

High-efficiency BIRA for embedded memories with a high repair rate and low area overhead

Joohwan Lee, Kihyun Park, and Sungho Kang

Abstract—High-efficiency built-in redundancy analysis (BIRA) is presented. The proposed BIRA uses three techniques to achieve a high repair rate using spare mapping registers with adjustable fault tags to reduce area overhead. Simulation results show that the proposed BIRA is a reasonable solution for embedded memories.

Index Terms—Built-in redundancy analysis, repair rate, area overhead, embedded memory

I. INTRODUCTION

Embedded memories will continue to dominate the increasing system-on-a-chip (SoC) area in the future. The yield of an SoC is dominated by the yield of the memory cores in the chip [1]. A built-in self-repair (BISR) technique can enhance the yield of embedded memories.

The most commonly used repair scheme is two-dimensional (2D) redundancy, which is composed of spare rows and spare columns. The optimal redundancy analysis (RA) problem for memories with 2D redundancy is known as NP-complete. Some heuristic built-in redundancy analysis (BIRA) approaches have been proposed as part of BISR, including CRESTA [2], LRM [3], ESP [3], and BRANCH [4]. CRESTA uses several parallel sub-analyzers to evaluate all possible repair solutions and produces the optimal repair rate. However, its area overhead is too large to be applied to memories with many spares. The LRM and ESP methods

require relatively low storage for failure information, but they cannot guarantee a high repair rate compared with those of other RA algorithms. BRANCH achieves the optimal repair rate with an exhaustive search algorithm. In order to reduce the analysis time, all detected faults are stored in content addressable memories (CAMs). However, the fault-storing CAM structure results in high area overhead.

A novel BIRA for embedded memories is proposed with a high repair rate and low area overhead. The proposed BIRA performs the RA process using spare mapping registers with fault tags. To reduce the area overhead, a detected fault is directly stored in a spare mapping register without using a bitmap, and the number of fault tags is adjustable. The repair rate of the proposed BIRA using three techniques is much higher than that of LRM, and the area overhead of the proposed BIRA is similar to that of ESP. Therefore, the proposed BIRA is a very efficient solution for repairing embedded memories.

In Section II, the proposed BIRA is illustrated with a detailed example. Simulation results of repair rates and area overhead are shown in Section III. Finally, Section IV concludes the letter.

II. PROPOSED BIRA ALGORITHM

The use of a BIRA results in additional cost because of area overhead. Reduced area overhead clearly decreases the overall cost of chip production. Since the proposed BIRA uses simple spare mapping registers with adjustable fault tags, the area overhead can be considerably reduced compared with those of previous RA algorithms. Fig. 1 shows the structure of the spare mapping registers for the proposed BIRA with R_s spare

Manuscript received Sep. 21, 2011; revised Feb. 29, 2012.
Graduate School of Electrical and Electronic Engineering, Yonsei
University, Seoul, Rep. of Korea.
E-mail : eldsich@yonsei.ac.kr

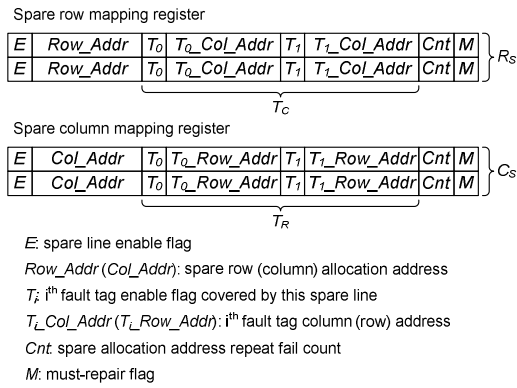


Fig. 1. Structure of the spare mapping registers.

rows and C_S spare columns. The number of spare row (column) mapping registers is the same as the value of R_S (C_S). If a detected fault, which does not share a row or column address with other existing faults, is repaired by a spare row (column), the addresses of the fault are stored in Row_Addr (Col_Addr) and $T_0_Col_Addr$ ($T_0_Row_Addr$). If a detected fault shares a row (column) address, the column (row) address of the fault is inserted into $T_1_Col_Addr$ ($T_1_Row_Addr$). Since the address of the fault is recorded at the fault tag address without duplication, the count (Cnt), which counts the repeated fail address, is needed to determine the value of the must-repair flag (M). Since the must-repair is considered, the maximal numbers of fault tags, T_C and T_R , are represented by C_S and R_S , respectively. Enable flags (E and T_i) are set to 1 when the pertinent address is used.

When a spare is used for repairing faults during the RA process, the decision is not changed in the previous BIRA algorithms. Unlike the previous BIRAs, the decision in the proposed BIRA can be changed by using three techniques for achieving a high repair rate, as follows.

- Conversion: When a single cell fault is repaired by a spare row (column), if a newly detected fault shares a column (row) address with the single cell fault and there is an available spare column (row), both faults are repaired by a spare column (row) instead of a spare row (column).
- Swap: When a fault is detected, if it is not repaired by pre-allocated spare lines and there is no available spare line, it can be repaired by swapping a spare row and a spare column.
- Inclusion: If all faults repaired by a spare line are covered by other spare lines, the spare line is no longer required for fault repair.

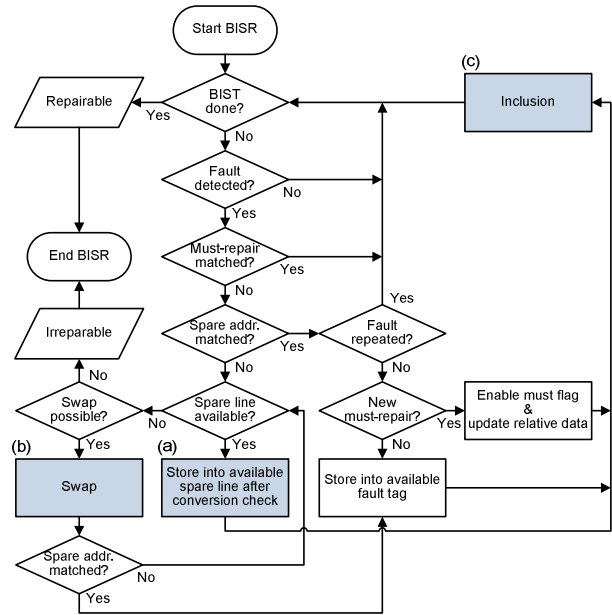


Fig. 2. BISR flow with the proposed BIRA.

The BISR flow with the proposed BIRA is shown in Fig. 2. If a fault which is not repaired by a must-repair spare line is detected by the built-in self test, the incoming fault is compared with previously stored spare allocation addresses. If there is no matched spare allocation address, the incoming fault is stored in more available spare line after the conversion check, as shown in Fig. 2(a). However, if no spare line is available, the fault can be stored in a fault tag using the swap technique when the swapped spare lines share a spare row or a spare column with the incoming fault, as shown in Fig. 2(b). If a spare allocation address is matched with the row address or the column address of the incoming fault, the fault is handled according to must-repair checks. After the incoming fault is recorded, the inclusion technique is applied for efficient use of redundancies, as shown in Fig. 2(c).

Fig. 3 shows an example of the proposed BIRA using an 8 by 8 memory block with 2 spare rows and 3 spare columns. In Fig. 3(a), when the fourth fault is detected, conversion occurs because the row address of the fourth fault is the same as that of the first fault. After the fifth and sixth faults are repaired by two spare columns, all the faults (#1 and #4) repaired by the second spare row are covered by the two spare columns; namely, inclusion occurs, as shown in Fig. 3(b). Since the second spare row is no longer used, it is available for next fault. When the



Fig. 3. An example of the proposed BIRA ($T_C=T_R=2$).

tenth fault is detected, there is no available spare. However, the tenth fault can be repaired by using swap technique. The tenth fault is repaired by swapping the first spare row and the second spare column, as shown in Fig. 3(c). Finally, after all the faults are processed, the repair solution is immediately determined because *Row_Addr* (*Col_Addr*) of each spare row (column) mapping register is the address of the repair solution.

III. SIMULATION RESULTS

Fig. 4 and 5 present the simulation results of the proposed BIRA compared with those of previous BIRAs. The size of the targeted memory blocks is 1,024 by 1,024, and the number of each fault tag is 2. To compare the normalized repair rates, each simulation was repeated 10,000 times for the number of randomly injected faults, which consisted of 37.07% faulty rows, 37.03% faulty columns, and 25.90% single cell faults. When a fault is generated, the chances of conversion, swap, and inclusion occurring in the fault distribution are 6.77%, 2.39%, and 0.32%, respectively. The normalized repair rates of CRESTA and BRANCH are 100% because their repair rates are optimal. Although the repair rate of the proposed BIRA is not optimal, it is much higher than those of ESP and LRM, as shown in Fig. 4. For other

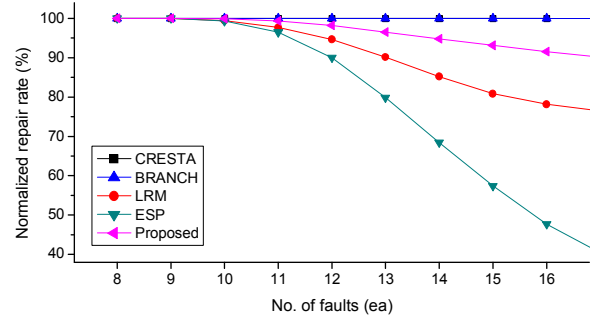


Fig. 4. Comparison of normalized repair rates ($R_S=C_S=4$, $T_C=T_R=2$).

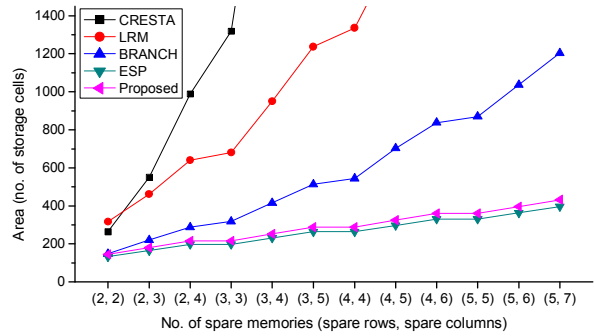


Fig. 5. Comparison of area overhead ($T_C=T_R=2$).

various cases, the repair rate of the proposed BIRA is always much higher than that of ESP and it is mostly much higher than that of LRM. We compared the area overhead with the number of spares in Fig. 5. The area overhead is estimated by using the same way as in [3, 4]. The area of storage requirement is a little bit less than that of the whole BIRA because of logic blocks; however, the area of storage cells dominates that of the BIRA. The area overhead of the proposed BIRA is much smaller than those of CRESTA, LRM, and BRANCH and similar to that of ESP, as shown in Fig. 5.

IV. CONCLUSIONS

The proposed BIRA with a high repair rate and low area overhead was presented. The three techniques of conversion, swap, and inclusion are used to achieve a high repair rate, and spare mapping registers with adjustable fault tags are used to minimize area overhead. In RA algorithms, there are trade-offs between repair rates and area overhead. The proposed BIRA has advantages in both of these areas. Therefore, the proposed BIRA is very suitable for embedded memories.

ACKNOWLEDGMENTS

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2010-0024707).

REFERENCES

- [1] Han, D., et al, "Novel hierarchical test architecture for SoC test methodology using IEEE test standards," *J. of Semi. Tech. and Sci. (JSTS)*, Vol. 12, No. 3, pp. 293-296, Sep., 2012.
- [2] Kawagoe, T., et al, "A built-in self repair analyzer (CRESTA) for embedded DRAMs," *Proc. Int. Test Conf., ITC*, pp. 567-574, Oct., 2000.
- [3] Huang, C.-T., et al, "Built-in redundancy analysis for memory yield improvement," *IEEE Trans. Relia.*, Vol. 52, No. 4, pp. 386-399, Dec., 2003.
- [4] Jeong, W., 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, pp. 2014-2026, Dec., 2010.



Joohwan Lee received the BS, MS, and PhD degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea in 2003, 2005, and 2012, respectively. He was also a Research Engineer with the ASIC Research Center, Yonsei

University. He is currently working in the S. LSI Division of Samsung Electronics as a senior engineer. His main research interests include very large scale integration (VLSI) design, fault simulation, fault diagnosis, built-in self-repair (BISR), built-in self-test (BIST), built-in redundancy analysis (BIRA), RA algorithm, reliability, 3D memory yield enhancement, and through-silicon via (TSV) repair.



Kihyun Park received the BS degree in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea in 2007. He was also a Research Engineer with the ASIC Research Center, Yonsei University.

He is currently working toward a combined MS/PhD degree in Electrical and Electronic Engineering at Yonsei University, Seoul, Korea. His current research interests include built-in self-repair (BISR), built-in self-test (BIST), built-in redundancy analysis (BIRA), test algorithms, and very large scale integration (VLSI) design.



Sungho Kang received the BS degree from Seoul National University, Seoul, Korea, and the MS and PhD degrees in Electrical and Computer Engineering from the University of Texas at Austin in 1992. He was a Research Scientist with the Schlumberger Laboratory for Computer Science, Schlumberger Inc., and a Senior Staff Engineer with the Semiconductor Systems Design Technology, Motorola Inc. Since 1994, he

has been a Professor with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea. His main research interests include VLSI/SOC design and testing, design for testability, BIST, defect diagnosis, and design for manufacturability.