A Low-overhead Fault tolerant Technique for TSV-based Interconnects in 3D-IC Systems

Abderazek Ben Abdallah*, Khanh N. Dang†, and Yuichi Okuyama‡
Adaptive Systems Labratory, Graduate School of Computer Science and Engineering
The University of Aizu, Aizu-Wakamatsu 965-8580, Japan
Email: {*benab, ‡okuyama}@u-aizu.ac.jp, †khanh.n.dang@ieee.org

***Abstract*—3D-Network-on-Chips (3D-NoCs) are considered as one of the most advanced and auspicious communication methodologies for future IC designs by exploiting the benefits of Network-on-Chips (NoCs) and 3D-Integrated Circuits (3D-ICs). However, their reliability still remains a major problem due to the vulnerability of Through Silicon Vias (TSVs). Because most of the existing techniques rely on correcting the TSV defects by using redundancy or employing routing algorithms, the TSV-cluster defect tolerances encounter costly area and power consumption penalties. In order to solve this issue, we propose a highly scalable and low-overhead TSV management for 3D-NoC systems where the TSVs of a router can be utilized by its neighbors and an adaptive online algorithm is also performed to assist. With this proposal, we aim to maintain a graceful performance for 3D-NoCs without the need for redundant links or employing routing algorithms.**

Keywords*—*Through-Silicon-Via, 3D-Network-on-Chip, cluster defect, fault-tolerance.

I. Introduction

In the last decade, the 3D-Network-on-Chip (3D-NoC) paradigm [3], which is a result of the fusion of 3D-Integrated Circuits (3D-ICs)and the mesh-based Network-on-Chips (NoCs) [1], is considered as one of the most promising architectures for IC design. As a consequence, the parallelism and scalability of NoCs can be further enhanced in the third dimension thanks to the short wire length and low power consumption of the Through-Silicon Vias (TSVs), that role the main inter-layer communication mediums. A TSV works as an inter-layer wire in 3D-NoCs. By performing stacking processes, TSVs are established and the two wafers can connect through them. While TSVs bring many advantages for 3D-NoCs, one of their major drawbacks is reliability.

The yield rates of 3D-ICs using TSVs have been considered as a critical factor due to the imperfection of the manufacturing process [14], [16]. In addition, due to the difference between thermal expansion coefficients of the implementation materials [15], 3D-ICs also encounter the stress issue. As reported in [13], the temperature variation between two layers can reach up to 10°C which negatively affects the devices. This can accelerate the *Electromigration*, *Time Dependent Dielectric Breakdown* and *Thermal Cycling*. In summary, TSVs in 3D-ICs are more fault sensitive than the traditional components, not only in the manufacturing phase; but, also during the operation time. Therefore, the future 3D-ICs demand more efficient fault-tolerance solutions.

The major TSV defects can be classified into three types: *Open* (or void), *Bridge*, and *Stuck-at*. Existing works presented so far have dealt with the high fault-rate of TSVs in different approaches: enhance the reliability of TSVs in the manufacturing process [11]; design awareness [15]; recovery the defected TSVs by using circuits [4], using redundancy [9], [12], or *Error Correction Codes* [5]; and using an alternative channel to avoid the defected TSV channel. Although these works have impressively enhanced the reliability of TSV-based systems, they are mostly suitable for random distribution of TSV defects. Unfortunately, the cluster defect distributions [9], [14], [17] are recently considered as the most realistic ones. In order to deal with the cluster TSV defect, most works aim to select a suitable grouping configuration [17] to distribute TSVs on different positions [14] or to enhance the redundancy correction rate [9]. Although these methods can improve the reliability of the system, using extra redundancies and complex arbitration penalty the area cost, wire latency and power consumption. Moreover, if the number of assigned redundant TSV is less than the number of defective TSVs, the vertical connection is not maintained. Therefore, we observe that with low utilization rates of the TSVs has been reported in 3D-NoCs [8], a management solution can efficiently deal with this issue.

This paper proposes a scalable TSV utilization algorithm and architecture to tackle the cluster defect in inter-layer links. To reduce this kind of defect, a router corrects its defected TSV communication by choosing one of its four neighbor TSV-clusters located on the same layer. To ensure timing constraints, in the design stage, we put the TSVs of two nearby routers in between them and a TSV-cluster is only shared between its two neighboring routers. Therefore, the solution can help 3D-NoCs to work around TSV-cluster defects without the need of redundancy. As a consequence, reliability at reasonable overhead is guaranteed. An extended version of this work can be found in [6].

The paper is organized as follows. Section II depicts the motivations and prior works. In Section III, we describe the proposed TSV fault-tolerant architecture. The algorithm and optimizations are later explained in Section IV. Section V shows our evaluation and comparison results. Finally, Section VI concludes the paper.

II. Motivations and Prior Works

The high defect-rate of TSVs negatively affects the final yield. In [10], the authors report that 0.63% of the TSVs are defected which leads the final yield without spare is only 15%. Not only the manufacturing stage, TSVs under operation also face several challenges with stress and thermal issues [15]. These make TSVs are one of the most vulnerable components in 3D-ICs.

However, the TSV failure distribution is still one of the matters that are still under investigation. In general, two main

assumptions for the failure distribution has been introduced: *Random* [12] and *Clustering* distributions [9], [14], [17]. Fortunately, *Random* TSV defect can be efficiently handled by using redundancy and recovery methods; but, *Clustering* defects still remain as a challenging issue. In addition, TSV misalignment [12], which is classified as a cluster defect, also may casually occur due to the inaccuracy in design and manufacturing. Due to the stress and thermal issues, the TSVs may also be defected during operating. On the other hand, *Mean Time To Failure* equations of 3D-ICs when affected by *Time Dependent Dielectric Breakdown*, *Thermal Cycling* and *Electro-migration*, shows an important role of the temperature values. Because of the clustering effect on hot-spot areas in 3D-ICs, this may lead to the TSV-cluster defect.

As we early mention in the introduction, the existing works have approached the TSV fault-tolerance using redundancies [9], [12], supporting circuits [4], coding techniques [5] or alternative channels [3]. Because the cluster defect is predicted to be frequently occurred, the most efficient solution for correcting random defects is grouping and adding redundancy. Unfortunately, they are still inefficient and demand costly extra area cost. Therefore, this paper focuses on fault-tolerance for cluster defect. On the other hand, several works [8] have been reporting the low utilization of the vertical connection using TSVs in 3D-NoC where the authors tried to reduce the number of TSVs to minimize the area overhead and maintain a low degradation in terms of performance. Therefore, we propose in this paper a low cost method for TSV fault-tolerance in 3D-NoCs by exploiting the low utilization rates and the capacity of using alternative channels.

III. Proposed TSV Fault Tolerance Architecture

In this paper, our solution handle the cluster defect by sharing TSVs between neighboring routers. If a TSV-cluster fails, its router find a healthy cluster from one of its neighbors to maintain the connection. Moreover, we also present several design optimizations to improve the reliability of the system (Section IV-B).

A. Fault assumptions

Before we present the system structure, this subsection clarifies the fault assumptions taken in this proposal. We assume there are no random defects. Designers, who also concern about the random defect, can use redundancies to handle it. In this work, we consider an occurred fault makes the whole cluster of TSVs defected. The detection modules, e.g. Built-In-Self-Test module [7], are assumed to be existing and connected to the fault-tolerance module. They are also assumed to be synchronized to exchange the configurations and faulty information.

B. System structure

Fig. 1 depicts a simplified layout example of $3 \times 3 \times 3$ 3D-NoC system using the proposed TSV management. For each vertical connection, each router needs a set of TSVs. Instead of grouping all TSVs together, we divide them into four groups in the design stage. As a result, every router has four TSV-clusters and has a maximum of four nearby TSV-clusters (the number of neighboring clusters is reduced in the borders). If a TSV-cluster of a router is defected, the router selects one of its four neighboring clusters as a replacement. Therefore, the connection can be maintain. In overall, the need for redundancy is not necessary with the low utilization rates of the vertical connections. To ensure the timing constraints, every router has to choose the closest TSV-cluster among its neighbor clusters. Borrowing further TSV-clusters is not considered because of the possible long wires that may create timing violations. Moreover, by structuring a set of four clusters for each router, the system can ensure the scalability of 3D-NoCs. Extending and reducing the size do not involve any significant changes in both architecture and algorithm wise.

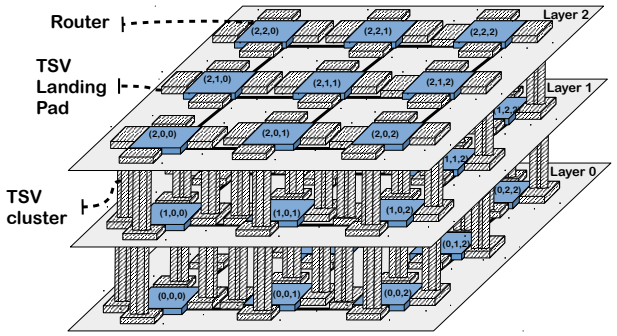

Fig. 1. Simplified block diagram illustrating the proposed system structure.

C. Sharing Circuit Design

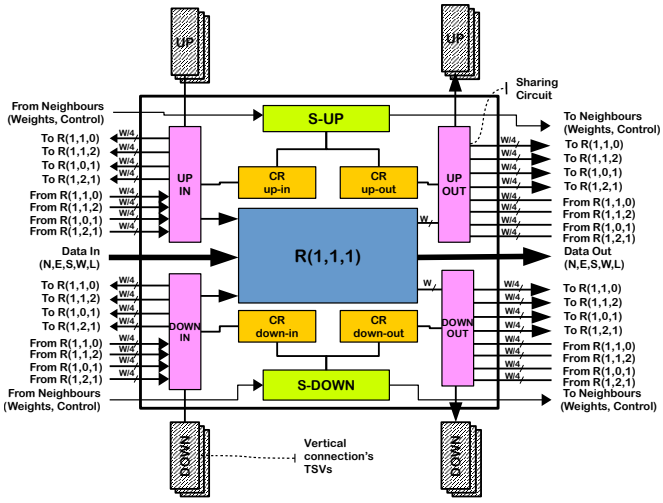

Fig. 2. The TSV fault-tolerance architecture. Red rectangles represent TSVs. *S-UP* and *S-DOWN* are the sharing arbitrators which manage the proposed mechanism. *CR* stands for configuration register and W is the flit width.

In oder to borrow a TSV-cluster from a neighbor, every router needs supporting/arbitrating modules. Fig. 2 shows the wrapper of a 3D-Router. The additional supporting modules that perform the sharing algorithm are later explained in Section IV. There are two identical sharing modules (S-UP and S-DOWN) for the two vertical up and down connections. For each connection, the router has two configuration registers (CR) for the input and output ports. In a 3D-NoC (e.g. Fig. 1), *R(1,1,1)* shares its TSV-clusters with its four neighbors: *R(1,1,0)*, *R(1,1,2)*, *R(1,0,1)*, and *R(1,2,1)*. The input of a TSV-cluster is shared between two routers. The output of a TSV-cluster is also shared the same routers. In the case where this TSV-cluster is defected, or borrowed, router can send its flits by using one of the four neighboring clusters through a sharing process.

As depicted in Fig. 2, the input and output ports can select the data from: (1) its original TSV-cluster, (2) one of its four neighboring clusters or (3) being disconnected. The selection is based on the value of the 6-bit CR. Because the CR only manages the connectivity, its value have to be set carefully to avoid the possible conflict of TSV-cluster usage and to optimize the performance. To this aim, an adaptive sharing algorithm is needed.

IV. Adaptive Online Sharing Algorithm

Algorithm 1: TSV Sharing Algorithm.

```
// Weight values of the current router and its N
   neighbors
Input: Weight_current, Weight_neighbor[1 : N]
// Status of current and neighboring TSV-clusters
Input: TSV_Status_current[1 : N], TSV_Status_neighbor[1 : N]
// Request to link TSV-clusters to neighbors
Output: RQ_link[1 : N]
// Current router status
Output: Router_Status
foreach TSV_Status_current[i] do
    if TSV_Status_current[i] == "NORMAL" then
        // It is a healthy TSV-cluster
        RQ_link[i] = "NULL"
    else
        // It is a faulty or borrowed TSV-cluster
        find c in 1:N with:
        Weight_neighbor[c] < Weight_current
        Weight_neighbor[c] is minimal
        and TSV_Status_neighbor[c] == "NORMAL";
        if (c==NULL) then
            return RQ_link[i] = "NULL"
            return Router_Status = "DISABLE"
        else
            return RQ_link[i] = c
            return Router_Status = "NORMAL"
        end
    end
end
```

In the previous section, we presented how a router can use its nearby TSV-clusters to maintain the connection. In order to deal with the TSV defects, every router need to arbitrate and configure its CR values. We proposed an algorithm of the arbitrating process to perform the mapping online so that the system can react immediately to the newly defected TSV-clusters and can consider the connectivity of the 3D-NoC system.

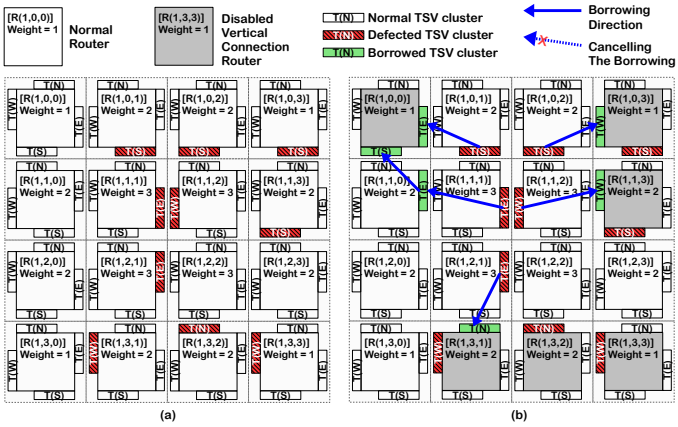

Fig. 3. An example of the sharing algorithm on a 4×4 layer: (a) Initial state with ten defected TSV-clusters; (b) Final result with six disabled routers.

Algorithm 1 shows the proposed algorithm for our sharing mechanism where each router has a weight for each of the vertical connections. This weight represents the connection's priority in sharing/borrowing and can be assigned at the design process or can be updated during operation. At the initial stage, the weights and the TSV-clusters states are exchanged between routers with their neighbors. After knowing the weights and the states, the algorithm performs the mapping process in every router at the same time. If a TSV-cluster is defected, its corresponding router finds from its neighbors a possible candidate by relying on the following conditions:

- The weight of the candidate has to be smaller than the current router.
- The candidate TSV-cluster has to be healthy and not borrowed.
- The weight of the final candidate is the smallest among all the possible candidates.

At the end of the algorithm, the router finds out the possible candidate for borrowing. If no candidates were found, the router's vertical connection is disabled or turned into Serialization (see Section IV-B). If there is a suitable candidate, the router sends a request signal to the owning router to use its TSV-cluster as a replacement for the defected one. The routers having borrowed TSV-clusters also look for a replacement among one of their neighbors. By using a weighted system, the system can avoid race conditions and the disabled TSV-clusters focus on smaller weight routers.

Fig. 3 shows an example of how the sharing algorithm works on a 4×4 layer with ten defected TSV-clusters. Initially, the routers in the center, which are predefined to have higher TSV utilization rates, have higher weights than those at the edges of the network, as depicted in Fig. 3(a). The sharing algorithm selects the best candidates by following the rules previously explained in Algorithm 1. As shown in the above example, the chain of sharing leads to disabling the routers on the edges. The final shape can be seen in Fig. 3(b). Instead of having ten defected TSV-clusters, the algorithm only disables six routers having the lowest weights. This is equivalent to a 40% of reduction from a no fault-tolerant system. Thanks to the weighting system, maintaining the connections of the center routers, which have higher weights and utilize more vertical communications, can reduce the impact of TSV defects in terms of overall performance.

A. Weight adjustment

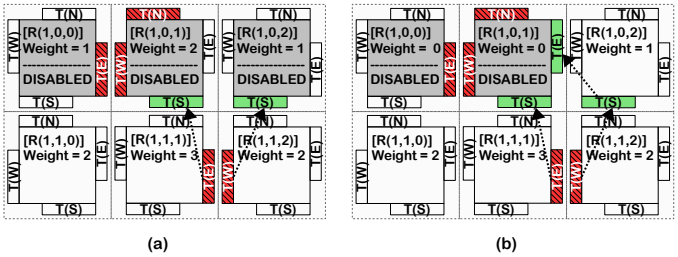

Fig. 4. Example of the weight adjustment performed to disable routers' sharing: (a) Before weight update; (b) After weight update.

After applying the sharing mechanism, the disabled TSV-clusters are usually shifted to areas of low weighted routers. As depicted in Fig. 4, the three routers (*R(1,0,0), R(1,0,1)* and *R(1,0,2)*) are disabled after the sharing process. However,

R(1,0,2) can borrow a TSV-cluster from *R(1,0,1)* to obtain a full connection. However, the weight of *R(1,0,2)* is lower than *R(1,0,1)* which prevents a borrowing process. Therefore, we need to perform an optimization called weigh adjustment.

To perform this optimization, the disabled routers, after the sharing process by Algorithm 1, are taken to a new process. First, every disabled router counts the number of possible TSV-clusters that it can borrow. The possible clusters alsos consist of clusters from a higher weight and disabled routers. Since these routers (*R(1,0,0), R(1,0,1)* and *R(1,0,2)*) are disabled, their TSV-clusters are free to be taken. At the end of this stage, *R(1,0,0), R(1,0,1)* and *R(1,0,2)* have 1, 3, and 1 borrowed/defected TSV-clusters and are able to take 0, 1 and 1 TSV-cluster from their disabled neighbors, respectively. At the second stage, the router checks whether it can take the disabled router's cluster to obtain a full connection. Because *R(1,0,2)* has one borrowed cluster and is able to borrow another one from *R(1,0,1)*, its weight is kept. The other routers (*R(1,0,1)* and *R(1,0,0)* weights are reduced to 0. As a result, *R(1,0,2)* can borrow a TSV cluster from *R(1,0,1)* despite the fact that it has a lower weight at the initial stage. The result is shown in Fig. 4 (b) where *R(1,0,2)* vertical connection is re-enabled.

B. Design optimization

Without adding redundancy, borrowing TSV-clusters to work around the defected ones makes some routers to have less than four accessible clusters. As a result, the communication of these routers have been disabled. To tackle this problem, the naive solution is using a fault-tolerant routing algorithm to re-route the packets to a neighboring router. However, we observe that the disabled routers may have a couple of clusters. Therefore, we implement the *Serialization* technique to help the vertical connection establishing with less than four TSV-clusters.

For the serialization, the router needs at least one TSV-cluster. The serialization mode (1:4 or 1:2) depends on the number of usable clusters. In order to perform serialization, the up and down directions's output of the crossbar is stored in a register. The serialization module transmits flits over the remained clusters within multiple clock cycles.

V. Evaluation Results

A. Evaluation Methodology

The proposed system was designed in Verilog-HDL, synthesized and prototyped with commercial CAD tools. We use NANGATE 45nm library and NCSU FreePDK TSV. The operating voltage is $1.1V$. The TSV's size, pitch, and Keep-out-Zone are $4.06\mu m \times 4.06\mu m$, 10 μm, and 15 μm, respectively. The system use a 7-port router for 3D Mesh-based topologies. The flow-control is Stall-Go and the forwarding mechanism is Wormhole. The input buffer and flit width are 4 and 44, respectively.

First, we evaluate the defect-rate by inserting faults (defects) into TSV-clusters and measure the reliability of the proposed 3D-NoC system. Second, we run both synthetic and realistic traffic patterns as benchmarks to study the performance of the proposed system. We also comparison to the baseline model [2]. Third, we design in hardware and evaluate the complexity of a single 3D router.

B. Defect-rate evaluation

In this section, we provide the impact of the different defect-rates. To demonstrate the scalability of the proposed architecture, we set up several layer sizes: 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64. TSVs are grouped in clusters as presented in Section III. We also use multiples TSV-cluster defect-rates: from 5% to 50%. Because our technique focuses on the cluster defect, random defects are assumed to be dealt with typical redundancy methods. The position of cluster defects are generated randomly and we perform the proposed algorithms with 100K different samples and calculate the average results. We measure the ratio of four types routers in the layer: *Normal* (healthy or corrected), *Serial* (router using serialization) and *Disabled* (disabled routers). We also compare the obtained results with *"Normal w/o FT"* (Normal without Fault-Tolerance), where no fault-tolerance method is used and the router vertical connection having defects is disabled.

As shown in Fig. 5, the system operates without disabling any vertical connections with fault-rates under 50%. Thanks to the *Serialization* techniques, the routers having less than four clusters are still able to work. Even at less than 20% of defect-rate, there are less than 10% of serialization connections in all simulated layer sizes. With 50% of defect-rate and a 2×2 layer size, the disabled router rate is negligible with about 1.565%. This can be easily dealt using a light-weight fault-tolerant routing algorithm. When the layer size increases to be larger than 8×8, the number of disabled connections is mostly insubstantial. At 50% defect-rate, the disabled router ratio is nearly 0.63%, 0.50%, 0.44% and 0.42% with 8×8, 16×16, 32×32, and 64×64 layer sizes, respectively. However, these defect-rates are extremely high; thus, our proposed mechanism can be considered as a highly reliable.

In comparison to the system without fault-tolerant methods, there is a significant improvement in terms of healthy connections, especially at large layer sizes. In Fig. 5, the percentage of routers having four healthy TSV-clusters is represented by the "Normal w/o FT" curve. At 50% defect-rate, the average ratio of normal routers has been improved by 29.83%, 186.26%, 280.76%, 324.42%, 346.74%, and 257.79% for 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 layer sizes, respectively. The improvements are lesser with small layer sizes such as: 2×2 or 4×4. However, thanks to the Serialization, the workable connection rates have nearly reached 100%.

C. Performance Evaluation

The previous section has proved the reliability of the proposed solution. In this section, we evaluate the system performance under TSV-cluster defects. To evaluate the performance of the proposed system and keep fair comparisons to the baseline, we adopted both synthetic and realistic traffic patterns as benchmarks. We selected Transpose, Uniform, Matrix-multiplication, and Hotspot 10% as the synthetic benchmarks. For realistic benchmarks, we chose H.264 video encoding system, Video Object Plane Decoder (VOPD), Picture In Picture (PIP) and Multiple Window Display (MWD). These realistic applications are carefully selected to study the performance of the system. The configurations of these benchmarks are shown in Table I.

1) Latency Evaluation: In this experiment, we evaluate the performance of the proposed architecture in terms of Average packet Latency (APL) over various benchmark programs and defect-rates. The simulation results are shown in Fig. 6 (a). From this graph, we notice that with a 0% of defect-rate, the system's tolerance has similar performance in comparison to the baseline system.

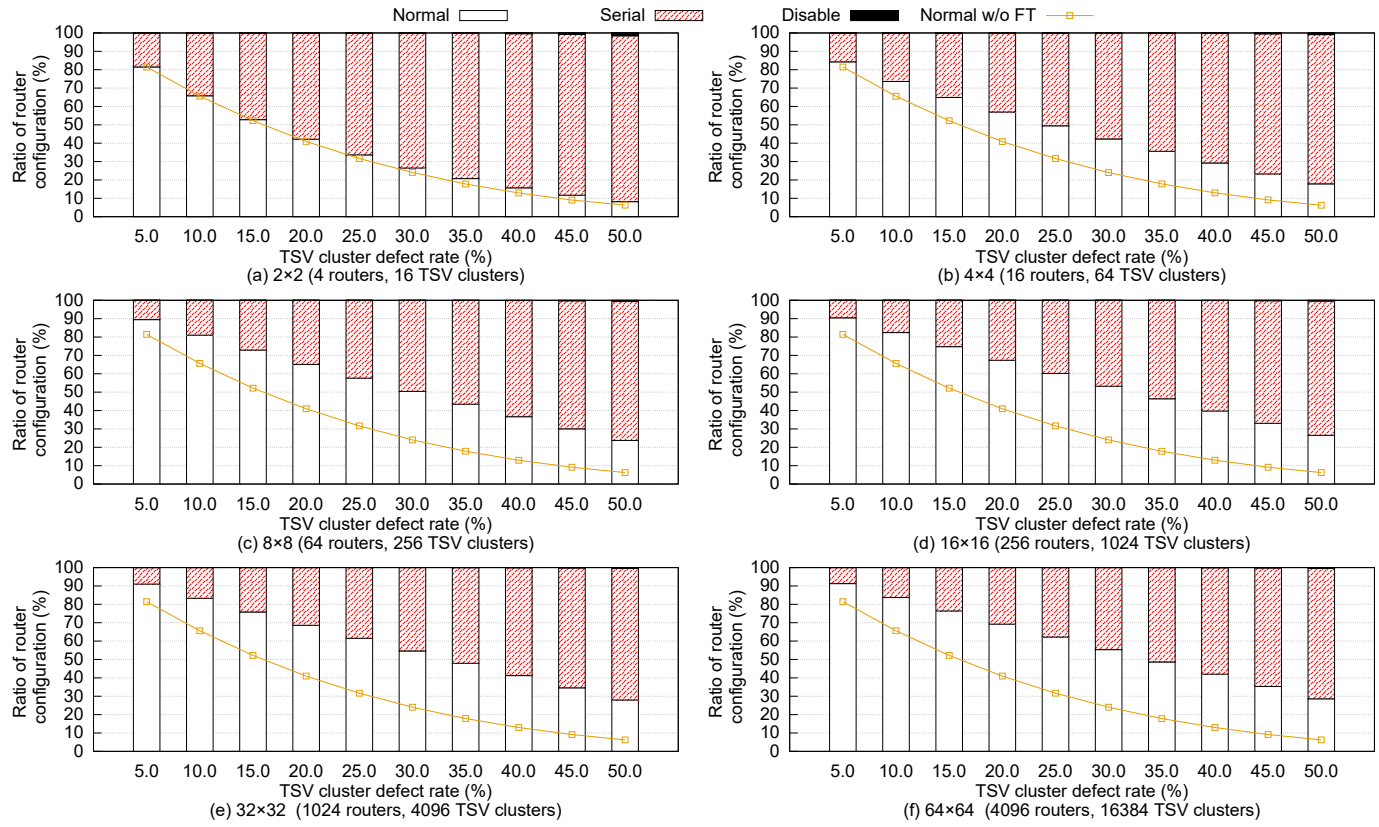

Fig. 5. Defect-rate evaluation: (a) Layer size: 2×2 (4 routers, 16 TSV clusters); (b) Layer size: 4×4 (16 routers, 64 TSV clusters); (c) Layer size: 8×8 (64 routers, 256 TSV clusters); (d) Layer size: 16×16 (256 routers, 1024 TSV clusters); (e) Layer size: 32×32 (1024 routers, 4096 TSV clusters); (f) Layer size: 64×64 (4096 routers, 16384 TSV clusters).

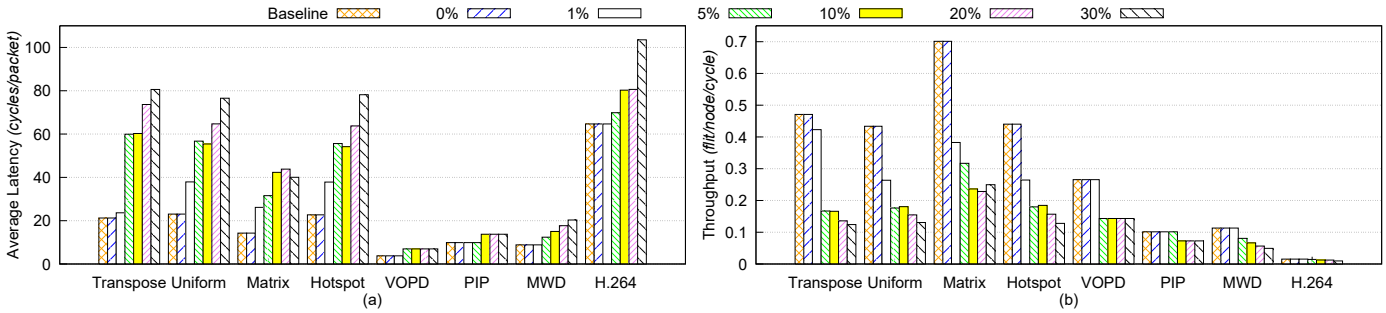

Fig. 6. Evaluation result: (a) Average Packet Latency; (b) Throughput.

TABLE I. SIMULATION CONFIGURATIONS.

Benchmark	**Matrix**	**Transpose**	**Uniform**	**Hotspot**
Network Size (x,y,x)	(6, 6, 3)	(4, 4, 4)	(4, 4, 4)	(4, 4, 4)
#Packets	1,080	640	8,192	8,192
Packet's Size	10	10	10[a]	10
Benchmark	**H.264**	**VOPD**	**MWD**	**PIP**
#Tasks	11	4	8	4
Network Size (x,y,x)	(3, 3, 3)	(3, 2, 2)	(2, 2, 3)	(2, 2, 2)
#Packets	8,400	3,494	1,120	512
Packet's Size	10	10	10	10

[a]For the hot spot nodes, there are additional 10% of flits.

When we increase the defect-rates in the proposed system, it has demonstrated additional impacts on APL. At a 1% fault-rate using Matrix, Uniform, Transpose, and Hotspot 10% benchmarks, the system increases the APL by 83.24%, 64.46%, 11.30% and 66,55%, respectively. These high impacts are due to the occurrence of bottlenecks inside the network caused by *Serialization*. The serialization is already a bottleneck technique. These bottlenecks effects are even higher at a 30% of defect-rate where the APL can be over 3 times that of the 0% case in the synthetic benchmarks.

With H.264, PIP, MWD and VOPD benchmarks, the APL incrementation are significantly reduced due to the low utilization of TSV. We can observe the identical performance of VOPD benchmark from a 1% to a 30% defect-rates. With the PIP benchmark, the system under 1% defect-rate has similar performance to 0% thank to the optimization process which disables the unused clusters. With the MWD and H.264

benchmarks, the impact on APL is gradually increased when increasing the defect-rate. Even at a 30% of defect-rate, the APL values of MWD and H.264 are increased by 129.91% and 60.04%, respectively. Because there is no optimized routing technique for these benchmarks, the bottleneck effect is expected to happen.

2) Throughput Evaluation: Fig. 6 (b) depicts the throughput evaluation with different benchmarks. At 0% defect-rate, the proposed system's throughput is similar to that of the baseline. When defects are injected into the system, we can observe some degradation in throughput caused by the bottleneck effects on the system. Similar to APL, the throughput degradation on realistic traffic benchmarks (VOPD, H.264, MWD and PIP) are significantly better than the synthetic ones. The system at a 20% defect-rate provides a decreased throughput by 71.17%, 64.36%, 67.44%, and 64.37% for Transpose, Uniform, Matrix, and Hotspot 10%, respectively. At the same defect-rate, VOPD, MWD, PIP and H.264 have 46.03%, 50.04% 28.17%, and 19.79% of throughput degradation.

Although there is a considerable degradation in the throughput evaluation, the system still maintains over 0.1 *flit/node/cycle* in the highly stressed benchmarks, even at extremely high defect-rates.

D. Router Hardware Complexity

TABLE II. HARDWARE COMPLEXITY OF A SINGLE ROUTER.

Model		**Area** (μm^2)	**Power** (mW)			**Speed** (Mhz)
			Static	Dynamic	Total	
Baseline router [2]		18,873	5.1229	0.9429	6.0658	925.28
Proposal	Router	29,780	10.017	2.2574	12.3144	613.50
	Serialization	3,318	0.9877	0.2807	1.2684	-
	TSV Sharing	5,740	0.7863	0.2892	1.0300	-
	Total	38,838	11.7910	2.8273	14.6128	537.63

Table II illustrates the hardware complexity results of the proposed router in terms of area, power (static, dynamic, and total), and speed. In comparison to the router in which we implement the proposed techniques, the area and power consumption have increased by 30.42% and 18.66%, respectively. The maximum speed has also slightly decreased by 12.37%. In comparison to the baseline model, the proposed system almost doubles the area cost and power consumption while decreasing the maximum frequency by about 50%. However, the TSV sharing and Serialization modules incur reasonable area and power consumption overheads which are 47.99% and 38.89% in comparison to the baseline router, respectively. Here, the TSV Sharing module handles the sharing algorithm and the Serialization module helps the router communicate in *Serialization* mode.

VI. CONCLUSION AND FUTURE WORK

This paper presented an adaptive and scalable sharing methodology for TSVs in 3D-NoC systems to deal with the TSV-cluster defects. The results have proven the system ability to provide high reliability that can reach up to 346.74% increase in functional routers. Moreover, the proposed approach can correctly work with a reasonable degradation, even under a 30% of defect-rate. The hardware complexity has shown a small overhead in terms of area cost (30.42%), power consumption (18,66%) and maximum frequency (12,37%) of router's logic. Since no TSV redundancy is not required in the proposed architecture and algorithm, we show that it is possible to provide a highly reliable system while maintaining the overhead reasonable.

As future work, the random TSV-defect is also an additional challenge for our 3D-NoC system. Furthermore, degradation factors on 3D-NoCs such as: thermal dissipation, stress, operating voltages should be investigated.

REFERENCES

[1] A. B. Abdallah and M. Sowa, "Basic Network-on-Chip Interconnection for Future Gigascale MCSoCs Applications: Communication and Computation Orthogonalization," in *Proc. of the Symp. on Science, Society, and Technology*, 2006, pp. 1–7.

[2] A. Ben Ahmed and A. Ben Abdallah, "LA-XYZ: low latency, high throughput look-ahead routing algorithm for 3D network-on-chip (3D-NoC) architecture," in *IEEE 6th Int. Symp. on Embedded Multicore Socs*. IEEE, Sep 2012, pp. 167–174.

[3] ——, "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.

[4] M. Cho *et al.*, "Design method and test structure to characterize and repair TSV defect induced signal degradation in 3D system," in *Proc. Int. Conf. on Computer-Aided Design*, 2010, pp. 694–697.

[5] K. N. Dang *et al.*, "A low-overhead soft–hard fault-tolerant architecture, design and management scheme for reliable high-performance many-core 3D-NoC systems," *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2705–2729, 2017.

[6] ——, "Scalable design methodology and online algorithm for tsv-cluster defects recovery in highly reliable 3d-noc systems," *IEEE Transactions on Emerging Topics in Computing*, in press.

[7] Y.-J. Huang and J.-F. Li, "Built-in self-repair scheme for the TSVs in 3-D ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 10, pp. 1600–1613, 2012.

[8] Y. J. Hwang *et al.*, "3D Network-on-Chip system communication using minimum number of TSVs," in *2011 Int. Conf. on ICT Convergence*. IEEE, 2011, pp. 517–522.

[9] L. Jiang, Q. Xu, and B. Eklow, "On effective through-silicon via repair for 3-D-stacked ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst*, vol. 32, no. 4, pp. 559–571, 2013.

[10] U. Kang *et al.*, "8Gb 3D DDR3 DRAM using through-silicon-via technology," in *IEEE Int. Solid-State Circuits Conf.-Dig. of Tech. Papers*. IEEE, 2009, pp. 130–131.

[11] J. U. Knickerbocker *et al.*, "Three-dimensional silicon integration," *IBM J. Research and Development*, vol. 52, no. 6, pp. 553–569, 2008.

[12] I. Loi *et al.*, "A low-overhead fault tolerance scheme for TSV-based 3D network on chip links," in *Proc. 2008 IEEE/ACM Int. Conf. on Computer-Aided Design*, 2008, pp. 598–602.

[13] Y. J. Park *et al.*, "Thermal Analysis for 3D Multi-core Processors with Dynamic Frequency Scaling," in *IEEE/ACIS 9th Int. Conf. on Computer and Information Science*, Aug 2010, pp. 69–74.

[14] F. Ye and K. Chakrabarty, "TSV open defects in 3D integrated circuits: Characterization, test, and optimal spare allocation," in *Proc. 49th Annual Design Automation Conf.* ACM, 2012, pp. 1024–1030.

[15] T. Zhang *et al.*, "Temperature-aware routing in 3D ICs," in *Asia and South Pacific Conf. on Design Automation*, Jan 2006, pp. 309–314.

[16] J. Zhao *et al.*, "Overview of 3D Architecture Design Opportunities and Techniques," *IEEE Des. Test.*, vol. PP, no. 99, pp. 2168–2356, Jul 2015.

[17] Y. Zhao *et al.*, "Cost-effective TSV grouping for yield improvement of 3D-ICs," in *Asian Test Symp.* IEEE, 2011, pp. 201–206.