ISSN No: 2348-4845
International Conference on Electronics, Communications and VLSI Circuits (ICECV-2015)
July 10, 2015 - Hyderabad, India
Papers Published in IJMETMR, A Peer Reviewed Open Access International Journal.

Design of a High speed Built-in Repair Analyser for Word-Oriented Memories

Mulagundla Laxmi
MTech Student
Department of ECE
Vaagdevi Engineering College
Bollikunta, Warangal

B.Shivakumar
Asst. Professor
Department of ECE
Vaagdevi Engineering College
Bollikunta, Warangal

M. Swetha
Asst. Professor
Department of ECE
Vaagdevi Engineering College
Bollikunta, Warangal

Abstract- ***Here we introduced a new method called Built in self-repair analyzer for the memory arrays. In this detection of errors is done by a single test. By performing the must-repair analysis on the fly during the test, it selectively stores fault addresses, and the final analysis is performed on the stored fault addresses to find a solution. The memory required to store the fault addresses is dominated by total area of our infrastructure. It grows quadratically with respect to the number of repair elements. This architecture also to support various types of word-oriented memories.***

Keywords- ***Must repair Analyzer, Solver, Spare rows and columns, Built in repair analyzer***.

1.Introduction

Today's system-on-chip (SoC) environment requires Significant changes in testing methodologies for memory arrays. The failure of embedded memories in a SoC is more expensive than that of commodity memories because a relatively large die is wasted. Due to the large die size and the complex fabrication process for combining memories and logic, SoCs suffer from relatively lower yield [1]. To improve the yield, memory arrays are usually equipped with spare elements, and external testers have been used to test the memory arrays and configures the spare elements. the SoC environment, combined with shrinking technology, allows us more area for on-chip test infrastructure at lower cost than before, which makes feasible a variety of built-in self test (BIST) and built in self-repair (BISR) techniques for reducing the test time.

A built-in self-test (BIST) or built-in test (BIT) is a mechanism that permits a machine to test itself. The main purpose of BIST [2] is to reduce the complexity, and thereby decrease the cost and reduce manufacturing tests. The IC has a function that reliance upon external test equipment. BIST is used to make faster, less-expensive integrated circuit verifies all or a portion of the internal functionality of the IC.

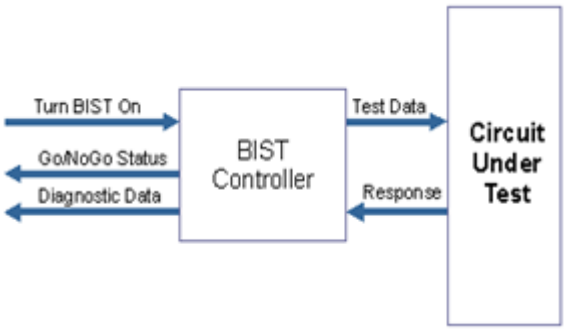

Fig.1. Basic BIST block diagram

Built-in redundancy allocation (BIRA) [3] approaches have been proposed as part of BISR. Kawagoe et al. propose a pioneering BIRA approach, CRESTA. CRESTA [9] has the sub-analyzers for all solution candidates, which provides the optimal repair rate with a single test. The sub-analyzer consists of a row content addressable memory (CAM) with r entries (r is the number of repair rows) and a column CAM with c entries (c is the number of repair columns) Since this may not be affordable in memories with many spare elements, subsequent studies have been focused on reducing hardware complexity.

In order to lower the hardware complexity and still guarantee the optimal repair rate, another approach is proposed. This evaluates each possible solution one by one and thus does not require the parallel sub-analyzers [6]. Such serial implementations may

ISSN No: 2348-4845
International Conference on Electronics, Communications and VLSI Circuits (ICECV-2015)
July 10, 2015 - Hyderabad, India
Papers Published in IJMETMR, A Peer Reviewed Open Access International Journal.

increase the overall test time, but the number of possible solutions is reduced using the must-repair analysis.

2. Preliminaries

In the classical spare allocation problem, we consider a bit oriented memory array with spare (repair) rows and spare (repair) columns [11]. Any fault row (column) can be replaced with a spare row (column). If a repair solution exists for a memory array, the memory array is repairable. If a row has more than c(r) faults and a repair row is not used for the row (Column) the memory is not repairable.

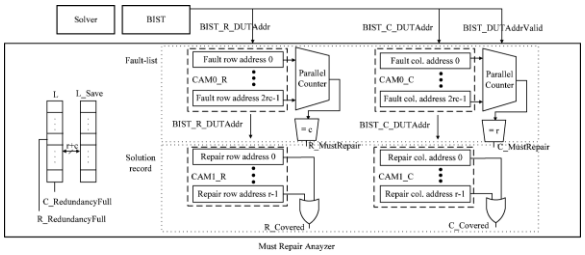

Fig. 2. Proposed on-chip infrastructure and must repair analyzer details

3. Implementation for WOM

There are various types of word-oriented repairable memories, and they impose different constraints on the spare allocation problem. Since it is difficult to capture all the different types of repairable memories into a generalized model and to design an universal repair analyzer, we categorize them into three types, which will be called type A, type B, and type C, respectively. A faulty row is replaced with a spare row, but the way to replace a faulty column varies, based on which they are classified.

In word-oriented memories, the data in a word is usually not placed in adjacent locations due to several issues such as the coupling effect, and the columns associated with the same bit position are clustered together. In type A memories, there are spare column-groups of ω columns each. A group of columns associated with a word is replaced with a space-column group. In other words, the column replacement is performed on a column group basis. If the first bit line in group 0 is faulty, and it is replaced with the first spare column in group 0, then the first bit lines in the other groups are also replaced with the associated first spare columns, respectively.

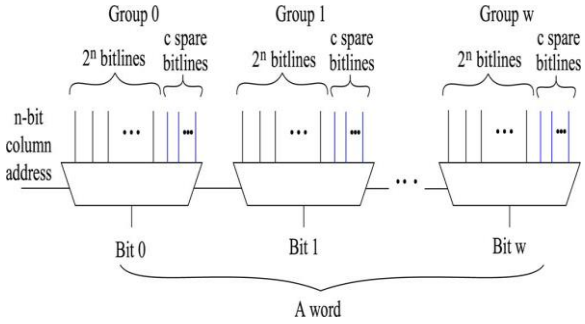

Fig. 3. Column circuitry of a word-oriented memory of type A

Type B:

In type B memories, a faulty column is replaced with a spare column, but among a group of columns associated with a word, only one column can be replaced. A word-oriented memory of type B has only spare columns, each spare column is selected when a programmed column address is accessed. Up to faulty columns can be replaced, but columns that constitute a word cannot be replaced together.

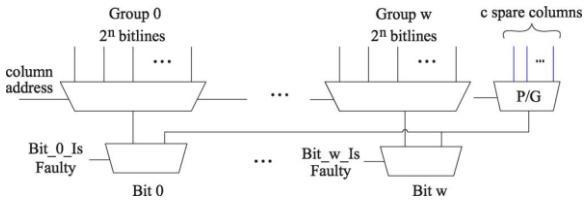

Fig. 4. Column circuitry of a word-oriented memory of type B

Type C:

In type c, any faulty column [7] can be replaced with an available spare column without any restriction.

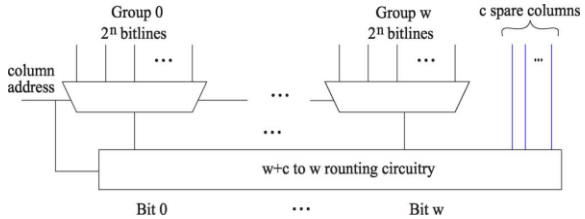

Fig. 5. Column circuitry of a word-oriented memory of type C

4. Proposed Infrastructure

ISSN No: 2348-4845
International Conference on Electronics, Communications and VLSI Circuits (ICECV-2015)
July 10, 2015 - Hyderabad, India
Papers Published in IJMETMR, A Peer Reviewed Open Access International Journal.

We propose an on-chip infrastructure for bit oriented memories, later it will be extended for word oriented memories. We assume that an arbitrary BIST engine tests a memory array and provides fault addresses whenever detected. Our infrastructure 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.

If a particular row or column is identified as must-repair, the MRA shown in fig.2 writes the row or column address in the solution record. The L registers are used as valid bits for the solution record and also determine the next available CAM entry. Since a must-repair row and a must-repair column can be identified by a fault at the same time, the MRA should be able to write a pair of row and column addresses simultaneously. Once a row or column address is stored as part of solution by the Must-repair condition, then all solution candidates considered by the SOLVER includes the address, and faults on the address do not affect the final analysis any more. Such faults do not need to be stored, and we can collect all necessary information for the final analysis during a single test.

Once the test is completed BIST_Done signal is asserted and the final analysis is started. In the final analysis, the SOLVER module controls the MRA. The operation of the SOLVER and the MRA in the final analysis phase is illustrated in Fig. 6. The SOLVER will generate repair strategies one by one and will check whether each repair strategy can fix all the faults captured in the fault-list.

5. Operation of SOLVER

The SOLVER generates repair strategy and the MRA reads each fault address in the fault-list in order until the RESTART signal is arrived. The MRA checks if each fault is covered by the current solution, stored in the solution record, and assert R_Covered or C_Covered. If both signals are low, the fault should be covered by a new repair row or column. The SOLVER [8] determines whether a repair row or column is used for the uncovered fault, and asserts R_Insert or C_Insert. If R_Insert (C_Insert) is high, the fault row (column) address is written in the row (Column) CAM of the solution record. If the CAM is full, the memory array cannot be repaired by the first repair strategy, and the SOLVER generates the next repair strategy and asserts the RESTART signal.

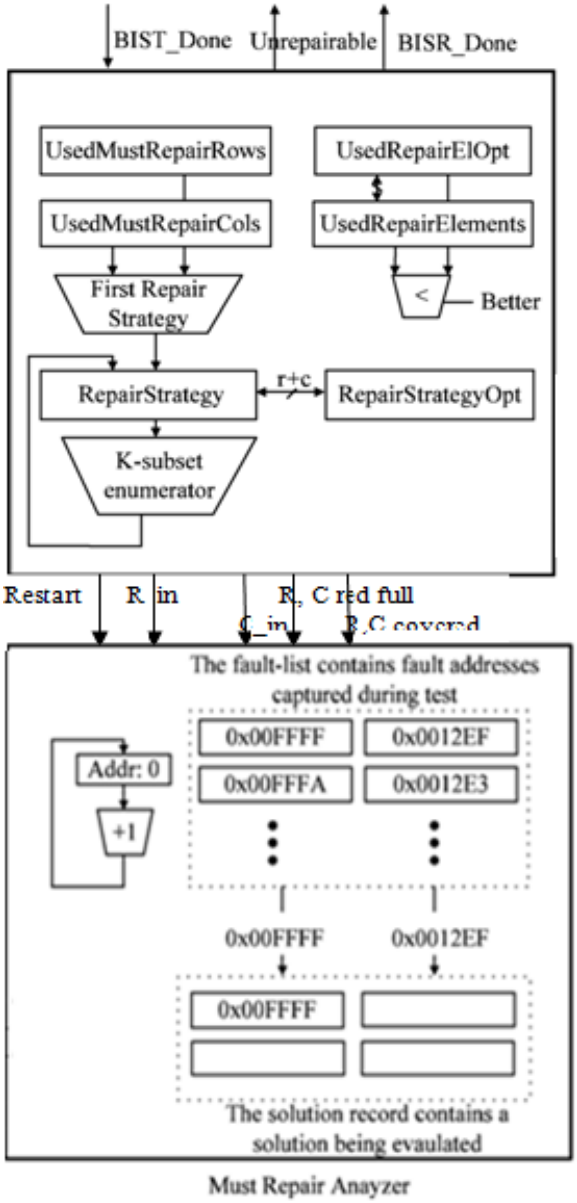

Fig.6. Solver details and MRA operation in the final analysis phase

ISSN No: 2348-4845
International Conference on Electronics, Communications and VLSI Circuits (ICECV-2015)
July 10, 2015 - Hyderabad, India
Papers Published in IJMETMR, A Peer Reviewed Open Access International Journal.

When the RESTART signal becomes high, the MRA restores the initial state, and the next repair strategy starts being evaluated. In this way, the SOLVER explores the solution space and can find a solution if one exists.

6. Extension for WOM

We will extend the proposed infrastructure for word-oriented memories. Unlike the bit-oriented memory case, from the BIST [10] engine, our infrastructure takes as input a triplet (R_X,C_X, S), where R_X (C_X) is the row (column) address, and S is the failure syndrome, which is the exclusive OR of the test response and the expected output of the word at 〖(R〗_X,C_X) . For word-oriented memories of type A, we can discard the failure syndrome and can input only the row and column addresses. Then without any modification, it will perform repair analysis for type A word-oriented memories.

Type B:

For a word-oriented memory of type B, Let ω be the word size of the device under test (DUT). We will map the word-oriented memory to a bit- oriented memory. Since every bit should be addressable in the bit-oriented memory, we expand the width of the column address by [log w] to distinguish each column within a word. We call the extended address the virtual column address [5]. In this case, a triplet can generate up to ω virtual column address for the bit-oriented memory. However, in the case that the number of "1"s in S is greater than 1, it is obvious that the row being tested is a must-repair row since the DUT is 1 column-per-word replaceable.

If this case is handled separately, one triplet will generate only one virtual column address. The pair of the incoming row address and the virtual column address is fed into the proposed infrastructure, which will work with the word-oriented memory of type B.

Type C:

In word-orient memories, ω bits (columns) have the same address. However, in order to repair such memories on a column basis, we need to distinguish each column. So we define the extended column address as a pair of a column address and a word of ω bits, each of which corresponds to one among the ω columns indicated by the column address. We call a pair of a row address and an extended column address the extended fault address. Multiple extended fault addresses indicating each fault with in a word can be combined into a single extended fault address.

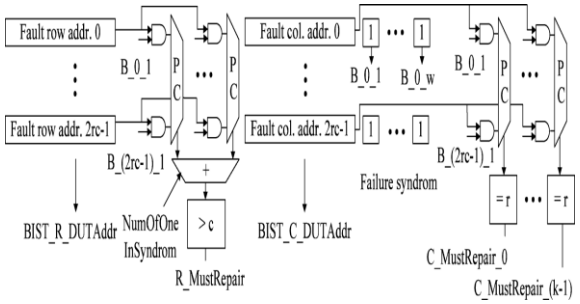

Fig. 7. Modified must repair analyzer for word-oriented memories of type C.

In the modified MRA, the Column CAM in the fault-list needs to store the extended Column addresses, and each entry has additional bits. Each bit has its own match signal, and each entry has ω+1 match signals including the original match signal. These signals are fed into the logic for generating the R_Must Repair signal. The number of the faulty cells is added to the number of 1's in the incoming fault syndrome, and if it is greater than c, then the must repair condition is satisfied and the R_MustRepairsignal is asserted. The new match signals are also used to generate C_Must Repair_1,….,C_Must Repair_ ω, each of which is asserted [4] when each column within a word satisfies the must-repair condition.

7. Simulation results

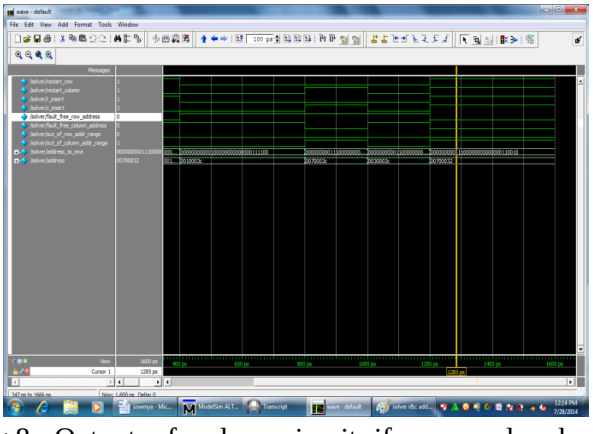

Fig.8. Output of solver circuit if row and column address is fault

ISSN No: 2348-4845
International Conference on Electronics, Communications and VLSI Circuits (ICECV-2015)
July 10, 2015 - Hyderabad, India
Papers Published in IJMETMR, A Peer Reviewed Open Access International Journal.

Fig.9. shows the simulation result of solver circuit for fault free row and column address.

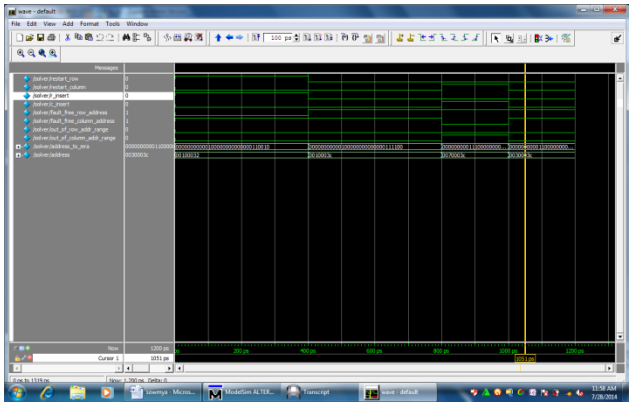

Fig.9. output of solver for fault free row and column address

Fig.10. shows the simulation result of Must repair analyser for row and column address within range.

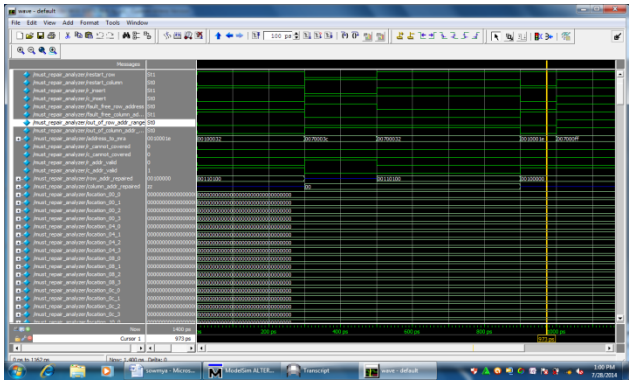

Fig.10. output of MRA for row & column address within range.

Fig.11. shows the simulation result of Must repair analyser for row and column address out of range.

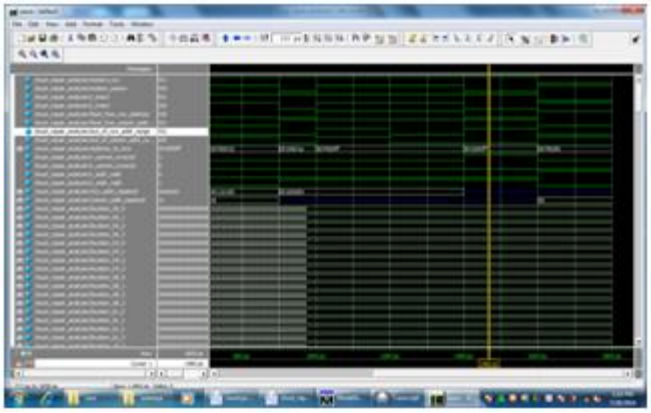

Fig.11.output of MRA for row & column address out of range

Fig.12. shows the simulation result of the final analysis is when row address is not valid.

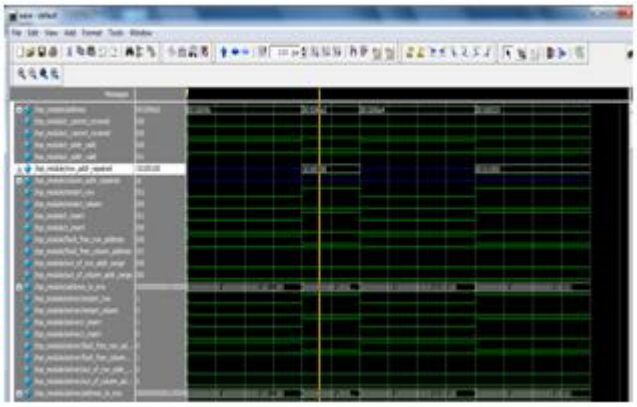

Fig.12. output of main circuit for row address not valid

Fig.13. shows the output result of the final analysis when column address is not valid.

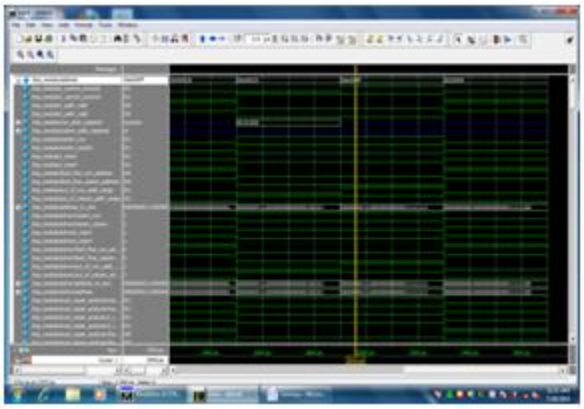

Fig.13. output of final analysis when column address is not valid

Fig.14. shows the simulation result of final analysis when row and column address are fault free.

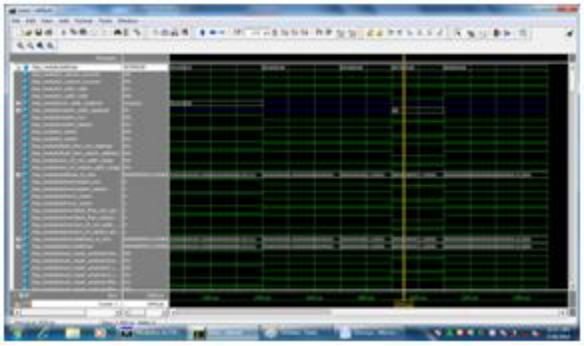

Fig.14. output of final analysis with row & column address valid.

ISSN No: 2348-4845
International Conference on Electronics, Communications and VLSI Circuits (ICECV-2015)
July 10, 2015 - Hyderabad, India
Papers Published in IJMETMR, A Peer Reviewed Open Access International Journal.

Fig.15. shows the simulation result of final analysis when row and column address are out of range.

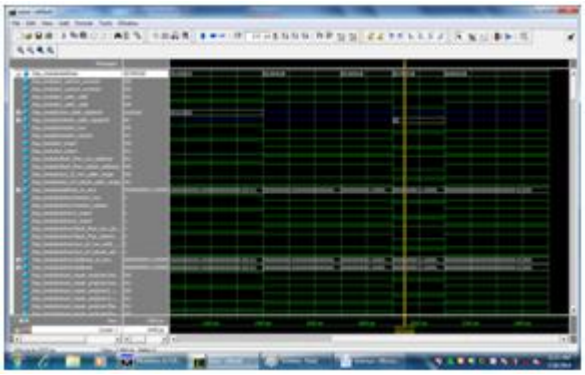

Fig.15. output of final analysis with row & column address out of range.

8. Conclusion

We have proposed an on-chip infrastructure for repair analysis with the optimal repair rate. Our infrastructure requires a single test. Most built-in repair analysers are developed for bit-oriented memories, whereas our repair analyser also aims at various types of word-oriented memories. To achieve this, we have extensively studied existing word-oriented repairable memories and have classified them into three types. For each type, we have showed how the bit-oriented version can be extended.

9. References

[1] Y. Zorian and S. Shoukourian, "Embedded-memory test and repair: Infrastructure IP for SOC yield," IEEE Design Test Compute., vol. 20, no. 3, pp. 58–66, May/Jun. 2003.

[2] P. Oehler,S. Hellebrand, and H.-H. Wunderlich, "An integrated built-in test and repair approach for memories with 2D redundancy," in Proc. Eur. Test Symp., 2007, pp. 91–96.

[3] W. Jeong, J. Lee, T. Han, K. Lee, and S. Kang, "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.

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

[5] S. Shoukourian, V. A. Vardanian, and Y. Zorian, "A methodology for design and evaluation of redundancy allocation algorithms," in Proc. VLSI Test Symp., 2004, pp. 249–255.

[6] E. E. Swartzlander, "Parallel counters," IEEE Trans. Comput., vol. 22, no. 11, pp. 1021–1024, Nov. 1973.

[7] C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, "Built-in redundancy analysis for memory yield improvement," IEEE Trans. Reliab., vol. 52, no. 4, pp. 386–399, Dec. 2003.

[8] A. Ferris and G. Work, "Memory circuit capable of replacing a faulty column with a spare column," U.S. Patent 5 163 023, Nov. 10, 1992.[9] T. Kawagoe, J. Ohtani, M. Niiro, and T. Ooishi, "A built-in self-repair analyzer (cresta) for embedded drams," in Proc. Int. Test Conf., 2000, pp. 567–574.

[10] S. Hamdioui, G. Gaydadjiev, and A. van de Goor, "The state-of-art and future trends in testing embedded memories," in Proc. Records Int. Workshop Memory Technol., Design, Test., 2004, pp. 54–59.

[11] P. Oehler, A. Bosio, G. D. Natale, and S. Hellebrand, "A modular memory BIST for optimized memory repair," in Proc. Int. On-Line Test. Symp., 2008, pp. 171–172.

Author-1:

Ms. Mulagundla Laxmi is persuing her M. Tech in Vaagdevi Engineering College, Bollikunta, Warangal and completed her B.Tech from Ganapathy

Engineering College, Hunter Road,Warangal. She is intrested in the area of VLSI.

Author-2:

Mr.B.Shivakumar (Asst.Prof in Vaagdevi College of Engineering). He has completed his M.Tech from Vaagdevi Engineering College and he is intrested in this area VLSI.

Author-3:

Mr. M. Swetha Maheswaram (Asst.Prof in Vaagdevi College Of Engineering). She has completed her M.Tech from Vaagdevi Engineering College and she is intrested in the field of VLSI.