Symmetric Transparent BIST for RAMs

V. N. Yarmolik
Computer Systems Department
Belarussian State University of Informatics and Radioelectronics, Minsk, Belarus

S. Hellebrand, H.-J. Wunderlich
Division of Computer Architecture
University of Stuttgart
Germany

Abstract

The paper introduces the new concept of symmetric transparent BIST for RAMs. This concept allows to skip the signature prediction phase of conventional transparent BIST approaches and therefore yields a significant reduction of test time. The hardware cost and the fault coverage of the new scheme remain comparable to that of a traditional transparent BIST scheme. In many cases, experimental studies even show a higher fault coverage obtained in shorter test time.

1 Introduction

Modern computer systems typically contain a variety of embedded memory arrays like caches, branch prediction tables or priority queues for instruction execution [4, 14]. Fault free memory operations are crucial for the correct behavior of the complete system, and thus, efficient techniques for production testing as well as for periodic maintenance testing are mandatory to guarantee the required quality standards. However, advances in memory technology and in system design turn memory testing into a more and more challenging problem. Due to the limited accessibility of embedded memories conventional methods for external testing can no longer provide satisfactory solutions. A number of theoretical and practical built-in self-test (BIST) approaches, which have been proposed in the past, offer the basis to overcome this problem in present-day systems-on-a-chip [1 - 3, 5 - 8, 12, 13, 17 - 20, 22]. These approaches range from on-chip random pattern generation and efficient mechanisms for repeated consistency checking to deterministic test schemes targeting specific fault models. With increasing memory densities the relative area overhead for the BIST implementation becomes negligible.

This work was supported by BMBF-Grant WEI-X023.2.

For deterministic memory BIST march tests have been widely accepted, because they combine a high fault coverage with a test time of order n, where n denotes the size of the memory [9 - 11, 15, 16, 21]. Furthermore, classical march tests can easily be extended to transparent tests which leave the memory contents unchanged and therefore are especially suitable for periodic maintenance testing [17]. However, since march tests scan the complete memory several times, test time becomes an issue of growing importance with increasing memory densities. In particular, transparent BIST for maintenance purposes may become infeasible, because it requires a considerable amount of extra time to compute a reference result each time before it is applied. Figure 1 illustrates this for a small example memory and a transparent version of the MATS+ algorithm [16].

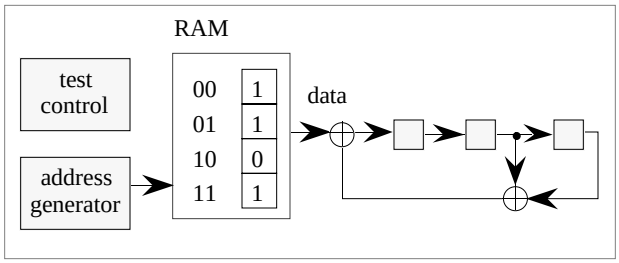


Figure 1: Example for transparent BIST.

Following the notations defined in Table 1 the original MATS+ algorithm can be written as $\{\Updownarrow(w0); \Uparrow(r0, w1); \Downarrow(r1, w0)\}$. It is transformed into a transparent test by deleting the initialization phase and replacing read and write operations with fixed „0“ or „1“ values by read and write operations with the appropriate variable values [17]. For the transparent version $\{\Uparrow(ra, wa^c); \Downarrow(ra^c, wa)\}$ a BIST session proceeds in two phases changing from increasing to decreasing address order. After each read operation the obtained data are fed into the signature analyzer, and at the end of the second phase the final signature σ has to be compared to a reference signature σ_{REF}.

This reference signature, however, depends on the memory contents and must be computed in an extra „signature prediction“ phase before the test can be started. In general, the signature prediction phase basically consists of all read-operations of the complete test. For the transparent MATS+ algorithm it is described by {⇑(ra); ⇓((ra)^c)} and requires an extra time of 2n. The time for a complete maintenance test based on MATS+ is 6n, which implies that one third of the test time is required for signature prediction. This ratio is similar for any of the known transparent BIST algorithms.

Symbol	Meaning
a^c	complementary value of a ∈ {0, 1}
⇑	increasing addressing order
⇓	decreasing addressing order
⇕	don’t care addressing order
w0, w1	write 0, 1 into memory cell
r0, r1	read memory cell, expected value is 0, 1
wa, wa^c	write a, a^c into memory cell
ra, ra^c	read memory cell and feed result into signature analyzer, expected value is a, a^c
$(ra)^c$	read memory cell with expected value a and feed a^c into signature analyzer

Table 1: Basic definitions and notations.

To avoid this inefficiency while retaining the benefits of transparent BIST, this paper introduces the concept of symmetric transparent BIST. This new approach will allow to completely omit the overhead for signature prediction. Concerning the overall hardware cost and the achievable fault coverage we will show that it is comparable to conventional transparent BIST approaches. Experimental data will demonstrate that, in many cases, the fault coverage is even increased.

The rest of the paper is organized as follows: The basic principles and ideas for the proposed symmetric transparent BIST are explained in Section 2. Section 3 demonstrates the general applicability of the new approach, and finally Section 4 analyzes the fault coverage properties.

2 Exploiting symmetries for signature prediction

Most of the march test algorithms used for transparent RAM BIST produce test data with a high degree of symmetry. In the sequel, the classical MATS+ algorithm will again serve as an example to illustrate this statement and also to show how certain symmetries can be exploited for an efficient BIST implementation. Having a closer look at the example of Figure 1 reveals the following symmetric behavior:

During the first phase the data sequence d = (1, 1, 0, 1) is read from the memory and fed to the signature analyzer. During the second phase we obtain d' = (0, 1, 0, 0) which can be produced from d by reversing the component order and complementing the entries.

Combining such kind of symmetry with a carefully chosen scheme for output data compression allows to completely omit the signature prediction phase of conventional transparent BIST approaches. More precisely, we will show that switching between the regular feedback polynomial and its reciprocal polynomial during signature analysis results in a precomputable signature. For a more detailed explanation some formal definitions have to be introduced first.

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

Definition 2: Let $h(X) = h_kX^k + h_{k-1}X^{k-1} + \ldots + h_1X + h_0$ be a polynomial of degree k. Then $h^*(X) := X^k \cdot h(X^{-1}) = h_k + h_{k-1}X + \ldots + h_1X^{k-1} + h_0X^k$ denotes the reciprocal polynomial.

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

As shown in Figure 2 the reciprocal polynomial corresponds to a signature register with feedback taps in reverse order. Using it for signature analysis reverts the data compression by a signature analyzer based on the original polynomial in the sense of the following Theorem.

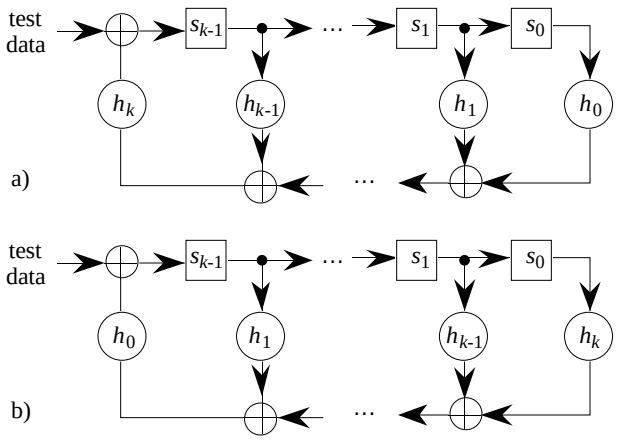


Figure 2: Serial signature analyzers with original (a) and reciprocal (b) feedback polynomial.

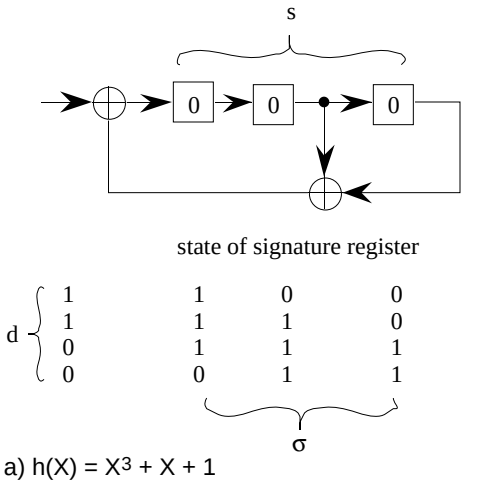
Theorem 1: Consider a serial signature analyzer with initial state s ∈ $\{0, 1\}^k$ and feedback polynomial h(X) ∈

GF(2)[X]. Furthermore let $d \in \{0, 1\}^n$ be a test data stream with corresponding signature $\sigma = sig(d, s, h)$. Then

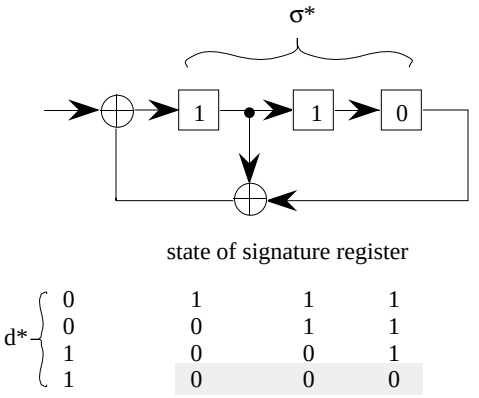
$$sig(\, d^*, sig(d, s, h)^*, h^*) = s^*,$$

i.e. signature analysis using h*(X) as feedback polynomial and the reversed signature σ* as initial state provides the reversed original state s* as signature for the reverse data stream d*.

Figure 3 illustrates the proposition of Theorem 1 for a small example. The general proof follows from elementary LFSR theory as sketched below.



a) $h(X) = X^3 + X + 1$



b) $h^*(X) = X^3 + X^2 + 1$

Figure 3: Signature analysis using the original and the reciprocal feedback polynomial.

Proof of Theorem 1: Without loss of generality we consider a test data stream consisting of just one bit d (the result for a data stream of length n follows by induction). In this case a signature analyzer with feedback polynomial $h(X) = h_k X^k + h_{k-1} X^{k-1} + \ldots + h_1 X + h_0$ and initial state $s = (s_{k-1}, \ldots, s_0)$ provides the signature

$$\sigma = \left(d + \sum_{i=0}^{k-1} s_i \cdot h_i, s_{k-1}, \ldots, s_1 \right).$$

Then

$$\sigma^* = \left(s_1, \ldots, s_{k-1}, d + \sum_{i=0}^{k-1} s_i \cdot h_i \right)$$

and d* = d. Since for a polynomial of degree k the highest coefficient h_k is 1, signature analysis based on h*(X) thus results in

$$\left(d + \left(d + \sum_{i=0}^{k-1} s_i \cdot h_i \right) \cdot h_k + \sum_{i=1}^{k-1} s_i \cdot h_i, s_1, \ldots, s_{k-1} \right)$$

$$= (s_0, s_1, \ldots, s_{k-1}) = s^*. \qquad \text{q.e.d.}$$

Theorem 1 is the key to construct efficient implementations for transparent RAM BIST. Figure 4 details the example of Figure 3 to show a BIST procedure for a test sequence of the type {⇑(ra, …); ⇓(ra, …)}.

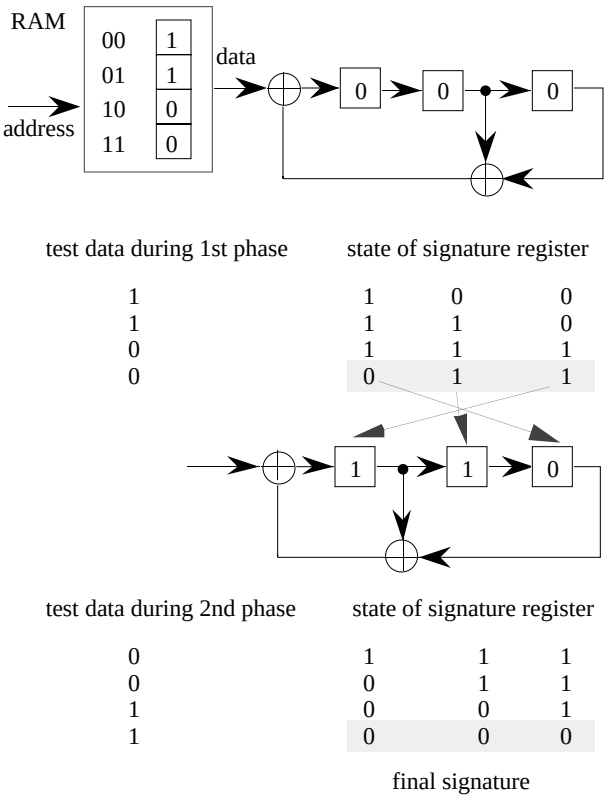


Figure 4: BIST for the test sequence {⇑(ra, …); ⇓(ra, …)}.

The feedback polynomial $h(X) = X^3 + X + 1$ is used to compact the test responses produced during the first phase ⇑(ra, …) of the test. During the second phase ⇓(ra, …) the memory is read in reverse addressing order, and the test data stream appears in reverse order. If the feedback polynomial is changed to $h^*(X) = X^3 + X^2 + 1$ and the contents of the signature analyzer is reloaded in reverse component

order, then the final signature will be zero according to Theorem 1 independent of the memory contents.

To be able to deal with the MATS+ algorithm Theorem 1 is extended as described by Theorem 2.

Theorem 2: Consider a serial signature analyzer with initial state $s \in \{0, 1\}^k$ and feedback polynomial $h(X) \in GF(2)[X]$. Furthermore let $d \in \{0, 1\}^n$ be a test data stream with corresponding signature $\sigma = sig(d, s, h)$. Then signature analysis using $h^*(X)$ as feedback polynomial and the reversed signature σ^* as initial state provides

$$sig(d^{*c}, sig(d, s, h)^*, h^*) =$$

$$s^* + sig((1, \ldots, 1), (0, \ldots, 0), h^*)$$

as signature for the complemented reverse data stream d^{*c}.

Proof: By definition $d^{*c} = (d_{n-1}^c, \ldots, d_0^c) = (d_{n-1} + 1, \ldots, d_0 + 1) = d^* + (1, \ldots, 1)$. According to the principle of superposition this yields:

$$sig(d^{*c}, sig(d, s, h)^*, h^*) =$$

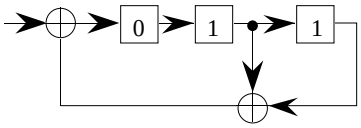
$$sig(d^* + (1, \ldots, 1), sig(d, s, h)^* + (0, \ldots, 0), h^*) =$$

$$sig(d^*, sig(d, s, h)^*, h^*) + sig((1, \ldots, 1), (0, \ldots, 0), h^*).$$

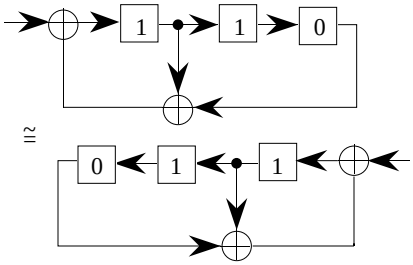
By Theorem 1 $sig(d^*, sig(d, s, h)^*, h^*) = s^*$, which completes the proof. q.e.d.

If we implement a transparent BIST based on the MATS+ algorithm analogously to the procedure described above for the test sequence $\{\Uparrow(ra, \ldots); \Downarrow(ra, \ldots)\}$, then we can precompute the reference signature with the help of Theorem 2 independently of the memory contents. The signature prediction phase of conventional transparent BIST schemes becomes completely superfluous, and the test time is reduced from 6n to 4n. As discussed in section 4 the fault coverage achievable with the new scheme is in many cases even higher than that guaranteed by the conventional approach. The additional hardware effort can be kept low, when the signature analyzer is implemented with flip-flops allowing both left and right shift (see Figure 5).

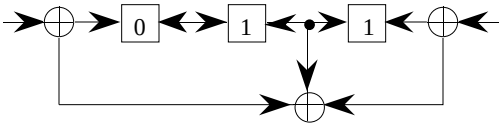
In this case, simply changing the direction of shifting corresponds to reloading the signature register with its reversed contents and to switching between the original and the reciprocal feedback polynomial. Test control has to incorporate this, but on the other hand the test control unit will be simplified, since it no longer has to handle an additional signature prediction phase. So the hardware overhead for the proposed technique mainly reduces to the extra cost for a bidirectional shift register and to an extra EXOR-gate ensuring that test data can be fed to the compactor from both directions.



a) $h(X) = X^3 + X + 1$



b) $h^*(X) = X^3 + X^2 + 1$



c) $h(X) = X^3 + X + 1$ combined with $h^*(X) = X^3 + X^2 + 1$

Figure 5: Combining original and reciprocal feedback polynomials.

3 Transforming transparent into symmetric algorithms

The proposed BIST schemes for the MATS+ algorithm and the test sequence $\{\Uparrow(ra, \ldots); \Downarrow(ra, \ldots)\}$ can be applied to 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 test data string D.

In general, it cannot be expected, that an arbitrary transparent march test is symmetric. However, typical transparent test algorithms contain symmetric subsequences and can be easily extended to fully symmetric versions. Consider the March C- algorithm as an example [10]. It is originally defined as

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

leading to the transparent version

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

The test data stream fed to the signature analyzer is not symmetric in the sense of Definition 4. The 4-bit memory of Figure 1 for example provides the sequence (1101,

0010, 1011, 0100, 1011) with a decreasing addressing order in the last phase. If this sequence is extended to (**0010**, 1101, 0010, 1011, 0100, 1011)), then it can be written as (d, d*^{c}) with the symmetry axis after the first 12 bits. Such an extended symmetric sequence is for example produced by the extended test

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

which guarantees at least the same fault coverage as the original test, but is suitable for an implementation as proposed in section 2. Although the extension will increase the test time from 9n to 10n, there is still a considerable gain in efficiency compared to the conventional 14n approach with signature prediction.

Similarly, for any of the known transparent algorithms it is possible to identify a potential symmetry axis and to add some additional read sequences to obtain symmetric versions of the algorithms. Table 2 shows the resulting test lengths for some commonly used transparent march tests [10].

Algorithm	**Transparent BIST**			**Symmetric Transparent BIST**
	Signature Prediction	**Test**	**Total Time**	**Total 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

Table 2: Comparison of test times for transparent and symmetric transparent BIST.

It can be observed that in all cases the symmetric versions of the considered transparent algorithms require a considerably shorter test time than the original versions.

4 Fault coverage issues

The properties of the proposed BIST technique with respect to error masking and fault coverage are summarized in the following observations

Observation 1: Faults which manifest themselves only in a subsequence of the test data stream which is compacted without changing the feedback polynomial have the same probability of fault masking as with conventional signature analysis.

To clarify Observation 1 assume a (d, d*^{c})-type symmetric transparent test based on a primitive feedback polynomial h(X) and an initial state s. Assume that faults lead to erroneous bits in the first part of the test data stream (d, d*^{c}) only. Then the erroneous data stream can be written as (d+e, d*^{c}) for some error vector e. By the principle of superposition, the signature after the first phase is *sig*(d, s, h) + *sig*(e, 0, h). This yields the final signature *sig*(d*^{c}, *sig*(d, s, h)*, h*) + *sig*(0, *sig*(e, 0, h)*, h*). Because of the constant zero input data stream, *sig*(0, *sig*(e, 0, h)*, h*) is obtained as the final state of an autonomous LFSR with initial state *sig*(e, 0, h)*. Error masking can only occur, if *sig*(e, 0, h)* already provides the expected value *sig*(0, 0, h)* = 0. Consequently, in the situation of Observation 1, the probability of error masking for a test data sequence of length 2n and a signature analyzer of degree k is the same as the probability of error masking for test data sequence of length n using conventional signature analysis.

For example all single stuck-at faults only change one of the data streams d or d*^{c} in a (d, d*^{c})-type symmetry and can therefore be detected without any additional risk of fault masking. Thus, problems can only arise with faults leading to erroneous data both in d and d*^{c}.

Observation 2: Faults which manifest themselves in multiple errors, such that the symmetry of the test data stream is preserved cannot be detected by the proposed scheme.

In practice, faults preserving the symmetry of the test data stream are very rare. Table 3 shows some experimental data for a 32 Kbit memory and several symmetric transparent tests.

	Fault Model			
Algorithm	**SAF**	**TF**	**CFid**	**CFin**
March C				
Transparent	99.992	99.991	99.995	99.996
Symmetric	100	100	100	100
March C-				
Transparent	99.992	99.991	99.996	99.997
Symmetric	100	100	100	100
March X				
Transparent	100	100	49.993	99.992
Symmetric	100	100	49.996	99.997

Table 3: Simulation results (fault coverage in % for single faults) for a 32 Kbit memory.

For a discussion of the investigated fault models (SAF: stuck-at faults, TF: transition faults, CFid: idempotent coupling faults, CFin: inversion coupling faults) see for example [10] or [11].

As a matter of fact it can be observed that in all cases the new symmetric approach achieves a slightly higher fault coverage than a conventional transparent BIST.

5 Conclusions

A new approach for transparent RAM BIST has been proposed which exploits symmetries in the test algorithms to implement schemes for output data compression with precomputable signatures. Compared to traditional transparent BIST schemes this new approach significantly reduces the test time while preserving the benefits of previous approaches with respect to hardware overhead and fault coverage. Experimental studies even show an increase in fault coverage in many cases.

6 Acknowledgement

The authors would like to thank Yuri Klimets for his major contribution to performing the simulations.

7 References

1 V. C. Alves, M. Nicolaidis, P. Lestrat, and B. Courtois: Built-in Self-Test for Multi-Port RAMs; Proc. IEEE Int. Conf. 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 a Telecommunication System; Proc. IEEE Int. Test Conf., Washington, DC, Oct. 1995, pp. 947-953.

3 P. H. Bardell, W. H. McAnney, and J. Savir: Built-In Test for VLSI: Pseudorandom Techniques; New York: John Wiley & Sons, 1987.

4 D. K. Bhavsar, J. H. Edmondson: Alpha 21164 Testability Strategy; IEEE Design&Test, Vol. 14, No. 1, January-March 1997, pp. 25-33

5 H. Cheung, S. K. Gupta: A BIST Methodology for Comprehensive Testing of RAM with Reduced Heat Dissipation; Proc. IEEE Int. Test Conf., Washington, DC, Oct. 1996, pp. 386-395.

6 B. Cockburn, Y.-F. N. Sat: Synthesized Transparent BIST for Detecting Scrambled Pattern-Sensitive Faults in RAMs; Proc. IEEE Int. Test Conf., Washington, DC, Oct. 1995, pp. 23-32.

7 R. David, A. Fuentes, and B. Courtois: Random Pattern Testing Versus Deterministic Testing of RAMs; IEEE Trans. on Computers, Vol. C-38, No. 5, May 1989, pp. 637-650

8 R. Dekker, F. Beenker, and L. Thijssen: Realistic Built-In Self-Test for Static RAMs; IEEE Design & Test of Computers, Vol. 6, No. 1, Feb. 1989, pp. 26-34.

9 R. Dekker, F. Beenker, and L. Thijssen: A Realistic Fault Model and Test Algorithms for Static Random Access Memories; IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 9, No. 6, June 1990, pp. 567-572

10 A. J. Van de Goor: Testing Semiconductor Memories, Theory and Practice; Chichester: John Wiley & Sons, 1991.

11 A. J. Van de Goor: Using March Tests to Test SRAMs; IEEE Design & Test of Computers, Vol. 10, No. 1, March 1993, pp. 8-14

12 K. Kinoshita, K. K. Saluja: Built-In Testing of Memory Using an On-Chip Compact Testing Scheme; IEEE Trans. on Computers, Vol. C-35, No. 10, October 1986, pp. 862-870.

13 K. T. Le, K. K. Saluja: A Novel Approach for Testing Memories Using a Built-In Self-Testing Technique, Proc. IEEE International Test Conference, Washington, DC, 1986, pp. 830-839.

14 M. E. Levitt: Designing UltraSparc for Testability; IEEE Design&Test of Computers, Vol. 14, No. 1, January-March 1997, pp. 10-17

15 M. Marinescu: Simple and Efficient Algorithms for Functional RAM Testing; Proc. IEEE Int. Test Conf., 1982, pp. 236-239

16 R. Nair: Comments on an Optimal Algorithm for Testing Stuck-at Faults in Random Access Memories; IEEE Trans. on Computers, Vol. C-28, No. 3, 1979, pp. 258-261

17 M. Nicolaidis: Transparent BIST for RAMs; Proc. IEEE Int. Test Conf., Baltimore, MD, Oct. 1992, pp. 598-607.

18 P. Olivo, M. Dalpasso: Self-Learning Signature Analysis for Non-Volatile Memory Testing; Proc. IEEE Int. Test Conf., Washington, DC, Oct. 1996, pp. 303-308.

19 I. M. Ratiu, H. B. Bakoglu: Pseudo-random built-in self-test methodology and implementation for the IBM RISC System/6000 processor; IBM Journal of Research and Development, Vol. 34, No. 1, 1990, pp. 78-84

20 N. Sakashita et al.: A Built-in Self-Test Circuit with Timing Margin Test Function in a 1Gbit Synchronous DRAM; Proc. IEEE Int. Test Conf., Washington, DC, Oct. 1996, pp. 319-324.

21 D. S. Suk, S. M. Reddy: A March Test for Functional Faults in Semiconductor Random-Access Memories; IEEE Trans. on Computers, Vol. C-30, No. 12, 1981, pp. 982-985

22 V. N. Yarmolik, S. Hellebrand, H.-J. Wunderlich: Self-Adjusting Output Data Compression: An Efficient BIST Technique for RAMs; Proc. Design and Test in Europe (DATE'98), Paris, February 1998, pp. 173-179