J Electron Test
DOI 10.1007/s10836-013-5398-4

A Fault Tolerant Hierarchical Network on Chip Router Architecture

M. H. Neishaburi · Zeljko Zilic

Received: 15 March 2012 / Accepted: 19 July 2013

Abstract Continuing advances in the processing technology, along with the significant decreases in the feature size of integrated circuits lead to increases in susceptibility to transient errors and permanent faults. Network on Chips (NoCs) have come to address the demands for high bandwidth communication among processing elements. The structural redundancy inherited in NoC-based design can be exploited to improve reliability and compensate for the effects of failures. In this paper, we propose an enhanced fault tolerant microarchitecture with deadlock-free routing for Hierarchical NoCs. The proposed router supplies dynamic Virtual Channel (VC) Allocation, and it employs a high-performance fault tolerant control flow, handling both transient and permanent faults in hierarchical networks without extra retransmission buffer requirements. Experimental results show a significant improvement in reliability as well as decreases in the average latency and energy consumption.

Keywords Network on chip · Hierarchical topology · Fault-tolerant

1 Introduction

Due to extensive advances in semiconductor process technology, a large number of processing elements (cores and resources) can be integrated into a single chip. To gain maximum utilization of these resources and cores, they need to get connected through an environment that enables a rapid interexchange of data [3]. Such embedded cores rely on the bandwidth and performance offered by an on-chip interconnection to fulfill their computational tasks. Traditional bus and crossbar architecture can no longer provide the raising scalability needs and the growing bandwidth requirement in Multiprocessor System on Chips (MPSoCs) within a reasonable area and power envelope. By supporting better modularity, scalability and higher bandwidth, NoC architectures supply a practical alternative for traditional SoC interconnect approaches [3]. While substantial bandwidth and concurrent communication capabilities are promised by NoC architecture, it is subject to performance degradation due to the reliability issues arising from manufacturing and testing challenges in deep submicron technology [1, 19]. Transient faults, including those caused by crosstalk, charge sharing, substrate and power supply noise pose a significant challenge to ensuring signal integrity in deep submicron process technologies [1, 19]. Additionally, as processing technology scales, the prominence of permanent faults resulting from electro migration and manufacturing challenges intensifies. Several factors, including high operating frequency, low voltage levels, small noise margins and reduced logic depth contribute to ever-increasing susceptibility of on-chip networks to faults [1, 19].

A flow control scheme coordinates the allocation of resources in the network while a packet progresses along a route. The key resources in most interconnection networks are the channels and the buffer. Hence, a flow-control method not only has to ensure that packet transmissions occur with no drops due to errors in on-chip network elements, but it has to also allocate these resources in an efficient manner. Buffers consume around 46 % of power [9] inside on-chip routers, as a result buffer management has become a challenge for NoC designers. In general, as the power density increases due to increase in the number of buffers, so does the temperature, accelerating device degradation and reducing reliability and lifetime of the circuit. Thus, the effective and resourceful management of buffers and input and output channels in

Responsible Editor: M. Violante

M. H. Neishaburi (✉) · Z. Zilic
Department of Electrical Engineering, McGill University, Montreal, QC, Canada
e-mail: mh.neishabouri@mail.mcgill.ca

Z. Zilic
e-mail: zeljko.zilic@mcgill.ca

NoC routers has a crucial effect on performance and reliability of interconnection networks [12, 19].

Another important consideration in the NoC domain is scalability. With a significant increase in the number of processing elements and the existence of heterogeneous subnets, many aspects of on-chip networks, such as routing, topology, and flow-control, should be revised from the perspective of scalability [4, 20]. One of the underlying concepts that can improve scalability of an on-chip network is hierarchy [2, 6, 20]. By resorting to hierarchical management paradigm, at low-level of hierarchy, large number of processing elements inside an on chip network can be partitioned into subnets (nodes), while at higher level of hierarchy the network can be seen as an interconnection of subnets. As a consequence of such hierarchical schemes, subnets may differ in terms of topology, routing and flow control. In fact, the hierarchy in the network topology will be instrumental for design, analysis, testing and management of NoCs with large number of cores [17, 20].

In this paper we focus on online error diagnosis and proposed an advanced fault tolerant hierarchical router. Major features of the proposed fault tolerant hierarchical router are 1) it offers more reliability and better performance in the case of failures in the network, 2) it offers a new fault-tolerant flow control that facilitates packet resubmission without the need for extra buffers, 3) it manages to keep network in the state of deadlock-free in the case of failures. To the best of our knowledge this paper is the first study that considers fault-tolerant issues in hierarchical networks.

2 Related Work & Motivation

Exploiting hierarchy to improve scalability and performance of both on-chip and off-chip interconnections is investigated in a large body of research [5–7, 13]. The authors in [5] proposed a hybrid ring/mesh interconnect topology to remove the limitations of long diameters in a large mesh based topology network. They reduce the average number of hops for global traffic by partitioning a two-dimensional mesh into several sub-meshes and then connecting them using a global interconnect. Compared to the traditional 2D-mesh, they have shown that hybrid ring/mesh architectures indeed have a positive effect on the average hop count.

In [7], a deadlock-free routing algorithm for hierarchical NoCs is proposed, providing better performance than using any single routing algorithm. This study also shows that the hierarchical routing algorithms lead to smoother flow of network traffic. However, the deadlock-free routing algorithm proposed in [7] ignored the possibility of failures inside subnets. In particular, as we will show later, if the fault-tolerant routing algorithm inside subnets performs dynamic reconfiguration and routing tables get updated, deadlocks might happen.

In [7, 20] authors have studied different aspects of hierarchical on-chip network, and they have noticed that such hybrid interconnections result to smoother flow of network traffic. However, they have not explored such hierarchical interconnections from the fault-tolerance perspective.

Figure 1 illustrates architecture of a generic NoC router; this router uses virtual channel flow control and wormhole switching and has two pipeline stages [11]. It consists of five basic elements: Routing Unit, VC allocator, Switch allocator, input channels and a Crossbar. Network Interface (NI) converts the local Processing Element (PE) messages to the acceptable format of NoC routers and vice versa.

A flit is a part of a packet that conveys data and control information - it is categorized into Header-Flit (H-F) that keeps the source and the destination addresses, Data-Flits (D-F) that involves a part of a packet and Tail-Flit (T-F) that represents end of a packet. A network topology determines how routers are interconnected; on-chip topologies can be regular such as mesh, irregular or combination of both (hierarchical).

Figure 2 illustrates a hierarchical network topology; The Subnet3 has a regular mesh topology, while routers inside subnet2 are connected according to an irregular topology. The routers that connect different subnets are called boundary nodes. It is important to note an existence of a fault in on-chip network makes its topology irregular. For instance once a failure occurs in one of the routers in subnet3 its topology is no longer a regular 2D mesh. Routing unit inside an on-chip network is in charge of leading an incoming packet to an

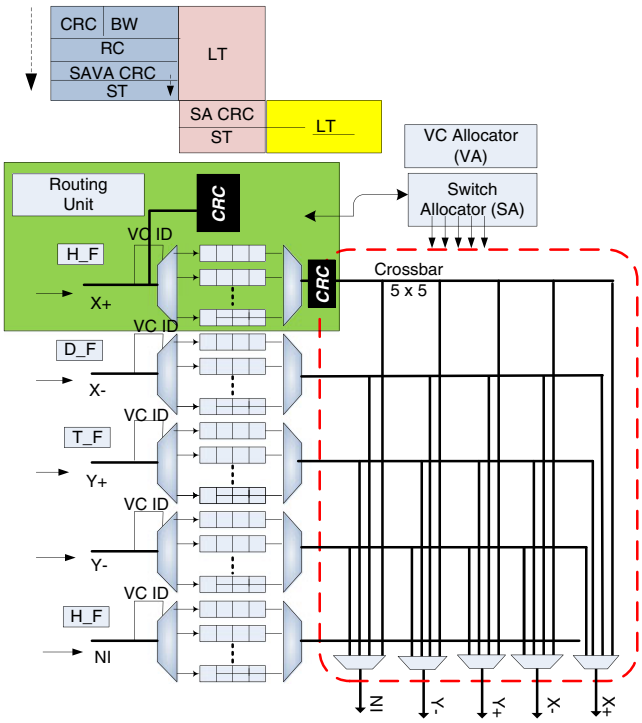


Fig. 1 A generic NoC router architecture

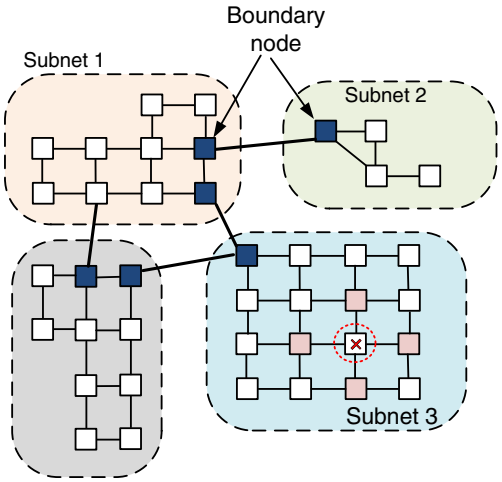


Fig. 2 A hierarchical NoCs

appropriate output port; it performs its task by means of a destination address inside an H-F as well as a routing algorithm.

A NoC flow control mechanism regulates packet propagation across an on-chip network by means of monitoring resource (buffer) allocations and releases. In the context of NoCs, most flow control mechanisms allocate resources at the granularity of a flit. A flit can be transferred through a physical channel in a single step or cycle.

Fault tolerance and reliability are two significant challenges IC designers are facing with. As chip design has become extremely cost-sensitive, the fault tolerance must be provided at a reasonable cost. Reliability-Aware Virtual Channel (RAVC) router as a NoCs fault tolerant architectures is introduced in [13]. This router addresses fault tolerance inside regular topologies like mesh by means of 1) isolating a faulty router from the healthy network 2) dynamic reconfiguring the adjacent routers of a faulty router through which the VCs associated with faulty routers will be reused and reassigned to other the ports. For instance, once a permanent failure occurs in a router inside the subnet3, the adjacent routers of that particular router will be reconfigured such that more resources (VCs) is assigned to the ports which are not connected to faulty routers.

The authors in [15] show that RAVC supplies better performance in the case of failures inside the regular network. However, RAVC is dedicated to the regular mesh topology and its routing unit is not scalable. Moreover, the flow control adopted in RAVC is unable to mitigate the transient failures effects.

In this work, we propose a hierarchical NoC router microarchitecture which can be used inside a hierarchical topology. Plus, a new fault-tolerant flow control mechanism will be

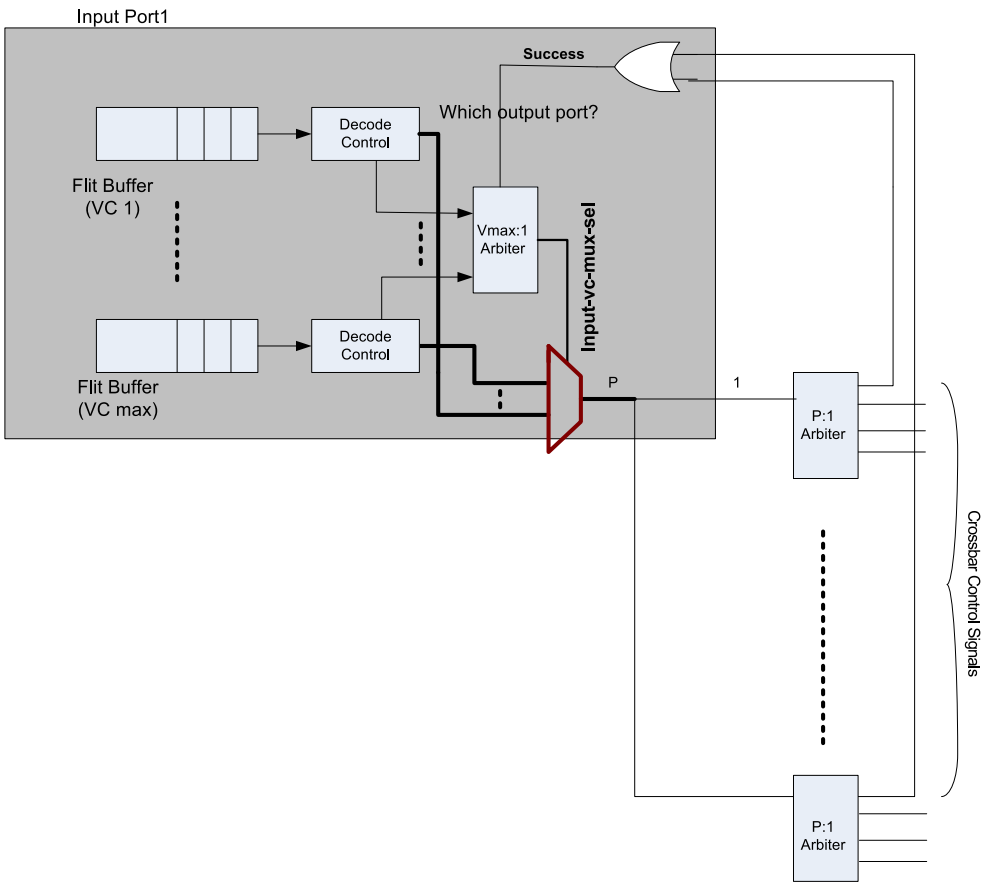


Fig. 3 Our modified Switch Allocation Unit

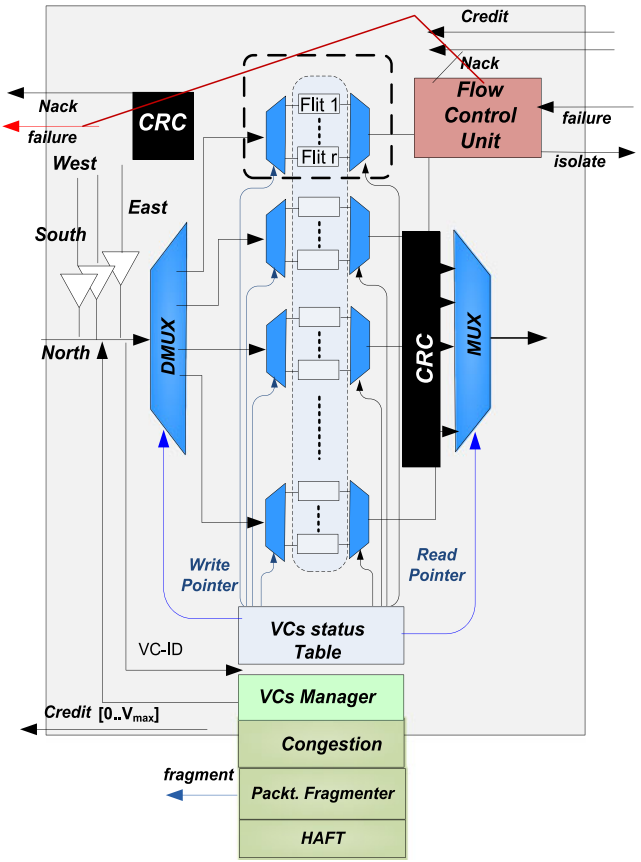


Fig. 4 The proposed input channel

proposed based upon packet fragmentation and resubmission will be proposed.

3 Generic Vitual Channel NoC Router

Figure 1 shows the architecture of a generic VC-based NoC routers. In a generic router each physical channel has a finite number of VCs; that is, there is not a mechanism for dynamic VC allocation. Once a flit arrives from an upstream router bandwidth allocation and CRC checks happen in parallel on that flit. During bandwidth allocation (BW stage) the incoming flits will be stored in the particular VC associated with it VC identifier (VC-ID). This value is defined by the upstream router. When the new incoming flit stored inside appropriate VC, the state of that VC is change to *Routing* sate. Routing Computation (RC) is carried out based upon the routing algorithm; usually the employed routing algorithms inside a regular mesh are simple and there is no need to any routing table information.

The Switch Allocation followed by the VC Allocation is completed in the same stage. A SA winner is selected through two separate stages of arbitration (local and global arbitration). The VC Allocation is then accomplished simply by finding a free output VC from the requested output port of the SA winner. After VC and Switch Allocation stage the flit is updated with new VC-ID and route field; therefore, the new CRC update for the modified field is needed. The CRC check starts in parallel with SA and VC allocator unit. Its results will be attached to the flits leaving the input channel.

4 Proposed Router Architecture

Here, we propose a fault tolerant hierarchical NoC router Microarchitecture. Different subnets inside a hierarchical network can be either aware or unaware of the fact that will be used in a hierarchical network. Here we assume that subnets are topology agnostic. However, there is a negligible difference between the operation boundary nodes any regular nodes in terms of VC allocation and release. Our proposed input channel is shown in Figs. 3 and 4. As shown in these figures input port of each input channel can be connected to the other input ports by means of a tri-state buffer. To supply maximum utilizations of buffers such a feature will be used during the reconfiguration. The reconfiguration will be launched as a consequence of router failures.

The congestion manager unit controls the interconnection of an input port to an input channel. Unbalanced load across VCs and the Head of Line (HOL) blocking resulting from the static VC allocation and hamper on-chip network performance [8, 9, 18].

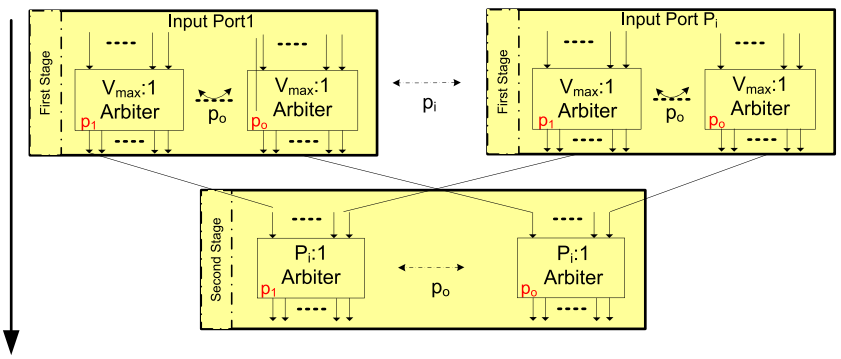


Fig. 5 Proposed router VC allocation unit

Basically, HOL is a situation that a block packet in a VC impedes t)he progress of other packets located in the same VC.

As Fig. 4 illustrates to manage dynamic VC allocation, we have adopted Unified Buffer Structure (UBS). Inside a UBS flits can be stored in slots not necessary consecutive; in other words, the requirements of storing all the flits of a packet in a consecutive space are removed. However, this enhancement comes with the price of having extra columns inside VC Status Table.

A. VC Status Table

VC status Table contains VC number range from 0 to VC_{max}, Output Port (OP), Output Virtual Channel (OVC), Read Point (RP), Write Pointer (WP), Header Pointer (HP), Credit and ESUB. The value that we assigned to the VC_{max} depends on the particular application which is supposed to map to the subnet, topology inside the subnet, and type of the nodes. For instance, in order to keep boundary nodes in the safe condition we usually let them have higher VC_{max} than other nodes. The ESUB is a new column added to the VC status table. This column indicates whether the output port of that particular VC is leaving the local subnet.

B. VC Allocation Unit

As a consequence of having a dynamic range of VCs ($v_{min} < VCs < v_{max}$) per input port, our proposed scheme needs larger vmax: 1 arbiters in comparison to v: 1 of conventional arbiter. However, the proposed VC allocator uses smaller arbiters in the second stage. As shown in Fig. 5, as opposed to the baseline routers which accept requests for each output channel, the second arbitration stage in RAVC is responsible for choosing a winner for each output port among all the competing input ports. In fact, instead of having ($P_{out} \times V$) arbiters, where each of which has to accept ($P \times V$) request, VC allocator at the second stage in RAVC router contains P_{out} arbiters.

C. Switch Allocation Unit

During the switching stage, the crossbar output port will be selected among the requesting input ports. The Switch Allocation unit (SA) arbitrates amongst all VCs requesting access to the crossbar and grants permission to the winning flits. The winners are then able to traverse the crossbar through an appropriate output link. Our modified Switch Allocation (SA) unit is similar to the baseline SA unit and carries out its operation in two stages, 1) the first stage selects a winner request among $\cup_{i=1}^{i=max} VC_i$, 2) the second stage arbitrates among each input channel winners which requests the same output port.

The conventional router SA unit is modified to be consistent with our dynamic VC allocation approach. To adopt the worst-case scenario, where an input channel dispenses all allowable VCs (VC_{max}), the proposed router employs a VCmax: 1 arbiter per each input channel at the first stage. The second arbitration stage is similar to that of baseline router.

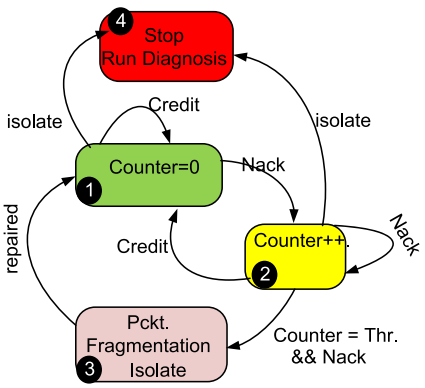


Fig. 6 State diagram of the flow control unit

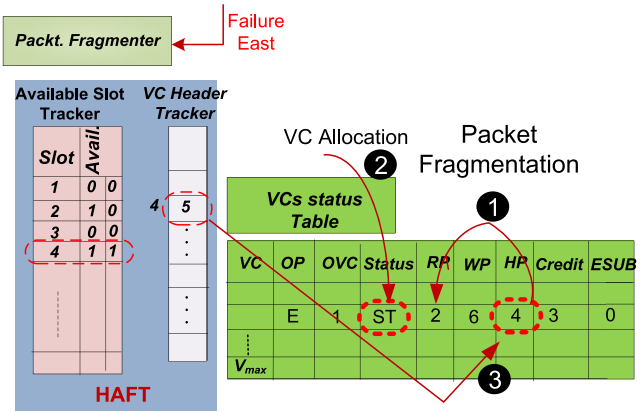


Fig. 7 VC status table and packet fragmentation

D. History-Aware Free slot Tracker (HAFT)

This unit is shown in Figs. 6 and 7. HAFT keeps track of the available slot inside the UBS. Once a new coming header-flit passes the CRC check stage, it needs to be stored into one of the UBS slots. In our mechanism, VC manager along with the history-aware free slot tracker storing available slot status carries out the dynamic VC allocation.

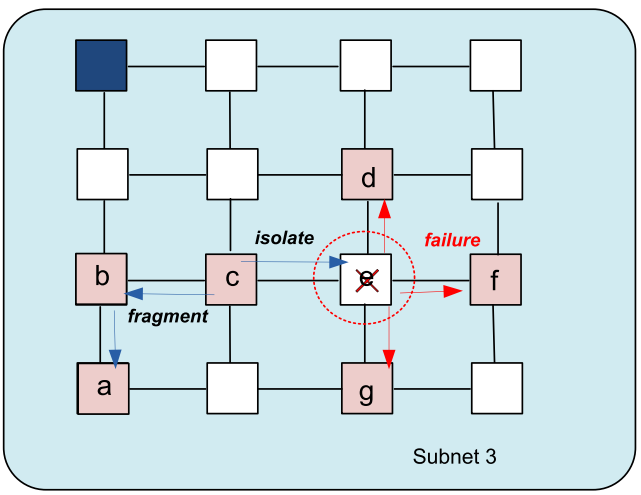


Fig. 8 Flow control mechanism

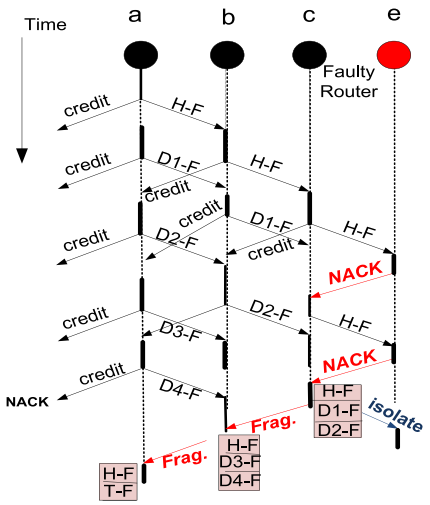


Fig. 9 Sequence diagram of the proposed fault tolerant flow control

It turns out that the VC Manager either dispenses a new VC or will place that new incoming flit at the end of already existing VC. As far as VC's counts not exceeding the VC_{max}, the VC manager dispenses a new VC. Otherwise, a new available VC should host incoming flits.

To provide a fault-tolerant flow control, it is required to keep track of all header-flits, as a result, HAFT needs to update its VC-Header tracker table. For example, if the header of the VC1 is placed at address adr1 and a new incoming header-flit located at adr2 placed inside the same VC, the content VC-Header tracker at the address adr2 is adr1. Since available slots in UBS are tracked by HAFT, once it reaches its capacity limits, it activates the Congestion Manger unit by means of triggering congestion signals. Once a buffer releases HAFT deactivates its trigger.

When there is a need for buffers in other input channels, HAFT will inform the congestion manager of that input channel which is already disconnected from the faulty router to store new incoming packets. When flits other than header flits leave an input channel, HAFT releases the corresponding buffer slot. HAFT store a 2-bits data with respect to each buffer slot as shown in Fig. 7. Such bits are '11' concerning to header-flit, '10' concerning to data-flit, '00' for empty slot.

E. Packet Fragmentor Unit

The VC status Table inside our proposed router stores the header-address as well as ESUB information, making it different from [9]. Such data stored inside HAFT is instrumental during packet fragmentation. As Fig. 7 illustrates, packet fragmentation happens in 3 steps. At the first step the read-point of a VC is replaced by header-address of that particular VC. Thereafter, at the second step, the VC status should be changed to (VC-allocation), making it going through the routing step again. Finally, the header-point field will be replaced by previous header flit address.

F. Flow ControlUnit

Figure 6 shows state the proposed flow control unit diagram. There exist a counter associated with each output port; the flow control unit uses these counters to perform its task. First, these counters are initialized with zeros. We made use of a Cyclic Redundancy Check (CRC) unit to discover Link errors and Data path upset. This CRC check occurs in parallel with the BW stage and it has no impact on the router critical path. The CRC calculation takes place during the link traversal stage, because the content of the flit (output port and output VC) is determined at this stage. The appropriate CRC can be attached to a flit while being transferred. Because the CRC calculation must be carried out in one cycle of the router pipeline, a simple yet effective polynomial ($x^8 + 1$) is used;

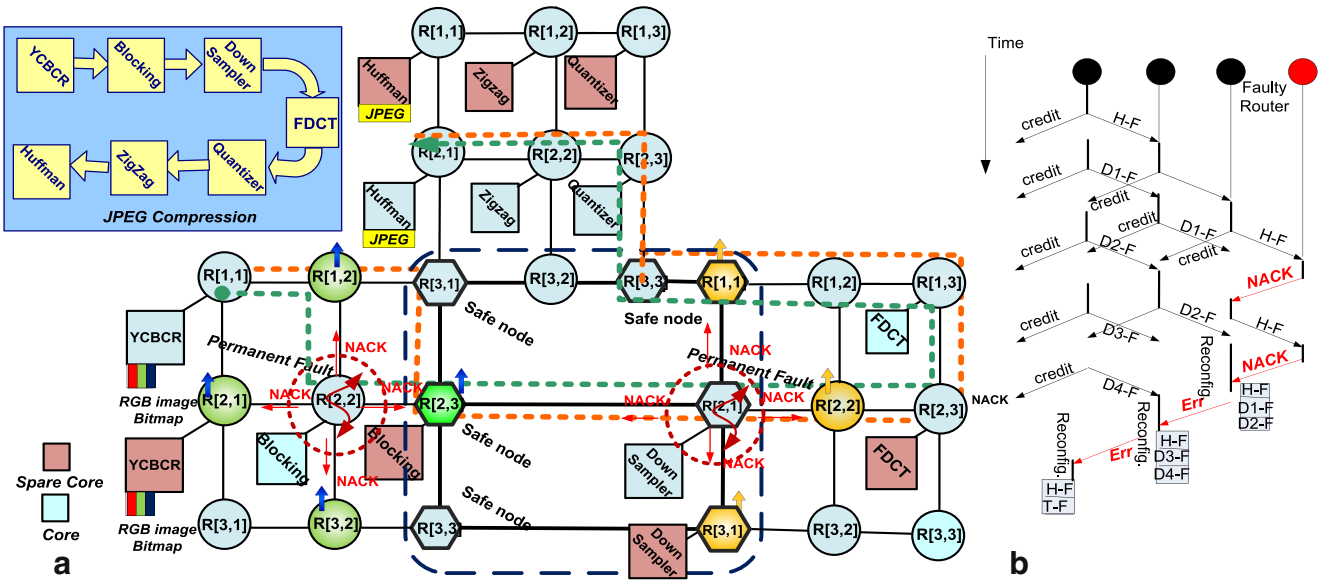


Fig. 10 **a** Experimental platform, (**b**) Proposed fault-tolerant flow control

this CRC unit can detect odd number of bit-errors; plus, a single burst error up to eight bits is diagnosable.

During a flit transmission, if either the downstream router or the link between upstream and downstream is faulty the upstream router will receive NACK signal. Once a flow control unit receives the ***"NACK"*** signal it will change its state to (state 2), shown in Fig. 6. As long as the flow control unit is in this state it will resubmit the same flit over and over again; that is the mechanism that we have used to address transient errors (soft-errors). Meanwhile by receiving every NACK signal the value of that counter is increased by one.

Once that counter reaches its threshold value and again NACK signal is received from the downstream router, flow control unit changes e to (state 3); at this moment, first, it sends an isolate signal to the downstream router and invokes packet fragmentation operation.

The downstream router which has exactly the same flow control unit by receiving isolate signal should first stop further transmitting of flits; second if it has a dedicated module for error diagnosis should activate that unit. Such an invocation may lead to propagation of fragment signals from the current router to other routers involved in the submission of the current packet. For instance assuming that router (a) in Fig. 8 is transmitting a packet (P1) that involves five flits. At the first step router (a) send a header flit H-F to router (b); afterward H-F is moved to router (C) and D1-F is moved from router (a) to router (b). However, it turns out that router (C) is unsuccessful to process D1-F after trying several time. Therefore, a fragment signal will be propagated from router c, b and a respectively where these routers still have part of the packet.

Figure 9 illustrates the sequence diagram of the proposed flow control. Associated with ach VC there is a credit value. This value indicates the maximum number of flits inside a VC. The proposed flow control have utilized credit based flow control and augmented that with NACK, 'isolated' and 'fragment' signals. In the credit-based flow control, once a router forwards a flit, it sends a credit to the downstream router. The same scenario that we explained before has been depicted in Fig. 9 by means of sequence diagram. Eventually, the packet (P1) which is already distributed among 3 routers needs to get fragmented.

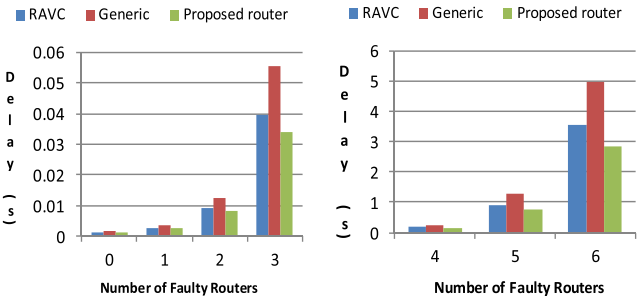


Fig. 12 Average latency

5 Deadlock- Free Routing

A deadlock-free routing algorithm for hierarchical Network on Chip is proposed in [7]. The authors in [7] proved that a hierarchical routing algorithm R_h is deadlock free if the following conditions are maintained: 1) each subnet routing RL (Local Routing) is dead-lock free, 2) the external routing (the routing among boundary nodes) RG is deadlock free, 3) all boundary nodes are safe. Moreover, they have also stated that boundary nodes are safe as long as there is no path of link of dependencies from an output link to an input link of such node.

Since our proposed router can dispense VCs dynamically, it can break any possible cycle of link dependencies that might cause deadlock. To manage VC allocation whenever a boundary node receives a 'fragment' signal as a result of a failure inside that subnet, it will dispense another VC. That is, once it receives a new header flit that that will be stored inside the new VC.

We have adopted Logic-Based Distributed Routing (LBDR) proposed in [20]. According to this routing algorithm each cardinal port only needs three bits (two bits for routing restriction and one connection bit). The values of such bits are determined by topology and routing restriction sets. Routing restriction bit (referred to as R_{xy}) indicates whether packets routed through some ports could make a turn at next hop. The connection bit $C_x(C_n, C_e, C_s, C_w)$ at each output port indicates whether a node is connected through this direction. As stated in [20] this routing algorithm provides a deadlock-free routing in the case of failures inside a regular network topology.

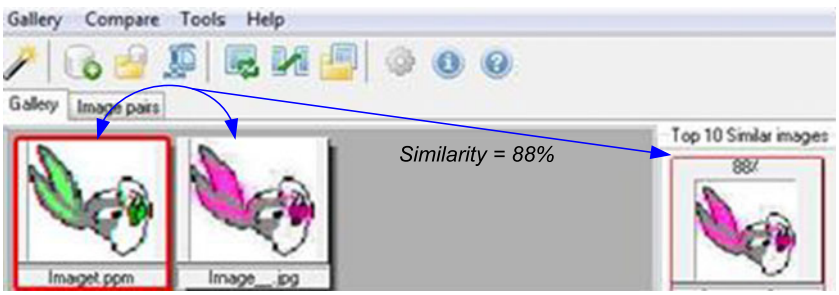


Fig. 11 Image comparison

Table 1 Average energy consumption over different failure scenario

#Perm. Failures	Power (W)		
	Generic	RAVC	Proposed Router
0	1.75	1.82	1.80
1	1.91	1.85	1.81
2	2.20	2.01	1.90
3	3.11	3.01	2.15
4	4.11	3.95	2.50
5	5.15	4.78	3.14

6 Experimental Results

A. Application Specific Traffic

To evaluate reliability and the performance metrics of the proposed router in a fault prone environment i.e. assuming both permanent and transient fault, we consider a $(3 \times 3) \times 3$ hierarchical topology. Figure 10(a) illustrates our experimental framework. The routers that connect subnets are called boundary nodes. We created a cycle accurate SystemC model of the router and augmented it with Orion power library to evaluate energy consumption of the network. We considered 70 nm as the process feature size and 250 MHz as working frequency. The flit size is 64 bits and every packet involves 8 flits. As shown in Fig. 10, a standard JPEG encoder SystemC model is mapped to the hierarchical interconnection. Our JPEG model consists of seven modules: YCBCR, Blocker, Down-Sampling, Digital Cosine Transform (DCT), Quantizer, ZigZag, and Huffman Coder.

In our framework, we assumed two instances for every module (spare and original). In the case of permanent failures in a router, as explained previously, first, that router will be isolated to transmit and receive more packets; second, its neighboring routers will be notified and reconfigured to allocate more buffer for their input channels; then, a spare core corresponded to the core connected to the faulty router take over the job of original core.

For instance, as Fig. 10 illustrates once the proposed fault-tolerant flow control detects failures inside R[2,2], first, its neighboring routers R[1,2], R[2,1], R[3,2] and R[2,3] will be reconfigured. Thereafter, fault tolerant routing algorithm will change the previous path (green line) to the new one (orange line), activating spare "Blocking". The remaining issue here is R[2,3], which is a boundary nodes.

As per our experiment, we consider a 64*64 bitmap image; this image is located inside the memory of YCBCR; eventually, final result will be ready by Huffman module. The fault injection module which is also written in SystemC injects both permanents and transient fault inside the routers and links. As long as the simulation is running, faults can be injected; however, no more than six permanent faults are allowed. Plus, it is not allowed to inject permanent fault to two routers that are connected to the same module, or else simulation platform no longer can generate the final JPEG image. To compare accurately reliability of the proposed router with Baseline router and RAVC router [13], we exercise these routers with the same failures. We have utilized "Image Comparer 3.7 Build 710" to assign a quantitative number to the reliability of these routers. This software can supply a quantitative measure regarding the similarities of various images. In other words, this software gives a number that represents the percentage of similarities between two images. For example, the similarity (difference) between two images illustrated in Fig. 9 is 88 % (12 %) based on this software

$$Fidelity \cong \frac{\sum_{j=1}^{\#Experiments} Similarity(original, Generated_i)}{\#Experiments} \quad (1)$$

As our first experiment, we measure the fidelity of image reproduction, as a means to estimate the reliability of an on-Chip network. We compute the fidelity, Eq. 1, by summing up the similarities of all the generated output images with faults against the correct image. Similarities were obtained by Image Comparer 3.7, and the total was averaged by dividing it over

Table 2 Features provided by our proposed router versus other related work

	Boundary node Safeness	Flow Control	Topology Agnostic Reconfiguration	Fault Mitigation		Hierarchical topology
				Permanent	Transient	
Our proposal in this paper	VC Expansion	Fragmentation/Wormhole	Yes	Yes	Yes	Yes
RAVC [13]	Relay on Subnet local routing	Wormhole	No	No	Yes	No
ERAVC [15]	Relay on Subnet local routing	Fragmentation/Wormhole	No	Yes	Yes	No
HiRA [7]	Relay on Subnet local routing	Wormhole	No	No	No	Yes
[2]	Static	Wormhole	No	No	No	Yes
[12]	N/A	Wormhole	No	Yes	No	No

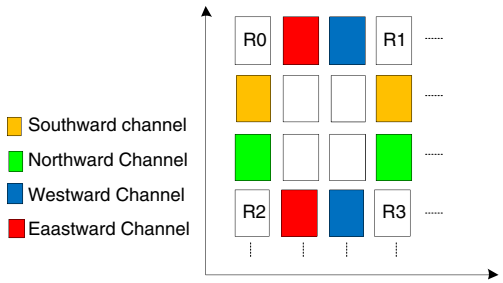


Fig. 13 Simulation graph

the number of experiments. This software can supply a quantitative measurement regarding the similarities of various images. In other words, this software gives a number that represents the percentage of similarities between two images. For example, the similarity (difference) between two images illustrated in Fig. 11 is 88 % (12 %) based on this software. We did our experiment over 1200 cases. The experiment shows that the proposed router provides 15 % and 34 % more fidelity than RAVC and Baseline router, respectively.

We evaluate the average latency of the network plugged with our proposed router, generic (baseline) router, and RAVC. Experimental in Fig. 12 shows that on average the proposed router offers 22 % and 45 % improvement on average latency with respect to RAVC and the baseline router in the case of failures in network. Tables 1 and 2 display the average energy consumption of the generic router, RAVC, and our proposed router when there are failures in system. When there are no failures in on-chip network generic router consume less energy; however, while number of permanent failure increases the proposed router provides better characteristics in terms of energy consumption.

B. Synthetic Traffic

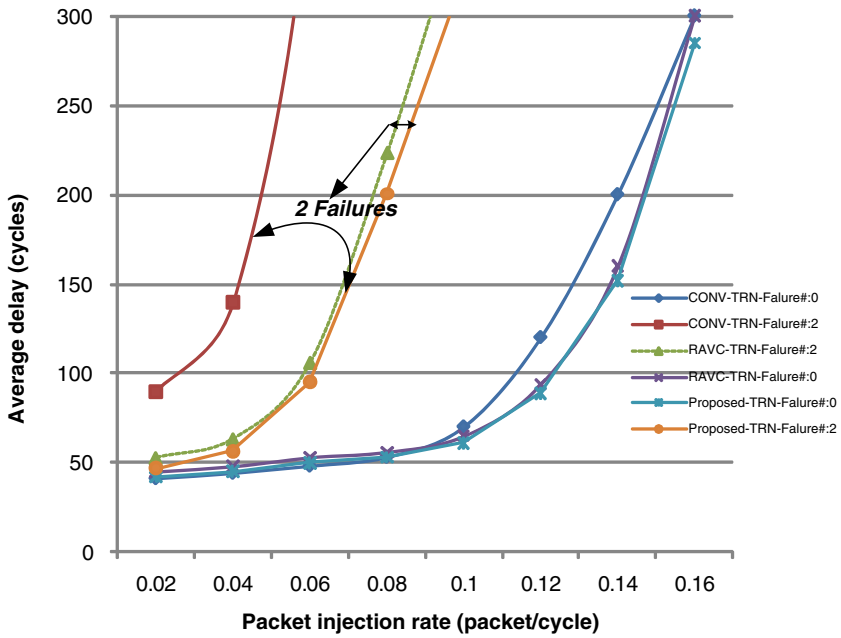
To evaluate performance of the proposed router in a fault-prone environment using synthetic traffic in hierarchical NoC, we consider the odd-even routing algorithm for the global routing. For each subnet we consider LBDR routing algorithm. As explained earlier LBDR routing is used as a fault tolerant deadlock-free routing algorithm. All simulations were performed in 27 nodes (3× 3) ×3) hierarchical network which is shown in Fig. 8 (A). Moreover, to compare our proposed router with the baseline and RAVC [14] router, these routers have been configured to see the network as a "flat network". We assume uniform and transpose traffic patterns, in such a way that traffic pattern inside a subnets are either uniform or transpose. But we assume just uniform traffic pattern between boundary nodes.

Figure 13 represents the performance metrics graphs. R0 – R4 illustrates the placement of tiles/routers. Red bar between R0 and R1 represents metric for east channel from R0 to R1. Blue bar between R0 and R1 represents metric for west channel from R1 to R0. Green bar between R0 and R4 represents metric for northward channel from R0 to R4. Orange bar between R0 and R4 represents metric for southward channel from R4 to R0 [11].

In our experiments, baseline router contains 4 VCs each of which encompasses 16 flits. Our proposed router contains 64 flits buffer with the default of minimum 4 VCs yet extendable to 16 (VC_{min}=4, VC_{max}=16).

Under uniform and transpose traffic pattern inside subnets, the average packet latency of the conventional router, RAVC and our proposed router, assuming router failures, are illustrated in Figs. 14 and 15, respectively. Here, we assume packets size equals to 8 flits. On the basis of extracted simulation environment, in a fault prone environment under transpose traffic

Fig. 14 Average latency transpose local traffic



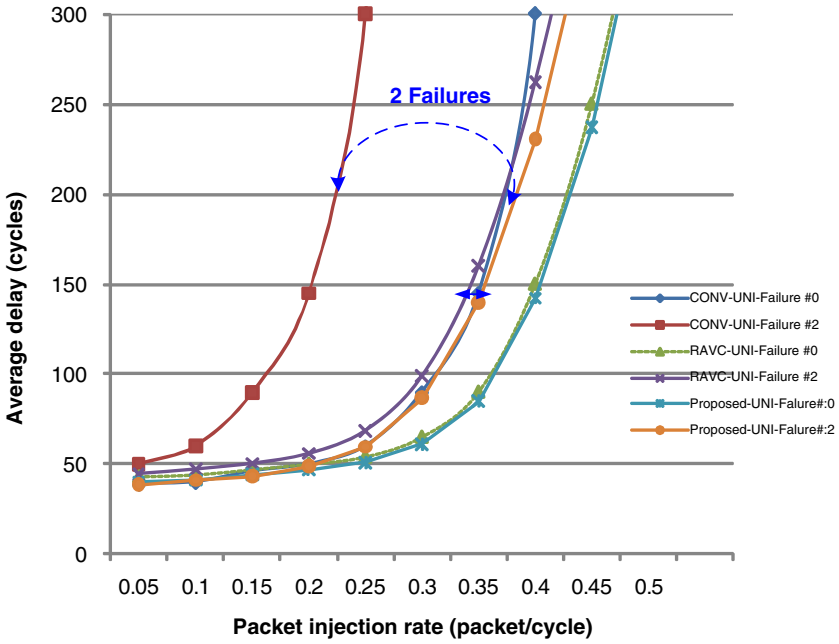


Fig. 15 Average latency uniform local traffic

pattern, the proposed router provides 10 % and 38 % decrease in the average packet latency over RAVC and baseline router. As per Fig. 15, the reduction in the average packet latency over RAVC and baseline router under uniform traffic pattern is 12.5 % and 28 %, respectively. To illustrate the effects of failures on traffic distribution, we plot the simulation results of one of the three subnets, Figs. 16, 17, 18 and 19.

As explained before, NIRGAM measures performance of NoC per-channel basis http://www.nirgam.ecs.soton.ac.uk. We armed NIRGAM with LBDR and changed VC allocation unit as well as the flow control algorithm to realize our proposed router. NIRGAM is configured to generate the transpose traffic. Figure 16 illustrates the average latency based on the number of clock cycles per flit in each channel. Figure 17 shows the power consumption inside each router. Figures 14 and 15 show non-faulty conditions, whereas Figs. 16 and 17 plot the average latency and power consumption inside a generic router in case of failures. Average flit latency and power consumption inside the proposed router in case of failures are shown in Figs. 18 and 19, respectively. As shown in these figures, the proposed router handles failures better than generic router and avoids the creation of hotspots.

On average, considering the average latency of channels altogether, the proposed router provides 15 % and 32 % decrease in the average packet latency over RAVC and Baseline router, under the transpose traffic pattern. Under uniform design traffic, in terms of energy consumption, in the presence of failures, the proposed router provides 25 % and 38 % better results over RAVC and Baseline router.

The design is implemented in Verilog and synthesized using Synopsys design Compiler tool and the TSMC 65 nm technology library at supply voltage 1 V and an operating frequency of

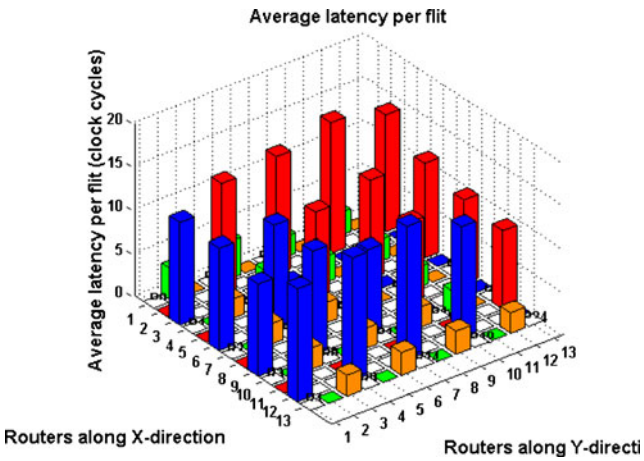


Fig. 16 Average latency per flit transpose pattern non-faulty

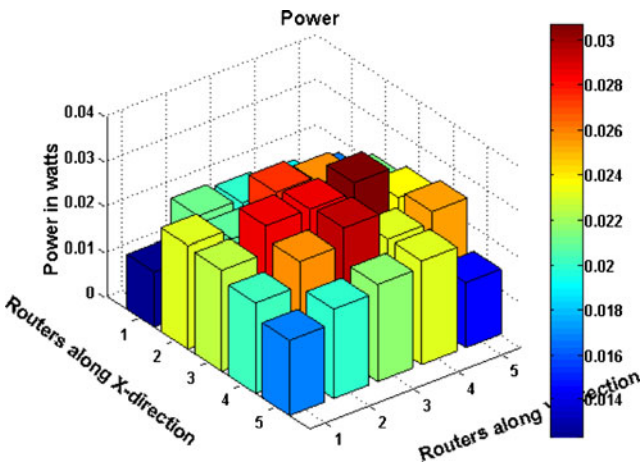


Fig. 17 Energy consumption non-faulty

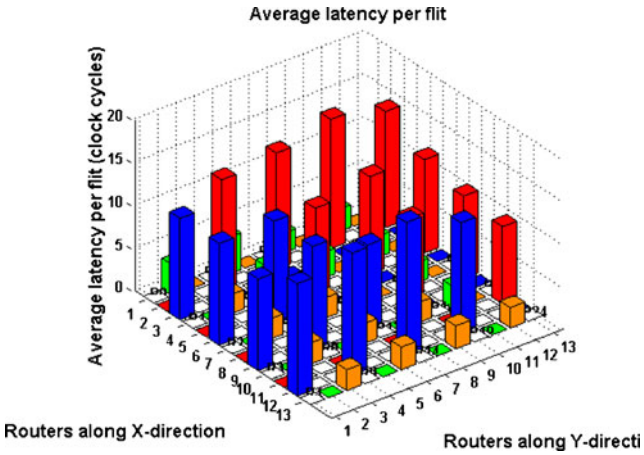


Fig. 18 Average latency per flit in Baseline router in case of failures (faulty)

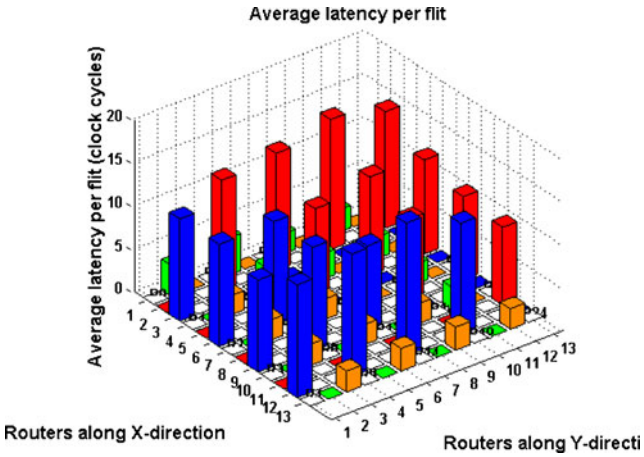


Fig. 20 Average latency per flit in the proposed router in case of failures

500 MHz. The area of the proposed router is 101,544.49 μm2 which has 2.3 % overhead with respect to RAVC router.

We have adopted packet completion probability metric introduced in [10] as another means to compare the proposed router versus baseline router, (Eq. 2). This is defined as the number of received packets divided by total number of injected packets into the on-chip network.

$$\textit{Packet Completion Probability} = \frac{\textit{Received Packets}}{\textit{Total number of injected packets}}, \quad (2)$$

where Packet completion probability = 1 in a fault free network

Figures 20, 21 and 22 compares the proposed router with RAVC and the baseline router. The proposed router provides approximately 14 % improvements with respect to RAVC and 51 % improvement with respect to the baseline router.

C. Comparison with the related work

Table 1 feature-wise compares this work with some of the closest related work [2, 7, 12, 13, 15]. These features are related to the employed flow-control mechanism, the VC allocation scheme and the support for transient and permanent failures. As listed in this table, as opposed to Partial VC Sharing (PVS) mechanism, proposed in [12] and RAVC [13] which overlook the effect of transient errors, the proposed router in this paper, makes an effective use of the packet-fragmentation, explained in Section 4, to alleviate the destructive effects of transient errors. We also leverage the fact that if there is a permanent error inside a particular link or router, the neighboring routers can benefit from inter-channel buffer sharing, in a sense that the buffers expected to host the traffic from faulty links or routers no longer need to maintain its connection, enabling resource reuses in favor of handling extra traffic. However, other routers [7] overlook the possibility of any performance gain by means of inter-channel buffer sharing. As explained in Section 4, the proposed router maintains network connectivity in a

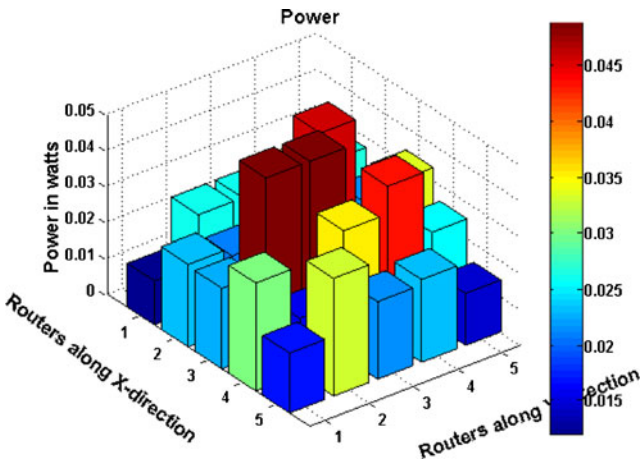


Fig. 19 Energy consumption in Baseline router in case of failures (faulty)

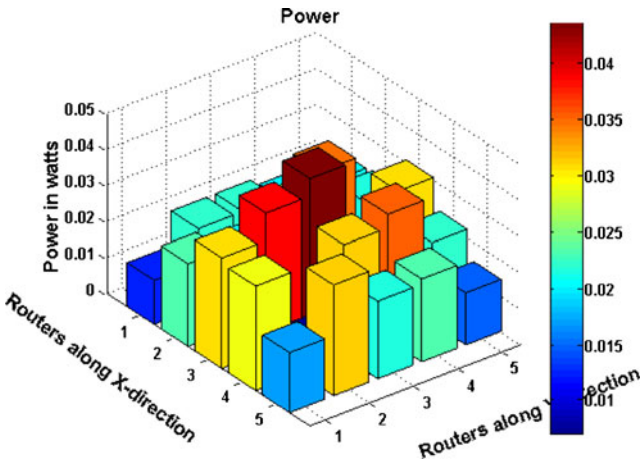


Fig. 21 Energy consumption in the proposed router in case of failures

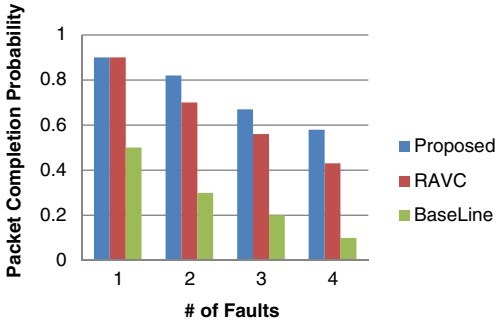


Fig. 22 Packet Completion Probability with different number of faults

hierarchical topology by means of VC expansion in boundary nodes upon the reception of NACK signals, whereas [7, 13] and [15], and relay on the adopted routing algorithm to keep the boundary node in the safe condition.

As shown in [18], if a subnet fault-tolerant routing algorithm involves the topology discovering and the routing table updates phases, it is impossible to always keep the boundary nodes in the safe condition by just relying on the routing algorithm.

7 Conclusion and Future work

A new fault-tolerant routing suited for hierarchical topology is proposed in this paper. The proposed router adopted a new VC allocation unit which enables Dynamic VC allocation. A new fault-tolerant flow control unit called "fragmentation-flow" control is introduced. This flow control mitigates both permanent and transient faults. Resorting to the fragmentation-flow control, the proposed router provided packet resubmissions without the need of extra buffers.

On average, the proposed router reduces the packet latency by 20 % in a fault-prone environment. Moreover, the energy consumption reduces when there is possibility of faults in an on-chip network. In the future work, we are planning to incorporate a dynamic fault-tolerant routing algorithm which involves topology discovery and update in routing per each update. In order to keep the whole hierarchical network in a deadlock-free condition, the VC expansion must be extended with a new mechanism that guarantees even with the new Channel Dependency Graph (CDG) all boundary nodes are in the safe condition [18]. The incorporation of the proposed router as means as a means to enhance on-chip observability is also another interesting future work [11, 16].

References

1. Alaghi A, Karimi N, Sedghi M, Navabi Z (2007) "Online NoC Switch Fault Detection and Diagnosis Using a High Level Fault Mode," *in Proceedings of International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems* (DFT), pp. 21–29
2. Bourduas S and Zilic Z (2007) "A Hybrid Ring/Mesh Interconnect for Network-on-Chip Using Hierarchical Rings for Global Routing," in *Proceedings of Network on Chip Symposium* (NOCS), pp.195–204
3. Dally WJ and Towles B (2001) "Route packets not wires: On-chip interconnection networks," in *Proceedings of Design Automation Conference* (DAC), pp. 684–689
4. Das R, Eachempati S, Mishra AK, Narayanan V, Das CR (2009) "Design and Evaluation of a Hierarchical On-Chip Interconnect for Next-Generation CMPs," *in Proceedings of International Symposium on High Performance Computer Architecture (HPCA),* pp. 175–186
5. Duato J, Yalamanchili S, Nionel L (2003) "Interconnection networks: An engineering approach," (Morgan Kaufmann Publishers)
6. Guree A, Ventroux N, David R, Merigot A (2010) "Hierarchical Network-on-Chip for Embedded Many-Core Architectures," in *Proceedings* of Network on Chip Symposium (NOCS), pp. 189–196
7. Holsmark R, Kumar S, Palesi M, Mejia A (2009) "HiRA: A Methodology for Deadlock Free Routing in Hierarchical Networks on Chip," in *Proceedings* of Network on Chip Symposium (NOCS), pp. 2–11
8. Kang YH, Kwon T, and Draper J (2009) "Dynamic Packet Fragmentation for Increased Virtual Channel Utilization in On-Chip Routers," in *Proceedings of International Symposium on Networks-on-Chip* (NOCS), pp. 250–255
9. Karanth Kodi A, Sarathy A, and Louri A (2008) "iDEAL: Inter-Router Dual-function Energy and Area-efficient Links for Network-on-Chip (NoC) Architectures," *in Proceedings of International Symposium on Computer Architecture* (ISCA), pp. 241–250
10. Kim J, Nicopoulos C, Park D, Vijaykrishnan N, Yousif MS, and Das CR (2006) "A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks," in Proceedings of International Symposium on Computer Architecture (ISCA)
11. Kumar A, Kundu P, Singh AP, Peh L and Jha N (2007) "4.6Tbits/s 3.6GHz Single-cycle NoC Router with a Novel Switch Allocator in 65nm CMOS," in *Proceedings of International Conference on Computer Design* (ICCD), pp. 63–70
12. Latif K, Rahmani AM, Nigussie E, Seceleanu T, Radetzki M, Tenhunen H (2013) Partial virtual channel sharing: a generic methodology to enhance resource management and fault tolerance in networks-on-chip. J Electron Test: Theory Appl (Springer-JETTA). doi:10.1007/s10836-013-5389-5
13. Neishaburi MH and Zilic Z (2009) "Reliability aware NoC router architecture using input channel buffer sharing," in *Proceedings* of Great Lake Symposium on VLSI (GLSVLSI), pp. 511–516
14. Neishaburi MH and Zilic Z (2011) "A Fault Tolerant Hierarchical Network on Chip Router Architecture," in *Proceedings of International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems* (DFT), pp. 445–453
15. Neishaburi MH and Zilic Z (2011) "ERAVC: Enhanced Reliability Aware NoC Router," in *Proceedings of International Symposium on Quality Electronic Design* (ISQED), pp.591–596
16. Neishaburi MH and Zilic Z (2011) "A distributed AXI-based platform for post-silicon validation," in *Proceedings of IEEE VLSI Test Symposium* (VTS), pp. 8 – 13
17. Neishaburi MH and Zilic Z (2011) "Hierarchical trigger generation for post-silicon debugging," in *Proceedings of IEEE VLSI Design, Automation and Test* (VLSI-DAT), pp. 1 – 4
18. Neishaburi MH and Zilic Z (2013) "NISHA: A fault-tolerant NoC router enabling deadlock-free interconnection of subsets in hierarchical architecture". J Syst Archit (JSA)
19. Park D, Nicopoulos C, Kim J, Vijaykrishnan N, Das CT (2006) "Exploring Fault-Tolerant Network-on-Chip Architectures," in *Proceedings of International Conference on Dependable Systems and Networks* (DNS), pp. 93–104
20. Rodrigo S, Flich J, Roca A, Medardoni S, Bertozzi D, Camacho J, Silla F, Dauto J (2011) Cost-efficient on-chip routing implementations for CMP and MPSoC systems. IEEE Trans Comput-Aided Des Integr Circ Syst 30:534–547

M. H Neishaburi received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Tehran, Tehran, Iran, in 2005 and 2007, respectively. He is currently a Ph.D. candidate with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada.

He has joined Synopsys in Canada as a Senior Verification Engineer in 2013. Now he has been working on verification of super speed serial interfaces such as USB3, PCIe Gen1-3, SATA Gen 1-3 and 3 and XAUI.

Mr. Neishaburi is the recipient of the prestigious Lorner Trottier and Provost's Graduate named fellowship as a Ph.D. student with the Faculty of Engineering, McGill University. His current research interests include ASIC Design, Verification and Network on Chip.

Zeljko Zilic received the B.Eng. degree from the University of Zagreb, Zagreb, Croatia, and the M.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, all in electrical and computer engineering. From 1996 to 1997, he was with Lucent Microelectronics, where as a Member of Technical Staff he was involved in the design, test, and verification of Orca FPGAs. He joined McGill University, Montreal, QC, Canada, in 1998, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He used a sabbatical leave from 2004 to 2005 to work with ST Microelectronics, Ottawa, ON. He conducts research on various aspects of the design and test of microsystems including programmable logic cores. He has published over 200 papers, 3 books, and 5 patents for which he received several awards. He has graduated over 40 M.Eng. and Ph.D. students, who have received numerous awards for their theses and have moved on to leading industrial an d academic institutions upon their graduation.