# ERAVC: Enhanced Reliability Aware NoC Router

M. H Neishaburi, Zeljko Zilic
McGill University, 3480 University Street, Montreal, Quebec Canada H3A-2A7
Mh.neishabouri@mail.mcgill.ca, zeljko.zilic@mcgill.ca

## Abstract

The continuing advances in processing technology result in significant decreases in the feature size of integrated circuits. This shrinking leads to increases in susceptibility to transient errors and permanent faults. Network on Chips (NoCs) are poised 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 digital systems. In this paper, we propose an enhanced fault tolerant micro-architecture for NoC routers. The proposed router supplies dynamic virtual channel allocation using Unified Buffer Structure (UBS) and History Aware Free-slot Tracker (HAFT). Plus, to reduce the associated performance costs of retransmissions in the case of failure, the proposed router employs a high-performance fault tolerant control flow, handling both transient and permanent faults 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

Reliability, Network on Chip (NoC)

## 1. Introduction

Large number of processing elements (cores and resources) can be integrated into a single chip due to extensive advances in semiconductor process technology. To gain maximum utilization of these resources and cores, they need to get connected through an environment that enables rapid inter-exchange of a large volume of data [6]. 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 Multi-processor System on Chips (MPSoCs) within a reasonable area and power envelope.

Networks on chip (NoCs) have emerged as a promising interconnects fabric to address such growing challenges in deep submicron technology. By supporting better modularity, scalability and higher bandwidth, NoC architectures supply a practical alternative for traditional SoC interconnect approaches [4][6][14].

While NoC architecture promises to supply substantial bandwidth and concurrent communication capabilities, it is subject to significant performance degradation due to the reliability issues arising from the signal integrity, as well as manufacturing and testing challenges in deep submicron technology [12]. 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 [23]. Additionally, as processing technology scales, the prominence of permanent faults resulting from electromigration 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 such faults [23]. Moreover, scaling implies that the circuits will become increasingly sensitive to temporary faults caused by terrestrial cosmic rays and alpha particles, resulting in unacceptable soft-error rates (SERs) and throughput in future on-chip networks[23].

While a packet progresses along a route, a flow control coordinates allocation of resources in network. 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.

We note that buffer management has become a challenge for NoC designers - buffers consume around 46% of power [1] inside on-chip routers. In general, as the power density increases, so does the temperature, accelerating device degradation and reducing reliability and lifetime of the circuit. Alternatively, network throughput and performance significantly suffer with reducing the number of router buffers. Thus, the effective and resourceful management of buffers and hence input and output channels in NoC routers has a crucial effect on performance and reliability of interconnection networks.

In this paper, we propose an Enhanced Reliability Aware Virtual Channel (ERAVC) router, which is architected to handle transient faults and alleviate the effects of permanent faults. Major benefits of the proposed router over its counterparts, specifically RAVC router [3] are: 1) it offers more reliability and better performance in the case of failures in network 2) it offers a new fault-tolerant flow control that facilitates packet resubmission without the need for extra buffers.

## 2. Related work

Reliability of on-chip networks has been addressed in literature from different aspects. To alleviate effects of permanent faults in NoCs, fault tolerant routing has been suggested in [7-11]. In general, such routing algorithms have been categorized as stochastic and adaptive. The authors in [7] provide the stochastic fault tolerant routing algorithm that transfers redundant packets through alternative paths.

In [8] authors suggest gossip routing that enables a router to forward packets to any of its neighbors with some

preordained probability. Direct flooding in [9] improves gossip flooding algorithm by giving priority to routers that bring packet closer to the destination. On the other hand, faulty routers are bypassed by locally or globally adapting the network. In fact, the distinctive feature of an adaptive routing algorithm with respect to a stochastic routing is that adaptive fault-tolerant routings sustain network connectivity in case of failures by exploiting structural redundancy of NoCs and without consuming bandwidth through data redundancy. However, such routing algorithms are subject to deadlocks or livelocks. DyAD [10] and Odd-even [11] are two adaptive routing algorithms that are deadlocks and livelock free. Although an adaptive routing algorithm can tackle permanent fault in an on-chip network, it still suffers from transient faults. In fact, the inability of fault tolerant routing algorithms to detect and avoid transient faults results from the fact that transient errors disappear faster than routing adaptation [5].

On-chip flow control schemes address reliable on-chip transmissions by end-to-end and link level fault recovery between routers. In [20] authors show that link-level recovery provides better solution, since there is no requirement for large retransmission buffers due to timeout period. Particularly, traffic congestion or failures inside the network result in increased timeout latency, and subsequently congestion propagation in network. Authors in [2] present a fault-tolerant flow control mechanism using fragmentation and a dedicated header buffer for each virtual channel. However, their mechanism needs dedicated header buffer for every VC, leading to large energy consumption.

Resorting to reconfigurable structures is another policy for dealing with faulty conditions. Neishaburi et al. in [3] proposed RAVC router which has a novel architecture for dynamic virtual channel allocation. RAVC mitigates the effects of failures in an on-chip network in that faulty routers are isolated and the virtual channels of the routers surrounding the one with permanent errors can be totally recaptured and reassigned to other input ports. However, that study provides no solution to handle transient faults.

## 3. Preliminaries

### 3.1. A generic NoC router

Figure 1 illustrates the architecture of a generic NoC router; this router uses virtual channel flow control and wormhole switching and has two pipeline stages [15]. It consists of five basic elements: Routing Unit, VC allocator, Switch allocator, input channels and a Crossbar. This router has four inputs corresponding to the four cardinal directions (North, East, South and West) and one from the Network Interface Controller (NIC); an NIC converts messages from local Processing Element (PE) to the acceptable format for routers and vice versa. A flit conveys data and control that will fall into Header-Flit (H-F) that keeps the source and the destination addresses, Data-Flits (D-F) that carry data parts of a packet and Tail-Flit (T-F) that represents end of a packet. The Routing unit based on a routing algorithm and the particular destination address inside an H-F leads an incoming packet to the appropriate output port. A routing algorithm determines the path along which a packet is transferred to its destination. A routing algorithm can be either deterministic or adaptive. A deterministic routing algorithms always supplies the same path between the given source/destination pair, while an adaptive routing algorithm employs global NoC characteristics such as traffic and the failure location to avoid congested area or faulty regions of a network. A network topology determines how these routers are connected together and consequently the ports number of each router. An NoC flow control mechanism controls packet propagation across on-chip network by monitoring the resource (buffer) allocation and release. In the context of networks on chip, most flow control mechanisms allocate resources at the granularity of a flit. A flit is a part of a packet that can be transferred through a physical channel in a single step or cycle.

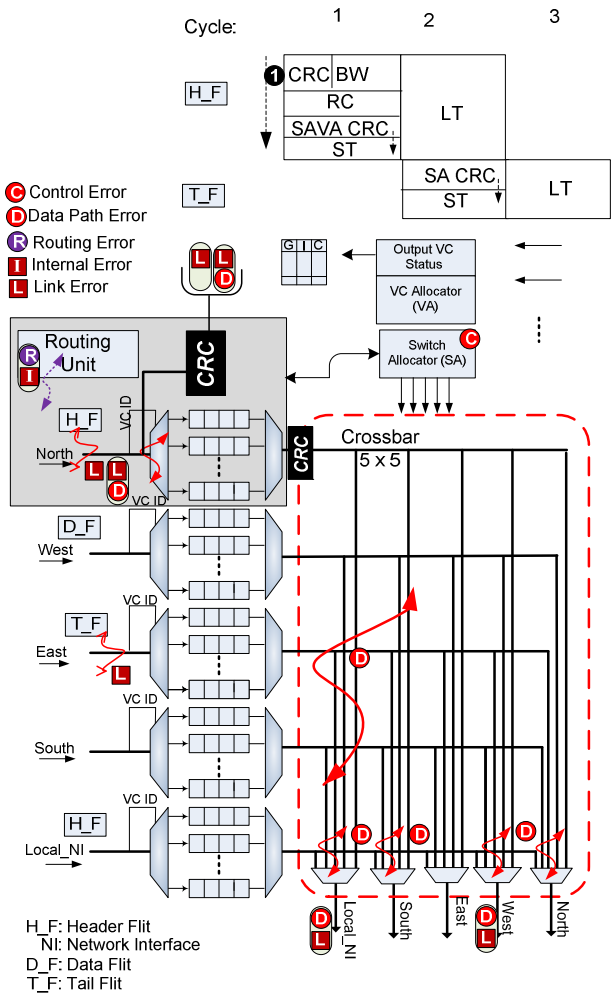


**Figure 1: A generic NoC router**

### 3.2. CRC stage

As Figure1 illustrates, soft errors fall into two categories: 1) Inter-router errors (link errors) 2) Intra-router errors (errors inside components of a router). An inter-router error results from cross-talk or noise on links. An intra-router error occurs as a consequence of an upset in data path or

control units of an on-chip router. The proposed mechanism in this paper handles Link errors (L) and Data path upsets in upstream router that can present themselves over link (D,L). In generic VC-based NoC routers, each physical channel involves the finite number of VCs. As Figure 1 illustrates, once a flit arrives from an upstream router to an input channel of the proposed router, its VC identifier (VC_ID) defined by an upstream router is decoded; the related H-F will be written to the buffer (BW stage) in the appropriate VC associated with its decoded VC-ID. Meanwhile, the CRC unit checks errors. In fact, the CRC check occurs in parallel with BW stage; therefore, it has no impact on the critical path of the router. Link errors (L) and Data path upset in upstream router (D,L) can be discovered at this stage. Meanwhile, the state of aforementioned VC will be changed to the routing state. Consequently, Routing Computation (RC), VC Allocation (VA) and Switch Allocation (SA) are done at the same stage, but VA followed by SA since we assume the scheme proposed in [15]. A winner of SA stage is selected through two separate stages of arbitration (local and global arbitration). VA is then accomplished simply by finding a free output VC from the requested output port of the SA winner. After SA -VC stage the flit is updated with a new VC ID and route field; therefore, the new CRC update for the modified field is needed. The CRC check starts its operation in the same clock cycle and forwards their results to SW traverse (ST) to be attached to the flit leaving the input channel.

### 3.3. RAVC Router

RAVC router with a novel architecture for dynamic virtual channel allocation is introduced in [3]. This architecture alleviates the effects of permanent faults in an on-chip network by isolating the router from sending packets and consuming network bandwidth. In addition, RAVC is architected in such a way that VCs inside the input channels of the routers surrounding a faulty router can be totally recaptured and reallocate to the other input ports.

Figure 2 illustrates reconfiguration in routers when there is a permanent fault inside a router. As shown in Figure 2, while a module connected to R[2,2] is transmitting packets to R[2,3], the occurrence of permanent faults in R[2,3] leads to reconfiguration of routers surrounding the R[2,3]. In particular, this reconfiguration acts in favor of R[2,2] and R[3,3] which are expected to receive more traffic.

In this paper, we use the same philosophy but with different implementation. RAVC router used a linked-list data structure placed in SRAM memories for managing Head-pointer and Next-Pointer [3]. In RAVC, Next-pointer memory should keep track of the flits addresses inside a shared memory, leading to infeasible hardware overhead.

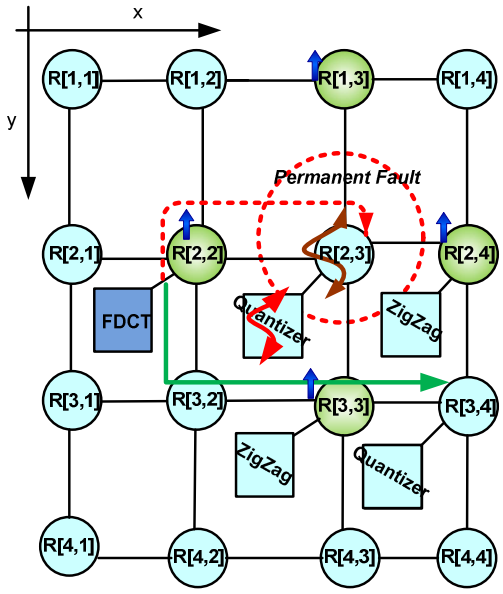


**Figure 2: RAVC router**

## 4. Proposed router architecture

Static VC allocation leads to unbalanced loading across VCs and Head of Line (HOL) blocking [16][3], particularly in the case of uneven traffic distribution among all direction such as burst traffic. To combat with HOL blocking, dynamic virtual channel regulation has been suggested [16][3]. Figure 3 shows our proposed input channel architecture. We have adopted Unified Buffer Structure (UBS) and History Aware Free-slot Tracker (HAFT) to regulate VC allocation. One of the distinctive features of the proposed router that will be used for reconfiguration is that the input port of each input channel can be connected to the other input ports. The congestion unit controls interconnection of input ports to input channels. This unit uses four global registers that indicate conditions of four neighbor routers. We use a counter for each output port.

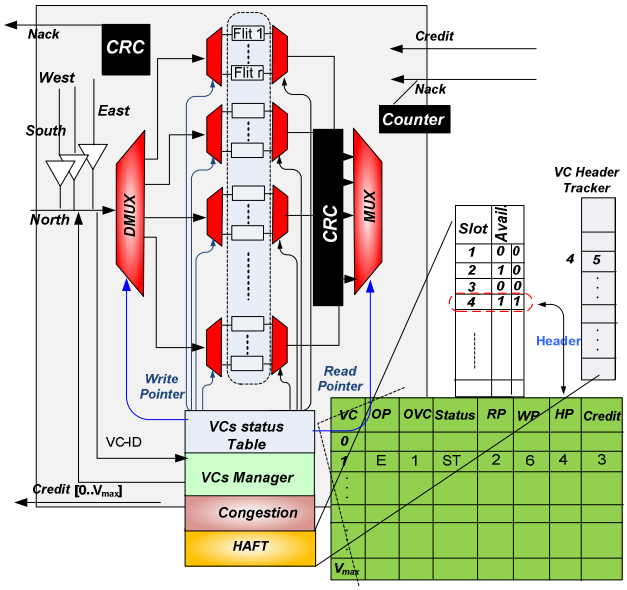


**Figure 3: The proposed input channel**

Whenever the proposed router receives a NACK signal from one of its neighboring routers, the associated counter of that particular port will be increased. After reaching a threshold value which is configurable, the status register of the downstream router will be changed to faulty. Then, congestion manager disconnects that faulty router. Plus, when there is a need for buffers in other input channels, HAFT will inform congestion manager of that input channel which is already disconnected from the faulty router to store new incoming packets.

The arriving header-flit after passing the CRC check is stored into one of the UBS slots, which is determined by VC manager and HAFT. In fact, since VC allocation caries out dynamically, an incoming flit cannot find the destined buffer slot using its VC-ID. HAFT stores status of the available slot. The VC Manager should then assign VC to the new header-flit. To carry out its task, VC Manager either dispenses a new VC or places that new incoming flit at the end of already existing virtual channels. As long as numbers of VCs do not exceed the $VC_{max}$, VC manager dispenses a new VC. Otherwise, it has to place the new incoming flit inside one of the available VC. Since in our proposed fault tolerant flow control, we need to keep track of all the header-flits, 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 at address adr2 will be placed inside the same VC. The content VC-Header tracker at address ad2 stores adr1.

When a flit other than the header flit leaves input channel VC manager and HAFT releases the buffer slot. As figure 3 shows associated with each slot HAFT store 2 bits, if a header-flit is stored on that slot these bits are (11) otherwise are (10). These bits will be used by Fault tolerant routing unit to decide if it is required to send a header-flit.

### 4.1. VC status table

VC status Table is used by the VC manager to dynamically regulate VCs in an input channel. The table contains VC number that ranges from 0 to $VC_{max}$, Output Port (OP), Output Virtual Channel (OVC), Read Point (RP), Write Pointer (WP), Header Pointer (HP) and Credit. The VC status Table is different form [16] in that the Header-address is also stored in the VC status table. In the initial condition, the number of credits is (Buffer-Size/$VC_{max}$).

As we discussed before, HAFT keeps track of the available slot in UBS; it triggers the congestion signal once it reaches the capacity limits, and as soon as a buffer slot get free by a departing flit, HAFT releases the congestion signal.

### 4.2. Fault-tolerant flow control

Figure 4 (A) (B) shows our proposed fault-tolerant flow control mechanism. The proposed flow control is a combination of the credit-based and ACK/NACK flow controls. In Figure4, router 1 (R1) transmits a packet (P1) that involves five flits to R4. In the credit-based flow control, once a router forwards a flit, it sends a credit to the downstream router. At the first step, R1 sends H1 to the downstream router (R2), while decreasing the credit value of the VC associated with that particular VC in its VC status table. Meanwhile, R2 stores H1 inside its UBS. At the next step, R1 sends D1 to R2, while R2 is sending H1 to its downstream router (R3). Thereafter, at the third step, D2 and D1 leave R1 and R2 respectively, and the corresponding credits are sent back to their upstream routers. However, due to the failure in R3's downstream router, it receives NACK signal while transmitting H1. R3 tries to resend H1 again as long as the numbers of unsuccessful retransmission do not exceed the threshold of the counter. After reaching the threshold value, counters will update condition of the downstream router to failure states. At this moment the packet (P1) that is already distributed among 3 routers needs to get restructured. Since HAFT tracks the header-flit of the packets in progress, it can easily change VC-Status table in such a way that read-pointer of that VC updated with its header-pointer and status of VC will be changed to VC-allocation. Hereafter, the VC which was in the transmission state need to compete again to get output VC.

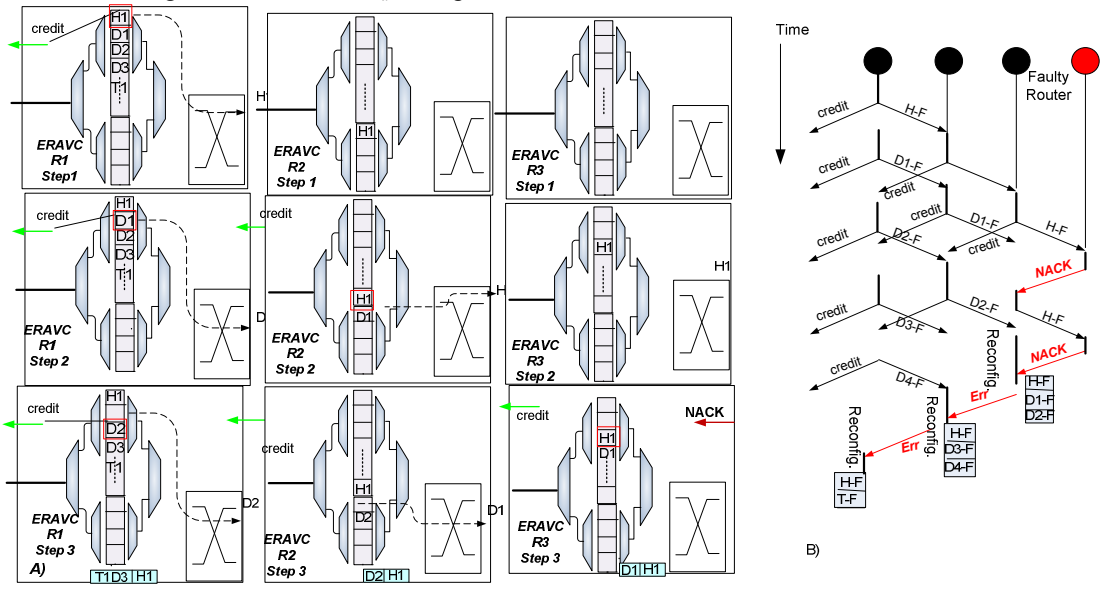


**Figure 4: The proposed flow control**

As Figure 4 shows, during the reconfiguration phase R1 can reroute part of the P1 involve (H1, D2 and T1), R2 reroutes (H1, D2) and R1 (H1, D1). Therefore, P1 can be rerouted and bypass the faulty router without loosing its data. Figure 4 (B) shows the back propagation of the Err signal from downstream router to the upstream which triggers reconfiguration of inside the VC status Table.

## 5. Experimental Result

To evaluate reliability and performance metrics of the proposed router in a fault-prone environment i.e. assuming both permanent and transient faults, we considered a 5 * 5 mesh-based topology. Figure 5 shows our experimental framework. We created a cycle-accurate systemC model of the router and augmented it with Orion power library [21] to evaluate energy consumption of the network. We assumed 70nm as the process feature size and 250 MHz as the working frequency. The flit size is 64 bits and every packet involves 8 flits.

We mapped a standard JPEG encoder systemC model to the mesh-bashed interconnect. JPEG is a commonly used lossy compression for images that typically achieves 10:1 compression with little perceptible loss in the image quality. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. Our JPEG model consists of seven systemC modules: YCBCR, Blocker, Down-Sampler, Digital Cosine Transform (DCT), Quantizer, ZigZag, and Huffman Coder. To have a fair reliability comparison between the proposed, generic and RAVC routers, we armed them with DY-AD fault-tolerant routing algorithm [10]. In our framework, we assumed two instances for every module (spare and original). In the case of permanent failures in a particular router, that router will be first isolated from transmitting and receiving packets; second, its neighboring routers will be notified and reconfigured to assign more buffers to their input channels; then, the spare core corresponding to the particular core connected to the faulty router will carry out the job of the faulty core.

For instance, Figure 5 (left side) shows that once R[2,3] find itself faulty, its neighboring routers R[1,3], R[2,2], R[2,4] and R[3,3] will be reconfigured. Thereafter, fault tolerant routing algorithm will change the previous path to the new one (Red line) that makes the spare Downsampler operate instead of original one. As we discussed earlier, the proposed fault tolerant flow control guarantees that packets which are distributed among several routers will be broken into smaller packets and rerouted again.

As our experiment, we consider the 64*64 bitmap image; this image will be placed inside the memory of YCBCR; eventually, final result will be prepared by Huffman module. The fault injection module which is also written in systemC injects permanents faults inside the routers. As long as simulation is running the fault injection module can inject faults; however, no more than six permanent faults are allowed, and the fault injection module cannot inject permanent faults to routers that are connected to the same module, or else simulation platform no longer can generate the final JPEG image. However, the fault injection module can inject transient faults to routers and links as long as simulation is running. To compare accurately reliability of the proposed router with generic and RAVC router, we exercised these routers with the same failure patters. We evaluate the average latency of the network plugged with model of Base-line router, RAVC and ERAVC. Experimental in Figure 5 shows that ERAVC provide in average 22% and 45% improvement in average latency with respect to RAVC and baseline router. Table 1 displays the average energy consumption of the generic router, RAVC, and our proposed ERAVC in the case of failures. When there are no failures in an on-chip network generic router consume less energy; however, while number of permanent failures increase ERAVC performs better in terms of energy consumption. The design is implemented in VHDL and synthesized using Synopsys design Compiler tool and the TSMC 90 nm technology library at supply voltage 1 V and an operating frequency of 500 MHz. The area of ERAVC is 101,544.49 µm2 which has 2.3 % overhead with respect to RAVC router.

**Table 1: Average energy consumption over different failure scenarios**

| #Perm. Failures | Power (W) | | |
|---|---|---|---|
| | Generic | RAVC | ERAVC |
| 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 |

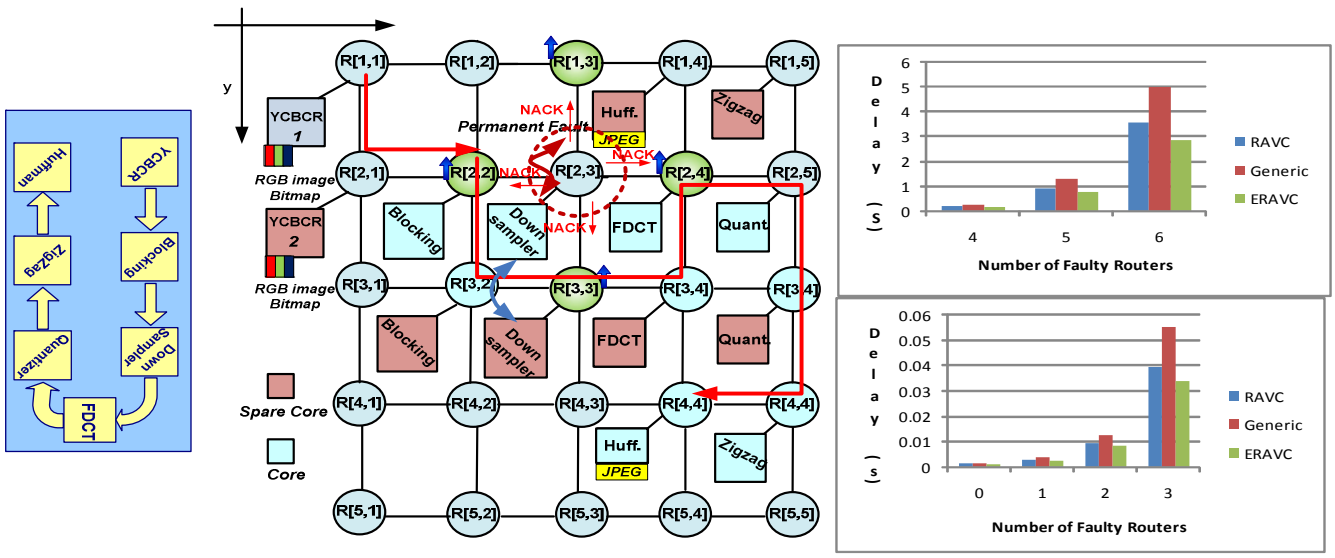


**Figure 5: Experimental platform**

## 6. Conclusion

In this paper, we proposed ERAVC router that supplies dynamic virtual channel allocation using UBS and HAFT. The proposed router reduces the associated performance costs of retransmission in the case of failure, by adopting a high-performance fault tolerant control flow. It handles both transient and permanent fault without extra retransmission buffer. Experimental results showed that the ERAVC improves reliability of on-chip network and decreases the average latency in a fault-prone environment. In our future work, we plan to extend ERAVC with debug enhancements and based on hardware assertions [24], and employ ERAVC in a hierarchical network [4].

## References


[1] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L. S. Peh, "Research challenges for on-chip interconnection networks," IEEE Micro, vol. 27, no. 5, pp. 96–108, 2007.

[2] Y. Kang, T. Kwon, J. Draper, "Fault-Tolerant Flow Control in On-Chip Networks," in Proceedings of the International Symposium on Network-on-Chip (NoCS), pp 79-86, 2010.

[3] M. H Neishaburi, Z. Zilic, "Reliability aware NoC router architecture using input channel buffer sharing," in Proceedings of Great Lake Symposium on VLSI (GLSVLSI), pp 511-516, 2009

[4] S. Bourduas and Z. Zilic, "A Hybrid Mesh/Ring Interconnect for Network-on-Chip Using Hierarchical Rings for Global Routing", Proceedings of ACM/IEEE International Symposium on Networks-on-chip, NOCS 2007, pp. 195-204, May 2007.

[5] A. Kohler, G. Schley, and M. Radetzki, "Fault Tolerant Network on Chip Switching with Graceful Performance Degradation," IEEE Transaction on computer-aided design (TCAD), VOL. 29, NO. 6, JUNE 2010 Symposium on Network-on-Chip (NoCS), pp 145-154, 2010

[6] W. J. Dally and B. Towles. "Route packets not wires: On-chip interconnection networks," in Proceedings of Design Automation Conference, pp 684-689, June 2001.

[7] P. Bogdan, T. Dumitras, and R. Marculescu, "Stochastic communication: A new paradigm for fault-tolerant network-on-chip," VLSI design 2007:17, 2007.

[8] T. Dumitras and R. Marculescu, "On-chip stochastic communication," in Proceedings of Design Automation and Test in Europe (DATE), 2003, pp. 790–795.

[9] M. Pirretti, G. Link, R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. Irwin, "Fault tolerant algorithms for network-on-chip interconnect," in Proc. Int. Symposium Very Large Scale Integr. (ISVLSI), 2004, pp. 46–51.

[10] J. Hu and R. Marculescu, "Dyad: Smart routing for network-on-chip," In Proceedings of Design Automation Conference (DAC), 2004, pp. 260-263.

[11] G. M. Chiu, "The odd-even turn model for adaptive routing," IEEE Tran. On Parallel and Distributed Systems, 11(7):729.738, July 2000.

[12] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, C. T. Das, "Exploring Fault-Tolerant Network-on-Chip Architectures," in Proceedings of International Conference on Dependable Systems and Networks, pp. 93-104, 2006.

[13] G. L. Frazier and Y. Tamir, "The design and implementation of a multiqueue buffer for VLSI communication switches," in Proceedings of the IEEE International Conference on Computer Design (ICCD), pp. 466-471, 1989.

[14] W. J. Dally, "Virtual-channel flow control," in Proceedings of the 17th Annual International Symposium on Computer Architecture (ISCA), pp. 60-68, 1990.

[15] A. Kumar, P. Kundu, A. P. Singh, L.Peh and N. Jha, "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, 2007

[16] A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, S. Yousif, R. Das, "ViChaR: A Dynamic Virtual Channel Regulator for Network-on-chip Router, 39th Micro, 2006.

[17] A. Karanth Kodi, A. Sarathy, and A. Louri,"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, 2008.

[18] J. Kim, C. Nicopoulos, D. Park, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks," in Proc. of the ISCA, 2006.

[19] Y. H. Kang, T. Kwon, and J. Draper. "Dynamic Packet Fragmentation for Increased Virtual Channel Utilization in On-Chip Routers", In Proc. of International Symposium on Networks-on-Chip, May 2009.

[20] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. De Micheli, "Analysis of error recovery schemes for etworks on chips," *IEEE Design Test Comput.*, vol. 22, no. 5, pp. 434–442, 2005.

[21] H.Wang, "A detailed Architectural-Level Power Model for Router Buffers, Crossbars and Arbiters," Technical report, Princeton University, 2004

[22] A. Pullini, F. Angiolini, D. Bertozzi, Luca Benini, "Fault Tolerance Overhead in Network-on-Chip Flow Control Schems," *in Proceedings of Symposium on Integrated Circuits and Systems Design*, pp. 224-229, 2005.

[23] P. Shivakumar, et al, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic", In Proceedings. Of International Conference on Dependable Systems and Networks, June 2002.

[24] M. Boulé, J-S. Chenard and Z. Zilic, "Adding Debug Enhancements to Assertion Checkers for Hardware Emulation and Silicon Debug", *Proceedings of IEEE International Conference on Computer Design, ICCD 06*, pp. 294-299, Oct. 2006.