Discussion of Cost-effective Redundancy Architectures

Keewon Cho, Jooyoung Kim, Hayoung Lee and Sungho Kang
Dept. of Electrical and Electronic Engineering
Yonsei University
Seoul, Korea
{ckw1505, kimjy9850, yseehy214}@soe.yonsei.ac.kr, and shkang@yonsei.ac.kr

Abstract—To get a reasonable yield, memories incorporate redundancies to substitute for faulty cells. As the performance of repair algorithm reaches some saturation point, recent studies focus on various redundancy architectures for higher repair rate. In this paper, three kinds of spares, i.e., local, common, and global spares, are discussed to analyze the efficiency of redundancy architectures in respect of the repair cost. In order to estimate the impact of each spare, more than a hundred redundancy architectures are simulated with different faulty patterns. This paper performs a data analysis and suggests cost-effective redundancy architectures.

Keywords— built-in redundancy analysis (BIRA); repair cost; yield; redundancy architecture

I. INTRODUCTION

Memory yield and reliability problems become more sensitive because an advanced technology induces more defects. One powerful solution is a built-in redundancy analysis (BIRA) methodology which repairs faulty cells with prepared redundancies. However BIRA algorithms targeting a single block with local spares have some limitations in repair efficiency. Even though all repairable memories are fixed with the algorithms, the percentage of unrepairable memories is increased by multiple faults induced circumstances. Therefore, recent studies turn their attentions to change redundancy architectures [1, 2]. [1] introduces three different types of the redundancy constraints to achieve higher repair rate. [2] shows an optimal repair algorithm which utilizes a block-based redundancy architecture.

Because repair rate is directly connected with the manufacturing yield, a main purpose of previous studies is increasing repair rate. However, the memory repair cost is decided by various factors. Especially, area overhead is one of the major factor of the BIRA cost since the area of chip is limited. Furthermore, area overhead become more important in multi-memory blocks considering redundancy architectures.

This paper conducts a data analysis based on numerous experimental results from a simulation tool in c-language. In addition, cost-effective redundancy architectures targeting four memory blocks in parallel are suggested considering both repair rate and area overhead.

II. PROPOSED ARCHITECTURES

This paper sets up redundancy architectures with 4-memory blocks utilizing three different kinds of spares. Local spares, which are commonly used in previous studies, can only be allocated in their located memory block. On the other hand, common and global spares can be allocated in multiple memory blocks. Common spares can be shared in their two adjacent memory blocks and global spares can be placed in multiple memory blocks. Figure 1 shows the basic redundancy architecture which utilizes all kinds of spares. Each memory block contains a local row and a local column spare. There are four common spares in Figure 1 and each of them can be used in its adjacent memory blocks. Global spare lines can repair two memory blocks simultaneously in this case. And also, these lines can be allocated wherever faults occurred.

Actual area overhead of various redundancy architectures can hardly be estimated without implementing all structures. Instead, previous study offers the storage requirement comparing method [3]. The area of storage requirement takes a most portion of the area of the whole BIRA and can easily be calculated with mathematical equations. To estimate area overhead of various redundancy architectures, storage requirements for each type of spare are evaluated. Since these values are varied depending on BIRA algorithms, this paper uses BRANCH algorithm [3] to observe the relationships among three kinds of spares.

This research was supported by the MOTIE (Ministry of Trade, Industry & Energy) (10052875) and KSRC (Korea Semiconductor Research Consortium) support program for the development of the future semiconductor device.
III. DATA ANALYSIS AND RESULTS

In order to acquire the data, we developed a simulation tool in C-language. When the number of each spare and fault counts is entered, the simulation tool generates faulty addresses at random. When the analysis is end, the tool shows each trial’s repair rate and area overhead. Changing the number of faults and each spare, repair rate and area overhead are observed to accumulate the data. Since hundreds of redundancy architectures are simulated with different faulty patterns, it is hard to show all the performance of each type of spare.

\[ \text{Repair rate ROI} = \frac{\sum_{i=1}^{m}(R_{1} - R_{Ref}) + \sum_{i=1}^{n}(R_{2} - R_{Ref}) + \sum_{i=1}^{o}(R_{3} - R_{Ref})}{m+n+o} \]

where \( R_{Ref} \) is repair rate of the reference set and \( R_I \) is repair rate of the set which has one more additional spare than the reference set and \( m \) is the number of \( R_{1} \) sets, \( R_{2}, R_{3}, n \) and \( o \) are decided as similar as \( R_{1} \) and \( m \).

\[ \text{Area overhead ROI} = \# \text{ of storage } (R_k - R_{Ref}). \]

This evaluation index helps to figure out the hardware impact of spare lines.

Figure 2 shows the performance of each spare’s repair rate ROI. Basically, injecting common spare lines shows great improvement in repair rate. This result is quite reasonable because common spare lines are more flexible than other spares. The fact that common spare lines can be shared in two memory blocks leads to a maximum 70% improvement in rate of increase in repair rate. On the other hand, global spare lines shows poor performance in repair rate ROI. Since a global spare line is twice the size of a common (or local) spare line, we gives a penalty to the number of global spare lines. This penalty affects whole results in Figure 2.

Figure 3 shows the performance of each spare’s area overhead ROI. The remarkable result in the graph is that the area increasing rate of common spare line is overwhelming. Also, we can observe that area overhead ROI of local spare lines is placed in the lowest position at first. However, as the number of adding spares is increased, its value moves up rapidly. It can be expected that the efficiency of global spare lines gets better in complicated faulty memories.

Because the faulty patterns can be varied in various memories, there is no perfect redundancy architecture. But cost-effective redundancy architectures can be recommended from the data analysis. For low area overhead, local and common spare lines should be injected to a minimum. Especially, the number of common spare lines should be carefully controlled. For high repair rate, enough number of global spare lines should be assigned. By doing so, a reasonable memory yield can be guaranteed with a proper area of BIRA module.

IV. CONCLUSION

A deep discussion of redundancy architectures in aspect of repair rate and area overhead is performed in this paper. Various redundancy architectures are used to simulate the performance of local, common and global spare lines. Two new evaluation indexes are suggested to describe the characteristic of each spare line. To improve both repair rate and area overhead, the number of each spare line should be treated properly. We expect that this data analysis helps to figure out cost-effective redundancy architecture.

ACKNOWLEDGMENT

This research was supported by the MOTIE (Ministry of Trade, Industry & Energy) (10052875) and KSRC (Korea Semiconductor Research Consortium) support program for the development of the future semiconductor device.

REFERENCES

