International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-8 Issue-2S2 December, 2018

Redundancy Algorithm Using Line Based Method for Memories achieving Less Area Overhead

V.R. Seshagiri Rao, Asha Rani. M

Abstract: ***Test cost and yield improvement is becoming important parameters with the growth of memory capacity and density. Built-in redundancy analysis (BIRA) is popularly used for embedded memories to solve yield and quality issues by removing faulty cells with available goods cells. Different BIRA approaches require different area overheads to get optimal repairs. It is difficult to get low area overhead and at the same time optimal repair rate. A new BIRA method is proposed and it uses a line-based search operation.The new BIRA reduces the storage capacity for storing faulty address data by ignoring the data of unnecessary faulty cell addresses. By using a line fail count comparison method, the proposed BIRA analyzes redundant spares quickly to get optimal repair rate. By simulation results it is verified that the proposed BIRA does the spare allocations with low area overhead.***

Keywords: ***Built-in self-repair (BISR), built-in self-test (BIST), redundancy analysis (RA), yield improvement***

I. INTRODUCTION

Due to the technological progress of semiconductor manufacturing, the capacity and density of semiconductor memories have steadily increased. This has increased the probability of occurrence memory faults, decreasing the yield causing quality degradation. Therefore, in semiconductor memory manufacturing, maintaining good quality and yield have become the important objectives. Faulty cells are replaced with spare cells to improve yield and quality.

Instead of external ATE, most SOCs adopt a built-in self-test (BIST) and built-in redundancy analysis (BIRA) for testing and repairing the embedded memories asthis method is less time consuming and more cost-effective. For memories the costs are reduced to a great extent by incorporating these functions in the chip itself. Earlier research work on BIRA techniques have concluded that this still has some more problems that have to be resolved to reduce the costs.

II. BUILT-IN REDUNDANCY ANALYSIS (BIRA)

To compare various BIRA, we define the repair rate parameter. The ability of an RA algorithm to find a appropriate repair solution is given by Repair rate and was introduced in [1]. Definitions of the repair rate and the normalized repair rate are as follows:

Repair Rate = No of repaired chips / No of Total Tested chips
Normalized Repair Rate=No of Total Tested chips/No of reparable chips

The no. of total tested chips includes the no. of irreparable chips; the repair rate is determined by no of unrepairable bad chips. The process variations, design originated faults, and human errors and some of the factors that may produce irreparable chips in chip manufacturing. These factors influence the accuracy of the developed algorithm. However Normalized repair rate is free from these variations. To develop efficient repair solutions normalized repair rate is more suitable. Optimum repair rate was defined in[1] and is used if the normalized repair is nearly 100%

The three main parameters of BIRA are 1. Repair Rate 2.Area head 3. Analysis Speed. Low value of overhead naturally reduces the cost of production. The lesser the repair rate more is the unwanted yield drop. So the repair rate should be increased to maximum extent more analyzing time is needed for High density memories than low density memories due to high processing time. As the test cost is increases with testing time, Faster RA is more advantageous.

There has been much lot of research for redundant spare allocation for reconfigurable memory arrays by various algorithms [6], [7], [15].The B&B algorithm is a simple and accurate fault-driven method. There are other methods such as intelligent solve and intelligent solve first [6] meet the requirements of low area overhead and optimal repair rate. In cases of complicated fault distributions both algorithms take a lot of time to complete the RA. Although intelligent solve first is faster than intelligent solve, its RA speed is not adequately fast enough for mass production purposes. Also there is no guarantee of an optimal solution bi Intelligent solve first.

Optimal repair solution is another important parameter of BIRA.The optimal repair solution is defined as the minimum number of spares used for a repairable memory module. The BIRA algorithm which uses least number of spares is more efficient. During the physical laser-fusing process extra cost is required to repair over allotted repair signatures. The unallocated redundancies during the wafer level repair process can be utilized to attend future field level new package level faulty cells. So arriving at the optimal repair solution is also prime objective of BIRA.

Revised Manuscript Received on December 08, 2018.
V.R.Seshagiri Rao, ECE Department, Institute of Aeronautical Engineering, Dundigal, Hyderabad, Telangana, India.
Dr. Asha Rani.M, Professor, ECE Department, JNTUH College of Engineering, Hyderabad, Telangana, India

In this paper, the analysis of various faults and categorization of various algorithms based on the time of execution is elaborated in Section 2. Definition of Optimum repair rate and it's calculation is presented in section III. A new Content addressable Memory structure for Fault Storage with illustration of example is presented in Section 4. A new algorithm namely Line Based Fault Comparison is explained with a typical fault distribution is also given. Simulation Results showing the impact on Area Overhead is given Section 5. Conclusions are given in section 6finally.

III. CLASSIFICATIONS OF REDUNDANCY ALGORITHMS

First it is proposed to classify and define different types of faults that may occur in have been planned to be repaired by specific spare rows and/or spare columns. Let Rs and Cs be the number of spare rows and columns in the memory block let K be the no of faults in the Row(Column). Then as per the value of K the faults are classified as shown below.

01234567									Sc1	Sc2	Sc3	Sc4
0				x								
1			x		x							
2						x						
3	x											
4		x										
5					x		x					
6		x										
7		x										
SR1												
SR2												

Fig 1. A Memory Block wit defects with two spare rows and Fours spare columns

Definition 1: A faulty line is said to be scattered Faulty Row (Column) if K<=Cs (Rs).

For Example Rows 1 and Row 5 and Column 4.

Definition 2: A faulty line is must repair Row(Column) if K> Cs (Rs). For Ex Column1.

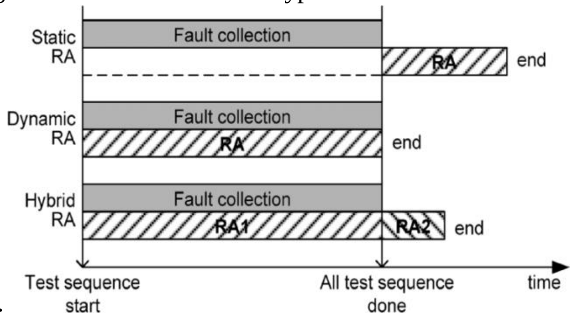
Definition 3: A faulty cell which does not have any common row or column with any

Other faulty cellis called as an orthogonal faulty cell. For ex Faults in the cells (0.3) (2.5), (3.0).

The BIST applies various test vectors through March tests to identify the faulty memory cells. The Faulty addresses are stored and analyzed by BIRA. So the important stages are,

1. Fault Collection
2. Redundancy Analysis.

The Redundancy Analysis is classified into three categories. The three different types of BIRA are shown in



ig 2.

Fig 2. Three types of RA approaches

In Static RA, the redundancy is commenced after the completion of Fault collection. This requires a large bitmap size. Also this process is slow as the redundancy analysis by BIRA can only start after the completion of Fault Collection phase. This is clearly shown in Fig 2. So this is not at test process.

In Dynamic RA, the faults are analyzed as and when they occur and the spares are also allotted concurrently.The RAis completedat the same time as the Fault collection is finished. Cresta and ESP follow this method. The disadvantage of this dynamic RA is that it is not optimum resulting in loss of repair rate. Also Cresta requires huge area overhead due toso many noparallel sub analyzers

In Hybrid type RA, spares are allotted for the high priority faults such as Must Type as per Definition 2. But the spare allocation for scattered faults and orthogonal faults are done after the completion of Fault Collection phase. So this has advantages of Dynamic and Static types. The analysis time is faster than Static type but less than Dynamic RA. But the repair rate is better than Dynamic RA. Intelligent Solve and LRM use hybrid RA.The proposed BIRA also uses hybrid RA.

IV. CALCULATION OF REPAIR RATE (OPTIMUM VALUE)

A. Criteria for Redundant Spare allocations

Obviously the objective of any BIRA is optimal repair rate. So the spare assignments is done as per the three proposals stated below.In previous papers [1], [6], [7], [12], [14] some of these proposals were also made. They are summarized as given below.

Rule1: Any spare row or column can be used to take care of orthogonal fault.

Rule2: Aspare row (column) can be used for scattered faulty row(column).Alternatively several spare columns (rows) also can be used for assignmentdepending on the availability.

Rule3: It should be noted that spare row (column) should only be compulsorily be used for the repair of must row (column).

As stated in the above rules there is a flexibility for choosing spares for orthogonal and scattered faults whereas there is no choice for must faults. So for optimum repair, the spares have to be assigned meticulously for a scattered Faults. First must repair faults are to be addressed. Then Scattered faults and finally the orthogonal faults.

For the faults storage the following guidelines are proposed.

Proposals 1: For must type of faults, it is sufficient to store the line addresses instead of all pairsof row and column cell details.

Proposal 2: For scattered faults, row and column addresses have to be stored in well define format to arrive at a optimum spare allocation during BIST operation.

Proposal 3: spare allocation for orthogonal faults should be done in the end.

In any BIRA the above proposals must be adhered to. But this is not sufficient.

In addition a good RA algorithm should be designed and should be evaluated for all permutations and combinations. The proposed BIRA is implemented in accordance with all the above three proposals with a new technique of spare assignment reducing primarily the area overhead.

As stated in the above rules there is flexibility for choosing spares for orthogonal and scattered faults whereas there is no choice for must faults. So for optimum repair, the spares have to be assigned meticulously for a scattered Faults. First must repair faults are to be addressed. Then Scattered faults and finally the orthogonal faults.

For the faults storage the following guidelines are proposed.

Proposals 1: For must type of faults, it is sufficient to store the line addresses instead of all pairs of row and column cell details.

Proposal 2: For scattered faults, row and column addresses have to be stored in well define format to arrive at a optimum spare allocation during BIST operation.

Proposal 3: spare allocation for orthogonal faults should be done in the end.

In any BIRA the above proposals must be adhered to. But this is not sufficient. In addition a good RA algorithm should be designed and should be evaluated for all permutations and combinations. The proposed BIRA is implemented in accordance with all the above three proposals with a new technique of spare assignment reducing primarily the area overhead.

V. NEW BIRA APPROACH FORMULATED

A. Basic considerations

spareRs redundant spare rows. Fig 4 shows the memory structure with BIST and BIRA blocks incorporated.

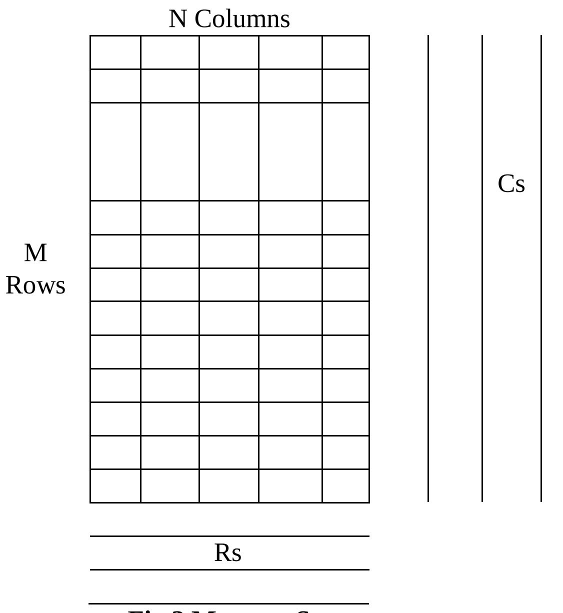


Fig 3.Memory Structure

Fig 3 shows the memory structure where Rs redundant rows and Cs redundant columns are incorporated to take care of faults which may arise in the Memory Block M X N either during manufacturing or field usage.

Fig 4 shows the Memory block with BIST and BIRA [12]blocks. The BIST block tests the memory block after Test Start signal is received. When all memory locations are tested, BIST generates Test End signal. The BIRA[5] module collects all the Faulty addresses at the end of the test does the spares allotment as per the novel algorithm. If any early terminations conditions are detected it generates irreparable signal or otherwise Repair signature with positive repair signal is asserted.

Fig 5 shows the various CAM structure formats for fault storage. In [1] CAM storage is done separately for Must repair faults. This is not done in the proposed BIRA to save the Area overhead. However this is taken care in the Parent address CAMs by inclusion of Must repair Flag. The No of Parent address CAMs is Rs+Cs, whereas the no of Child address CAMs is Rs(Cs-1)+Cs(Rs-1).The CAM structure is outlined in Fig 5 which is self-explanatory. In [3] &[7] the procedure to finalize CAM dimensions are given.

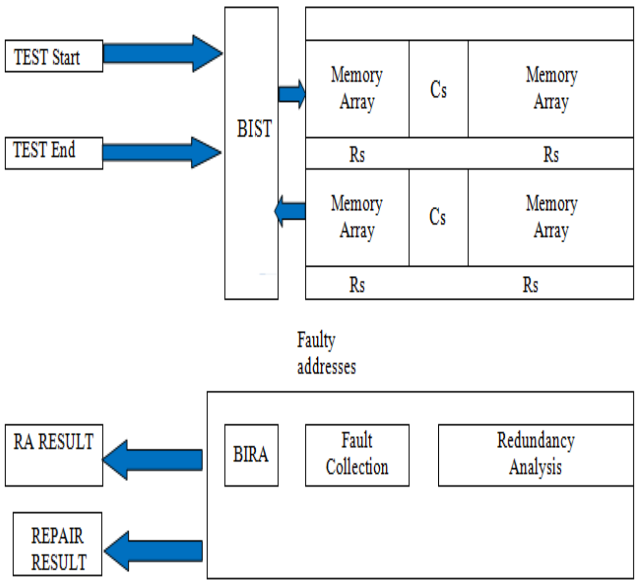


Fig 4. Structure of BIST and BIRA

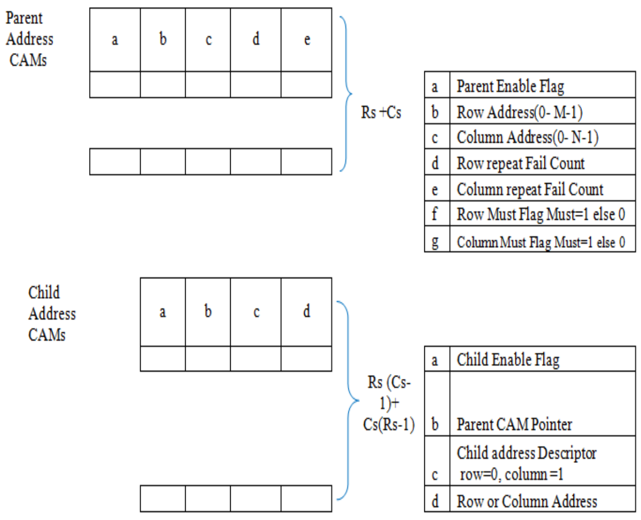


Fig 5 CAM structure for Fault Storage as per the proposed BIRA.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

B. Initial Flow chart for Fault Detection

During Fault collection phase the faults which come under the category of Must repair are identified and spare assignment is done in Fault collection stage itself. Subsequently if any faults occur in the must repair line line they are ignored as spare has already been allotted for the faulty line.The detail flow chart for fault collection is given in Fig 6 which is self- explanatory. If the fault come under the category of Parent or child faults then the availability of respective CAM is checked. If the CAM space is unavailable then irreparable signal is generated.

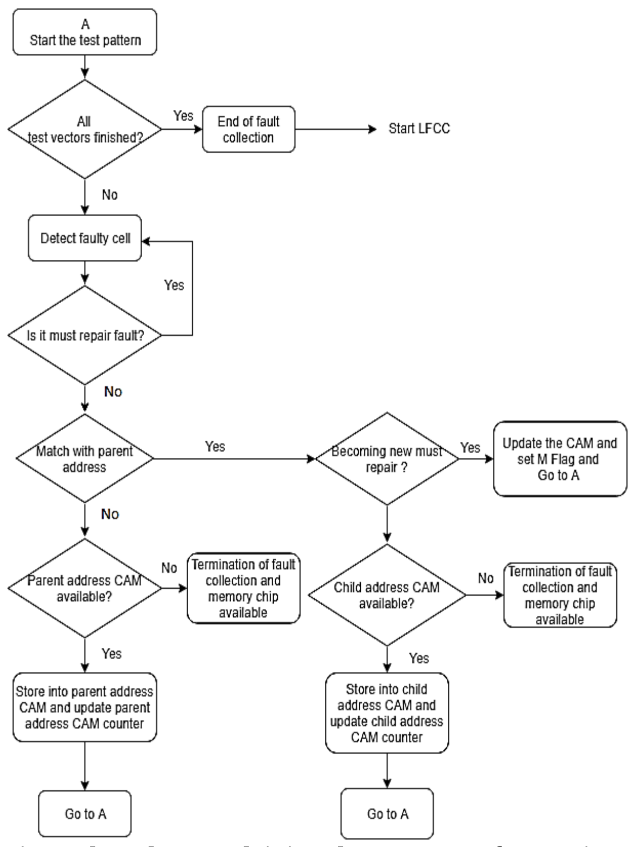

A
Start the test pattern
All test vectors finished?
Yes
End of fault collection
Start LFCC
No
Detect faulty cell
Yes
Is it must repair fault?
No
Match with parent address
Yes
Becoming new must repair ?
Yes
Update the CAM and set M Flag and Go to A
No
Parent address CAM available?
No
Termination of fault collection and memory chip available
Child address CAM available?
No
Termination of fault collection and memory chip available
Yes
Store into parent address CAM and update parent address CAM counter
Yes
Store into child address CAM and update child address CAM counter
Go to A
Go to A

Fig 6. Flow chart explaining the sequence of operations for the proposed BIRA

After the faults are collected there are two outcomes at this stage either the memory chip is not reparable or the novel BIRA namely LFCC(Line Fail Count comparison) should be used for optimum spare allocation.

C. LFCC Algorithm for allotment of spares

At the end of fault collection, the must repair faults have already been taken care as explained above. Only the Orthogonal Faults and Scattered faults are left out. Either spare row or column can be allotted for orthogonal fault as they do not have any common row or column with other faults, So they are processed after finalization of spares for scattered faults. So if minimum number of spares are used for repair of scattered faults optimal repair rate is achieved.

If the total no of orthogonal faults (Sc), Row fault lines(LRf) and column fault lines(LCf) is less than total available row(Ra) and column spares(Ca), then it is a simple case then spares allotment is done liberally without any difficulty. This is stated by the condition 3 as given below.

Condition 3: Sc + LRf + LCf<=Ca + Ra

If condition 3 is not satisfied, then the fault set of the memory structure is not a simple case, then the LFCC algorithm is applied. The faults are arranged in descending order of fail count in the scattered faulty line buffer structure. The structure format of Buffer structure is shown in Fig 7.

Buffer Enable Flag	R/C Flag Row=0 Column=1	Line No	Fault Count	Intersection Flag

Fig 7. Scattered Faulty line Buffer structure

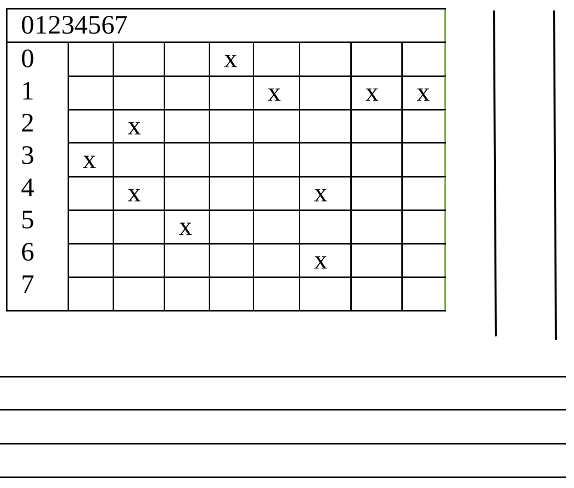
After the Scattered Faults are listed in the above Buffer the following conditions are verified. The combination which meets both the conditions 4 & 5 is the appropriate solution.

Condition 4: LFC> = TFC- (RA+CA-SP COUNT)

Condition 5:LFC> = TFC- (RA+CA-SP COUNT)-INTERSECTION FAULT

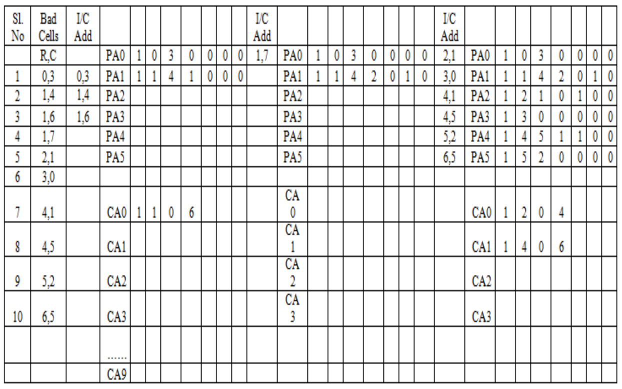
Condition 4 is a compulsory. If condition4 is met then condition 5 is also to be met. This is illustrated in Fig.9. The final spares allocation is shown in Fig 11.

The LFCC processis explained for the Example memory shown inFig 8. It has four Row spares and Two Column spares. The Fault collection sequence as per the algorithm in Fig 6 is shown in Fig 9.



01234567								
0				x				
1					x		x	x
2		x						
3	x							
4		x				x		
5			x					
6						x		
7								

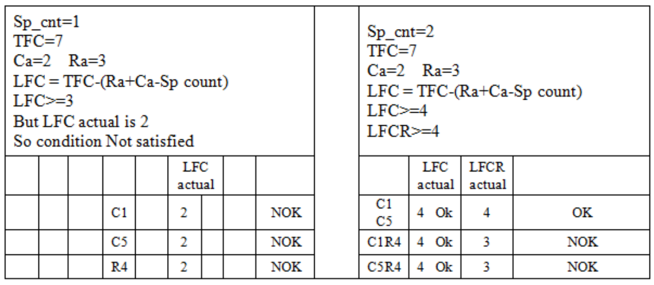
Fig 8. Memory Block 7 X 7 with four Row spares and Two Column spares



Sl. No	Bad Cells	I/C Add									I/C Add									I/C Add								
	R,C		PA0	1	0	3	0	0	0	0	1,7	PA0	1	0	3	0	0	0	0	2,1	PA0	1	0	3	0	0	0	0
1	0,3	0,3	PA1	1	1	4	1	0	0	0		PA1	1	1	4	2	0	1	0	3,0	PA1	1	1	4	2	0	1	0
2	1,4	1,4	PA2									PA2								4,1	PA2	1	2	1	0	1	0	0
3	1,6	1,6	PA3									PA3								4,5	PA3	1	3	0	0	0	0	0
4	1,7		PA4									PA4								5,2	PA4	1	4	5	1	1	0	0
5	2,1		PA5									PA5								6,5	PA5	1	5	2	0	0	0	0
6	3,0																											
7	4,1		CA0	1	1	0	6					CA 0									CA0	1	2	0	4			
8	4,5		CA1									CA 1									CA1	1	4	0	6			
9	5,2		CA2									CA 2									CA2							
10	6,5		CA3									CA 3									CA3							
			......																									
			CA9																									

Fig. 9 Fault Collection Sequence for the Memory Block shown in Fig 8

As condition 3 is not met by the example memory in Fig 8 LFCC algorithm is carried out as shown in Fig 10.



Sp_cnt=1
TFC=7
Ca=2 Ra=3
LFC = TFC-(Ra+Ca-Sp count)
LFC>=3
But LFC actual is 2
So condition Not satisfied

					LFC actual			
			C1		2			NOK
			C5		2			NOK
			R4		2			NOK

Sp_cnt=2
TFC=7
Ca=2 Ra=3
LFC = TFC-(Ra+Ca-Sp count)
LFC>=4
LFCR>=4

	LFC actual	LFCR actual	
C1 C5	4 Ok	4	OK
C1R4	4 Ok	3	NOK
C5R4	4 Ok	3	NOK

Fig 10.LFCC Analysis process

Buffer Enable Flag	R/C Flag Row=0 Column=1	Line No	Fault Count	Intersection Flag
1	1	1	2	00100
1	1	5	2	00100
1	0	4	2	00011

Fig 11. Contents of Scattered Buffer for the example Memory

01234567								
0				x				
1					x		x	x
2		x						
3	x							
4		x				x		
5			x					
6						x		
7								

12 (a) NOT OK

01234567								
0				x				
1					x		x	x
2		x						
3	x							
4		x				x		
5			x					
6						x		
7								

12 (b) OK as per C1 C5 combination

1. SIMULATION RESULTS

The faulty memory simulation using Xilinx software is shown in Fig 12.

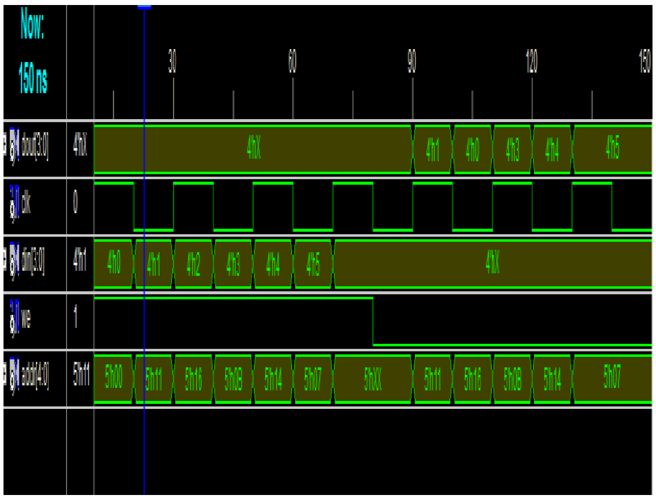


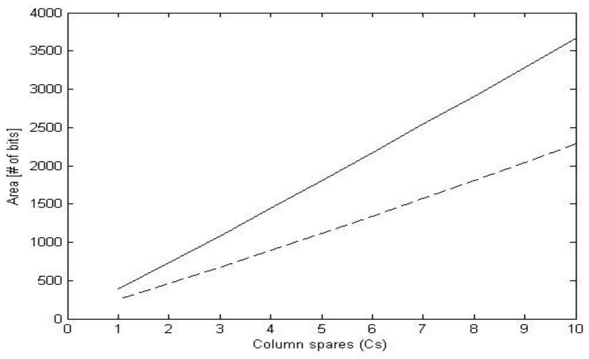
Fig12. Faulty memory simulation

Area overhead is one of the most important parameter to evaluate the BIRA algorithms. The area overhead estimate is not the same as the area of the whole BIRA. The area of storage cells is considered to calculate the areas per the below formulae.

$$A_{intelligent} = 2\ RsCs(\log 2M + \log 2N+1) + 2\ RsCs(\log 2Rs + \log 2Cs)+Aspare\_register$$

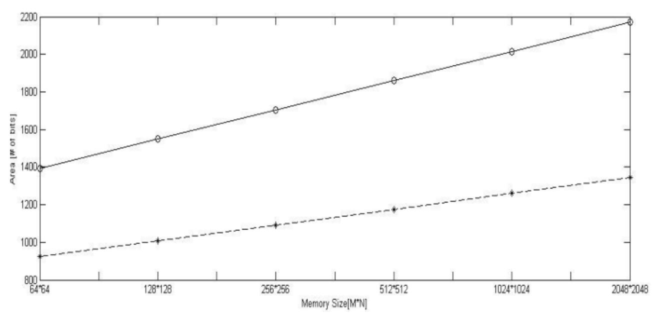
$$A_{LFCC} = RsCs\ (\log 2M+ \log 2N+1) + 2\ RsCs(\log 2Rs + \log 2Cs) +Aspare\_register$$

The graph are obtained through MATLAB simulation and shown in Fig 13 and Fig 14. It can be verified that the proposed algorithm can minimized the area overhead to nearly 2%.



__________ Intelligent Solve first::::-------------- LFCC

Fig13. Calculation of Area overhead with varying column spares with Rs=6, M=n=2048X2048



_________ Intelligent Solvefirst::::---------------- LFCC

Fig 14. Calculation of Area Overhead with varying Memory sizes with Rs= Cs =6

VI. CONCLUSION

A novel algorithm by analyzing cell faults by Line based search operation is proposed in this paper. This method results in saving of Area Overhead with respect to latest algorithms such as 'Intelligent first'. Results have been validated using MATLAB simulations. The area overhead is reduced approximately 35% compared to Intelligent first method. Also it relatively faster as extra time to save must repair faults in separate CAM as per paper [1] is avoided.

REFERENCES

1. WoosikJeong, Ilkwon Kang, KyowonJin, and Sungho Kang " A fast Built-in redundancy analysis for Memories with Optimal Repair Rate Using a Line-Based Search Tree" IEEE transactions on very large scale integration(vlsi) systems, vol.17, no 12, December 2009
2. Lee, H.,Kim., Cho, K., & Kang, S. (2018). Fast Built-In Redundancy Analysis Based on Sequential Spare Line Allocation. IEEE Transactions on Reliability
3. C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, "Built-in redundancyanalysisformemoryyieldimprovement,"IEEETrans.Reliab. ,vol.52, no. 4, pp. 386–399, Dec.2003.
4. S.-K.Lu,Y.-C.Tsai,C.-H.Hsu,K.-H.Wang,andC.-W.Wu,"Efficient built-inredundancyanalysisforembeddedmemorieswith2Dredun-dancy," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 4, no. 1, pp. 34–42, Jan.2006.
5. S. Bahl, "A sharable built-in self-repair for semiconductor memories with 2-D redundancy scheme," in Proc. 22nd IEEE Int. Symp. Defect Fault-Tolerance VLSI Syst. (DFT), Sep. 2007, pp.331–339.
6. T.-W.Tseng,J. F.Li,A.Pao, K.Chiu,and E.Chen,"Are configurable built-in self-repair scheme for multiple repairable RAMs in SOCs,"in Proc. Int. Test Conf. (ITC), Oct. 2006, pp.1–9.
7. Y.-J. Huang, D.-M. Chang, and J.-F. Li, "A built-in redundancy-anal-ysis scheme for self-repairable RAMs with two-level redundancy," in Proc. 21st IEEE Int. Symp. Defect Fault-Tolerance VLSI Syst. (DFT), Oct. 2006, pp.362–370.
8. P.Öhler,S.Hellebrand,andH.J.Wunderlich,"Anintegratedbuilt-in test and repair approach for memories with 2D redundancy," in Proc. Eur. Test Symp. (ETS), May 2007, pp.91–96.
9. H.-Y.Lin,F. M.Yeh, and S.Y.Kuo," An efficient algorithmfor spare allocation problems,"IEEETrans.Reliab.,vol.55,no.2,pp.369–378, Jun.2006.
10. S.-K. Lu, Y.-C. Tsai, and S.-C. Huang, "A BIRA algorithm for em-bedded memories with 2D redundancy," in Proc. IEEE Int. Workshop Memory Technol., Des., Test. (MTDT), Aug. 2005, pp.121–126.
11. 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. Workshop Compon.-Based Softw. Eng., Softw. Arch. Reuse (ICIS-COMSAR), Jul. 2006, pp.355–360.
12. R.-F. Huang, C.-H. Chen, and C.-W. Wu, "Economic aspect of memory built-in self repair," IEEE Des. Test Comput., vol. 24, no. 2, pp. 164–172, Mar.2007.
13. J.R.Day,"Afault-driven compreh ensivere dundancy algorithm,"IEEE Des. Test Comput., vol. 2, no. 3, pp. 35–44, Jun.1985.
14. M.Tarr,D.Boudreau,andR.Murphy,"Defectanalysissystemspeeds test and repair of redundant memories," Electronics, vol. 57, pp. 175–179, Jan.1984.
15. T.-W.Tseng,J.-F.Li,andD.-M.Chang,"Abuilt-inredundanc y-analy sisscheme for RAMswith 2Dre dundancyusing 1Dlocalbitmap,"in Proc. Des., Autom. Test Eur. (DATE), Munich, Germany, Mar. 2006, pp.53–58.