IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A thermal-aware on-line fault tolerance method for TSV lifetime reliability in 3D-NoC systems

KHANH N. DANG[1], AKRAM BEN AHMED[2], (MEMBER, IEEE), ABDERAZEK BEN ABDALLAH[3] AND XUAN-TU TRAN[1], (SENIOR MEMBER, IEEE)

[1]VNU Key Laboratory for Smart Integrated Systems (SISLAB), VNU University of Engineering and Technology (VNU-UET), Vietnam National University, Hanoi (VNU), Hanoi 123106, Vietnam
[2]National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, 305-8568, Japan
[3]Adaptive Systems Laboratory, The University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan.

Corresponding authors: Khanh N. Dang (e-mail: khanh.n.dang@vnu.edu.vn) and Xuan-Tu Tran (e-mail: tutx@vnu.edu.vn).

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.312.

ABSTRACT

Through-Silicon-Via (TSV) based 3D Integrated Circuits (3D-IC) are one of the most advanced architectures by providing low power consumption, shorter wire length and smaller footprint. However, 3D-ICs confront lifetime reliability due to high operating temperature and interconnect reliability, especially the Through-Silicon-Via (TSV), which can significantly affect the accuracy of the applications. In this paper, we present an online method that supports the detection and correction of lifetime TSV failures, named IaSiG. By reusing the conventional recovery method and analyzing the output syndromes, IaSiG can determine and correct the defective TSVs. Results show that within a group, R redundant TSVs can fully localize and correct R defects and support the detection of $R+1$ defects. Moreover, by using G groups, it can localize up to $G \times R$ and detect up to $G \times (R+1)$ defects. An implementation of IaSiG for 32-bit data in eight groups and two redundancies has a worst-case execution time (WCET) of 5,152 cycles while supporting at most 16 defective TSVs (50% localization). By integrating IaSiG onto a 3D Network-on-Chip, we also perform a grid-search based empirical method to insert suitable numbers of redundancies into TSV groups. The empirical method takes the operating temperature as the factor of accelerated fault due to the fact that temperature is one of the major issues of 3D-ICs. The results show that the proposed method can reduce the number of redundancies from the uniform method while still maintaining the required Mean Time to Failure.

INDEX TERMS Fault-Tolerance, Fault Detection, Parity Check, Through Silicon Via, Real-time, Thermal aware

I. INTRODUCTION

Serving as vertical wires between two adjacent layers in Three Dimensional Integrated Circuits (3D-ICs), Through-Silicon-Vias (TSVs) offer extremely short lengths and low latency, which could bring high speeds and lower power to inter-chip communication [1]–[3]. TSV-based 3D-ICs also have smaller footprints despite the TSV's overheads [4] due to the three dimensional structure.

However, one of the major concerns of TSVs is reliability due to their low yield rates [5], vulnerability to thermal and stress, and the crosstalk issues of parallel TSVs [6], [7]. A single defective TSV in the manufacturing phase can corrupt the connection between two layers or even separate the system in parts. Therefore, identifying and correcting the faulty TSVs is necessary to improve the overall yield rate. On the other hand, by having higher operating temperature and high temperature differences between layers [8], the thermal and stress impacts on 3D-ICs reliability are also critical, which can shorten the lifetime expectation. Consequently, there is an imperative need to not only improve the yield rate at the manufacturing phase but also on lifetime reliability due to the vulnerability to the thermal and stress of TSVs. To solve these issues, researchers have been focusing on dealing with TSV faults in several phases, such as pre-bond, post-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

bond, and post-manufacture and different aspects: detection, recovery, online, or offline.

To localize the defective position, most systems use Built-in-self-test (BIST) [9], external testing [10], [11] and online testing [12], [13]. The mentioned works mostly focused on issuing test patterns and capturing them on a different terminal of a TSV to identify their healthiness. To tolerate defects, three main approaches: *(i)* hardware fault-tolerance circuits [14], redundancies [15]–[17], or reliability mapping [7], [18]; *(ii)* information redundancy such as coding techniques [19], [20] or modular redundancies [21]; or *(iii)* algorithm-based fault-tolerance [22], [23]. While the hardware fault-tolerance circuit [14] uses a specific design to analyze the output of a TSV to identify its status and correct (i.e., raising or lowering the output voltage), the redundancy-based approaches [15]–[17] use spare TSVs to handle the tasks of the faulty TSVs. On the other hand, the reliability mapping approach tried to analyze the potential critical issues and optimize the design for a certain requirement. The information redundancy methods deal with the defective TSVs as flipped bits where they use coding techniques to detect and correct the corruptions in transmitted bits. As we can observe that both hardware and information redundancies keep the faulty TSV group and provide a correction method; however, there are cases where the fault rates are higher than the limitation of these approaches. Therefore, algorithm-based fault-tolerance [22], [23] methods, which provide an alternative way to execute the system, can be useful. This type of approach can help the system maintain its reliability under a reasonable trade-off on performance. Although commercial CAD tools and existing solutions have become mature for defect localization and detection, having an online, non-blocking, and low-cost solution helps preventing expensive consequences of operating systems under faults.

Despite having numerous methods to solve the reliability issues of TSVs, they mostly focus on offline testing and recovery in three phases: pre-bond, post-bond and post-production. However, the high operating temperature is one of the critical issues of 3D-IC [8]. The fault rates are expected to exponentially accelerate with the operating temperatures in most academic and industry models [24], [25]. TSVs made out of Copper also have higher activation energy than general CMOS, which makes their fault rate further accelerated with the temperature. By having higher temperatures than conventional 2-D ICs and high temperature differences between layers, life-time reliability is one of the critical issues for 3D ICs.

From the design perspective, having a light-weight and graceful-degradation localization and recovery method is necessary for TSV-based 3D-ICs. However, there are some issues that motivate our method to solve:

1) To ensure the real-timeliness, *fault detection and recovery need to be completed on a timely basis with acceptable execution time extension* [26]. In other words, the system must respond to new faults (as an event) before a certain deadline. The existing works using BIST [9] or external testing [10], [11] periodically can offer a high coverage; however, they cannot satisfy these requirements as they need an enormous amount of cycles to complete and can also block the communication, which degrades the overall performance.
2) A possible approach that can help the system to operate in real-time while ensuring the quality of connections is to use ECCs [19]. However, *ECCs are usually limited by the number of detectable and correctable faults*, which is inefficient to clustering defects [15] in TSVs.
3) The integration of recovery is not well considered in most TSV localization works, especially for online and lifetime reliability. Most works deal with defect localization and recovery separately. Meanwhile, *safety-critical systems require high availability, which needs self-correction on the fly*.
4) While adding redundancies [16], [17] for manufacturing defects considers the fault-rate as a uniform distribution, lifetime reliability of TSVs is heavily affected by the operating temperature [24], [25]. Since the temperature is also shown with a non-uniform distribution in 3D-ICs [8], *designers must consider how we group TSVs to improve system reliability.*

Motivated by the above problems, TSV-based 3D-ICs desire to have a method to monitor, detect, localize, and recover from TSV defects. In [27], an on-communication test (OCT) method was previously presented by testing the communication medium along with its operations. The main idea in the aforementioned work is to use Error Correction/Detection Code as the baseline and improve its detectability and correctability using augmented algorithms. On-communication test ing only offers non degradation; but, also short response time for TSV. Compared to periodic test [15], OCT executes and finishes after a new fault occurred, which guarantees the real-time requirements. In this paper, we propose a novel method named **I**solation **a**nd **S**hift **i**n **G**roup (IaSiG), which is not only an OCT method, it also provides recovery right after its execution. This work is based on our preliminary work in [27] [1] with the additional new contributions as follows:

- A low response time on-communication test (OCT) algorithm to isolate and check the possible defects in a group of TSVs. Double-check mechanism is also integrated to reduce the probability of hidden defects.
- The proposed OCT reuses the unused spare TSVs for the testing purpose. Therefore, the proposal utilizes all the redundancies for both detection and recovery.
- A grouping algorithm to help reduce the test time and increase the detectability and localizability of the isolation and check algorithm.
- Integration of the proposed method onto a 3D-Network-on-Chip (3D-NoC) and performance evaluation. An empirical redundancy insertion also helps to reduce the

[1] [27] K. N. Dang et al., "TSV-IaS: Analytic analysis and low-cost non-preemptive on-line detection and correction method for TSV defects,"in Proc. IEEE Computer Society Annual Symp. VLSI (ISVLSI), 2019, pp. 501–506.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

number of redundancies while maintaining the required reliability.

In summary, this paper aims to provide an online and non-blocking method for detecting and correcting failures in TSVs. We also investigate the impact of operating temperatures and adapt the number of redundancies to that. Along with mathematical analyses, this paper provides a comprehensive platform for dealing with the lifetime reliability of TSVs and could be widely used. We also demonstrate the efficiency of the proposed methodology using a 3D-NoC under PARSEC benchmarks. The organization of this paper is as follows: Section II presents the related work. In Section III, we introduce the proposal and Section IV is dedicated to illustrate the evaluation experiments and findings. Finally, Section V concludes the paper.

II. RELATED WORKS

In this section, we present the related works on ECC methods, TSV testing, TSV recovery and scheduling approach.

Since parity calculation requires only XOR gates, it has been the backbone of most low-cost error detection and correction codes. Hamming code and its extension SECDED (Single Error Correction Double Error Detection) [19] are also two common methods based on parity-check. The delay and area complexity of these methods are only $O(n)$ and $O(log_2 n)$, which make them more suitable for the encoding and decoding scheme for high-speed TSV links. To correct more faults, Orthogonal Latin Square Code is also another option for TSV with low cost and modular design [28]. In [29], the authors presented a method named SEC-DAEC (Single Error Correction Double Adjacent Error Correction) to correct not only a single flipped bit but also two adjacent flipped bits.

In this proposal, we adopt the parity check, which can detect one fault. Therefore, the proposed technique can be integrated into any ECC scheme that is based on the parity check.

To correct faulty TSVs, there are two major approaches: (1) correction circuits [14] such as using voltage comparator to detect and correct open defects (2) double [30] or shared spare TSVs [16], [17] to replace the faulty ones. While correction circuit is low cost, it is limited in terms of correctability. Therefore, recent researches tent to focus on spare TSVs instead. There are four major methods of shared spare TSV recovery: (a) shifting [31], (b) switching [16], (c) crossbar [17] and (d) network [15]. To redirect the TSV signal to the spare ones, a multiplexer/de-multiplexer [16], [17], [31] or tri-state gate [23] could be used. Double-TSV [30] is another method using redundancy; but, it is not cost efficient. In [32], the authors approached the redundancy placement using cobweb-like shapes. The redundancies are placed at the border of the cobweb and the signals of failed TSVs are shifted outwardly in chains. Another shape of the TSV group could be a honeycomb [33] that provides lower effect on area than conventional 2D Mesh while still maintaining the scalability feature. By combining a single or several honeycombs shaped groups of TSVs with the time-division multiplexing technique, the approach in [33] can even lower the cost of the design. Although alternating different shapes of TSV groups can end up with a better result, the 2D-Mesh-like placement still fits best the ASIC design. *Park et al.* [34] introduce the TSV set structure and redundancy re-usage to not only detect and correct TSV faults, but can deal with the intervention of soft errors during online testing/repairing. One thing we can easily notice is most of the above methods provide detection/recovery using spare TSVs regardless of the TSV position or the repairing structure. By considering them as the optimization problem, work in [18] use Integer Linear Programming to find the optimal value for spare and functional TSV positions and the group structure.

The built-in Error Detection/Correction Code [19] (EDC/ECC) could help detect and locate faults in TSVs as normal wires. However, in [35], the inconsistent behavior between TSVs and wires was discussed unveiling that the flipped bit is not consistent in TSVs. On the other hand, despite the immediate response time, ECC/DEC can localize and detect a certain number of defects. Also, additional bits require extra TSVs, which are costly since the size of TSV is significant.

Another common method is to use testing circuits or BISTs. Works in [11], [12], [14] depicted a more fine-grained method, which could detect open defects using a simple circuit. Depending on the level of defects, they can even provide the recovery method. The grouping method allowing column and row check is presented in [36]. This method supports open, short and bridge defects and could reduce the testing time from $O(n)$ to $O(\sqrt{n})$. For online testing, injected test patterns to the TSV can be captured at the output and analyzed to find open defects using a NAND gate with logic threshold voltage [12]. *Serafy et al.* [13] presented a lifetime reliability using a resistance tracking method and *BIST* to overcome the aging in TSVs. In [17], the authors proposed a test access point for injecting and collecting test vectors while *Van der Plas et al.* [6] used a test pattern generator to test open TSV defects. In [9], other methods of TSV's *BIST* for pin-hole and void defects are also presented.

One thing in common between these types of tests (BISTs and dedicated circuits) is that they require system interruption (partly or totally) when they detach tested devices/modules from the system. This is not affordable in real-time and safety-critical applications, as we previously discussed.

A. TEST SCHEDULING

Since naively allowing to run BISTs is not preferable because of the costly execution they demand, scheduling this type of tests based on period is more suitable. This method is called *Periodic BIST* (P-BIST) that activates *BIST* periodically. Here, we focus mainly on Network-on-Chip testing as the scope of this work. Works in [37] and [38] activates periodically their testing circuit; however, they only execute during the free time slots to avoid conflicts between data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

traffic and test traffic. The tested routers still maintain their functions as usual. For the detached core, they also provide alternative connections during the test time. *Huang et al.* [39] also presented another non-blocking testing for Network-on-Chips. A testing for NoC fabrics is presented in [40], which can be applied for 3D-NoCs, uses dedicated test data and structure. The consensus of these methods is to smartly schedule in order to avoid creating congestion/conflict on the system to reduce the performance degradation. Because their experiments are limited in terms of size, more complicated systems might lead to costly degradation.

III. PROPOSED ARCHITECTURE AND ALGORITHM

This section first shows the TSV organization. Then, we overview the Isolation and Shift (IaS) algorithm, which we enhance to a group-based method named Isolation and Shift in Group (IaSiG). Furthermore, we provide mathematical analyses for the proposal. Later, we present the integration of the method onto the 3D Network-on-Chip to support detection and correction faults. In this work, we consider the on-line monitoring and correcting for TSVs. We assume that the manufacturing test has already been performed and the system can correct TSVs if needed.

A. ISOLATION AND SHIFT (IAS) OVERVIEW

This section provides an overview about Isolation and Shift (IaS), which is our preliminary work in [27]. We later discuss the advantages and drawbacks of the *IaS* technique.

1) TSV Organization

In this work, we adopt the TSVs as one-dimensional arrays and the shifting mechanism for recovery [31]. We also note that our technique is general and can be applied for others organization and recovery methods. Here, we assume a TSV group of M original TSVs and R spare TSVs. We symbolized each TSV as t_i where i is the index of the TSV. Also, the input signals for TSVs are organized as a similar one-dimension array and symbolized as s_i.

For a set of M bits data, we use parity to check correctness of the data. This leads to $M+1$ bit codeword and the TSV set now has a size of $M + R + 1$. Depending on the ECC/EDC, the number of TSVs may vary.

2) TSV model

Despite having parity check, there is no guarantee that we can detect odd number of defects. The main reason is the inconsistent behavior of defective TSVs [35]. For instance, with short-to-substrate defects, the defect in a TSV could be hidden when it transmits value '0', which is at $0V$. Here, we adopt TSV defect models from [35] as follows:

- **Short-to-substrate**: the value of TSV is stuck at '0'.
- **Open**: a certain latency, which is added to slow down the transition of TSV, delays the value of TSV by one clock cycle.
- **Bridge**: Two or more TSVs are shorted, which prevents them from having different values. If all of the bridged TSVs send the same logic value, the output is corrected. However, if they are different, the output follows the majority voting. If the numbers of '0' and '1' are equal, the output is randomly assigned as a post-metastability result.

3) Mechanism

Isolation and Shift is based on two phases. The isolation phase is performed by considering the isolated TSV as unusable. Then the system shifts its signal to utilize the spare TSV to support communication. For instance, if the TSV with index f is isolated, signals $s_{i\geq f}$ are routed to TSVs t_{i+1}.

By isolating each TSV in the group, the decoder can remove each TSV from the communication. It then can decide whether TSVs have defects based on the syndrome of Parity-check. If the system finds out that isolating t_f gives non-faulty output while using the actual t_f TSV gives faulty outputs, t_f is determined as faulty. Let $S_{(f=i)}$ be the output (syndrome) of the decoder while isolating t_i:

$$t_i = \begin{cases} \text{faulty} & \text{if } S_{(f=i)} = 0 \text{ and } S_{(\forall f!=i)} \neq 0 \\ \text{healthy} & \text{if } S_{(f=i)} \neq 0 \text{ and } S_{(\exists f!=i)} \neq 0 \end{cases} \quad (1)$$

The detection process is based on statistics, which collect the faulty behavior (failed the parity check) for a certain number of cycles. Because the TSV defect is inconsistent, it can take several cycles to detect the faulty cases. We define a healthy set of TSVs is to have less than T faulty outputs after K transmissions ($K > 2$). To distinguish defects from soft errors, the threshold T could be set to 1; otherwise, $T = 0$. Here, we define the syndrome S as:

$$S = \begin{cases} 0 & \text{if less than T fault after K transmissions} \\ 1 & \text{if T+ faults after K transmissions} \end{cases} \quad (2)$$

A demonstration of multiple faults detection is shown in Fig 1. This illustration of using shifting can help to detect and correct two faults by using Parity-check. The proof of using Parity-check to detect multiple faults is shown in Lemma III.1 at the Section III-D.

Table 1: Detection results with Monte-Carlo simulation with random data.

M	R	K	# hidden defects	detection rate
5	1	8	352	0.9648
5	1	16	2	0.9998
5	1	32	0	1.00
5	2	8	1828	0.8172
5	2	16	4	0.9996
5	2	32	0	1.00
9	1	8	679	0.9321
9	1	16	3	0.9998
9	1	32	0	1.00
9	2	8	3921	0.9321
9	2	16	16	0.9998
9	2	32	0	1.00

Table 1 shows the Monte-Carlo simulation with 10,000 cases to verify our solution with the TSV model. We can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

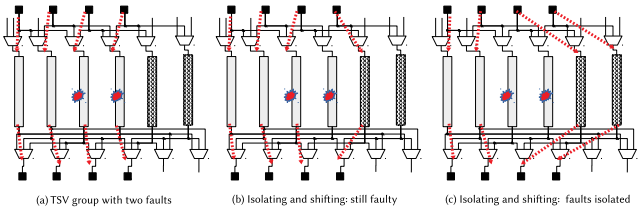

Figure 1: Illustration of shifting method with two spare TSVs. The parity check is used to detect and localize the fault. Red dotted arrow is the configuration to isolate and localize the fault.

observe that with *K* smaller than 32, there is a certain number of hidden defects. However, under the random data case, it can detect with *K=32*. However, as analyzed in Lemma III.1, there is still a chance of having hidden defects. Therefore, in Algorithm 1, we use double-check to improve the correctness of the decision.

4) Algorithm

Algorithm 1 shows the IaS (Isolation and Shift) algorithm for detecting and localizing faults in a TSV group. It starts without isolating any TSVs and keeps calculating the parity check of incoming flits. Once an error occurs, the system starts isolating and checking. Without that, it keeps checking until meeting errors (see line 1-2 of Algorithm 1). Because of this, redundant TSVs are not used frequently, which could avoid aging in those TSVs. Note that we assume that the redundant TSVs are always healthy.

In comparison to the previous work [27], the major improvement of this algorithm is to use double-check. As shown in Table 1 and Lemma III.1, there is a chance of hidden defects. To avoid this scenario, we use double check to naturally increase the value of *K*. When *IaS* detects the potential case, it runs one additional *K* transmission instead of concluding immediately.

Once an error is hit, the system isolates one TSV and uses redundant ones to handle the communication. The parity check will find whether there is a faulty output. If the non-isolated is faulty ($S_{(f=NULL)} = 1$) and the isolated case is non-faulty ($S_{(f=i)} = 0$), the system indicates the faulty positions are the isolated ones. To avoid silent errors, the system performs double check: once a satisfied case is met after K transmissions, it keeps checking for extra K (double-check) transmissions to ensure the correctness (lines 6-12). Note that the system scans through all TSVs in order to isolate them. At the end of the scan, the system checks whether it finds out the faulty positions. If not, it indicates that more than R defects occurred and marks the whole group of TSVs as faulty.

To support real-time applications, fully hardware architecture is used and no connections are blocked during testing. In

Algorithm 1: IaS Algorithm.

```
   Input: S;                          // output of detector
   Output: Fault_Idx;                 // fault indexes
1  while (S == 0) do
2      use M+1 TSVs for communication.
3  foreach i in 1:R do
4      while no case left do
5          isolate i TSVs
6          if S_(f=i) = 0 and S_(f≠i) = 1 then
               // Double-check
7              run for extra K transmissions;
8              if S_(f=i) = 0 then
                   // Faults are localized
9                  Fault_Idx[all i TSVs] = 1;
10                 match = 1;
11                 break;
12             else
13                 next isolation;
14 if (match==0) then
       // More than R defects
15     Fault_Idx = 2^(M+1) − 1;
       /* mark all TSVs as faulty          */
16 return Fault_Idx;
```

terms of checking time, the proposal needs

$$K \times (\sum_{r=1}^{R} \binom{M+R}{r} + 1) \tag{3}$$

transmissions (cycles) to complete its loops. This is also the maximum response time to newly occurred faults or the *worst-case-execution-time* (WCET) of the algorithm.

In [31], the authors proposed testing TSVs by transmitting two values '0' and '1'. A test generator is also used in [12], [17]. Here, once TSVs are isolated, they could be tested using a dedicated tester.

5) Architecture

Figure 2 shows the simplified architecture for IaS with $[M = 4, R = 1]$ TSVs. The input data width is $M - 1$ and the encoded data width is M for parity check. Note that the system can be adopted with another ECC, which has a different coding rate. The encoded data is shifted with the configuration from "TSV-Fuse", as represented in Figure 2. This box receives the isolated TSV value ($isol\_TSV$) from the controller.

At the bottom layer, the output of data from TSVs is unshifted using a corresponding configuration from "TSV-Fuse". The unshifted data is checked with ECC to find possible data corruption. The parity check is sent to the controller to monitor the case. After looping all cases, the controller decides what is the proper configuration, which also means the faulty indexes.

To communicate between two layers, we use a synchronization TSV (s_TSV) for matching the operations of the two controllers. The data is transferred via functional TSVs (f_TSV) with the help of redundant TSVs (r_TSV). Note that,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

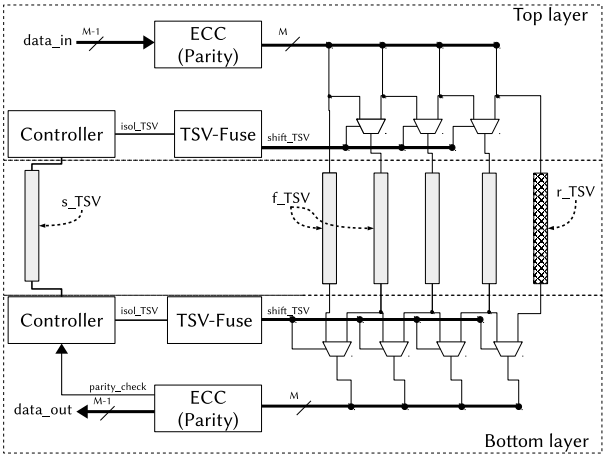

Figure 2: IaS architecture for $M = 4$, $R = 1$ [27]. *s_TSV*: synchronization TSV; *f_TSV*: functional TSV; *r_TSV*: redundant TSV.

in order to perform our algorithm, s_TSV must be healthy. To protect it, several methods could be used such as majority voting (3 TSVs for voting) and Double-TSVs. Note that our previous work [27] does not provides double-checking feature; therefore, s_TSV is not needed.

6) Discussion

Despite having a certain WCET and providing non-blocking testing, IaS still has two major drawbacks:

1) The testing time is not scaling well with IaS. With higher M and R values, the testing time can become enormous increasing the WCET. For instance, with $M = 32$ and $R = 2$, the WCET of IaS is over 17,000 cycles (see Figure 4).
2) With large M values there are more than R defects in a group, marking the whole group not efficient for both detection and recovery. Here, providing a certain set of defective TSVs can offer more efficiency.

In order to solve the above two issues, we present hereafter the group-based test, named IaSiG.

B. PROPOSED ISOLATION AND SHIFT IN GROUP

This section presents Isolation and Shift in Group (IaSiG) algorithm (see Algorithm 2) that works based on IaS. Assuming the system has M functional TSVs and R spare TSVs. Instead of testing M with R, we divide them into G groups of C ($C < M$) TSVs where $M = \sum_{i=0}^{G-1} C_i$. Note that we can even use heterogeneous clustering (groups with different C_i values). The current testing cluster has access to R spare TSVs and IaSiG uses IaS to test each group.

Note that by dividing into groups, each IaS of a group can locate R and detect more than R defects in a group. Therefore, IaSiG can locate $G*R$ and detect more than $G*R$ defects. If a whole group is defected, IaSiG could mark it as faulty while still considering other groups as healthy. This solves the second issue of IaS. As shown in Algorithm 2,

Algorithm 2: IaSiG algorithm.

```
Input: S;                          // output of detector
Output: Fault_Idx;                 // fault indexes
1 while (S == 0) do
2     use M+1 TSVs for communication.
3 foreach group do
4     if (S == 1 after K transmissions) then
          // Run IaS for a group
5         Group_Fault_Idx = IaS(S);
6         Fault_Idx = Fault_Idx + Group_Fault_Idx ;
7     else
8         next group;
9 return Fault_Idx;
```

the system first runs parity check for the whole TSVs (M TSVs) until faults are detected (S==1). Then, it runs a one-by-one clustering check by attaching R redundant TSVs to C_i functional TSVs. Once a clustering test is hit with a fault (S==1), Isolation and Shift (IaS) is used to find the defective TSVs. Otherwise, it switches to test a different cluster. If there are faults, the execution time is:

$$\text{WCET} = \text{WCET(group)} + \text{WCET(IaS)}$$

$$\text{WCET} = \begin{cases} G * K + (\sum_{r=1}^{R} \binom{C_{G-1}+R}{r} + 1) * K, & \text{if } \textbf{case-1} \\ \sum_{g=0}^{G-1} [1 + (\sum_{r=1}^{R} \binom{C_{G-1}+R}{r} + 1) * K] & \text{if } \textbf{case-2} \end{cases} \quad (4)$$

where:

- **case-1:** There are at R+ defects in every group, which makes the Algorithm 2 jumps into group every time.
- **case-2:** There are at R+ defects in one group, which makes the Algorithm 2 jumps into group one time. This is similar to the number of faults in IaS.

The WCETs are substantially smaller than normal IaS as shown in Fig. 4. As we can observe, with $R = 2$, the WCET of IaS becomes higher than 20,000 cycles with 32+ bit; however, IaSiG stays under 20,000 cycles even with 128-bit. By diving into groups, we now can tackle the first issue of IaS: testing time scalability.

Fig. 4 also shows the case where the number of TSVs per group is equal to the number of redundant TSVs. In this case, the testing time is increased; however, the system obtains the best granularity where no false positive cases occurs. In terms of detection rate, the successful rate of *IaSiG* is:

$$P_{\text{succesful-detection}} \simeq 1 - \left[(1/2)^{2K} \times \sum_{r=0}^{R} \binom{M+R}{r} \right] \quad (5)$$

The proof of the above equation can be found in Lemma III.1 of Section III-D. Apparently, increasing the value of K can help increase the detection rate; however, the WCET is increased. Therefore, designers should consider this trade-off in the design phase. If we consider the successful detection rate is sustainable for detection, the proposed method can guarantee the detection of $R + 1$ and the correction of R

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

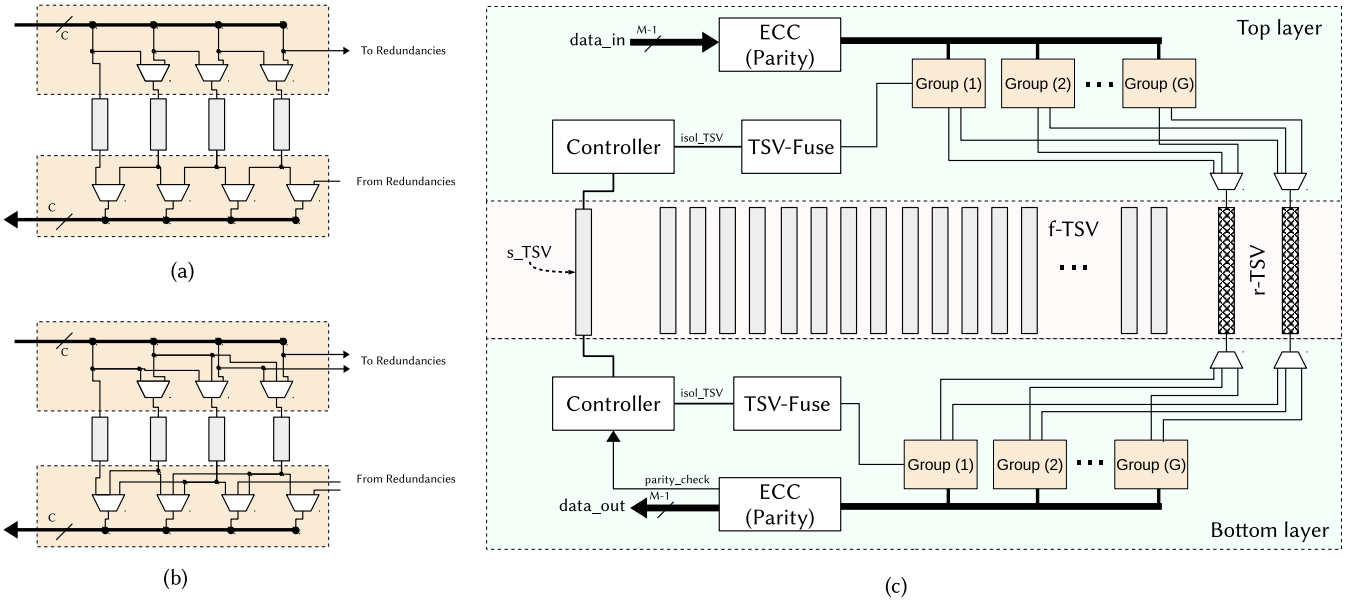

Figure 3: Proposed architecture for group: (a) Group with one redundancy; (b) Group with two redundancies; (c) Overall architecture. *s_TSV*: synchronization TSV; *f_TSV*: functional TSV; *r_TSV*: redundant TSV.

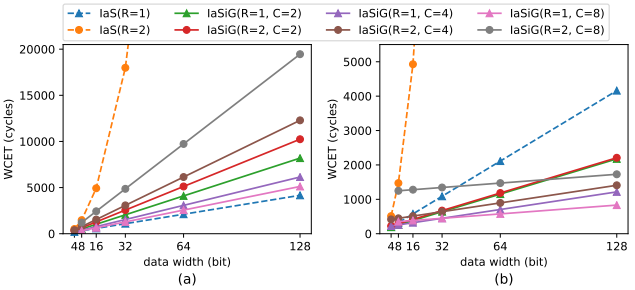

Figure 4: Comparison of worst case execution time (WCET) between IaS and IaSiG with K=32: (a) case-1; (b) case-2.

faults. The upper bound of detection and localization are $G(R+1)$ and GR, respectively. The proof of these bounds can be found in the Lemma III.2 of Section III-D.

C. ARCHITECTURE

Figure 3 illustrates the design for IaSiG where Figure 3 (a) and (b) shows the configuration for one and two redundant TSVs. For more redundant TSVs, we simply add multiplexers/demultiplexers for connecting selections. Here, we support shifting using multiplexers, which is similar to the TSV recovery method in [31].

The architecture of IaSiG is similar to IaS where they share the ECC (parity check), "TSV-Fuse" blocks , synchronization TSV (*s_TSV*) as in Figure 3. Because the redundant TSVs (*r-TSVs*) are shared between groups, we add multiplexers and demultiplexers to support switching groups. The controller block gives instructions for switching between group/redundant TSVs and synchronization.

1) Network-on-Chip Integration

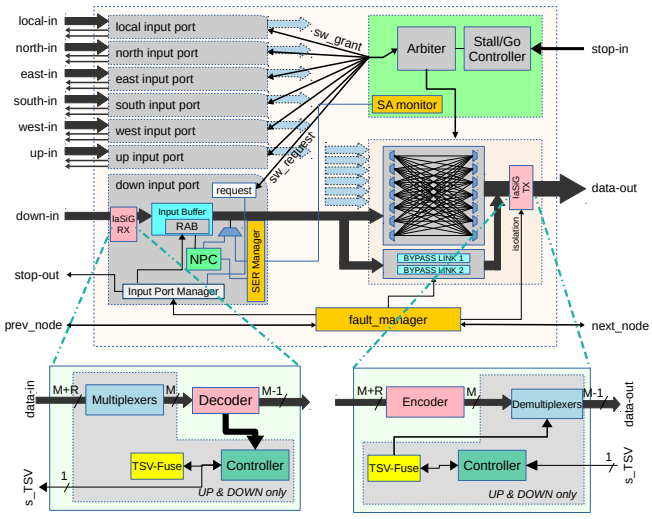

Figure 5: The 3D-NoC router with the proposed on-line fault detector. Note that there are only two modules in the router for UP and DOWN connections.

In order to understand the cost of the design, we integrate the *IaSiG* into our previously designed 3D-NoC router [23] as shown in Figure 5. Please note that the proposed approach is totally independent from our opted router architecture and could be implemented into any TSV-based architecture. The IaSiG is integrated as an ECC module for only two vertical ports (UP and DOWN) to monitor and detect faults of TSVs. *IaSiG-TX* and *IaSiG-RX* are two additional modules, which help handling the TSV fault detection: the data from TSV is brought to the *Multiplexer* for selecting the proper connection

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

then sent to the decoder. Parity check information of the decoder is sent to the controller of *IaSiG-TX*. The controller manages the "TSV-Fuse" (for multiplexer) and synchronizes between layers via s_TSV.

Previously in [23], we used two SECDED (16,22) codes to handle potential soft errors in the data. In this work, we use one parity check (32,33) for simplicity since the soft-error tolerance is not the main focus. We add only two spare TSVs (r_TSVs) and one synchronization TSV (s_TSV). Consequently, IaSiG uses 36 TSVs for each vertical connection. Also, we adopt the four-clustering structures of [23] where each connection is divided into four clusters of 8 TSVs (s_TSV is independent) and a cluster of 4 TSVs consists of parity check, spare TSV and synchronization TSV. The IaSiG also performs checking for four group of 8 TSVs.

For each vertical connection, if a cluster is defected, the router can choose one of its four neighboring clusters as a replacement without the need for redundancy. To satisfy the timing constraints, the router chooses the closest TSV-cluster among its neighbor clusters.

D. ANALYTICAL ANALYSIS

Lemma III.1. *Assuming the error probability of TSVs is independent, the probability of having a silent error after K transmissions is less than or equal to* $(1/2)^{2K}$.

Proof. The probability of a silent error for open and a short defects are:

$$P_{\text{silent 1-bit open}} = P_{0\to0} + P_{1\to1} \simeq 1/2 \tag{6}$$

$$P_{\text{silent 1-bit short-to-substrate}} = P_0 \simeq 1/2 \tag{7}$$

where P_i is the probability of transmitting a logic value i in TSV and $P_{i\to j}$ is the probability of transition from logic value i to logic value j. Here, we use $P_{\text{silent 1-bit}} = 1/2$ as the probability of having a silent 1-bit.

The probability of having a silent error while having f faulty TSVs ($0 \le f \le M$) is:

$$\begin{aligned} P_{\text{silent f-bit}} &= \sum_{i|2}^{f} \binom{f}{i} (1 - P_{\text{silent 1-bit}})^{f-i} (P_{\text{silent 1-bit}})^{i} \\ &= \sum_{i|2}^{f} \binom{f}{i} (1/2)^{f} \simeq 1/2 \end{aligned} \tag{8}$$

Equation 8 can be proven using the binomial theorem:

$$(x+y)^f = \sum_{i=0}^{f} \binom{f}{i} x^{f-i} y^i \tag{9}$$

By using $x = 1$ and $y = -1$:

$$\begin{aligned} (1+-1)^f = 0 &= \sum_{i=0}^{f} \binom{f}{i} (-1)^i \\ 0 &= \sum_{i|2}^{f} \binom{f}{i} - \sum_{i\nmid 0}^{f} \binom{f}{i} \end{aligned} \tag{10}$$

$$\sum_{i|2}^{f} \binom{f}{i} = \sum_{i\nmid 2}^{f} \binom{f}{i} \tag{11}$$

In other words, the number of having odd cases is equivalent to the number of having even cases. Meanwhile, the total number of cases is 2^f.

$$\begin{aligned} \sum_{i|2}^{f} \binom{f}{i} + \sum_{i\nmid 2}^{f} \binom{f}{i} &= 2^f \\ \sum_{i|2}^{f} \binom{f}{i} &\simeq \frac{2^f}{2} = 2^{f-1} \end{aligned} \tag{12}$$

Equation 8 could be proven as follows:

$$\sum_{i|2}^{f} \binom{f}{i} (1/2)^f \simeq 2^{f-1}(1/2)^f = 1/2 \tag{13}$$

Because the error probability of each TSV is independent, the probability of having a full silent error after K transmissions is:

$$P_{\text{silent}} \simeq (1/2)^K \tag{14}$$

Since Algorithm 1 uses double-check when it finds a healthy case, it performs one more K transmission. As a result, the silent probability is:

$$P_{\text{silent}} \simeq (1/2)^{2K} \tag{15}$$

□

Note that Algorithm 1 iterates from the least significant index TSV to the most significant one. The successful rate of the model is:

$$P_{\text{succesful-detection}} \simeq 1 - \left[(1/2)^{2K} \times \sum_{r=0}^{R} \binom{M+R}{r} \right] \tag{16}$$

Base on Eq. 16, we can conclude that:

- Having longer transaction length K can reduce the probability of silent faults.
- Having higher number of redundancies could enhance the reliability; however, it increases the testing time.
- In real-applications, there is a chance that the system sending all-zeros or all-ones during the test (even the chance is relatively small), we think this issue could be fixed by a dedicated data.

As shown in Algorithm 1, we use double-check to reduce the chance of hidden defects. By performing one more K transmission, we reduce the probability of silent defects to $\simeq (1/2)^{2K}$, which is a half of the probability in [27].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

Lemma III.2. *The method can guarantee the detection of $R+1$ and the correction of R faults. Furthermore, the detection and localization bound are $G(R+1)$ and GR, respectively.*

Proof. First, we assume F is the number of faults. Regardless of F being odd or even, silent errors after K transmissions make the decoder detect the failed case. If $F \leq R$, any cases of F faults in M TSVs could be covered by iterations from 0 to 2^M in Algorithm 1. Therefore, after a heuristic search through all possible cases, the system can match the F fault patterns once.

Given R redundancies, the maximum number of isolated faults is R since the system needs at least $M-R$ TSVs to work. If after reducing to M TSVs, the output of decoder (S) is equal to "1", since there is one fault left. Therefore, $R+1$ is the maximum number of detectable faults.

Now considering G groups in *IaSiG*, where each group is performed separately. If the faults occur in all groups, *IaSiG* can localize up to $G \times R$ defects and detect the case of $G(R+1)$ defects. □

E. EMPIRICAL THERMAL AWARE REDUNDANCY INSERTION

As we previously mentioned, most existing works inserted redundancies uniformly to TSV groups to help recover the manufacturing defects. However, lifetime reliability is heavily affected by the operating temperature as in Black's model [25] where the fault rate at the temperature T (Kelvin) is accelerated by a factor π_T as follows:

$$\pi_T = A(J)^n \times e^{-\frac{E_a}{k_B T}} \tag{17}$$

where J is the current density and A, E_a and k_B are the pre-exponential factor, activation energy and Boltzmann constant, respectively. Since the temperature map of the 3D-ICs is not uniform, the fault rates are varied between the TSV groups. Therefore, we observe that with the ability to predict the critical area, we can efficiently insert more redundancies into the less reliable area.

To find the vulnerable area, we normalize the acceleration factor π_T to a reference temperature T_{ref} to obtain the normalized fault rate (NFR_T) of each region of the 3D-IC:

$$NFR_T = \frac{\pi_T}{\pi_{T_{ref}}} = \frac{A(J)^n e^{-\frac{E_a}{k_B T}}}{A(J)^n e^{-\frac{E_a}{k_B T_{ref}}}} = e^{\frac{E_a}{k_B}(\frac{1}{T_{ref}} - \frac{1}{T})} \tag{18}$$

To predict the number of needed redundancies for lifetime reliability recovery, we proposed an empirical approach as follows:

- **Step 1**: Analyze the 3D-NoC system with the desired applications. Here, we use PARSEC benchmarks as examples.
- **Step 2**: Simulate the temperature of each TSV group and extract the predicted fault rate with the Black's model [25] by using Equation 18. We choose the coolest area of the 3D-IC as the reference temperature.
- **Step 3**: Insert the corresponding number of redundancies needed to the TSV group. Designers can set their target Mean Time to Failure (MTTF) or fault rate. Then, we perform a search to find the optimal redundancies for the target MTTF.

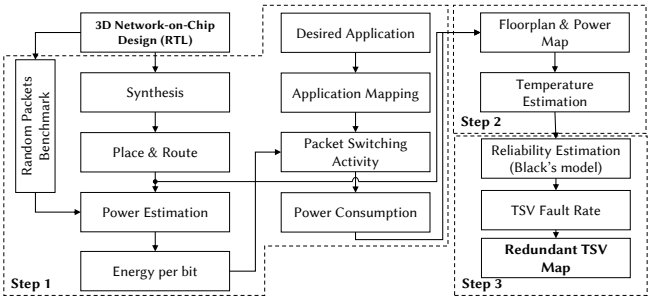

Figure 6: The empirical method for inserting redundant TSVs into the 3D NoC.

Figure 6 shows the flow to predict the number of needed redundancies in our 3D NoC. Obviously, designers can choose different fault model, application, reference temperature, and target MTTF (MTTF_{target}) to have the suitable results. In **Step 1**, we first use commercial CAD tools to design and estimate the power consumption. Then, we can extract the energy per bit value. To estimate the power consumption of each router, we multiply the energy per bit value to the packet switching activities from the desired applications. In **Step 2**, we use the power consumption map and floor-plan to estimate the temperature of each router and its nearby TSV groups in the NoC. We further estimate the normalized fault rate by using Equation 18. The redundant TSV map for each router is used for the final design of 3D-NoC in **Step 3**. To minimize the WCET, we split the group with larger number of redundancies into several sub-groups with maximum two redundancies ($R \leq 2$). We decide the number of redundancies based on two thresholds of fault rates (θ_1 and θ_2):

$$\#\text{redundancies} = \begin{cases} 0 & \text{if fault-rate} \leq \theta_1 \\ 1 & \text{if } \theta_1 < \text{fault-rate} \leq \theta_2 \\ 2, & \text{otherwise.} \end{cases} \tag{19}$$

To find the thresholds of fault rate (θ_1 and θ_2), we use a heuristic search. The tuning process of these two thresholds can be obtained via a grid search approach to find the optimal point for these two values.

In summary, this method helps estimate the needed redundancies for correcting TSVs under thermal awareness. By estimating the number of failed TSVs, we can reduce the area cost by removing the redundancies in low fault rate areas.

IV. EVALUATION

A. METHODOLOGY

The proposed architecture is designed in Verilog HDL using NANGATE 45nm library [41] and NCSU FreePDK TSV

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

[42]. The design is implemented using Synopsys CAD tools. We evaluate the hardware complexity of the proposed design in comparison to other TSV recovery methods. We also perform the execution time analysis and compare this work with existing ones. Last, we discuss the possible optimizations for this proposal.

B. HARDWARE COMPLEXITY

The implementation results with different configurations of the proposed method are shown in Figure 7. Compared to our preliminary work IaS [27], we observe a lower area cost due to the reduction in the number of registers needed for configuration. Consequently, the power consumption is also smaller. When we increase the number of redundancies, we observe a significant increase in both area and power of *IaS*. However, the increase in *IaSiG* is insignificant. In terms of power, we even observe lower consumption of our design thanks to the fewer registers and lower complexity.

Among the different configurations of *IaSiG*, we can observe a slight drop in area cost with $R = 1$ when we increase the value of C. This is due to the lower number of registers needed for each group. However, with $R = 2$, we observe an increase because of the complexity added for shifting two TSVs. On the other hand, since the switching between clusters incurs additional power consumption, we still observe higher power consumption with higher number of groups. We also would like to note that with low defect rates, it could reduce the power consumption by skipping groups.

The hardware complexity results and comparison for 32-bit implementations between this work and Error Correction Codes are shown in Table 2. We can observe that the proposed *IaSiG* has higher area cost and power consumption than most low complexity ECCs (Hamming, SECDED). With higher complicated ECCs (SEC-DAEC and TAEC), the total area cost of the encoder and decoder of *IaSiG* is much smaller. Note that *IaSiG* uses the least codeword bit (35), which also leads to smaller TSV area cost.

Table 2: Hardware implementation results.

Scheme		SECDED [19]	SEC-DAEC [29][a]	**IaSiG** (4×8)
Tech. (nm)		45	45	45
k (bit)		32	32	32
n (bit)		40	39	45
Area Cost (μm^2)	Encoder	111.7200	322	511.252
	Decoder	253.7640	1902	818.748
Latency (ns)	Encoder	0.60	0.53	0.90
	Decoder	1.44	1.33	1.82
Power (μW)	Encoder	36.9622	-	154.2795
	Decoder	103.1422	-	305.1569

[a] All designs are area optimized.

C. COMPARISON OF TSV TESTING AND RECOVERY MECHANISM

Table 3 shows the comparison between our method and existing testing frameworks for TSVs. We only compare the TSV testing while the system testing is shown in Table 5. As we can see in Table 3, this work at its best can capture the

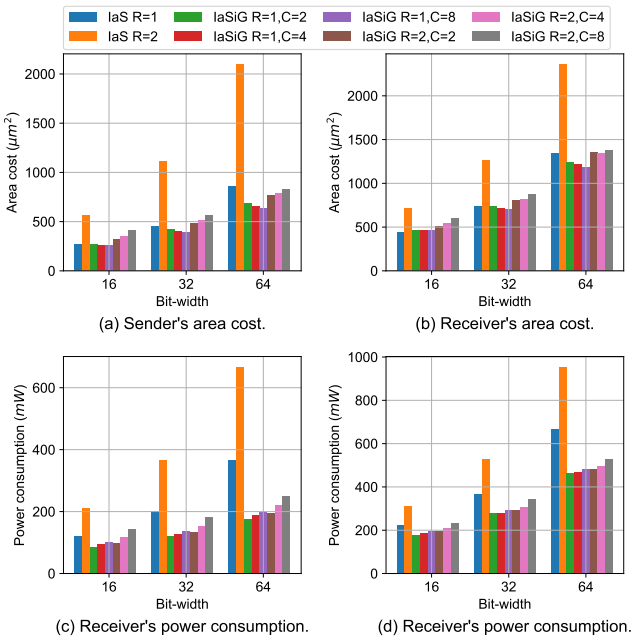

Figure 7: Implementation results of the proposal with different configurations.

fault within 32 cycles, which is 1 cycle per TSV. However, on the longer end, it can require in average 49 and 101 cycles per TSV with R=1 and R=2, respectively. These values are higher than other works; however, our work is for online and non-blocking test. Therefore, the long testing time can be easily justified since the system can continue to perform as usual. In terms of area cost, our method also has a higher area than others due to its need to store the status of TSVs. However, its area cost is still reasonable since the overhead of a TSV is not overwhelming. One notable thing we can observe is that our approach is suitable for a small group of TSVs. It is certainly possible to have a huge number of TSVs per group; however, as we analyzed, the testing time is not scalable. To handle this problem, we divide the TSVs into groups and support parallel testing. Therefore, we can make the testing time to remain constant and scale with the number of TSVs. On the other hand, collecting test results in the pre-bond and post-bond test require a scan chain, which lead to testing time and area complexity of $O(n)$ (n is the number of TSVs). Meanwhile, our method only requires $O(1)$ for testing time and could maintain the same testing time while up-scaling the system complexity (more cores or layers).

In comparison to alternative shapes of TSV groups such as cobweb [32], sharing spares [34], or honeycomb [33], our area overhead is also higher. For instance, in the honeycomb and cobweb design (w.o. TSV), the area per TSV is 17.61 μm^2 and 4.85 μm^2, respectively, while our value is 42.4 μm^2. However, their hardware complexity does not include the controlling for detecting and localizing TSV defect. Moreover, our design adopts the shifting mechanism from [31] and can be compatible with the other recovery

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

Table 3: Comparison table with existing works in TSV testing. S: number of signal TSVs; R: number of redundant TSVs.

Work	*Zhang et al.* [43]	*Zhao et al.* [12]	*Cho et al.* [14]	*Jani et al.* [44]	*Lee et al.* [36]	**This work**	
Test type	post-bond	online	pre-bond	post-bond	post-bond	online	
Technology	45 nm	130 nm	90/45 nm	28 nm	45 nm	45 nm	
Configuration	$S = 96$ $R = 24$	$S = 9112$ $R = 114$	$S = 1444$	$S = 10,000$	$S = 1,000$	$S = 32$ $G = 8$ $R = 1$	$S = 32$ $G = 8$ $R = 2$
Area w.o. TSV (μm^2)	self-test: 1, 128 control logic: 281.3 per TSV:11.7	detection: 111,403 recovery: 506,310 routing: 312,542 per TSV: 33.876	area: ≃46,656 per TSV: ≃21 (45nm)	passive: 61,370 active logic: 28,600 per TSV: 8.997	total: 1130.5[1] per TSV: 1.1305	total: 1097.0 per TSV: 33.2	total:1440.1 per TSV: 42.4
Test Time (cycles)	short: 3/TSV open: 2/TSV total: 1.04/TSV	test: S+R (1/TSV) repair: S+R (1/TSV) total: 2(S+R) (2/TSV)	1/TSV	alignment: 1/TSV RC: 1/TSV	Total: 0.5/TSV	best: 32 (1/TSV) worst: 1568 (49/TSV)	best: 32 (1/TSV) worst: 3232 (101/TSV)

[1] The estimation is based on the results represented in the paper.

designs.

D. WORST CASE EXECUTION TIME ANALYSIS

In order to understand the performance of *IaSiG* under different number of faults, we perform the test under different scenarios as shown in Figure 8. The numbers of data-bit (M) are 8, 16, 32, and 64. Here, we varied different group sets (C) between 2 and 4 TSVs. The number of redundancies (R) are from 1 to 6. Note that the value of R is also the number of injected defects. For real-time applications, the worst case execution time (WCET) is one of the most important criterion; therefore, we focus on measuring the WCET in this section. Although normal behavior could be faster, WCET represents the expected time of resulting test results. It is worth mentioning that with all designs (IaS, TSV-OCT, IaSiG), we use the K = 32, 64 and 128 since the experiment we conducted in *IaS* [27] shows 32 cycles is the best value that can guarantee no missing faults under random data. In this analysis, we first conduct Monte-Carlo simulation under 10,000 cases to find the WCET. Then, WCET analyses (see Section III-B) are conducted and verified with simulations. If there is an unmatched case, we conduct a specific case to verify the results.

In comparison to *IaS*, we easily observe lower WCET of *IaSiG* with a high number of redundancies. Notably, with a large number of data TSVs, *IaS* accelerates the WCET significantly while *IaSiG* maintains a reasonable one. For instance, with 32-bit TSVs, the WCET values of *IaSiG* are less than 20,000 cycles in all cases while *IaS* is even higher than 10^8 cycles. Even with the highest value configuration ($R = 6, K = 128$, and $M = 64$), *IaSiG* is still less than 30,000 cycles while *IaS* is above 10^{10} cycles. In summary, we observe that with a large number of redundancies *IaSiG* dominates *IaS* in terms of WCET. For smaller numbers of redundancies, *IaS* is better; however, the WCET of *IaSiG* is still smaller than 20,000 cycles, which is a reasonable test time for on-chip communications [37], [38].

In comparison to *TSV-OCT* [35], which is an error correction code with augmented algorithm, we can easily observe faster WCET under lower defect rates. This is due to the fact that the built-in ECC in this work already support localization of one fault; therefore, the execution time is much faster. However, with higher numbers of defects, we can easily observe the increase of WCET. With 4 defects, and $K > 32$, *TSV-OCT* has longer WCET than our proposal. While *TSV-OCT* is limited to 5-6 fault localization depending on the value of K, our method is only limited by the value R and can support defect detection in multiple groups. For instance with 32 bit ($M = 32$), cluster size $C = 4$ ($G = 8$ groups), and redundancy $R = 4$, *IaSiG* can detect at least $R = 4$ defects and at most $M \times R = 32$ defective TSVs. Also, while *TSV-OCT* does not support recovery, *IaSiG* can support the recovery of R defects. The number of additional TSVs is also lower, IaSiG uses $R = 4$redundant TSVs while *TSV-OCT* requires 9×5 TSVs for 32-bit (13 extra TSVs.)

As discussed in Section III-B, *IaSiG* under **case-2** (R defects in all TSVs) has less WCET than **case-1**, where *IaSiG* encounters R defects in each group. A significant reduction in WCET could be seen in Figure 8, where **case-2** has half of the execution time in comparison to **case-1**. However, we would like to note that with **case-1**, *IaSiG* can detect up to $G \times R$ defects. Since the WCET only scales with the value of G, R, and K, the WCET of the system is identical to a group. To maintain a reasonable WCET, designers can choose proper values to work with. However, if the WCET should be smaller than 1000 cycles, the choice is limited to 8 TSVs and 1 redundant one. We would like to note that reducing the number of cycles per test (K) can scale down the testing time; however, as we illustrated in Table 1, it can reduce the accuracy, which leads to hidden faults.

In summary, among the on-communication test techniques, *IaSiG* offers better scalability when the WCET slightly increases with the number of redundant TSVs. For the other works (IaS and TSV-OCT), we could observe the rapid increase of WCET.

E. NETWORK-ON-CHIP LINK TESTING PERFORMANCE

Table 4 shows the hardware complexity of the router design. The current work is based on the baseline router in [23]. Here, we compare with our previous works in [23] and [35] that use SECDED and PPC coding techniques, respectively. In comparison to these works, we observe that tour proposal has lower area cost than the *TSV-OCT* router [23] thanks to the lower complexity of the encoder and decoder and the smaller

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

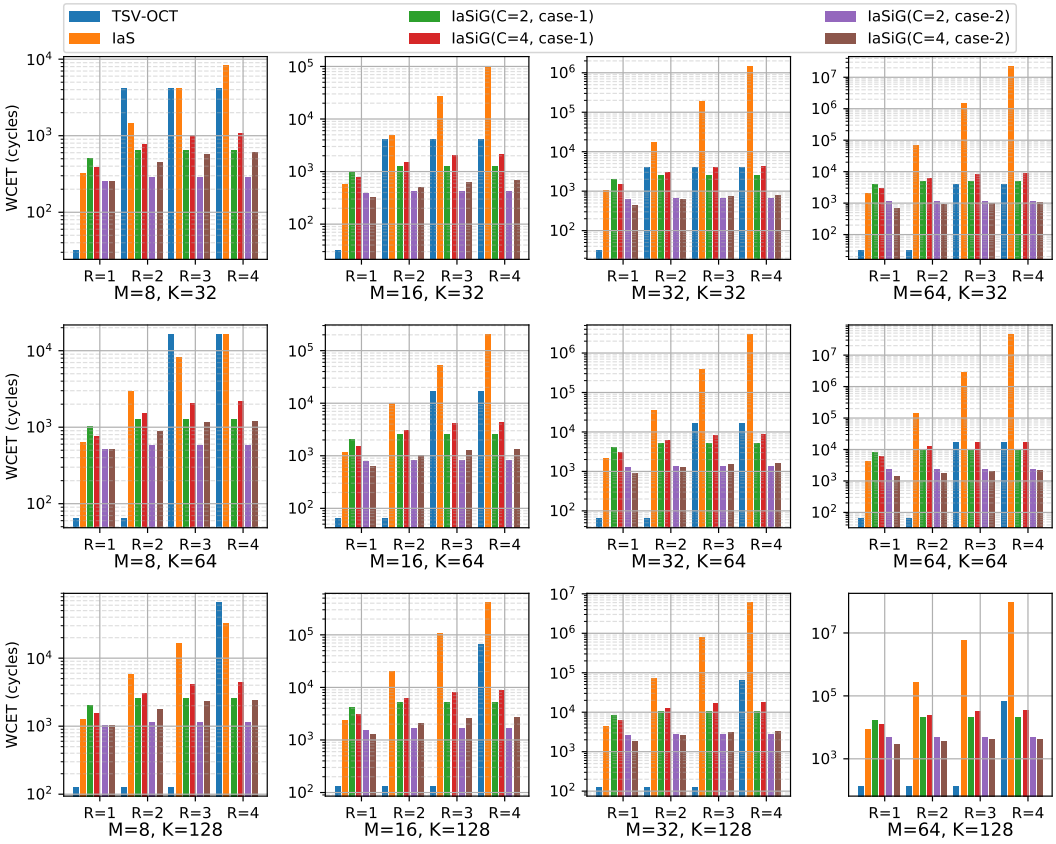

Figure 8: Comparison of worst case execution time (WCET). Note that the WCET of *IaSiG* are in **case-1** and **case-2** (see Section III-B). With TSV-OCT [35], R is the number of defective TSVs.

number of TSVs in a single connection (36 instead of 45 per connection). The area cost is similar to the router using SECDED code [23]. Note that both previous works use ECC so they can deal with soft errors while this work needs re-transmission.

Table 4: Hardware complexity of 3D NoC router (32-bit).

Design	**Specification**	**ECCs**	**Module**	**Area**	
				(μm^2)	(ratio to baseline)
Baseline router [23]	Wormhole, 4-flits buffer, 32-bit data, 3D Mesh	None	Router	18,873	-
			TSVs[1]	1,054.95	-
			Total	19,927.95	-
SECDED router [23]		2× SECDED (22,16)	Router	24,519	(129.92%)
			TSVs[1]	1,451.56	(137.50%)
			Total	25,970.56	(130.32%)
TSV-OCT router [35]		64 cycles PPC(4 × 8)	Router	26,843	(142.23%)
			TSVs	1,483.52	(140.63%)
			Total	28,326.52	(142.14%)
This work		Parity 32-bit R=2 K=32	Router	24,921	(132.05%)
			Encoder	511.252	-
			Decoder	818.748	-
			TSVs[1]	1,186.82	(112.50%)
			Total	26,107.82	(132.01%)

[1] TSV area: $4.06\mu m \times 4.06\mu m = 16.4836\mu m^2$. This work requires two connections (up and down), 35 signal TSVs and 1 feedback TSV (s_TSV) per connection.

Table 5 shows the testing time for Network-on-Chips. Here, we could easily observe our method is the fastest where the worst case (5,152 cycles) is smaller than other methods. Work in [35] can have a fast execution time; however, the worst case can reach up to 16,000 cycles, which is 5× the value obtained with proposed approach. We would like to note that in [35] the baseline test requires64 cycles, while in this work we confirm that 32 cycles provide the best results. With the double-check mechanism, we could elapse the test time twice to 64 cycles. For a 32-cycle baseline test, the work in [35] requires from 32 to 4128 cycles to perform, which is still larger than this work. This method supports a limited number of fault localization due to having only two redundancies while it is the only method providing online recovery among all the compared works.

F. EMPIRICAL THERMAL AWARE REDUNDANCIES INSERTION

In this part, we perform an empirical method to insert a suitable number of redundant TSVs in each group. Using the flow in Figure 6, we first use our baseline 3D-NoC to perform energy simulation by using Synopsys CAD tool. Here, we extract the average energy per bit of the proposed design. Then, by extracting the switching activities of PARSEC benchmarks [47] under gem5 [48] simulator with 3D mesh topology of Garnet NoC (64 cores: $4 \times 4 \times 4$), we obtain the power consumption for each router inside the NoC. We then

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

Table 5: Online testing circuit for Network-on-Chip comparison (32-bit and 45 nm technology).

Design	**Coverage**	**NoC size and topology**	**Connection preemption**	**Performance (normalized)**		**Test period/ time** (*cycles*)	**Area** (μm^2)	**Test circuit**	**Recovery circuit**
				Synthetic Period: 20K[b]	PARSEC Period:40K[b]				
Tran et al. [45][a]	Link & Router	10×8 Mesh	✓	$> 10\times$	$> 1.4\times$	200K-1M[c]	700	✗	✗
Kakoee et al. [38][a]	Link	10×8 Mesh	✓	$> 10\times$	$> 1.8\times$	500K-1M[c]	700	✗	✗
Dang et al. [35]	Link (TSVs)	any NoC	✗	$1.0\times$	$1.0\times$	64-16,448	2291.59	✓	✗
Liu et al. [46][a]	Router	10×8 Mesh	✓	$1.0 - 2.5\times$	$0.9 - 1.0\times$	20K-	2200	✗	✗
Wang et al. [37][a]	Link & Router	10×8 Mesh	✓	$1.5 - 2\times$	$0.9 - 1.0\times$	16,840-	2400	✗	✗
IaSiG ($R = 2, K = 32, G = 8$)	Link (TSVs)	any NoC	✗	$1.0\times$	$1.0\times$	32-5,152	1440.124	✓	✓

[a] Result are extracted from paper [37] by subtracting to the non-test router design (0.0251 mm^2).
[b] Performance normalization are approximated from paper [37].
[c] Test time or period (these terms are exchangeable because of on-operation test) is taken from paper [37] with a $> 5\times$ average latency and $> 1.8\times$ execution time performance degradation.

emulate the temperature using HotSpot 6.0 [49] to predict the temperature of each router. As we use the coolest area as the reference temperature, we can extract the normalized fault rate of each TSV group using Equation 18. Based on the fault rate, we insert proper redundant TSVs and arbitration modules to help detect and correct defects. Since the number of redundancies depends on the target MTTF (MTTF_{target}), designers can select a target value for their design. In this evaluation, we choose $1.5\times$, $2\times$ and $2.5\times$ the corresponding MTTF of the system at T_{ref} ($\mathsf{MTTF}_{T_{ref}}$).

Figure 9(b) illustrates the result of our empirical thermal aware insertion with $\mathsf{MTTF}_{target} = 2 \times \mathsf{MTTF}_{T_{ref}}$. As we can observe, under non-redundancy case (10:0), the MTTF of the system significantly drops to less than 50% of the target MTTF. Meanwhile, inserting one redundancy per group nearly reaches the target MTTF while inserting two redundancies overcomes the target MTTF. Here, our empirical method tries to optimize the number of redundancies per TSV group. As a result, the empirical approach approximately reaches the target MTTF. Meanwhile, the number of redundancies with the empirical approach is lower than 11:2 and higher than 11:1. This can be easily explained by Equation 19 where the proposal method chooses between either 0, 1 or 2 redundancies.

Figures 9(a) and 9(c) show the results with the target MTTF as $1.5\times$ and $2.5\times$ the $\mathsf{MTTF}_{T_{ref}}$. In both cases, the method approaches the target MTTF by using a grid search. With the target MTTF as $1.5\times$ of $\mathsf{MTTF}_{T_{ref}}$, it is natural that we need lesser TSVs than Figure 9(b).We can easily observe that constantly inserting either 1 or 2 redundant TSVs leads to higher MTTF than our desire. Despite the fact it can be more resilient, it needs extra area cost. With the $1.5\times$ case, the number of redundancies is lower than $R = 1$. Here, we can observe grid search can provide better area cost by reducing the number of TSVs. Here, the distribution of redundancy is no longer uniform; however, the overall reliability is still approximately as desired. On the other hand, the $2.5\times$ case still keeps the number of redundancies less than $R = 2$ and larger than $R = 1$, as can be seen in Figure 9(c). By selecting the optimal values of R for each group, we can have less TSVs while still maintaining the desired MTTF.

In summary, by heuristically searching for the optimal threshold values, the proposed approach can match the target MTTF while reducing the number of redundant TSVs. We would like to note that the WCET is equivalent to the WCET of the group with the highest R value. Therefore, grid search can optimize the number of TSVs; however, it cannot ensure the optimal WCET. On the other hand, we still cannot vary the value of M with grid search.

G. DISCUSSION

In the previous evaluations, we have presented the efficiency of *IaSiG*. Despite the obtained advantages, there are some challenges that should be addressed in order to further enhance the detection ability of *IaSiG*, as discussed hereafter.

First, this work does not take into account the occurrence of metastability, which could be solved by an immune circuit or a voltage comparator [14]. Also, the metastability phenomenon could be stabilized using several flip-flops and samplings.

Second, *IaSiG* is based on parity-check. Unlike *TSV-OCT* [27], which is based on ECC, parity-check cannot localize faults but can only detect them. As a result, *IaSiG* is not resilient to soft errors, which is out of the scope of this work. However, we can wrap *IaSiG* on top of an ECC to protect the data without encountering any incompatibility.

Third, the impact of real-chip implementation and process–voltage–temperature (PVT) variations has not been taken into account in this work. However, the efficiency of our algorithm is independent from the mentioned variations. Note that our target is lifetime reliability; therefore, we here assume the manufacturing defects are well recovered. Our execution times are analyzed in cycles, which can be varied with the frequency of the design.

Fourth, in Section III-E, we show an empirical method to decide the number of redundancies in each TSV group. Apparently, the selection is constrained by several parameters such as design, reference temperature, target MTTF. We also want to note that inserting one or two TSVs nearby a router can slightly reduce its temperature. While the reduction is ineligible, designers can modify the configuration of HotSpot 6.0 to obtain the updated temperature. Moreover, MTTF represents the average time to failure and may not reflect the worst case scenario.

Fifth, the selection of two threshold values is obtained via a grid search approach in our empirical method. This can help the designer to find an approximate solution after a certain execution time. However, we would like to note that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3022904, IEEE Access

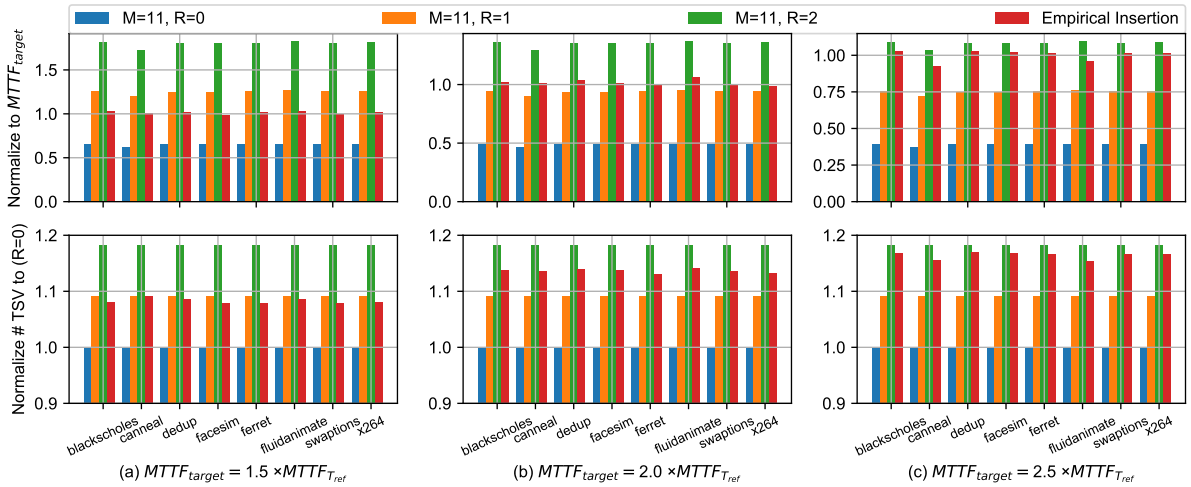

Figure 9: The result of our empirical method for inserting redundant TSVs into the 3D NoC with different target MTTFs.

selection process is still an open problem and out of the scope of this work. Our approach only demonstrates the ability to reduce the number of redundancies while maintaining the target MTTF. Since the examples in our evaluation still require up to two redundancies per group, the WCET is still as same as the $R = 2$. However, by lowering the target MTTF, the system can reduce the number of redundancies and can lower the WCET to $R = 1$.

Sixth, the main hypothesis of this work is to assume that the redundant TSVs (r_TSV) and feedback TSV (s_TSV) must be healthy. Obviously, this is a post manufacturing test and recovery so we can ensure the correctness after production. Furthermore, we avoid to use the redundant and feedback TSVs during operation to avoid wear-out. Nevertheless, the defect on these TSVs could lead to incorrect behaviors. While defective redundant TSVs could be detected, defects on feedback TSVs could lead to synchronization issues.

Seventh, although we only demonstrate the usage of our approach on a 3D-NoC under the PARSEC benchmarks, there is no limit on the application of this work with TSVs in 3D-ICs in general. Depending on the number of needed TSVs and the target reliability (or MTTF), designers can build their detection and recovery using either *IaS* or *IaSiG*. The empirical thermal aware insertion can also provide a certain confident number of redundancies needed for obtaining a target MTTF. Also, designers can adopt different reliability models rather than Black's model. The impact of different parameters (i.e., voltage, frequency, feature size) can also be integrated into the reliability estimation.

Finally, since our design has been completed in commercial CAD tools, the implementation of our approach is ready to be integrated onto any TSV-based 3D-ICs for production. Designers must carefully choose the parameters (R, K, or G) to satisfy their specifications. The computation of the controlling mechanism is fully implemented in hardware; thus, there is no extra computation needed.

Despite the above-mentioned limitations, we believe that *IaSiG* still provides extra defects' localization while maintaining short execution time. The exhibited overhead in a 3D-NoC implementation is also reasonable, which makes *IaSiG* a totally promising solution for integration into highly reliable 3D ICs.

V. CONCLUSION

This work has presented a light-weight method to enhance the online detectability and provide online localization and recovery for TSV's faults. The proposal isolates the possible fault position and checks the output syndrome to indicate the fault-free situation. To reduce the testing time and area cost, the proposed method divides TSVs into groups, which will have less TSVs per test. The results shows that the proposal has a reasonable area cost while guaranteeing localization and recovery up to the number of redundancies. An implementation of 32-bit version with 3D-NoC has less than 6000 cycles of testing time, which is significantly lower that the state-of-the-art works. Thermal adaptation for spare TSV insertion is also discussed in our proposal to help reduce the number of redundancies while still guaranteeing the desired MTTF. In the future, applying the proposed method for neuromorphic applications will be investigated. Different TSV group shapes and repairing mechanisms such as cobweb or honeycomb is another considerable approach.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.312.

References

[1] J. Cho et al., "Modeling and analysis of through-silicon via (TSV) noise coupling and suppression using a guard ring," IEEE Trans. Compon. Packag. Manuf. Technol., vol. 1, no. 2, pp. 220–233, 2011.

[2] J. Kim et al., "High-frequency scalable electrical model and analysis of a through silicon via (TSV)," IEEE Trans. Compon. Packag. Manuf. Technol., vol. 1, no. 2, pp. 181–195, 2011.
[3] W. R. Davis et al., "Demystifying 3D ICs: The pros and cons of going vertical," IEEE Des. Test. Comput., vol. 22, no. 6, pp. 498–510, 2005.
[4] X. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs)," in Proc. of the 2009 Asia and South Pacific Des. Automation Conf., 2009, pp. 234–241.
[5] J. U. Knickerbocker et al., "Three-dimensional silicon integration," IBM Journal of Research and Development, vol. 52, no. 6, pp. 553–569, 2008.
[6] G. Van der Plas et al., "Design issues and considerations for low-cost 3-D TSV IC technology," IEEE J. Solid-State Circuits, vol. 46, no. 1, pp. 293–307, 2011.
[7] F. Ye and K. Chakrabarty, "TSV open defects in 3D integrated circuits: Characterization, test, and optimal spare allocation," in Proc. of the 49th Annu. Des. Automation Conf. ACM, 2012, pp. 1024–1030.
[8] D. Cuesta et al., "Thermal-aware floorplanner for 3D IC, including TSVs, liquid microchannels and thermal domains optimization," Applied Soft Computing, vol. 34, pp. 164–177, 2015.
[9] Y. Lou et al., "Comparing through-silicon-via (TSV) void/pinhole defect self-test methods," Journal of Electronic Testing, vol. 28, no. 1, pp. 27–38, 2012.
[10] B. Noia et al., "Pre-bond probing of TSVs in 3D stacked ICs," in 2011 IEEE Int. Test Conf. (ITC). IEEE, 2011, pp. 1–10.
[11] P.-Y. Chen et al., "On-chip TSV testing for 3D IC before bonding using sense amplification," in 2009. ATS'09. Asian Test Symp. IEEE, 2009, pp. 450–455.
[12] Y. Zhao et al., "Online Fault Tolerance Technique for TSV-Based 3-D-IC," IEEE Trans. VLSI Syst., vol. 23, no. 8, pp. 1567–1571, 2015.
[13] C. Serafy and A. Srivastava, "Online TSV health monitoring and built-in self-repair to overcome aging," in Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2013 IEEE International Symposium on. IEEE, 2013, pp. 224–229.
[14] 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 Comput.-Aided Des., 2010, pp. 694–697.
[15] L. Jiang et al., "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.
[16] 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.
[17] 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.
[18] Q. Xu et al., "Clustered fault tolerance TSV planning for 3-D integrated circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 8, pp. 1287–1300, 2017.
[19] M.-Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," IBM J. Res. Dev., vol. 14, no. 4, pp. 395–401, 1970.
[20] R. Kumar and S. P. Khatri, "Crosstalk avoidance codes for 3D VLSI," in Automation and Test in Europe. EDA Consortium, 2013, pp. 1673–1678.
[21] B. Fu and P. Ampadu, "On hamming product codes with type-II hybrid ARQ for on-chip interconnects," IEEE Trans. Circuits Syst. I, vol. 56, no. 9, pp. 2042–2054, 2009.
[22] 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.
[23] K. N. Dang et al., "Scalable design methodology and online algorithm for TSV-cluster defects recovery in highly reliable 3D-NoC systems," IEEE Trans. Emerg. Topics Comput., in press.
[24] M. White, Microelectronics reliability: physics-of-failure based modeling and lifetime evaluation. JPL Publ., 2008.
[25] J. R. Black, "Mass transport of aluminum by momentum exchange with conducting electrons," in 6th Annual Reliability Physics Symposium (IEEE). IEEE, 1967, pp. 148–159.
[26] R. Dearden et al., "Real-time fault detection and situational awareness for rovers: Report on the mars technology program task," in 2004 IEEE Aerospace Conference Proceedings, vol. 2. IEEE, 2004, pp. 826–840.
[27] K. N. Dang et al., "TSV-IaS: Analytic analysis and low-cost non-preemptive on-line detection and correction method for TSV defects," in Proc. IEEE Computer Society Annual Symp. VLSI (ISVLSI), 2019, pp. 501–506.
[28] M. Hsiao, D. Bossen, and R. Chien, "Orthogonal latin square codes," IBM Journal of Research and Development, vol. 14, no. 4, pp. 390–394, 1970.
[29] A. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in 25th IEEE VLSI Test Symp. IEEE, 2007, pp. 349–354.
[30] M. Laisne et al., "Systems and methods utilizing redundancy in semi-conductor chip interconnects," 2013, US Patent 8,384,417.
[31] A.-C. Hsieh and T. Hwang, "TSV redundancy: Architecture and design issues in 3-D IC," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 4, pp. 711–722, 2012.
[32] T. Ni, D. Liu et al., "Architecture of cobweb-based redundant TSV for clustered faults," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 7, pp. 1736–1739, 2020.
[33] T. Ni et al., "A novel TDMA-based fault tolerance technique for the TSVs in 3D-ICs using honeycomb topology," IEEE Transactions on Emerging Topics in Computing, 2020.
[34] J. Park, M. Cheong, and S. Kang, "R2-TSV: A repairable and reliable TSV set structure reutilizing redundancies," IEEE Transactions on Reliability, vol. 66, no. 2, pp. 458–466, 2017.
[35] K. N. Dang et al., "TSV-OCT: A Scalable Online Multiple-TSV Defects Localization for Real-Time 3-D-IC systems," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2019.
[36] Y.-w. Lee, H. Lim, and S. Kang, "Grouping-based TSV test architecture for resistive open and bridge defects in 3-D-ICs," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 36, no. 10, pp. 1759–1763, 2017.
[37] J. Wang, M. Ebrahimi, L. Huang, X. Xie, Q. Li, G. Li, and A. Jantsch, "Efficient design-for-test approach for networks-on-chip," IEEE Transactions on Computers, vol. 68, no. 2, pp. 198–213, 2019.
[38] M. R. Kakoee, V. Bertacco, and L. Benini, "At-speed distributed functional testing to detect logic and delay faults in NoCs," IEEE Transactions on Computers, vol. 63, no. 3, pp. 703–717, 2014.
[39] L. Huang et al., "Non-blocking testing for network-on-chip," IEEE Trans. Comput., vol. 65, no. 3, pp. 679–692, 2016.
[40] C. Grecu et al., "Testing network-on-chip communication fabrics," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 26, no. 12, p. 2201, 2007.
[41] NanGate Inc., "Nangate Open Cell Library 45 nm," http://www.nangate.com/, (accessed 16.06.16).
[42] NCSU Electronic Design Automation, "FreePDK3D45 3D-IC process design kit," http://www.eda.ncsu.edu/wiki/FreePDK3D45:Contents, (accessed 16.06.16).
[43] J. Zhang et al., "Self-test method and recovery mechanism for high frequency TSV array," in 2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip (VLSI-SoC). IEEE, 2011, pp. 260–265.
[44] I. Jani et al., "BISTs for post-bond test and electrical analysis of high density 3D interconnect defects," in 2018 IEEE 23rd European Test Symposium (ETS). IEEE, 2018, pp. 1–6.
[45] X.-T. Tran et al., "Design-for-test approach of an asynchronous network-on-chip architecture and its associated test pattern generation and application," IET computers & digital techniques, vol. 3, no. 5, pp. 487–500, 2009.
[46] J. Liu et al., "Online traffic-aware fault detection for networks-on-chip," Journal of Parallel and Distributed Computing, vol. 74, no. 1, pp. 1984–1993, 2014.
[47] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, January 2011.
[48] N. Binkert et al., "The gem5 simulator," ACM SIGARCH computer architecture news, vol. 39, no. 2, pp. 1–7, 2011.
[49] R. Zhang, M. R. Stan, and K. Skadron, "Hotspot 6.0: Validation, acceleration and extension," University of Virginia, Tech. Rep, 2015.

KHANH N. DANG is currently an assistant professor at VNU Key Laboratory for Smart Integrated Systems, Vietnam National University Hanoi (VNU), Hanoi Vietnam. He received his B.Sc., M.Sc., and Ph.D. degree from VNU University of Engineering and Technology, University of Paris-Sud XI, and The University of Aizu, Japan in 2011, 2014, and 2017, respectively. His research interests include System-on-Chips/Network-on-Chips, 3D-ICs, and fault-tolerant systems.

Dr. Khanh N. Dang was visiting researcher at University of Aizu in 2019. This work has been partly done during his time in University of Aizu.

AKRAM BEN AHMED received his M.S.E. and Ph.D. degrees in Computer Science and Engineering from the University of Aizu, Japan, in 2012 and 2015, respectively. He later joined Keio University, Japan, as a postdoctoral researcher. He is currently a Research Scientist at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. His research interests include on-chip interconnection networks, reliable and fault-tolerant systems, and low-power embedded systems.

ABDERAZEK BEN ABDALLAH Abderazek Ben Abdallah is a full Professor of Computer Science and Engineering and the Head of the Division of Computer Engineering, the University of Aizu. He has been a faculty member at the University of Aizu since 2007. Before joining the University of Aizu, he was a research associate at the Graduate School of Information Systems, the Univ. of Electro-Communications at Tokyo from 2002 to 2007. He received the h.D. degree in computer engineering from the Univ. of Electro-Communications at Tokyo in 2002. His research falls primarily in the area of computer system and architecture, with an emphasis on adaptive/self-organizing systems, networks-on-chip/SoCs, processor micro-architecture and power & reliability-aware architectures. He is also interested in neuro-inspired systems and VLSI design for 3D-ICs. He has authored three books, published more than 150 journal articles and conference papers in these areas and given invited talks as well as courses at several universities. He has been a PI or CoPI of several projects for developing next generation high-performance reliable computing systems for applications in general purpose and pervasive computing. He is a senior member of IEEE and ACM and a member of IEICE.

XUAN-TU TRAN received a Ph.D. degree in 2008 from Grenoble INP (at the CEA-LETI), France, in Micro Nano Electronics. He is currently an associate professor at VNU-UET, Vietnam National University, Hanoi (VNU). He was an invited professor at the University Paris-Sud 11, France (2009, 2010, and 2015), at Grenoble INP, France (2011), at the University of Electro-Communications, Japan (2019), and adjunct professor at UTS, Australia (2017-2020). He is currently the director for the VNU Key Laboratory for Smart Integrated Systems (SISLAB). His research interests include design and test of systems-on-chips, networks-on-chips, design-for-testability, asynchronous/synchronous VLSI design, low power techniques, and hardware architectures for multimedia applications, cryptography.

He is a Senior Member of the IEEE, IEEE Circuits and Systems (CAS), IEEE Solid-State Circuits and Systems (SSCS), member of IEICE, and the Executive Board of the Radio Electronics Association of Vietnam (REV). He serves as Chairman of IEICE Vietnam Section, Chairman of IEEE SSCS Vietnam Chapter.