IEEE TRANSACTIONS ON

CIRCUITS AND SYSTEMS

I: REGULAR PAPERS

A PUBLICATION OF THE IEEE CIRCUITS AND SYSTEMS SOCIETY

WWW.IEEE-CAS.ORG

CAS
IEEE CIRCUITS AND SYSTEMS SOCIETY

MAY 2015 VOLUME 62 NUMBER 5 ITCSCH (ISSN 1549-8328)

SPECIAL SECTION ON THE 2014 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS 2014)

GUEST EDITORIAL

SPECIAL SECTION PAPERS

REGULAR PAPERS

Analog and Mixed Mode Circuits and Systems

(Contents Continued on Back Cover)

IEEE

(Contents Continued from Front Cover)

Digital Circuits and Systems and VLSI

Nonlinear Circuits and Systems

Circuits and Systems for Communications

Logic-DRAM Co-Design to Exploit the Efficient Repair Technique for Stacked DRAM

Minjie Lv, Hongbin Sun, *Member, IEEE*, Qiwei Ren, Bing Yu, Jingmin Xin, *Senior Member, IEEE*, and Nanning Zheng, *Fellow, IEEE*

***Abstract*—Three-dimensional (3D) integration is promising to provide dramatic performance and energy efficiency improvement to 3D logic-DRAM integrated computing system, but also poses significant challenge to the yield. To address this challenge, this paper explores a way to leverage logic-DRAM co-design to reactivate unused spares and thereby enable the cost-efficient technique to repair 3D integration-induced defective DRAM cells after die stacking. In particular, we propose to make the DRAM array open its spares to off-chip access by a small architectural modification and further design the defective address comparison and redundant address remapping with an efficient architecture on logic die to achieve equivalent memory repair. Simulation results demonstrate that the proposed repair technique for stacked DRAM can significantly alleviate potential yield loss, with minimal area and power consumption overhead and negligible timing penalty.**

***Index Terms*—Memory repair, redundancy, 3D DRAM, 3D integration, yield.**

I. Introduction

As AN EMERGING technology, three-dimensional (3D) integration provides a viable and promising option to address the well-known memory wall problem in high-performance computing systems. By stacking multiple high-capacity DRAM dies with one or more logic dies and providing massive inter-die interconnect bandwidth with through silicon vias (TSVs), 3D logic-DRAM integrated systems can have drastically reduced memory access latency and increased memory access bandwidth. The performance and power efficiency benefits of the 3D logic-DRAM integrated computing system were extensively evaluated and well demonstrated in many recent works [1], [2]. However, 3D integration poses significant challenge to the yield, which has become the major bottleneck for the mass production of 3D integrated circuits.

Conventionally, the yield of DRAM can be significantly improved by using redundancy repair, which replaces defective cells with spare rows or columns [3]. Two types of one-time programmable fuses, i.e., laser fuse and electrical fuse, are exploited to enable redundancy repair [4]. In general, laser fuse is much more cost-efficient and is thus commonly employed in wafer-level repair, whereas electrical fuse is mainly employed to cope with a small number of failures during post-package repair. However, these two conventional redundancy repair techniques tend to become severely inadequate for 3D stacked DRAM, because 3D stacking always involves many aggressive process steps that tend to incur much more failures than packaging and may even cause a certain amount of cluster defects [5]. The continuous use of laser fuse or electrical fuse in this situation will become either infeasible or inefficient. Therefore, exploiting more cost-efficient repair techniques to reduce the potential yield loss of stacked DRAM has attracted considerable attention [6]–[9]. However, prior works mainly focused on evaluating the potential benefits by sharing the memory array or redundancies across multiple DRAM dies at the architecture level, how to efficiently implement the repair circuit and thus address the yield loss remains unclear.

Actually, once the manufacturing process of DRAM becomes stable, a large percentage of spares equipped with laser fuses will likely be left unused after wafer level repair. These unused spares have been suggested to be used to improve reliability, e.g., to enhance the existing error control coding [10], [11]. Recently, Chou *et al.* [12] proposed to reactivate these unused spares for DRAM repair after die stacking by adding extra controlling circuit and TSVs to each redundant row or column. Although the overall repair scheme and its yield improvement are appealing, the proposed technique inevitably incurs a significant cost in case of a large number of redundancies and imposes a severe limitation on the floorplan of logic die.

This paper is also interested in how to use the aforementioned unused spares to repair DRAM after die stacking in the 3D logic-DRAM integrated system. In particular, it aims to explore a much more cost-efficient solution by fully exploiting the availability of both logic and DRAM dies. Accordingly, we first propose to make the DRAM array open its redundant rows and columns to off-chip access by slight architectural modifications. Then, we further propose to move the defective address comparison and redundant address remapping that are used to be designed on DRAM die into memory controller on logic die, and implement them with an efficient architecture to fully leverage the design flexibility of standard CMOS process. Thus, upon a memory request, the access address is first compared with recorded defective addresses and remapped to redundant rows or columns if necessary on logic die. Both the corresponding normal and redundant DRAM arrays can then be accessed as a common memory access through generic DRAM I/Os. Interestingly, the proposed logic-DRAM co-designed repair technique can inherently enable the global sharing of

Manuscript received October 21, 2014; revised December 29, 2014 and January 31, 2015; accepted February 01, 2015. Date of publication April 15, 2015; date of current version April 28, 2015. This work was partially supported by the National Natural Science Foundation of China (No. 61274028) and National Science and Technology Major Project of China (No. 2013ZX01033-001-001). This paper was recommended by Associate Editor M. Seok.

M. Lv, H. Sun, J. Xin and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: lv.minjie@stu.xjtu.edu.cn; hsun@mail.xjtu.edu.cn; jxin@mail.xjtu.edu.cn; nnzheng@mail.xjtu.edu.cn).

Q. Ren and B. Yu are with Xi'an SinoChip Semiconductors Co. Ltd., Xi'an, Shaanxi, China, 710075 (e-mail: qiwei.ren@scsemicon.com; bing.yu@scsemicon.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2015.2403031

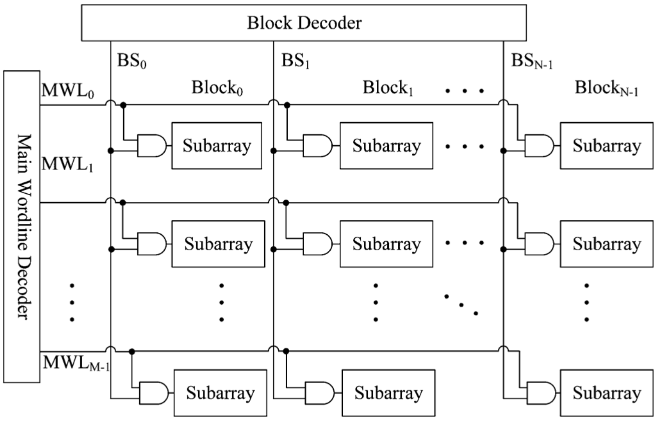

Fig. 1. Hierarchical structure of one DRAM bank.

redundant rows across DRAM dies and hence significantly improve repair capability and efficiency. Evaluation results show that the proposed logic-DRAM co-designed stacked DRAM repair technique can significantly reduce 3D integration-induced yield loss, while its area and power consumption overhead is very low, and its timing penalty is negligible.

The remainder of this paper is organized as follows. Section II reviews the architecture for conventional 2D DRAM and presents the motivation for this work. Section III presents the basic idea and detailed implementation of the proposed logic-DRAM co-designed memory repair architecture. Yield improvement and hardware overhead of the proposed architecture are evaluated in Section IV. Finally, Section V concludes this paper.

II. Background and Motivation

A. Architecture and Repair Flow for Conventional 2D DRAM

In the current large-capacity DRAM design, each memory die generally contains one or several memory banks where each bank is essentially an independent array that has its own address and data bus and can be accessed concurrently. Modern DRAM design usually employs memory array division technique to improve the performance of high-density memory. Thus, DRAM bank is actually organized as a hierarchical structure, where each memory bank is composed of several blocks and each block is composed of dozens of subarrays. Fig. 1 illustrates the hierarchical structure in one memory bank for high-density DRAM [3]. During the DRAM access, several of the most significant bits (MSBs) of the address input are first provided to *block decoder* to select the corresponding memory block, and then several other address bits are provided to *main wordline decoder* to activate the corresponding subarray in the selected memory block. For a DRAM array with wide data bus, a combination of several subarrays can be concurrently activated and accessed [13]. These subarrays share the same address bus and each provides several data bits to form the wide data bus. The hierarchical structure can effectively reduce the parasitic capacitance for signal/noise ratio enhancement and charging/discharging current reduction, and thus improve overall access speed for large-capacity DRAM.

Subarray is the basic element of the above-mentioned DRAM hierarchical structure. Fig. 2 illustrates the memory architecture of one subarray in conventional 2D DRAM [14]. Each subarray has its own address decoder and sense amplifier. Since failure cells are inevitable in the fabrication process of DRAM memory, conventional 2D DRAM subarray is usually designed with several redundant rows or columns to repair the defective cells identified during memory test. Redundancy repair, which replaces faulty memory cells with spare memory cells, is a widely used technique in large-capacity DRAM chips. As shown in Fig. 2, the input address of the DRAM access is connected to both the address decoder and the redundancy address comparator. The redundancy address comparator is also connected to recorded defective addresses, which are provided by an address fuse box. When the address input matches the recorded defective address, the match signal will enable the redundant row or column driver and access the corresponding redundant row or column; otherwise, it turns to access the row and column in normal memory array.

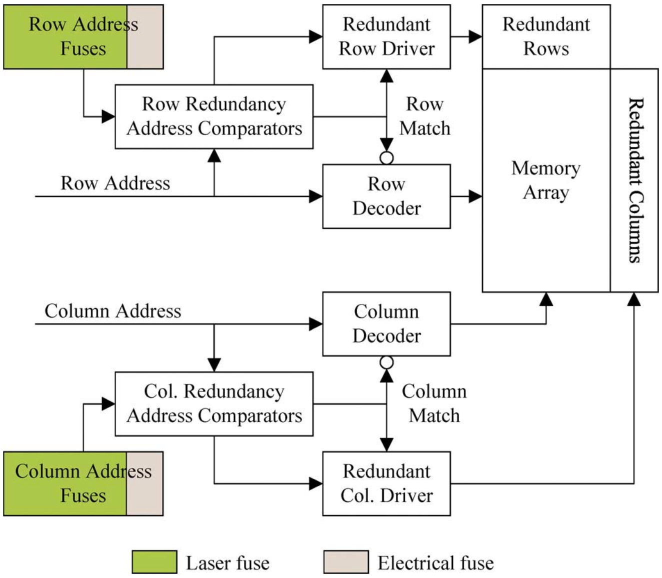

Fig. 2. Memory architecture of one DRAM subarray.

Two types of fuses are used to record the defective addresses, i.e., laser fuse and electrical fuse, both of which are one-time programmable fuses. Laser fuse is blown by using laser energy, whereas electrical fuse is blown by applying a higher-than-normal voltage. These fuses are exploited in different repair procedures [4], as shown in Fig. 3. In general, compared with electrical fuse, laser fuse can be processed significantly faster and is more cost-efficient, hence is mainly employed for wafer-level repair. While electrical fuse is mainly employed for post-package repair, because failures that occur during packaging and burn-in procedures are very few, and laser energy is no longer applicable. In practice, DRAM chip only integrates a few electrical fuses. Most vendors tend to only use electrical fuse to repair single-bit failures [15].

B. Repair Challenges for Stacked DRAM

The aforementioned test and repair flow is adequate for conventional 2D DRAM, but not for 3D stacked DRAM. In general, 3D stacking always involves many aggressive process steps that tend to incur more failures than packaging and assembly and may even cause a certain number of cluster failures [5]. Moreover, since laser machine tools cannot be employed after die stacking, to repair these stacking-induced failures by simply

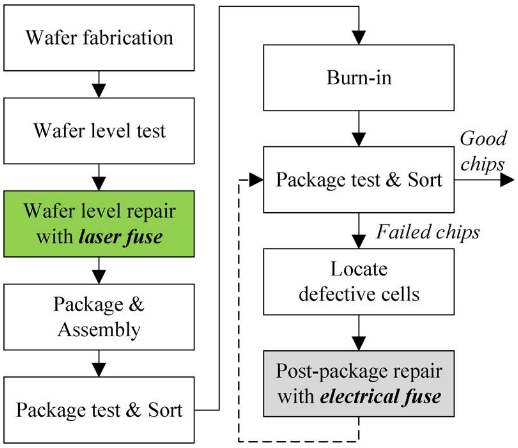

Fig. 3. Test and repair flow for conventional 2D DRAM.

using electrical fuse is neither feasible nor cost-efficient. Failures on the redundancy may also cause those rows and columns that have already been repaired to fail again during die stacking process. Therefore, the conventional repair scheme will become very insufficient for stacked DRAM design, and more efficient repair techniques must be exploited to address these challenges.

Many works have been proposed to reduce the yield loss of 3D stacked DRAM [7]–[9], [16]–[18]. However, these works mainly propose to share the resource across multiple DRAM dies at the architecture level and evaluate the potential benefits on yield improvement. Hence, the cost overhead of these works is usually very high, and how the circuit can be implemented to enable the desired resource sharing is largely unclear. In addition, many flexible and effective repair techniques proposed for memory on logic die seem to also work for stacked DRAM [6], [16]. However, these techniques are inherently designed for small-capacity memory, and the direct use of them for stacked DRAM is very expensive. For example, Axelos *et al.* [6] propose a cache-based redundancy architecture to replace faulty memory words with a set of direct-mapped cache banks. Since this repair technique has to locate the redundancy on logic die, the area overhead will be unacceptable for high-density DRAM, especially when the error rate increases and cluster errors (defective rows/columns) are inevitable.

Since simply repairing on either DRAM die or logic die is inadequate for stacked DRAM, we propose to exploit logic-DRAM co-design to address the aforementioned challenges. In most current 3D circuit design practices, a plurality of DRAM dies are often stacked on logic die. As circuit design on logic die is much more flexible and efficient than that on DRAM die, those memory repair circuits, such as address comparison and remapping, used to be very costly on DRAM die can be implemented very efficiently on logic die. Meanwhile, the redundancy that is very costly in terms of area overhead can be located on the DRAM die, where the memory array is extensively optimized for high density. Actually, DRAM venders usually tend to provide more than enough redundancies to guarantee the high yield because conventional redundant rows/columns located on the DRAM die are restricted within each subarray. Once the DRAM manufacturing process becomes stable and the yield is ramped up, most redundancies equipped for laser-fuse-based repair will likely be left unused and can be further exploited. Chou *et al.* [12] proposed an off-chip memory repair scheme that reactivates unused spares for memory repair by controlling the redundancy repair circuits of memory with TSVs. This scheme is interesting and can effectively compensate for the 3D stacking-induced yield loss. However, it requires to scatteredly allocate a large amount of TSVs and related controlling circuits right beside each redundancy on the logic die. This inevitably incurs significant cost and further imposes severe limitation on the floorplan of logic die. This intuitively motivates us to exploit logic-DRAM co-design to efficiently repair those die-stacking-induced failures with unused spares.

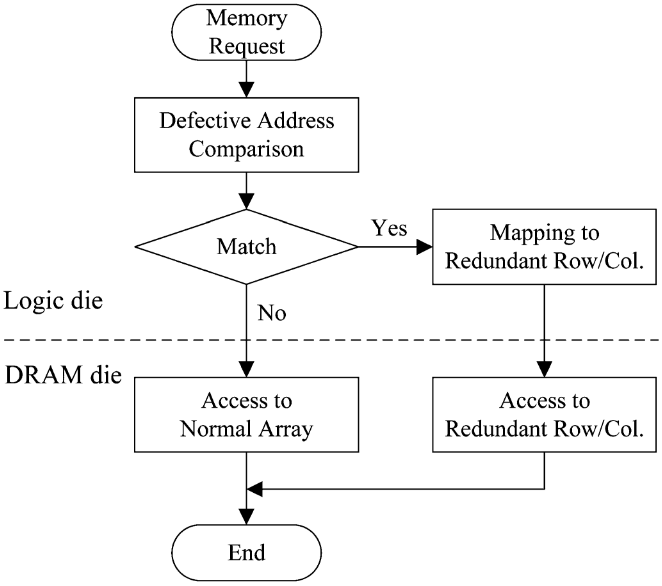
Memory Request
Defective Address Comparison
Match
Yes
Mapping to Redundant Row/Col.
Logic die
No
DRAM die
Access to Normal Array
Access to Redundant Row/Col.
End

Fig. 4. Overall flow of the proposed logic-DRAM co-designed repair technique.

III. Logic-DRAM Co-Designed Memory Repair for Stacked DRAM

A. Overall Co-Designed Architecture

To address the yield loss, we propose a logic-DRAM co-design architecture that enables efficient stacked DRAM repair with unused spares for 3D integrated circuit. The underlying idea can be illustrated as Fig. 4, where defective address comparison and remapping are moved from DRAM die to logic die, and DRAM opens its redundant rows and columns to off-chip access through generic I/Os. Thus, upon a memory request, the access address is first compared with recorded defective addresses and remapped to a redundant row or column if necessary. Then, we can access the corresponding normal or redundant array by using a common memory access through generic DRAM I/Os to achieve the equivalent memory repair with unused spares.

To implement the proposed logic-DRAM co-designed repair architecture, we have to exploit the corresponding architecture and circuit-level design in DRAM array, memory controlling logic, and the interface between them. This procedure will be described in the following subsections. Exploiting the logic-DRAM co-design can significantly alleviate the yield loss while considerably lowering the design overhead compared with that in previous repair techniques.

In particular, the proposed logic-DRAM co-design architecture can easily enable the global sharing of redundant rows,

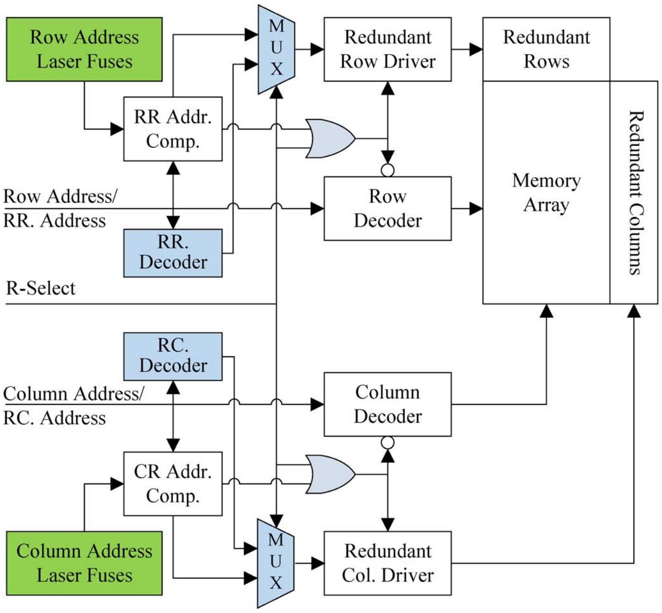

Fig. 5. Modified DRAM subarray architecture enabling the off-chip access to redundant rows and columns.

which can significantly enhance the capability and efficiency of redundancy repair. Since redundancy is located inside each subarray, repair failure may occur when the local error rate is high and the redundant rows or columns in certain subarray are used up, even though the number of redundant rows or columns in other subarrays remains sufficient. Therefore, inter-subarray redundancy sharing has the potential to significantly improve the repair rate and has been exploited in previous works. However, the previously proposed redundancy-sharing designs should only be among spatial adjacent subarrays in one DRAM die [19], [20] or several vertical-stacked DRAM dies [8], [21]. Hence, their improvements on repair rate are also very limited. In the proposed co-designed repair architecture, because of the design flexibility enabled by moving the defective address comparison and remapping from the DRAM die to the logic die, the DRAM repair can utilize all the available redundancies from any subarray on any die. This can significantly improve the repair rate, as shown in the evaluation section.

B. Modified Architecture for DRAM Array

Opening redundant rows and columns to off-chip access is the key to the proposed logic-DRAM co-designed stacked DRAM repair. As described in Section II, the high-density DRAM array usually employs memory array division technique and is organized as a hierarchical structure. The redundant rows and columns are located inside each subarray, which has its own address decoder and sense amplifier. Hence, opening redundant rows and columns to off-chip access is mainly due to the design of the DRAM subarray. Actually, the design effort required to open redundant rows and columns is relatively simple and may only need minimal modifications. Fig. 5 illustrates the modified DRAM subarray architecture, which enables off-chip access to redundant rows and columns by simply adding two small address decoders, two MUXs, and two OR gates.

In the proposed DRAM array architecture, accessing normal memory array or redundant rows/columns is mainly controlled by *R-Select*. Let us take row activation as an example to explain how this architecture works. When we activate redundant rows, the R-Select is set. Hence, the row decoder to normal memory array is disabled and the redundant row driver is enabled. The additional MUX controlled by R-Select will also chooses the redundant row (RR) address to activate the specific redundant row. While when normal memory rows are activated, we reset R-Select and enable the row decoder to normal memory array. In addition, the row address bus provides the normal row address. Access to the normal or redundant column (RC) is similar that the R-Select signal is shared by both the redundant rows and columns in different time slots.

In the proposed DRAM architecture (Fig. 5), normal and redundant addresses share the same address bus routing network to subarray, and subarrays share the same single R-Select signal. This can substantially mitigate the potential area overhead because of extra address wires. However, for the DRAM configuration with wide data bus that multiple subarrays are activated and accessed concurrently, the address bus sharing solution tends to seriously degrade the repair efficiency, as all the involved subarrays have to be simultaneously repaired as long as one of them has a defective cell in the target address and the majority of redundancies are wasted. Nevertheless, as the redundancies provided in DRAM array are more than enough and the global sharing of redundant rows enabled by the proposed logic-DRAM co-design can significantly enhance the repair efficiency, the final repair rate is still satisfactory according to evaluation in Section IV.

Note that the proposed redundancy repair architecture that opens redundant rows and columns to off-chip access exerts no negative effect on the redundancy already used for laser-fuse-based repair. As when the R-Select is invalid, the address comparators can normally operate. Leveraging the proposed logic-DRAM co-designed repair also makes the DRAM array do not need electrical fuse on the DRAM die any more. Removing the electrical fuses and related voltage charge pump circuits can substantially reduce the cost of DRAM design.

C. Address Remapping in Memory Controlling Logic

As DRAM array opens both its normal array and redundant array to off-chip access, we can leverage address remapping on logic die to achieve memory repair instead of the fuse-based circuits on DRAM die. Moreover, as logic circuit design is much more flexible, we can design much more advanced and efficient architecture to effectively repair defective cells.

Fig. 6 illustrates the proposed address remapping architecture design used in memory controlling logic. We can see that the address remapping is mainly realized by leveraging two fully associative remapping buffers. Let us take *Row Remapping Buffer* as an example. The fully associative Row Remapping Buffer holds the defective row addresses in its tag portion and the corresponding remapped redundant addresses in its data portion. Considering that the Row Remapping Buffer may not be always full, we can add one valid bit to each tag to indicate whether or not it contains the defective row address, which can be leveraged to reduce access energy of the Row Remapping Buffer. The operation of this fully associative Row Remapping Buffer itself is very simple. During DRAM memory row activation, the target row address is compared with all the valid tags and the corresponding redundant row address is accessed in case of

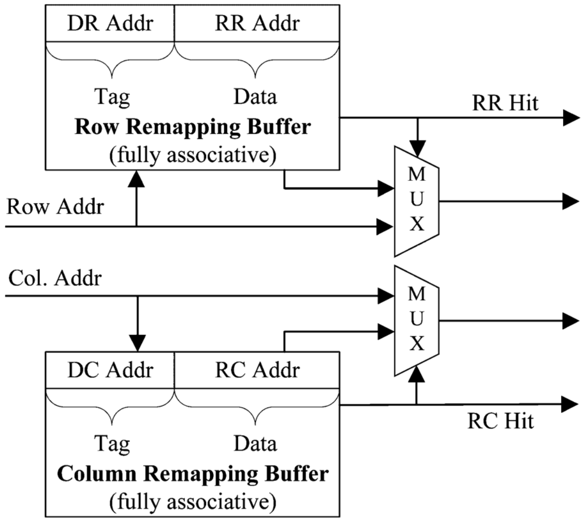

Fig. 6. Proposed address remapping architecture design in memory controlling logic.

a hit. Finally, both the remapped memory address and the redundant row hit signal are provided to memory controlling logic to decide the final target memory location.

The proposed address remapping buffer design is critical for enabling the global sharing of redundant row. As all the defective addresses and available redundancies on DRAM dies are collected and managed together, defective row repair is no longer limited to use redundant rows within the same subarray. Instead, the Row Remapping Buffer can determine to use all the available redundancies from any subarray on any die without any extra design effort. We should also point out that, it is inefficient to share the column redundancies, as column redundancy can only be accessed after a certain row activation. Sharing column redundancies across different DRAM rows may incur extra row activation and hence complicate the DRAM access. Therefore, global sharing is restricted only to redundant row repair in this study.

The tag and data portions of these remapping buffers, i.e., defective and the corresponding redundant addresses, can be supplied from embedded or external non-volatile memories such as antifuse one-time programmable (OTP) ROM or Flash memory. During the test procedure after die stacking, when DRAM dies have stacked on logic die, the automatic test equipment (ATE) tests the stacked DRAM, identifies all defective cells and allocates the unused spares by repair analysis for each 3D IC. The repair data, which include defective row/column addresses and corresponding redundant addresses, are stored in the non-volatile memory within the system and loaded into Row and Column Remapping Buffers in case of power on.

D. A Case Study on Interface Design

Given that the DRAM array has opened both its normal and redundant arrays, and memory request can correctly identify the defective address and remap it to redundancy, the last thing to do is to properly deliver failure and repair information from logic die to DRAM dies. The current standard DRAM interface I/O needs to be extended to deliver extra data, such as remapping buffer hit and redundant address. With 3D integration technology, the way to extend DRAM interface as required is easy and quite diversified. In this paper, we just present and discuss a case study by extending the JEDEC Wide I/O Single Data Rate standard [22].

TABLE I
COMMAND TRUTH TABLE FOR THE PROPOSED DRAM I/O INTERFACE

CMD	RAS#	CAS#	WE#	RES#	ADD
ACT.	L	H	H	H	Row Address
ACT. *w/ Repair*	L	H	H	L	RR Address
READ	H	L	H	H	Col. Address
READ *w/ Repair*	H	L	H	L	RC Address
WRITE	H	L	L	H	Col. Address
WRITE *w/ Repair*	H	L	L	L	RC Address

Table I shows the command truth table for the proposed DRAM I/O interface by extending the aforementioned Wide I/O standard. We only add the I/O of RES# into Wide I/O interface. RES# is a single bit I/O signal that indicates whether or not the target address is defective and is set as valid (L) by RR Hit or RC Hit from Row or Column Remapping Buffer. Accordingly, the additional RES# adds three new commands to the truth table (Table I), i.e., ACTIVE with Repair, READ with Repair, and WRITE with Repair, where all the commands share the same address bus I/Os.

The timing diagram is illustrated in Fig. 7 to demonstrate the use of these commands. During row activation, if the target row is failure free and Row Remapping Buffer misses, RES# is set as invalid (H), normal ACTIVATE command is sent to DRAM, and ADD delivers the address of normal row; however, if the target row contains failure cells and Row Remapping Buffer hits, RES# is set as valid (L), ACTIVATE with Repair command is sent to DRAM, and ADD delivers the corresponding address of redundant row. During column write/read, if the target column is failure free and Column Remapping Buffer misses, RES# is set as invalid (H), normal WRITE/READ command is sent to DRAM, and ADD delivers the address of the normal column. If the target column contains failure cells and Column Remapping Buffer hits, RES# is set as valid (L), WRITE/READ with Repair command is sent to DRAM, and ADD delivers the address of the corresponding redundant column.

We note that Table I is not a complete truth table for standard DRAM I/O interface; only three representative commands and their extension are illustrated in this table. However, other commands can also be easily extended in a similar way. More important, the extended I/O interface is fully compatible with the existing JEDEC Wide I/O standard, and the extended commands are only issued when RES# is set as valid (L). In addition, as we only add an extra I/O signal to the interface, the required modification to control logic on DRAM die is relatively very small. Moreover, the proposed interface design by adding a single RES# I/O signal is a relatively straightforward and cost-efficient solution. The additional I/O signal can be further eliminated by fully multiplexing the original command and address I/O signals. However, this procedure tends to complicate the interface timing and may even degrade the throughput. We can also add more I/O signals to deliver additional information, such as the location address of defective subarray among all the concurrently accessed subarrays. Thus, together with the DRAM array design, the repair capability can be improved further at the cost of increased overhead.

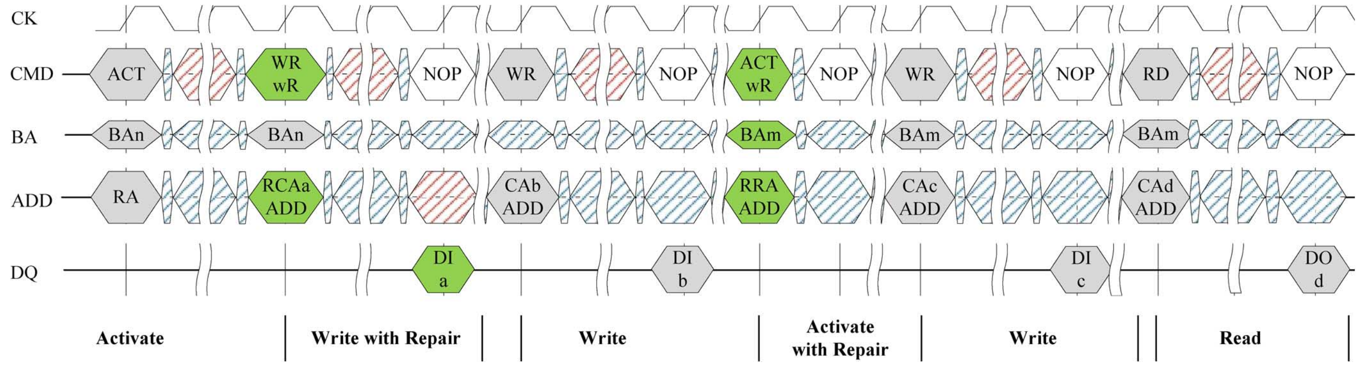

Fig. 7. Timing diagram of the proposed DRAM I/O interface.

IV. Evaluation Results

A. Yield Improvement

Let us use a 1 Gb Wide I/O Single Data Rate DRAM presented in [23] as an example to discuss the yield improvement and the following overhead and timing penalty. The DRAM die has 4 channels, each comprising 2 banks and each bank containing 32 blocks. Each block is further divided into 32 subarrays. Considering that each single channel has 128 DQs, we assume that 32 subarrays in one block is concurrently activated and that each subarray provides 4 data bits. Thus, each subarray contains 256 wordlines and 512 bitlines. We assume that each subarray is equipped with 4 redundant rows and 16 redundant columns. During redundant column repair, 4 consecutive bitlines are repaired together by 4 consecutive redundant columns in each subarray. The redundant rows can be utilized across different subarrays enabled by the global sharing, whereas the redundant columns can only be used to repair within each subarray. Moreover, due to address bus sharing, those concurrently accessed subarrays must be simultaneously repaired as long as one of their target addresses has a defective cell.

To evaluate the yield improvement of the proposed repair architecture, we build a simulator with the following characteristics. We use a negative binomial distribution defect model [24] and consider both single-bit defective cells and cluster defects. The percentages of single-bit defective cells, defective rows, and columns are set as 50%, 25%, and 25%, respectively. This simulator can support us to inject defects into random locations across the DRAM die and evaluate the repair rate. The *repair-most* analysis approach proposed in [25] is used to allocate the spare rows and columns for defective cells and rows/columns. Since pre-stacking repair may use some spares and post-stacking repair rate is partially determined by the unused spares, the simulator needs to inject and repair defects in each die twice. During the first round, we randomly inject defects into each die, repair them with spares, and record the occupied spares. During the second round, we randomly inject defects into the known good dies, repair them with unused spares, and analysis the repair rate. The location of defect injection covers the spares, and the situations that spares either used for pre-stacking repair or remaining unused become defective during die stacking are taken into consideration.

To simplify the yield improvement evaluation, we assume that DRAM yield is only determined by the repair rate and can be affected by the defect density incurred both prior to and during 3D integration. Accordingly, we let D_0 represent the average defect density (faults/die) incurred prior to the 3D integration and let D_1 represent the average defect density (faults/die) incurred during 3D integration. For each defect density scenario, we simulate 10 000 instances. Fig. 8 illustrates the simulated yield of 3D DRAM with and without repair after die stacking by varying D_0 from 0 to 300 faults/die and D_1 from 0 to 100 faults/die respectively. Fig. 8(a) shows that without the repair after die stacking, the compound yield of the 3D stacked DRAM drastically decreases from 100% to 0% with the increase of D_1, even when D_0 is very low. Whereas the compound yield

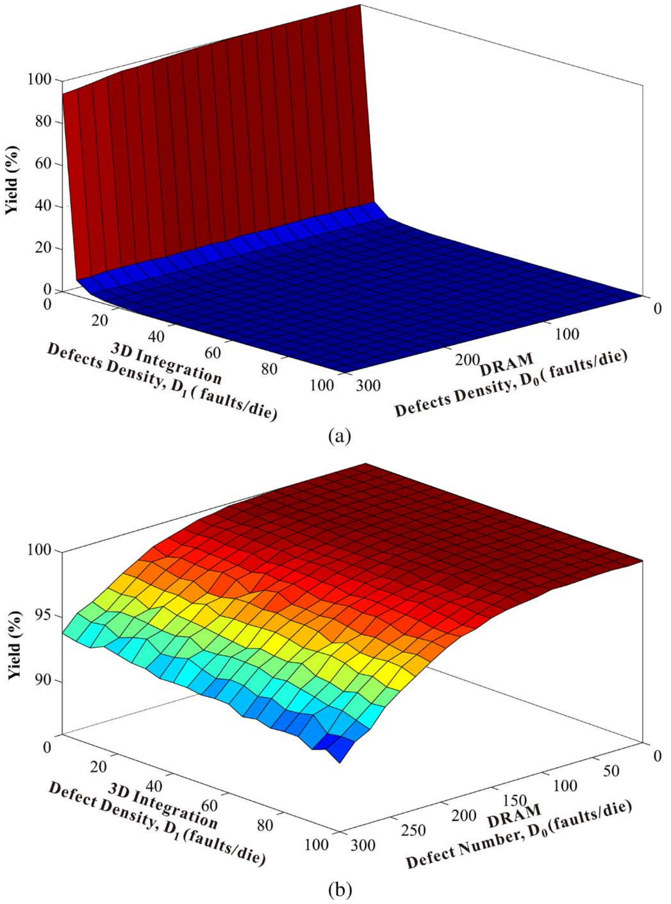

Fig. 8. Yields of the evaluated 1 Gb stacked DRAM with and without the proposed repair after die stacking: (a) without repair after die stacking, the yield decreases dramatically as D_1 increases, and (b) with the proposed repair after die stacking, the yield is stable and kept above 91.27%.

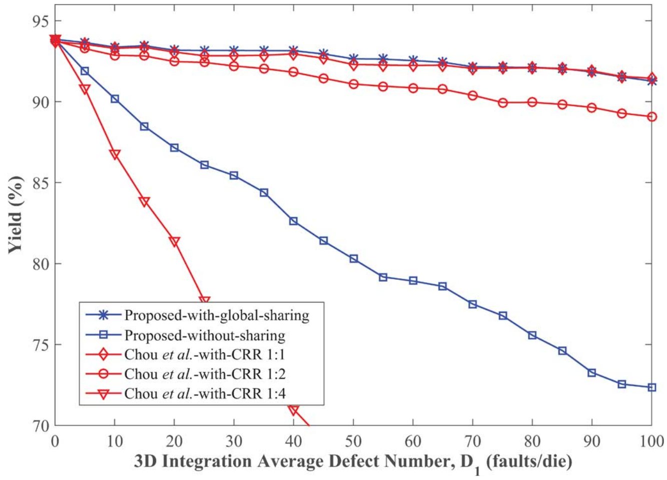

Fig. 9. Yield comparison for the evaluated 1 Gb stacked DRAM among the proposed repair techniques with and without global sharing of redundant rows, and Chou et al.'s technique [12] with different configuration, when the average defect density incurred prior to 3D integration D_0 is 300 faults/die.

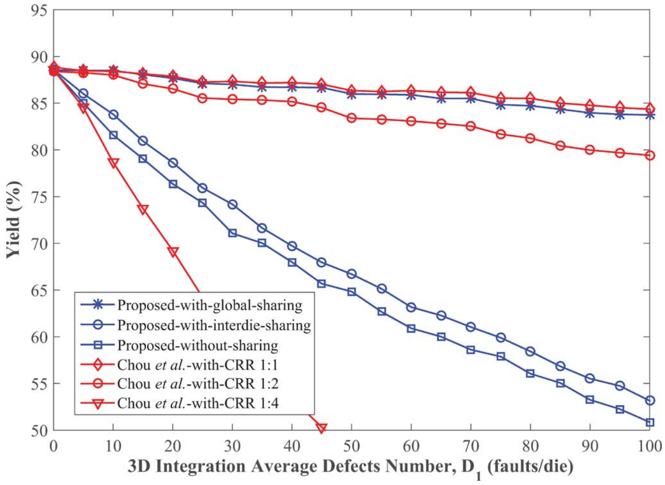

Fig. 10. Yield comparison for the evaluated 2-die stacked 2 Gb DRAM among the proposed repair without global sharing, with vertical inter-die redundancy sharing [8], [21] and global sharing of redundant rows, and Chou et al.'s technique [12] with different configurations, when the average defect density incurred prior to 3D integration D_0 is 300 faults/die.

keeps relatively much more stable for D_0, due to the memory repair before die stacking. Fig. 8(b) shows that, with the proposed repair after die stacking, the compound yield is significantly improved and can be kept above 91.27% even when D_1 is as high as 100 faults/die. This result further demonstrates that repair after die stacking is indispensable for the 3D logic-DRAM integrated system and that the proposed logic-DRAM co-designed repair approach can significantly alleviate potential yield loss.

The global sharing of redundant rows enabled by the proposed logic-DRAM co-designed repair is important to improve the yield of stacked DRAM. Fig. 9 shows the yield comparison between the proposed repair technique with and without global sharing of redundant rows, when the average defect density incurred prior to 3D integration D_0 is set as 300 faults/die. We can see that the global sharing of redundant rows can steadily increase the compound yield from 72.35% to 91.27%. Moreover, the global sharing of redundant rows tends to be more effective when more DRAM dies are stacked and the capacity of stacked DRAM increases. The main reason is that multi-die stacking tends to require a higher yield for each single die and provide more redundancies for sharing as well. Accordingly, we evaluate a 2-die stacked 2 Gb DRAM and compare the yield of the proposed repair technique with and without global sharing.

Moreover, we further compare the proposed global sharing with other vertical inter-die redundancy sharing approaches [8], [21]. Fig. 10 illustrates the evaluation results for the yield comparison when the average defect density incurred prior to 3D integration D_0 is 300 faults/die for each single die. We can see that without global sharing, the yield of the 2-die stacked DRAM is lower and decreases more sharply than that of the single-die stacked DRAM. However, the global sharing of redundant rows across the two dies can improve the yield from 50.86% to 83.73%. Although vertical inter-die redundancy sharing can also improve the yield to some extent, its efficiency is much lower compared with that of global sharing.

The repair capability of the proposed repair is also compared with Chou *et al.*'s technique [12] in Fig. 9 and Fig. 10. As Chou *et al.*'s technique repairs stacked DRAM by adding extra controlling circuit to redundancy, its repair capability is proportional to the overhead incurred by the controlling circuit. To maximize its repair capability, each redundancy should be configured with its dedicated control circuit and be used independently. However, this tends to incur significant overhead, as discussed in the following subsection, and is thus infeasible. Reference [12] further mentioned that the design overhead can be significantly reduced if several redundancies share the same control circuit and are simultaneously repaired. Therefore, the repair capability of Chou *et al.*'s technique depends on the Control circuit/Redundancy Ratio (CRR). In the evaluation, we simulate Chou *et al.*'s technique with three CRR configurations, i.e., 1:1, 1:2, and 1:4. Figs. 9 and 10 show that the repair capability of the proposed technique is comparable with the best result of Chou *et al.*'s technique. Moreover, the yield of Chou *et al.*'s technique dramatically decreases when control circuit sharing is employed to redundancies to reduce the overhead.

TABLE II
YIELD OF DIFFERENT CONFIGURATIONS ON CONCURRENTLY ACCESSED SUBARRAYS FOR THE EVALUATED 1 Gb STACKED DRAM

Block Size	Subarrays per Block[1]	DQs per Channel	Yield
2 Mb	16	64 bits	92.98%
4 Mb	32	128 bits	91.27%
8 Mb	64	256 bits	89.55%

[1] All the subarrays in one block are concurrently accessed.

As we discussed in Section III-B, due to the address bus sharing solution, all the concurrently accessed subarrays have to be simultaneously repaired as long as one of them has a defective cell in the target address. This tends to seriously waste redundancies and degrade the repair efficiency in case of wide data bus. Accordingly, we simulate the yield of different configurations on concurrently accessed subarrays for the evaluated 1 Gb stacked DRAM and illustrate the results in Table II. We can tell that, due to the global sharing of redundant rows, the yield remains relatively steady as the number of concurrently accessed subarrays increases. Of course, the repair capability will be reduced when the data bus becomes wider and more subarrays are concurrently accessed. However, it is not efficient for DRAM design to concurrently activate and access too many subarrays, as this will significantly reduce the signal/noise ratio

TABLE III
CACTI REPORT FOR FULLY ASSOCIATIVE ROW AND COLUMN REMAPPING BUFFER WITH DIFFERENT ENTRIES AT 50 nm TECHNOLOGY NODE

Entries	Row Remapping Buffer			Column Remapping Buffer		
	Area (μm^2)	Dyn.E (pJ)	Access time (ns)	Area (μm^2)	Dyn.E (pJ)	Access time (ns)
128	2054.8	1.14	0.34	1806.5	0.59	0.27
256	4051.6	1.79	0.39	3045.8	0.88	0.29
512	6616.7	2.41	0.43	5019.9	1.46	0.32

and increase charging/discharging current. Therefore, the wider data bus should be achieved by exploiting the multi-channel design as the referred wide-I/O DRAM work [23] does.

B. Overhead Analysis

The additional area overhead of the proposed repair design is mainly due to 1) the fully associative address remapping buffers on logic die that hold the defective addresses and the corresponding redundant addresses; 2) the redundant address decoders, MUXs, and OR gates that are added into each subarray on DRAM die; 3) the routing wire of R-Select delivered to each subarray on DRAM die; and 4) the additional interface and related control circuit design.

For the evaluated 1 Gb wide I/O DRAM, the defective row address in each remapping buffer entry should be 16 bits, i.e., 3 bits for 8 banks, 5 bits for 32 blocks, and 8 bits for 256 intra-subarray rows. Meanwhile, the redundant row address should be 10 bits because each subarray only has 4 redundant rows. Similarly, the defective column address should be 15 bits, i.e., 3 bits for 8 banks, 5 bits for 32 blocks, and 7 bits for 128 intra-subarray columns, and the redundant column address should be 2 bits for 4 intra-subarray redundant columns. CACTI 6.0 [26], a widely used cache modeling tool, is used to estimate the performance and overhead of the proposed address remapping buffers at 50 nm technology node. During the simulation for yield improvement, we found that for the defect density D_1 of 100 faults/die, the maximum numbers of used redundant rows and columns are 243 and 177 respectively. Hence, we simulate the fully associative buffers with 128, 256, and 512 entries. Table III lists the corresponding modeling results produced by CACTI, including area, dynamic energy, and access time. Clearly, the area cost of address remapping buffers is quite low; in particular, it only takes 7097.4 μm^2 for a 256-entry Row Remapping Buffer and a 256-entry Column Remapping Buffer. Non-volatile memory may also be needed to store repair data for address remapping buffers. However, this extra storage overhead can be neglected because the computing system or system-on-chip (SoC) always integrates large-capacity non-volatile memory in itself for program and configuration data.

To estimate the overhead of the redundant address decoders and logic gates added to each DRAM subarray, we implement two 2–4 decoders and associated logic gates in 65 nm technology. Their overall area is only 5.76 μm^2, whereas the area of DRAM subarray (256 × 512) is approximately 3666 μm^2 in our design practice. In fact, commercial DRAM has already integrated similar redundant address decoders and logic gates for internal testing. Therefore, the logic circuit can be efficiently reused for the proposed repair, and the extra area overhead is negligible. The area overhead incurred by the routing wire of R-Select is roughly estimated in accordance with the result in [2]. The H-tree network occupies approximately 26% of the total area in a 1 Gb DRAM and only add 1 bit is added to the 159-bit bus (data and address). Hence, the area overhead is below 0.16% of the overall DRAM die.

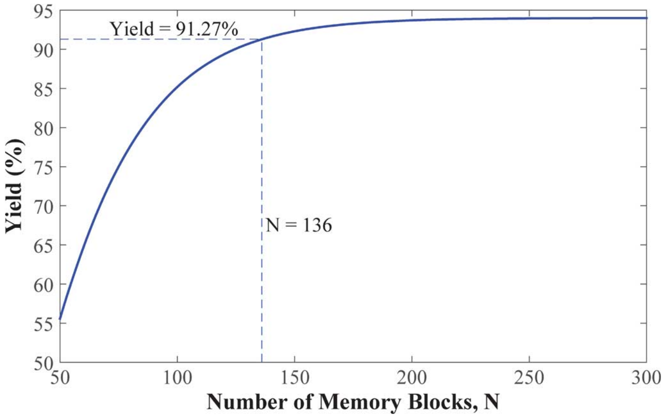

Fig. 11. Yield vs. number of required memory blocks.

The area overhead for interface design includes the extra TSV for RES# and the modification to controlling circuit. We assume the use of the cuboid copper TSV with a 5 μm × 5 μm square cross section, a 50 μm length, a 200 nm silicon dioxide linear thickness, and a 50 nm diffusion barrier. The landing pad has a square area of 7.8 μm × 7.8 μm, and a keep out zone of 1.04 μm [27]. The area overhead (approximately 78.15 μm^2) is very small because only one extra TSV is needed for RES#. The induced area overhead can also be neglected because the modification to controlling circuit is only due to RES#.

In summary, the area overhead of the proposed repair technique is mainly incurred by address remapping buffer on logic die and is relatively very low (7097.4 μm^2) compared with that of typical processor or SoC design. Compared with [12], this paper provides a much more efficient technique to reactivate the unused spare for 3D integration repair. The technique in [12] has to take approximately 1855 μm^2 for each redundancy repair after 3D integration; thus, it becomes too expensive when we need to repair with hundreds of redundancies. In addition, the TSV and related controlling circuit must be located scatteredly on the logic die right beside each redundancy, which will further impose severe limitation on the floorplan of logic die. Considering the aforementioned yield comparison in Fig. 9 and Fig. 10, we can conclude that the repair technique proposed in the present study is much more cost-efficient than that proposed in [12].

We should also point out that, the idea to reactivate unused redundancies saves a considerable amount of area overhead. Prior works [6], [16] that proposed to provide redundancy with extra SRAM memory on logic die is very inefficient for DRAM repair, especially when the DRAM capacity is large and the defect density is high. Although single-bit defective cells can be repaired at a very low cost, defective rows and columns can only be repaired by the memory block of hundreds of memory cells. Fig. 11 illustrates the required number of memory blocks for cluster defect repair at different yield rates. A total of 136 memory blocks are needed for the yield of 91.27%, which means at least 8.7 KB SRAM memory for a 1 Gb single die

DRAM. The area overhead is too expensive for logic design because the required memory blocks are linearly increased with the capacity of DRAM.

The power consumption overhead should also be considered and carefully evaluated. For the proposed memory repair design, the memory controller on the logic die must access the fully associative address remapping buffers for each DRAM access, which tends to increase the dynamic power consumption. According to [23], when the wide I/O DRAM runs at its full speed of 200 Mb/s data rate and 12.8 GB/s bandwidth, the dynamic power consumption of both DRAM array and I/O is approximately 484.2 mW. According to the CACTI report in Table III, the dynamic energy for one access to a 256-entry Column Remapping Buffer is 0.88 pJ. For the data access rate of 200 MHz, the dynamic power consumption of the Column Remapping Buffer is below 0.176 mW. Hence, the power consumption overhead of the proposed address remapping buffer design is only as low as 0.036%.

C. Performance Analysis

As shown in Fig. 7, the proposed interface I/O protocol is fully compatible with the existing wide I/O standard and does not add any delay cycle even when accessing a defective address. The modification to the control logic and memory array on DRAM die is also quite small and will not incur noticeable delay to the overall access time. Therefore, the performance evaluation in this study mainly focuses on the timing penalty induced by the address remapping buffers on the logic die.

According to the CACTI report in Table III, access times for the 256-entry Row Remapping Buffer and 256-entry Column Remapping Buffer are 0.39 ns and 0.29 ns, respectively. Meanwhile, the highest working frequency of memory controlling circuit on logic die should be 200 MHz to match the maximum I/O data rate. Hence the minimum cycle time should be 5 ns. Access time to the 256-entry address remapping buffer only occupies less than 7.8% of the total cycle time of the memory controlling circuit and can be easily hidden inside each cycle. Moreover, the time budget for accessing the address remapping buffer is actually sufficient because the access addresses are usually first stored into the scheduling queue and then transferred after several clock cycles. Therefore, the proposed Row and Column Remapping Buffers will not incur any noticeable timing penalty to the memory controlling circuit design.

V. Conclusion

This paper aims to leverage logic-DRAM co-design to efficiently repair 3D integration-induced defective cells in stacked DRAM with unused spares. By making the DRAM array open its redundancy to off-chip access and designing the defective address comparison and redundant address remapping in the memory controller on the logic die, we can directly access the corresponding redundancy as a common DRAM access through generic interface I/Os if necessary. Hence, the unused spares after wafer level repair can be efficiently reactivated to repair the defective DRAM cells incurred by 3D integration. Evaluation results show that the proposed repair technique for stacked DRAM can significantly alleviate the 3D integration-induced potential yield, while only incurring very low area and power consumption overhead and negligible timing penalty.

Acknowledgment

The authors are grateful to the anonymous reviewers for their valuable comments.

References

[1] G. H. Loh, “3D-stacked memory architectures for multi-core processors,” in *Proc. ACM SIGARCH Comput. Archit. News*, 2008, pp. 453–464.

[2] H. Sun *et al.*, “3D DRAM design and application to 3D multicore systems,” *IEEE Design Test Comput.*, vol. 26, pp. 36–47, 2009.

[3] M. Horiguchi and K. Itoh, “Redundancy,” in *Nanoscale Memory Repair*. New York: Springer, 2011, pp. 19–67.

[4] J. K. Wee *et al.*, “An antifuse EPROM circuitry scheme for field-programmable repair in DRAM,” *IEEE J. Solid-State Circuits*, vol. 35, pp. 1408–1414, 2000.

[5] A. W. Topol *et al.*, “Three-dimensional integrated circuits,” *IBM J. Res. Develop.*, vol. 50, pp. 491–506, 2006.

[6] N. Axelos *et al.*, “Efficient memory repair using cache-based redundancy,” *IEEE Trans. Very Large Scale Integr. (VLSI) Systems*, vol. 20, pp. 2278–2288, 2012.

[7] C.-W. Chou *et al.*, “Yield-enhancement techniques for 3D random access memories,” in *Proc. IEEE Int. Symp. VLSI Design Autom. Test (VLSI-DAT)*, 2010, pp. 104–107.

[8] L. Jiang *et al.*, “Yield enhancement for 3D-stacked memory by redundancy sharing across dies,” in *Proc. Int. Conf. Comput.-Aided Design*, 2010, pp. 230–234.

[9] Y.-F. Chou *et al.*, “Yield enhancement by bad-die recycling and stacking with though-silicon vias,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, pp. 1346–1356, 2011.

[10] C. L. Su *et al.*, “An integrated ECC and redundancy repair scheme for memory reliability enhancement,” in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2005, pp. 81–89.

[11] R. Datta and N. A. Touba, “Exploiting unused spare columns to improve memory ECC,” in *Proc. IEEE VLSI Test Symp.*, 2009, pp. 47–52.

[12] Y. F. Chou *et al.*, “Reactivation of spares for off-chip memory repair after die stacking in a 3-D IC with TSVs,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, pp. 2343–2351, 2013.

[13] K. Itoh, “High performance subsystem memories,” in *VLSI Memory Chip Design*. New York: Springer, 2001, pp. 354–357.

[14] W. Yang, “Antifuse circuitry for post-package DRAM repair,” U.S. Patent 6 240 033, 2001.

[15] J.-K. Wee *et al.*, “A post-package bit-repair scheme using static latches with bipolar-voltage programmable antifuse circuit for high-density DRAMs,” *IEEE J. Solid-State Circuits*, vol. 37, pp. 251–254, 2002.

[16] X. Wang *et al.*, “Global built-in self-repair for 3D memories with redundancy sharing and parallel testing,” in *Proc. IEEE Int. 3D Syst. Integr. Conf. (3DIC)*, 2011, pp. 1–8.

[17] M. Taouil *et al.*, “Yield improvement for 3D wafer-to-wafer stacked memories,” *J. Electron. Testing*, vol. 28, pp. 523–534, 2012.

[18] C.-C. Chi *et al.*, “3D-IC BISR for stacked memories using cross-die spares,” in *Proc. Int. Symp. VLSI Design, Autom., Test (VLSI-DAT)*, 2012, pp. 1–4.

[19] M. Horiguchi *et al.*, “A flexible redundancy technique for high-density DRAMs,” *IEEE J. Solid-State Circuits*, vol. 26, pp. 12–17, 1991.

[20] M. Horiguchi, “Redundancy techniques for high-density DRAMs,” in *Proc. Annu. IEEE Int. Conf. Innov. Syst. Silicon*, 1997, pp. 22–29.

[21] M. Lefter *et al.*, “Is TSV-based 3D integration suitable for inter-die memory repair?,” in *Proc. ACM Conf. Design, Automat., Test Eur. (DATE)*, 2013, pp. 1251–1254.

[22] “Wide I/O single data rate,” JEDEC Standard, JESD229, 2011 [Online]. Available: https://www.jedec.org/standards-documents/docs/jesd229

[23] J. S. Kim *et al.*, “A 1.2 V 12.8 GB/s 2 Gb mobile wide-I/O DRAM with 4 × 128 I/Os using TSV based stacking,” *IEEE J. Solid-State Circuits*, vol. 47, pp. 107–116, 2012.

[24] C. H. Stapper, “Yield model for fault clusters within integrated circuits,” *IBM J. Res. Develop.*, vol. 28, pp. 636–640, 1984.

[25] M. Tarr *et al.*, “Defect analysis system speeds test and repair of redundant memories,” *Electronics*, vol. 57, pp. 175–179, 1984.

[26] N. Muralimanohar *et al.*, “CACTI 6.0: A tool to model large caches,” *HP Lab.*, 2009.

[27] M. Pathak *et al.*, “Through-silicon-via management during 3D physical design: When to add and how many?,” in *Proc. Int. Conf. Comput.-Aided Design*, 2010, pp. 387–394.

Minjie Lv received the B.S. degree from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China in 2010. Currently, he is a Ph.D. candidate in the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His current research interests include high-performance computer architecture and high-reliability, low-power memory systems.

Hongbin Sun (M'11) received B.S. and Ph.D. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003 and 2009, respectively. He was a visiting Ph.D. student in electrical, computer and systems engineering department at Rensselaer Polytechnic Institute, Troy, NY, USA, from 2007 to 2008. Currently he is an Associate Professor at Xi'an Jiaotong University. His current research interests include memory hierarchy design and VLSI architecture for digital video processing.

Qiwei Ren received the B.S and M.S. degrees from Tsinghua University, Beijing, China, in 1992 and 1995, respectively, and received Ph.D. degree from Delft University of Technology, The Netherlandsin 2001.

In 2006, he joined Qimonda Technologies (Xi'an) as the director of DRAM product design. Currently, he is the General Manger of Xi'an Sinochip Semiconductors Co. Ltd. and a part-time professor at Xi'an Jiaotong University. He has rich experiences in memory design and testing, process technology, and yield analysis.

Bing Yu received the M.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2005. In 2007, he joined Qimonda Technologies (Xi'an) as a design engineer for array column and row. In 2009, he joined Xi'an Sinochip Semiconductors Co. Ltd. and was involved in DRAM product design, especially for array, redundancy, and column/row. Currently, he is a department manager of product design at Xian Sinochip Semiconductors Co. Ltd. and in charge of DRAM product design and yield improvement.

Jingmin Xin (S'92–M'96–SM'06) received the B.E. degree in information and control engineering from Xi'an Jiaotong University, Xi'an, China, in 1988 and the M.S. and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1993 and 1996, respectively. Since 2007, he has been a Professor at Xi'an Jiaotong University. His research interests are in the areas of adaptive filtering, statistical and array signal processing, system identification, and pattern recognition.

Nanning Zheng (SM'93–F'06) graduated from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975, and received the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981 and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985. He jointed Xi'an Jiaotong University in 1975, and is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems.