

Transparent Word-oriented Memory BIST Based on Symmetric March Algorithms

V.N. Yarmolik^{1,2}, I.V. Bykov¹, S. Hellebrand³, H.-J. Wunderlich³

¹ Computer Systems Department, Belarussian State University of Informatics and Radioelectronics, P.Brovki 6, Minsk, 220027, Belarus
yarmolik@yvn_poit.minsk.by

² Bialystok University of Technology, Department of Computer Science, Poland

³ Division of Computer Architecture, University of Stuttgart, Breitwiesenstr. 20/22, 70656 Stuttgart, Germany

Abstract. The paper presents a new approach to transparent BIST for word-oriented RAMs which is based on the transformation of March transparent test algorithms to the symmetric versions. This approach allows to skip the signature prediction phase inherent to conventional transparent memory testing and therefore to significantly reduce test time. The hardware overhead and fault coverage of the new BIST scheme are comparable to the conventional transparent BIST structures. Experimental results show that in many cases the proposed test techniques achieve a higher fault coverage in shorter test time.

1 Introduction

Today semiconductor memory testing becomes one of the strategic position in the area of computer systems design and diagnosis. It can be explained by the following two main reasons: memory devices are used in any computer and most other digital systems and play a key role in terms of correct functioning the complete system, and on the other hand, advances in memory technology make memory testing more and more complicated due to appearance of the new defect mechanisms in memory devices and constraints of fault coverage and the time spent on the test procedures.

In this paper we concentrate our attention on one of the most important group of test techniques used for solving the problem of memory testing: transparent Built-In Self-Test (BIST) for Random Access Memories. These techniques were introduced due to the problem of the limited accessibility of embedded memories which are widely used in many critical applications (medical electronics, railway control, avionics, telecommunications etc.) and allow to restore the initial contents of memories at the end of the test session.

A number of BIST approaches were developed in the past [1-10]. These approaches are devoted to many aspects of BIST for RAMs: test algorithms, test hardware implementation, efficient mechanisms for consistency checking etc.

Taking into account the results related to transparent memory testing [2, 4, 10, 11] it should be concluded that the most frequently used test algorithms for transparent RAM BIST are march tests [12-14] which combine a high fault coverage with an acceptable test time even for memories of the large sizes (1M and higher). Another advantage of these algorithms is that they can be easily extended to transparent versions [11]. However, the traditional approaches to transforming non-transparent march algorithms to transparent tests implies significant increasing test time needed for calculating the learnt signature. It can be illustrated by the following example. Let us consider four-bit memory and transparent version of the MATS+ algorithm [12] (see Fig 1.) which can be written as $\{\Leftrightarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w1)\}$. Throughout the paper we will use the notations presented below:

1. a° : complementary value of $a \in \{0, 1\}$;
2. \Uparrow : increasing addressing order;
3. \Downarrow : decreasing addressing order;
4. \Leftrightarrow : arbitrary addressing order;
5. $w0, w1$: write 0, 1 into memory cell;
6. $r0, r1$: read memory cell, expected value is 0, 1;
7. wa, wa° : write a, a° into memory cell;
8. ra, ra° : read memory cell and feed result into signature register, expected value is a, a° ;
9. (ra°) : read memory cell with expected value a and feed a° into signature register.

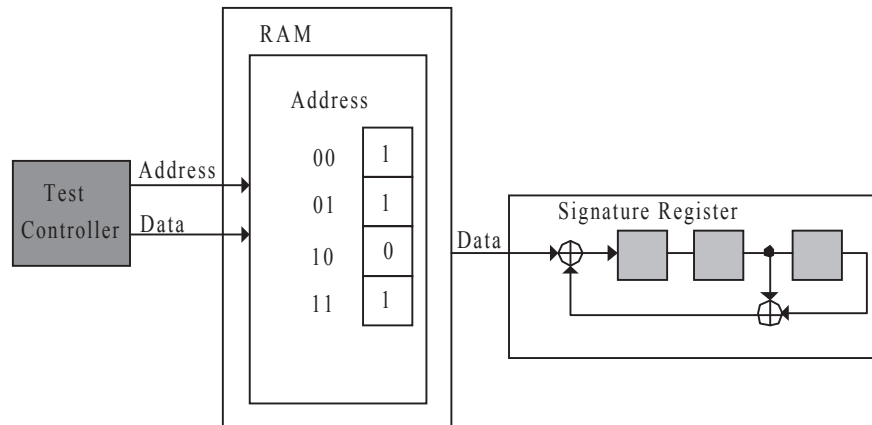


Fig. 1. BIST Scheme for RAMs

MATS+ algorithm can be transformed into a transparent test by removing the first march element ($w0$) which is used for RAMs initialization and replacing read and write operation with fixed values "0" and "1" with the appropriate values which depend on the sequence of Read and Write operations of the initial algorithm [11]. For MATS+ its transparent version consists of the following two phases:

1. $\{\Uparrow (ra); \Downarrow (ra^\circ)\}$;

2. $\{\uparrow (ra, wa^c); \downarrow (ra^c, wa)\}$.

The former phase called "signature prediction phase" is needed for calculating the learnt signature which depends on the memory contents, and the latter phase is used for RAM's testing. During the testing process after each Read operation the obtained data is fed into the signature register, and at the end of the test session the final signature S is compared with the learnt signature S^* .

In general, the signature prediction phase consists of all Read operations of the complete march algorithm. For the considered example it implies that one third of the test time is required for signature prediction, and this ratio is similar for any transparent march algorithm. To cope with this problem a new technique for transparent testing bit-oriented RAMs was proposed in [15]. In this paper we extend it to the case of word-oriented memories and show that this systematic approach allows to avoid spending extra time on performing signature prediction phase without any losses in fault coverage. The paper is organized as follows: The basic principles of the proposed BIST schemes are presented in Section 2. Section 3 demonstrates the general applicability of the new approach and Section 4 analyzes the experimental results and the fault coverage of the proposed technology for transparent RAMs testing.

2 March Algorithms Symmetry and Signature Prediction

It should be noticed that most of the march test algorithms used for transparent RAM BIST produce test data with a high degree of symmetry. To illustrate this statement, consider MATS+ algorithm again and the BISTed four-bit RAM shown in Figure 1. For the fault-free memory during the both phase the data sequence read from the memory can be represented the binary vector $(a_1, a_2, a_3, a_4, a_4^c, a_3^c, a_2^c, a_1^c) = (1, 1, 0, 1, 0, 1, 0, 0)$. The so-called property of symmetry of march algorithms is concluded in generation of bit sequence on the output of the RAM in reverse order and with complemented entries compared to the sequence produced by the previous march elements. Combining such a kind of symmetry with a certain type of signature register allows to completely omit the signature prediction phase of conventional transparent memory BIST techniques. The proposed transparent technique is based on the switching between the regular feedback polynomial and its reciprocal polynomial during signature analysis. To detail the new test methodology, consider the following formal definitions first introduced in [15].

Definition 1: The signature obtained for a test data string $d \in \{1, 0\}^n$ using single input signature register with feedback polynomial $h(X) \in GF(2)[X]$ and initial state $s \in \{1, 0\}^k$ is denoted by $sig(d, s, h)$.

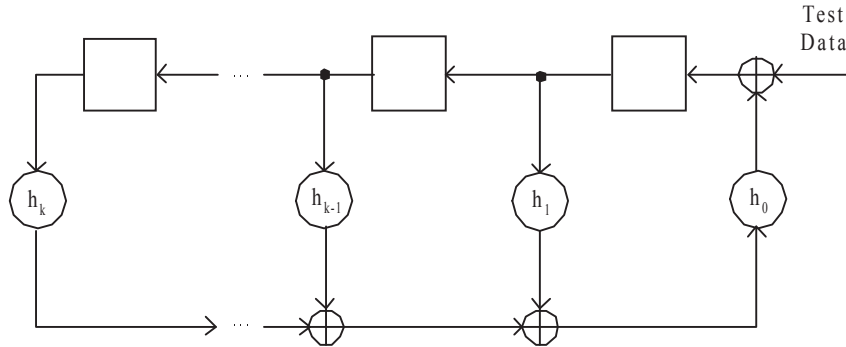
Definition 2: Let $h(X) = h_k X^k + h_{k-1} X^{k-1} + \dots + h_1 X + h_0$ be a polynomial of degree k . Then $h^*(X) = X^k h(X^{-1}) = h_k + h_{k-1} X + \dots + h_1 X^{k-1} + h_0 X^k$ denotes the reciprocal polynomial (the general structures of signature registers corresponding to $h(X)$ and $h^*(X)$ are shown in Figure 2).

Definition 3: Let $d=(d_0, \dots, d_{n-1}) \in \{1, 0\}^n$ be a data stream, then $d^*=(d_{n-1}, \dots, d_0) \in \{1, 0\}^n$ denotes the data stream with components in reverse order, and $d^c=(d_0^c, \dots, d_{n-1}^c) \in \{1, 0\}^n$ denotes the data stream with inverted components.

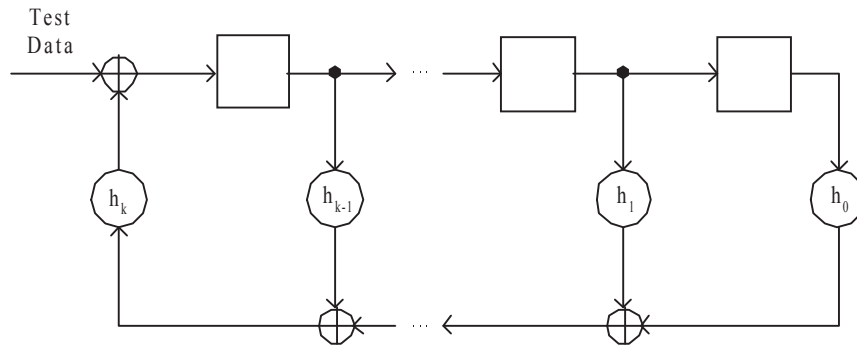
Consider a single input signature register with initial state $s \in \{1, 0\}^k$ with structure defined by feedback polynomial $h(X) \in GF(2)[X]$. Let $d \in \{1, 0\}^n$ be a test data stream with corresponding signature $\sigma = sig(d, s, h)$. It was shown [15] that signature analysis using $h^*(X)$ as feedback polynomial and the reversed signature σ^* as an initial state provides the original state s as signature for the data stream d^* . In other words,

$$sig(d^*, sig(d, s, h)^*, h^*)=s.$$

Figure 3 illustrates the above proposition.



a) single-input signature register with feedback polynomial $h(X)$



b) single-input signature register with feedback polynomial $h^*(X)$

Fig. 2. Single-input signature registers with original and reciprocal feedback polynomial

The results obtained for single output signature analyzer can be extended to the case of multi-input signature register (MISR) and word-oriented RAM which is typical in practice. Let us consider BILBO-type MISR [16] presented in Figure 4.

Transparent Word-oriented Memory BIST Based on Symmetric March Algorithms 5

This kind of signature analyzer (Type 1) is described by the original polynomial $h(X)=h_kX^k+ h_{k-1}X^{k-1}+...+ h_1X+h_0$, however a state of the register depends on a k-bit word $d_{k-1}...d_1d_0$ compacted instead just one bit d . The functioning of such an analyzer can be described by the following recursive expressions:

$$s_{k-1}(n)=d_{k-1}(n)+ \sum_{i=0}^{k-1} s_i(n-1)h_i \tag{1}$$

...

$$s_1(n)=d_1(n)+s_2(n-1) \tag{2}$$

$$s_0(n)=d_0(n)+s_1(n-1) \tag{3}$$

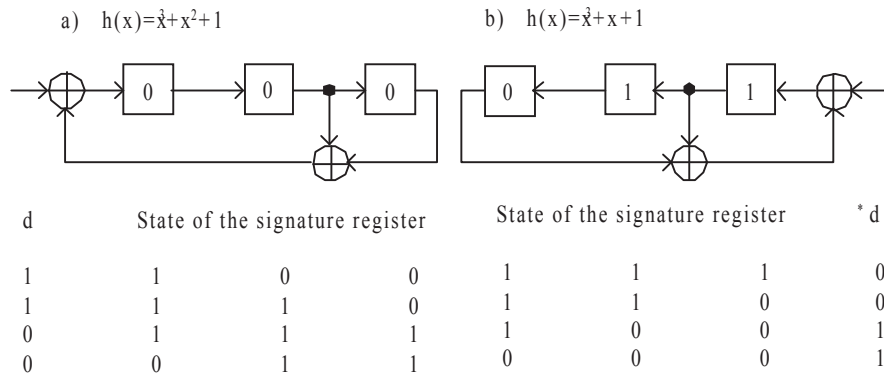


Fig. 3. Signature analysis with the original and reciprocal feedback polynomial

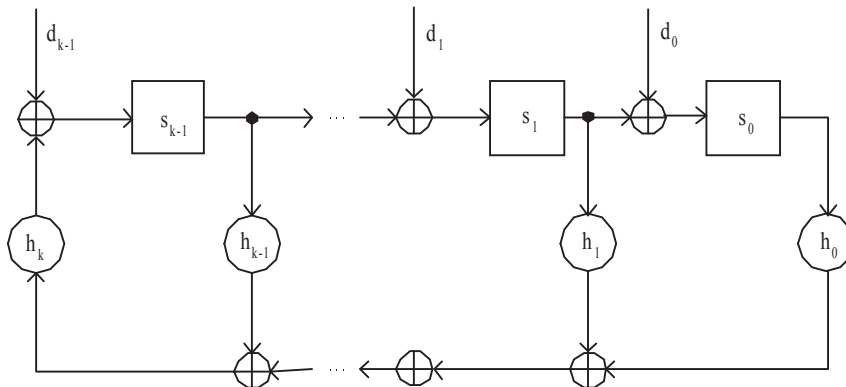


Fig. 4. Multi-Input Signature Register (Type1)

It is obvious, that the previous states of the register $s_{k-1}(n-1), \dots, s_1(n-1), s_0(n-1)$ can be found from (1)-(3) as follows:

$$s_{k-1}(n-1) = s_{k-2}(n) + d_{k-2}(n) \tag{4}$$

...

$$s_1(n-1) = s_0(n) + d_0(n) \tag{5}$$

$$s_0(n-1) = s_{k-1}(n) + d_k \tag{6}$$

$$s_1(n) + \sum_{i=0}^{k-2} (s_i(n) + d_i(n)) h_{i+1}$$

The expressions (4)-(6) define the structure of the MISR (Type 2) corresponding to the reciprocal polynomial $h^*(X) = h_k + h_{k-1}X + \dots + h_1X^{k-1} + h_0X^k$ (see Figure 5).

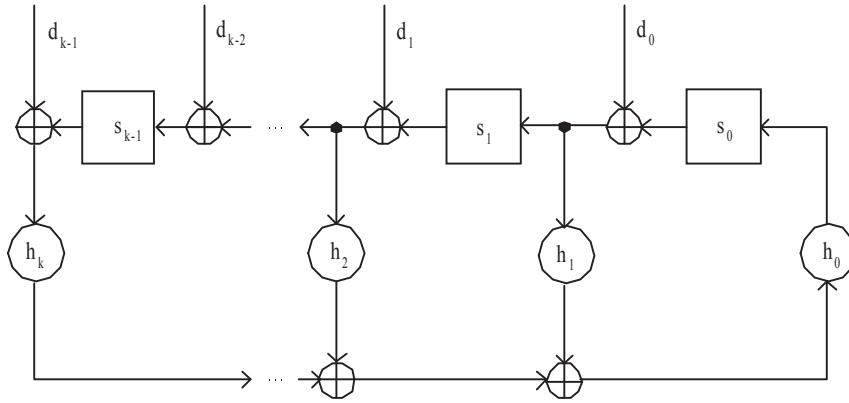


Fig. 5. Multi-Input Signature Register (Type2)

Let $msig(D, S, h)$ be a signature obtained using MISR of Type 1 for a sequence of binary-valued vectors $D = D_0 D_1, \dots, D_m$, where $D_i = (d_0, d_1, \dots, d_{k-1})$, $S \in \{0, k\}^k$ is initial state of MISR, and h is polynomial $h(X) = h_k X^k + h_{k-1} X^{k-1} + \dots + h_1 X + h_0 \in GF(2)[X]$. Denote $D^* = D_m, D_{m-1}, \dots, D_0$ the sequence of binary-valued vectors in reverse order to the sequence D . Then the following theorem is valid for the case of signature analysis based on symmetric algorithms and MISRs of Type 1 and Type 2.

Theorem: Let $D = D_0 D_1, \dots, D_m$ be a data stream compressed to the signature $\sigma = msig(D, S, h)$ by MISR of Type 1. Then signature analysis based on MISR of Type 2 results in the following final signature $msig^{-1}(D^*, \sigma, h^*) = S$, where σ is the initial state of MISR of Type 2.

Proof: Consider a compressed sequence consisting of the only binary-valued vector $(d_0, d_1, \dots, d_{k-1})$. In this case signature analysis based on MISR of Type 1 and initial state $S = (s_{k-1}, \dots, s_0)$ provides the signature

$$\sigma = (d_{k-1} + \sum_{i=0}^{k-1} s_i h_i, s_{k-1} + d_{k-2}, \dots, s_1 + d_0).$$

Signature analysis based on MISR of Type 2 results in

$$(s_{k-1} + d_{k-2} + d_{k-2}, \dots, s_1 + d_0 + d_0, d_{k-1} + (d_{k-1} + \sum_{i=0}^{k-1} s_i h_i) h_k + \sum_{i=1}^{k-1} (s_i + d_{i-1}) h_i + \sum_{i=1}^{k-1} d_{i-1} h_i) = (s_{k-1}, \dots, s_0) = S.$$

The proof for a sequence of binary-valued vectors with an arbitrary length m follows by induction.

q. e. d.

An example of signature analysis with MISRs is presented in Figure 6.

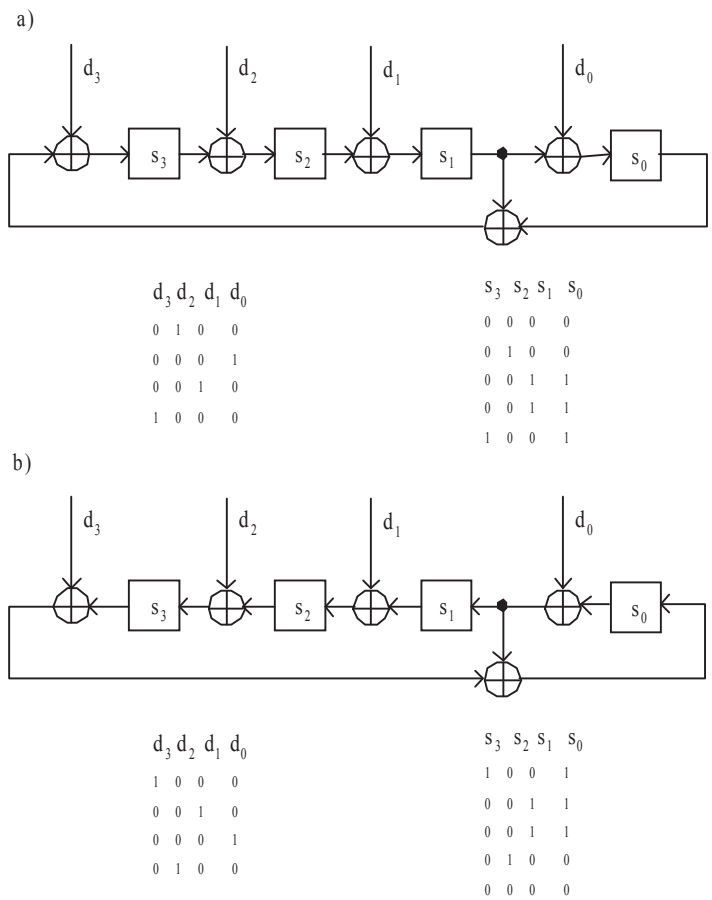


Fig. 6. Signature analysis with multi-input registers

3 Symmetric March Algorithms Construction

The proposed BIST schemes for bit- and word-oriented RAMs can be used with any transparent march test which is symmetric in the following sense.

Definition 4: Let $D \in \{0, 1\}^{2n}$ be a data string. D is called symmetric, if there exists a data string $d \in \{0, 1\}^n$ with $D=(d, d^*)$ or $D=(d, d^c)$. A transparent march test is called symmetric, if it produces a symmetric data string D .

In general, it cannot be expected, that an arbitrary transparent march test is symmetric. However, any march test can be easily extended to fully symmetric version. Consider the March C- algorithm as an example [14]. The initial algorithm version is defined as

$$\{\Leftrightarrow(w0); \Uparrow(r0, w1); \Uparrow(r1, w0); \Downarrow(r0, w1); \Downarrow(r1, w0); \Leftrightarrow(r0)\}$$

and its transparent version is

$$\{\Uparrow(ra, wa^c); \Uparrow(ra^c, wa); \Downarrow(ra, wa^c); \Downarrow(ra^c, wa); \Downarrow(ra)\}.$$

The test data stream fed to the signature analyzer is not symmetric in sense of Definition 4. To satisfy its requirements an additional march element (shown in bold) has to be introduced:

$$\{\Uparrow(\mathbf{ra}^c); \Uparrow(ra, wa^c); \Uparrow(ra^c, wa); \Downarrow(ra, wa^c); \Downarrow(ra^c, wa); \Downarrow(ra)\}.$$

Despite the extended algorithm version will increase the test time from $O(9n)$ to $O(10n)$, there is still a considerable gain in efficiency compared to the conventional $14n$ transparent Marc C- test with signature prediction. Similarly, any known transparent algorithms can be transformed to its symmetric versions. Table 1 shows the resulting test lengths for some commonly used transparent march tests [12].

Table 1. Comparison of test complexities for conventional transparent and symmetric transparent march tests

Algorithm	Transparent BIST		Symmetric Transparent BIST	
	Time for Signature Prediction	Time for Transparent Test	Overall Time	Overall Time
MATS+	2n	4n	6n	4n
March C-	5n	9n	14n	10n
March A	4n	14n	18n	16n
March B	6n	16n	22n	18n
March X	3n	5n	8n	6n
March Y	5n	7n	12n	8n

4 Fault Coverage Issues

For investigation the ability of the proposed approach to transparent BIST for word-oriented memories 16K RAM of organization $8 \times 2K$ was simulated (discussion of the investigated fault models (SAF: stuck-at faults, TF: transition faults, CFid:

idempotent coupling faults, CFin: inversion coupling faults can be found in [12, 13]). As can be observed (see Table 2), the new BIST scheme provides fault coverage comparable with conventional BIST technique.

5 Conclusions

A new approach to transparent BIST for word-oriented RAMs has been proposed which exploits symmetries in march algorithms. Compared to traditional transparent BIST schemes this new approach provides significant reduction in test time while preserving the benefits of conventional approaches with respect to hardware overhead and fault coverage.

Table 2. Simulation results (fault coverage in % for single faults) for 8x2K RAM

Algorithm	March C		March C-		March X	
	Conventional	Symmetric	Conventional	Symmetric	Conventional	Symmetric
SAF	100	100	99.59	100	100	100
TF	100	100	99.69	100	99.70	99.95
Cfid	99.98	100	99.91	100	49.98	49.98
Cfin	99.94	100	99.78	100	99.87	99.89

References

1. V.C. Alves, M. Nicolaidis, P.Lestrat, and B.Courtois: Built-In Self-Test for Multi-Port RAMs; Proceedings IEEE International Conference on Computer-Aided Design, ICCAD-91, November 1991, pp. 248-251.
2. S. Barbagallo, F. Corno, P.Prinetto, M.Sonza Reorda: Testing a Switching Memory in Telecommunication System; Proceedings IEEE International Test Conference, Washington, DC, Oct. 1995, pp. 947-953.
3. H. Cheung, S. K. Gupta: A BIST Methodology for Comprehensive Testing of RAM with Reduced Heat Dissipation; Proceedings IEEE International Test Conference, Washington, DC, Oct. 1996, pp. 386-395.
4. B. Cockburn, Y.-F. Sat: Synthesized Transparent BIST for Detecting Scrambled Pattern-Sensitive Faults in RAMs; Proceedings IEEE International Test Conference, Washington, DC, Oct. 1995, pp. 23-32.
5. R. Dekker, F. Beenker, and L. Thijssen: Realistic Built-In Self-Test for Static RAMs; IEEE Desing & Test of Computers, Vol. 6, No. 1, Feb. 1989, pp. 26-34.
6. K. Kinoshita, K. K. Saluja: Built-In Testing of Memory Using an On-Chip Compact Testing Scheme; IEEE Transactions on Computers, Vol. C-35, No. 10, October 1986, pp. 862-870.
7. K.T. Le, K.K. Saluja: A Novel Approach for Testing Memories Using a Built-In Self-Testing Technique; Proceedings IEEE International Test Conference, Washington, DC, Oct. 1986, pp. 830-839.

8. P. Olivo, M. Dalpasso: Self-Learning Signature Analysis for Non-Volatile Memory Testing; Proceedings IEEE International Test Conference, Washington, DC, Oct. 1996, pp. 303-308.
9. N. Sakashita et al.: A Built-in Self-Test Circuit with Timing Margin Test Function in a 1 Gbit Synchronous DRAM; Proceedings IEEE International Test Conference, Washington, DC, Oct. 1996, pp. 319-324.
10. V. N. Yarmolik, S. Hellebrand, H.-J. Wunderlich; Self-Adjusting Output Data Compression: An Efficient BIST Technique for RAMs; Proceedings Design and Test in Europe (DATE'98), Paris, February 1998, pp. 173-179.
11. M. Nicolaidis: Transparent BIST for RAMs; Proceedings IEEE International Test Conference, Baltimore, MD, Oct. 1992, pp. 598-607.
12. A. J. Van de Goor: Testing Semiconductor Memories, Theory and Practice; Chichester: John Wiley & Sons, 1991.
13. A. J. Van de Goor: Using March Tests to Test SRAMs; IEEE Desing & Test of Computers, Vol. 10, No. 1, March 1993, pp. 8-14.
14. M. Marinescu: Simple and Efficient Algorithms for Functional RAM Testing; Proceedings IEEE International Test Conference, 1982, pp. 236-239.
15. V.N.Yarmolik, S.Hellebrand, H.-J.Wunderlich: Symmetric Transparent BIST for RAMs, DATE-99, Munich, March 9-12, 1999, pp. 702-707.
16. B. Koenemann, J. Mucha, G. Zwihoff: Built-In Logic Block Observation Technique; Proceedings IEEE International Test Conference, 1979, pp. 37-41.