IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723

A Low Latency BIRA Based on A Counting Row Column Threshold Approach

Layyina PV.[1] Bency Varghese A.[2] Linu Babu[3]

Student, Department of Electronics and Communication Engineering,

IES College of Engineering, Thrissur, India[1]

Assistant Professor, Department of Electronics and Communication Engineering,

IES College of Engineering, Thrissur, India[2,3]

Abstract— Semiconductor memories are considered one of the most important aspects of modern VLSI Systems. Memories are the most important universal components in System on chip today. Almost all SOC's contain some type of embedded memories, such as ROM, RAM, DRAM and flash memory. Due to use of deep micron technology the cells are becoming more susceptible to manufacturing defects. Semiconductor memories will occupy the 90% ofthe total chip area by 2016. Testing the memory IP in SOC becomes more important because the memory density higherthan the logic part, which means the chance to have a defect,is higher in memory. The quality of embedded memorydominates the overall quality and profitably of the whole chip. The BIST methodologies offer solutions for testability of embedded memories and minimize the embedded memory tester's requirements and reduce memory test time.

A novel BIRA approach that focuses on a 100% repair rate and a minimal area overhead is proposed in this paper. In the modified method, depending on the error sensitivity and presetting a threshold, the rows with more errors are automatically replaced. After replacement any fault which isnot recovered are replaced with column spare wires. Thus minimal time is only required. Our experimental results confirm that the modified approach exhibits outstanding performance for delay that have 100% repair rates are designed using the tool Xilinx14.2. Coding is done using VHDL and simulation is verified using Xilinx ISE Design Suite. The software used is XILINIX ISE simulator. Implemented using VHDL module.

Keywords: Built-in redundancy analysis (BIRA), area overhead, redundancy analysis (RA).

I. INTRODUCTION

Testing embedded memory is more difficult than testingcommodity memory. The first testing issue is accessibility. Accessing the DRAM core from an external memory testeris costly—in terms of pin/area overhead, performance penalty, and noise issues—when the DRAM core isembedded in a CPU or ASIC and surrounded by logic blocks. Proper Design For Testability (DFT) methodology must be provided for core isolation and tester access, and a price has to be paid for the resulting hardware overhead, performance penalty, and noise and parasitic effects. Even if these are manageable, memory testers for full qualification and testing of Embedded DRAM (EDRAM) will be much more expensive due to increased speed and I/O data width, and if we also consider engineering change the overall investment will be even higher. A promising solution to this dilemma is BIST. With BIST, the tester requirement forEDRAM can be minimized, and memory tester time can be greatly reduced throughout the entire test flow of the DRAM. Also, the total test time can be reduced because parallel testing at the memory bank and chip levels is easier. Therefore, BIST has been widely considered as a must for EDRAMs. In order to enhance the area overhead and achieve a 100% repair rate, we have proposed a novel BIRAapproach. In order to minimize the area overhead, only the essential parts of fault addresses are stored in a new fault- storing CAM structure, with the remainder being stored in spare memories during the fault collection phase. At the endof the fault collection phase, the proposed CAM structure collects the minimum information needed to perform the RA procedure. The proposed RA process greatly reduces solution search trees by using only small parts of the proposed CAM structure. Chapter two of this paper, background information is provided. The proposed BIRA algorithm and some simple examples are provided in Section three Section four describesexperimental results used to evaluate the performance of the proposed BIRA, and Section V concludes this paper

II. RELATED WORK

1) K W. Jeong, J. Lee, T. Han, K. Lee, and S. Kang proposes a new algorithm called BRANCH analyser. Many BIRA approaches require extra hardware overhead in order to achieve optimal repair rates, orthey suffer a loss of

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723

repair rate in minimizing the hardware overhead. An innovative BIRA approach is proposed to achieve optimal repair rates, lower areaoverhead, and increase analysis speed. The proposed BIRA minimizes area overhead by eliminating somestorage coverage for only must-repair faulty information. The proposed BIRA analyzes redundanciesquickly and efficiently by evaluating all nodes of a branch in parallel with a new analyzer

2) A built-in self-repair analyzer (CRESTA) for embeddedDRAMs proposed by T. Kawagoe, J. Ohtani, M. Niiro, T. Ooishi, M. Hamada, and H. Hidaka. A new practical built- in self-repair analyzer algorithm for embeddedDRAMs (e-DRAM) achieves 100% detection ability of the repairable chips with 1% area penalty of the target32 Mb embedded DRAM by 4 parallel analyzers. Itworks at as fast as 500 MHz, well beyond targeted e- DRAMs' maximum operation speed around 200 MHz+. and relatively small

3) W. K. Huang, Y. N. Shen, and F. Lombardi adopted approach, the faulty line covering technique, is a refinement of the fault-driven approach. This approach finds the optimal repair solution within a smaller number of iterations than the fault-driven algorithm.The second approach exploits a heuristic criterion in thegeneration of the repair solution. This heuristic criterionpermits a fast repair. The criterion is based on the calculation of efficient coefficients for the rows andcolumns of the memory. A considerable reduction in processing and complexity (number of records generated in the repair process for finding the optimal repair solution) is accomplished.

4) Nakahara, K. Higeta, M. Kohno, T. Kawamura, and K. Kakitani describes paper presents a built-in self-test (BIST) scheme, which consists of a flexible pattern generator and a practical on-macro two-dimensional redundancy analyzer, for GHz embedded SRAMs Inorder to meet the system requirements and to detect wide variety of faults or performance degradation resulting from recent technology advances, themicrocode-based pattern generator can generate flexible patterns. It can be implemented with simple hardware and can show fairly good performance compared

5) Yervant Zorian and Samvel Shoukourian (2003) has published a research work is used to test and diagnose problems in embedded memories as well as repair them to improve fabrication yield. The proposed design integrates memory core with test and repair corein a composite infrastructure that ensures manufacturing and field repair efficiency and optimizes SOC yield. The resulting repair is permanentand leads to no memory performance degradation. The external-bypass technique also uses redundancymultiplexers and comparators, typically implementedat the RTL outside the memory. This leads to a performance and area penalty compared to the star memory, which has the multiplexers integrated in its layout.

III. SCHEME OF MEMORY BISR

This BISR consisted of a BIST and a BIRA. The BIST generates test patterns that allow for faulty cells within a memory array to be detected. When a faulty cell is detected, the BIST sends the fault address to the BIRA.The fault address is then stored within the fault collection logic of the BIRA. If the BIRA uses CAMs for the fault collection, the incoming fault address can be comparedwith the stored fault addresses within one clock cycle [32]. After this test process, the BIRA algorithm performs a RA procedure based on the information collected about thefaults

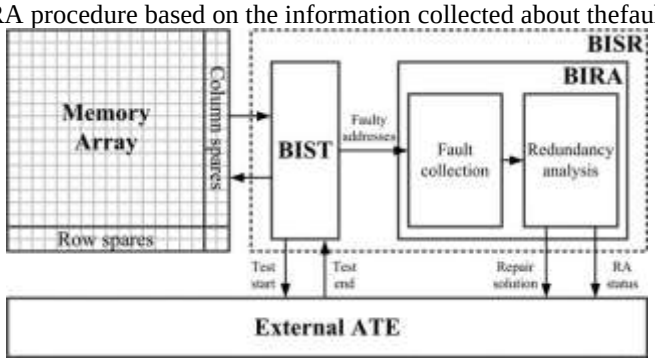


Fig. 3.1: block diagram of a memory BISR

The concept of spare/non-spare pivot faults has been defined. In order to expedite the repair process, all faultsare classified into two groups: spare pivot faults and non- spare pivot faults. During the fault collection phase, if a newly detected fault has a unique row address and column address compared to the collected faults, then the faultbecomes a spare pivot fault. Therefore, spare pivot faultsdo not share row and column addresses with each other. In addition, a set of spare pivot faults or the number of spare pivot faults in an individual memory can differ depending on the order of

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723

fault detection. All faults except spare pivotfaults are defined as non-spare pivot faults. After a spare pivot set is decided, every non-sparepivot fault should share its row or column address with at least one of the spare pivot fault.Fig.3.2 shows an example of a faulty memory with two row spares and two column spares. Inthis example, there are eight faults into an 8-by-8 memory array. A fault detection order is shown in as a number in the top left corner of each fault. At the end of the fault collection phase, the first, second, fifth, and sixth detected faults make the set of sparepivot faults, {(2, 1), (5, 3), (1, 4), (7, 7)}, which is depicted as red-highlighted faults shown in Fig. 2. Because each of these four faults has unique row and column addresses, and every non- spare pivot faults shares a row or column address with one or more of the faults in the set, this set of faults meets the conditions of the spare pivot set

In most cases, spare pivot faults play important roles in the spare allocation problem. Because of their characteristics, spare pivot faults are frequently treated as leading elements of RA procedures. Some theorems and proofs of spare pivot faults are represented in another study.

The most critical concern about spare pivotfaults is that, if the number of spare pivot faults is greater than the number of total spares, then the faulty memory cannot be repaired. At least one spare line should be allocated to each spare pivot fault, because of the uniqueness of the row and column addresses. Asa result, the maximum number of sparepivot faults is the same as the number of total spares. This feature indicates that most of therepair solution addresses can be derived from spare pivot faults. However, the information about non-spare pivot faults is mainly used to assist decisions made for the spare pivot faults.

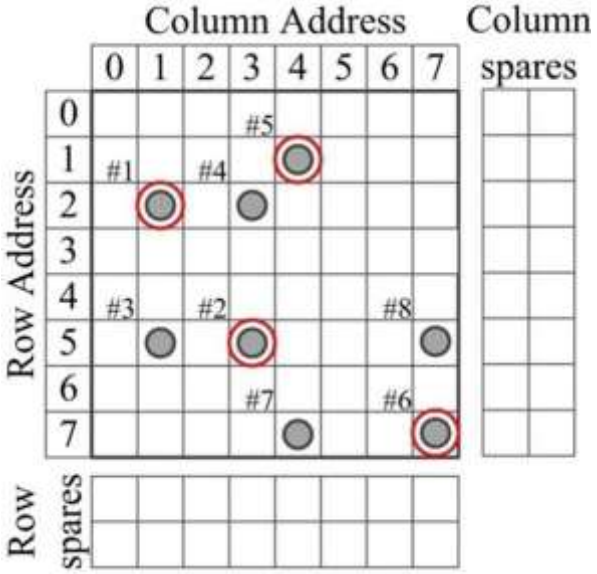


Fig. 3.2: Example for set of spare pivot faults

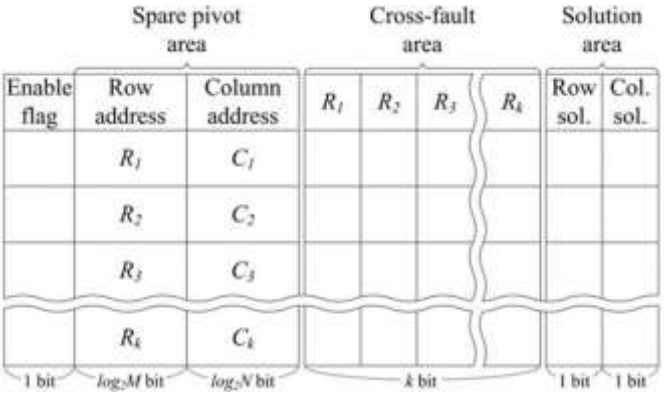


Fig. 3.3: Proposed CAM structure method

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723

IV. FAULT COLLECTION

During a memory test process, the addresses of non-spare pivot faults are stored in spare memories. After the memory test process, the proposed BIRA searches for repair solutions based on the stored addresses. However,the access time for the spare memories is much greater thanthat of the CAMs because of the differences in the search mechanism.

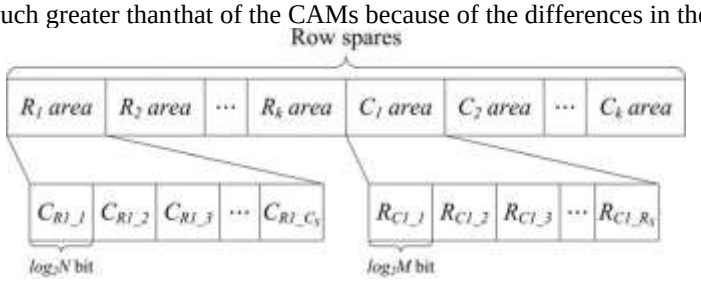


Fig 4.1: Usage of row spare memories

Basically, row spare memories are used to store the addresses of non-spare pivot faults, because they are more convenient to access than column spare memories. Column spare memories also can be utilized as storage elements in caseof the number of required bits for storing non-spare pivotfaults is greater than the number of bits of row spare memories. The storage process is relatively straightforward; once the number of row spares and column spares are settled, certain lengths of bits are assigned to each element of S. Then, whenever anon-spare pivot fault is detected, its address information is stored into the pre-assigned space

V. REDUNTANCY ANALYSIS

After the fault collection process, the proposed BIRAalgorithm determines which addresses can be repairsolutions. In order to obtain a 100% repair rate, anexhaustive search algorithm is adopted using the information of the fault-storing CAM. As mentioned inSection I, the RA problem is known to be NP-complete; however, if the characteristics of the spare pivot faults areproperly utilized, the RA procedure can be greatlysimplified. Under certain conditions, repair solutions can beeasily found The position of the related non-spare pivotfaults that are not cross-faults, is the most critical factor inthe repairing direction of the spare pivot fault. For the sakeof convenience, non- spare pivot faults that are not cross-faults are defined as tail-faults in this paper. Unlike cross-faults, tail-faults are connected with only one spare pivotfault. Therefore, if a related tail-fault exists in a rowdirection, a spare pivot fault should be repaired by a rowspare, and vice versa. In this case, related repair solutionsare directly stored into the solution area of the proposedCAM with no additional RA procedures. This processgreatly reduces the whole trial numbers to find repairsolutions. At this point, some spare lines are allocated forthe repair decisions of the must-repair lines and tail-faults.This means that repair decisions of the cross-fault area areperformed with the row spares and column spares available. If the number of spare pivot faults is less than the number of total spares then the number of solutions searching cases increases. In this condition, the use of spare lines becomes more flexible; for example, both a row spare and column spare can be allocated to one spare pivot fault. In addition, it becomes possible for a spare line to be directly attributable to a tail-fault without the need to go through the related spare pivot fault. Therefore, if needed, the information for the tail-faults can added to the cross-fault area in order to find the repair solution. The basic mechanism of an exhaustive search in this condition is similar to the previous one; however, in this case, the exhaustive search was performed for all of the spare pivot faults.The first step is to check the tail-faults and store the related solution into the solution area. At this point, if thereis no available row spare, then the related column address of the tail-fault can be stored into the CAM so that the proposedBIRA seeks the possibility of repairing this tail-fault using column spares.

However, if there is no available column spare, the related row address of the tail-fault is stored into the CAM inthe same manner The second step is to choose the column addresses in the spare pivot area.. If all of the 1 bits in the cross-fault area can be covered, then the repair solution is stored into the solution area There are four sparepivot faults in the faulty memory depicted within this set. Because the number of total spares is four in this example, only one spare line can be allocated to each spare pivot fault. A close look at the non-spare pivot faults reveals that there isno tail-fault in this example; every non-spare pivot fault is simultaneously connected with two spare pivot faults in the figure shown.

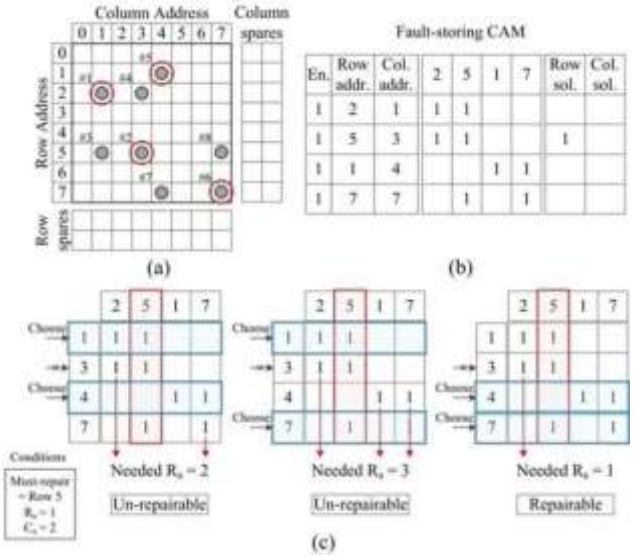
IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723



Fig, 5.1: Example of RA process

As a result, there are four spare pivot faults and four cross-faults in this faulty memory block. During the fault collection process, must-repair lines are detected so that the solution of the spare pivot fault (5, 3) is decided by the repairing row address 5. Because one row spare is already allocated, the number of available row spares is one, and the number of available column spares is two..However, because only one row spare is left, this faulty memory cannot be repaired. Similarly, if column addresses 1and 7 are chosen, the faulty memory cannot be repaired, because it needs three available row spares. If column addresses 4 and 7 are chosen, faults (2, 1) and (2, 3) are left, and these faults can be repaired by one row spare. Therefore,the repair solution is to repair row addresses 2 and 5 and column addresses 4 and 7 From the definition, faults (2, 3), (1, 4), and (5, 3) become tail-faults. However, the fault (1, 4)cannot be covered by a column spare, because there is no available column spare. As a result, the related row address 1 is stored into the CAM. In order to repair the rest of Tail faults ,and the must repair line, row addresses 2 and 5 have to be repaired. The related information is stored into the solution area, because there is no column spare. Only one case is considered in exhaustive search.

VI. MODIFIED APPROACH

Today's deep submicron technologies allow the implementation of multiple memories on a single chip. Dueto their high density, memories are more prone to faults. These faults impact the total chip yield. One way to solve this problem is to enhance the memory by redundant memory locations. In order to do so, a memory test is needed to identify the faulty regions. The memory is tested by external test hardware or by on chip dedicated hardware (memory BIST). The second testing strategy is the preferred method for embedded memories The redundant spare rows and spare columns are often included into the memory.

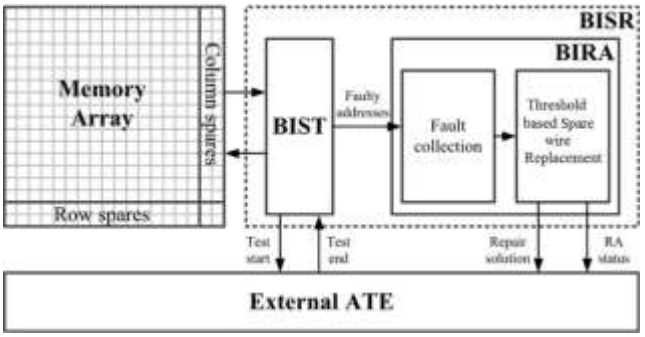


Fig:6.1 Block Diagram of Modified Approach

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723

The existing infrastructure is capable of detecting faults in memory array and repairing the memory array using spare wires. BIST. The Memory BISR (MBISR) concept contains an interface between memory BIST (MBIST) logic and redundancy wrapper for storing faulty addresses. However, redundancy increases the silicon area and thus has a negative impact on yield. memory. The memory is repaired during testing by storing faulty addresses in registers. To maximize the yield, redundancy analysis is necessary. BISR (Built in Self Repair) technique is a promising and popular solution for enhancing the yield of memories with the redundancy logic

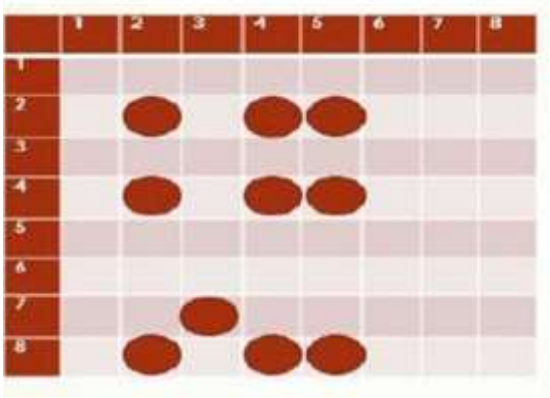


Fig. 6.2: Example of Faulty memory

This infrastructure adopts the framework of the Counting threshold based row column algorithm for repair Analysis and consists of must-repair analysis and final analysis. The must-repair analysis identifies must-repair rows and columns, and the final analysis searches a repair solution. The must-repair analysis is performed concurrently with the test, while the final analysis is done after the test is completed. The MRA consists of a pair of CAMs for fault addresses, called the fault-list, and a pair of CAMs for a repair solution, called the solution record.

Counting threshold based row column algorithm is a heuristic method based on a binary search tree. Although thebranch-and-bound algorithm is faster than the exhaustivebinary search algorithm, it does not significantly reduce the time or memory space needed to search for a solution. This algorithm determines an optimal solution for repairing memory faults in a short amount of time. It is also the fastestand easiest algorithm to develop or implement in ATE tests. Furthermore, it uses a small amount of memory to solve the spare location problem.Therefore, to overcome the disadvantages of existing algorithms, here algorithm is based on threshold logic in proposed system model. This achieves optimal repair rate with moderate area overhead, increased test speed and reduced test time. Cost of each repair solution generated by the algorithm is evaluated and the solution which consumes minimum cost is chosen

Figure 6.2 shows an example of a faulty memory. In this example, there are ten faults into an8-by-8 memory array. In the second, fourth and eighth row, the 2nd,4thand 5th bits are faulty and in the seventh row the 3rd bit is faulty. Once the number of faults and the faulty locations are found they have to be replaced with spare wires. In the Threshold technique, a threshold limit is kept. In this example the threshold is kept as

2. All the row shaving faults greater than 2 will get replaced parallelly. That is the second, fourth and eighth row will get replaced. The seventh row will still be left out as it has only 1 fault which is less than the threshold. Now the spare columnis allocated for column 3. Hence all the faulty addresses are corrected using the threshold-based approach resulting in reduced area, delay and power consumption.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723

VII. COMPARISON AND RESULTS

The coding is done using VHDL language and thesimulation is verified using Xilinx. Figures below shows thesimulation result of efficient BIRA using Spare Pivot fault and row column threshold algorithm

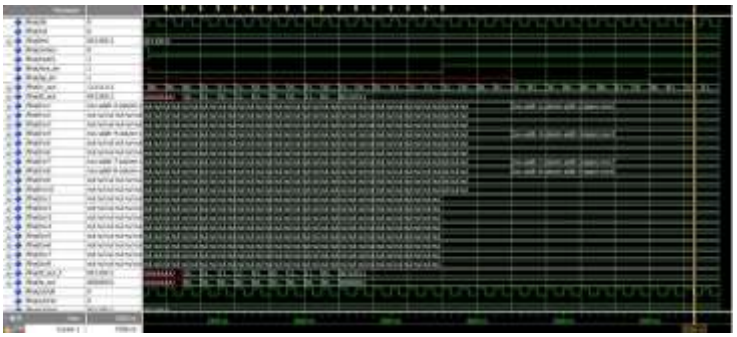


Fig 7.1: Simulation Results of existing Method

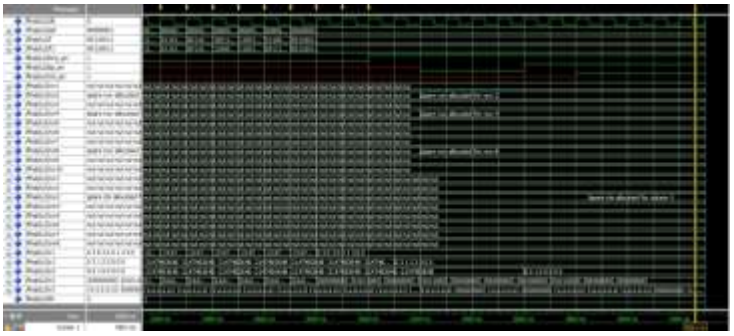


Fig 7.2: Simulation results of modified approach

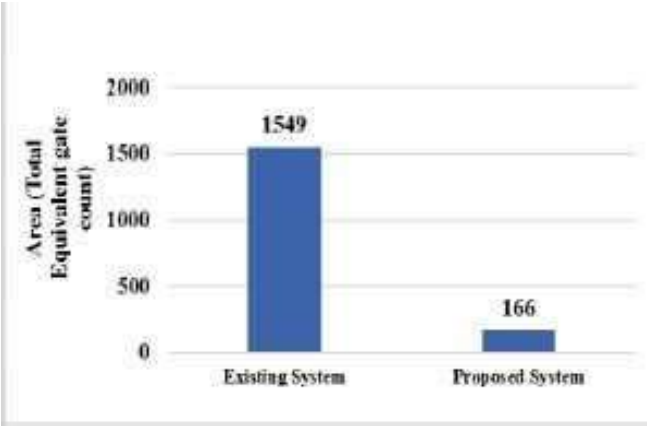


Fig 7.3: Area comparison of both existing and modified

As the logic devices are reduced the power taken for each algorithms also differ greatly.Figure below shows the total estimated power consumption of the existing method

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 7, July 2021

DOI 10.17148/IJIREEICE.2021.9723

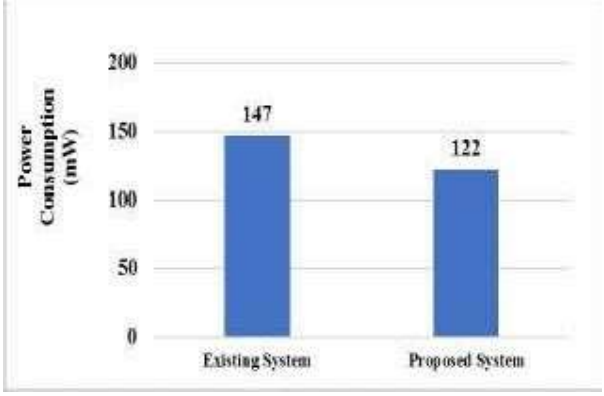


Fig 7.4: power Comparison of existing and modified

Figure below shows the comparison of existing and modified BIRA The delay for the existing architecture is 8.39ns and it consists of 3 levels of logic And using threshold row column approach. Here the delay taken is3.140ns which is more than 50% faster than existing method.

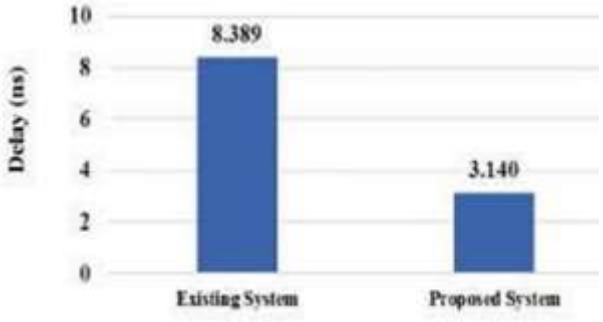


Fig 7.5 : Delay comparison of both existing and modified

VIII. CONCLUSION

This paper carries out a detailed investigation of the faults, and it proposes an efficient BIRA approach using the characteristics of these faults. The proposed BIRA algorithm using threshold technique can acquire the smallest area over head and can have a 100% repair rate. The experimental results from both the circuit an dapplication levels demonstrate that theproposed system delivers greater improvements in energy saving, design area and reducing delay. From this paper it can be concluded that the proposed BIRA approach using threshold technique and spare wire allocation has the smallest area (166 Total Equivalent gate count for design), delay (3.140 ns) and power consumption (122 mW) compared with the existing method. In conclusion, it is expected that the proposed BIRA approach will prove to be practical solution to the yield and reliability problems commonly encountered in commodity memories

ACKNOWLEDGMENT

I would like to convey my heartfelt gratitude towards my guide Ms. Linu Babu for her constant guidance, encouraging help and inspiring words. I am thankful to the department of Electronics and communication engineering for their support

REFERENCES

[1] W. Jeong, I. Kang, K. Jin, and S. Kang, "A fast built-in redundancy analysis for memories with optimal repairrate using a line-based search tree," IEEE Trans. Very Large Scale Integr. Syst., vol. 17, no. 12, pp. 1665-1678, Dec. 2009

[2] T. Kawagoe, J. Ohtani, M. Niiro, T. Ooishi, M. Hamada, and H. Hidaka, "A built-in self-repair analyzer (CRESTA) for embedded DRAMs," in Proc. Int. Test Conf., Oct. 2000, pp. 567-574.

[3] W. K. Huang, Y. N. Shen, and F. Lombardi, "New approaches for the repairs of memories with redundancy by row/column deletion for yield enhancement," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 9, no. 3, pp. 323-328, Mar. 1990.]

[4] S. Nakahara, K. Higeta, M. Kohno, T. Kawamura, and K.Kakitani, "Built-in self-test for GHz embedded SRAMs using flexible pattern generator and new repairalgorithm," in Proc. ITC, 1999, pp. 301–310.

[5] Y. Zorian and S. Shoukourian, "Embedded-memory test and repair: Infrastructure IP for SoC yield," IEEE Des. Test Comput., vol. 20, no. 3, pp. 58–66, May–Jun. 2003