, consider this situation for PVS approach as shown in Figure 4(b). If the fault occurs on 'Channel_0', the 'Channel_1' can utilize the VC buffers and control logic to enhance the system throughput and avoid the unnecessary static power consumption by the VC buffers and control logic.

CASE-2:Deadlock Situation: If all the VC buffers become faulty except one VC buffer for a single port. The architecture becomes equivalent to the non-VC architecture and deadlock can occur as can be seen in Figure 5(a). For input port of 'Channel_0', the fault occurs all the connected VC buffers except the lowest VC buffer. The architecture is equivalent to the non-VC architecture and deadlock can occur. Now consider the PVS architecture shown in Figure 5(b). If only one VC buffer is left in upper half of VCs, 'input channel_0' can use the VC buffers of lower half and packets can still be transmitted on 'input channel_0' in interleaved manner. Thus, deadlock can be avoided in this case.



(a) Conventional architecture.

(b) PVS Architecture.

Figure 5: Effect of fault on load management.

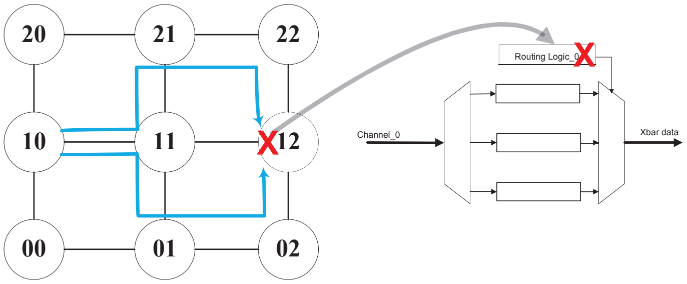


Figure 6: Impact of faulty routing logic on packet routing.

CASE-3:Load Management: To balance the load on input buffers and provide a relief to the loaded ports, if the input ports share the VC buffers, the extra load due to faulty links will be shared by all the ports. PVS-NoC architecture uses this sharing approach to retain the performance in case of fault on any communication resource.

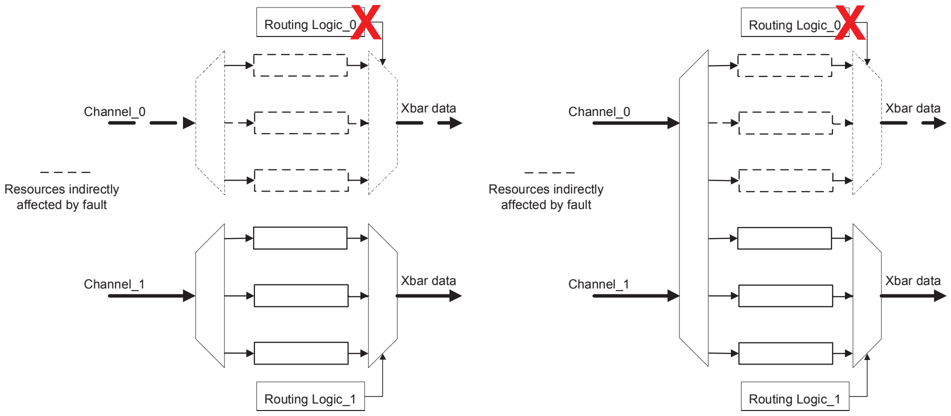
In typical VC architecture, if one VC becomes faulty, the port becomes overloaded as shown in Figure 5(a). The 'Channel_0' port becomes overloaded as many packets are waiting to be routed but due to less number of functioning VCs, the waiting packets cannot be routed through 'Channel_0'. In case of PVS architecture, it can be observed from Figure 5(b), the available VC buffers in lower half can be used to route the blocked packets. Thus, the fault impact will be distributed equally over 'Channel_0' and 'Channel_1'.

CASE-4:Faulty Routing Logic: Consider the 3×3 NoC mesh shown in Figure 6. If the fault occurs at node '12' in the routing logic for input port from node '11'. The packets for node '12' form node '11' will be re-routed through the paths shown with blue color. In case of such a fault, all the resources on input port cannot be used by the NoC system as shown in Figure 7(a). The packets cannot be routed through 'Channel_0' and the VC buffers cannot be utilized, if the fault occurs on 'Routing_Logic_0'.

In case of PVS approach, if the fault occurs in any routing logic element, the VC allocator marks the routing logic faulty and the VC buffers controlled by this routing logic are not allocated to any packet for the purpose of transmission. If the fault occurs on 'Routing_Logic_0', the 'Channel_0' can still be used with 'Routing_Logic_1' as shown in Figure 7(b). Only the lower half of buffers will be available for 'Channel_0' and 'Channel_1'. The packets do not need to be rerouted and thus the fault can be tolerated with minimized overhead. Similarly, the faults occurred in multiplexer or multiplexer output link of 'Routing_Logic_0' can be tolerated. Another interesting case is that if fault occurs on 'Routing_logic_0', its multiplexer, and 'Channel_1' simultaneously, the PVS architecture is still able to pass the packets by using 'Routing_Logic_1' and 'Channel_0'. However, in the conventional architecture this case leads to two channels failure.

VI. Simulation results

To demonstrate performance characteristic of the proposed architecture (PVS-NoC) under faults, a cycle-accurate NoC simulation environment was implemented in HDL. The packets had a fixed length of seven flits, the buffer size was eight flits and the data width was set to 32 bits. The 5×5 2D mesh topology was used for interconnection. Each input port had 4 VCs. With same parameters, typical virtual channel and FVS-NoC architectures were analyzed. Static XY wormhole routing algorithm was used for both non-faulty and faulty scenarios. For the faulty scenario, it is assumed that an appropriate fault detection mechanism (test unit) similar to the one used in [9] detects the faulty links, and stores the fault information in the



(a) Conventional architecture: Faulty routing logic requires re-routing of packets.

(b) PVS Architecture: Routing logic fault tolerated.

Figure 7: Routing logic fault.

configuration register of the routers connected to the faulty link. In this case, these routers will not send any traffic to the corresponding links and will reroute packets using one of the other adjacent routers. This results in modification of static XY routing algorithm in deadlock free manner for faulty scenarios. The PVS approach with grouping combination of (2, 2, 1) was used, where '1' represents the buffer dedicated to the local PE and '2' shows that the group of two ports share the VC buffers.

A. *Synthetic Traffic Analysis*

We compare the simulation results in terms of average packet latency (APL) and saturation points for two cases: a normal network with no fault using typical, PVS, and FVS virtual channel management policies, and an example of a faulty network with two faulty links using typical and PVS virtual channel management schemes. The system sustainability under link faults with PVS approach has been discussed in CASE-1 of Section V.

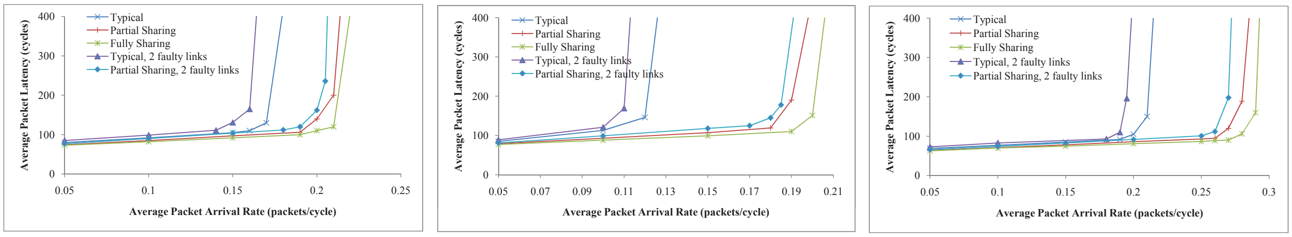
In traffic analysis, the performance of the network was evaluated using latency curves as a function of the packet injection rate. The packet latency was defined as the time duration from when the first flit is created at the source node to when the last flit is delivered to the destination node. For each simulation, the packet latencies were averaged over 50,000 packets. Latencies were not collected for the first 5,000 cycles to allow the network to stabilize. To perform the simulations, uniform, transpose and Negative Exponential Distribution (NED) [18] traffic patterns were used. The NED is a synthetic traffic model based on Negative Exponential Distribution where the likelihood that a node sends a packet to another node exponentially decreases with the hop distance between the two cores. This synthetic traffic profile accurately captures key statistical behavior of realistic traces of communication among the nodes.

The latency curves for uniform, transpose and NED traffic patterns are shown in Figure 8. It can be observed for all the traffic patterns, the PVS-NoC architecture saturates at higher injection rates as compared to the typical VC architecture but slightly less than FVS-NoC architecture. In case of proposed architecture, bandwidth limitations are managed by proper resource utilization without increasing the communication resources and making the load more balanced. The saturation point of PVS-NoC is just before FVS-NoC because FVS-NoC provides more buffer utilization by sharing the VC buffer among all input ports, however FVS-NoC is not a power efficient solution as verified for realistic application traffic pattern. In case of faulty networks, the curves reveal that for all the traffic patterns the proposed architecture has less performance loss compared to the typical architecture under faulty conditions. The reason is that the VCs connected to the faulty links are utilized by the other channel which helps to reduce the average packet latency.

B. *Fault Tolerance for Routing Logic*

As discussed in CASE-4 of Section V, if the fault occurs on routing logic, PVS architecture can tolerate the fault and packets do not need to be re-routed. To demonstrate that, the 6×6 2D mesh NoC with two VCs per port and two faulty routing logics was simulated. The values of the other system parameters are the same as in Section VI.

The latency curves with two faulty routing logics for the 6×6 mesh 2D network are shown in Figure 9 for uniform, transpose and NED traffic patterns. It can be observed for all the traffic patterns, the PVS-NoC architecture saturates at higher injection rates as compared to the typical VC architecture with faulty routing logic. The reason is that the PVS

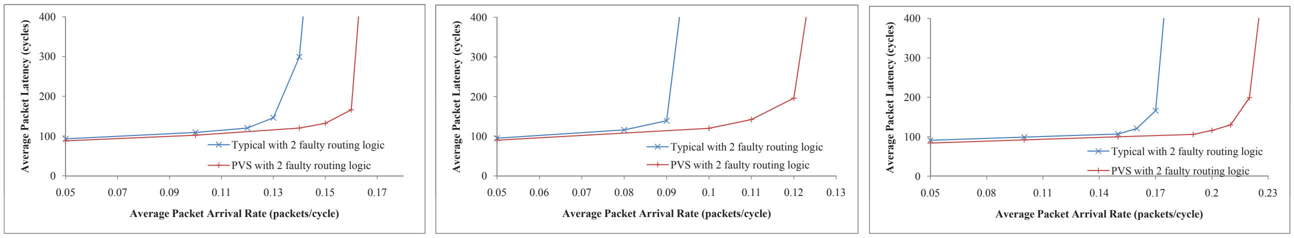


(a) Uniform traffic load.

(b) Transpose traffic load.

(c) NED traffic load.

Figure 8: APL vs. Packet injection rate for 5×5 mesh 2D NoC with (2, 2, 1) combination of PVS approach.



(a) Uniform traffic load.

(b) Transpose traffic load.

(c) NED traffic load.

Figure 9: APL vs. Packet injection rate for 6×6 mesh 2D NoC with two faulty routing logics.

architecture does not re-route the packets in case of fault on routing logic and other routing logics with in the sharing group can be used to route the packets through same channel, and thus the average hop-count is not affected.

C. Real Benchmark

The encoding part of video conference application with sub-applications of H.264 encoder, MP3 encoder and OFDM transmitter was simulated. The details of this application model are presented in [19]. In order to estimate the power consumption, the high level NoC power simulator presented by [8] was extended to support 3D-NoC architectures. The power consumption analysis of the interconnection network (NoC switches, bus arbiters, intermediate buffers, and interconnects) is performed using 35nm standard CMOS technology. The simulation results for average packet latency (APL), power consumption and average router silicon area for video conference encoding application are shown in Table.I.

The area of the 3D-symmetric-mesh-based routers (with 7×7 crossbars) were computed after synthesized on CMOS 65nm *LPLVT STMicroelectronics* standard cells using Synopsys Design Compiler. The simulation results for the average router silicon area for a 3×3×3 3D NoC are shown in Table.I. Each input port has 4 VCs buffers. The buffers size was 8 flits and the data width was set to 32 bits. Here, the average silicon area is reported because different sharing combinations for PVS according to algorithm I have different crossbar sizes and thus different silicon area. The figures given in the table demonstrate, the area overheads of the proposed PVS technique is reasonable when compared to a fully shared virtual channel technique.

The PVS-NoC showed around 21% reduction in power consumption and about 7% reduction in silicon area but about 6% more APL over the FVS-NoC architecture. On other hand, the PVS-NoC showed approximately 22% reduction in APL value but about 8% more power consumption and about 4% more silicon area over symmetric 3D-NoC architecture. Thus, the proposed architecture PVS-NoC provides an optimal tradeoff between APL, power consumption and silicon area.

3D NoC Architecture	Power Consumption (W)	Average Packet Latency (cycles)	Average Silicon Area (μm^2)
Typical Symmetric 3D NoC (7x7)	1.587	186	195154
PVS-3D-NoC	1.713	144	203596
FVS-3D-NoC	2.182	136	220282

Table I: Experimental result for average power consumption and APL of video conference encoder application.

VII. Conclusions

A novel NoC architecture which retains the system performance during occurrence of faults has been presented. This architecture can also tolerate routing logic fault without additional logic. The presented architecture has better tradeoff between resource utilization, system performance and power consumption than conventional VC based architectures. The proposed architecture was simulated with uniform, transpose, and negative exponential distribution (NED) synthetic benchmarks. Also, the architecture was simulated for video conference encoder application. Simulation results showed that average packet latency of PVS-NoC was significantly lower compared to typical VC based NoC architecture. In case of video conference encoder application, the PVS-NoC has consumed 21% less power and takes 7% smaller area than Full Virtual channel Sharing Network-on-Chip. On other hand, the PVS-NoC has consumed 8% more power and takes 4% more area than typical virtual channel architecture.

REFERENCES

[1] BALFOUR, J. AND DALLY, W. J. Design tradeoffs for tiled cmp on-chip networks. In *Proceedings of the 20th annual international conference on Supercomputing (ICS)*. pp.187-198.

[2] BANERJEE, N., VELLANKI, P., AND CHATHA, K. A power and performance model for network-on-chip architectures. In *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*. Vol.2. pp.1250-1255.

[3] CONCATTO, C., MATOS, D., CARRO, L., KASTENSMIDT, F., SUSIN, A., COTA, E., AND KREUTZ, M. Fault tolerant mechanism to improve yield in NoCs using a reconfigurable router. In *Proceedings of the 22nd Annual Symposium on Integrated Circuits and System Design: Chip on the Dunes*. SBCCI '09. ACM, New York, NY, USA.

[4] CONSTANTINESCU, C. Trends and challenges in vlsi circuit reliability. In *Proceedings of IEEE Micro'03*. pp.14-19.

[5] DUMITRAS, T. AND MARCULESCU, R. On-chip stochastic communication [soc applications]. In *Design, Automation and Test in Europe Conference and Exhibition, 2003*. pp.790-795.

[6] FICK, D., DEORIO, A., CHEN, G., BERTACCO, V., SYLVESTER, D., AND BLAAUW, D. A highly resilient routing algorithm for fault-tolerant NoCs. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09*. pp.21-26.

[7] FRANTZ, A. P., KASTENSMIDT, F. L., CARRO, L., AND COTA, E. Dependable network-on-chip router able to simultaneously tolerate soft errors and crosstalk. In *Test Conference, 2006. ITC '06. IEEE International*. pp.1-9.

[8] GUINDANI, G., REINBRECHT, C., RAUPP, T., CALAZANS, N., AND MORAES, F. NoC power estimation at the rtl abstraction level. In *Symposium on VLSI, 2008. ISVLSI '08. IEEE Computer Society Annual*. pp.475-478.

[9] JANTSCH, A. AND (EDS.)., H. T. 2003. *Networks on Chip*. Kluwer Academic Publishers.

[10] LATIF, K., RAHMANI, A.M., RAO, V.K., SECELEANU, T., LILJEBERG, P., AND TENHUNEN, H. Enhancing Performance of NoC-Based Architectures using Heuristic Virtual-Channel Sharing Approach. In *proceedings of 35th IEEE Annual Computer Software and Applications Conference (COMPSAC)*. 2011. pp. 442-447.

[11] LEHTONEN, T. On fault tolerance methods for networks-on-chip. *Ph.D. Thesis, University of Turku, Finland*, 2009.

[12] LEHTONEN, T., LILJEBERG, P., AND PLOSILA, J. Fault tolerance analysis of NoC architectures. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*. pp.361-364.

[13] LOTFI-KAMRAN, P., RAHMANI, A. M., DANESHTALAB, M., AFZALI-KUSHA, A., AND NAVABI, Z. EDXY - A low cost congestion-aware routing algorithm for network-on-chips. *J. Syst. Archit.* 2010. *56,* 7, pp.256-264.

[14] LUCA BENINI, G. D. M. 2006. *Networks On Chips: Technology And Tools*. Morgan Kaufmann Publishers.

[15] MULLINS, R., WEST, A., AND MOORE, S. Low-latency virtual-channel routers for on-chip networks. In *Computer Architecture, 2004. Proceedings. 31st Annual International Symposium on*. pp.188-197.

[16] NEISHABURI, M. AND ZILIC, Z. Reliability aware NoC router architecture using input channel buffer sharing. In *ACM Great Lakes Symposium on VLSI*. pp.511-516.

[17] PASRICHA, S. AND DUTT, N. 2008. *On-Chip Communication Architectures: System on Chip Interconnect*. Morgan Kaufmann.

[18] RAHMANI, A., AFZALI-KUSHA, A., AND PEDRAM, M. 2009. NED: A Novel Synthetic Traffic Pattern for Power/Performance analysis of Network-on-chips Using Negative Exponential Distribution. In *Journal of Low Power Electronics (American Scientific Publishers)*. Vol.5. pp.396-405.

[19] RAHMANI, A., LATIF, K., RAO, V., LILJEBERG, P., PLOSILA, J., AND H, T. Congestion aware, fault tolerant, and thermally efficient inter-layer communication scheme for hybrid noc-bus 3D architectures. In *NOCS '11: In Proceedings of ACM/IEEE International Symposium on Networks-on-Chip*. pp.65-72.

[20] SHEN, J.-S., HUANG, C.-H., AND HSIUNG, P.-A. Learning-based adaptation to applications and environments in a reconfigurable network-on-chip. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*. pp.381-386.

[21] VADDINA, K. R., NIGUSSIE, E., LILJEBERG, P., AND PLOSILA, J. Self-timed thermal monitoring of multicore systems. In *12th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS'09)*. pp.246-251.

[22] WILLIAM J DALLY, B. P. T. 2004. *Principles and Practices of Interconnection Networks*. The Morgan Kaufmann Series in Computer Architecture and Design.

[23] YAGHINI, P., EGHBAL, A., PEDRAM, H., AND ZARANDI, H. Investigation of transient fault effects in an asynchronous NoC router. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*. pp.540-545.

[24] YE, T., BENINI, L., AND DE MICHELI, G. Analysis of power consumption on switch fabrics in network routers. In *Design Automation Conference (DAC), 2002. Proceedings. 39th*.pp.524-529.

[25] ZHANG, Z., GREINER, A., AND TAKTAK, S. A reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on-chip. In *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*. pp.441-446.

[26] ZONOUZ, A., SEYRAFI, M., ASAD, A., SORYANI, M., FATHY, M., AND BERANGI, R. A fault tolerant NoC architecture for reliability improvement and latency reduction. In *12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools*. pp.473-480.