# An Efficient March Test Algorithm for Detection of Resistive Shorts in Multi-Port SRAMs

Gurgen Harutyunyan
Virage Logic
Yerevan, Armenia
gurgen.harutyunyan@viragelogic.com

Valery A. Vardanian
Virage Logic
Yerevan, Armenia
valery.vardanian@viragelogic.com

## ABSTRACT

The paper describes the resistive shorts of multi-port SRAMs. New test operations and a new March test algorithm are proposed. The paper shows how to use new test operations and two background patterns to generate an efficient March test algorithm for detection of resistive shorts in multi-port SRAMs.

## Keywords

Multi-port SRAM, March test algorithm, resistive shorts.

## 1. INTRODUCTION

The advent of Deep Submicron technologies requires from the test industry to model physical defects manifesting themselves as shorts or opens, as accurately as possible. The main approach of Defect-based testing [1] is to apply accurate models of physical defects targeting improvement of the quality of the product. Since the well-known stuck-at fault model does not cover all possible defects, it is important to consider new defects trying to improve the quality of the test algorithm.

In [2]-[3], [5], resistive open defects are considered. In recent technologies (see [3]), the significance of resistive-open defects has considerably increased In particular, in [4], Intel reports show that vias are the most common root cause of test escapes in submicron technologies. In [3], resistive defects have been injected in the core-cell of the Infineon 0.13μm synchronous embedded-SRAM family. For each defect location, electrical simulations have been performed with many parameters, e.g. defect size, supply voltage, operating temperature and process comer.

However, resistive short defects are also important and their importance grows continuously [7]-[8]. Earlier resistive short defects were modeled mainly as bridging faults [9]-[10]. However, those bridging faults represent very simple fault models without taking into account the resistance of the defect. Note that resistance of the defect may be considered as a random entity in the whole range of its variance. In this paper, we consider the model in a higher degree of accuracy with all possible values of the resistance. It is known as Analogue Detectability Interval (ADI) [7] and may be a union of disjoint intervals of resistance. March algorithms are widely used to detect resistive opens and shorts (see [2]-[6]).

In this paper, we have shown the importance of resistive short defects only and propose a new test algorithm for detection of resistive short defects for multi-port bit-oriented, as well as word-oriented, SRAM memories. The next major contribution of this paper is the usage of background patterns, i.e. the patterns that are applied in the March algorithm. We have shown that application of background patterns Row stripe and Column stripe and their inverse, is enough to detect all those resistive short defects that are considered in this paper.

It should be noted that for the usage of the correct background pattern we need to know the correct scrambling [11] of the memory. Briefly saying, scrambling is the logical to physical mapping of the memory and, thus, for setting the correct background pattern, the logical one, in general, may not match with the physical one. To know the scrambling of the memory, we have to analyze the structure of the memory by taking into account all types of scrambling (structural primitives).

The third major contribution of this paper is introduction of new concurrent operations applied in the March algorithm. We have shown that there are specific resistive short defects that require the existence of special concurrent operations in the test algorithm. Thus, the common Write and Read operations do not allow us detect all possible resistive short defects and we have to develop new operations applied concurrently.

In Section 2, we give the main definitions and notations used in the paper.

In Section 3, we introduce the new operations to be used in the March algorithm and describe the class of resistive short defects considered in this paper. Note that we will not consider the class of well-known bridging faults (that can be considered as a model of the class of shorts with small or negligible resistance), as well as the shorts between a node and the VDD or GND, and will mainly concentrate on shorts between different bit-lines / word-lines with sufficiently higher resistance. However, note that we cover all values of the magnitude of the resistance. In section 4, we describe the new March algorithm proposed for detection of resistive shorts. Finally, in Section 5, we give the main conclusions.

## 2. DEFINITIONS AND NOTATIONS

Three types of access ports are considered for multi-port memories:

- Read-only port – Only operation Read can be applied through this port.
- Write-only port – Only operation Write can be applied through this port.
- Read/Write (dual) port – Both operations Read and Write can be applied through this port.

A March test algorithm *M* is a test algorithm with a finite number of March elements $M = \{M_1, M_2, ..., M_k\}$ where each March element $M_i$ consists of an addressing direction and a finite number of test operations [9]. The test operations use values 0 or/and 1 for bit-oriented memories and physical background patterns for word-oriented memories.

The addressing direction can be ascending (⇑), descending (⇓) or arbitrary (⇔) address order. However, in this paper, we used ascending and descending address orders only.

In Fig.1 we can see the general view of a resistive short between two bit-lines or word-lines or a bit-line and a word-line in the memory. R denotes the magnitude of resistance: $0 < R < \infty$. Depending on the actual value of R we have either a bridging defect (for smaller or negligible values of resistance R) that is symmetric with respect to the values on A and B, or a resistive short (for higher values of resistance R) when A and B may have different values. For other values of resistance, one should do simulations to find out the actual values of A and B. However, we cover all possible cases for the value of R. The proposed March algorithm detects all resistive shorts provided the corresponding operations are used in the March algorithm and the corresponding background patterns are applied with the March algorithm. Also we assume that the scrambling information for the memory is used for setting the correct background pattern.

Only traditional Read and Write operations are not enough for detection of resistive shorts considered in this paper. There are defects that require specific operations for their sensitization and detection. If only one operation can sensitize a given defect then any generated March test that does not use that operation will not detect the defect. Thus, it is very important to develop the accurate and full set of test operations for March test generation to detect the given set of defects.

The next important factor that affects the defect coverage for a given March test is the set of physical background patterns and their correct usage in test operations. The same March test algorithm using different background patterns can detect or not detect the given defect. On the basis of our considerations, we came to the conclusion that only Row stripe (RS), Row stripe bar (~RS), Column stripe (CS) and Column stripe bar (~CS) physical background patterns were enough for detection of resistive shorts in multi-port SRAMs.

The following background patterns are used if the physical content of the memory contains:

Row stripe (RS) – all 0s in the first row, all 1s in the second row, all 0s in the third row, etc, i.e. all 0s in rows of even (or odd) number and all 1s in rows of odd (or, respectively, even) number;

Row stripe bar (~RS) – all 1s in the first row, all 0s in the second row, all 1s in the third row, etc;

Column stripe (CS) – all 0s in the first column, all 1s in the second column, all 0s in the third column, etc, all 0s in columns of even (or odd) number and all 1s in columns of odd (or, respectively, even) number;

Column stripe bar (~CS) – all 1s in the first column, all 0s in the second column, all 1s in the third column, etc.

Finally, the scramble information of the memory is very important for defect detection. If the scrambling information is unknown, then it is impossible to have an accurate physical background pattern. There are resistive shorts that require the certain test operation with the certain background pattern in order to be sensitized. Thus, if accurate scramble information is missing, then the expected defect coverage will be lost.

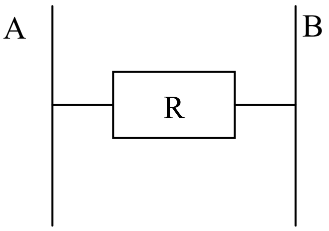

**Figure 1. A resistive short between lines A and B**

## 3. TEST OPERATIONS AND RESISTIVE SHORTS

Figures 2 shows an example of resistive shorts in a 3-port SRAM, where 2 ports are Read-only and the third one is Write-only. In this figure, the cell-level representation of memory array structure is given.

Resistive short F1 (see Fig. 2) occurred between "true" bit-lines of two different Read-only ports of two cells that belong to different columns (BL-A1 and BL-B2). Let us assume that cell C11 has value 0 and cell C22 has value 1. Operation Read through port A from cell C11 will set value 0 on BL-A1. A simultaneous operation Read through port B from cell C21 will activate WL-B2 and the corresponding access transistors of cells C21 and C22 will be open. We have the following situation in this case:

- The access transistors of the port B in the cell C22 are open;
- BL-A1 and BL-B2 are connected;
- BL-A1 and BL-B2 have opposite values (the value of BL-B2 is 1 since the value of cell C22 is 1).

In this case, the value of BL-A1 or BL-B2 will change. This will lead to flipping in cell C11 or C22. It is necessary to have opposite values in C11 and C22 in order to sensitize this defect. For example, background pattern Column stripe can be used. After the sensitization of the resistive short, the latter can be detected by a Read operation that immediately follows the sensitization operation.

### Test operations:

The following test operations are used for testing resistive shorts in multi-port SRAMs:

**W -** Operation Write is applied through one of Write-only or Read/Write port.

**R -** Operation Read is applied through one of Read-only or Read/Write port.

**RSH -** Operation Read is applied through one of Read-only or Read/Write port (active port) and simultaneously Read operations are applied through the rest of Read-only and Read/Write ports (inactive ports). All operations are applied at the same address.

**RSHR -** Operation Read is applied through one of Read-only or Read/Write port (active port) and simultaneously Read/Write operations are applied through the rest of inactive ports (operations Read from Read-only ports, operations Write from Write-only and Read/Write ports). All operations applied to inactive ports are addressed to the memory cell with inverted

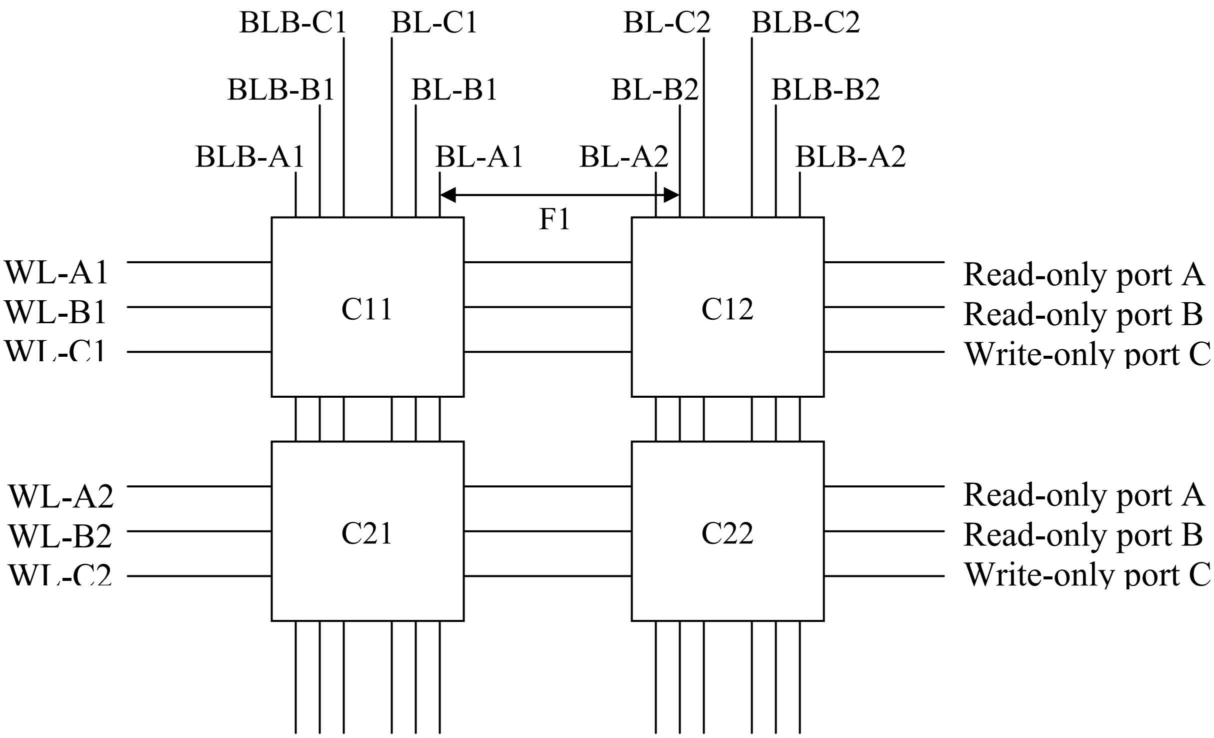

**Figure 2. Resistive short F1**

column bit COL(0). All operations use the same background pattern. The results of Read operations from inactive ports are ignored. In case CM=1, operation Read should be applied through active port only.

**RSHC -** Operation Read is applied through one of Read-only or Read/Write port (active port) and simultaneously Read/Write operations are applied through the rest of inactive ports (operations Read from Read-only ports, operations Write from Write-only and Read/Write ports). All operations applied to inactive ports are addressed to the memory cell with inverted row bit ROW(0). All operations use the same background pattern. The results of Read operations from inactive ports are ignored.

**WSHC -** Operation Write is applied through one of Write-only or Read/Write port (active port) and simultaneously Read/Write operations are applied through the rest of inactive ports (operations Read from Read-only ports, operations Write from Write-only and Read/Write ports). All operations applied to inactive ports are addressed to the memory cell with inverted row bit ROW(0). All operations use the same background pattern. The results of Read operations from inactive ports are ignored.

## Resistive shorts:

Table 1 shows all possible resistive shorts considered in this paper and the conditions for their sensitization.

We split all possible resistive shorts in 5 groups depending on between which lines they occurred (listed in the first column). Resistive shorts can occur between:

- two bit-lines in the same column;
- two bit-lines of different columns;
- two word-lines in the same row;
- two word-lines of different rows;
- a bit-line and a word-line.

The resistive shorts are listed in the second column. The third and the forth columns show correspondingly the required test operation and background pattern for sensitization of the resistive short. “All” listed in the fourth column means that all four RS, ~RS, CS and ~CS background patterns are needed for sensitization of the resistive short. Although only one resistive short is described in each row in Table 1, the short can bring to different impacts on the memory cell depending on the value of resistance. Thus, for each type of resistive short more than one condition for sensitization is described.

## 4. THE PROPOSED MARCH TEST ALGORITHM

Table 2 describes the proposed March test algorithm for detection of resistive shorts in multi-port SRAMs. “MP” stands for “multi-port”. The complexity of March MP is 28Np, where N is the number of memory words (bits) in case of word-oriented memories (bit-oriented memories) and p is the number of memory access ports. The March test is applied through each port. It uses Row stripe, Row stripe bar , Column stripe and Column stripe bar background patterns.

**Table 1. Resistive shorts and their sensitization**

| Groups | Resistive shorts | Test operations | Background patterns |
|---|---|---|---|
| Two bit-lines in the same column | Two bit-lines of the same Read port | R | (RS and ~RS) or (CS and ~CS) |
| | Two bit-lines of two different Read ports | RSHC | All |
| | Two bit-lines of the same Write port | W | (RS and ~RS) or (CS and ~CS) |
| | Two bit-lines of two different Write ports | WSHC | All |
| | Two bit-lines of Read and Write ports | RSHC | All |
| Two bit-lines of different columns | Two bit-lines of the same Read port | RSH | All |
| | Two bit-lines of two different Read ports | RSHC | RS and ~RS |
| | Two bit-lines of the same Write port | W | All |
| | Two bit-lines of two different Write ports | WSHC | RS and ~RS |
| | Two bit-lines of Read and Write ports | RSHC | RS and ~RS |
| Two word-lines in the same row | Two word-lines of two different Read ports | RSHC | RS and ~RS |
| | Two word-lines of two different Write ports | WSHC | RS and ~RS |
| | Two word-lines of Read and Write ports | RSHC | RS and ~RS |
| Two word-lines of different rows | Two word-lines of the same Read port | R | RS and ~RS |
| | Two word-lines of two different Read ports | RSH | RS and ~RS |
| | Two word-lines of the same Write port | W | RS and ~RS |
| | Two word-lines of two different Write ports | RSHC | RS and ~RS |
| | Two word-lines of Read and Write ports | RSHR | RS and ~RS |
| Bit-line and word-line | Bit-line and word-line of the same Read port | R | (RS and ~RS) or (CS and ~CS) |
| | Bit-line and word-line of two different Read ports | RSH | (RS and ~RS) or (CS and ~CS) |
| | Bit-line and word-line of the same Write port | W | (RS and ~RS) or (CS and ~CS) |
| | Bit-line and word-line of two different Write ports | WSHC | (RS and ~RS) or (CS and ~CS) |
| | Bit-line and word-line of Read and Write ports | RSHC, W | RS and ~RS |

**Table 2. March MP: applied through each port.**

⇑(W(*CS*));
⇑(RSHC(*CS*), RSH(*CS*), WSHC(*CS*), RSH(*CS*));
⇑(RSH(*CS*));
⇑(W(~*CS*));
⇓(RSHC(~*CS*), RSH(~*CS*), WSHC(~*CS*), RSH(~*CS*));
⇑(RSH(~*CS*));
⇑(W(*RS*));
⇑(RSHC(*RS*), RSH(*RS*), RSHR(*RS*),
RSH(*RS*), WSHC(*RS*), RSH(*RS*));
⇑(RSH(*RS*));
⇑(W(~*RS*));
⇓(RSHC(~*RS*), RSH(~*RS*), RSHR(~*RS*),
RSH(~*RS*), WSHC(~*RS*), RSH(~*RS*));
⇑(RSH(~*RS*)).

## 5. CONCLUSIONS

A wide class of resistive shorts in multi-port SRAMs is described in this paper. The traditional Read and Write operations were not enough to detect such defects. A new set of test operations are developed for sensitization and detection of such defects. The corresponding background patterns are also defined for sensitization of such defects. Based on these, an efficient March test algorithm March MP is proposed for detection of the considered class of resistive shorts in multi-port SRAMs.

## 6. REFERENCES

[1] W. Maly, “Realistic fault modeling for VLSI testing”, In Design Automation Conference, 1987, pp. 173-180.

[2] S. Borri, M. Hage-Hassan, P. Girard, S. Pravossoudovitch and A. Virazel, "Defect-Oriented Dynamic Fault Models for Embedded-SRAMs", Proc. IEEE ETW, 2003, pp. 23-28.

[3] L. Dilillo, P; Girard, S. Pravossoudovitch, A. Virazel, S. Borri and M. Hage-Hassan, "Resistive-Open Defects in Embedded SRAM Core-Cells: Analysis and March Test Solution", Proc. of IEEE ATS, 2004, pp. 266-271.

[4] W. Needham et al, “High Volume Microprocessor Test Escapes - An Analysis of Defects Our Tests are Missing”, Proc. Int. Test Conference, 1998, pp. 25-34.

[5] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, M. Bastian. “Resistive-Open Defect Injection in SRAM Core-Cell: Analysis and Comparison between 0.13 µm and 90 nm Technologies” Proc. Design Aut. Conf., 2005, pp. 857-862.

[6] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, M. Bastian., “Resistive-Open Defect Influence in SRAM Pre-Charge Circuit: Characterization and Analysis”, Proc. European Test Symposium, 2005, pp. 116-121.

[7] P. Engelke, I. Polian, J. Schloeffel, B. Becker, “Resistive Bridging Fault Simulation of Industrial Circuits”, In Proc. of Design Automation and Test in Europe, 2008, pp. 628-633.

[8] R. Rodriguez-Montanes, E. M. J. G. Bruls, and J. Figueras, “Bridging defects resistance measurements in a CMOS process”, In International Test Conference, 1992, pp. 892-899.

[9] A.J. van de Goor, *Testing semiconductor memories: Theory and Practice*, John Wiley & Sons, Chichester, England, 1991.

[10] S. Hamdioui and A.J. Van de Goor, "An Experimental Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests", Proc. of IEEE ATS, 2000, pp. 131-138.

[11] A.J. van de Goor, I. Schanstra, “Address and Data scrambling: Causes and Impact on Memory Tests”, Proc. of DELTA 2002, pp. 128-136.