Dynamic Self-Repair Architectures for Defective Through-silicon Vias

Joon-Sung Yang, Tae Hee Han, Darshan Kobla and Edward L. Ju

Three-dimensional integration technology results in area savings, platform power savings, and an increase in performance. Through-silicon via (TSV) assembly and manufacturing processes can potentially introduce defects. This may result in increases in manufacturing and test costs and will cause a yield problem. To improve the yield, spare TSVs can be included to repair defective TSVs. This paper proposes a new built-in self-test feature to identify defective TSV channels. For defective TSVs, this paper also introduces dynamic self-repair architectures using code-based and hardware-mapping based repair.

Keywords: Through-silicon via, built-in self test, dynamic self repair, code-based repair, hardware mapping based repair.

I. Introduction

The three-dimensional (3D) integration technology enables systems with direct die stacking and die-to-die bonding. Through-silicon vias (TSVs), which are vertical interconnection links, provide a plurality of benefits such as high-density, high-bandwidth, and low-power operation [1]-[3]. Multiple device layers are placed together using TSV technology forming 3D stacked integrated circuits (3D SICs) [1], [2], [4]-[7]. For example, 3D stacked memory may include coupled layers or packages of DRAM memory elements. A number of semiconductor companies have announced 3D stacked memory products [3], [8]. TSV technology shows a blueprint in the semiconductor business to overcome the technology scaling limitation and continue Moore's Law [1], [3]. These advantages are helping companies meet the increasing market demands for high-density and low-power applications such as server applications and low-power system-on-a-chip (SoC) systems used in hand-held device applications.

While TSV technology has been shown to be a promising solution, there are concerns regarding the yield for 3D SICs [4]-[7], [9], [10]. The yield issues are dependent on many factors. The 3D SIC processing step includes wafer thinning, handling, and a post bonding process. Foreign particles caught between wafers, dies, or die-wafers can reduce the yield. Edge effects such as vulnerability to damage at the bonding edges, and alignment problems caused by via misalignment errors, cause a yield problem [11].

We can mainly classify the yield loss into two categories as shown below [6] and [12].

(a) Stack yield loss caused by defects in the stacked dies or wafers.

(b) Assembly yield loss caused by defects in the assembly

Manuscript received June 11, 2013; revised Aug. 28, 2013; accepted Oct. 9, 2013.

This paper was supported by Samsung Research Fund, Sungkyunkwan University, 2013.

Joon-Sung Yang (phone: +82 31 299 4325, js.yang@skku.edu) and Tae Hee Han (than@skku.edu) are with the Department of Semiconductors System Engineering, Sungkyunkwan University, Suwon, Rep. of Korea.

Darshan Kobla (darshan.kobla@intel.com) and Edward L. Ju (edward.l.ju@intel.com) are with Intel Corporation, Austin, TX, USA.

process.

Yield losses may result in significant costs for 3D SICs for device manufacturers.

For 3D SICs, testing plays a key role in guaranteeing the operation of 3D SICs since the assembly process and TSV manufacturing can potentially introduce defects. The test for 3D SICs has three main phases: a known-good die test, a known-good stack test, and a final test with the packaged product [11]. A known-good die (KGD) test of individual dies is performed after fabrication. For stacked dies, the known-good stack test is then run for intermediate stacked and final stacked dies. In this step, flaws or defects in the stacking process are identified. After packaging, the packaged product is tested as a final test. Some researches have been conducted on built-in-self-test (BIST) approaches to enhance a test flow for TSV technologies [2]- [6], [12], [13].

To overcome the yield issues, hardware redundancy has been widely applied. In 3D SICs, spare TSVs, such as column or row redundancies in memory, can be mainly used as a hardware redundancy technique to repair defective TSVs between stacked dies. Previous researches have focused on hardware-redundancy based TSV repair approaches [1], [3], [6]-[7], [14]. In this paper, we propose a new BIST feature to identify defective TSVs in 3D SICs and introduce cost-effective dynamic TSV repair techniques.

The remainder of this paper is organized as follows. In section II, some previous related works are described, and in section III, dynamic TSV repair approaches are proposed. Analysis and experimental results are given in section IV, and section V ends the paper with some concluding remarks.

II. Related Works

The low reliability of TSVs has brought about yield issues in 3D SICs [4]-[7], [9], [10]. Hardware redundancy is a commonly used technique to enhance a low yield problem. As a hardware redundancy in TSVs, spare TSVs are placed in 3D SICs to repair defective TSVs. TSV repair architectures can be implemented in a chain or grid type.

Figure 1 shows chain-type TSV redundancy approaches. In Fig. 1(a), doubling and tripling redundancy approaches are proposed. For doubling redundancy, a spare TSV is placed in every other TSV channel and a defective channel is re-routed to the spare TSV. With a tripling redundancy approach, spare TSV channels are located to tolerate one or two defective channels. This approach provides high fault-tolerance; however, there is a huge area overhead required to implement doubling and tripling TSV redundancy architectures [7]. Figure 1(b) shows a signal switching repair approach with two spare TSVs. Two spare TSVs are placed at the beginning and end of a group of TSV channels. Because I/O data can be sent to the left- or right-neighboring TSV to avoid defective TSVs, the signal switching method can tolerate two TSV failures [3]. This approach requires less hardware overhead than the doubling or tripling approach in Fig. 1(a). TSVs with spare channels are bundled to provide data routing toward the left- or right-direction using spare TSVs on each side. To reduce hardware costs, a TSV signal shifting approach, shown in Fig. 1(c) was proposed [1]. This repair architecture has one spare TSV, and the data can be shifted to one direction where a spare TSV is located. One defective TSV channel can be tolerated by shifting the signals, which forms a TSV chain.

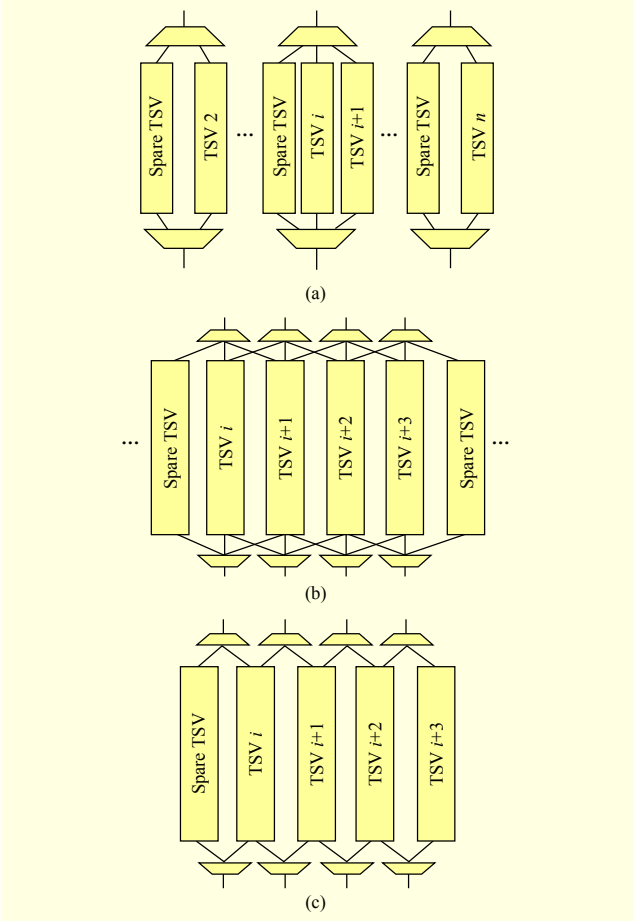

Fig. 1. Chain-type TSV repair architectures: (a) doubling and tripling redundancy, (b) signal switching, and (c) signal-shifting repair architectures.

Grid-type repair implementation approaches are shown in Fig. 2. Figure 2(a) shows an $n \times n$ TSV grid architecture forming a crossbar [14]. Because a crossbar connects multiple inputs to multiple outputs in a matrix manner, each defective TSV is connected to a spare TSV. If m spare TSVs are used, the spares can be re-routed up to m defective TSV channels. Figure 2(b) shows a TSV repair architecture based on a switch and routing path such as a Network-on-Chip (NoC) [6]. It

ETRI Journal, Volume 36, Number 2, April 2014
http://dx.doi.org/10.4218/etrij.14.0113.0578

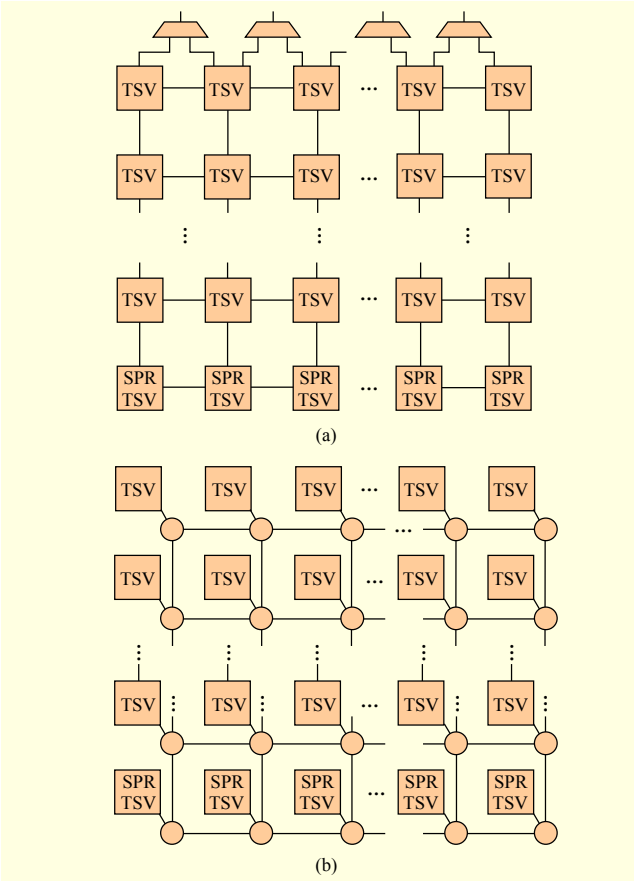

Fig. 2. Grid-type TSV repair architecture: (a) crossbar-based repair architecture and (b) NoC-based repair architecture.

allows for fully defective TSV channel replacement by spare TSVs using minimum routing paths.

The repair approaches shown in Fig. 1 and Fig. 2 may be useful in 3D SICs; however, conventional methods do not consider TSV interconnect effects or TSV stress on the transistors. This paper proposes a code-based TSV repair architecture and hardware-mapping based approaches considering the electrical effects from TSVs.

III. Details of Proposed Dynamic Repair Architectures with Defective TSV Detection

The following subsections describe the TSV interconnection effects and reliability issues, the new BIST feature to identify defective TSVs, and the two proposed TSV repair architectures.

1. TSV-Induced Stress and Reliability Issues

The widely used filling material for TSVs is copper, which causes TSV-related stresses [11], [15], [16]. TSV-induced stresses can cause reliability issues in several ways. Owing to the thermal coefficient difference between a TSV fill and silicon, thermo-mechanical stress may be introduced during the complex fabrication process [15]. Stress can cause a mechanical reliability issue such as crack growth in the interconnection. Tensile stress is also induced by a TSV, which can cause electrical effects around the TSV channels [16]. The change in the stress to silicon can change the mobility of carriers, which can result in device performance degradations near the TSVs. Transistors around the TSVs are influenced by stress, which can thereby degrade their performance and impact upon timing and reliability issues. In addition, there is an electromigration reliability impact problem of TSVs on the nearby metal wires [11].

Other factors such as temperature and current can cause a reliability issue in TSVs. Different causation profiles of this issue are provided in [11], which describes a stress modeling with respect to changes in stress, temperate, current, and overall combined impact. These combined impacts cause severer device performance degradations and reliability issues in silicon.

Considering TSV stress and reliability issues, we propose a defective TSV repair architecture minimizing the influences of the above factors. Unlike conventional methods, the proposed method minimizes the use of logic near the TSVs.

2. Identifying Defective TSVs

As addressed in section I, there are many factors causing a yield problem in 3D SICs. One way to enhance the yield issue is to replace/repair defective TSVs. To perform defective TSV channel repair, it is necessary to identify where defective TSVs are located. The locations of the defective channels are then found, and a defect map is generated through testing. This is used for defective TSV repair.

We propose a new BIST feature to support dynamic repair architectures for 3D SICs. Owing to manufacturing problems, wear-out, and TSV performance degradation, defective TSVs can occur at any time in the lifespan of 3D SICs. Conventionally, a defect map is generated only at the test stage. To find any defective TSVs occurring in its life cycle, we propose a BIST feature that runs a self-test at each system power up or reset cycle to generate a new defect map. Figure 3(a) shows a 3D SIC with a BIST module, and TSV channels connected to a BIST module in Die 1 and Die 2. The detailed structure of the proposed BIST is shown in Fig. 3(b). The hardware implementation includes a pattern generator with a shift register. Certain patterns are generated from the pattern generator and are sent to the TSVs. A comparator in the BIST block compares the read TSV data with the expected value and a comparator keeps track of the number of read mismatches. If the number of mismatches equals the expected

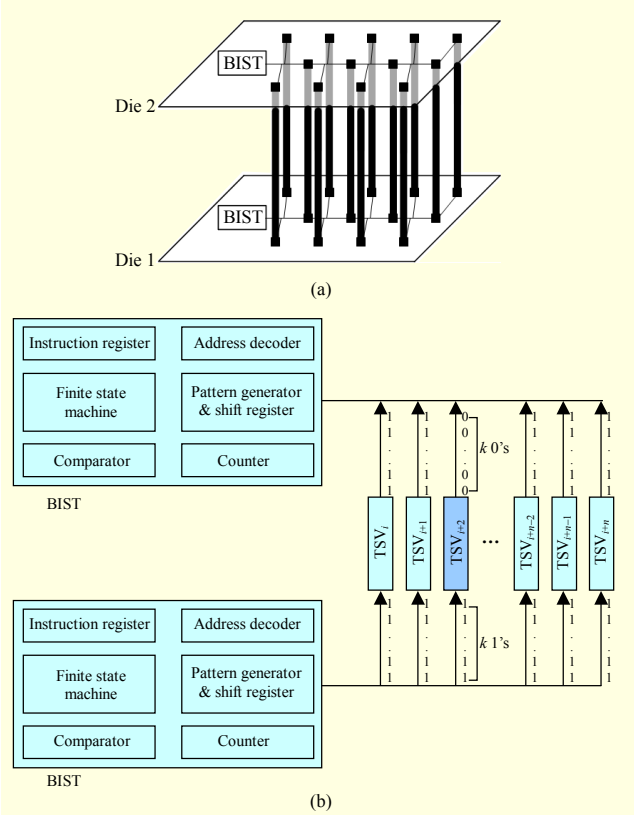

Fig. 3. Proposed BIST hardware implementation: (a) 3D SIC with BIST module and (b) proposed BIST structure with TSV.

value, the failing TSV channel can be identified.

As an example of a stacked DRAM, a DRAM may have four entries with each entry having a width of 32-bits. BIST writes all 1's to each entry in Die 1, and reads each entry of the four entries back in Die 2. Considering stuck-at faults, if a read mismatch occurs four times and if all of the mismatches are found at the same failing bit, then the TSV channel corresponding to the mismatch data bit is the failing one and needs to be replaced. In Fig. 3(b), all 1's are sent k times from Die 1, and BIST in Die 2 reads the value k times. TSV_{i+2} highlighted in gray shows k 0's while 1's are sent k times from Die 1. A comparator and a counter are used to identify defective TSVs. Hence, TSV_{i+2} is identified as a defective TSV. In this fashion, a TSV defect map can be generated.

The BIST operation is performed as a part of the initial firmware bring-up at every power up or reset sequence. This allows the detection of a dynamic failing TSV.

3. Dynamic Hardware-Based Repair Algorithm

In this paper, unlike conventional repair approaches [1], [3], [6]-[7], [14], we minimize the use of logic around the TSVs owing to the stress and reliability issues addressed in section III. Based on this guideline, we propose a dynamic hardware-mapping based failing TSV repair architecture.

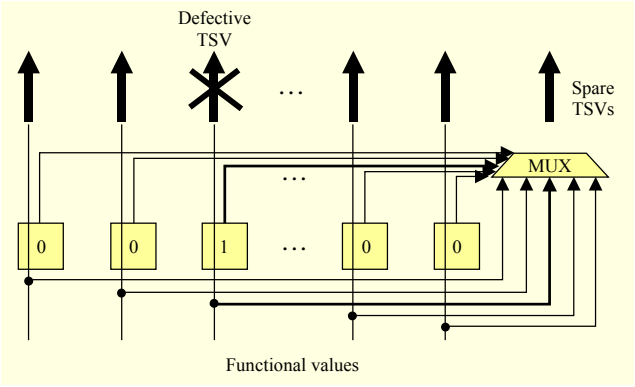

Fig. 4. Dynamic hardware-mapping based self-repair architecture.

During a system reset or power-up, the proposed BIST operation in section III detects defective TSVs, and a defect map is generated. The BIST sends data from Die 1 to Die 2 to produce a defect map for Die 1, and a Die 2 defect map is generated by sending data from Die 2 to Die 1. In this manner, the locations of the failing TSVs are found from the BIST at each die.

Figure 4 shows the hardware architecture of a dynamic hardware-mapping-based approach. Instead of placing a number of multiplexers at every TSV channel, the proposed method places a multiplexer at a spare TSV channel to minimize the TSV stresses on the logic. Defect map information is used as selection bits for the multiplexer. Because a boundary scan chain is not used after manufacturing testing, selection bits can be stored in the boundary scan cells. This reroutes a defective TSV channel to a spare channel instead of performing a shift-left or shift-right operation for all TSV channels [1], [3], [7]. A multiplexer is used to choose a data line for which there is a defective TSV. The multiplexer logic can be implemented using a standard cell, pass transistor, or primitive logic gates. If a pass transistor is used, the pass transistor logic needs to be added to all selection bits of the multiplexer for delay balancing. Figure 4 shows an example in which the third TSV is determined to be defective. Hence, the selection bit for the third TSV is assigned to 1, and the other TSVs have 0 as their selection bit. In a similar manner, the data receiving side has the same selection bit setting, which is used for de-multiplexing.

For a static self-repair, the testing and self-repair of a TSV operation can occur during the manufacturing to identify defective TSVs. A fusing technique can be applied to permanently write values for a defective TSV channel re-routing. A fused boundary scan chain stores selection bits for the multiplexer and de-multiplexer.

4. Code-Based Self-Repair Architecture

Self-repair logic can dynamically perform the repair process regardless of the location of the defects. Self-repair logic such as an error-correcting code (ECC) can be adopted. For example, the ECC approach can be used for the self-repair logic. In this approach, on the data transmitting side, a check bit (or other ECC) is generated based on the data to be transferred by the TSVs. The data are transferred through the TSVs, with the check bits being transferred through spare TSVs. Figure 5(a) illustrates a general self-repair architecture implementation using the generation of an error-correction code. On the data receiving side, decoding logic decodes the data (raw data + check bit(s)) and corrects the data regardless of a defective TSV channel. The general self-repair architecture of the data receiving side is shown in Fig. 5(b). Error-correction logic or a syndrome generator is implemented to perform the repair operation. Hence, even though there is a defective TSV, the error-correcting logic corrects the corrupted data from the channels and thus provides a replacement of the TSV operation. Since this approach does not require multiple multiplexers or de-multiplexers, it reduces the use of logic around the TSVs, and helps minimize the TSV stress impact and reliability influence.

Multiple kinds of self-repair logic may be utilized in the system, with ECCs and error-detecting codes being common examples. For example, for a single-bit ECC, check bits are generated using the data word. If the size of the data word is D, and the number of required check bits to have single-error correction and double-error detection (SEC-DEC) capability is C, then C is determined when D and C meet the requirements of $2^C \geq D + C + 1$. Hence, if the data words are 32 bits, 64 bits, and 128 bits, then 6, 7, and 8 check bits are required to perform a single-error correction, respectively. All 32 TSVs, 64 TSVs, or 128 TSVs may thus have 6, 7, or 8 spare TSVs to perform the repair process using an ECC.

In certain implementations, errors may be detected and corrected, or detected and not corrected, such as when there are excessive numbers of defective TSVs. For example, logic may provide for SEC-DED, single-error correction and double-adjacent error correction (SEC-DAEC) [1], or other correction and detection operations. In one example, SEC-DAEC may be particularly useful during TSV operation because defects in a device may cause issues for adjacent TSVs, and there may therefore be a particular value in correcting double-adjacent errors. It should be noted that the proposed code-based self-repair architecture can be used to implement any general codes. The architecture of the data transmitting and receiving side (check bit generation and syndrome generator architectures) will be the same for any exiting codes.

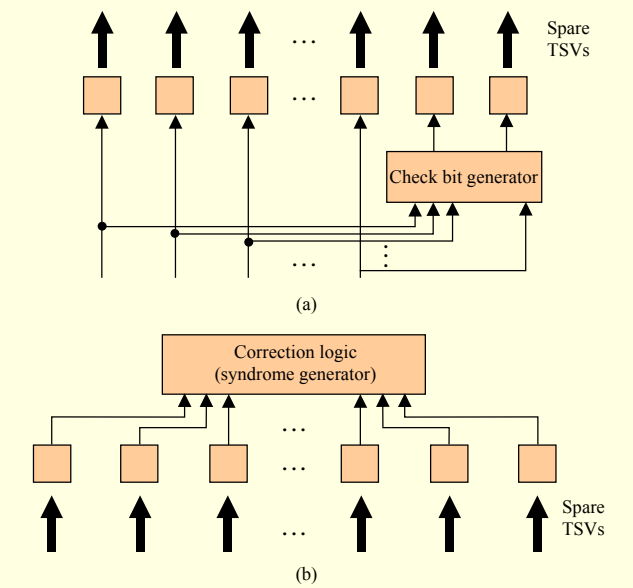

Fig. 5. Code-based self-repair architecture: (a) data transmission side architecture and (b) data receiving side architecture.

IV. Analysis and Experimental Results

In this section, an analysis of the proposed dynamic TSV repair approaches is presented.

In this paper, we propose two dynamic repair architectures. Regardless of the implementations, at a system reset or power good, the μCode (micro-code) or initial firmware initiates a test of the TSVs to identify defective TSV channels. The code-based method or hardware-mapping-based approach is selected depending on the system application. For example, if a fault tolerant operation is focused, the code-based approach will be chosen because checker bits enhance the reliable operation when there are no defective TSVs. It should be noted that both approaches can also be implemented together to further enhance the reliable TSV and repair operations.

To analyze the TSV impact on transistors, we used the configuration used in [10] and [16]. There are many factors influencing the reliability of TSVs such as distance and degree between a transistor and a TSV channel. Transistor mobility ($\Delta\mu/\mu$) and threshold voltage ($\Delta V_{th}/V_{th}$) are mainly influenced by the TSV stress.

$$\frac{\Delta\mu}{\mu} = -\Pi \times \sigma_{rr} \times \alpha(\theta), \text{ where } \sigma_{rr} = -\frac{B\Delta\alpha\Delta T}{2}\left(\frac{R}{r}\right)^2.$$

Mobility variation modeling is based on the *Lamé* stress solution [16], which is a representation of stress state vectors. Here, $\alpha(\theta)$ is an orientation factor defined as the degree between the TSV and a transistor, B is the biaxial modulus, $\Delta\alpha$ is the thermal expansion coefficient difference between

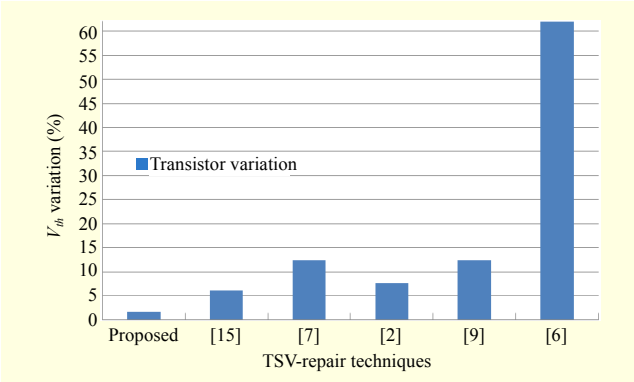

Fig. 6. Comparison of transistor variation with different TSV-repair approaches.

copper and silicon, ΔT is the temperate difference between copper annealing and operation, and R and r are the TSV radius and distance from the TSV to a transistor, respectively. The threshold voltage variation modeling can be described as follows.

$$\Delta V_{th}(\theta) = -m\Delta E_C + (m-1)\Delta E_V,$$

where ΔE_C and ΔE_V are the energy changes in the silicon conduction band and valence band, respectively, and m is the body-effect coefficient [17]. For comparison purposes, mobility variations and threshold variations are considered to capture the variation effects on the transistors. Assuming the same TSV radius and distance from the TSV to a transistor, transistor variations for different repair architectures are compared using the above equations. Figure 6 shows the stress level of the transistors around the TSV with different architectures. The X-axis and Y-axis show different TSV repair approaches, and threshold voltage variations normalized by the proposed method, respectively. The proposed method shows the least impact on the transistors by the TSV channels, and an NoC-type architecture [6] shows the highest stress on the transistors caused by adjacent TSVs.

Table 1 shows various aspects of TSV repair methods. Comparison results for H/W overhead, routing congestion, and dynamic repair capability are described. The proposed method provides only a dynamic repair approach through a BIST operation to identify the failing TSV channels. Hardware overhead for each technique is estimated based on the number of multiplexers (MUXes) used to implement the repair hardware. Assuming that there are n TSVs and one spare TSV (a total of n + 1 TSVs), the second row in Table 1 shows the number of MUXes required. Routing heavily depends on the placement of TSV channels in 3D SICs. The third row roughly shows the estimated routing congestion based on the regular TSV channel placement assumption.

Table 1. TSV-repair architecture comparisons.

	[15]	[7]	[2]	[9]	[6]	Proposed
Dynamic repair	N/A	N/A	N/A	N/A	N/A	Yes
Num. of MUXes required	$2(n+1)/5$	$2(n-1)$	$2n$	n	$4n^2$	2
Routing	Low	Low	Low	Med	High	High

V. Conclusions

In this paper, a new BIST is proposed. The new BIST feature helps to identify defective TSV channels, which may occur during any lifespan of 3D SICs. This enables dynamic TSV repair by providing TSV channel test results at every power-up or reset. The proposed BIST operation is compatible with conventional repair approaches.

We also introduced self-repair architectures for defective TSVs identified through a BIST operation. Hardware-mapping-based and code-based approaches were proposed for SoCs. As the experimental results show, the proposed self-repair methods significantly minimize TSV stresses and reliability problems by placing repair logic only near a spare TSV. Hence, the proposed repair architectures help to enhance the serious yield issues arising in 3D SICs.

References

[1] T.-C. Huang, "High-Performance Built-in Self-Routing for Through-Silicon Vias," *Electron. Lett*., vol. 48, no. 9, Apr. 2012, pp. 480-482.

[2] H.-Y. Huang, Y.-S. Huang, and C.-L. Hsu, "Built-in Self-Test/Repair Scheme for TSV-Based Three-Dimensional Integrated Circuits," *Proc. IEEE Asia Pacific Conf. Circuits Sys. (APCCAS)*, Kuala Lumpur, Malaysia, Dec. 6-9, 2010, pp. 56-59.

[3] U. Kang et al., "8 Gb 3D DDR3 DRAM Using Through-Silicon-Via Technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, Jan. 2010, pp. 111-119.

[4] L. Jiang, R. Ye, and Q. Xu, "Yield Enhancement for 3D-Stacked Memory by Redundancy Sharing Across Dies," *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD),* San Jose, CA, USA, Nov. 7-11, 2010, pp. 230-234.

[5] L. Jiang et al., "Modeling TSV Open Defects in 3D-Stacked DRAM," *Proc. IEEE Int. Test Conf. (ITC)*, Austin, TX, USA, Nov. 2-4, 2010, pp. 1-9.

[6] L. Jiang, Q. Xu, and B. Eklow, "On Effective TSV Repair for 3D-Stacked ICs," *Proc. Conf. Exhibition (DATE)*, Dresden, Germany, Mar. 12-16, 2012, pp. 793-798.

[7] D. Velenis, E.J. Marinissen, and E. Beyne, "Cost Effectiveness of 3D Integration Options," *Proc. IEEE Int. 3D Sys. Integr. Conf. (3DIC)*, Munich, Germany, Nov. 16-18, 2010, pp. 1-6.

[8] M. Kawano et al., "Three-Dimensional Packaging Technology for Stacked DRAM with 3-Gb/s Data Transfer," *IEEE Trans. Electron Devices*, vol. 55, no. 7, July 2008, pp. 1614-1620.

[9] R.S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs," *Proc. IEEE*, vol. 94, no. 6, June 2006, pp. 1214-1224.

[10] L. Yu et al., "Methodology for Analysis of TSV Stress Induced Transistor Variation and Circuit Performance," *Proc. Int. Symp. Quality Electron. Des. (ISQED)*, Santa Clara, CA, USA, Mar. 19-21, 2012, pp. 216-222.

[11] M. Pathak et al., "Electromigration Modeling and Full-Chip Reliability Analysis for BEOL Interconnect in TSV-Based 3D ICs," *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, San Jose, CA, USA, Nov. 2011, pp. 555-562.

[12] Q. Xu et al., "Yield Enhancement for 3D-Stacked ICs: Recent Advances and Challenges," *Proc. Asia South Pacific Des. Automation Conf. (ASP-DAC)*, Sydney, NSW, Australia, Jan. 30 - Feb. 2, 2012, pp. 731-737.

[13] E.J. Marinissen and Y. Zorian, "Testing 3D Chips Containing Through-Silicon Vias," *Proc. IEEE Int. Test Conf. (ITC)*, Austin, TX, USA, Nov. 1-6, 2009, pp. 1-11.

[14] A.P. Karmarkar, X. Xu, and V. Moroz, "Performance and Reliability Analysis of 3D-Integration Structures Employing Through Silicon Via (TSV)," *Proc. IEEE Int. Rel. Physics Symp.*, Montreal, QC, Canada, Apr. 26-30, 2009, pp. 682-687.

[15] M. Jung et al., "TSV Stress-Aware Full-Chip Mechanical Reliability Analysis and Optimization for 3D IC," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 31, no. 8, Aug. 2012, pp. 1194-1207.

[16] J.-S. Yang et al., "Robust Clock Tree Synthesis with Timing Yield Optimization for 3D-ICs," *Proc. Asia South Pacific Des. Automation Conf. (ASP-DAC)*, Yokohama, Japan, Jan. 25-28, 2011, pp. 621-626.

[17] A. Dutta and N.A. Touba, "Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code," *Proc. IEEE VLSI Test Symp. (VTS)*, Berkeley, CA, USA, May 6-10, 2007, pp. 349-354.

Joon-Sung Yang received his BS from Yonsei University, Seoul, Rep. of Korea in 2003, and his MS and PhD degrees in electrical and computer engineering from the University of Texas at Austin, Austin, TX in 2007 and 2009, respectively. Upon graduation, he worked in Intel Corporation, Austin, TX, USA for three years. He is an assistant professor at Sungkyunkwan University in Korea. He was the recipient of a Korea Science and Engineering Foundation (KOSEF) Scholarship in 2005. He received a Best Paper Award at the 2008 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems and was nominated for Best Paper Award at the 2013 IEEE VLSI Test Symposium. His research interests are VLSI testing, silicon debug and nanometer scale test and design methodologies.

Tae Hee Han received his BS, MS, and PhD degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1992, 1994 and 1999, respectively. From 1999 to 2006, he had been with the Telecom R&D center of Samsung Electronics, where he developed 3G wireless, mobile TV, and mobile WiMax handset chipsets. Since March 2008, he has been at Sungkyunkwan University, Suwon, Rep. of Korea as an associate professor. His current research interests include Networks-on-Chip, 3D IC technologies, wearable system architecture, and embedded software. From 2011 to 2013, he had been working as Program Director under Ministry of Knowledge Economy, Korea Government, in the field of system semiconductors.

Darshan Kobla is a Micro Architect with DFx CoE team at Intel's Atom Product Development business unit. He is responsible for overall micro-architecture in the areas of design for debug and structural testing across multiple Atom SoCs for the tablet and smartphone market segments. He received his MS in electrical engineering from Arizona state university, Tempe, AZ in 2005. His research interests are SoC debug, low-cost DFT, 3-D test and adaptive testing. He has five pending US patents filed in 2012.

Edward L. Ju is an engineering manager of DFx CoE (Center of Excellence) at Intel's Atom-SoC Product Development business. He is responsible for overall DFx methodologies, architecture/micro-architecture, RTL implementation, and pre-silicon validation across multiple Atom SoCs for the tablet and smartphone market segments. He received his BS degree in electrical

and computer engineering from National Taiwan University, Taipei, Taiwan, in 1990, and his MS in electrical and computer engineering from the University of Texas at Austin, Austin TX, in 1995. He was admitted to the PhD program and completed all coursework at the University of Texas at Austin, Austin TX, while he was working full-time at Motorola during 1999 to 2001. He has two US patents pending filed in the last two years. His research interests are SoC test strategy, SoC system/platform/form-factor debug, and TSV test.

ETRI Journal, Volume 36, Number 2, April 2014
http://dx.doi.org/10.4218/etrij.14.0113.0578