International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

High Speed Symmetric Transparent BIST on FPGA

Srikanth Lammatha, Prof. A. S. Srinivasa Rao, E. Jaya

Abstract **- Symmetric Transparent BIST has been proposed as a means to skip the signature prediction phase during RAM testing thereby testing time can be reduced. The proposed symmetric transparent BIST scheme is for single bit and multiple bit (word) RAM's. The principle of this is a characteristic of a polynomial is modified based on the application of sequence of March events. In this paper an FSM is used to generate the mach events and are applied for the RAM to verify it. Compared to previous BIST techniques, controller complexity is reduced. Due to the reduction in the controller complexity the hardware required for the BIST also reduces and thereby reducing the power and delay.**

Keywords **– Built in self test; design for testability; integrated circuit reliability; Self-testing; RAM testing.**

I. INTRODUCTION

Advances in the memory technology tend to make memory devices more and more complicated. The sizes of these memories are decreasing with increase in technology. These memories are tested after the fabrication and also to be tested in the field repeatedly. Testing these memories in the field is not an easy task for the user. If a failure occurs as some point in the field, user may not distinguish and a chance of erroneous outputs during read/write operations. For the testing these memories, a March algorithm is employed for effective and to minimize the verification time. March algorithms impose a series of march events that perform a predetermined sequence of Read/Write operations in every cell for bit organized RAM or Word for Word organized RAM's [8].

II. MARCH ALGORITHMS

A. Traditional March Algorithms

A March algorithm consists of n march elements, denoted by M_i, with 0<=i<n. Each March element comprises zero (or more) write operations, denoted byw_0/w_1 meaning that 0/1 is written to the RAM cell, and zero (or more) read operations denoted byr_0/r_1, meaning that 0/1 is expected to be read from the memory cell. For example, the C algorithm [see Fig. 1(a)] consists of six march elements denoted byM0 to M5. In Fig. 1, denotes an increasing addressing order (which can be any arbitrary addressing order) and denotes a decreasing addressing order (which is the inverse addressing order of).

B. Transparent BIST Algorithms

Traditional march algorithms erase the memory contents prior to testing; therefore, they do not serve as good platforms for periodic BIST. Nikolaidis [2] proposed the concept of transparent BIST where the initial w_0 phase is bypassed, and a signature prediction phase is used instead. The signature prediction phase consists of read operations equivalent to all the read operations of the March algorithm and it is utilized in order to calculate a signature that will be compared against the compacted signature calculated during the (remaining) march test. The transparent version of the C-algorithm is shown in Fig. 1(b). The notation for the transparent versions of the algorithms differs slightly from the one used in traditional march algorithms. Instead of r_0, r_1, w_0, w_1 the notations r_a, r_a^c, w_a, w_a^c and $(r_a)^c$ are utilized. Their meanings are as follows.

r_a - Read the contents of a word of the RAM, expecting to read the initial contents of the RAM word (i.e., before the beginning of the test).

r_a^c - Read the contents of a word of the RAM, expecting to read the complement of the initial contents of the RAM word.

$(r_a)^c$ - Read the contents of a word of the RAM expecting to read the initial word contents and feed the complement value to the compactor.

w_a - Write to the memory word; the value that was stored in this memory word at the beginning of the test is (assumed to be) written to the word.

w_a^c - Write to the memory word; the inverse of the value that was stored in this memory word at the beginning of the test is (assumed to be) written to the word.

By default, the data driven to the compactor with the $(r_a)^c$ operation are identical to the data driven by the r_a^c . The importance of the $(r_a)^c$ operation is the following: during the signature prediction phase the contents of the RAM are equal to the initial contents (since no write operation has been performed); therefore, in order to simulate the r_a^c operation we invert these contents prior to driving them to the compactor.

It has been shown that, with the transparent BIST algorithms, the contents of the memory at the end of the test are identical to those before the test. Also, since the read elements of the signature prediction phase (M0) are identical to the read elements of the testing phase (M1-M5), then if we store the result of the compaction of M0 and compare it to the result of the compaction ofM1-M5, then we can detect faults that occur due to the write operations of the March algorithm.

Traditional transparent BIST [1] has the disadvantage that the signature prediction phase adds up to the total testing time with a percentage of (more than) 30% [8]. In order to confront this problem, Yarmolik *et al.* introduced the concept of symmetric transparent BIST, which is explained in the next subsection.

C. Symmetric Transparent BIST

In order to define a symmetric transparent algorithm, some notations will be introduced first. Let $d=\{d_0,d_1,\text{---}d_{n-1}\}\in\{0,1\}^n$ be a data stream, then $d^*=\{d_{n-1},d_{n-2},\text{---},d_1,d_0\}$denotes the data stream with components in reverse order and $dc=\{d_0^c\text{-----}d_{n-1}^c\}$ denotes the data stream with inverted components. For example, if $d=\{1011\}$,$d^*=\{1101\}$ and $d^c=\{0100\}$.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

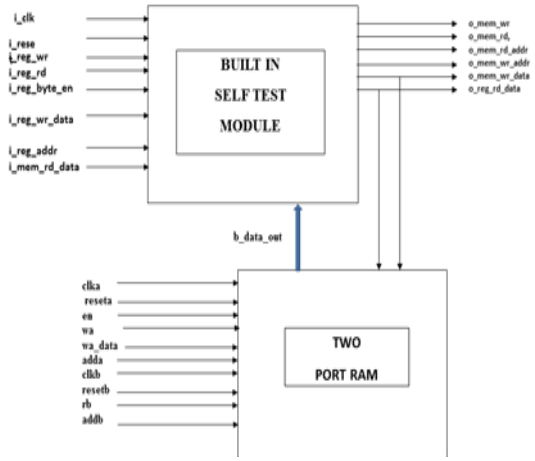
A data string D € $\{0,1\}^{2n}$ is called ***symmetric***, if there exists a data string d € $\{0,1\}^{n}$ with D={d, d*} or D={d, d^{*c}}. For example, D1= {1010 0101} andD2= {1010 1010} [11] are symmetric data strings, since {0101}={1010}* and{1010}={1010}^{*c}. Furthermore, a *transparent march test is called symmetric* if it produces a symmetric test data string.

In order to derive a symmetric transparent algorithm, the March series is modified in such way that the expected output response is equal to a known value. Therefore, the signature prediction phase can be skipped and the time required for the test is reduced.

In order to achieve this, Yarmolik *et al.* noticed that most of the March algorithms used for transparent BIST produce test data with a high degree of symmetry [3]. For example, the read elements of the transparent C- march algorithms [see Fig. 1(b)], ignoring the signature prediction phase (and the write elements) are: (r_a), (r_a^c), (r_a), (r_a^c), (r_a).It is easy to detect the approximate symmetry; furthermore, it is also easy to derive a symmetric sequence by adding an additional read element, resulting in the following sequence of read elements: ((r_a)^c); (r_a); (r_a^c); (r_a); (r_a^c); (r_a). For example, for a bit-organized memory with five words whose initial contents are (11010), the result of the latter sequence is (00101 11010 00101 ! 01011 10100 01011) which is easily shown to be symmetric with respect to the given definition. Yarmolik *et al.* have shown that by exploiting the previously mentioned symmetry and by using linear structures as compactors for the outputs of the RAM [21], the final value of the compactor is equal to a known value, i.e., the all-zero value. For the case of bit-organized memories, SISRs were utilized, while for word-organized memories MISRs were exploited. It was proven that by toggling between a primitive polynomial and its reciprocal one during the r and r operations, the final signature is equal to the all-zero state. They even reported marginal increase in the fault coverage of the symmetric schemes compared to the respective transparent ones with signature prediction. For example, in Fig. 1(c) the symmetric transparent version of the C- algorithm is presented.

M_0.	⇑(w_0);	⇑ (r_a); ⇑ ((r_a)^c); ⇓ (r_a); ⇓ ((r_a)^c); ⇓ (r_a);	⇑ ((r_a)^c);
M_1.	⇑ (r_0,w_1);	⇑ (r_a,w_a);	⇑ (r_a,w_a^c);
M_2.	⇑ (r_1,w_0);	⇑ (r_ac,w_a);	⇑ (r_ac,w_a);
M_3.	⇓ (r_0,w_1);	⇓ (r_a,w_a^c);	⇓ (r_a,w_a^c);
M_4.	⇓ (r_1,w_0);	⇓ (r_ac,w_a);	⇓ (r_a^c,w_a);
M_5.	⇓ (r_0);	⇓ (r_a);	⇓ (r_a);
	(A)	(B)	(C)

5. Proposed architecture for STBIST



The accumulator-based response compaction scheme proposed in this paper stems from the following two observations.

1) Observation 1: If the March algorithm is symmetric (as in the case of symmetric transparent BIST) then the number of r_a elements equals the number of r_a^c elements plus the number of (r_a) ^{c}) elements (without taking into account the addressing order, of the March element).

2) Observation 2*:* The accumulator-based compaction of the responses holds the *order-independent* property (i.e., the final signature is independent of the order of the incoming vectors. Observation 2 stems directly from the per-mutational property of the addition operation. Accumulator-based compaction for symmetric transparent BIST for the case of word-organized memories is based on Lemma 1.

Lemma 1: If a symmetric transparent march algorithm is applied to a word-organized memory whose word length is n and the responses are captured in an n-stage accumulator comprising a 1's complement adder (starting from the all-0 state), then the final content of the accumulator is equal to the all-1 state.

III. MATHEMATICAL PROOF

Let M be the number of elements of the march algorithm; since the algorithm is symmetric, the total number of r_a elements is equal to the total number of r_a (plus the number of (r_a)^c) elements. Therefore, for every vector **a** driven to the inputs of the accumulator, its complement $\mathbf{a}^c$ is also driven to the inputs of the accumulator exactly once. But

$$\mathbf{a} + \mathbf{a}^c = 2^{n-1}$$

Furthermore, for 1's complement addition it holds that the sum of two numbers A and B is given by (A-1+B (mod 2^n-1))+1, therefore, the sum of 2^n-1 and 2^n-1 is (2^n-1-1+2^n-1) mod (2n-1)+1=(2^n-2)+1=2^n-1. Therefore, it is trivial to show (by induction) that (2) holds for any value of

$$\sum \mathbf{i} * (2^n\text{-}1) = 2^n\text{-}1$$

From (1) and (2), and taking into account that the addition operation is per mutative (Observation 2), we have the proof.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

For example, let us consider the 4-word 3-bit RAM presented in Fig. 2(a). The outputs of the memory are driven to an n=3-stage accumulator comprising a 1's complement adder, Fig. 2(b). For the implementation of the $(r_a)^c$ march element, the subtraction operation of the accumulator can be utilized. In order to apply march elements of the form (r_a,w_a^c) or (r_a^c,w_a) the output of the RAM must be inverted and then fed back to its Inputs; with the proposed scheme, this can be one by forcing the all-1 vector to one input of the adder/subtractor and perform a subtract operation. This is done with the OR gates whose one input is driven by the inv signal in Fig. 2. Therefore, the inverse of the read vector appears at the outputs the adder/subtractor and applied to the RAM inputs.

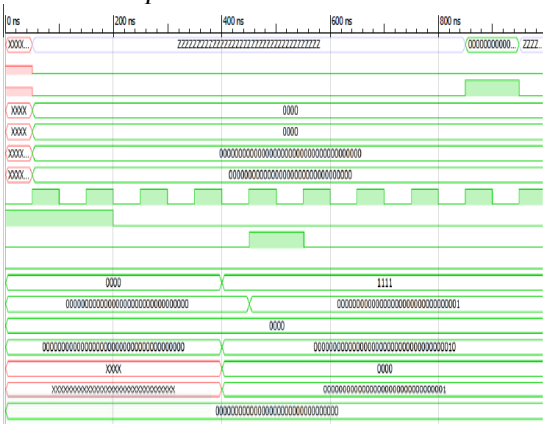
Taking into account the results related to transparent memory testing it should be concluded that the most frequently used test algorithms for transparent RAM BIST are march tests which combine a high fault coverage with an acceptable test time even for memories of the large sizes (1M and higher). Another advantage of these algorithms is that they can be easily extended to transparent versions. 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.

IV. Synthesis and Simultion Results

a. Synthesis report

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	191	3,840	4%
Number of 4 input LUTs	256	3,840	6%
Number of occupied Slices	175	1,920	9%
Number of Slices containing only related logic	175	175	100%
Number of Slices containing unrelated logic	0	175	0%
Total Number of 4 input LUTs	274	3,840	7%
Number used as logic	256		
Number used as a route-thru	18		
Number of bonded IOBs	123	173	71%
Number of BUFGMUXs	1	8	12%
Average Fanout of Non-Clock Nets	3.65		

b. Simulation report



c. Power report

V. Conclusion

Periodic testing of small-and medium size RAMs existing into current VLSI DSP and processor chip requires that the contents of the RAM are not rubbed-out, in order for the system to be able to continue its normal operation after test completion. Transparent BIST proposed by Nikoladius meets this requirement, however testing tie is increased, compared to traditional matching tests due to the requirement for a signature prediction phase. Symmetric transparent BIST proposed by yarmolik et al skips the signature prediction phase. The proposed scheme presents lower hardware overhead and requires less complicated control compared to the schemes proposed in [8],[9],therefore may prove variable solution for periodic testing of RAMs embedded into current VLSI chips.

References

[1] M. Nicolai is, "Theory of transparent BIST for RAMs," *IEEE Trans.Comput.*, vol. 45, no. 10, pp. 1141–1156, Oct. 1996.

[2] M. Nicolaidis, "An efficient built-in self test for functional test of embedded RAMs," in *Proc. 15th Symp. Fault Tolerant Comput.*, Jun. 1985,pp. 118–123.

[3] A. Castro, M. Nicolaidis, P. Lestrat, and B. Courtois, "Built-in self test for multi-port RAMs," presented at the ICCAD, Santa Clara, CA, Nov. 1991.

[4] A. J. van de Goor, *Testing Semiconductor Memories, Theory and Practice*. Chichester, U.K.: Wiley, 1991.

[5] A. J. van de Goor, "Using march tests to test SRAMs," *IEEE Des. Test Comput.*, vol. 10, no. 1, pp. 8–14, 1993.

[6] A. J. van de Goor and C. A. Verruijt, "An overview of deterministic functional RAM chip testing," *ACM Comput. Surveys*, vol. 22, no. 1, pp. 5–33, Mar. 1990.

[7] M. Marinescu, "Simple and efficient algorithms for functional RAM testing," in *Proc. IEEE Int. Test Conf.*, 1982, pp. 236–239.

[8] V. N. Yarmolik, S. Hellebrand, and H.-J. Wunderlich, "Symmetric transparent BIST for RAMs," presented at the DATE, Munich, Germany, Mar. 1999.

[9] V. N. Yarmolik, I. V. Bykov, S. Hellebrand, and H.-J. Wunderlich, "Transparent word-oriented memory BIST based on symmetric march algorithms," in *Proc. Eur. Dependable Comput. Conf.*, 1999, pp.339–350.

[10] I. Voyiatzis, "Test vector embedding into accumulator-generated sequences: A linear-time solution," *IEEE Trans. Comput.*, vol. 54, no. 4, pp. 476–484, Apr. 2005.
[11] A. Stroele, "BIST patter generators using addition and subtraction operations," *J. Electron. Test.: Theory Appl.*, vol. 11, pp. 69–80, 1997.
[12] R. Dorsch and H.-J. Wunderlich, "Accumulator-based deterministic BIST," in *Proc. Int. Test Conf.*, 1998, pp. 412–421.
[13] I. Voyiatzis, A. Paschalis, D. Gizopoulos, N. Kranitis, and C. Halatsis, "A concurrent built-in self-test architecture based on a self-testing RAM," *IEEE Trans. Reliab.*, vol. 54, no. 1, pp. 69–78,Mar. 2005.
[14] W. L.Wang, K. J. Lee, and J. F.Wang, "An on-chip march pattern generator for testing embedded memory cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 5, pp. 730–735, Oct. 2001.
[15] C.-T. Huang, J.-R. Huang, C.-F. Wu, C.-W. Wu, and T.-Y. Chang, "A programmable BIST core for embedded DRAM," *IEEE Des. Test Comput.*, vol. 16, no. 1, pp. 59–70, Jan./Mar. 1999.
[16] J. -F. Li, R.-S. Tzeng, and C.-W. Wu, "Diagnostic data compression techniques for embedded memories with built-in self-test," *J. Electron. Test.: Theory Appl.*, vol. 18, pp. 515–527, 2002.
[17] S. Hamdioui and J. E. Q. D. Reyes, "New data-background sequences and their industrial evaluation for word-oriented random-access memories," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 6, pp. 892–904, Jun. 2005.
[18] S. Hamdioui, Z. Al-Ars, and A. J. van de Goor, "Opens and delay faults in CMOS RAM address decoders," *IEEE Trans. Comput.*, vol. 55, no.12, pp. 1630–1639, Dec. 2006.
[19] W.-L. Wang, K.-J. Lee, and J.-F. Wang, "An on-chip march pattern generator for testing embedded memory cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 5, pp. 730–735, Oct. 2001.
[20] D.-C. Huang and W.-B. Jone, "A parallel built-in self-diagnostic method for embedded memory arrays," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 4, pp. 449–465, Apr. 2002.
[21] B. H. Fang and N. Nicolici, "Power-constrained embedded memory BIST architecture," in *Proc. 18th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst. (DFT)*, 2003, pp. 451–458.

AUTHOR'S PROFILE

Srikanth Lammatha
Completed his B. Tech. degree from Aditya Institute of Technology and Management, Tekkali in 2009.Now he is studying M.Tech degree with VLSI-SD specialization in Aditya Institute of Technology and Management, Tekkali .His interested areas,VLSI system design.
Email id: lammathasrikanth@gmail.com

A. S. Srinivasa Rao
completed his M. Tech. degree from Andhra University, Vishakapatnam in 2004 and submitted his thesis for Ph. D. in Andhra University. He is having 14 years of teaching experience in various engineering colleges. Presently he is working as Professor, Dept. of ECE, Aditya Institute of Technology and Management, Tekkali. His interested areas, Signal processing, Adaptive antenna arrays and Communication systems. He has published 11 papers in various journals/proceedings in the field of Antenna Arrays.
Email id : asr47@rediffmail.com

E. Jaya
Completed her M. Tech. degree from Aditya Institute of Technology and Management, Tekkali in 2011.Now she is working as Asst.professor in Aditya Institute of Technology and Management,Tekkali .Her interested areas, VLSI, Signal processing
Email id: jaya.baratam@gmail.com