BIST LOGIC DESIGN FOR A REDUCED MODEL OF 3–COUPLING FAULTS IN RANDOM–ACCESS MEMORIES

Caşcaval Petru

Department of Computer Science and Engineering, "Gh.Asachi" Technical University of Iaşi, Bd. D. Mangeron, 53A, 700050, Iaşi, Romania, e-mail: cascaval@cs.tuiasi.ro.

Abstract – A logic design for a built-in self-testing implementation of a march test able to cover a reduced model of 3–coupling faults in $n \times 1$ random–access memories (RAMs) is discussed. The logic design is focused on the march test *MT-R3CF* with $30n$ operations given by Caşcaval, Bennett, and Huţanu in [1]. To reduce the length of the test, only the coupling faults between physically adjacent memory cells have been considered. To compare marh test *MT-R3CF* with other published tests, simulation results are also presented in this paper.

Keywords: RAM testing, reduced 3–coupling faults, march test, built–in self–testing.

INTRODUCTION

Built–in self–testing (BIST) is a design for testability technique that places test functions physically on chip with the circuit under test. System designers use BIST for periodic testing that guarantees the detection of all target faults within a fixed time. Two kinds of testing methods exist: deterministic testing and random testing. This paper is focused on the deterministic testing of random–access memories. Taking into consideration the number of simultaneously tested arrays and the number of simultaneously accessed bits within an array, four test architectures exit: single–array single bit (SASB), single–array multiple bit, multiple–array single bit (MASB), and multiple–array multiple bit. In this work only SASB and MASB test architectures have been considered. SASB test architectures are those in which a single array of the RAM chip is tested at a time and a single bit of the tested array is accessed at a time. MASB test architectures can be used if a memory chip is organised as a number of independent arrays, allowing multiple arrays to be tested simultaneously.

Ensuring that fault coverage is sufficiently high and the number of tests is sufficiently low are the main problems with a BIST implementation. In this paper we focus on the model of 3–coupling faults. Many test algorithms have been devised to detect 3–coupling faults, see for example [4] and [6]. But, for the memory chips currently available, these tests take a long time to perform. For example, to test a 64 Mb memory chip assuming a cycle time of 60 ns, the test *S3CTEST* with the

length of $5n\log_2 n+22.5n$ given by Cockburn in [4] takes about 10 min 54s. To reduce the length of the test, the realistic coupling faults that may affect only the physically adjacent memory cells has been considered. The model of 3–coupled faults which comprises only physically adjacent cells is called reduced 3–coupling. The march test *MT–R3CF* with $30n$ operations given by Caşcaval, Bennett, and Huţanu in [1] is able to cover this model of reduced 3–coupling faults.

MARCH TEST *MT–R3CF*

The most widely used test algorithms for RAM–BIST are the march tests. As defined in [7], a march test consists of a sequence of m march elements, $<M^{(0)}; M^{(1)}; ...; M^{(m-1)}>$, where a march element (M) consists of a sequence of operations applied to each cell in the memory before proceeding to the next cell. The whole memory is checked homogeneously in either ascending ($\Uparrow$) or descending address order ($\Downarrow$). As defined in [1],

$$MT\text{–}R3CF = < \Uparrow (w_0)^{(0)}; \Uparrow(rw_1)^{(1)}; \Uparrow(rw_0)^{(2)}; \Downarrow(rw_1)^{(3)}; \Downarrow(rw_0)^{(4)}; I_1{}^{(5)}; \Uparrow(rw_c rw_c)^{(6)};$$
$$I_2{}^{(7)}; \Uparrow(rw_c rw_c)^{(8)}; I_3{}^{(9)}; \Uparrow(rw_c rw_c)^{(10)}; I_4{}^{(11)}; \Uparrow(rw_c rw_c)^{(12)}; \Uparrow (r)^{(13)} >, \qquad (1)$$

where I_1 , I_2 , I_3 , and I_4 are test sequences which initialise the memory as follows: I_1 initialises the odd columns with 0 and the even columns with 1, and I_3 vice versa (column–stripe data background); I_2 and I_4 initialise the memory with a checkerboard data background and its complement (Fig. 1).

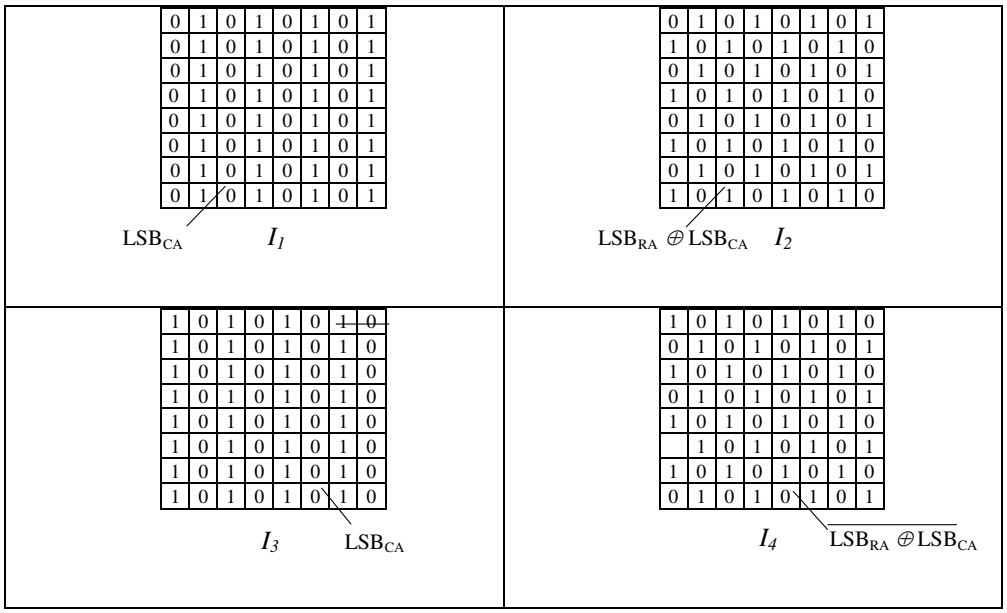

Fig. 1 –Data background used by *MT–R3CF*.

This march test contains fourteen sequences as identified with a superscript (x), where $x \in \{0, 1,\ldots,13\}$. Note that when changing from one background to the next, only the cells that

must change states are written. Also, each write operation is preceded by a read operation. Fig. 1 shows that any background change affects only a half of the cells, so that each sequence I_1, I_2, I_3 or I_4 performs $\frac{n}{2}$ read operations and $\frac{n}{2}$ writes operations.

To compare this test with other march tests, simulation results are presented in Table 1. The following march tests have been considered in the simulation study: algorithm *March C–* given by van de Goor [7]; *March LR* given by Yarmolik, van de Goor, Gaydadjiev and Mikitjuk [8]; algorithm A given by Nair, Thatte, and Abraham [6] (*NTA*(*A*) in this paper); Symmetric *March G* algorithm given by van de Goor [7]; march test given by Caşcaval and Bennett [2] (*CB* in this paper – a BIST logic design for this march test is discussed in [3]). In the simulation study, 288 simple 3–coupling faults have been considered.

Table 1 – Fault coverage of simple reduced 3–coupling faults (expressed as %)

March test	**March C–**	*March LR*	*March G*	*NTA*(*A*)	*CB*	*MT–R3CF*
Length	10*n*	18*n*	24*n*	30*n*	38*n*	30*n*
Fault coverage	50	62.5	62.5	63.89	94.91	100

LOGIC DESIGN FOR MARCH TEST *MT–R3CF*

As shown in Fig. 2, the BIST logic can be divided into three parts: address–generation logic, data–generation and response–verification logic, and control logic.

The address generation logic is composed of two up/down counters, for row address (RA) and column address (CA), respectively. Each address counter can be initialised with 0 (InitZero) or 1 (InitUnu) in all the locations. The memory is checked in ascending or descending order, so that the control logic increments (Up) or decrements (Dn) the address counters, depending on the current test sequence (TS).

The data-generation logic supplies data to be written in the memory and the expected data for response monitoring. An unique logic to generate both data for writing operations and expected data for response monitoring can be used. Except on the first initialisation (w_0) and the final checking of the memory (r), the test algorithm is composed of two kinds of march elements, (rw_c) and (rw_crw_c). Every write operation into a cell is preceded by a read operation, and data that must be written into a cell is the complement of the expected data from the previous read operation of the cell. The expected data in the first read operation of a cell is presented in Table 2, where LSB_{CA} and LSB_{RA} denote the less significant bit of the column address of the cell and of the row address of the cell, respectively (see Fig. 1). Data generator is basically composed of a multiplexor with fourteen input

variables, as specified in Table 2, and a counter that supplies the selection code (x) for the input variable.

Table 2 – Expected data in the first read operation of a cell depending on the test sequence

TS	(1)	(2)	(3)	(4)	(5)	(6), (7)	(8), (9)	(10), (11)	(12), (13)
Data	0	1	0	1	0	LSB_{CA}	$LSB_{CA} \oplus LSB_{RA}$	$\overline{LSB_{CA}}$	$\overline{LSB_{CA} \oplus LSB_{RA}}$

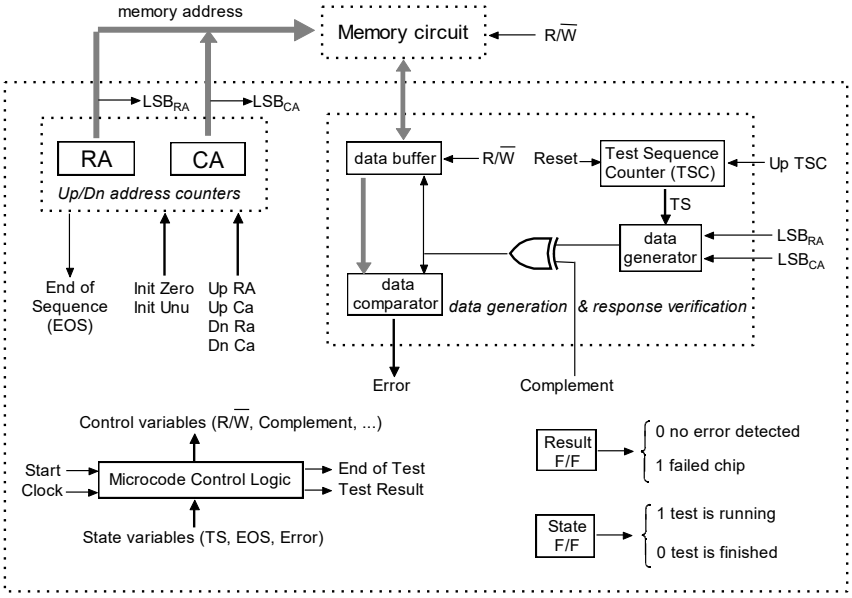

Fig. 2 – The block diagram of BIST logic for mach test *MT–R3CF*

FINAL REMARKS

To rich a high fault coverage, *MT–R3CF* uses different data–backgrounds: solid, column–stripe, and checkerboard data–background, in contrast with other march tests. Nevertheless, the BIST logic for *MT–R3CF* is not significantly more complicated than BIST logic for other march tests (see for example [3]). Regarding the MASB architectures, another response verification method is comparing the outputs of symmetrically placed bits in the tested arrays. Consequently, for these architectures the expected values need not be generated.

REFERENCES

[1] CAŞCAVAL, P., BENNETT, S., HUŢANU, C., (2004), "EFFICIENT MARCH TESTS FOR A REDUCED 3–COUPLING AND 4–COUPLING FAULTS IN RANDOM–ACCESS MEMORIES", *J. ELECTRONIC TESTING*, **20** (3), 227–243.

[2] CAŞCAVAL, P., BENNETT, S., (2001), "EFFICIENT MARCH TEST FOR 3–COUPLING FAULTS IN RANDOM–ACCESS MEMORIES", *J. MICROPROCESSORS & MICROSYSTEMS*, **24** (10), 501–509.

[3] Caşcaval, P., Onofrei, V., (2000), "Built–in Self Testing for Coupling Faults in Random–ACCESS MEMORIES", *BUL .INST. POL. IAŞI*, **XLVI (L)**, 1–4, *AUTOM. & COMP*., 93–101.

[4] Cockburn, B.F., (1994), "Deterministic Tests for Detecting Single V–coupling Faults in RAMs", *J. Electronic Testing*, **5** (1), 91–113.

[5] Franklin, M., Saluja, K., (1990) "Built–in Self Testing of RAMs", *IEEE Comp.*, **23** (10), 45–56.

[6] Nair, R., Thatte, S., Abraham, J. (1978), "Efficient Algorithms for Testing Semiconductor Random–Access Memories", *IEEE Trans. Comput.*, **C–27** (6), 572–576.

[7] van de Goor, A.J. (1993), "Using March Tests to Test SRAMs", *IEEE D&T of Comp.*, **10** (1), 8–14.

[8] Yarmolik, V.N., van de Goor, A.J., Gaydadjiev, G.N., Mikitjuk, V.G. , (1996), "March LR: A test for realistic linked faults", *14th IEEE VLSI Test Symp*., Princeton, USA, 272–280.