International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.4, July-Aug. 2012 pp-2458-2463 ISSN: 2249-6645

Global Block-Based Redundancy Architectures For Self-Repairing Of Embedded Memories

T. Govinda Rao[1], K. V. V. Satyannarayana[2], J. Sathish Kumar[3]
*(Department of ECE, Usha Rama College of Engg. & Tech., Telaprolu, India)
**(Department of ECE, Usha Rama College of Engg. & Tech., Telaprolu, India)
***(Department of ECE, Usha Rama College of Engg. & Tech., Telaprolu, India)

ABSTRACT: *In this paper, global block-based redundancy architectures are proposed for self-repairing of embedded memories. The main memory and the redundant rows/columns are divided into row blocks/column blocks. The replacement of faulty memory cells can be performed at the row/column block level instead of the traditional row/column level. Note that the redundant row/column blocks are global, i.e., they can be used to replace faulty cells anywhere in the memory array. This global characteristic is helpful for repairing cluster faults. Since the redundancy can be designed independently. It can be easily integrated with the embedded memory cores. To perform redundancy analysis, the MESP algorithm suitable for built-in implementation is also proposed. According to experimental results, the area overhead for implementing the MESP algorithm is almost negligible. Due to the efficient usage of the redundancy, the manufacturing yield, repair rates, and reliabilities can be improved significantly*

Keywords: *Cluster fault, embedded memory, modified essential spare pivoting (MESP) algorithm, repair rate, yield.*

I. INTRODUCTION

As we migrate to the nanometer technology era, we are seeing a rapid growing density of system-on-a-chip (SOC) designs. This trend makes the manufactured chips more susceptible to sophisticated defects. Moreover, according to the Semiconductor Industry Association (SIA) and Technology Roadmap for Semiconductors (ITRS) 2007, the relative silicon area occupied by embedded memories will approach 94% by 2014 [1]. Since embedded memories also have higher density than logic cores, the overall SOC yield is dominated by these memory cores. For example, the yield of a 24 M-bit embedded memory is about 20% [2]. In order to boost the manufacturing yield of embedded memories, one promising solution is the built-in self-repair (BISR) technique. To achieve the goals of BISR, three basic functions are usually required—memory built-in self-test (BIST), built-in redundancy analysis (BIRA), and address reconfiguration (remapping) (AR).

There are many BIRA/BISR techniques proposed in the past [3]–[12]. A comprehensive exhaustive search for built-in self analysis algorithm (CRESTA) has been proposed in [8]. Al-though this BISR scheme can achieve optimal repair rates, the hardware overhead for implementing the CRESTA scheme is very high.Wey and Lombardi in [9] propose a branch-and-bound technique with early screening in the repair process. A bipartite graph is applied to obtain the least required number of spare allocation, i.e., the least number of spare rows/columns to repair the faulty rows/columns in the main memory array. In our previous work [10], hybrid redundancy architectures are proposed. We have also shown that the complexity of the redundancy allocation problem is nondeterministic-polynomial-time-complete (NP-complete). From the simulation results, the manufacturing yield, repair rate (the ratio of the repaired memories to the number of defective memories), and reliability can be improved significantly.

Divided word-line (DWL) and divided bit-line (DBL) techniques are proposed in [13] and [14] to reduce the power consumption and increase the access time. This divided characteristic can also be used for fault-tolerant applications, i.e., redundant rows and columns can be divided into row blocks and column blocks. The replacement can be performed at the block level instead of the traditional row/column level. However, if the redundant rows (columns) are used locally to replace faulty row (column) blocks in the same row (column) bank, there are still some drawbacks, which should be dealt with. First, if there are many faulty cells located within a row/column bank (to be defined later), there may not be sufficient spare blocks to re-place these faulty cells. The spare blocks allocated for other banks may not be used in this case. Therefore, for some defect distribution resulting in such condition, more redundant rows (columns) should be included for successful repair of the faulty memory. The usage of spares then is inefficient. Second, if *cluster faults* are considered, this dilemma becomes more severe. This is because cluster faults usually gather within a small memory area (though there are small-area, medium-area, and large-area clusters). The local characteristic of redundancy is not suitable (or efficient) to repair such faults. Third, the local spare row (column) blocks should be implemented together with the main memory array, but then it is difficult for SOC integration.

In this paper, the *global* block-based redundancy architectures is proposed. Redundant rows/columns are still divided into row/column blocks. However, the redundant row/column blocks can be used to replace faulty row (column) blocks anywhere in the memory array.This *global* characteristic is helpful for repairing cluster faults.

Based on the proposed global redundant architectures, a heuristic modified essential spare pivoting (MESP) algorithm suitable for BISR is proposed. The area overhead for implementing the MESP algorithm is very low for easier discussion of the proposed techniques.

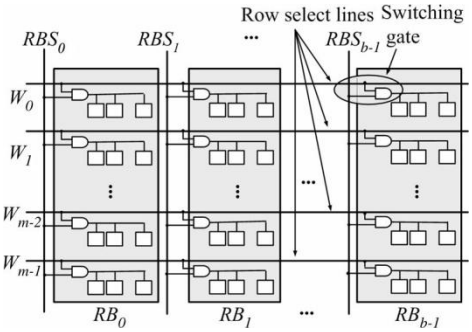

Fig. 1. DWL architecture.

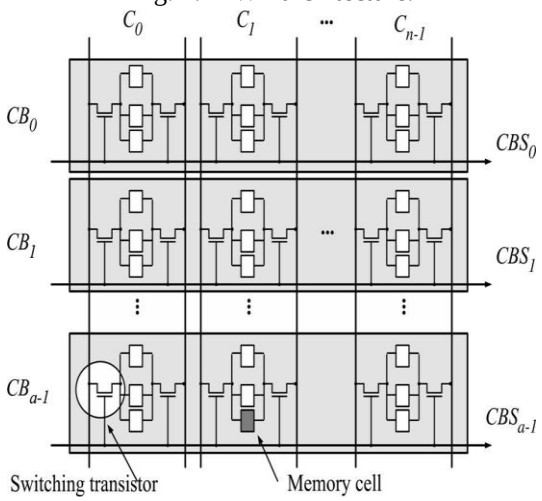

Fig. 2. DBL architecture.

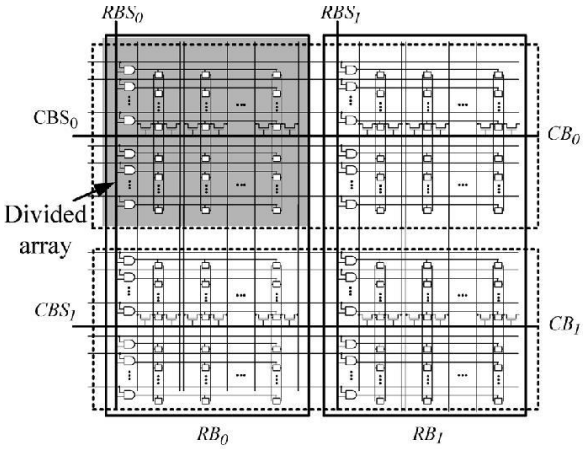

Fig. 3. DWL and DBL integration.

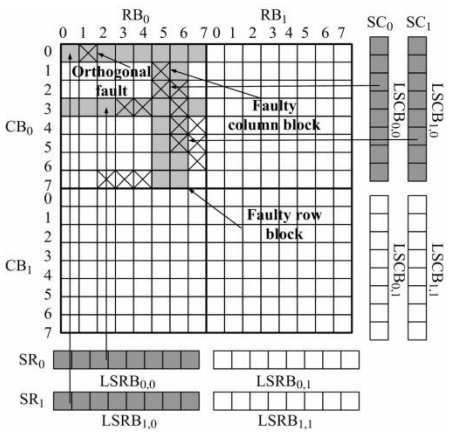

Fig. 4. Local block-level redundancy architecture.

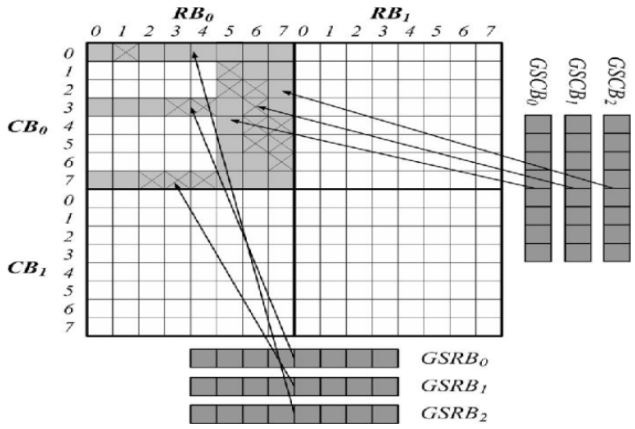

Fig. 5. Novel redundancy architecture (GBLRA).

The remainder of this paper is organized as follows. Section II reviews the DWL and DBL architectures. Section III introduces the proposed local and global block-based redundancy architecture. A heuristic built-in redundancy analysis algorithm named the MESP algorithm is described in Section IV. An example of the proposed MESP algorithm is delineated in Section V. Section VI describes the BISR circuits and the repair procedures. Experimental results are shown in Section VII. A practical 16 K $\times$ 32-bit memory chip with BISR is implemented and described in Section VIII. Finally, some conclusions are given in Section IX.

II. LOCAL AND GLOBAL BLOCK BASED REDUNDANCY ARCHITECTURES

As described above, the local block-based redundancy archi-tectures have three major drawbacks. An example is shown in Fig. 4 where faulty memory cells are marked X's. From this figure, we can see that all faulty cells are contained in DA_{00}. Therefore, only the spare blocks in RB_0 and CB_0 can be used to replace faulty cells. An *orthogonal fault* (to be defined later) should be repaired by one spare row or column block. It is ev-ident that the remaining faulty cells cannot be successfully re-paired by using the remaining spare blocks.

The four used spare blocks are shaded, as shown in Fig. 4.

However, there are still two available spare row blocks ($LSRB_{01}$ and $LSRB_{11}$) and two spare column blocks ($LSCB_{01}$ and $LSCB_{11}$). If these four spare blocks can also be used to replace the remaining faulty cells, it is evident that this faulty memory array can be repaired successfully. This scenario is more severe if cluster faults are considered. There-fore, in order to solve these drawbacks, the *global block-based redundancy architectures* are proposed in this paper.In Fig. 5, the faulty main memory array is the same as that shown in Fig. 4. However, three *global spare row blocks*(*GSRB*s) and three *global spare column blocks* (*GSCB*s) are added into the memory array. Based on this strategy, spare row blocks and column blocks are global, i.e., they can be used to replace the faulty cells anywhere in the memory array. Due to the global characteristic, we can reconfigure the memory more efficiently—the faulty memory array can be repaired successfully as indicated by the arrows shown in Fig. 5.

III. BUILT-IN REDUNDANCY ANALYSIS

The proposed MESP algorithm is based on the *essential spare pivoting (ESP) algorithm* [15]. However, we use the global block-based redundancies instead of the whole rows/columns. Before illustrating the proposed algorithm, we define some basic fault types (FTs) first.

1) **Faulty row (column) block:**
A row/column block that has more faulty cells than the threshold number (E_{th}) is called *a faulty row/column block*. We can use a $GSRB/GSCB$ to replace the faulty row/column block. In this work, we assume that the threshold is 2.

2) **Orthogonal fault** [15]: For a faulty memory cell, if there is no other faulty cell located in the same row and column containing the faulty cell, then this faulty memory cell is

Fig. 6. Fields of the FCR.

said to have an *orthogonal fault*. For each orthogonal fault,a GSRB is required to replace it.To implement the proposed MESP algorithm, it is necessary to use the *fault collection registers* (FCRs) for fault collection. The FCRs contain r+c entries, where and denote the numbers of GSRBs and GSCBs, respectively. During the BIST session, when a faulty cell is detected, fault information is stored into the FCR registers. There are four fields contained in each FCR, as shown in Fig. 6. The D'A field is used to store the index of the divided array containing the faulty cells. The local row (column) addresses of the faulty cells are stored in the LRA (LCA) field. The FT field identifies if the faulty memory cells are orthogonal faults (FT=0) or faulty row (FT=1)/column blocks (FT=2).

IV. BISR ARCHITECTURES AND PROCEDURES

Fig. 9 depicts the block diagram of the proposed BISR scheme that includes the BIST module, the BIRA module, and the redundant memory module containing the GSRBs and the GSCBs. The BIST module performs a specified March algorithm to detect functional faults in the redundant memory module and the main memory module. It can also locate the faulty addresses. The BIRA module performs the proposed MESP algorithm and includes two components named the *fault collection register (FCR)* and the *address remapping content-addressable memory (ARCAM)*. The FCR collects and analyzes the faulty cell information detected by the BIST module. The ARCAM is the address remapping mechanism when the memory operates in normal operation mode. After finishing the BIST session, the faulty information stored in the FCR will be shifted into the ARCAM. Fig. 10 shows the BISR procedures. In *test/repair* mode, the BIST circuit tests the spare elements first to identify fault-free spares. If a fault is detected from the spare memory, the signal is activated and the corresponding faulty flag in the ARCAM module will be set which means that the faulty spare element is unusable. After spare memory testing, the BIST circuitry will test the main memory. If a fault is detected in the main memory, the signal will be activated. The BIST will be suspended, and the BISR performs built-in redundancy analysis (BIRA). When this procedure is completed and the memory test is not finished yet, the BIRA module will send the signal to the BIST module to resume the test procedure. After the BIST session is finished, the BIRA module shifts the *faulty information (FI)* stored in the FCRs to ARCAM.When all fault information is shifted, the ARCAM serves as an address remapping mechanism in the normal operation mode. Since the BIST controller will be suspended each time when a fault is detected. The only possible loss of fault detection capability is during the instant when the BIST controller is resumed (not at-speed testing). However, the impact of fault detection quality is very low since the number of faults existed in the memory array is usually very small.

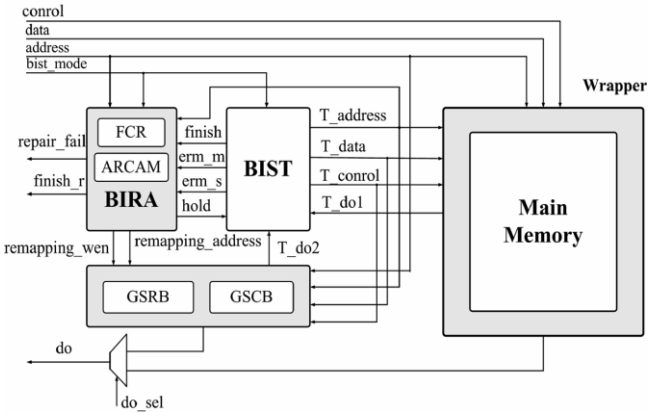

Figure .9: Block diagram of the proposed BISR scheme

The architectures of the FCR and ARCAM components are shown in Figs. 11 and 12, respectively. FCR component consists of the *FCR controller* and the *FCR*s. In test/repair mode, the reset signal is activated for one clock cycle to initialize all the internal registers. When a faulty cell in the spare elements is detected , the corresponding faulty flag in the ARCAM component will be set. After spare element testing, the ARCAM will send the signal to the FCRs to update the counters which count the numbers of available spare elements. If a faulty cell is detected in the main memory, this signal will force the FCR controller to send the hold signal to suspend the BIST circuit. Subsequently, the FCR controller will perform the MESP algorithm. The FCR controller is constructed with a finite-state machine with the state transition graph, as shown in Fig. 13. In this transition graph, if the BIST module does not detect faults or the fault has been repaired, it will remain in the state. Otherwise, it will enter the state. In the repair procedure state, if the detected fault is identified as a faulty row block and there are available
GSRBs , the BIRA module will enter the state, which means that we can use GSRBs to repair the faulty cells. In the GSRB_repair state, the output signal will be activated. Similarly, if the incoming fault is identified as a faulty column block and there are still available GSCBs , the BIRA module will enter the state which means that we can use GSCBs to repair the faulty cells. In the GSCB_repair state, the output signal will be activated. If the incoming

fault is not identified as a faulty row block or a faulty column block and there are available GSRBs or GSCBs, the BIRA module will enter the state. The signal will be activated and the incoming faulty information will be stored into the FCRs. If the GSRBs and GSCBs are all used and there still exists some faulty cells, the *repair_fail* signal will be activated. When the BIST module finishes testing the spare memory and the main memory , the BISR procedure is finished. The signal will be activated.

V. EXPERIMENTAL RESULTS

In order to evaluate the proposed fault-tolerance techniques, the simulation model proposed in [18] is used to analyze our techniques he number of such adjacent circuits is denoted as . The parameter denotes the cluster factor of the th adjacent circuit. Similarly, the number of faults that have already occurred on the th adjacent circuit is represented as . Finally, the probabilities that a defect results in a faulty cell, a faulty column, a faulty row, and a cluster fault (a combination of several cell faults where the faulty cells are a cluster of any shape) are assumed to be 70%, 15%, 10%, and5%, respectively. A simulator is implemented to simulate the proposed MESP algorithm. In our simulator, we set as a constant. The values of are set equal to for simplification. The values of and are 0.65 and 1, respectively.We define *repair rate* as the probability of successful reconfigurations. In Fig. 14, we how the repair rates for the proposed MESP algorithm with different amounts of redundancy. The memory size is 1024 1024 bits. The average number of defects injected into a chip is assumed to be 15. From this figure we can see that if more redundancies are added, we get higher repair rates. Similarly, if the memory array is divided into more row and column banks,we will also obtain higher repair rates.

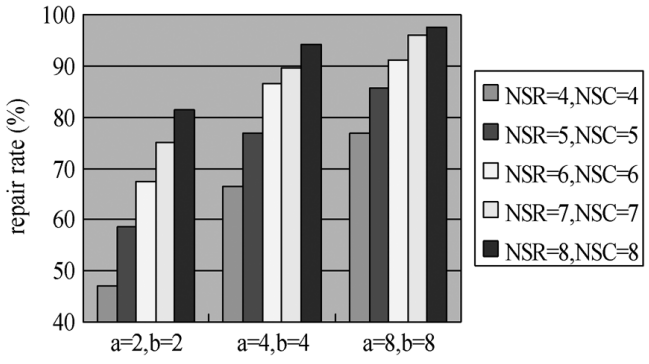

Fig. 14. Repair rates for different redundancy architectures based on the MESP algorithm.

We also compare the repair rates of the proposed MESP algorithm with those of the MESP (redundant row/column blocks are confined to their corresponding row/column banks) and ESP [15] (redundant rows/columns are not divided into row/column blocks) algorithms. The results are shown in Fig. 15. In this figure, the values of a and b are both set to 4. If the number of spare rows (columns) increases, the repair rates will also increase. From this figure we see that the MESP and MESP algorithms are much better than the ESP algorithm. It should be noted that if cluster faults are considered, the MESP algorithm will have 10% improvement over the MESP algorithm on average.This improvement is very useful to further increase the manufacturing yield.

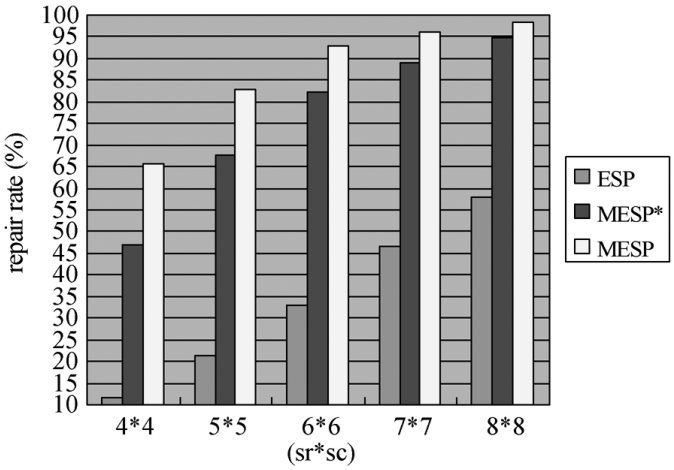

Fig. 15. Repair rates for the ESP, MESP and the MESP algorithms

If we change the values of and to 0.4 and 1.5, respectively, the results are, as shown in Fig. 16. The average number of defects injected into a chip is also the same. However, the injected faults are further clustered. Since the locations of defects are random and cluster faults may locate beyond the boundary of the chips or the divided arrays, the improvements of the MESP algorithm over the MESP algorithm are nearly the same as that shown in Fig. 15. From these figures, we can conclude that the proposed MESP algorithm is better than the other two algorithms. We have also tried to find the optimal solution by exhaustively searching the solution space. Therefore, massive simulations are conducted with different number of redundancies

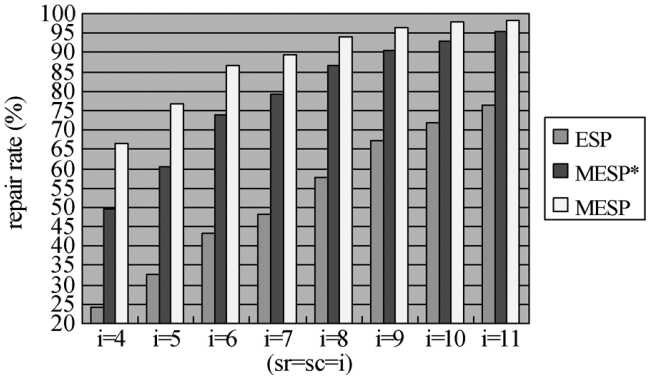

Fig. 16. Repair rates for the ESP, MESP□, and the MESP algorithms

and compared the results with the optimal solution. According to our simulation results, the average repair rate is only 0.92% lower than the optimal repair rate. Therefore, by using the proposed MESP algorithm, near-optimal repair rate can be achieved.

VI. A PRACTICAL EXAMPLE

We have implemented a 16 K 32 word-oriented SRAM chip based on the TSMC 1P6M 0.18 m process. The specifications of this design are shown in Table II. The

memory blocks in this design are generated by a commercial memory compiler. We combine four 8 K 16-bit SRAMs into one 16 K 32-bit SRAM, and integrate all categories of the redundancy such aspire words, spare rows, and spare column group blocks intone 336 32-bit spare SRAM.The chip layout of the 16 K 32-bit SRAM with BISR is shown in Fig. 17. The number of I/O pins is 118. The *core size* and *chip size* is 5.0501 and 8.551 mm , respectively. According to the area of the physical layout shown in this figure, the hardware overhead of the spare elements and the BISR module is about 8.7%.

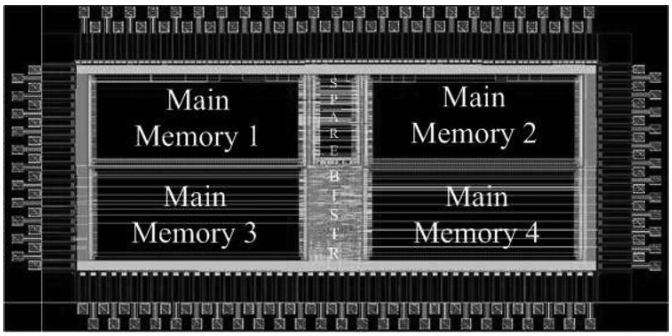

Fig. 17. Physical layout view of the 16 K x 32-bit SRAM with BISR.

VII. CONCLUSION

Instead of the traditional spare row/column redundancy architectures, *block-based* redundancy architectures are proposed in this paper. The redundant rows/columns are divided into row/column blocks. Therefore, the repair of faulty memory cells can be performed at the row/column-block level. Moreover, the redundant row/column blocks can be used to replace faulty cells anywhere in the memory array. This *global* characteristic is helpful for repairing cluster faults. The proposed redundancy architecture can be easily integrated with the embedded memory cores. Based on the proposed global redundancy architecture, a heuristic *modified essential spare pivoting* (*MESP*) algorithm suitable for built-in implementation is also proposed. According to experimental results, the area overhead for implementing the MESP algorithm is very low. Due to efficient usage of redundancy, the manufacturing yield, repair rate and reliability can be improved significantly.

REFERENCES

[1] A. Allan, "2001 technology roadmap for semiconductors," *Computer*,vol. 35, no. 1, pp. 42–53, Jan. 2002.

[2] 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 2003.[3] D. K. Bhavsar, "An algorithm for row-column self-repair of RAM's and its implementation in the Alpha 21264," in *Proc. Int. Test Conf.*, Sep. 1999, pp. 311–318.

[4] I. Kim, Y. Zorian, G. Komoriya, H. Pham, F. P. Higgins, and J.L. Lewandowski, "Built-in self-repair for embedded high density SRAM," in *Proc. Int. Test Conf.*, 1998, pp. 1112–1119.

[5] W. K. Huang, Y. H. 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.

[6] P. Mazumder and Y. S. Jih, "A new built-in self-repair approach to VLSI memory yield enhancement by using neural-type circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 12, no. 1, pp.24–36, Jan. 1993.

[7] S. Y. Kuo and W. K. Fuchs, "Efficient spare allocation in reconfigurable arrays," *IEEE Des. Test Comput.*, vol. 4, no. 1, pp. 24–31, Jan.1987.

[8] 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.

[9] C. L.Wey and F. Lombardi, "On the repair of redundant RAM's," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. CAD-6, no. 2, opp. 222–231, Mar. 1987.

[10] S. K. Lu, C. H. Hsu, Y. C. Tsai, K. H.Wang, and C. W.Wu, "Efficient built-in redundancy analysis for embedded memories with 2-D redundancy," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 1, pp. 31–42, Jan. 2006.

[11] M. Nicolaidis, N. Achouri, and S. Boutobza, "Dynamic data-bit memory built-in self-repair," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2003, pp. 588–594.

[12] M. Nicolaidis, N. Achouri, and L. Anghel, "Memory built-in self-repair for nanotechnologies," in *Proc. Int. On-Line Testing Symp.*, Jul. 2003, pp. 94–98.

[13] M. Yoshimoto, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, S.Nagao, S. Kayano, and T. Nakano, "A divided word-line structure in the static RAM and its application to a 64 K full CMOS RAM," *IEEEJ. Solid-State Circuits*, vol. SC-18, no. 5, pp. 479–485, Oct. 1983.

[14] A. Karandidar and K. K. Parhi, "Low power SRAM design using hierarchical divided bit-line approach," in *Proc. Int. Conf. Comput. Des.*, Oct. 1998, pp. 82–88.

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

[16] R. F. Huang, J. F. Li, J. C. Yeh, and C. W. Wu, "A simulator for evaluating redundancy analysis

algorithms of repairable embedded memories," in *Proc. IEEE Int. Workshop Mem. Technol., Des. Testing(MTDT)*, Jul. 2002, pp. 68–73.

[17] S. K. Lu, C. L. Yang, and H. W. Lin, "Efficient BISR techniques for word-oriented embedded memories with hierarchical redundancy," in *Proc. IEEE/ACIS Int. Conf. Comput. Inform. Sci.*, Jul. 2006, pp. 355–360.

[18] C. H. Stapper, "Simulation of spatial fault distributions for integrated circuit yield estimations," *IEEE Trans. Comput.-Aided Des.*, vol. 8, no.12, pp. 1314–1318, Dec. 1989.

T. GOVINDA RAO received the M.Tech (VLSISD) degree from Jawaharlal Nehru Technological University, Kakinada in 2011 in Electronics and Communication engineering. he is an Assistant Professor in the Department of Electronics and Communication engineering, Usha Rama College Of Engineering and Technology, Vijayawada. His current research interests include the areas of very large scale integration (VLSI) testing and fault-tolerant computing, video coding techniques, and Architectures design.

K. V. V. Satyanarayana received the M.Tech degree from Jawaharlal Nehru Technological University, Kakinada in 2008 in Electronics and Communication engineering. he isan Associate Professor in the Department of Electronics and Communication engineering, Usha Rama College Of Engineering and Technology, Vijayawada.

J. Sathish Kumar received the M.Tech degree from Jawaharlal Nehru Technological University, Kakinada in 2011 in Electronics and Communication engineering. he is an Assistant Professor in the Department of Electronics and Communication engineering, Usha Rama College Of Engineering and Technology, Vijayawada.