International Journal of Engineering & Technology, 7 (4.29) (2018) 235-238

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Two Cell Fault Models and Parasitic RC Test Method for Embedded SRAM

M. Parvathi*

**Professor, ECE Dept, BVRIT Hyderabad College of Engineering for Women, Telangana, India*
**Corresponding author E-mail: parvathi.m@bvrithyderabad.edu.in*

Abstract

The existing research on two cell memory faults was not adequate to identify the current technology prone defects. The gaps in the invention of test methods and fault models in related to two-cell SRAM is lead to the development of new test techniques, that are presented in this paper. The cell size reduction in present day technologies will give effect on bit line and coupling capacitance, due to capacitive nature through coupling, each cell will get influence of its neighbouring cells, prone to the faulty behaviour. In addition, parasitic node capacitance and faulty node voltage of a defective node can induce serious parasitic effects on the electrical behaviour of SRAMs. This paper is focused on analysis of characterization of two-cell fault models using bridge or short as defect model in the electrical environment and further evaluates the necessary conditions to induce worst-case coupling effects. The proposed method guarantees detecting all two-cell faults in the presence of capacitive coupling and worst-case neighbourhood data for any possible open or short defect.

Keywords: *Two cell SRAM, parasitic extraction, defect induced layout, capacitance effect*

1. Introduction

From the past literature [1-5] on two cell faults, noticed that very few studies were available on coupling faults, linked faults, and dynamic faults. The primary reason is, when more than one cell considered in the fault model, the fault number becomes either be doubled or tripled in comparison to single cell faults. Due to this fault exaggeration, more number of primitive compositions 4n to maximum 22n are required using March algorithms, that may consumes lot of test time. Some faults are linked faults [2-4], composed of two or more simple faults. The behavior of each simple fault influenced by the remaining faults, and in some cases, the fault may be masked. The problem of Traditional March tests for linked faults is masking of cell value. A large FMM (Functional Memory Model) that includes all static simple two-cell coupling faults are been defined by Hamdioui, van de Goor, and Rodgers [5] using March SS. Single-Port Faults are the faults that require at the most one port in order to sensitize a fault. Note that the single-port faults sensitize single-port as well as in multi-port memories [6]. Assume #P be defined as the number of ports required simultaneously to apply a test input on cell C. In the case of single cell, the same cell only will be effected using either read or write operation. In such case #P =1. If two simultaneous read operations applied on cell C_1 via two different ports cause that cell to flip, then #P =2. If two cells are sensitized using either write or read operation, the effect will be seen in one of the cells considering other cell condition. In this case #P=1, but cells are two. This paper is focusing on completeness in test models for single port two cell fault models including memory parasitic effects into consideration. Single port faults (1PFs) can be categorized as single cell single port faults (1PF1s) and single port 2 cell faults (1PF2s) as shown in Fig.1.

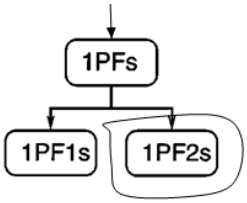

Fig. 1: Categorization of Single Port faults

The faults found in two-cell SRAM are different from the fault category that is observed in a single cell. These faults are purely coupling faults (i.e. the faults that involve more than one cell). These faults are classified based on how a cell influences the other cell in the memory array. If data is sensitized at one cell, the other cell data might change based on the fault type. Sometimes both the sensitized as well as coupled cell will be affected by a fault at a time, results in correct/incorrect/random data. Based on these possibilities, set of faults are defined in terms of two cell coupled fault models. A cell that sensitizes the fault into other cell is called an aggressor cell (a-cell) or coupling cell, and the other one is called a victim cell (v-cell) or coupled cell. The a-cell is the cell to which the sensitizing operation (or state) should be applied in order to sensitize the fault, while the v-cell is the cell where the fault appears. These faults are majority of dynamic faults and need more March sequences to identify [7-9]. In addition, the test without considering parasitic memory effect is not a complete test

[10]. Compared to single cell fault models, two cell fault models include more number of bits to be tested. If number of bits for testing increases then fault behavior becomes more complex, also the test time increases exponentially. This reflects on test cost too. These issues have become the motivational causes behind our proposed work and provided a solution to the problems addressed. Our proposed parasitic extraction method follows few steps such that, maximum possible two cell faults have been collected initial-

ly by considering bridge or short as defect model in an electrical circuit environment. As a next step, the corresponding defect layout has been extracted that pops up additional layers or missing of any node due to defect injection through circuit. These additional layers or missing node will change the overall parasitic resistance and capacitance of the cell, through this, through this step the faulty behavior of that particular cell can be analyzed. By repeating the procedure for all possible fault injection, an analysis has been carried out on the type of fault corresponding to their parasitic RC values. Hence, one can follow this as a fault behavior dictionary while designing the SRAM for two-cell architecture to predict the fault in advance. The method fault detection using parasitic extraction is shown in Fig.2. Section II briefs about two cell faults and fault model. Section III will discuss about parasitic resistance and capacitance effect on the chosen fault models.

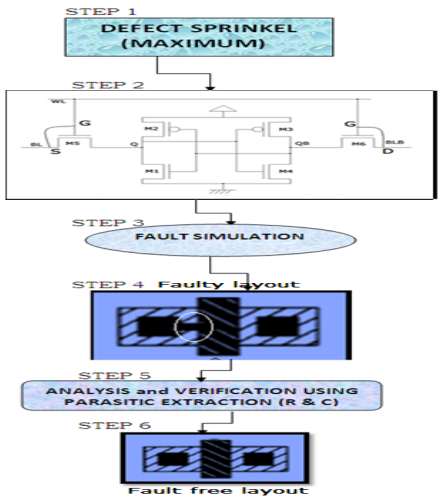

Fig. 2: Method of fault detection flow using parasitic extraction technique

Results are presented in Section IV, and conclusions are given in Section V.

2. Two-cell Faults and Condition for Fault Occurrence

2.1. Two Cell Faults

As shown in Fig.3 two 6T SRAM cells are tied together with a single port WL through which both the cells are given either read or write operation simultaneously. Each cell comprises seven nodes. Two nodes Q and QB are for identifying the cell state. Two nodes BL and BLB are for either to perform read/write with the help of WL node. Two more are supply and ground nodes V_{DD} and V_{SS} respectively. Altogether 14 nodes are considered for applying the proposed test.

In general, a two-cell FP is denoted by < Sa; Sv/F/R > (or < Sa; Sv/F/R >a;v). In this type of fault, the sensitizing cell is different from the cell in which the fault occurs.

In this notation, Sa describes the sensitizing operation or state of the aggressor cell (a-cell); while Sv describes the sensitizing operation or state of the victim cell (v-cell). Here, the set Si is defined as: Si ϵ {0; 1; 0w0; 1w1; 0w1; 1w0; r0; r1} (i ϵ {a; v}), Fϵ {0; 1; ↑; ↓; ?}, and R ϵ {0; 1; ?; -}. Table.1. shows the type of fault and the corresponding fault occurrence condition accordingly.

2.2. Fault Model for Two Cells SRAM Using Bridge/Short

From the chosen two-cell structure, each node of one cell is tied to the other cell node in order to create a bridge or short defect model. One example model is shown in Fig.4, in this 0^{th} cell is considered as 'v' cell and 1^{st} cell is considered as 'a' cell. Node Q0 of 0^{th} cell is shorted to node QB 1 of 1^{st} cell. Considering all possible node bridges, 16 fault models are identified as shown in Table.2. In each model, the operation applied on 'a' cell (i.e. cell 1) is specified and corresponding fault observation in 'v' cell (i.e. cell 0) is noted. Few fault models are found to be nonexistent yet and are Q0-QB1, Q1-Q0, QB1-QB0, Q1-BL0, BL1-BLB0, BL0-BLB1, BL0-BL1, and QB0-BLB1. These fault models were compared with existing fault models and identified the additional changes in their faulty behavior. Fault model for Undefined disturb coupling fault is new in which the return data is incorrect or undefined. Few fault models are identified with multi fault behavior by different sensitizing operations and are Q0-BL1, BL0-BLB1 and QB0-BLB1.

3. Parasitic RC Extraction from Defect Model

For the defect induced fault model developed in electrical circuit environment, the corresponding layout must be extracted in order to identify the additional parasitic components present in it. Fig. 5 & 6 are fault free and defect induced layouts respectively, shown as an example for how a fault can influence the nodes and layers on a layout. Defect induced layout using Q0-QB1 as fault model results an invisible node Q0. For all fault models under consideration the corresponding layouts are extracted and identified the fault location accordingly [11, 12]. Once fault induce layout is extracted, extraction of parasitic resistance and capacitance step should be carried out. Table 3&4 shown with various R and C extracted from layouts of all the fault models under consideration. These parasitic R and C are further compared with fault free in order to characterize the fault.

4. Results and Comparisons

The parasitic R & C values at each node are observed and are shown in Table 3 & 4. Fault detection through parasitic R, C is accomplished by comparing each faulty model parasitics with fault free model. The key observation from this procedure is wherever the fault is imposed in the circuit that corresponding node only is effected by resulting more resistance and capacitance than the fault free R, C values while remaining node parasitic RCs unchanged. The graphical representation of fault detection based on parasitic RC variation using 120nm Technology is shown in Fig.7 & 8 respectively. From the results, one can observe that, at least one node with fault affected will be highlighted with its corresponding parasitic R,C values, and the same is easily identified in the process of testing with parasitic values. This proposed method avoids the need of more test time due to huge test length involved in traditional March algorithms [5-7]. In addition, the test coverage also improved by detecting undetected faults, which is a limitation of existing algorithmic based test methods

Table 1: Faults with their cause and effects

S.No	Action on a-Cell	Effect on v-Cell	Fault Type
1	given a state	forced to given logic state	State Coupling Fault (CFst)
2	given a state	forced to undefined logic state	Undefined State Coupling Fault (CFus)
3	read/write	flip state	Disturb coupling fault CFds
4	read/write	undefined state	Undefined Disturb coupling fault (CFud)
5	read/write	flip state	Idempotent Coupling Fault (CFid)
6	transition write operation	inversion	Inversion Coupling Fault (CFin)
7	a given logic,	No transition write opera-	Transition Cou-

	value in the aggressor cell prevents	tion	pling Fault (CFtr)
8	transition in a-cell	non-transition write operation on v-cell	Write Destructive Coupling Fault (CFwd)
9	given a state	read operation on the v-cell changes the data in the v-cell and returns an incorrect value on the output	Read Destructive Coupling Fault (CFrd)

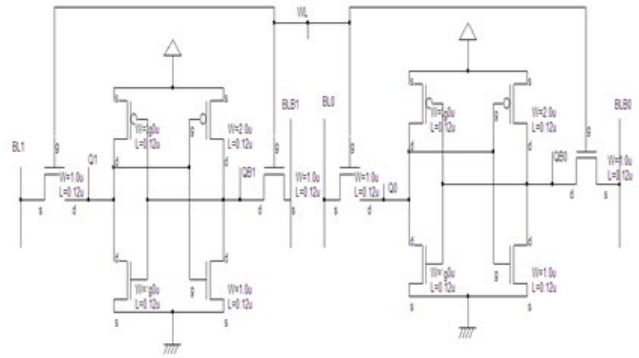

Fig. 3: Two cell SRAM model

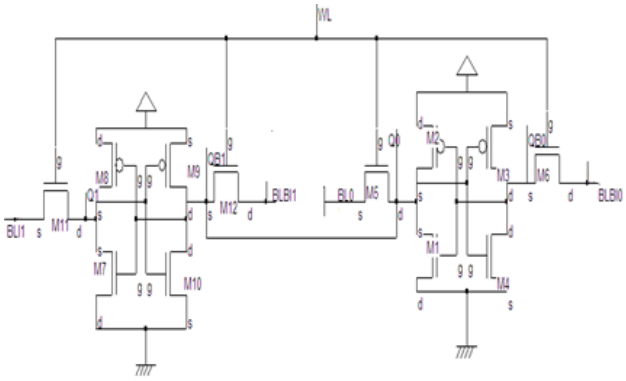

Fig. 4: Fault Model for CF_{st} using Short between QB1 and Q0

Table 2: Possible fault models using 'Bridge/short' model

S.No	DEFECT MODEL	FAULT OBSERVED
1	Q0-QB1	1. Writing '0' on 'a' cell causes 'v' cell to **undefined** state (simultaneous read operation on 'v' cell) hence mapped to **Read Destructive Coupling Fault (CFrd)** as well as both 'a' and 'v' cells are forced to toggle state.
		2. Writing '0' on 'v' cell causes 'a' cell **undefined** state and simultaneous read on 'a' cell results in toggle state.
2	Q1-QB0	1. Writing '0' on 'a' cell causes 'v' cell undefined state, Writing '0' on 'v' cell causes 'a' cell undefined state. hence mapped to **Undefined State Coupling Fault (CFus).**
3	Q1-Q0	1. Writing on 'a' cell causes 'v' cell read operation that changes the data in the v-cell and returns an **incorrect/undefined** value on the output. Hence results in **Read Destructive Coupling Fault (CFrd).**
		*Writing on 'v' cell causes 'v' cell data flips and returns an incorrect value on the output. **(Write destructive fault in 'v' cell WDF).**
4	QB1-QB0	1. Writing/reading on 'a' cell causes 'v' cell read operation returns an incorrect/undefined value on the output hence mapped to **Undefined Disturb coupling fault (CFud)**
5	Q0-BLB1	1.Transition write operation performed on the 'a' cell, causes the v-cell to flip mapped to **Idempotent Coupling Fault (CFid).**
6	Q1-BLB0	1. Write on 'v' cell causes the a-cell to flip results in **Disturb coupling fault CFds**
7	BL1-BLB0	1. **Transition write operation performed on the a-cell, causes the inversion of the 'a'-cell (WDF).**
		2. Non-transition write operation on 'V' causes transition in a-cell results in **Write Destructive Coupling Fault (CFwd)**
8	BL0-BLB1	1. A transition write operation performed on the a-cell, causes the inversion of the v-cell results in **Inversion Coupling Fault CFin.** 2. **WDF** in 'v' cell.
9	BL0-BL1	1. Given 'a' logic 1 state, 'v'-cell is forced into state '1' but read/write operation is needed. ***State Coupling Fault (CFst) on read/write**
10	BLB0-BLB1, QB0-BLB1, QB0-BL1, QB1-BL0	1. Writing/reading on 'a' cell causes 'v' cell read operation returns an incorrect/undefined value on the output results in new fault i.e. **Undefined Disturb coupling fault (CFud)**

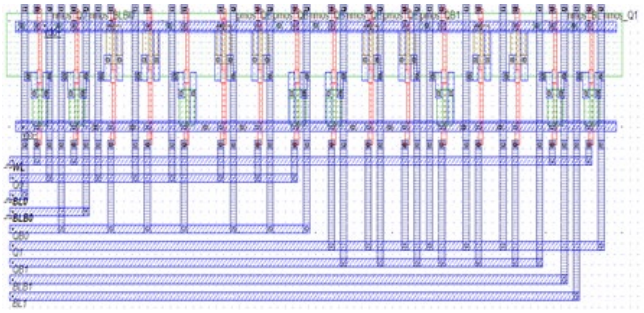

Fig. 5: Fault free layout for two cell SRAM

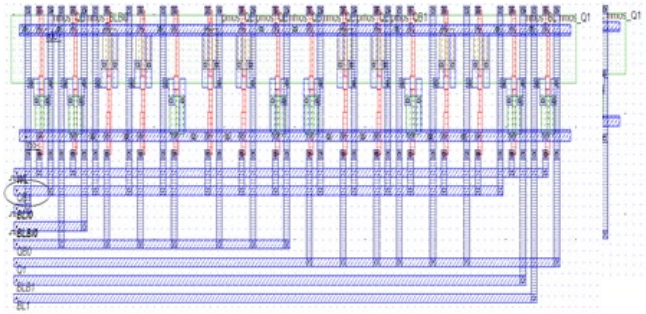

Fig. 6: Defect induced layout using Q0-QB1 as fault model

Table 3: Fault detection through parasitic Resistance

NODE	FF R,ohm	Q0-QB1 R,ohm	Q1-QB0 R,ohm	Q1-Q0 R,ohm	QB1-QB0 R,ohm	Q0-BL1 R,ohm	Q1-BL0 R,ohm	Q0-BLB1 R,ohm	Q1-BLB0 R,ohm	BL1-BLB0 R,ohm	BL0-BLB1 R,ohm	BL0-BL1 R,ohm	BLB0-BLB1 R,ohm	QB0-BLB1 R,ohm	QB1-BLB0 R,ohm	QB0-BL1 R,ohm	QB1-BL0 R,ohm
BL1	1161	1161	1161	1161	1161	7675	1161	1161	1161	3253	1161	2315	1161	1161	1161	1161	1161
Q1	6496	6479	13108	13007	6496	6469	7649	6496	8587	6496	6496	6496	6496	6496	6496	6496	6496
QB1	6496	12907	6495	6495	13108	6496	6495	6496	6495	6496	6496	6496	6496	6496	8586	6496	7648
BLB1	2098	2098	2098	2098	2098	2098	2098	8611	2098	2098	3252	2098	4190	8711	2098	8711	2098
BL0	1155	1030	1155	1155	1155	1155	NA	1155	1155	1155	NA	NA	1155	1155	1155	1155	NA
Q0	6516	1161	6516	NA	6516	NA	6516	NA	6516	6516	6516	6516	6516	6516	6516	6516	6516
QB0	6616	6720	NA	6616	NA	6616	6616	6616	6616	6616	6616	6616	6616	NA	6616	NA	6616
BLB0	2093	2093	2093	2093	2093	2093	2093	2093	NA	NA	2093	2093	NA	2093	NA	2093	2093
WL	793	792	793	793	793	793	793	793	793	793	793	793	793	793	793	793	793
TOTAL	33424	34441	33419	33418	33420	33395	33421	33421	33421	33423	33423	33423	33423	33421	33421	33421	33421

Table 4: Fault detection through parasitic Capacitance

	FF	Q0-QB1	Q1-QB0	Q1-Q0	QB1-QB0	Q0-BL1	Q1-BL0	Q0-BLB1	Q1-BLB0	BL1-BLB0	BL0-BLB1	BL0-BL1	BLB0-BLB1	QB0-BLB1	QB1-BLB0	QB0-BL1	QB1-BL0
Node	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	C,fF	
BL1	1.53	1.51	1.51	1.51	1.51	5.55	1.51	1.51	1.51	1.86	1.51	1.83	1.51	1.51	1.51	1.51	1.51
Q1	5.73	5.8	10.3	9.4	5.77	5.73	5.79	5.73	5.95	5.73	5.73	5.73	5.73	5.73	5.77	5.73	5.77
QB1	5.81	9.35	5.66	5.66	10.2	5.81	5.66	5.81	5.66	5.81	5.81	5.81	5.81	5.81	5.86	5.81	5.69
BLB1	1.48	1.46	1.46	1.46	1.46	1.48	1.46	5.53	1.46	1.48	1.8	1.48	1.84	6.06	1.46	6.06	1.46
BL0	0.58	0.55	0.58	0.58	0.58	0.58	NA	0.58	0.58	0.58	NA	NA	0.58	0.58	0.58	0.58	NA
Q0	4.69	1.51	4.69	NA	4.69	NA	4.69	NA	4.69	4.69	4.69	4.69	4.69	4.69	4.69	4.69	4.69
QB0	5.17	5.21	NA	5.17	NA	5.17	5.17	5.17	5.17	5.17	5.17	5.17	5.17	NA	5.17	NA	5.17
BLB0	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	NA	NA	0.67	0.67	NA	0.67	NA	0.67	0.67
WL	3.65	3.61	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65	3.65

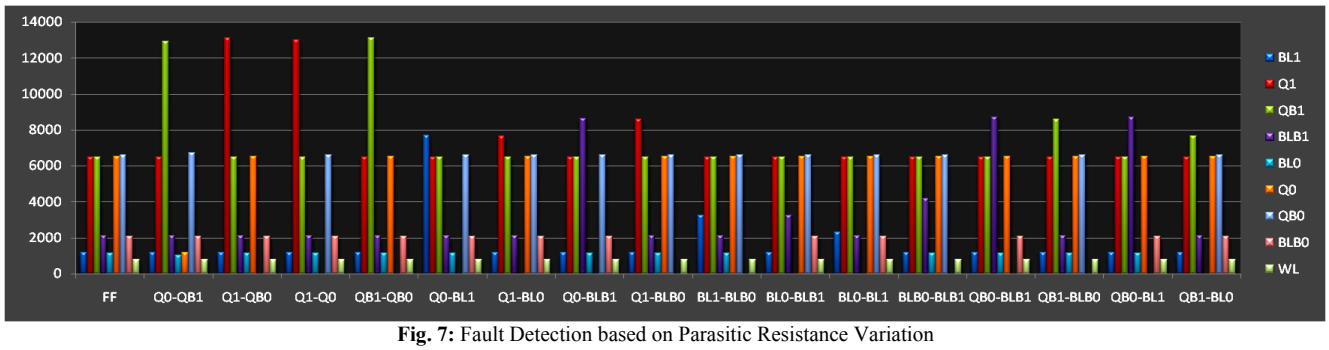

Fig. 7: Fault Detection based on Parasitic Resistance Variation

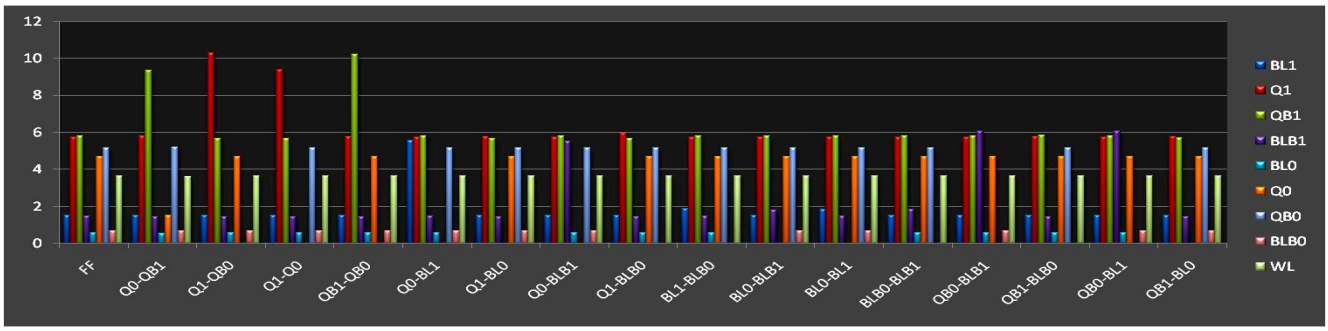

Fig. 8: Fault Detection based on parasitic capacitance Variation

5. Conclusion

Our work draws the attention to problems in the process variation as the technology advances from submicron to deep submicron level, emphasizes on the defect and fault mapping using circuit analogy. The proposed method overcomes the drawback of large test time used more number of test primitives [5-7], also results in full fault coverage with fault location. The test method proposed for two-cell fault model identifies new features in the behavior of faults such that undetected and undefined faults can also be modeled. Any undetected faults escape from traditional algorithm based tests can be found using the variations in R and C values from the fault induced layouts.

References

[1] Said Hamdioui, "Dynamic Faults in Random-Access-Memories: Concept, Fault Models and Tests", Journal of Electronic Testing: Theory and Applications, Kluwer Academic Publishers, 2003,pp: 195–205.

[2] Said Hamdioui, Ad J. van de Goor, Mike Rodgers, "Linked Faults in Random Access Memories: Concept, Fault Models, Test Algorithms, and Industrial Results", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 23, No. 5, May 2004, 0278-0070/04, 2004 IEEE, pp:737-756.

[3] A. Benso, A. Bosio, S. Di Carlo, G. Di Natale, P. Prinetto, "Automatic March Tests Generations for Static Linked Faults in SRAMs", 3-9810801-0-6/2006 EDAA.

[4] Bosio A., Di Carlo S., Di Natale G., Prinetto P., "March AB, a State-of-the-Art March Test for Realistic Static Linked Faults and Dynamic Faults in SRAMs", IET Computers & Digital Techniques Vol. 1, No. 3, 2007, pp. 237-245.

[5] Petru Caşcaval, Radu Silion, Doina Caşcaval and Cristina Huzum, "A fault primitive based model of all static four-cell coupling faults in Random-Access Memories", Universitatea Tehnica, Gheorghe Asachi din Iaşi Tomul LIV (LVIII), Fasc. 1, 2008, pp:55-65.

[6] Said Hamdioui, and Ad J. van de Goor, "Efficient Tests for Realistic Faultsin Dual-Port SRAMs", IEEE Transactions on Computers, Vol. 51, No. 5, May 2002, 0018-9340/02/2002 IEEE, pp:460-473.

[7] Pradeep Nagaraj, Shambhu Upadhyaya, Kamran Zarrineh, Dean Adams, "Defect Analysis and a New Fault Model for Multi-Port SRAMs", Proceedings of the 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'01), 1063-6722/2001 IEEE.

[8] Said Hamdioui, Ad J. van de Goor, David Eastwick, Mike Rodgers, "Detecting Unique Faults in Multi-Port SRAMs", Proceedings of the 10th Asian Test Symposium (ATS.01), 1081-7735/2001 IEEE.

[9] Said Hamdioui, and Ad J. van de Goor, "Thorough Testing of Any Multiport Memory With Linear Tests", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, No. 2, February 2002, 0278–0070/2002 IEEE, 0278-0070(02)01047-3., pp:217-231.

[10] Said Hamdioui, Ad J. van de Goor, Mike Rodgers, "Linked Faults in Random Access Memories: Concept, Fault Models, Test Algorithms, and Industrial Results", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 23, No. 5, May 2004, 0278-0070/04, 2004 IEEE, pp:737-756.

[11] M. Parvathi, K. Satya Prasad, N. Vasantha, "Testing of Embedded SRAMs using Parasitic Extraction Method", Robotic, Vision, Signal Processing and Power Applications (ROVISP), Empowering Research and Innovation, Editors: Ibrahim, H., Iqbal, S., Teoh, S.S., Mustafa, M.T. (Eds.). ISBN 978-981-10-1721-6 ISSN: 1876-1100, Feb-2016, Springer LNEE.

[12] M. Parvathi, "New March Elements for faults due to Open Defects in eSRAM", International Conference on Digital Content and Applications (ICDCA-STEM Conference-Jan 2018), Sydney, Australia.