Simple March Tests for PSF Detection in RAM

Ireneusz Mrozek
Bialystok Technical University
Computer Science Department
Wiejska 45A, 15-351 Bialystok
POLAND
imrozek@ii.pb.bialystok.pl

Eugenia Buslowska
Bialystok Technical University
Computer Science Department
Wiejska 45A, 15-351 Bialystok
POLAND
ebus@ii.pb.bialystok.pl

Abstract

Previous research has outlined that the only march tests can be in use now to test modern memory chips. Their transparent versions are very efficient for the simple fault testing and diagnoses. In the case of Pattern Sensitive Faults (PSF), they are not such efficient. Conventional memory tests based on only one run march test have constant and low faults coverage for PSF. To increase the probability of the detection of such type of faults, multiple run March test sessions can be used. As shown earlier, the key element of multiple run March test algorithms are memory backgrounds. Only in the case of optimal set of backgrounds the high fault coverage can be achieved. There are many different march tests which we can use in multiplying test scheme. However according to weighted fault coverage measure for march tests, it is not necessary to use complex tests to achieve high fault coverage in multibackground tests. The high fault coverage (espesially for PSF) we can achieve using the simplest tests. Efficiency of mulibackground test sessions based on two simplest march tests (MATS+ and MPS(3N)) and optimal selected backgrounds is the main subject of this paper. All of the analytical calculations are confirmed and validated by adequate experiments.

1 Introduction

Faults modeled from the memory defects can be summarized as follows [1, 2]; Stuck-at-Fault (SF): Either a cell or a line is stuck to logical *0* or *1*. Transition Fault (TF): The $0 \rightarrow 1$ (or $1 \rightarrow 0$) transition is impossible on a cell or a line. Coupling Fault (CF): When in a cell there is a transition $0 \rightarrow 1$ (or $1 \rightarrow 0$), the content of the other cell is changed. CF is generalized to a k-coupling fault when $k-1$ cells are changed and is classified into Inversion or Idempotent coupling faults depending upon what content changed [3]. Retention Faults (RF): A cell fails to retain its logic value after some time. This fault is caused by a broken pull-up resistor. Neighborhood Pattern Sensitive Fault (NPSF): a typical neighborhood pattern sensitive faults preventing the base cell from being transited to a certain value is called 'static' NPSF, and an NPSF is called 'dynamic' when a transition on the neighborhood cells triggers a transition on the base cell. The neighborhood pattern sensitive fault (NPSF) model is not new, but it is still widely discussed in the literature of memory testing, and becoming more and more important for memory testing. The problems with testing of semiconductor memories are very different from testing logic. The main reason is that the fault behaviour of memories is inherently analog, while the used fault models have a digital (logical) nature. Traditional March algorithms [1] have been widely used in RAM testing because of their linear time complexity, high fault coverage, and ease in built-in self-test (BIST) implementation. It is known that the traditional March algorithms do not generate all neighborhood patterns that are required for testing the NPSFs, however, they can be modified to get detection abilities for NPSFs. Based on traditional March algorithms, different approaches have been proposed to detect NPSFs, such as the tiling method [1, 4], two-group method [1], row-March algorithm [4] and transparent testing [3, 5, 6].

2 Transparent memory testing

March tests are superior in terms of test time and simplicity of hardware implementation and consisting of sequences of March elements. The March element includes sequences of read/write (r/w) operations, which are all applied to a given cell, before proceeding to the next cell. The way of moving to the next cell is determined by the address sequence order. During the testing, March tests make use of address sequences called "up" and "down" sequences, denoted as $\Uparrow$ and $\Downarrow$. The notation $\Updownarrow$ means don't care the direction of address order. It should be mentioned that the address sequences do not necessarily have to be counting sequences. As an example of the standard memory tests MATS+ test $\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0)\}$ can be considered, which includes just three phases. The first phase is memory initialization (writing all zero background), while the other two phases are sets of read and write operations allow detecting target faults. The MATS+ test detects all stuck-at faults, address faults, some transition faults and some coupling faults, as well as small portion of NPSF [1]. The transparent technique is a well known memory testing approach that retrieves the initial contents of the memory once the test phase has been finished. It is therefore suitable for periodic field testing while allowing preserving the memory content. A transparent BIST is based on a transparent March test that uses the memory initial data to derive the test patterns. The write data can be either the read value or its opposite value. A transparent test algorithm ensures that the last write data is always equal to the first read value in order to satisfy the transparency property. The procedure to derive a transparent test algorithm from a non transparent one (see [6]) can be summarized by the following steps:

1. Remove the initial sequence (initialization sequence). In most cases, removing such a sequence allows to reduce the test length, without affecting the fault coverage.

2. Add read operations at the beginning of all sequences starting with write operations.

3. Add extra sequence to preserve transparency that is, to retrieve the initial content of the cells.

4. Derive the prediction algorithm. This is done by deleting all write operations from the test sequences, thereafter the resulted sequences are appended at the beginning of the test algorithm.

The transparent version of the MATS+ test has the next notation $\{\Uparrow (ra, w\overline{a}); (r\overline{a}, wa)\}$ [6], where $a \in \{0, 1\}$ and $\overline{a}$ is the negation of the value of a. Transparent tests are particularly suitable for Built-In Self-Test.

A (pure) Transparent March memory BIST should implement the test algorithm issued from the above procedure. However, step 4 is mandatory only if an in-situ signature comparison capability is required. In this case, the signature resulting from the prediction sequences and the one resulting form the remaining test sequences are compared against each other to produce the test's status. In some cases like Adaptive Signature Analyses [7] and Symmetric Memory Tests [8] this step should be avoided. For further discussions, let us focus on two March memory tests, namely MATS+:$\{\Uparrow (ra, w\overline{a}); \Downarrow (r\overline{a}, wa)\}$, which allow to avoid prediction phases based on Symmetric memory testing [8], and modified PS(3N) test - MPS(3N): $\{\Uparrow (ra, w\overline{a}, r\overline{a}, wa, ra)\}$ [9]. It is quite important to emphasize that implementation of MPS(3N) in some cases do not need the value of fault free signature, and the MPS(3N) testing procedure can be interrupt by the system at any time due to the preserving the initial memory contents at any stage of testing. In a case of MATS+ test initial contents will be at the end of test procedure. The main advantage of the transparent memory testing is the test pattern flexibility to initialize the memory with the desired background to cover specific fault models not covered by standard March tests.

3 MATS+ and MPS(3N) memory tests efficiency analyses

To investigate the memory march tests, let us suppose that NPSF*k* includes memory cells with increasing order of addresses $\alpha(0), \alpha(1), \alpha(2), ..., \alpha(k-1)$, such a way that $\alpha(0) < \alpha(1) < \alpha(2) < ... < \alpha(k-1)$ and base cell has the address $\alpha(i)$, where $0 \le i \le k-1$.

Let us focus on the Passive NPSF*k* (PNPSF*k*) as the most difficult faults to be detected. First of all, it should be emphasized that due to scrambling information, as well as specific optimization techniques, there is a huge amount of such type of faults. Any k arbitrary memory cells out of all N memory cells can be involved into the PNPSF*k*. This notation means that k arbitrary cells are involved into the PNPSF*k* and one of the cells is a base cell (b) and the rest $k-1$ cells are the neighbors (n). Any arbitrary cell out of k cells is the base cell for which two transitions can be considered, so there is $2k$ distinct PNPSF*k* depending on the base cell position (k positions). For neighborhood pattern there are 2^{k-1} different patterns. Then the exact number of all PNPSF*k* within the N memory cells is determined according to the following equation [9]:

$$L_1(PNPSFk) = 2 \cdot 2^{k-1} k \begin{pmatrix} N \\ k \end{pmatrix} = k 2^k \begin{pmatrix} N \\ k \end{pmatrix} \quad (1)$$

The number $Q_{M+}(PNPSFk)$ of detectable faults during the one MATS+ memory test run is:

$$Q_{M+}(PNPSFk) = k \begin{pmatrix} N \\ k \end{pmatrix} \quad (2)$$

And the fault coverage FC for MATS+ is:

$$FC_{M+} = \frac{Q_{M+}(PNPSFk)}{L_1(PNPSFk)} 100\% = \frac{1}{2^k} 100\% \quad (3)$$

The exact values of fault coverage for different k and for MATS+ test are presented in Table 1.

In the case of MPS(3N) test we can do the same investigation in some other way. Any k arbitrary memory cells out of all N memory cells can be involved into the PNPSF*k*. It means that any arbitrary $k-1$ cells out of N may be the neighborhood cells and one arbitrary cell out of rest

Table 1. PNPSFk fault coverage for MATS+ test

k	3	4	5	6	7	8	9
FC_{M+}	12.5	6.25	3.12	1.56	0.8	0.4	0.2

$N-k+1$ may be a base cell. Because in base cell two transitions can be considered, so there is $(N-k+1)$ distinct PNPSF*k* for chosen neighborhood cells. In neighborhood cells, there are 2^{k-1} possible patterns. Therefore the number of all PNPSF*k* which was determined by (1) can be determined by (4) too.

$$\begin{aligned} L_2(PNPSFk) &= 2 \times (N-k+1) \times 2^{k-1} \times \begin{pmatrix} N \\ k-1 \end{pmatrix} \\ &= 2^k \times (N-k+1) \begin{pmatrix} N \\ k-1 \end{pmatrix} \qquad (4) \end{aligned}$$

It is easy to show that $L_1(PNPSFk) = L_2(PNPSFk)$. Now, the number $Q_{M3N}(PNPSFk)$ of detectable faults during the one MPS(3N) memery test run is (5):

$$Q_{M3N}(PNPSFk) = 2 \times (N-k+1) \begin{pmatrix} N \\ k-1 \end{pmatrix} \qquad (5)$$

And the fault coverage FC for MPS(3N) is:

$$FC_{M3N} = \frac{Q_{M3N}(PNPSFk)}{L_2(PNPSFk)} 100\% = \frac{1}{2^{k-1}} \qquad (6)$$

The exact values of fault coverage for different k and for MPS(3N) test are presented in Table 2.

Table 2. PNPSFk fault coverage for MPS(3N) test

k	3	4	5	6	7	8	9
FC_{M3N}	25	12.5	6.25	3.125	1.6	0.8	0.4

There are some solutions which allow to increase the values of FC_{M+} shown in Table 1 and Table 2. Among those the most promising is the multiple run memory testing. The key idea behind this approach is the different background selection for increasing the fault coverage. Let us analyze the efficiency of this approach for different number of backgrounds. In the case of two backgrounds the fault coverage will be twice as high as when the second background is inverse version of the first one. More complicated problems arise for three and four runs of memory test with the different backgrounds.

4 Background selection

To achieve high fault coverage of PNPSF*k* for multi-run memory testing, it is quite important to choose appropriate backgrounds. In the case of tests which allow to generate only one pattern (like MATS+, MPS(3N)), the selection algorithm for optimal background selection can be based on the following statements [10].

Statement 1 *In the case of* m *runs of the memory test which allow to generate only one pattern within neighboring cells based on backgrounds* $a_0, a_1, a_2, ..., a_{m-1}$*, an optimal set of such type of backgrounds should have the maximal Hamming distance* $HD(a_k, a_j)$ *between any pair* (a_k, a_j)*,where* $k, j \in \{0, 1, 2, ..., m-1\}$.

There are not known efficient algorithms to generate more then four backgrounds which satisfied statement 1. For the case of four runs of memory test, taking into account Statement 1, we have to estimate the maximum minimal possible Hamming distance between any pairs (B_i, B_j), (B_i, B_l), (B_i, B_r), (B_j, B_l), (B_j, B_r) and (B_l, B_r) out of four backgrounds $\{B_i, B_j, B_l, B_r\}$ $i \neq j \neq l \neq r \in \{1, 2, ..., 2^N\}$. Mathematically this problem can be formulated as [11]:

$$\begin{aligned} &MMHD(B_i, B_j, B_l, B_r) = \underset{\forall i \neq j \neq l \neq r \in \{1,2,\ldots,2^N\}}{MAX} \{ \\ &\quad MIN[\qquad (7) \\ &\quad HD(B_i, B_j), HD(B_i, B_l), HD(B_i, B_r), \\ &\quad HD(B_j, B_l), HD(B_j, B_r), HD(B_l, B_r) \\ &\quad]\}. \end{aligned}$$

Let we have got three first arbitrary backgrounds B_i, B_j and B_l with optimal value of $MMHD(B_i, B_j, B_l)$. As shown earlier, for large N this value can be regarded as integer number $2N/3$, then $HD(B_i, B_j) = HD(B_i, B_l) = HD(B_l, B_j) = 2N/3$. We have to emphasize that in the case of three backgrounds, it is impossible to get greater value of $MMHD(B_i, B_j, B_l)$, that is why in the case of four backgrounds, $MMHD(B_i, B_j, B_l, B_r)$ also can not be greater than $2N/3$. It means that in the case of four backgrounds, the best solution will be obtained when distances between the fourth background and the first three backgrounds are equal: $HD(B_r, B_i) = HD(B_r, B_j) = HD(B_r, B_l) = 2N/3$. Due to the fact that backgrounds B_i and B_j have $S_{01}(B_i, B_j) + S_{10}(B_i, B_j) = 2N/3$ different bits, the third background B_l have been generated by the selection of part of its bits from background B_i and the another part from B_j, as well as inversion of all $S_{00}(B_i, B_j) + S_{11}(B_i, B_j) = N/3$ equal bits for B_i and B_j. In the case when we create the next background B_r as the selection of another parts of $S_{01}(B_i, B_j) +$

$S_{10}(B_i, Bj_{)} = 2N/3$ different bits from backgrounds B_i and B_j and inversion of all $S_{00}(B_i, B_j) + S_{11}(B_i, B_j) = N/3$ bits, this background can be regarded as the third background compared with B_i and B_j. It follows from the conclusion that the background B_r has the same distances $HD(B_r, B_i) = HD(B_r, B_j) = 2N/3$ as the background B_l $HD(B_l, B_i) = HD(B_l, B_j) = 2N/3$. From the procedure of generation of B_l and B_r, we can conclude that in the $S_{01}(B_i, B_j) + S_{10}(B_i, B_j) = 2N/3$ positions with the different bits for B_i and B_j backgrounds, B_l and B_r have inverse value of bits. Then $HD(B_l, B_r) = 2N/3$. To summerize it is easy to show that $HD(B_i, B_j) = HD(B_i, B_l) = HD(B_j, B_l) = HD(B_i, B_r) = HD(B_j, B_r) = HD(B_l, B_r) = 2N/3$.

For the previous example, in the case of two backgrounds $B_i = 011100$, $B_j = 010011$ the third background $B_l = b_{l1}b_{l2}b_{l3}b_{l4}b_{l5}b_{l6} = 101111$ have been generated to satisfy the equality $HD(B_i, B_j) = HD(B_i, B_l) = HD(B_j, B_l) = 2N/3$. To generate new fourth background B_r, its first and second bits have to be inverse value compared with B_i and B_j, namely $b_{r1} = 1$, due to $b_{i1} = b_{j1} = 0$ and $b_{r2} = 0$, because $b_{i2} = b_{j2} = 1$. Then another part (compared with the case of the B_l generation) of the bits (two bits) with opposite value in B_i and B_j should take the value from one background, let it be B_j (for example, $b_{r3} = b_{r4} = 0$) and the second part the values from background B_i (for example, $b_{r5} = b_{r6} = 0$). The final result is $B_r = b_{r1}b_{r2}b_{r3}b_{r4}b_{r5}b_{r6} = 100000$ which is satisfy to the next statement:

Statement 2 *In the case of four runs of the memory test which allows to generate only one pattern within neighboring cells based on four backgrounds B_i,B_j,B_l and B_r ($i \neq j \neq l \neq r \in \{1, 2, ..., 2^N\}$ when N is one bit-wide memory size), an optimal set of such type of backgrounds should satisfy the following equality :*

$$\begin{aligned} HD(B_i, B_j) &= HD(B_i, B_l) = HD(B_j, B_l) = \\ &= HD(B_i, B_r) = HD(B_j, B_r) = \\ &= HD(B_l, B_r) \approx 2N/3 \end{aligned} \tag{8}$$

According to algorithm presented in [11], we can generate the set of four optimal backgrounds. As an example of such type of optimal backgrounds for the case of transparent four runs of memory testing the following set of backgrounds can be chosen for $N = 9$ and all bits of first background are zero:

$$\begin{aligned} B_1 &= b_{11}b_{12}b_{13}b_{14}b_{15}b_{16}b_{17}b_{18}b_{19} = 000000000 \\ B_2 &= b_{21}b_{22}b_{23}b_{24}b_{25}b_{26}b_{27}b_{28}b_{29} = 111111000 \\ B_3 &= b_{31}b_{32}b_{33}b_{34}b_{35}b_{36}b_{37}b_{38}b_{39} = 000111111 \\ B_4 &= b_{41}b_{42}b_{43}b_{44}b_{45}b_{46}b_{47}b_{48}b_{49} = 111000111 \end{aligned}$$

This set of backgrounds will be used in the next section to calculate four runs of MATS+ and MPS(3N) test efficiency especially in terms of PSF.

5 Four runs of MATS+ and MPS(3N) memory test efficiency analyses

Let assume that N is divisible by 3. Let's start our investigation with MATS+ test.

For the first background B_1 $Q_{M+}(B_1) = Q_{M+}(PNPSFk)$. Background B_2 generates

$$Q_{M+}(B_2) = k\left(\binom{2N/3}{k} + \sum_{i=1}^{k-1}\binom{2N/3}{k-i}\binom{N/3}{i}\right)$$

new patterns and background B_3 due to its structure allows to generate the additional portion of patterns calculated as:

$$\begin{aligned} Q_{M+}(B_3) = k\Bigg(\binom{N/3}{k} &+ \sum_{i=1}^{k-1}\binom{2N/3}{k-i}\binom{N/3}{i} \\ &+ \sum_{i=1}^{k-1}\binom{N/3}{k-i}\binom{N/3}{i}\Bigg). \end{aligned}$$

Full amount of the patterns generated after three runs of the MATS+ test based on B_1, B_2, B_3 patterns equals to:

$$\begin{aligned} Q_{M+}(B_{1\_2\_3}) =& k\Bigg(\binom{N}{k} + \binom{N/3}{k} + \binom{2N/3}{k} \\ &+ 2\sum_{i=1}^{k-1}\binom{2N/3}{k-i}\binom{N/3}{i} \\ &+ \sum_{i=1}^{k-1}\binom{N/3}{k-i}\binom{N/3}{i}\Bigg). \end{aligned}$$

Background B_4 due to its structure allows to generate the additional portion of patterns calculated as

$$\begin{aligned} Q_{M+}(B_4) =& k\Bigg(\sum_{i=1}^{k-1}\binom{2N/3}{k-i}\binom{N/3}{i} \\ &+ \sum_{i=1}^{k-1}\binom{N/3}{k-i}\binom{N/3}{i}\Bigg) \end{aligned}$$

Full amount of the patterns generated after four runs of the MATS+ test based on the set of B_1, B_2, B_3, B_4 patterns equals to:

$$\begin{aligned} Q_{M+}(B_{1\_2\_3\_4}) =& k\Bigg(\binom{N}{k} + \binom{N/3}{k}\binom{2N/3}{k} \\ & + 3\sum_{i=1}^{k-1}\binom{2N/3}{k-i}\binom{N/3}{i} \\ & + 2\sum_{i=1}^{k-1}\binom{N/3}{k-i}\binom{N/3}{i}\Bigg) \end{aligned}$$

Taking into account that for real applications, N is a big integer number, $k << N$ and $N^k >> N^{k-1}$ last equation in the case of even N can be simplified to

$$\begin{aligned} Q_{M+}(B_{1\_2\_3\_4}) \approx& kN^k\Bigg(\frac{1}{k!} + \frac{1+2^k}{3^k k!} \\ & + \frac{2}{3^k}\sum_{i=1}^{k-1}\frac{1}{(k-i)! \times i!} \\ & + \frac{3}{3^k}\sum_{i=1}^{k-1}\frac{2^{k-i}}{(k-i)! \times i!}\Bigg) \end{aligned}$$

Then the fault coverage $FC_{MATS+}((B_1, B_2, B_3, B_4), k)$ for the four runs of the MATS+ test with investigated backgrounds can be estimated as [12]:

$$\begin{aligned} FC_{MATS+}((B_1, B_2, B_3, B_4), k) \approx& \Bigg(\frac{1}{2^k} + \frac{1}{3^{k-1}} \\ & - \frac{1}{2^k 3^{k-1}} + \frac{1}{2^k 3^{k-1}}\sum_{i=1}^{k-1} 2^{k-i}\binom{k}{i}\Bigg)100\% \end{aligned} \tag{9}$$

In the case of MPS(3N) for the first all zero bit background (B_1) $Q_{M3N}B_1 = Q_{M3N}(PNPSFk)$. Now let k_n means the number of neighbourhood cells ($k_n = k - 1$). Then the background B_2 generates:

$$\begin{aligned} Q_{M3N}(B_2) =& 2(N-k_n)\Bigg(\binom{2N/3}{k_n} \\ & + \sum_{i=1}^{k_n-1}\binom{2N/3}{k_n-i}\binom{N/3}{i}\Bigg) \end{aligned} \tag{10}$$

new patterns and background B_3 due to their structure allow to generate the additional portion of patterns calculated as:

$$\begin{aligned} Q_{M3N}(B_3) =& 2(N-k_n)\Bigg(\binom{N/3}{k_n} \\ & + \sum_{i=1}^{k_n-1}\binom{2N/3}{k_n-i}\binom{N/3}{i} \\ & + \sum_{i=1}^{k_n-1}\binom{N/3}{k_n-i}\binom{N/3}{i}\Bigg) \end{aligned} \tag{11}$$

Full amount of the patterns generated after three runs of the MPS(3N) test equals to:

$$\begin{aligned} Q_{M3N}(B_{1\_2\_3}) =& 2(N-k_n)\Bigg(\binom{N}{k_n} \\ & + \binom{N/3}{k_n} + \binom{2N/3}{k_n} \\ & + 2\sum_{i=1}^{k_n-1}\binom{2N/3}{k_n-i}\binom{N/3}{i} \\ & + \sum_{i=1}^{k_n-1}\binom{N/3}{k_n-i}\binom{N/3}{i}\Bigg) \end{aligned} \tag{12}$$

Background B_4 due to its structure allows to generate the additional portion of patterns calculated as:

$$\begin{aligned} Q_{M3N}(B_4) =& 2(N-k_n)\Bigg(\sum_{i=1}^{k_n-1}\binom{2N/3}{k_n-i}\binom{N/3}{i} \\ & + \sum_{i=1}^{k_n-1}\binom{N/3}{k_n-i}\binom{N/3}{i}\Bigg) \end{aligned} \tag{13}$$

Full amount of the patterns generated after four runs of the MPS(3N) test based on the set of B_1, B_2, B_3, B_4 backgrounds equals to

$$\begin{aligned} Q_{M3N}(B_{1\_2\_3\_4}) =& 2(N-k_n)\Bigg(\binom{N}{k_n} \\ & + \binom{N/3}{k_n}\binom{2N/3}{k_n} \\ & + 3\sum_{i=1}^{k_n-1}\binom{2N/3}{k_n-i}\binom{N/3}{i} \\ & + 2\sum_{i=1}^{k_n-1}\binom{N/3}{k_n-i}\binom{N/3}{i}\Bigg) \end{aligned} \tag{14}$$

According to the same assumptions for k and N defined earlier:

$$Q_{M3N}(B_{1\_2\_3\_4}) \approx \left(\frac{1}{k_n!} + \frac{1+2^{k_n}}{3^{k_n}k_n!} + \frac{2}{3^{k_n}} \sum_{i=1}^{k_n-1} \frac{1}{(k_n-i)! \times i!} + \frac{3}{3^{k_n}} \sum_{i=1}^{k_n-1} \frac{2^{k_n-i}}{(k_n-i)! \times i!} \right) \tag{15}$$

Then the fault coverage $FC_{MPS3N}((B_1, B_2, B_3, B_4), k)$ for the four runs of MPS(3N) test with the investigated backgrounds can be estimated as:

$$FC_{MPS3N}((B_1, B_2, B_3, B_4), k) \approx \left(\frac{1}{2^{k_n}} + \frac{1}{3^{k_n-1}} - \frac{1}{2^{k_n}3^{k_n-1}} + \frac{1}{2^{k_n}3^{k_n-1}} \sum_{i=1}^{k_n-1} 2^{k_n-i} \binom{k_n}{i} \right) \tag{16}$$

6 Experimental results

In this section, we want to confirm the analytical results by adequate experiments. In Table 3 the fault coverage of 4rMATS+ and 4rMPS(3N) are presented. These results have been calculated according to (9) and (16).

Table 3. Analytical results: 4rMATS+ and 4rMPS(3N) fault coverage for PNPSF*k*

Test/k	3	4	5	7
MATS+	47.23%	24.54%	12.42%	3.12%
MPS(3N)	83.33%	47.23%	24.54%	6.24%

To validate analytical results from Table 3, many experiments have been done. The experiments was done for PNPSF3 and PNPSF5 faults, different sizes of memory and selected optimal backgrounds from section 4. In each cases, all PNPSF3 and then PNPSF5 faults were generated. It allowed to obtain exact number of activated by 4rMATS+ and 4rMPS(3N) test session faults. So, each time we could calculate exact value of fault coverage of this test sessions. All experimental results are presented in Tables 4.

As presented in the Table 4 fault coverage confirmed the analytical results obtained according to (9) and (16). It should be noticed that in such type of memory testing, the fault coverage minimally depends on N. The especially high influence on the fault coverage can be observed for very small N what in real applications has no matter.

Table 4. Experimental results: 4rMATS+ and 4rMPS(3N) fault coverage for PNPSF3 and PNPSF5

Fault/N	9	33	66	129
MATS+				
PNPSF3	49.11%	47.73%	47.48%	47.29%
PNPSF5	12.50%	12.46%	12.44%	12.43%
MPS(3N)				
PNPSF3	87.50%	84.38%	83.85%	83.38%
PNPSF5	25.00%	24.70%	24.65%	24.58%

7 Conclusions

In this paper, the fault coverage of the test sessions based on simple march algorithms (MATS+ and MPS(3N)) was presented. In one test session, the same test had been running four times on different background each time. The background was selected according to algorithm which allows to generate optimal (for MATS+ and MPS(3N) tests) set of backgrounds. The main attention in this paper was focused on the efficiency of PSF detection. We can see that both of the simple tests can be successfully used in multi-backgrounds test scheme. If we compare the efficiency of the MATS+ and MPS(3N) test sessions, we can say that from point of view of PSF detection, MPS(3N) gives us better results than MATS+. For MPS(3N) test and optimal set of backgrounds, we have received very high fault coverage (for 4rMPS(3N) and PNPSF3 the fault coverage equal to 83.38%). Moreover, in transparent version of MPS(3N), testing procedure can be interrupted by the system at any time due to the preserving the initial memory contents at any stage of testing. In the case of MATS+ test, the initial contents will be at the end of test procedure. However the big drawback of MPS(3N) test is that it doesn't allow to detect address decoder faults which are detectable by MATS+ test. The simple solution of this problem is to add to the test session, which is based on MPS(3N) test, one iteration of MATS+ test or another very short test which can detect address decoder faults.

Acknowledgement

This work was supported by the Rector of Bialystok University of Technology and the Dean of Faculty of Computer Science at Bialystok University of Technology (grant number: S/WI/6/08).

References

[1] A. J. v. d. Goor, *Testing Semiconductor Memories: Theory and Practice*. Chichester, England: John Wiley & Sons, 1991.

[2] J. P. Hayes, “Detection of pattern-sensitive faults in random-access memories,” *IEEE Trans. Computers*, vol. 24, no. 2, pp. 150–157, 1975.

[3] B. F. Cockburn, “Deterministic tests for detecting scrambled pattern-sensitive faults in RAMs,” in *MTDT ’95: Proceedings of the 1995 IEEE International Workshop on Memory Technology, Design and Testing*. Washington, DC, USA: IEEE Computer Society, 1995, pp. 117–122.

[4] M. Franklin and K. K. Saluja, “Testing reconfigured RAM’s and scrambled address RAM’s for pattern sensitive faults.” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 15, no. 9, pp. 1081–1087, 1996.

[5] M. G. Karpovsky and V. N. Yarmolik, “Transparent memory testing for pattern-sensitive faults,” in *Proceedings of the IEEE International Test Conference on TEST: The Next 25 Years*. Washington, DC, USA: IEEE Computer Society, 1994, pp. 860–869.

[6] M. Nicolaidis, “Theory of transparent BIST for RAMs,” *IEEE Trans. Comput.*, vol. 45, no. 10, pp. 1141–1156, 1996.

[7] V. N. Yarmolik, Y. Klimets, and S. Demidenko, “March ps(23n) test for DRAM pattern-sensitive faults,” in *ATS ’98: Proceedings of the 7th Asian Test Symposium*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 354–357.

[8] V. N. Yarmolik, “Contents independent RAM built in self test and diagnoses based on symmetric transparent algorithm,” in *DDECS ’00: Proceedings of the 3rd Workshop on Design and Diagnostics of Electronic Circuits and Systems*, Smolenice - Slowakia, April 5-7 2000, pp. 220–227.

[9] V. Yarmolik, I. Murashko, A. Kummert, and A. Ivaniuk, *Transparent Testing of Digital Memories*. Minsk, Belarus: Bestprint, 2005.

[10] S. V. Yarmolik and I. Mrozek, “Multi background memory testing,” in *MIXDES 07: Proceedings of the 14th International Conference Mixed design of integrated circuits and systems*. Ciechocinek, Poland: IEEE Computer Society, June 21-23 2007, pp. 511–516.

[11] I. Mrozek and V. N. Yarmolik, “Optimal backgrounds selection for multi run memory testing,” in *DDECS ’08: Proceedings of the IEEE International Workshop on Design and Diagnostics of Electronic Circuits and Systems*, Bratislava, Slovakia, April 16-18, 2008, pp. 332–338.

[12] ——, “MATS+ transparent memory test for pattern sensitive fault detection,” in *MIXDES ’08: Proceedings of the 15th International Conference Mixed design of integrated circuits and systems*, Poznan, Poland, June 19-21, 2008, pp. 493–498.