Yield Enhancement Techniques for 3D Memories by Redundancy Sharing among All Layers

Joohwan Lee, Kihyun Park, and Sungho Kang

Three-dimensional (3D) memories using through-silicon vias (TSVs) will likely be the first commercial applications of 3D integrated circuit technology. A 3D memory yield can be enhanced by vertical redundancy sharing strategies. The methods used to select memory dies to form 3D memories have a great effect on the 3D memory yield. Since previous die-selection methods share redundancies only between neighboring memory dies, the opportunity to achieve significant yield enhancement is limited. In this paper, a novel die-selection method is proposed for multi-layer 3D memories that shares redundancies among all of the memory dies by using additional TSVs. The proposed method uses three selection conditions to form a good multi-layer 3D memory. Furthermore, the proposed method considers memory fault characteristics, newly detected faults after bonding, and multiple memory blocks in each memory die. Simulation results show that the proposed method can significantly improve the multi-layer 3D memory yield in a variety of situations. The TSV overhead for the proposed method is almost the same as that for the previous methods.

Keywords: Yield enhancement, multi-layer 3D memory, redundancy sharing, inter-die redundancy, die-selection method.

I. Introduction

Three-dimensional (3D) integrated circuits (ICs) with through-silicon vias (TSVs) were introduced to overcome the well-known wall problems of two-dimensional (2D) ICs, such as interconnect problems [1], [2]. Memory plays an important role in high performance systems and will likely be the first commercial application of 3D IC technology [3], [4]. A 3D memory is implemented with TSVs as vertical buses across memory layers. Thus, the 3D memory can reduce memory access latency and increase memory access bandwidth [3].

Since a 3D memory has an extremely high capacity and density, defects are easily introduced during the manufacturing of multi-layer 3D memories. Furthermore, additional defects can arise during bonding, which results in a yield drop and quality degradation [5], [6]. Memory repair is used for improving the yield of 3D memories as well as that of 2D memories [7]-[17]. To effectively repair memories, a number of redundancy analysis (RA) algorithms have been presented [11]-[17].

A memory die typically contains many memory blocks and spare rows and columns are employed for each memory block. After the memory repair process is completed, a memory block can be categorized into one of two basic types: a repairable memory block or an irreparable memory block. If a memory block is repairable, unused redundancies may remain. However, if a memory block is irreparable, the number of redundancies is insufficient to repair the block. Unfortunately, the entire memory die must be discarded if just one of its memory blocks is irreparable. Intuitively, a memory yield can be increased by letting memory blocks share the precious spare rows and columns. In [11], the unused redundancies are shared to increase the memory yield in traditional 2D memories, but

Manuscript received Oct. 10, 2011; revised Nov. 29, 2011; accepted Dec. 12, 2011.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2010-0026822).

Joohwan Lee (phone: +82 2 2123 2775, eldsich@yonsei.ac.kr), Kihyun Park (cimabear@yonsei.ac.kr), and Sungho Kang (corresponding author, shkang@yonsei.ac.kr) are with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.12.0111.0643

high routing overhead makes this strategy unpopular for 2D memories. However, in 3D memories, redundancy sharing among vertical memory layers is a practicable and effective strategy because the short electrical path length of TSVs can make routing easy. When inter-die redundancies are used, a memory block that is not self-repairable can borrow redundant resources from its vertical memory blocks and may be repairable after bonding. Thus, the 3D memory yield can be significantly enhanced with the use of inter-die redundancies, when compared with no use of inter-die redundancies.

With the vertical redundancy sharing strategy, die-selection methods are increasingly important. For example, when a 3D memory is composed of two memory dies, if a self-repairable memory die is bonded with a memory die that is not self-repairable but they cannot form a good 3D memory, the 3D memory yield might even be sacrificed. Therefore, to enhance the 3D memory yield, the selection of adequate memory dies is both important and relevant. Recently, three die-selection methods [8]-[10] have emerged for yield enhancement in two-layer 3D memories. Chou and others [8] exactly matched one die to another with inter-die redundancies. Specifically, the number of unused redundancies in a self-repairable memory die was the same as the number of insufficient redundancies in a memory die that was not self-repairable. Jiang and others [9] selectively matched memory dies together using irrespective sub-bipartite graphs. Lee and others [10] selected memory dies to manufacture 3D memories using three search-space conditions (and this method is the die-selection method that we previously used). The die-matching algorithms in [8], [9], and [10] enhanced the 3D memory yield. However, although these die-selection methods enhance the 3D memory yield, they can only share redundancies between neighboring vertical memory dies.

Sharing vertical redundancies among multiple memory dies would enable significant yield enhancement. Namely, when a memory die that is not self-repairable is not repaired by sharing redundancies with a self-repairable memory die, it may be repaired by sharing redundancies with multiple self-repairable memory dies. Furthermore, the previous methods [8], [9] do not consider the memory fault characteristics, and the methods [8]-[10] do not consider additionally detected faults after bonding or multiple blocks in a memory die.

If redundancies are shared among all of the memory dies, additional TSVs are required when compared with the previous methods [8]-[10]. However, TSV overhead is less of a concern because of the continuing improvement of TSV manufacturing technology. The number of TSVs can also be reduced without a severe 3D memory yield loss by decreasing the redundancy sharing ratio of global spares to local spares.

In this paper, we propose an efficient die-selection method for yield enhancement in multi-layer 3D memories. Unlike the previous methods [8]-[10], the proposed die-selection method shares redundancies among all of the memory dies within a multi-layer 3D memory. Furthermore, since the proposed method considers both memory fault characteristics and additionally detected faults, its 3D memory yield can be higher than those of previous methods. The proposed method also takes multiple blocks into account, which is practical because a memory die contains a number of memory blocks.

The rest of this paper is structured as follows. Section II introduces background information related to the proposed die-selection method. Section III illustrates the proposed die-selection method with a simple example to provide a more detailed explanation of the method. The fault distributions used in simulations, the yield of 3D memories in various situations, and the TSV overhead are shown and analyzed in section IV. Finally, section V concludes the paper.

II. Background

1. 3D Memory Stacking

There are three basic ways to stack 3D memories [8]-[10], [18]: wafer-to-wafer (W2W), die-to-wafer (D2W), and die-to-die (D2D). Each integration method has advantages and disadvantages. W2W integration technology has a simple manufacturing process and creates hundreds or thousands of memories at once. However, its yield of 3D memories can be quite low because W2W integration technology cannot use known-good-die information. On the other hand, D2W and D2D integration technologies require more complex manufacturing processes. However, the 3D memory yield from the D2W and D2D methods can be much higher than that of the W2W method because of the use of known-good-die information. In this paper, we assume 3D memory stacking with D2D integration technology. However, D2W integration technology can also be used to build 3D memories by adding the position constraint of a die in a wafer.

In 3D memories, memory dies are stacked vertically upon each other and TSVs are utilized as buses to link the stacked dies together, as shown in Fig. 1. This organization enables redundancy sharing across memory dies using short TSVs while causing very little overhead routing.

2. Memory Die Classification

After the pre-bond test and repair, memory dies with inter-die redundancies are classified into the following four types: fault-free, self-repairable, inter-repairable, and irreparable. A fault-free die has no faults and uses no redundancies. A self-repairable die can be repaired with its self-contained

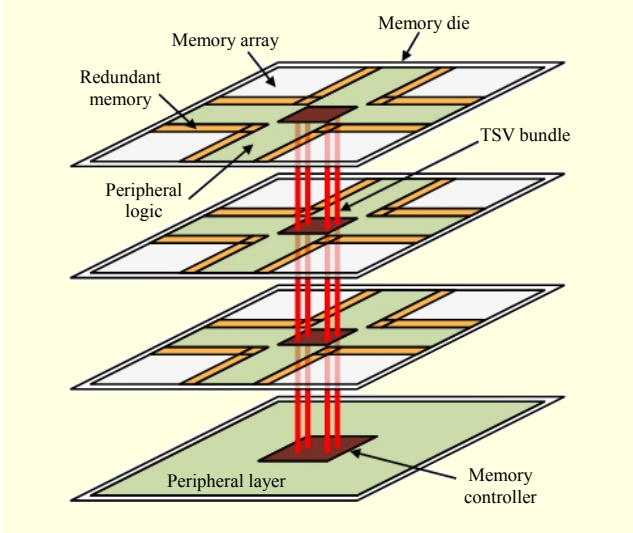


Fig. 1. Overview of 3D memory.

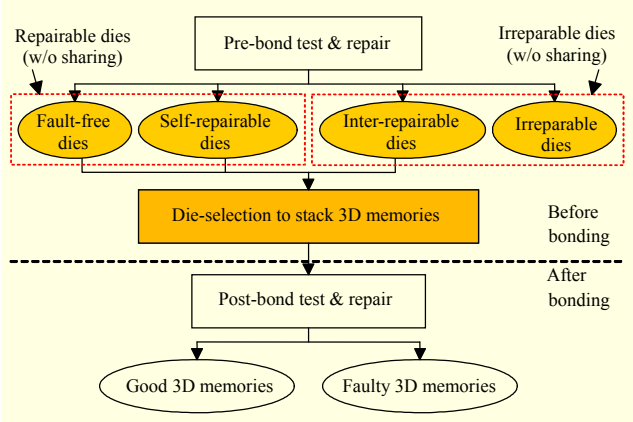


Fig. 2. Procedure to manufacture 3D memories with inter-die redundancies.

redundancies and may leave unused redundancies. An inter-repairable die cannot be repaired with the self-contained redundancies, but can be repaired by the inter-die redundancies. An irreparable die cannot be repaired by either the self-contained redundancies or the inter-die redundancies.

Without the use of inter-die redundancies, a 3D memory can only be composed of repairable dies, that is to say fault-free dies and self-repairable dies. However, 3D memories using inter-die redundancies are stacked with fault-free dies, self-repairable dies, and inter-repairable dies. Therefore, the redundancy sharing strategy can significantly enhance the 3D memory yield.

The procedure to manufacture 3D memories with inter-die redundancies based on D2D integration technology is shown in Fig. 2. The procedure in Fig. 2 is similar to the stacking flow in [8]. However, four die types are used in Fig. 2 instead of three die types in [8]. In Fig. 2, first of all, memory dies are classified according to their states after the pre-bond test and repair. Secondly, the classified dies are carefully selected for stacking as 3D memories since a die-selection can significantly affect the 3D memory yield. After bonding, the post-bond test and repair is carried out to ensure the reliability of the 3D memory. Good 3D memories, which do not have any faults after the test and repair, are then shipped while faulty 3D memories are scrapped.

3. Memory Fault Characteristics

The majority of memories popularly use spare row lines and spare column lines [12]-[17]. A memory with spare row lines and spare column lines obeys the line replacement policy [14], [17], which dictates that any fault in a memory has to be replaced with a spare line. To form an operational 3D memory, all of the faults should be allocated to spare row lines or spare column lines. Among the faults assigned to spare lines, there are single cell faults, which do not share row and column addresses with other faults. A single cell fault can be repaired by either a spare row line or a spare column line. Therefore, as in 2D memory repair [14], [17], the repair decision for single cell faults can be postponed until the dies have been properly selected to assemble a 3D memory. This characteristic of single cell faults is used to enhance the 3D memory yield in the proposed die-selection method.

III. Proposed Yield Enhancement Techniques

1. 3D Memory Architecture with Multi-layer Redundancy Sharing

Two types of redundancies have been considered when repairing defective memories with inter-die redundancies. The shift reconfiguration mechanism is used to exchange a defective element with an inter-die redundancy, as described in [8]. In [9] and [10], both a programmable decoder and multiplexers are used for redundancy sharing. However, these existing 3D memory architectures for inter-die redundancies are only used when redundancies are shared between neighboring vertical memory dies. Thus, a new 3D memory architecture is required for sharing redundancies among multiple memory dies.

The inter-die redundancy schemes used in the previous die-selection methods [9], [10] and the proposed die-selection method are depicted in Fig. 3. Redundancies are shared between two layers in the previous methods (Fig. 3(a)), and among multi-layers in the proposed method (Fig. 3(b)). In Fig. 3(a), a spare row and two spare columns in each memory block are connected to the programmable decoder of a neighboring layer using TSVs as well as the decoder in its own layer.

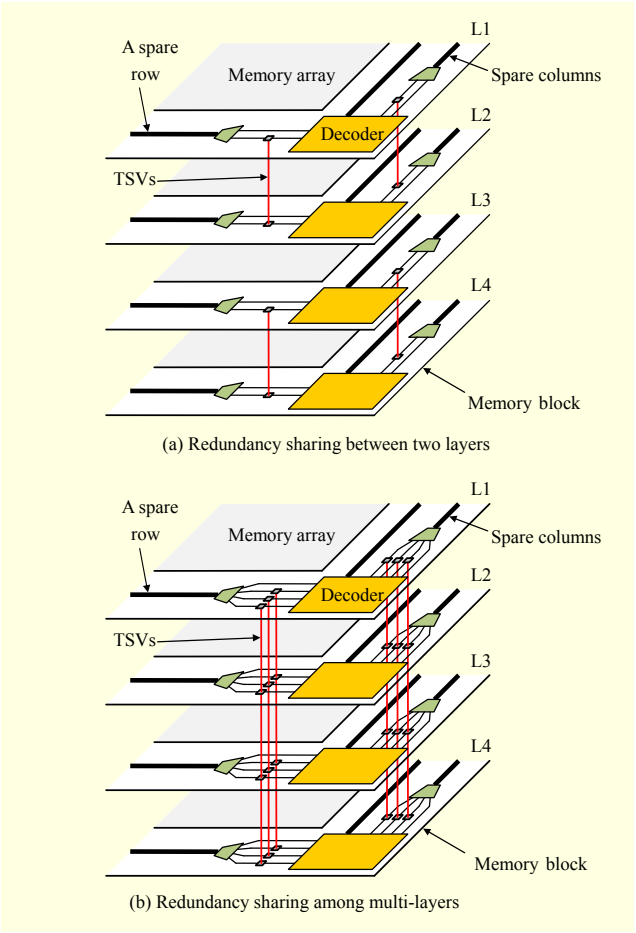


Fig. 3. Inter-die redundancy scheme.

However, in Fig. 3(b), the redundancies are not only connected to the decoder in their own layer but also linked to the decoders of all the other layers. In both cases, the multiplexers determine which memory blocks use the redundancies. The routing overhead to support inter-die redundancies is quite low due to the use of short TSVs.

In the previous methods [9], [10], two global redundancies can be used in the two dies with a common TSV because redundancies are shared between neighboring dies, as shown in Fig. 3(a). However, in the proposed method, nine common TSVs are required for sharing four global redundancies because all the relevant redundancies can simultaneously be used in a die, as shown in Fig. 3(b). Generally, when a memory die contains B memory blocks and when R_S global spare rows and C_S global spare columns are added to each memory block, sharing redundancies between two neighboring dies requires $B \times (R_S + C_S) \times [L/2]$ TSVs to form an L-layer 3D memory. However, the proposed method, which shares redundancies among all the dies, requires $B \times (R_S + C_S) \times (L-1)^2$ TSVs to stack an L-layer 3D memory. For example, in each of the methods illustrated in Figs. 3(a) and 3(b), there is a four-layer 3D memory (L=4) that has a memory block (B=1) with a global spare row (R_S=1) and a global spare column (C_S=1) in each memory die. Thus, the 3D memory that shares redundancies between neighboring dies requires four TSVs, as shown in Fig. 3(a). On the other hand, the 3D memory that shares redundancies among all the dies uses eighteen TSVs, as shown in Fig. 3(b).

The number of TSVs used in the proposed scheme is the same as that used in the previous scheme when the number of layers in a 3D memory is two. But, when the number of layers is greater than or equal to three, the proposed method requires more TSVs to share redundancies among all the dies than the previous method requires. However, with the continuing improvement of TSV manufacturing technology, TSV overhead is less of a concern. Furthermore, since some redundancies can only be used to repair their own blocks without sharing, the number of TSVs can be reduced. In other words, they can be locally used. For instance, in Fig. 3, a spare row and a spare column (global R_S = 1 and global C_S = 1) in each memory block are used for redundancy sharing but a spare column (local C_S = 1) is not shared.

Multi-layer redundancy sharing, in the proposed 3D memory architecture, improves the efficiency of redundancy utilization because global redundancies can be used in all layers. When the proposed scheme is applied, more redundancies are used in each memory block, yet the overall number of redundant resources is unchanged. Using the previous scheme, a memory block can only utilize two spare rows and three spare columns, as shown in Fig. 3(a). Yet that increases to four spare rows and five spare columns with the proposed scheme, as shown in Fig. 3(b). In other words, when the proposed scheme is used, it is possible to share fewer redundancies while maintaining the 3D memory yield. Therefore, the number of TSVs needed for the proposed method can be reduced. Moreover, when there are many memory layers, it is also possible to reduce the number of redundancies. For example, each memory block in Fig. 3(a) can use five redundancies. However, even if a local spare column is not used in Fig. 3(b), each memory block can use eight redundancies, which is still three more than that in Fig. 3(a).

Although the number of TSVs used in the 3D memory can be reduced by sharing fewer redundancies, it is possible that one of the additional TSVs added for the proposed scheme is defective. Since a defective TSV does not transmit a proper signal, the 3D memory yield can possibly be dropped. However, the proposed scheme can tolerate some TSV defects. For example, if one of the three TSVs connecting the global spare columns from the L1 layer to the L2 layer in Fig. 3(b) is defective, the corresponding spare columns can be used as partially global redundancies. As long as a memory block does not require borrowing all the three global spare columns to

become reparable, it is still sufficient.

2. Basic Die-Selection Method

A memory die typically consists of multiple memory blocks. However, to simplify the problem, we assume that a memory die is composed of only one memory block. First, the basic die-selection method, which uses the simplified memory dies, is proposed in this subsection. Then in section III.4, we discuss the advanced die-selection method for multiple memory blocks.

The redundancy sharing ratio of global spares to local spares greatly affects the 3D memory yield. With different redundancy sharing ratios, 3D memory yields are evaluated and compared in section IV. However, for the sake of simplicity, in the rest of this paper, only shared redundancies are considered, unless otherwise specifically stated. In other words, unless otherwise noted, all of the spare rows and columns are used globally.

During the pre-bond test and repair, memory repair information for the proposed die-selection method is collected. Like our previous method [10], the proposed method requires information about the number of faulty row lines, the number of faulty column lines, and the number of single cell faults in each memory block because both methods use the memory fault characteristics. However, the previous method [10] does not consider redundancy sharing among multiple memory dies, additionally detected faults after bonding, or multiple blocks in a memory die. Therefore, the proposed method is different from the previous method [10].

Unlike the previous method [9], the proposed method does not require fault bitmaps. Therefore, the proposed method is cost-effective. Using the collected information, the proposed basic die-selection method selects a target memory die to be stacked and then instantly identifies counterpart memory dies that meet three basic selection conditions. The memory fault characteristics stated in subsection II.3 are reified as the basic selection conditions. These three conditions are as follows:

Basic selection condition 1: $\sum_{k=1}^{m} R_k \le L \times R_S$,

Basic selection condition 2: $\sum_{k=1}^{m} C_k \le L \times C_S$,

Basic selection condition 3:

$$\sum_{k=1}^{m} (R_k + C_k + S_k) \le L \times (R_S + C_S),$$

where m is the present number of stacked memory dies and L is the total number of layers in a 3D memory. R_k, C_k, and S_k represent the numbers of faulty row lines, faulty column lines, and single cell faults for the k-th stacked memory die, respectively.

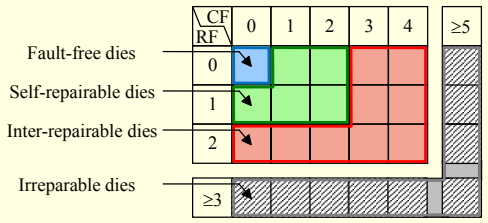


Fig. 4. A memory die classification map ($L = 2$, $R_S = 1$, and $C_S = 2$).

Input: the sorted list of N memory dies
Output: sets of memory dies to form a good 3D memory

```
1  SELECT_DIES (the sorted list) {
2    while (there is die data to be checked in the sorted list) {
3      t_die = SELECT_T_DIE ();
4      for all k, 1 < k ≤ L {
5      the k-th selected die
       = SELECT_C_DIE (t_die, the already selected dies); }
6      if (the selected dies are suitable for a good 3D memory) {
7        OUTPUT_GOOD_DIES (t_die, the selected dies);
8        DELETE_GOOD_DIES (t_die, the selected dies); }
9      else {
10       DELETE_IRREPARABLE_DIE (t_die); }
11     RESET_EXCLUDED_DIES (); } }
```

Fig. 5. Pseudo-code for proposed die-selection method.

If the first basic selection condition is satisfied, then the total number of faulty row lines in the stacked memory dies is fewer than or equal to the total number of spare rows in a 3D memory. Therefore, all the row faulty lines in the stacked memory dies can be repaired. Similarly, if the second basic selection condition is satisfied, all of the column faulty lines in the stacked memory dies can be repaired. When considering single cell faults, the third basic selection condition should also be satisfied to repair the 3D memory. Unlike the previous method [8], the proposed method uses the memory fault characteristics, thus enabling a higher 3D memory yield.

A memory die classification map is used to detect the counterpart memory dies as quickly as possible. Memory repair information for the proposed die-selection method is immediately recorded in the memory die classification map after the memory die is classified. The memory die classification map for a two-layer 3D memory is shown in Fig. 4; each memory die has one spare row and two spare columns. In Fig. 4, *RF* (*CF*) denotes the number of faulty row (column) lines in a memory die after the pre-bond test and repair. Since there are two layers in the 3D memory, the total number of spare rows (columns) is two (four). Thus, if *RF* (*CF*) exceeds two (four), the memory die is irreparable. There

is no need to store irreparable dies in the memory die classification map because irreparable dies are not used for 3D memory stacking. Classified memory dies that are not irreparable are recorded in the memory die classification map.

Pseudo-code for the proposed die-selection method is shown in Fig. 5. *SELECT_DIES* represents the process of selecting memory dies to form 3D memories (line 1). This function does not terminate until all dies are checked (line 2). It uses the sorted list of N memory dies as input, where N is the total number of memory dies to be checked. *SELECT_DIES* outputs sets of memory dies to form a good 3D memory. In *the sorted list*, the memory dies are arranged using the sums of *RF*, *CF*, and *SF*, where *SF* represents the number of single cell faults in a memory die after the pre-bond test and repair, and they are organized in descending order. If the total of *RF*, *CF*, and *SF* is large, the memory die is generally hard to repair. On the other hand, it can be easily repaired when the sum is small. In Fig. 5, if there is die data to be checked in the sorted list, the target memory die (*t_die*) is determined by *SELECT_T_DIE* (line 3). The target die is located at the very front of the sorted list. Once the target memory die is chosen, candidates for its counterpart are analyzed with *SELECT_C_DIE* (lines 4 and 5), which uses the three basic selection conditions and the die classification maps. If there is at least one candidate, the first available memory die in *the sorted list* is selected as the counterpart. This process using *SELECT_C_DIE* is repeated L–1 times (line 4) because L–1 counterpart memory dies are required to stack. If the selected memory dies can stack into a good 3D memory (line 6), they are outputted at *OUTPUT_GOOD_DIES* (line 7) and deleted from *the sorted list* at *DELETE_GOOD_DIES* (line 8). Otherwise, the target memory die is deleted at *DELETE_IRREPARABLE_DIE* (line 10) because it cannot be repaired by redundancy sharing. Finally, the memory die status is reset by *RESET_EXCLUDED_DIES* (line 11). For a worst-case evaluation, the outer loop (line 2) is executed N+1 times and the function (line 5) in the inner loop (line 4) is carried out approximately $(L-1) \times N^2$ times, where N is the number of memory dies in the sorted list, as shown in Fig. 5. Hence, the computational complexity of the proposed method is $O(N^2)$.

An example for the proposed die-selection method is shown in Fig. 6. The proposed method finds the counterpart dies to form two-layer 3D memories using four memory dies with one spare row and two spare columns, as shown in Fig. 6. Memory repair information for four memory dies is displayed in Fig. 6(a). Memory dies are sorted in descending order (dies B-A-D-C), and then they are recorded in the sorted list. With the sorted list, proper memory dies are selected to stack good 3D memories with the proposed die-selection, as shown in Fig. 6(b). In the memory die classification map, *SF* is written as the

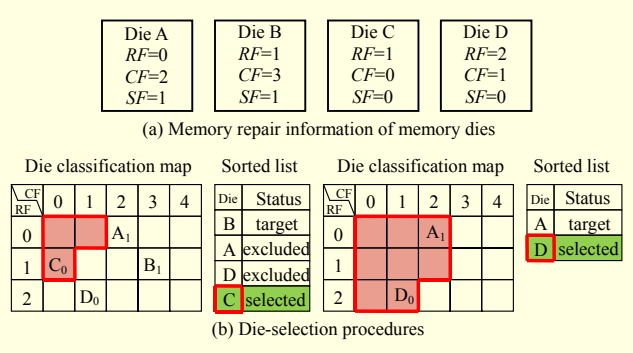


Fig. 6. An example for proposed die-selection method (L = 2, R_S = 1, and C_S = 2).

number positioned at the right bottom of the die. When die A in Fig. 6(a) has a single cell fault, this is written in Fig. 6(b) as die A_1. Similarly, dies B, C, and D can be expressed as dies B_1, C_0, and D_0. In the left side of Fig. 6(b), the inter-repairable die B_1 is selected as the first target memory die since it is located on the first line of the sorted list. Since the three basic selection conditions instantly determine the search space for finding counterpart dies, the self-repairable die C is matched with the first target die B. In the right-hand side of Fig. 6(b), the data of dies B and C is deleted from the sorted list and the die classification map because these dies can make a good 3D memory. The self-repairable die A is matched with the inter-repairable die D to create a good 3D memory stack, in the same way.

3. 3D Memory Repair for Additionally Detected Faults after Bonding

After memory dies are selected to stack good 3D memories, memory dies are bonded to each other. Additional faults can arise during bonding, and they should be repaired during the post-bond test and repair. To repair newly detected faults, unused redundancies are required. However, since the three basic selection conditions do not consider additionally detected faults after bonding, the proposed basic die-selection method cannot guarantee the existence of unused redundancies and the repair of newly detected faults. To prevent the unexpected yield drop, the three basic selection conditions are modified as follows:

Modified selection condition 1: $\sum_{k=1}^{m} R_k \le L \times R_S - R_M$,

Modified selection condition 2: $\sum_{k=1}^{m} C_k \le L \times C_S - C_M$,

Modified selection condition 3:

$$\sum_{k=1}^{m} (R_k + C_k + S_k) \le L \times (R_S + C_S) - S_M,$$

where R_M, C_M, and S_M denote the spare row margin, the spare column margin, and the single cell fault margin, respectively. The margin range can be determined based on the characteristics of the manufacturing process. With the modified selection conditions, the proposed die-selection method can guarantee the 3D memory yield, in spite of additional faults detected after bonding.

4. Advanced Die-Selection Method Considering Multiple Memory Blocks

A memory die is usually composed of multiple memory blocks. If a single memory block is irreparable after bonding, the entire 3D memory is scrapped. Therefore, all the memory blocks in a 3D memory should be repairable. Unlike the previous die-selection methods [8]-[10], our proposed die-selection method specifically considers how to form 3D memories with multiple memory blocks. Thus, the 3D memory yield with the proposed method is higher than that from previous methods, especially when there are many memory blocks. To stack good 3D memories, multiple memory die classification maps are used. Each memory die classification map records memory repair information for a given memory block position, so the number of multiple memory die classification maps is determined by how many memory blocks are in a memory die. A counterpart memory die is determined when the three modified selection conditions are simultaneously satisfied in all the memory die classification maps. The memory die classification maps share the sorted list. The standard sum for ordering is calculated using the hard to repair memory block.

An example of the proposed die-selection method for multiple blocks is shown in Fig. 7; four-layer 3D memories are stacked using six memory dies. The memory dies consist of two memory blocks with one spare row and two spare columns added to each memory block. All-zero margins are used. Memory dies are arranged in descending order (dies A-B-C-D-E-F) and recorded in the sorted list. With the sorted list, proper memory dies are selected to stack good 3D memories with the proposed die-selection, as shown in Fig. 7. In Fig. 7(a), the inter-repairable die A is first selected as the target memory die. Four memory dies (B, C, E, and F) can be used as the counterpart for die A in the memory die classification map of the first memory block; however, only two memory dies (D and F) can be used in the second one. Therefore, die F is selected as the counterpart memory die for die A because it can simultaneously satisfy all three selection conditions. However, since there are no more unchecked memory dies in the sorted list, dies A and F cannot form a 3D memory. After die A is deleted from the sorted list and the die classification maps, the

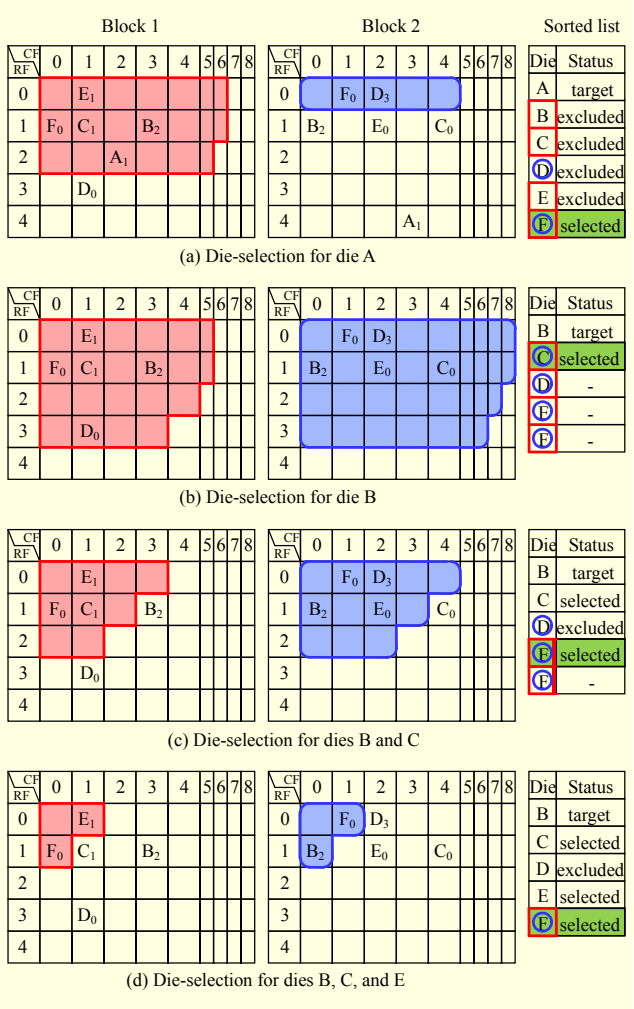


Fig. 7. An example of proposed die-selection method for multiple blocks ($L = 4$, $R_S = 1$, $C_S = 2$, $R_M = 0$, $C_M = 0$, $S_M = 0$, and $B = 2$).

inter-repairable die B is selected as the target memory die. According to the memory die classification maps, all the remaining memory dies can be a counterpart memory die for die B, as shown in Fig. 7(b). However, die C is adopted as the counterpart memory die for die B because it is the most difficult memory die to repair among the remaining memory dies. In other words, using a hard-to-repair memory die first provides more opportunities for stacking 3D memories. Similarly, die E is chosen as the counterpart memory die for dies B and C, as shown in Fig. 7(c). Finally, die F is selected to form a good 3D memory in Fig. 7(d).

IV. Simulation Results and Analysis

To evaluate the performance of the proposed die-selection method, we upgraded our yield enhancement simulator, named *YES*, which had been developed in C language to measure the

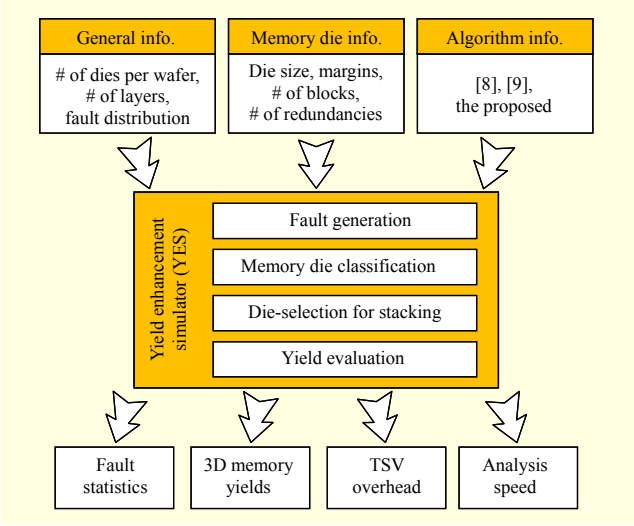


Fig. 8. Overview of a yield enhancement simulator (YES).

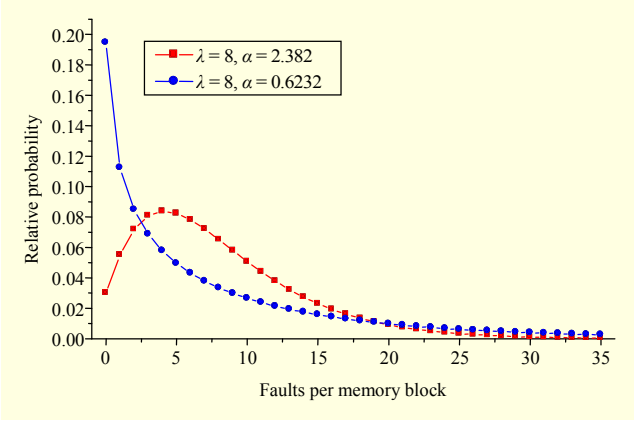


Fig. 9. Polya-Eggenberger distributions with $\lambda = 8$, $\alpha = 2.382$ and $\lambda = 8$, $\alpha = 0.6232$.

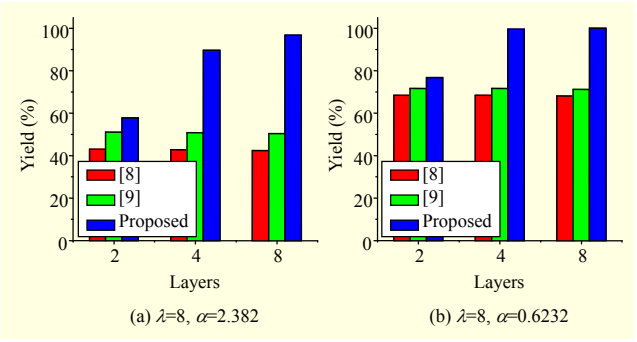


Fig. 10. Comparison of yields with different layers ($R_S = 3$, $C_S = 4$, $R_M = 0$, $C_M = 0$, $S_M = 0$, and $B = 16$).

performance of various die-selection methods during our previous research, as in [10]. The overall diagram of *YES* is shown in Fig. 8. In reference to the information associated with several inputs, *YES* generates faulty addresses. Then, memory dies are classified with their states after the pre-bond test and repair. The classified memory dies are selected for 3D memory stacking. After evaluating 3D memory yields, *YES* generates output data as follows: fault statistics, 3D memory yields, TSV overhead, and analysis speed.

For the comparison of yields, various memory dies are considered for forming 3D memories with 2, 4, or 8 layers. We use 1,000 memory dies to obtain each simulation result. Each memory die contains 4, 8, or 16 memory blocks, and each memory block has 1,024×1,024 bit-cells. Different combinations of spares are used, and the redundancies are vertically shared with different ratios. To handle additional faults detected after bonding, margins are also considered.

In our simulation, a different number of faults is injected into each memory block at random locations using Polya-Eggenberger distribution [9], [19]-[22], as shown in Fig. 9. Polya-Eggenberger distribution is suitable for modeling integrated circuit yields [19]-[22]. The two distributions share the parameter λ and have different parameters for α: (a) Polya-Eggenberger distribution with $\lambda = 8$ and $\alpha = 2.382$; (b) Polya-Eggenberger distribution with $\lambda = 8$ and $\alpha = 0.6232$. The two distributions represent the cases with clustered faults ($\lambda = 8$ and $\alpha = 2.382$) and evenly distributed faults ($\lambda = 8$ and $\alpha = 0.6232$), respectively. Thus, simulation results are obtained in both cases.

The proposed die-selection method is compared with the two previous die-selection methods [8], [9]. However, since the proposed method is based on the previous method [10], it is not compared with the previous method [10]. Without the additional considerations (redundancy sharing among multiple memory dies, additionally detected faults after bonding, and multiple blocks in a memory die) in the proposed method, simulation results of the proposed method are identical with those of the previous method [10].

When different numbers of layers are used, the yield of the proposed die-selection method is greater than that of the previous die-selection methods, as shown in Fig. 10. The previous methods [8], [9] share redundancies only between neighboring memory dies. However, the proposed method shares the redundancies among all the memory dies. Therefore, the gaps between the 3D memory yield with the proposed method and the 3D memory yields with the previous methods are great, especially when the number of layers is large. Furthermore, since the proposed redundancy sharing strategy is used, the 3D memory yield using the proposed method increases as the number of layers grows.

Next we compare the 3D memory yield of the proposed method with that of the previous methods, when there are different numbers of memory blocks, as shown in Fig. 11. The yield using the proposed method is higher in all cases. As the number of memory blocks increases, the difference between

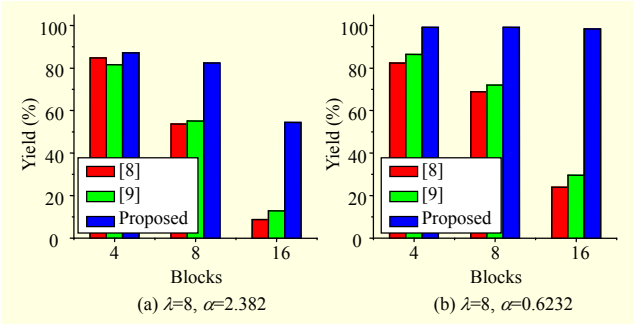


Fig. 11. Comparison of yields with different blocks ($L = 8$, $R_S = 3$, $C_S = 4$, $R_M = 0$, $C_M = 0$, and $S_M = 0$).

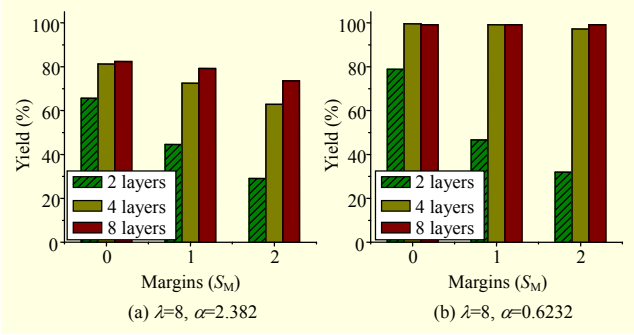


Fig. 12. Comparison of yields with different margins ($R_S = 3$, $C_S = 4$, $R_M = 0$, $C_M = 0$, and $B = 8$).

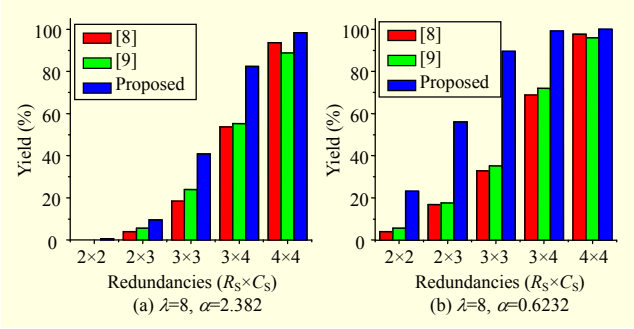


Fig. 13. Comparison of yields with different redundancies ($L = 8$, $R_M = 0$, $C_M = 0$, $S_M = 0$, and $B = 8$).

the 3D memory yield with the proposed method and that with the previous methods grows. Therefore, the proposed die-selection method is very practical.

The 3D memory yield using the proposed method with different margins is shown in Fig. 12. In Fig. 12, the spare row margin (R_M) and spare column margin (C_M) are fixed at zero but the single cell fault margin (S_M) varies from 0 to 2. It is natural that the 3D memory yield is low when the single cell fault margin is large. However, when the number of layers increases, the effect of the margin decreases because the proposed method shares inter-die redundancies among all the memory dies. Thus, the proposed die-selection method has the

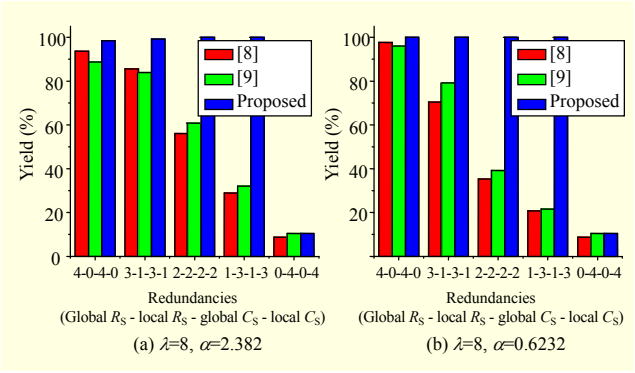


Fig. 14. Comparison of yields with different redundancy sharing ratios ($L = 8$, $R_M = 0$, $C_M = 0$, $S_M = 0$, and $B = 8$).

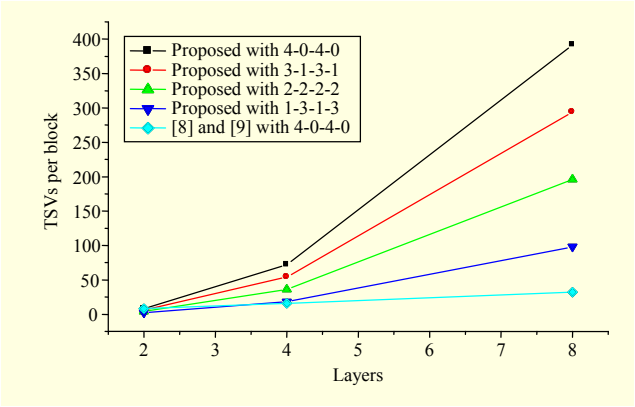


Fig. 15. Comparison of numbers of TSVs in each block with different redundancy sharing ratios.

advantage of repairing additionally detected faults after bonding.

As the number of redundancies increases, the 3D memory yield drastically increases in all cases, as shown in Fig. 13. The average number of faults in each memory die is 8 because the parameter λ of the Polya-Eggenberger distribution is 8. When the same number of redundancies is used, the 3D memory yield with the proposed method is higher than the yields with the previous methods. Therefore, the proposed die-selection method is superior to previous methods.

Different redundancy sharing ratios are used in Fig. 14. When all of the redundancies are shared, the difference of the 3D memory yield is not large. However, as global spares decrease and local spares increase, the gap in the 3D memory yield grows. Furthermore, the 3D memory yield with the proposed method does not drop until all the redundancies become local. Therefore, since there is no need to share all the redundancies in the proposed die-selection method, the TSV overhead can be significantly reduced.

The number of TSVs in each block is shown in Fig. 15 according to three die-selection methods ([8], [9], and the

proposed method) and different redundancy sharing ratios. When all of the redundancies are shared, there is a great gap between the number of TSVs for the previous methods ([8] and [9] with 4-0-4-0, L=8) and that for the proposed method (the proposed with 4-0-4-0, L=8), as shown in Fig. 15. This is because the number of TSVs for implementing L-layer 3D memories with the previous methods [8], [9] is proportional to L but with the proposed method is proportional to $(L-1)^2$. With the continuing improvement of TSV manufacturing technology, this overhead is likely to be less of a concern. However, as the redundancy sharing ratio is decreasing, the difference between the previous methods and the proposed method can be reduced. Nevertheless, if the number of TSVs is critical, a lesser number of layers can be used for redundancy sharing. For example, when four layers are used for sharing and the redundancy sharing ratio is low, the number of TSVs for the proposed method is almost the same as that for the previous methods, as shown in Fig. 15. In this case, the yield of the proposed method is also greater than the yields of the previous methods.

V. Conclusion

A die-selection method using three selection conditions was proposed for yield enhancement in multi-layer 3D memories. Previous die-selection methods share redundancies only between neighboring memory dies. However, the proposed die-selection method shares redundancies among all the memory dies within a multi-layer 3D memory. Furthermore, the 3D memory yield with the proposed method is higher than the yields with previous die-selection methods since the proposed die-selection method considers both memory fault characteristics and additionally detected faults. The proposed die-selection method also takes multiple blocks into account, which is practical because a memory die typically contains a number of memory blocks. Simulation results showed that the proposed die-selection method can further improve 3D memory yields. The TSV overhead for the proposed method is almost the same as that for the previous methods. In conclusion, the proposed die-selection method using three selection conditions effectively enhances the yield of multi-layer 3D memories with inter-die redundancies.

References

[1] V.F. Pavlidis and E.G. Friedman, "Interconnect-Based Design Methodologies for Three-Dimensional Integrated Circuits," *Proc. IEEE*, vol. 97, no. 1, Jan. 2009, pp. 123-140.

[2] W.R. Davis et al., "Demystifying 3D ICs: The Pros and Cons of Going Vertical," *IEEE Design Test Comput.*, vol. 22, no. 6, Nov. 2005, pp. 498-510.

[3] H. Sun et al., "3D DRAM Design and Application to 3D Multicore Systems," *IEEE Design Test Comput.*, vol. 26, no. 5, Sept. 2009, pp. 36-47.

[4] S.S. Iyer et al., "Process-Design Considerations for Three Dimensional Memory Integration," *Proc. Symp. VLSI Tech.*, Jun. 2009, pp. 60-63.

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

[6] H.-H.S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits," *IEEE Design Test Comput.*, vol. 26, no. 5, Sept. 2009, pp. 26-35.

[7] Y.-F. Chou, D.-M. Kwai, and C.-W. Wu, "Memory Repair by Die Stacking with Through Silicon Vias," *Proc. Int. Workshop Memory Tech., Design, and Testing (MTDT)*, Aug. 2009, pp. 53-58.

[8] C.-W. Chou, Y.-J. Huang, and J.-F. Li, "Yield-Enhancement Techniques for 3D Random Access Memories," *Proc. Int. Symp. VLSI Design Automat. Test (VLSI-DAT)*, Apr. 2010, pp. 104-107.

[9] L. Jiang, R. Ye, and Q. Xu, "Yield Enhancement for 3D-Stacked Memory by Redundancy Sharing Across Dies," *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2010, pp. 230-234.

[10] J. Lee, K. Park, and S. Kang, "A Die-Selection Method Using Search-Space Conditions for Yield Enhancement in 3D Memory," *ETRI J.*, vol. 33, no. 6, Dec. 2011, pp. 904-913.

[11] T. Yamagata et al., "A Distributed Globally Replaceable Redundancy Scheme for Sub-Half-Micron ULSI Memories and Beyond," *IEEE J. Solid-State Circuits*, vol. 31, no. 2, Feb. 1996, pp. 195-201.

[12] C.-T. Huang et al., "Built-in Redundancy Analysis for Memory Yield Improvement," *IEEE Trans. Relia.*, vol. 52, no. 4, Dec. 2003, pp. 386-399.

[13] M.-H. Yang et al., "A Novel BIRA Method with High Repair Efficiency and Small Hardware Overhead," *ETRI J.*, vol. 31, no. 3, June 2009, pp. 339-341.

[14] W. Jeong et al., "A Fast Built-in Redundancy Analysis for Memories with Optimal Repair Rate Using a Line-Based Search Tree," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 17, no. 12, Dec. 2009, pp. 1665-1678.

[15] H. Cho, W. Kang, and S. Kang, "A Built-In Redundancy Analysis with a Minimized Binary Search Tree," *ETRI J.*, vol. 32, no. 4, Aug. 2010, pp. 638-641.

[16] T. Han et al., "High Repair Efficiency BIRA Algorithm with a Line Fault Scheme," *ETRI J.*, vol. 32, no. 4, Aug. 2010, pp. 642-644.

[17] W. Jeong et al., "An Advanced BIRA for Memories with an Optimal Repair Rate and Fast Analysis Speed by Using a Branch Analyzer," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 12, Dec. 2010, pp. 2014-2026.

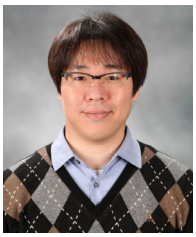
[18] A. Papanikolaou, D. Soudris, and R. Radojcic, *Three Dimensional System Integration: IC Stacking Process and Design*, Springer, 2010.

[19] C.H. Stapper, "On a Composite Model to the IC Yield Problem," *IEEE J. Solid-State Circuits*, vol. 10, no. 6, Dec. 1975, pp. 537-539.

[20] C.H. Stapper, A.N. McLaren, and M. Dreckmann, "Yield Model for Productivity Optimization of VLSI Memory Chips with Redundancy and Partially Good Product," *IBM J. Research Development*, vol. 24, no. 3, May 1980, pp. 398-409.

[21] C.-L. Wey and F. Lombardi, "On the Repair of Redundant RAM's," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 6, no. 2, Mar. 1987, pp. 222-231.

[22] R.-F. Huang et al., "A Simulator for Evaluating Redundancy Analysis Algorithms of Repairable Embedded Memories," *Proc. Int. Workshop Memory Tech., Design, and Testing (MTDT)*, Jul. 2002, pp. 68-73.



Joohwan Lee received his BS, MS, and PhD in electrical and electronic engineering from Yonsei University, Seoul, Rep. of Korea, in 2003, 2005, and 2012, respectively. He was also a research engineer with the university's ASIC Research Center. Currently, he is working in the S. LSI Division of Samsung Electronics as a senior engineer. His main research interests include VLSI design, fault simulation, fault diagnosis, BISR, BIST, BIRA, RA algorithm, reliability, 3D memory yield enhancement, and TSV repair.

Kihyun Park received his BS in electrical and electronic engineering from Yonsei University, Seoul, Rep. of Korea in 2007, then went on to work as a research engineer with the university's ASIC Research Center. Currently, he is working toward a combined MS/PhD in electrical and electronic engineering at Yonsei University. His current research interests include BISR, BIST, BIRA, test algorithms, and VLSI design.

Sungho Kang received his BS in control and instrumentation engineering from Seoul National University, Seoul, Rep. of Korea, and MS and PhD in electrical and computer engineering from the University of Texas at Austin, Austin, USA, in 1992. He was a research scientist with the Schlumberger Laboratory for Computer Science, Schlumberger Inc., and a senior staff engineer with Semiconductor Systems Design Technology, Motorola Inc. Since 1994, he has been a professor in the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Rep. of Korea. His main research interests include VLSI/SOC design and testing, design for testability, and design for manufacturability.