Soft-Error Resilient 3D Network-on-Chip Router

Khanh N. Dang, Michael Meyer, Yuichi Okuyama,
and Abderazek Ben Abdallah
The University of Aizu
Graduate School of Computer Science and Engineering
Aizu-Wakamatsu 965-8580, Japan
Email: {d8162103, d8161104, okuyama, benab}@u-aizu.ac.jp

Xuan-Tu Tran
Smart Integrated Systems Laboratory
VNU University of Engineering and Technology
Vietnam National University, Hanoi
Hanoi, Vietnam
Email: tutx@vnu.edu.vn

***Abstract*—Three-Dimensional Networks-on-Chip (3D-NoCs) have been proposed as an auspicious solution, merging the high parallelism of the Network-on-Chip (NoC) paradigm with the high-performance and low-power of 3D-ICs. However, as feature sizes and power supply voltages continually decrease, the devices and interconnects have become more vulnerable to transient errors. Transient errors, or soft errors, have severe consequences on chip performance, such as deadlock, data corruption, packet loss and increased packet latency. In this paper, we propose a soft-error resilient 3D-NoC router (SER-3DR) architecture for highly-reliable many-core Systems-on-Chips. The proposed architecture is able to recover from transient errors occurring in different pipeline stages of the SER-3DR. We implemented the architecture in hardware with 45 nm CMOS technology. Evaluation results show that SER-3DR is able to achieve a high level of transient error protection with a latency increase of 18.16%, an additional area cost of 14.98% and a power overhead of 5.90% when compared to the baseline router architecture.**

I. Introduction

Global interconnects are becoming the major performance bottleneck for high-performance Multi/Many-core Systems-on-Chips (MSoCs). For more than a decade, Network-on-Chip (NoC) interconnects have been proposed as a promising solution for future MSoC designs [1]. The NoC paradigm offers more scalability than conventional shared bus interconnects and allows more processing elements (PEs) to be efficiently integrated into a single chip. Despite the higher scalability and parallelism offered by a NoC system over traditional shared-bus based systems, it is still not an ideal solution for future large scale MSoCs. This is due to some limitations such as high power consumption and low throughput. Merging NoC to the third dimension (3D-NoCs) has been proposed to deal with the above problems, as it was a solution offering lower power consumption and higher speeds [2]–[5].

As feature sizes and supply voltages continually decrease, systems implemented with these interconnects have become more vulnerable to soft errors. *Shivakumar et al*. [6] analyzed the transient error trends for smaller transistors and showed that the occurrence rate of transient faults is significantly higher than the permanent faults. In particular, they expect the transient error rate for combinational logic to increase dramatically.

There are several causes of transient faults that affect the operation of a circuit for a small period of time, typically for about one clock cycle. Common causes are: cosmic radiation [7], process variation [8] and alpha particles [9]. Faults result in severe consequences on overall chip performance, such as deadlock, data corruption, packet loss and increased packet latency. Therefore, without efficient protection mechanisms, transient errors, or soft errors, can compromise system reliability.

There are two main methods for achieving soft-error recovery in MCSoC systems. The first approach is software-based methods, where additional copies of a program are executed in order to obtain soft-error resilient results [10]. Although software-based methods have less modifications to the hardware, they introduce large overheads on task execution time and power consumption. The second approach is hardware-based methods, where additional circuits are designed in conjunction with common functional units to provide error protection. For example, *Triple Modular Redundancy* (TMR) [11] uses three identical subsystems to process the same task and a majority voting of the results is used to determine the correct output.

Previously, in [2]–[5], we proposed hardware techniques and smart routing algorithm to tackle hard-errors in the router. Specifically, our architecture is capable of recovering from faults in links, input buffers and crossbars [5].

In order to deal with the soft errors in Network-on-Chip, there are several existing works targeted to numerous layers. In case of data corruption, the most efficient solution is using *Error Correcting Code/Error Detecting Code* (ECC/EDC) such as: *SEC* (Single Error Correction), *SECDED* (Single Error Correction, Double Error Detection), *ED* (Error Detection), *PAR* (Parity Code), *CRC-4* (Cyclic Redundancy Check) and *CRC-8* [12]. For adaptive code, *Yu et al.* [13] presents a dynamic *ECC* of two *Hamming Code* which reconfigured based on quality of connection. For the logic corruption, most of works perform in cross network layers. With End-to-End flow control, *Shamshiri et al.* [14] presents an error-correcting and on-line diagnosis using a specific code named *2G4L*. *NoCAlert* [15] implements module's constraints to obtain computational accuracy from sub-module of router to end-to-end connection. *FoReVer framework* [16] also presents a network level method to periodically detect and recover from routing errors: loss, duplicated, and misrouted packets. Although the above works present several efficient solutions

to deal with soft-errors on data and routing logic, the pipeline stages of routers are still need to be protected from soft errors. Since the pipeline stage failure simultaneously impacts to the software and network correctness, we need an on-line, low-latency and low-cost technique to detect and recover from such failures. Therefore, this paper presents a detection and recovery solution which satisfies these requirements.

In this paper, we propose a soft-error resilient 3D-NoC router (SER-3DR) architecture for highly-reliable many-core Systems-on-Chips. The proposed architecture is able to recover from transient errors occurring in different pipeline stages of the SER-3DR. The rest of this paper is organized into five sections. Section II presents a brief overview of the baseline OASIS-3D-NoC system. Section III and Section IV present the proposed soft-error resilient 3D-NoC router (SER-3DR) architecture and algorithm respectively. Section V presents the implementation and evaluation results. Finally, the last section presents concluding remarks and future work.

II. 3D-OASIS Network-on-Chip

The 3D-OASIS-NoC (3D OASIS Network-on-Chip) system architecture and the router block diagram, with its three main pipeline stages: (Buffer Writing, Routing calculation/Switch Arbitration and the Crossbar Traversal), are shown in Fig. 1(c). 3D-OASIS-NoC adopts *Wormhole-like* switching. The forwarding method, chosen in a given instance, depends on the level of the packet fragmentation. For instance, when the buffer size is greater than the number of flits, *Virtual-Cut-Through* is used. However, when the buffer size is less than or equal to the number of flits, *Wormhole* switching is used. In this way, packet forwarding can be executed in an efficient way while maintaining a small buffer size [4], [5].

The router is the back-bone component of the 3D-OASIS-NoC design. Each router has a maximum number of 7-input and 7-output ports, where 6 input/output ports are dedicated to the connection to the neighboring routers and one input/output port is used to connect the switch to the local computation tile. The number of input-ports depends on the router position in the network because we need to eliminate any unused ports to reduce the area and power consumption.

The 3D-OASIS-NoC router contains seven *Input-port* modules for each direction in addition to the *Switch-Allocator* and the *Crossbar* module, which handle the transfer of flits to the next neighboring node. The *Input-port* module is composed of two main elements: *Input-buffer* and the *Next-Port-Routing* module. Incoming flits from different neighboring routers, or from the connected computation tile, are first stored in the *Input-buffer*. This step is considered as the first pipeline stage of the flits life-cycle, Buffer-Writing (BW).

Since 3D-OASIS-NoC is targeted for various applications, the payload size can be easily modified in order to satisfy the requirements of specific applications. After being stored, the flit is read from the FIFO buffer and advances to the next pipeline stage. The addresses ($xdest$, $ydest$ and $zdest$) are decoded in order to extract the information about the destination address, in addition to the *Next-Port* identifier

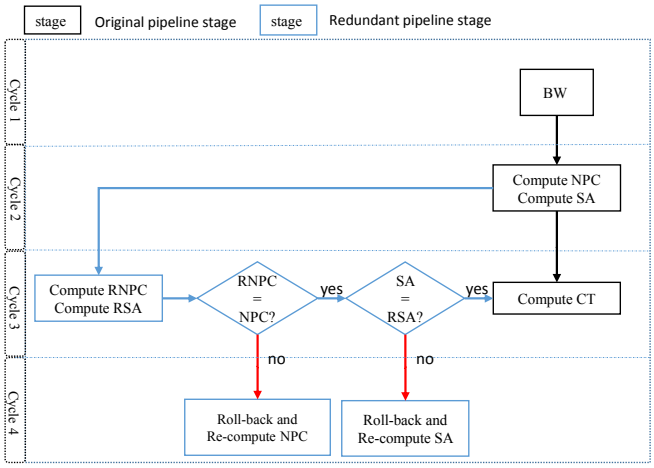

Fig. 2: SER-3DR pipeline stages.

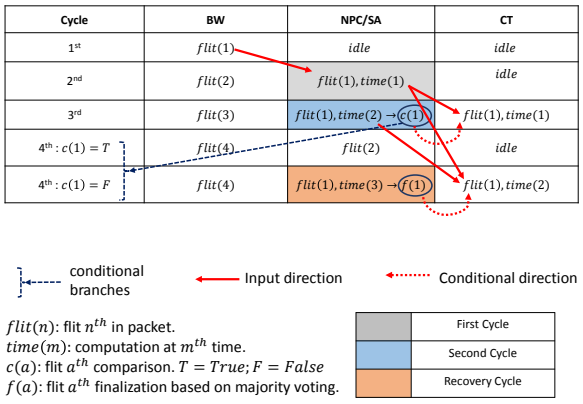

Fig. 3: SER-3DR pipeline timeline chart.

which is pre-calculated in the previous upstream node, and the fault information is received from *Fault Controller*. These values are sent to the *Next-Port-Routing* circuit where *LAFT* (Look-Ahead-Fault-Tolerance) is executed to determine the *New-next-Port* direction for the next downstream node. At the same time, the *Next-Port* identifier is also used by the *Switch Request Controller* to generate the request for the *Switch-Allocator* asking for permission to use the selected output port via *sw-req* and *port-req* signals.

III. Soft-error Resilient Router Architecture

Our main goal in proposing SER-3DR (Soft-Error Resilient 3D-NoC Router) is to develop a highly-reliable and low-cost technique to recover from soft-errors in all pipeline stages of the router. For ease of understanding, we provide a high-level view of the pipeline stages in Fig. 2 and the timeline-chart of the SER-3DR pipeline stages in Fig. 3. As shown in Fig. 2, the baseline OASIS router has three pipeline stages: (1) BW (buffer writing), (2) NPC/SA (Next Port Computation and Switch Allocation), and (3) CT (Crossbar Traversal).

To deal with the soft-error, the data corruption can be efficiently removed by using an ECC [12], [17]. Therefore,

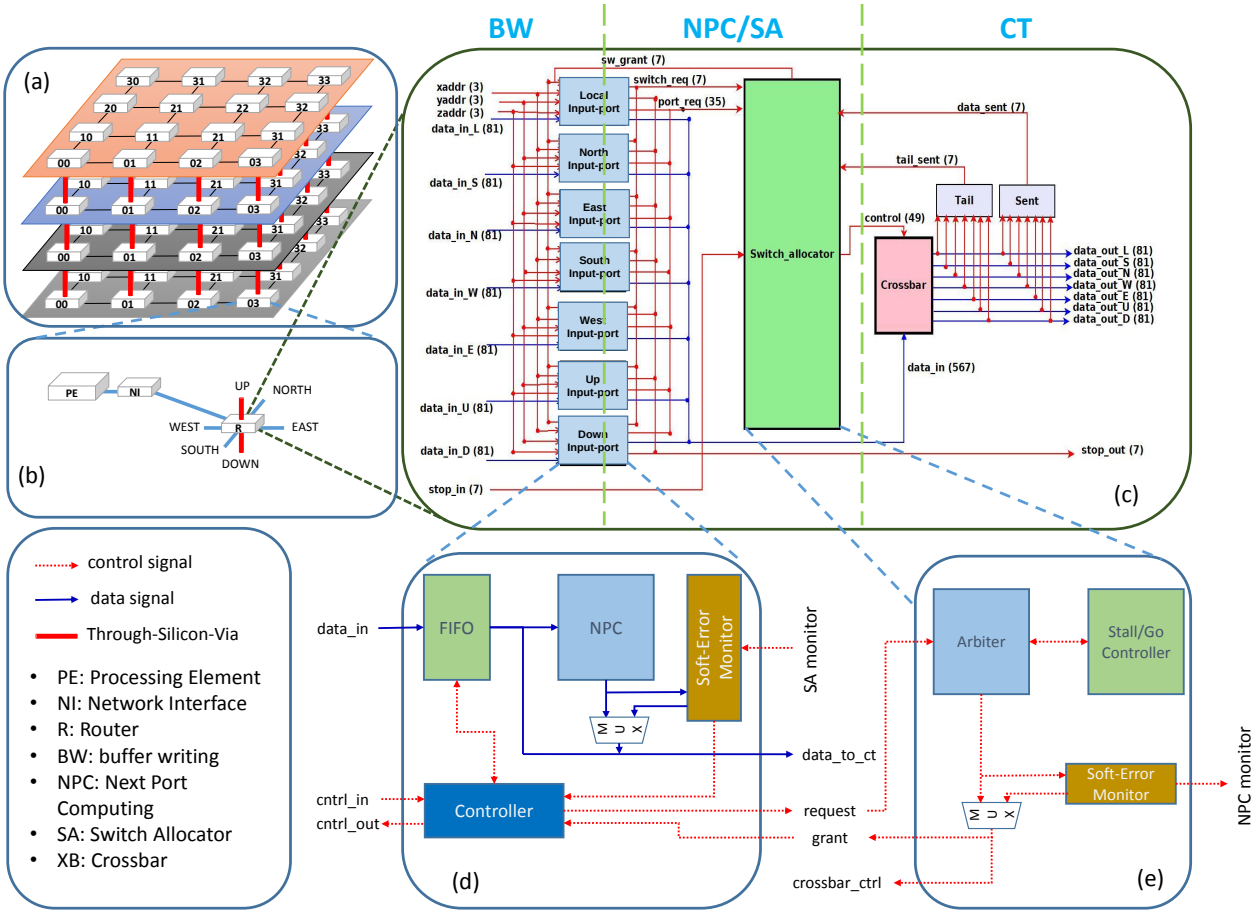

Fig. 1: 3D-NoC architecture high-level view.

this paper only focuses on the soft-error on router's logic. Since the NPC/SA stage (*Routing and Arbitrating*) consists of the most complexity combinational logic in the router, this stage is selected to apply our proposal technique. As shown in Fig. 2, the SER-3DR architecture extends the finite state-machine (FSM) of the baseline router so that the NPC and SA stages are recomputed (RNPC and RSA) in parallel with the CT stage. In terms of architecture, we add two lightweight monitoring modules into the input-port and the switch allocator, as shown in Fig. 1(d) and 1(e). These modules manage redundant computation, detect the appearance of soft-errors and decide to roll-back and re-compute NPC/SA when a soft-error occurs. The details of their operations are given in Section IV.

In Fig. 3, we present a timeline chart of a soft-error resilient router. $[flit(n)]$ presents the flit in the n^{th} position of the packet. $[time(m)]$ illustrates the m^{th} time of computation. In the first clock cycle, BW handles $[flit(1)]$ while NPC/SA and CT are idle or handle another packet. In the second cycle, NPC/SA computes $[flit(1), time(1)]$, meaning computation of the first flit at the first time. In the third cycle, NPC/SA computes $[flit(1), time(2)]$, meaning it computes the first flit for the second time also known as redundant computing. $[c(1)]$ compares the results of $[flit(1), time(1)]$ and $[flit(1), time(2)]$ to detect the occurrence of a soft-error. If there is no error, CT processes $[flit(1), time(1)]$ to finish the pipeline stages of the first flit. If there is an error on NPC/SA, the system requires the recovery fourth cycle. In this cycle, NPC/SA re-calculates the first flit for the third time as recovery: $[flit(1), time(3)]$ and finalizes an accurate result by using majority voting: $[f(1)]$. After getting the final result of the first flit, CT completes the pipeline stage of the first flit based on the correct result of the two previous computations: $[flit(1), time(1)]$ or $[flit(1), time(2)]$. As shown in Fig. 3, SER-3DR requires one clock cycle for detecting the soft-error and one optional cycle for recovery each time a error occurs.

IV. Soft-error Resilient Router Algorithm

The proposed Soft-Error Resilient Algorithm (SERA) of SER-3DR resolves soft-errors which appear inside the router's pipeline stages. At every processing header flit, SERA computes the monitored pipeline stage in two clock cycles to judge when soft-errors occur. When a soft-error occurs, SERA requires one additional clock cycle to roll-back and re-calculate the faulty pipeline stage. After re-calculating, SERA can

Algorithm 1 SERA Algorithm for SER-3DR

```
1: procedure SERA
2:     // input flit's data
3:     Input: in_flit;
4:     // output flit's data
5:     Output: out_flit;

6:     // Write flit's data into buffers
7:     BW(in_flit);

8:     // Compute first time of NPC and SA
9:     next_port[1] = NPC(in_flit);
10:    grants[1] = SA(in_flit);

11:    // Compute redundant of NPC and SA
12:    next_port[2] = RNPC(in_flit);
13:    grants[2] = RSA(in_flit);

14:    // Compare orginal and redundant to detect soft-error
15:    // Soft-error on NPC
16:    if next_port[1] ≠ next_port[2] then
17:        // roll-back and recalculate NPC
18:        next_port[3] = NPC(in_flit);
19:        final_next_port = MajorityVoting(next_port[1,2,3]);
20:    else
21:        // No soft-error on NPC
22:        final_next_port = next_port[1];
23:    end if

24:    // Soft-error on SA
25:    if grants[1] ≠ grants[2] then
26:        // roll-back and recalculate SA
27:        grants[3] = SA(in_flit);
28:        final_grants = MajorityVoting(grants[1,2,3]);
29:    else
30:        // No soft-error on SA
31:        final_grants = grants[1];
32:    end if
33:    // After detection and recovery, SERA finishes with CT
34:    out_flit = CT(in_flit, final_next_port, final_grants);
35: end procedure
```

decide the accurate output of a faulty pipeline stage based on the three consecutive results using majority voting.

As shown in Algorithm 1, SERA routes a flit from an input port to an output port. The input flit's data (in_flit) is first written into the input buffer by BW stage (line 7). Second, SERA computes the first-time NPC and SA stages which output the next_port[1] and grants[1] respectively (lines 8-9). Third, the redundant processes of NPC and SA (RNPC and RSA) are performed with these outputs: next_port[2] and grants[2] (lines 12-13). In the next step, SERA compares the outputs of the original and the redundant processes. If next_port[1] is different from next_port[2], a soft-error occurred in the NPC, the algorithm calculates NPC a third time and uses majority voting to decide the final value. Otherwise, the final value is assigned as the first result. SA is also processed in a similar fashion to NPC: determining error's occurrence, finalizing value or assigning first value. After detection and recovery, SERA finishes with crossbar traversal.

V. Design and Evaluation Results

A. Methodology

Our proposed system (SER-3DR) is integrated into OASIS 3D-NoC [4], [5]. We designed the system in Verilog-HDL, and synthesized using 45nm technology library [18]. For the Through-Silicon-Via (TSV) integration, we used FreePDK3D45 kit compiler [19]. We evaluated the hardware complexity, power consumption and speed. We also evaluated the throughput and End-To-End (ETE) delay using Matrix-multiplication, Transpose and Uniform benchmarks. For comparison, we also implemented and simulated the baseline LAFT-OASIS [4], HLAFT-OASIS [5], and Triple Modular Redundancy of NPC/SA based on OASIS (TMR-OASIS).

The Matrix multiplication benchmark is selected due to its complexity in terms of throughput requirement and computational parallelism. To perform the multiplication of two 6×6 matrices, we establish a $6 \times 6 \times 3$ 3D-Mesh based network, which consists of two layers for the input matrices and one layer for the result. We also execute transpose traffic pattern based on matrix transposition. Each node in the network sends flits to its index-reversed position. Finally, Uniform traffic pattern is chosen to analyze network performance. In this benchmark, each node sends flits to every other node with equal probability and data size.

To study the soft-errors affect on the proposed architecture, we create "injection modules" to inject errors into NPC/SA stage of SER-3DR. We also injected to the baseline LAFT-OASIS similar error rates. We measured the system execution time as the interval from the first sent flit to the last delivered flit. The crash events are also recorded as the soft-error reliability of LAFT-OASIS. Since our recovery method is based on the majority voting of three consecutive results, the maximum error rate of our proposal architecture is 1 error in every 3 clock cycles ($\simeq 33.33\%$). We also select independent rates for NPC and SA stages. For convenience, we use $A\%$ to denote the injection rates of both NPC and SA ($A\%$). Rate $A\%\&B\%$ denotes the injection rate of NPC and SA are $A\%$ and $B\%$, respectively.

B. Hardware Complexity

Table I depicts the implementation result of the original OASIS system, the TMR-OASIS, and the proposed SER-

3DR on 45 nm CMOS process and FreePDK3D45 TSV's technology. Table II presents the Network-on-Chip configuration. Table III depicts the ASIC parameters to implement the proposal architecture. Layout of SER-3DR is shown in Fig. 4. In comparison with the original LAFT-OASIS router architecture, the SER-3DR requires slightly more logic's area cost: 14.98% while the TMR-OASIS costs more 45.20% since it duplicates three times NPC and SA stage. The frequency decreases from $801.28\ MHz$ to $655.74\ MHz$ (-18.16%) due to additional combinational logic (compare and majority voting) in the critical path. TMR-OASIS adds only a majority voting in the critical path, therefore its impact is slightly better. On the other hand, TMR-OASIS increases the power consumption to $30.31\ mW$ ($+18.30\%$). The proposed design slightly increases the power consumption from $25.62\ mW$ of baseline to $27.13\ mW$ ($+5.90\%$). Notice that the TSVs cost the major part of area cost and power consumption.

TABLE I: Hardware complexity comparison results.

Design	Max Freq. (MHz)	Total Power mW)	Logic's area (μm^2)	# TSVs
LAFT OASIS	801.28	25.62	14,920	164
TMR-OASIS	763.36	30.31	21,664	164
SER-3DR	655.74	27.13	17,154	164

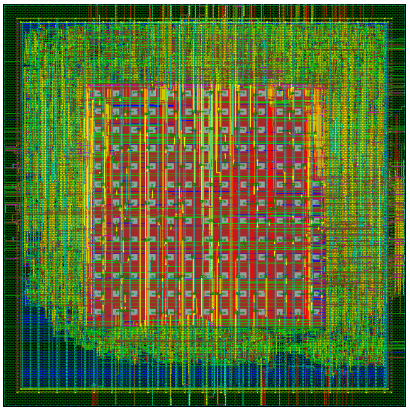

Fig. 4: SER-3DR router layout with 45 nm CMOS process.

C. End-to-End Delay Evaluation

We evaluate the End-to-End Delay (ETE) over different Flits/Packet from 1-100 $flits/packet$ and three injection rates (0%, 11.11%&6.67%, 33%). Figure 5 shows the ETE evaluation. From this figure, we can see that with the smallest

TABLE II: Network configuration.

Parameter	Value
# ports	7
Topology	3D Mesh
Routing Algorithm	Look-ahead routing
Flow Control	Stall-Go
Forwarding mechanism	Wormhole
Input buffer	4

TABLE III: Technology parameters.

Parameter	Value
Technology	Nangate 45 nm FreePDK3D45
Voltage	1.1 V
Chip's size	$300\mu m \times 300\mu m$
TSV's size	$4.06\mu m \times 4.06\mu m$
TSV pitch	10 μm
Keep-out Zone	15 μm

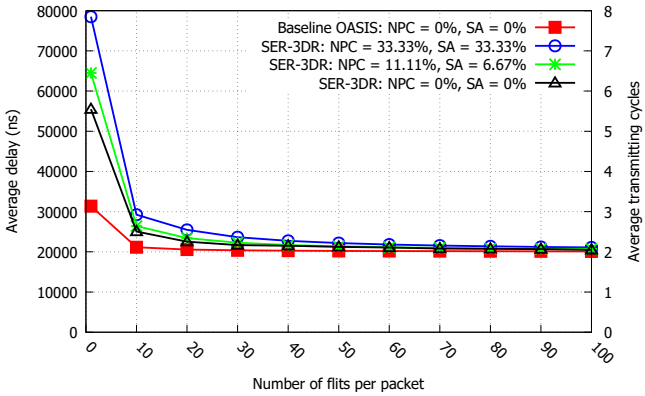

Fig. 5: Average End-to-End delay of Transpose Benchmark: Network size: 64 ($4 \times 4 \times 4$).

packet length (1 $flit/packet$), the proposed SER-3DR based architecture outperforms the unprotected OASIS NoC baseline architecture with the worst case of the ETE evaluation is a 33% error rate. Since the redundant computing cycles are required with each header flit, smaller flits sizes suffer higher impact in ETE latency. Furthermore, the routers have to wait for the diagnosis and the recovery process, therefore the network also imply more arbitrating time. However, for medium packet lengths (10 to 30 $flits/packet$), the ratio of the redundant cycles per the total transferring cycles is reduced. Therefore, the ETE delay is also decreased. Moreover, we can see significant performance benefits from using the SER-3DR with long packet's size. For example, for 100 $flits/packet$, the ETE is reduced by about 73.13% with a 33% error rate in SER-3DR. It is worth noting that a higher number of flits per packet leads to a slight convergence of all models and error rates. This small impact can be explained by the ratio of redundant cycles per total transferring cycle is insignificant, for example: about $1/100$ for 100 $flits/packet$. This ratio creates a light effect to the system performance. For the highest number of flits per packet (100 $flits/packet$) and Transpose benchmark, the baseline systems's ETE is $20,113\ \mu s$ with a 0% error rate and $21,092\ \mu s$ for SER-3DR with a 33% error rate.

D. Execution Time Evaluation

For this evaluation, we used the three benchmarks over five injection rates : 0%, 8.33%, 16.67%, 11.11%&6.67% and 33%. The evaluation results with Transpose, Uniform, and Matrix are shown in Figure 6, 7, and 8, respectively. We

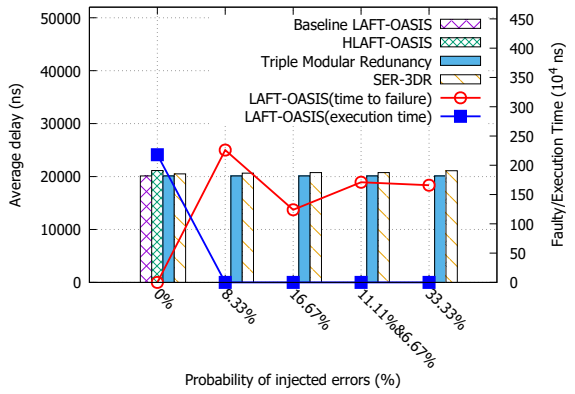

Fig. 6: Transpose Benchmark: Network size: 64 ($4 \times 4 \times 4$).

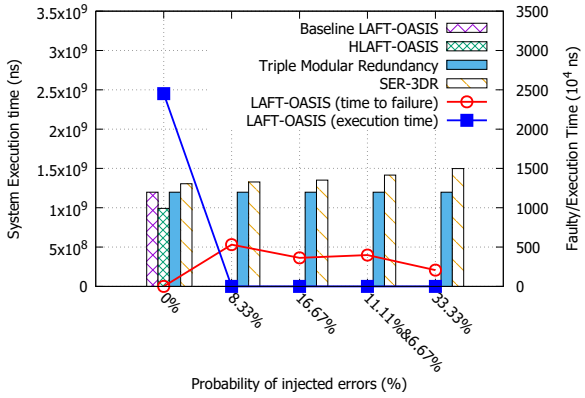

Fig. 7: Uniform Benchmarks: Network size: 64 ($4 \times 4 \times 4$).

perform these benchmarks for 4 models (SER-3DR, LAFT-OASIS, HLAFT-OASIS and TMR-OASIS). The system execution time or average delay is presented as bar graph. We also inject the soft-errors inside the baseline model (LAFT-OASIS) and measure the execution time. Its *time to failure* or complete *execution time* is depicted as line graph format.

For Transpose benchmarks in Fig. 6, we found that the average execution time slightly increases from $20,113$ μs to $20,505$ μs ($+1.95\%$) for an error injection rate of 0%. With different error injection rates, we can see that the average execution time slightly increases from $20,505$ μs for a 0% error rate to $21,092$ μs for a 33% error rate. Uniform benchmark has about 9.06% increase in execution time with an absence of faults, while Matrix has 10.02% additional execution time. In the faulty cases, SER-3DR requires additional time for detecting and recovery.

With the baseline LAFT-OASIS, we inject similar error rates to study the impact of soft-errors. According to the results, LAFT-OASIS system crashed in every error rates. The system easily falls to deadlock or the router is hang up because of inaccurate arbitration in NPC and SA. Notably, uncompleted faulty LAFT-OASIS in transpose benchmark even cost more time than finished non-faulty LAFT-OASIS. This behavior is explained by mis-routing packets inside network. Obviously, with 0% of error rate, LAFT-OASIS runs correctly.

E. Throughput Evaluation

To perform the throughput evaluation, we also used the above three benchmarks with five injection rates as shown in Figures 9, 10, and 11. For Uniform and Matrix benchmarks, the throughput is slightly degraded due to the short packet length. The Transpose benchmark has a insignificant change in the throughput as shown in Fig. 9. In conclusion, we note that SER-3DR provides a soft-error tolerant solution, even with an error rate of 33.33%.

F. Architecture Comparison

As we can see in the execution time and throughput evaluation, TMR-OASIS made no impact to the system performance due to no additional clock cycle; however, this technique leads to an extremely high area cost (45.20%) and power consumption overhead (18.30%). Our proposal has a slightly impact to system area cost (14.08%), power consumption (5.90%) while supporting similar soft-error resilient ability. The proposed architecture outperforms with short packet-size but mostly insignificant changes for medium and large packet-size.

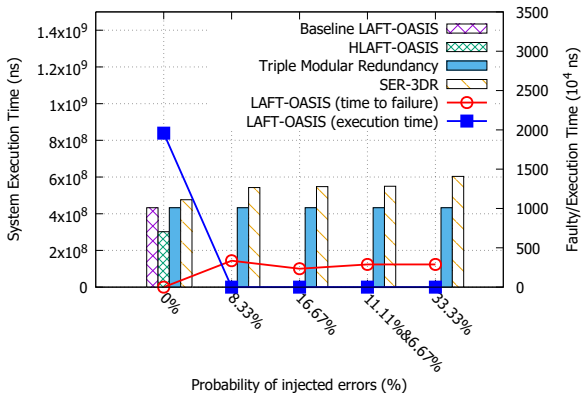

Fig. 8: Matrix Benchmarks: Network size: 72 ($3 \times 6 \times 6$).

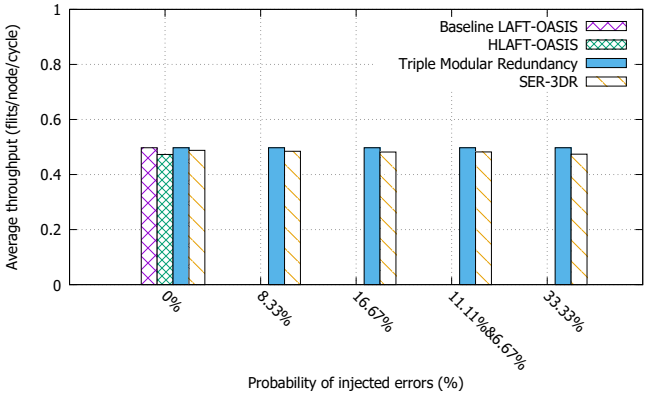

Fig. 9: Transpose Benchmark: Network size: 64 ($4 \times 4 \times 4$).

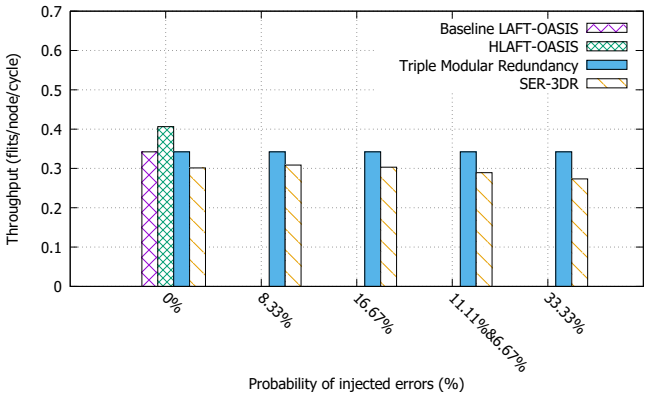

Fig. 10: Uniform Benchmark: Network size: 64 ($4 \times 4 \times 4$).

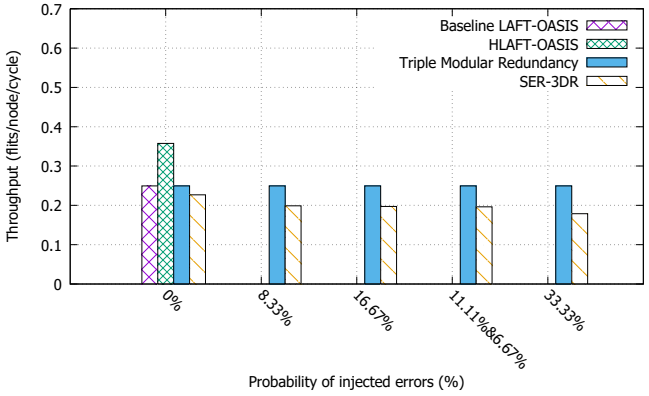

Fig. 11: Matrix Benchmark: Network size: 72 ($3 \times 6 \times 6$).

VI. Conclusion

In this paper, we proposed a soft-error resilient 3D-NoC router (SER-3DR) architecture. The proposed architecture is able to recover from transient errors occurring in different pipeline stages of the SER-3DR. We implemented the architecture in hardware with 45 nm CMOS process. Evaluation results show that SER-3DR is able to achieve a high level of transient error protection with a small latency increase of 18.16%, a power overhead increase of 5.90% and an additional area cost of 14.08% when compared to the baseline router architecture.

As a future work, an in-depth hybrid software-hardware error detection and recovery mechanism will be implemented. In addition, a thermal power study should be conducted to observe how the performance gain obtained with the proposed algorithm would affect this design requirement, as it is very crucial for 3D-Network-on-Chip architectures.

Acknowledgment

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo, Japan, in collaboration with Synopsis, Inc. and Cadence Design Systems, Inc. This project is also supported by Competitive research funding, Ref. UoA-CRF 2014 and P-5 2015, Fukushima, Japan.

The work of Xuan-Tu Tran is partially supported by Nafosted under the project No. 102.01-2013.17.

References

[1] A. B. Abdallah and M. Sowa, "Basic Network-on-Chip Interconnection for Future Gigascale MCSoCs Applications: Communication and Computation Orthogonalization," in *JASSST2006*, 2006.

[2] A. Ben Ahmed, A. Ben Abdallah, and K. Kuroda, "Architecture and design of efficient 3D network-on-chip (3D NoC) for custom multicore SoC," in *International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, pp. 67–73, IEEE, 2010.

[3] A. Ahmed and A. Abdallah, "Low-overhead Routing Algorithm for 3D Network-on-Chip," in *Networking and Computing (ICNC), 2012 Third International Conference on*, pp. 23–32, Dec 2012.

[4] A. B. Ahmed and A. B. Abdallah, "Architecture and design of high-throughput, low-latency, and fault-tolerant routing algorithm for 3D-network-on-chip (3D-NoC)," *The Journal of Supercomputing*, vol. 66, no. 3, pp. 1507–1532, 2013.

[5] A. Ben Ahmed and A. Ben Abdallah, "Graceful deadlock-free fault-tolerant routing algorithm for 3D Network-on-Chip architectures," *Journal of Parallel and Distributed Computing*, vol. 74, no. 4, pp. 2229–2240, 2014.

[6] P. Sivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on soft error rate of combinatorial logic," in *Proc. Intl. Conf. Dependable Sys. & Networks DSN02*, pp. 23–26, 2002.

[7] J. F. Ziegler, "Terrestrial cosmic ray intensities," *IBM Journal of Research and Development*, vol. 42, no. 1, pp. 117–140, 1998.

[8] K. J. Kuhn, "Reducing variation in advanced logic technologies: Approaches to process and design for manufacturability of nanoscale cmos," in *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, pp. 471–474, IEEE, 2007.

[9] T. C. May and M. H. Woods, "Alpha-particle-induced soft errors in dynamic memories," *Electron Devices, IEEE Transactions on*, vol. 26, no. 1, pp. 2–9, 1979.

[10] M.-L. Li, P. Ramachandran, S. K. Sahoo, S. V. Adve, V. S. Adve, and Y. Zhou, "Swat: An error resilient system," *Proceedings of SELSE*, 2008.

[11] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch, "Methods for fault tolerance in networks-on-chip," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, p. 8, 2013.

[12] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: the energy-reliability tradeoff," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 24, pp. 818–831, June 2005.

[13] Q. Yu and P. Ampadu, "Transient and permanent error co-management method for reliable networks-on-chip," in *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pp. 145–154, IEEE, 2010.

[14] S. Shamshiri, A.-A. Ghofrani, and K.-T. Cheng, "End-to-end error correction and online diagnosis for on-chip networks," in *Test Conference (ITC), 2011 IEEE International*, pp. 1–10, IEEE, 2011.

[15] A. Prodromou, A. Panteli, C. Nicopoulos, and Y. Sazeides, "Nocalert: An on-line and real-time fault detection mechanism for network-on-chip architectures," in *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*, pp. 60–71, Dec 2012.

[16] R. Parikh and V. Bertacco, "Formally enhanced runtime verification to ensure noc functional correctness," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-44, (New York, NY, USA), pp. 410–419, ACM, 2011.

[17] Q. Yu and P. Ampadu, *Transient and Permanent Error Control for Networks-on-Chip*. Springer, 2012.

[18] NanGate Inc., "Nangate Open Cell Library 45 nm," *Avaialable: http://www.nangate.com/*, 2014.

[19] NCSU Electronic Design Automation, "FreePDK3D45 3D-IC process design kit," *Avaialable: http://www.eda.ncsu.edu/wiki/FreePDK3D45:Contents*, 2015.