Evaluation of Fault Tolerant Channel Buffers for Improving Reliability in NoCs

Dominic DiTomaso †, Travis Boraten †, Avinash Kodi †, and Ahmed Louri ‡
†*Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701*
‡*Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721*
E-mail: dd292006, tb286706, kodi@ohio.edu, louri@ece.arizona.edu

***Abstract*—Elastic or channel buffers can improve the overall power and area overhead of Network-on-Chip (NoC) architectures by reducing or replacing large, power hungry router buffers. In this paper, we design three fault tolerant schemes for our channel buffers which are used in a concentrated torus (CTorus) topology to reduce power consumption and improve throughput and latency. Our proposed fault tolerant techniques on CTorus topology are evaluated using the Synopsys Design Compiler and our results show (i) an improvement in energy-delay product (EDP) ranging from 20% to 43%, (ii) improvement in saturation throughput of 32% and (iii) an overall reduction in area overhead by 53-68% over other state-of-the-art electrical topologies.**

I. Introduction

As the scalability of Chip Multiprocessors (CMPs) increases, the Network-on-Chips (NoCs) architectures that connects hundreds to thousands of cores [1] must overcome the problems of global wire delay and power consumption. With technology scaling to smaller devices, both power dissipation (static and dynamic) and failures in router components are becoming crucial for future CMPs. While the previous design of 80-core Intel TeraFlops consumed more than 28% of the total chip power [2], more recent 48-core Intel SCC design [3] reduced the overall communication power to 10% of the total power budget by implementing several power optimization techniques. Clearly, energy-efficient and fault tolerant NoCs architectures are required to sustain and continue the performance gains achieved by increasing the number cores on a single chip with every successive generation.

Three research areas to improve NoCs architectures include (a) buffering and switching to improve energy-efficiency, (b) topologies designed for high throughput, and (c) fault tolerance for robust or reliable performance. Several design techniques have been proposed to minimize the high power consumption of router buffers, including (i) replacing the repeaters along the link to duplicate as hold and store (channel buffers) when desired [4] and (ii) replacing all buffers with elastic buffers along the link by replacing repeaters with flip-flops [5]. Minimizing crossbar area and power has also been studied by using smaller, segmented and split crossbars for improved energy and area-efficiency [6]. Flattened Butterfly (FBfly) is a high-radix NoC router architecture which reduces any extra hops along a dimension, thereby restricting the diameter of the network to two at the cost of increased router radix [7]. Finally, with technology feature sizes continuing to decrease, the mitigation of hard and soft errors are becoming a critical issue that will impact the performance of NoCs architectures. Fault tolerant techniques are needed to provide the expected high availability and data integrity that applications today require when full Dual Modular Redundancy (DMR) systems are not feasible or expensive to implement. This includes, variable strength error correction coding, detection and sampling techniques [8], built-in system test (BIST), and system adaptability. While prior work has shown the benefits of reducing power and area with channel buffers [4], there has not been an integrated evaluation that takes router optimizations (buffers and crossbars) into topology evaluation while overcoming component failures in NoCs.

In this paper, we propose fault tolerant channel buffers in a concentrated torus (CTorus) topology with the goals of minimizing power consumption, eliminating Head-of-Line (HoL) blocking, and further improving network performance. We avoid HoL blocking in CTorus by using dual channels (dc) between routers to increase the number of inputs. Furthermore, we take advantage of the dual inputs by separating the monolithic crossbar into smaller multiple crossbars (mx). A CTorus topology is used which balances the traffic load while providing performance comparable or better to the FBfly topology. To overcome failures in the channel buffers, we propose three different fault tolerant schemes which utilize the redundancy of the small, power efficient dual channel buffers. The first design, called No Escape Channel, does not add any additional links or escape channels, and simply uses the inherent redundancy of the dual channel organization when a failure is detected. The second design, called Full Escape Channel, adds an additional escape link that is used only when there is a failure in one of the links. The third design, called One Escape Buffer, is a mixture of the first two designs in that only a portion of a full escape channel is used when a failure is detected. We used the Synopsys Design Compiler to evaluate the power, area and router pipeline latencies for various configurations. The major contributions of this work are as follows: (1) we utilize a CTorus topology with energy efficient channel buffers and crossbars, (2) we propose three fault tolerant schemes which can overcome failures with marginal area and power overhead, and (3) we show a performance improvement of up to 32%, an improved EDP from 20% to 43%, and a total area reduction of 53-68% when compared to other NoC topologies such as CMesh and FBfly.

978-1-4673-2527-1/12/$31.00 ©2012 IEEE

II. Topology

We use a concentrated torus (CTorus) topology as shown in Figure 1(a). Torus topologies reduce the traffic contention at the center of the network by balancing the traffic load better than a mesh due to the wrap-around links allowing packets to travel in both directions. Network diameter can be reduced by concentrating four cores to a single router leading to a savings in power and area overhead. The CTorus topology is organized in a grid fashion on the chip with links between each adjacent router and wrap around links connecting routers on the edge of the chip. Figure 1(a) shows the bottom half of the 64 core topology and the wrap around links are shown as arrows for simplicity.

A single stage of each channel buffer is comprised of a single three-state repeater along all wires as shown in Figure 1(b) and 1(c). Each repeater stage acts as a conventional transmission repeater with the addition of an active low control input. When the control line for a stage is high, each repeater within that stage is tri-stated and the data is held from traversing. The dual-functionality of the links reduces the overhead of expensive buffers within the router and saves considerable power and area. Our baseline design uses no fault tolerance and requires a single control block per inter-router link in order to manage all the stages along the link. The dual channel (dc) design naturally overcomes Head-of-Line (HoL) blocking in the proposed architecture.

The dual links in the CTorus topology become the two input ports for the multi-crossbar (mx) organization. The mx organization splits a large monolithic crossbar into four smaller crossbars to reduce area and power consumption. The crossbars take advantage of xy dimension ordered routing (DOR) in which a packet travelling in the x direction will continue in the same x direction or move to a y direction. Once in the y direction, a packet will stay in the same y direction. This creates four quadrants of traffic: (+x, +y) [North-East], (-x, -y) [South-West], (-x, +y) [North-West] and (+x, -y) [South-East]. A router has two input and one output port for each direction. Therefore, by limiting the crossbar connections and combining select crossbar outputs, we provide more opportunities for the output ports to be occupied than a conventional crossbar. Deadlocks are avoided using VCs and dimension ordered routing which eliminates circular dependencies.

III. Fault Tolerance Channel Buffers

In this section, we describe in detail the implementation of the fault tolerant dual channel buffers and the control logic for overcoming faults using three different fault tolerance approaches. Our goal of implementing fault tolerance is to overcome device failure in our channel buffers with marginal power or area overhead while sustaining performance similar to the baseline. Our first design can achieve fault tolerance without the area overhead of an escape channel, but with slight performance penalty. The second and third design can maintain performance by adding either a full escape channel with four buffers or just one escape buffer. It should be noted that we

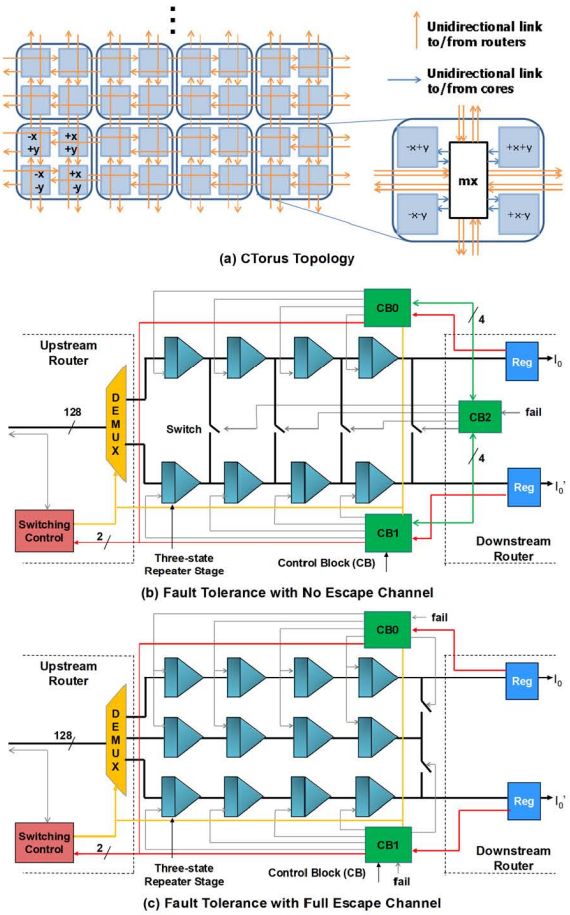

Fig. 1. (a) The CTorus topology and two different fault tolerant channel buffer designs including (b) No Escape Channel and (c) Full Escape Channel.

discuss how to recover from faults in this paper; the testing will be evaluated in future publications.

The No Escape Channel design shown in Figure 1(b), uses the inherent redundancy of the dual-channel design to provide a means for rerouting flits around faulty stages. When a stage is declared faulty, the control block labeled CB2 in Figure 1(b) is notified. From then on, if a flit attempts to traverse through a failed stage, the associated control block (CB0 or CB1) requests permission from CB2 to send the flit across the switch to the adjacent line. The flit bypasses the failed buffer and continues to the next stage on the secondary link. If contention arises, CB2 manages round robin arbitration to promote fairness to both lines. After the faulty stage is bypassed, the secondary control requests access at the next available active switch, and if granted the flit returns to its original line. While the design in Figure 1(b), takes a minimal area approach, the control logic suffers from complexity and is the only fault tolerance design implemented of the three that hinders performance due to arbitration delay after one faulty stage. The No Escape Channel design, however, can withstand multiple faults in both links given that no two parallel link stages are bad. This fault tolerant design achieves the goal of having marginal area and power overhead by not adding an

escape channel. However, when one channel buffer fails, a link must be shared between two packets causing one to be halted which will impact performance.

The Full Escape Channel design shown in Figure 1(c), adds a third full parallel channel buffer link. This channel is only used when there is a fault in one of the other channels. When a failure is detected, packets are re-routed into the escape channel then back out to the original input port. Full Escape allows for multiple stages or a single full channel to become defective before performance is hindered. However, there is an area overhead due to the additional escape channel. No additional power is added because the distance a packet traverses is the same even when there is a failure.

A third fault tolerance design, called One Escape Buffer, takes a Dual Modular Redundancy (DMR) methodology, and uses one redundant buffer stage per link. This design is similar to the Full Escape Channel except that there is only one tri-state buffer per inter-router link and a packet can be routed to this buffer after each stage. The input for this buffer is fed from a 1×4 MUX and output to a 1×4 DEMUX connected to each intermediate stage. This design allows faults at any one stage before affecting performance. When a buffer stage is declared defective, the associated control block is notified, and the respective select lines are activated to allow the flit to continue through the bypass stage. Therefore, similar to the Full Escape Channel design, the One Escape Buffer can maintain performance for a certain number of failures. However, there is a power and area overhead due to the MUX/DEMUX and additional wiring.

IV. Performance Evaluation

In this section, we first evaluate our three proposed fault tolerant channel buffer designs in terms of power, timing, and area overhead compared to a baseline dc buffer organization with no fault tolerance. We assumed a BIST architecture similar to [9] where they showed that 117 cycles (200 MHz clock) were required to detect faults in the channel. Each packet consists of 4 flits where each flit is 128 bits. The channel buffer and crossbar organization was synthesized and optimized using the Synopsys Design Compiler tool using the TSMC 65 nm technology libraries with a nominal supply voltage of 1.0 V and an operating frequency of 2 GHz.

TABLE I
Power and Area estimation using Synopsys Design Compiler for 65 nm technology node at 1.0 V and 2 Ghz clock.

Design	Power (mW) Link + Reg + CB + DEMUX/MUX	% Change
Baseline	180.81 + 2.78 + 0.09 + 0.20	-
No Escape Channel	180.81 + 2.78 + 0.63 + 0.20	+0.29
Full Escape Channel	180.81 + 2.78 + 0.09 + 0.20	0
One Escape Buffer	180.94 + 2.78 + 0.10 + 0.78	+0.39
	Total Area (mm^2) Link + Reg + CB + DEMUX/MUX	
Baseline	234.67 + 2.60 + 0.08 + 1.34	-
No Escape Channel	234.67 + 2.60 + 0.58 + 1.34	+0.21
Full Escape Channel	256.00 + 2.60 + 0.08 + 1.34	+6.3
One Escape Buffer	240.00 + 2.60 + 0.10 + 10.94	+8.9

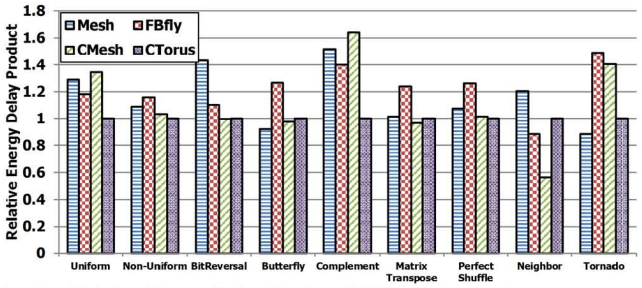

Fig. 2. Relative Energy-Delay Product (EDP) of 64 core topologies for all synthetic traffic.

Second, we evaluate our CTorus topology using the best fault tolerant channel buffer design and compare to mesh, Concentrated mesh (CMesh), and Flattened Butterfly (FBfly). For fair comparison, the bisectional bandwidth was maintained equal for all designs by adjusting the link widths.

A. Power

The total power dissipation of baseline dual-channel buffer, shown in Table I, is 183.8 mW for 65 nm comprising of the link, register, control block, and DEMUX/MUX. No Escape Channel slightly increased total power to 184.4 mW, while maintaining the same link cost as baseline with the control block and arbitration overhead providing the largest penalty. The Full-Escape Channel design dissipated 183.8 mW, with the majority of the increase in the link due to the redundant channel, but incurred minimal additional control block overhead. One Escape Buffer dissipated 184.60 mW mainly due to the 128 bit wide 1-to-4 DEMUX and MUX. The leakage power of the baseline, No Escape, Full Escape, and One Escape and was found to be 2.16, 3.27, 3.01, and 2.31 nW respectively.

B. Timing

The latency for the baseline was found to be 0.37 ns in 65 nm technology and is due to the channel buffer latency (0.20 ns), register buffering (0.09 ns), and the DEMUX (0.08) latency which falls within our specified clock period of 0.50 ns. For the three fault tolerance schemes, the same latency was incurred in each design except One Escape which employed additional MUX (0.07 ns) and DEMUX (0.08 ns) yielding 0.52 ns total delay. This delay does not fall within our clock period of 0.50 ns therefore it will not be considered for CTorus using 65 nm but can be lowered with future technology nodes.

C. Area

The baseline area, shown in Table I, of 238.6 mm^2 was estimated from the link, register buffers, control block, and DEMUX. No Escape yielded the best area of the three fault tolerance schemes with 239.1 mm^2, due to the advantage of the inherent redundancy natively seen in the design, eliminating the need of additional redundant components at the cost of control block complexity. One Escape provided the second best area and contained overhead penalties in link cost due to a single redundant buffer stage per channel and complexity of

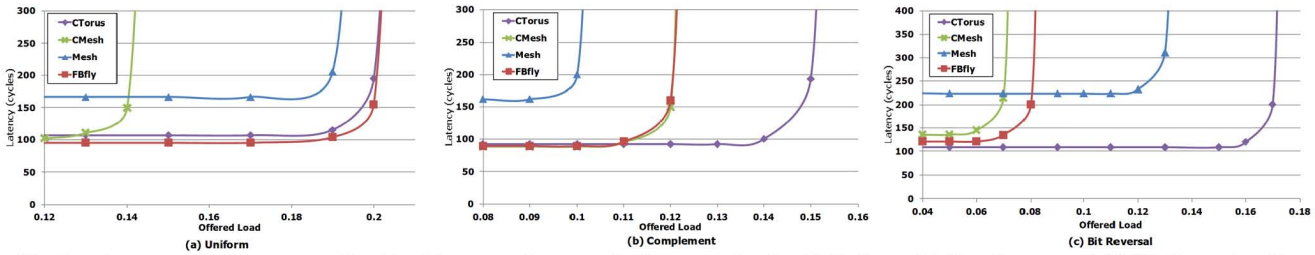

Fig. 3. Average packet latency as offered load increases from zero load to saturation for (a) Uniform, (b) Complement, and (c) Bit Reversal traffic.

the required MUX/DEMUX components needed to drive the bypass stage. The Full Escape required the most area overhead due to an additional full channel.

D. CTorus Results

A cycle-accurate on-chip network simulator was used to conduct a detailed evaluation of each network. For open-loop measurement, the packet injection rate is varied from 0.1 to 0.9 of the network capacity, and packets are injected according to the Bernoulli process based on the given network load. The simulator was warmed up and allowed to run until all the packets reached their destinations. We implement the full escape channel design in CTorus as this provides the best overall power and area combination while maintaining performance.

1) EDP: Figure 2 shows the average energy-delay product (EDP) per packet normalized to CTorus. Mesh has a high EDP for some cases but not all with an average EDP 18% higher than CTorus. This is due to the large network diameter of mesh causing high latency. CTorus has the lowest latency and energy for most traffic patterns with an average EDP 15-21% lower than all the other networks. CTorus has the same network diameter as CMesh; however, the low energy from the crossbar design and buffer channels allows CTorus to have a lower EDP for many cases.

2) Latency: Figure 3 shows the average latency per packet for all the networks. With uniform traffic, CTorus has similar performance to FBfly. The dual channels of CTorus allow packets to injected into the cores faster than the single channel links in FBfly. However, the extra links between routers in FBfly causes the packet latency to be similar to CTorus. Figure 3(b) shows the average latency for complement traffic. CTorus saturates a load of 0.15 which is 25% higher than FBfly and CMesh. Figure 3(c) shows the latency results for the bit reversal traffic pattern. The saturation of CTorus is at a load of 0.17 which is 30% higher than mesh and 70% higher than CMesh and FBfly. The concentrated networks start at a lower average latency than mesh with CTorus being the lowest. However, CMesh and FBfly saturate earlier because of the contention at the concentrated router. CTorus saturates at higher loads because of the increased input and output ports.

3) Area: CTorus saves 62% area over mesh, 53% area over CMesh, and 68% area over FBfly. The baseline 8×8 crossbar used in CMesh and was estimated to have an area of 0.492 mm^2. The split mx crossbars in CTorus reduces the crossbar area overhead by 79% compared to the baseline. The high radix router as well as the many wired links in the FBfly topology causes the total network area to be over 16 mm^2. While the mesh topology has lower radix routers, the area is still almost 14 mm^2 due to the large number of routers needed at each core.

V. Conclusions

In this paper, we design three fault tolerant schemes for channel buffers. Our fault tolerant designs maintain performance when faults occur with minimal power and area overhead. In addition, we compare leading topologies such as mesh, CMesh, and FBfly to a CTorus topology which utilizes the proposed fault tolerant channel buffers and improves EDP by 20% to 43%.

Acknowledgment

This work was partially supported in part by the National Science Foundation grants ECCS-0725765, CCF-0953398, CCF-0915418, CCF-1054339 (CAREER) and ECCS-1129010.

References

[1] L. Benini and G. D. Micheli, "Networks on chips: A new soc paradigm," *IEEE Computer*, vol. 35, pp. 70–78, 2002.

[2] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-ghz mesh interconnect for a teraflops processor," *IEEE Micro*, pp. 51–61, September/October 2007.

[3] S. Borkar, "Will interconnect help or limit the future of computing?" Presented at the 19th Annual IEEE Symposium on High-Performance Interconnects (Hot Interconnects), Santa Clara, California, 2011.

[4] A. K. Kodi, R. Morris, D. DiTomaso, A. Sarathy, and A. Louri, "Co-design of channel buffers and crossbar organizations in nocs architectures," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2011.

[5] G. Michelogiannakis, J. Balfour, and W. J. Dally, "Elastic-buffer flow control for on-chip networks," in *Proceedings of the Fifteenth International Symposium on High-Performance Computer Architecture*, 2009, pp. 151–162.

[6] F. Gilabert, M. Gomez, S. Medardoni, and D. Bertozzi, "Improved utilization of noc channel bandwidth by switch replication for cost-effective multi-processor systems-on-chip," in *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, May 2010, pp. 165 –172.

[7] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: Cost-efficient topology for high-radix networks," in *Proceedings of 34th Annual International Symposium on Computer Architecture(ISCA)*, June 2007, pp. 126 – 137.

[8] S. Nomura, M. D. Sinclair, C.-H. Ho, V. Govindaraju, M. de Kruijf, and K. Sankaralingam, "Sampling + dmr: practical and low-overhead permanent fault detection," *SIGARCH Comput. Archit. News*, vol. 39, no. 3, pp. 201–212, Jun 2011.

[9] J. H. Collet, A. Louri, V. T. Bhat, and P. Poluri, "ROBUST: a new self-healing fault-tolerant noc router," in *Proceedings of the 4th International Workshop on Network on Chip Architectures*, 2011, pp. 11–16.