TSV-IaS: Analytic analysis and low-cost non-preemptive on-line detection and correction method for TSV defects

Khanh N. Dang*‡, Akram Ben Ahmed†, Abderazek Ben Abdallah‡ and Xuan-Tu Tran*
*SISLAB, University of Engineering and Technology, Vietnam National University Hanoi, Hanoi, 123106, Vietnam
†Department of Information and Computer Science, Keio University, Yokohama, 223-8522, Japan
‡Adaptive Systems Laboratory, The University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan.
Email: khanh.n.dang@vnu.edu.vn; khanh@u-aizu.ac.jp

***Abstract*—Through-Silicon-Via (TSV) is one of the most promising technologies to realize 3D Integrated Circuits (3D-ICs); however, the reliability issues due to the low yield rates and the sensitivity to thermal hotspots and stress issues are threating TSV-based 3D-ICs. On the other hand, real-time applications demand short response time to the new occurrence faults which make online testing become a critical challenge. In order to solve this problem, this paper presents an online method named TSV-IaS which supports detecting and correcting open and short defect TSVs by isolating and shifting the signals of TSVs then analyzing the output syndromes. Furthermore, we also present an analytical analysis on detectability and response time of the proposal. Results show that with R redundant TSVs in a group, the proposed method can fully correct R defects and detect R+1 defects while keeping a low-cost structure. Monte-Carlo simulations also demonstrate a 100% detection rate with 32-cycles based tests.**

***Index Terms*—Fault-Tolerance, Error Correction Code, Through Silicon Via, Real-time**

I. Introduction

Through-Silicon-Vias (TSVs) serve as vertical wires between two adjacent layers in Three Dimensional Integrated Circuits (3D-ICs). Thanks to their extremely short lengths, their latencies are low which could offer extremely high speeds of communication [1], [2]. Moreover, as a 3D-IC technology, TSV-based ICs can have smaller footprints despite the TSV's overheads [3], and lower power consumption thanks to the shorter wires [4].

Despite the aforementioned advantages, reliability has been a major concern of Through-Silicon-Vias due to their low yield rates [5], [6], vulnerability to thermal and stress, and the crosstalk issues of parallel TSVs [7], [8]. Built-in-self-test (BIST) [9], [10], external testing [11], [12] and on-line testing [13], [14] techniques are common methods to help the system to determine whether a TSV has a defect. On-line testing could be also handled by using an Error Correction Code (ECC). For recovery, there are three main approaches: *(i)* hardware fault-tolerance such as correction circuits [15], redundancies [16], reliability mapping [8]; *(ii)* information redundancy such as coding techniques [17], [18] or re-transmission request [19]; or *(iii)* algorithm-based fault-tolerance [20], [21].

Although numerous methods have been proposed to solve the reliability issues of TSVs, they are mostly off-line method and target for manufacturing defects. However, 3D-ICs confront the thermal and stress issue which could dramatically affect the lifetime reliability. Therefore, having light-weight and graceful degradation testing method could help the system aware of new defects. However, here are several issues need to be tackled:

1) For real-time applications, *new fault occurrences need to be detected, alerted and corrected on a timely basis* [22]. The test should be done by a specific deadline which could be used for check-pointing and recovery. Therefore, using BIST [9], [10] or external testing [11], [12] periodically (as known as Periodic Test)) may not satisfy these requirements due to its cost an enormous amount of cycles.
2) *Non-deterministic transmission/execution time is not preferred.* Real-time systems usually need to ensure the completion time of each task (computation/communication); therefore, the testing process should be deterministic. Also, other running tasks, which under various priorities, need to be completed by a specific deadline, are not always ready for being preempted for testing [23].
3) The only approach that can help the system to operate in real-time while ensuring the quality of connections is to use ECCs [17], [18]. However, *ECCs are usually limited by the number of detectable and correctable faults* which is not suitable for the clustering defect [16] in TSVs.

Because of the above problems, TSV-based 3D-ICs desire to have an on-line method to monitor, detect, localize and recover TSV defects. For real-time applications, the response times to the new occurred faults need to be limited by specific deadlines. Here, we use a method named on-communication test (OCT). Fig. 1 demonstrates our motivation by comparing periodic test [16] and our on-communication test. In Fig. 1(a), on-communication test (OCT) detects, localizes and corrects the fault after a new fault occurred which reduce the response time significantly. Fig. 1(b) shows the case of OCT can detect

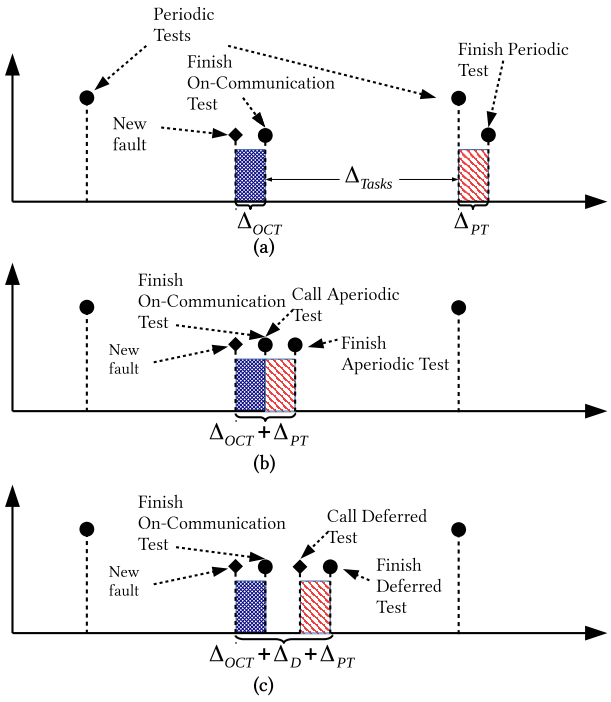

Figure 1. Motivation of On-communication Test (OCT): (a) OCT successfully corrects faults; (b) OCT unsuccessfully corrects faults then call an aperiodic test; (c) A deferred case where the test is called after completing higher priority transactions. The OCT reduces the response time by $\Delta_{PT} + \Delta_{Tasks} - \Delta_{OCT}$ and $\Delta_{Tasks} - \Delta_{OCT}$ in case (a) and (b), respectively. Δ_{PT}: worst-case execution time of the periodic tests (PT). Δ_{OCT}: worst-case execution time of OCTs. Δ_{Tasks}: execution time of remain system's tasks from the occurrence time of faults. Δ_D: deferred time to complete higher priority transactions.

but failed to correct; however, the system can initialize an aperiodic to test the TSVs. In the worst case, if the OCT fails to detect, the response time is equal to using a periodic test. Also, by having shorter response time, the OCT method requires lower recovery cost (latency, performance or power consumption). On the other hand, executing periodic test may not capable due to the real-time constraints of the system, deferred test could be used as shown in Fig. 1(c). Here, after getting results from OCT, the periodic test has to wait until being executed.

Therefore, in this paper, we propose a novel method named TSV **I**solation **a**nd **S**hift (TSV-IaS) which is specially designed for correcting and detecting faults in TSV-based links. The contributions of this paper are as follows:

- A method to isolate and check the possible defects in a group of TSV.
- An on-communication test (OCT) algorithm which provides real-time responsiveness to new occurred faults.
- An analytical analysis of IaS to demonstrate the efficiency of the proposal.

The organization of this paper is as follows: Section II presents the preliminary works. Then, Section III describes the proposal. Section IV shows the evaluation and Section V concludes the paper.

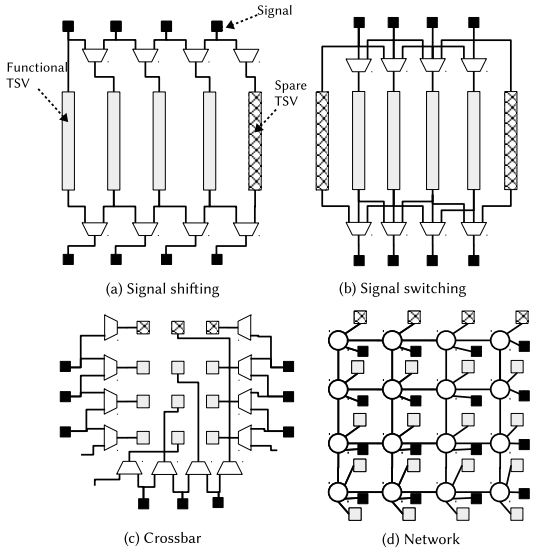

Figure 2. Methods of recovery TSV failures. Black boxes are redundant TSV; black circles are the signal terminal; white circles are the routing circuits.

II. Preliminary

In this section, we first present the preliminary work parity-based ECC methods and TSV recovery.

Since parity calculation requires only XOR gates, its simplicity in terms of area, power, and latency makes it become popular in integrated systems' error detection and correction codes. The most basic method is Parity-check which is commonly used in communication. Hamming [17] and its extension [18] (SECDED: Single Error Correction, Double Error Detection) is also two common methods. The delay and area complexity of those 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-link. 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.

There are two common methods in TSV recovery: (1) double [24] and (2) shared spare TSVs [25], [26]. While doubling TSV may offer high reliability, it also doubles the area cost which is already a major issue of TSV. Therefore, recent researches tend to focus on shared spare TSV instead. Figure 2 shows four major methods of shared spare TSV recovery: (1) shifting [27], (2) switching [25]; (3) crossbar [26] and (4) network [16]. Among the shared spare TSV, multiplexers and multiplexers are commonly used. To redirect the TSV signal, a multiplexer [25]–[27] or tri-state gate [21], [28] could be used.

III. Proposal

In this work, we consider the on-line monitoring and correcting for TSVs. We assume that the manufacturing test has already performed and the system can correct TSVs if needed.

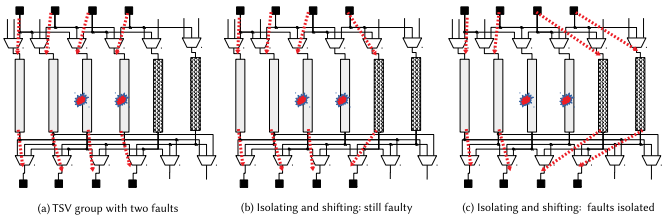

Figure 3. Illustration of shifting method with two spare TSVs. The parity check is used to detect and localize the fault. Red dotted arrow is the configuration to isolate and localize the fault.

A. TSV Organization

In this work, we adopt the TSVs as one-dimensional arrays and the shifting mechanism for recovery [27]. However, our technique is general and can be applied for other organization and recovery methods. Assuming the TSV group is organized in a group of M original TSVs and R spare TSVs. Each TSV is symbolized as t_i where i is the element indexes. Also, the signal group is organized as a similar one-dimension array and symbolized as s_i.

For a group of M bits data, we adopt the parity check as the based coding technique. This could help the proposal easily integrates into any parity-based ECCs.

B. Isolating and Shifting

The isolation is performed by considering the isolated TSV as faulty and using a spare TSV to continue to work via the shifting method. For instance, if the TSV with index f is isolated, signals $s_{i \geq f}$ are routed to the TSV t_{i+1}.

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

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

Note that the detection process is based on statistics. Here, we define a healthy set of TSVs is to have less than T faulty outputs after K transmissions ($K > 2$). The threshold T could be set to 2 in order to distinguish defect from single event upset; otherwise, $T = 1$.

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

As a demonstration of multiple fault detection, Fig 3 shows the illustration of using shifting to help detect and correct two faults by using Parity-check. Please note that our method satisfies for being OCT since it does not need to preempt any data transaction for testing. The proof of using Parity-check to detect multiple faults is shown on Lemma III.1.

C. On-Communication Test Algorithm

Algorithm 1 shows the OCT algorithm for detecting and localizing the possible faults in a TSV group. It starts with isolating one TSV and using redundant TSVs to handle the communication. The parity check will find whether 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 position. If the non-isolated is not faulty ($S_{(f=NULL)} = 0$) and the isolated case is faulty ($S_{(f=i)} = 1$), the system indicates there are two faulty positions. Then, the system starts to isolate more TSVs until reaching the limitation (R spare TSVs).

Algorithm 1: OCT Algorithm.

```
   Input: S;                          // output of detector
   Output: Fault_Idx;                 // fault indexes
1  foreach i in 1:R do
2      while no case left do
3          isolate i TSVs
4          if S_(f=i) = 0 and S_(f≠i) = 1 then
               // Faults are localized
5              Fault_Idx[all i TSVs] = 1;
6  return Fault_Idx;
```

To support real-time applications, fully hardware architecture is used. We start with the isolated index array $isol\_TSV = 0$ then increase it after K transmissions. If the new value has less than or equal R number of bit '1' which mean less than or R isolated TSVs, the new value will be used for fusing TSVs. If the new value has more than R number of bit '1', the new value is skipped until the satisfied value is found. By keep looping, the controller can heuristically check all possible cases.

In term of execution time, the proposal needs

$$K \times \sum_{r=0}^{R} \binom{M+R}{r}$$

cycles to complete its loops. It is also the worst case execution time (WCET) to new occurred faults. To reduce WCET, we can simply reduce the size of the TSV group or provide dynamic group (smaller M and R).

In [27], the authors presented testing TSV using transmitting two values '0' and '1'. Test generator is also used in [13], [26]. Here, once TSVs are isolated, they could be tested using a dedicated tester.

D. Architecture

Figure 4 shows the brief architecture for a TSV group with [M=4, R=1] TSVs. Here, we adopt the parity-check as ECC. The input data's width is M-1 and the encoded data width is M. 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. This box receive 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.

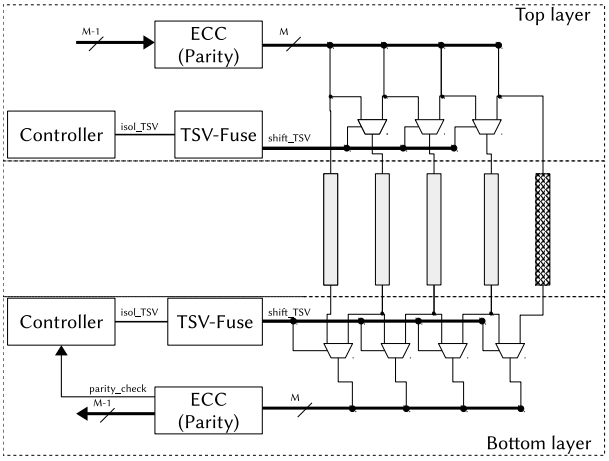

Figure 4. Proposed architecture for M=4, R=1.

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.

E. Analytical Analysis

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

Proof. The probability of silent error of an open and a short defect is:

$$P_{\text{silent 1-bit open}} = P_{0\to0} + P_{1\to1} \simeq 1/2 \quad (3)$$

$$P_{\text{silent 1-bit short-to-substrate}} = P_0 \simeq 1/2 \quad (4)$$

where P_i is the probability of transmitting 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 silent 1-bit.

The probability of encountering a silent error while having f faulty TSVs ($0 \le f \le M$) is[1]:

$$P_{\text{silent f-bit}} = \sum_{i|2}^{f} \binom{f}{i} (1 - P_{\text{silent 1-bit}})^{f-i} (P_{\text{silent 1-bit}})^{i}$$

$$= \sum_{i|2}^{f} \binom{f}{i} (1/2)^f \simeq 1/2 \quad (5)$$

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

$$P_{\text{silent}} \simeq (1/2)^K \quad (6)$$

□

[1]Proof is shown in the Appendix.

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

$$P_{\text{succesful-detection}} \simeq 1 - \left[(1/2)^K \times \sum_{r=0}^{R} \binom{M+R}{r} \right] \quad (7)$$

Base on Eq. 7, we can conclude that:

- Having longer transaction length K can reduce the probability of silent fault.
- Having higher redundancy could enhance the reliability; however, it both reduces the detection rate and increase the testing time.

Lemma III.2. *The system can detect* $R+1$ *and correct* R *faults.*

Proof. Assume F is the number of faults. Regardless of F being odd or even, silent errors after K transmission make the decoder detects the failed case. If $F \le R$, any cases of F faults in M TSVs could be covered by iteration from 0 to 2^M in Algorithm 1. Therefore, after a heuristic search through all possible case, 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=1, since there is one fault left. Therefore, $R+1$ is the maximum number of detectable faults. □

IV. Evaluation

A. Methodology

The proposed architecture is designed in Verilog HDL using NANGATE 45nm library [31] and NCSU FreePDK TSV [32]. The design is implemented using Synopsys tools. We evaluate hardware complexity of the proposed design in comparison to TSV recovery method.

The TSV defects are modeled as:

- **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, delay the value of TSV by one clock cycle.

Please note that this work does not take into account the occurrence of metastability which could be solved by using an immune circuit [33], a voltage comparator [15], or several flip-flops and samplings.

B. Hardware Complexity

Table I shows the hardware complexity of the proposed mechanism in comparison to existing methods. In general, the normalized area cost of our IaS is reasonably small but it still larger than the large-scale BIST [29], [30]. This is due to the BIST only use one module to test all TSVs; however, the area of the BIST is more than 200 times significantly bigger than ours.

In comparison to online methods [13], [15], [16], ours area cost is smaller while providing shorter response times from new faults. Our method is the only one could provide short response time.

Table I
HARDWARE IMPLEMENTATION RESULTS AND COMPARISON.

Technique		*Zhao et al.* [13]	*Park et al.* [29]	*Cho et al.* [15]	*Jani et al.* [30]	*Jiang et al.* [16]	Ours	
Brief Description		Online detection and recovery	Fast BIST extraction for TSV	Detection and correct for open-defect	BIST engines for post-bond test and electrical analysis	Online recovery for TSV cluster defect	On-communication test and shifting	
Technology		130 nm	32 nm	45 nm	28 nm	65 nm		45 nm
TSV pitch		5 μm	N/A	N/A	3.45 μm	10 μm		10 μm
Grouping ratio (M:R)		80:2	N/A	N/A	N/A	4:2	5:1	5:2
# TSVs		1,362	240	1	10,000	1000	6	7
Cost (μm^2)	Detection	16,857	3,780	21	89,970	CPU-based test	269.7240	395.5420
	Recovery	112,407	N/A	N/A	N/A	N/A	17.8220	42.5600
	All	129,264	3,780	$\simeq 21$	89,970	152,010	287.5460	438.1020
	Normalize (w/o TSV)	69.9075	240	$\simeq 21$	8.9970	52.010	57.5092	87.6204

C. Detection Evaluation

Table II shows the detection evaluation using Monte-Carlo simulation with 10,000 tests (each test consists of a full loop). The simulation values satisfy the theoretical estimation based on Eq. 7. With small values of K (8, 16), the proposed method encounter hidden faults. However, if the value K raises to 32, there is no case that the method fails to detect. There is no incorrect detection this those evaluation.

Table II
DETECTION RESULTS WITH MONTE-CARLO SIMULATION.

M	R	K	# hidden error	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

D. Response time

Table III shows the comparison between our method and existing testing frameworks. We choose SoC-based and NoC-based testing frameworks as the target because TSVs are the near-future integration technology for SoCs and NoCs. As we can see in Table III, the testing time for our method is not only non-preemptive but it also support short response time.

For the response time, most existing works on SoC-testing [34]–[38] requires hundreds of thousand or million cycles to complete. However, they are mostly preemptive testing which requires disabling completely or partially the execution of the system for test. On the other hand, our method provides shorter response time. For 32-bit data, 2 redundancies and 32 transactions, it only take 16,896 cycles for completing the test. Lower data bits could provide faster response times.

Because no scan chain is needed in our proposal, OCT can act in parallel to reduce the test time. In other words, our method could maintain the same testing time while up-scaling the system complexity (more cores or layers). On the other hand, BIST methods cannot solve the scalability issue.

V. CONCLUSION

This work has presented a light-weight method, to enhance the on-line detectability and provide on-line 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.

Using adaptive test cycles (K) and integrating on realistic application (3D-RAM, 3D-Network-on-Chip) is the future work.

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] 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.

[4] 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.

[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] U. Kang *et al.*, "8 Gb 3-D DDR3 DRAM using through-silicon-via technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 111–119, 2010.

[7] 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.

[8] 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.

[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] M. Tsai *et al.*, "Through silicon via (TSV) defect/pinhole self test circuit for 3D-IC," in *IEEE Int. Conf. on 3DIC*. IEEE, 2009, pp. 1–8.

[11] 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.

[12] P.-Y. Chen *et al.*, "On-chip TSV testing for 3D IC before bonding using sense amplification," in *Asian Test Symp.* IEEE, 2009, pp. 450–455.

[13] 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.

Table III
COMPARISON OF RESPONSE TIME.

Work	Brief. description	**Tech.** (nm)	**Area per TSV** (μm^2)	**Test Time** (cycles)
Li et al. [34]	On-chip test framework for 3D-IC	90	134.505	system: 11,009,580 data TSV: 114 address TSV: 307
Grecu et al. [35]	NoC testing	gate count	0.3299	unicast: 223,368 multicast: 15,233
Liu et al. [36]	Test scheduling NoC	-	-	105,775
Amory et al. [37]	Test scheduling NoC	-	-	209,720
Xiang et al. [38]	NoC testing (8× 8)	-	-	221,119
Ours	Statistical detector, isolation and check for TSV	45	M=5, R=1: 57.5092 M=5, R=2: 87.6204	K=32, M=5, R=1: 192 K=32, M=5, R=2: 512 K=32, M=9, R=1: 320 K=32, M=9, R=2: 1,472 K=32, M=33, R=1: 1,088 K=32, M=33, R=2: 16,896

[14] C. Serafy and A. Srivastava, "Online tsv health monitoring and built-in self-repair to overcome aging," in *Int. Sym. on Defect and Fault Tolerance in VLSI and Nanotechnol. Sys. (DFT),*. IEEE, 2013, pp. 224–229.

[15] 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.

[16] 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.

[17] R. W. Hamming, "Error detecting and error correcting codes," *Bell Labs Tech. J.*, vol. 29, no. 2, pp. 147–160, 1950.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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 (IEEE Cat. No. 04TH8720)*, vol. 2. IEEE, 2004, pp. 826–840.

[23] G. C. Buttazzo, *Hard real-time computing systems: predictable scheduling algorithms and applications*. Springer, 2011, vol. 24.

[24] M. Laisne *et al.*, "Systems and methods utilizing redundancy in semiconductor chip interconnects," 2013, US Patent 8,384,417.

[25] 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.

[26] 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.

[27] 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.

[28] A. B. Abdallah *et al.*, "A low-overhead fault tolerant technique for tsv-based interconnects in 3d-ic systems," in *18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. IEEE, 2017, pp. 179–184.

[29] J. Park *et al.*, "Fresh: A new test result extraction scheme for fast tsv tests," *IEEE Trans Comput -Aided Design Integr Circuits Syst*, vol. 36, no. 2, pp. 336–345, 2017.

[30] 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.

[31] NanGate Inc., "Nangate Open Cell Library 45 nm," http://www.nangate.com/, (accessed 16.06.16).

[32] NCSU Electronic Design Automation, "FreePDK3D45 3D-IC process design kit," http://www.eda.ncsu.edu/wiki/FreePDK3D45:Contents, (accessed 16.06.16).

[33] K. A. Bowman *et al.*, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, 2009.

[34] L.-C. Li *et al.*, "An efficient 3d-ic on-chip test framework to embed tsv testing in memory bist," in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*. IEEE, 2015, pp. 520–525.

[35] 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.

[36] C. Liu *et al.*, "Reuse-based test access and integrated test scheduling for network-on-chip," in *Proceedings of the conference on Design, automation and test in Europe: Proceedings*. European Design and Automation Association, 2006, pp. 303–308.

[37] A. M. Amory *et al.*, "A scalable test strategy for network-on-chip routers," in *IEEE International Conference on Test, 2005.*, Nov 2005, pp. 9 pp.–599.

[38] D. Xiang *et al.*, "Multicast-based testing and thermal-aware test scheduling for 3d ics with a stacked network-on-chip," *IEEE Trans Comput*, vol. 65, no. 9, pp. 2767–2779, Sep. 2016.

APPENDIX

Equation 5 can be proven using the binomial theorem:

$$(x+y)^f = \sum_{i=0}^{f} \binom{f}{i} x^{f-i} y^i \tag{8}$$

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

$$(1+-1)^f = 0 = \sum_{i=0}^{f} \binom{f}{i} (-1)^i = \sum_{i|0}^{f} \binom{f}{i} - \sum_{i\nmid 0}^{f} \binom{f}{i}$$

$$\sum_{i|0}^{f} \binom{f}{i} = \sum_{i\nmid 0}^{f} \binom{f}{i} \tag{9}$$

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

$$\sum_{i|0}^{f} \binom{f}{i} + \sum_{i\nmid 0}^{f} \binom{f}{i} = 2^f \Rightarrow \sum_{i|0}^{f} \binom{f}{i} \quad \simeq \frac{2^f}{2} = 2^{f-1}$$

The Eq. 5 could be proven as follows:

$$\sum_{i|0}^{f} \binom{f}{i} (1/2)^f \simeq 2^{f-1}(1/2)^f = 1/2 \tag{10}$$