Self-Adjusting Output Data Compression: An Efficient BIST Technique 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

After write operations, BIST schemes for RAMs relying on signature analysis must compress the entire memory contents to update the reference signature. This paper introduces a new scheme for output data compression which avoids this overhead while retaining the benefits of signature analysis. The proposed technique is based on a new memory characteristic derived as the modulo-2 sum of all addresses pointing to non-zero cells. This characteristic can be adjusted concurrently with write operations by simple EXOR-operations on the initial characteristic and on the addresses affected by the change.

1 Introduction

High density memories constitute an integral part of most present day micro-electronic systems. To guarantee reliable storage of data, particularly in systems for safety critical applications, efficient test procedures are indispensable. Besides production testing, periodic maintenance testing is especially important for memories to deal with the problem of data retention, e. g. checking for data losses after power-off conditions or during run-time. Moreover, frequent operations on the memory contents require on-line or periodic consistency checks. When memories are embedded in larger integrated circuits, such as microprocessors or digital communication devices, conventional techniques for external memory testing can no longer be applied due to the limited accessibility of the memories. To solve this problem, and also to reduce the long testing times inherent to conventional external testing, a number of theoretical and practical built-in self-test (BIST) solutions have been proposed in the past [1-5, 7, 10-13, 16, 18].

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

This paper addresses the problem of periodic consistency checking for embedded memories. The proposed strategy is based on the BIST architecture sketched in Figure 1. The test pattern generator produces a predetermined sequence of memory addresses, and the corresponding sequence of data is fed into the output data compressor, the final state of which represents a characteristic of the memory contents. For the initial correct memory contents a reference characteristic C_{REF} is „learned“ this way on chip and saved in a specific location [14]. The same procedure is repeated periodically, and the respective characteristics C_{TEST} are compared to C_{REF} to reveal inconsistencies. The test pattern generator can also be used to generate the data to be written into the memory for the production test. In this mode standard algorithms may be applied which are not considered within the framework of this paper [6, 8].

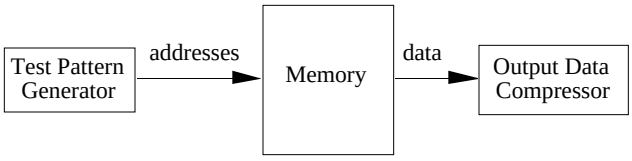

Figure 1: Block diagram of self-testing memory.

Since the described technique checks the memory off-line, there is a certain latency of error detection compared to conventional on-line checking schemes [15]. But this is more than outweighed by a much smaller area overhead and a simpler design. In [14] a checking scheme based on the architecture of Figure 1 has been proposed relying on signature analysis for output data compression. It has been demonstrated by the authors that this kind of BIST architecture offers some very important advantages, whereof the most valuable are: low area overhead, rather unaffected access time during normal operation, and high fault coverage due to the low probability of aliasing errors [3, 9]. The main disadvantage, however, is the necessity to

adjust the learned signature after write operations. Updating the learned signature requires to run through a complete cycle of the test pattern generator and thus repeating the complete initial learning phase.

In this paper, we propose a new scheme for output data compression which avoids this overhead while retaining the advantages of schemes relying on learned signatures. It is based on a new characterization of the fault free memory contents and can easily be adjusted during write operations, which eliminates the repeated learning phases of previous approaches and significantly reduces the test time. It will be shown that the scheme is particularly suitable for DRAMs, although it can be efficiently implemented for various types of memory. The achievable test quality is the same as for schemes using conventional signature analysis: the probability of aliasing errors is 2^{-k}, where k denotes the length of the memory characteristic. The basic concepts of the proposed technique are introduced in Section 2 for bit-oriented as well as for word-oriented RAMs, a complete BIST architecture is described in Section 3.

2 A Self-Adjusting Characteristic for RAMs

In the following a characteristic for fault free RAMs is introduced which can easily be adjusted concurrently with write operations. The updated characteristic is obtained as a combinational function of the initial characteristic and the addresses affected by the change. The basic principles of the proposed approach will be explained in Section 2.1 for bit-oriented RAMs, the generalization to word-oriented memories is described in Section 2.2.

2.1 Bit-Oriented Memories

As illustrated in Figure 2, the proposed reference characteristic C_{REF} of the correct initial memory contents is defined as the modulo-2 sum of all addresses pointing to cells containing a „1“, and it will therefore be called „modulo-2 address characteristic“ throughout the paper.

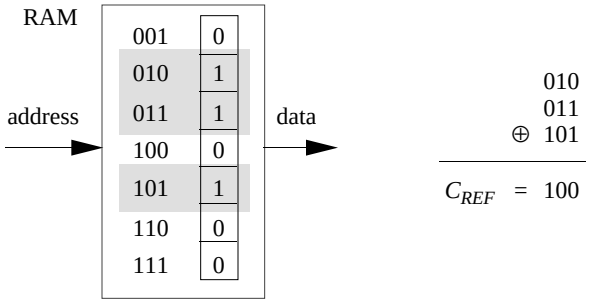

Figure 2: Modulo-2 address characteristic for bit-oriented RAMs.

As shown in Figure 3, during BIST a test pattern generator (an LFSR with primitive feedback polynomial or a counter, e.g.) produces all possible addresses in an arbitrary order. The output data compressor determines the modulo-2 sum C_{TEST} of all addresses corresponding to „1“, and if $C_{REF} = C_{TEST}$ holds, the memory is considered as fault free.

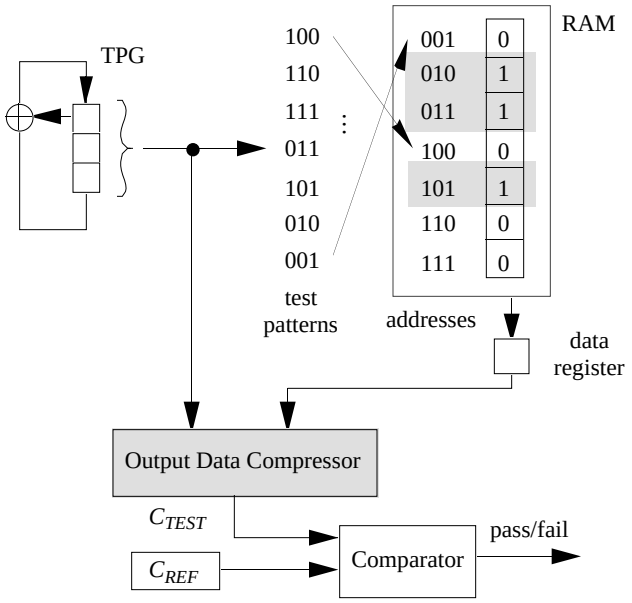

Figure 3: Memory BIST based on the modulo-2 address characteristic.

The compressor circuit, which is shown in more detail in Figure 4, simply has to perform bitwise EXOR-operations on addresses controlled by the value stored in the data register.

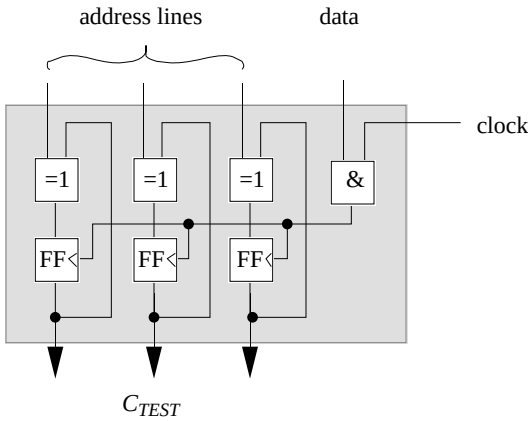

Figure 4: Output data compressor based on the modulo-2 address characteristic.

In case of write operations, the reference characteristic must be adjusted to the new characteristic C_{REF}^{new}, which is obtained by modulo-2 addition of all addresses with

changes from „0“ to „1“ and modulo-2 subtraction (which is equivalent to modulo-2 addition) of all addresses with changes from „1“ to „0“. More specifically, altering the contents of a single cell corresponds to the modulo-2 addition of the cell address to the initial characteristic C_{REF}. In particular, this implies that the compressor circuit of Figure 4 is also capable of adjusting the reference characteristic during write operations in system mode (a detailed description of the complete RAM implementation with BIST will be given in Section 3).

To summarize the basic concepts of the proposed approach more formally let the memory be represented by an array $M[1, \ldots, m]$ with address space $A = \{1, \ldots, m\}$, and let $A_1 := \{a \in A \mid M[a] = \text{„1“}\}$ denote the set of all memory addresses pointing to „1“ entries. With

$$\bigoplus_{a \in A_1} a$$

representing the bitwise modulo-2 sum of the binary representation of all addresses in A_1, the initial characteristic is obtained as

$$C_{REF} = \bigoplus_{a \in A_1} a,$$

and the new reference characteristic after a write operation at a specific address a^* is given by

$$C_{REF}^{new} = C_{REF}^{old} \oplus a^* \cdot \left(M[a^*]^{new} \oplus M[a^*]^{old} \right).$$

Besides the simple mechanism of mapping changes in memory to changes in the characteristic, the proposed scheme has a number of additional advantageous properties:

1) All single errors are detectable and diagnosable, since if only single errors are assumed, the expression $C_{REF} \oplus C_{TEST}$ provides the address of the faulty memory cell.
2) All double errors are detectable, since in this case $C_{REF} \oplus C_{TEST}$ corresponds to the sum of two addresses a_r and a_s, and $a_r \neq a_s$ implies $C_{REF} \oplus C_{TEST} \neq 0$.
3) As shown below, output data compression based on the modulo-2 address characteristic is equivalent to serial signature analysis and the probability of aliasing errors is thus estimated by 2^{-k}, where k denotes the length of the characteristic.

Property 3 is an immediate consequence of the following observation.

Observation: Let $\varphi(X) \in$ GF(2)[X] be a primitive polynomial of degree k, and let

$$\varphi^{-1}(X) := X^k \varphi\left(\frac{1}{X}\right)$$

denote the reciprocal polynomial. An LFSR with feedback polynomial $\varphi^{-1}(X)$ and initial state (1, 0, …, 0) generates the same state transition sequence (in reverse component order) as the LFSR with feedback polynomial $\varphi(X)$ „counting“ backward from (0, …, 0, 1).

The example shown in Figure 5 exploits this observation to verify property 3 for a 7-bit RAM. A conventional BIST is implemented using a 3-bit LFSR with primitive feedback polynomial $\varphi(X) = 1 + X + X^3$ as test pattern generator and a serial signature analyzer with the reciprocal feedback polynomial $\varphi^{-1}(X) = 1 + X^2 + X^3$.

With an all-zero initial state the signature register does not change its contents before the first cell containing a „1“ is addressed. The new contents is (1, 0, 0), and as the remaining memory cells only contain „0“ entries, the signature analyzer works like an autonomous LFSR with initial state (1, 0, 0) for the rest of the test procedure. Since $\varphi^{-1}(X)$ is the reciprocal of $\varphi(X)$, this implies that the signature analyzer basically behaves like the test pattern generator counting backward from (0, 0, 1) (with reversed component order), thus the final signature is (1, 1, 0) and corresponds exactly to the address of the memory cell with a „1“ entry.

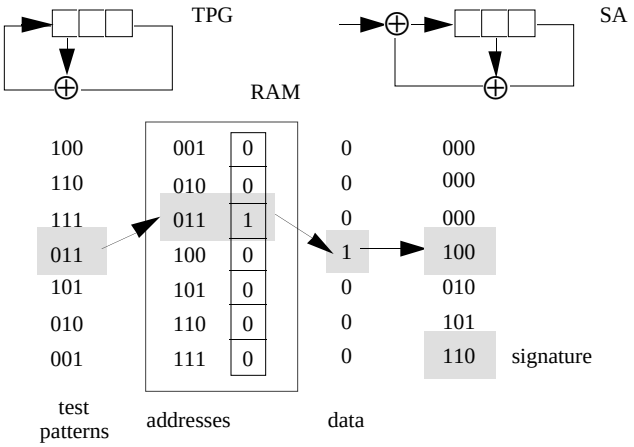

Figure 5: Correspondence between signature analysis and modulo-2 address characteristic.

If the RAM contains more than one non-zero entries, then similarly the signature is obtained as modulo-2 sum of all addresses (in reversed component order) corresponding to memory cells with contents „1“.

In general the correspondence between the modulo-2 address characteristic and signature analysis is described by the following theorem which was proven in [9, 17].

Theorem 1: Let M be a bit-oriented memory with $m = 2^k$-1 cells, $\varphi(X) \in$ GF(2)[X] a primitive polynomial of degree k, and let $A_1 \subset \text{GF}(2)^k \setminus \{0\}$ contain the memory

addresses pointing to „1“ entries. Furthermore, for $a = (a_0, \ldots, a_{k-1}) \in GF(2)^k$ let $a^r := (a_{k-1}, \ldots, a_0)$ denote the vector with components in reversed order. Then a BIST

- using a test pattern generator with feedback polynomial $\varphi(X)$,
- a serial signature analyzer with feedback polynomial $\varphi^{-1}(X)$,
- initial states (1, 0, …, 0) and (0, …, 0) for the test pattern generator and the signature analyzer, respectively,
- and a test length of m

is characterized by the fault free signature

$$S = \bigoplus_{a \in A_1} a^r .$$

The theorem remains true, when the number of memory cells is $m < 2^k$-1, and the initial state of the test pattern generator is selected, such that the final state is (0, …, 0, 1). This implies that for any memory BIST based on the modulo-2 address characteristic there exists an equivalent BIST configuration based on signature analysis with a primitive feedback polynomial, and consequently the same test quality is guaranteed.

In contrast to signature analysis, however, the proposed scheme is capable of updating the reference characteristic after write operations in one step. This enormous gain in time (as mentioned before, updating the reference signature in a conventional scheme requires to run through a complete cycle of the test pattern generator), is paid by only little hardware overhead. As explained above, the proposed output data compressor for an m-bit memory can be implemented using $\lceil \log_2 m \rceil$ flip-flops, $\lceil \log_2 m \rceil$ EXOR-gates and one AND-gate.

2.2 Extension to word-oriented RAMs

The scheme for output data compression introduced in the previous section can easily be applied to word-oriented RAMs. As illustrated in Figure 6, for this purpose the word-oriented RAM is considered as bit-oriented memory with addresses of the form (a_w, a_b), where a_w denotes the word address and a_b the bit position within the word.

Analogous to Section 2.1 the memory can then be modeled as a two-dimensional array $M[1\ldots m, 0\ldots n]$ with address space $A = \{1, \ldots, m\} \times \{0, \ldots, n\}$ and „1“-space $A_1 := \{(a_w, a_b) \in A \mid M[a_w, a_b] = \text{„1“}\}$. The reference characteristic C_{REF} for the initial correct memory contents is determined as

$$C_{REF} = \bigoplus_{(a_w, a_b) \in A_1} (a_w, a_b),$$

where the modulo-2 sum of address pairs is defined by $(a_w, a_b) \oplus (a_w', a_b') = (a_w \oplus a_w', a_b \oplus a_b')$.

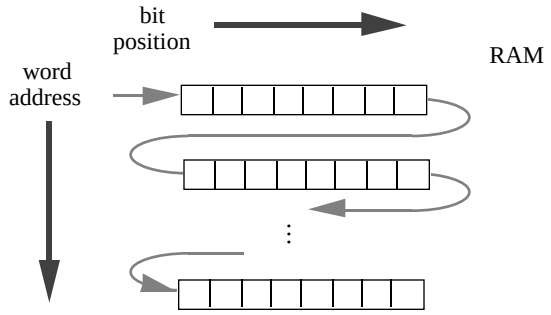

Figure 6: Bit-oriented representation of a word-oriented RAM.

To calculate the corresponding characteristic C_{TEST} during BIST an LFSR or counter generates all word addresses in an arbitrary order. For each fixed word w^* with address a_{w^*} the modulo-2 sum of all addresses (a_{w^*}, a_b) pointing to „1“ entries is computed in one step, i. e. with $A_1(w^*)$ denoting $\{a_b \mid M(a_{w^*}, a_b) = \text{„1“}\}$ a partial characteristic

$$C_{w^*} = \bigoplus_{a_b \in A_1(w^*)} (a_{w^*}, a_b)$$

is determined by a combinational circuit. This way, C_{TEST} can be derived as

$$C_{TEST} = \bigoplus_{1 \le a_w \le m} C_w = \bigoplus_{1 \le a_w \le m} \bigoplus_{a_b \in A_1(w)} (a_w, a_b),$$

which is just the sum of all bit addresses containing a „1“.

To design an output data compressor accomplishing this task, it is crucial to derive an efficient implementation for the partial characteristic C_{w^*}. As introduced above

$$C_{w^*} = \bigoplus_{a_b \in A_1(w^*)} (a_{w^*}, a_b) = \left(\bigoplus_{a_b \in A_1(w^*)} a_{w^*}, \bigoplus_{a_b \in A_1(w^*)} a_b \right).$$

It is easily verified that

$$\bigoplus_{a_b \in A_1(w^*)} a_{w^*} = a_{w^*},$$

if the number of ones in w^* is odd, and

$$\bigoplus_{a_b \in A_1(w^*)} a_{w^*}$$

yields the all-zero vector, if the number of ones in w^* is even. Consequently, the first component of C_{w^*} can be implemented as

$$F_0 \cdot a_{w^*} = \bigoplus_{a_b \in A_1(w^*)} a_{w^*},$$

where F_0 is a function which determines the parity of the word w*.

Since the bit addresses within the word w^* are represented by binary l-bit vectors, $l = \lceil \log_2 n \rceil$,

$$\bigoplus_{a_b \in A_1(w^*)} a_b$$

is obtained by bitwise EXOR-operations

$$\bigoplus_{a_b \in A_1(w^*)} a_b = \left(\bigoplus_{a_b \in A_1(w^*)} a_b^1, \ldots, \bigoplus_{a_b \in A_1(w^*)} a_b^l \right),$$

with a_b^i denoting the i-th component of a_b,, $1 \le i \le l$. As only bit-addresses with $a_b^i = 1$ can contribute to the i-th sum

$$\bigoplus_{a_b \in A_1(w^*)} a_b^i,$$

it is sufficient to implement functions F_i which count (modulo 2) the number of ones at all the addresses with this property. The second component of C_{w^*} is then derived as

$$\bigoplus_{a_b \in A_1(w^*)} a_b = (F_1, \ldots, F_l),$$

Figure 7 shows an example for a memory consisting of seven 4-bit words.

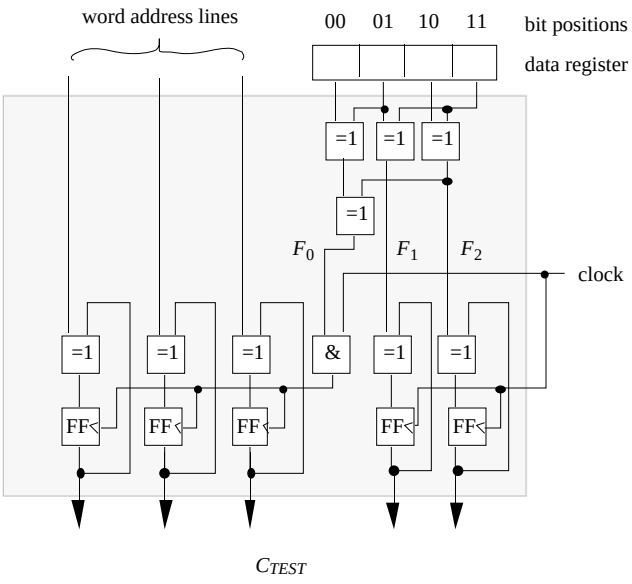

Figure 7: Output data compressor for word-oriented RAMs.

To adjust the reference characteristic C_{REF} in case of a write operation at a specific word-address a_{w^*}, all bit positions a_b have to be determined with $M[a_{w^*}, a_b]^{old} \neq M[a_{w^*}, a_b]^{new}$, and the partial characteristic C_{w^*} must be adjusted to

$$C_{w^*}^{new} = C_{w^*}^{old} \oplus \bigoplus_{a_b \in A_{diff}(w^*)} (a_{w^*}, a_b),$$

where $A_{diff}(w^*) := \{ a_b \mid M[a_{w^*}, a_b]^{old} \neq M[a_{w^*}, a_b]^{new} \}$. As one can easily verify, to accomplish this task it is sufficient to feed the output data compressor of Figure 7 with the „difference-word“ $M[a_{w^*}]^{old} \oplus M[a_{w^*}]^{new}$.

Concerning the hardware cost for the word-oriented case, it can be shown that the EXOR-tree for implementing the functions $F_0, \ldots, F_l$ requires at most

$$\sum_{1 \le j \le l} \left(2^j - 1 \right)$$

EXOR-gates. Overall, the output data compressor of Figure 7 can be implemented using $\lceil \log_2 m \rceil + l$ flip-flops,

$$\lceil \log_2 m \rceil + l + \sum_{1 \le j \le l} \left(2^j - 1 \right)$$

EXOR-gates, and 1 AND-gate. Table 1 summarizes the hardware cost for some example configurations of word-oriented RAMs.

# Words (m)	Word-length (n)	# Flip-Flops	# EXOR-Gates	# AND-Gates
2^{20}	8	23	34	1
2^{20}	16	24	50	1
2^{20}	32	25	82	1
2^{30}	8	33	44	1
2^{30}	16	34	60	1
2^{30}	32	35	92	1

Table 1: Hardware overhead for the proposed output data compressor.

The length of characteristic is $k := \lceil \log_2 m \rceil + \lceil \log_2 n \rceil$, and the probability of aliasing errors for word-oriented RAMs is $2^{-k} = 2^{-\lceil \log_2 m \rceil - \lceil \log_2 n \rceil}$.

3 The BIST architecture

In this section, a complete BIST architecture based on the proposed scheme for output data compression is described for DRAMs. The diagram in Figure 8 shows the necessary BIST equipment in shaded blocks.

The BIST control unit distinguishes between three main phases:

1) During an initialization phase the reference characteristic C_{REF} has to be determined by running through a complete cycle of the test pattern generator TPG, which may be implemented as a maximum period LFSR or counter. The output data compressor is reset to the all-zero vector at the beginning of this phase, and

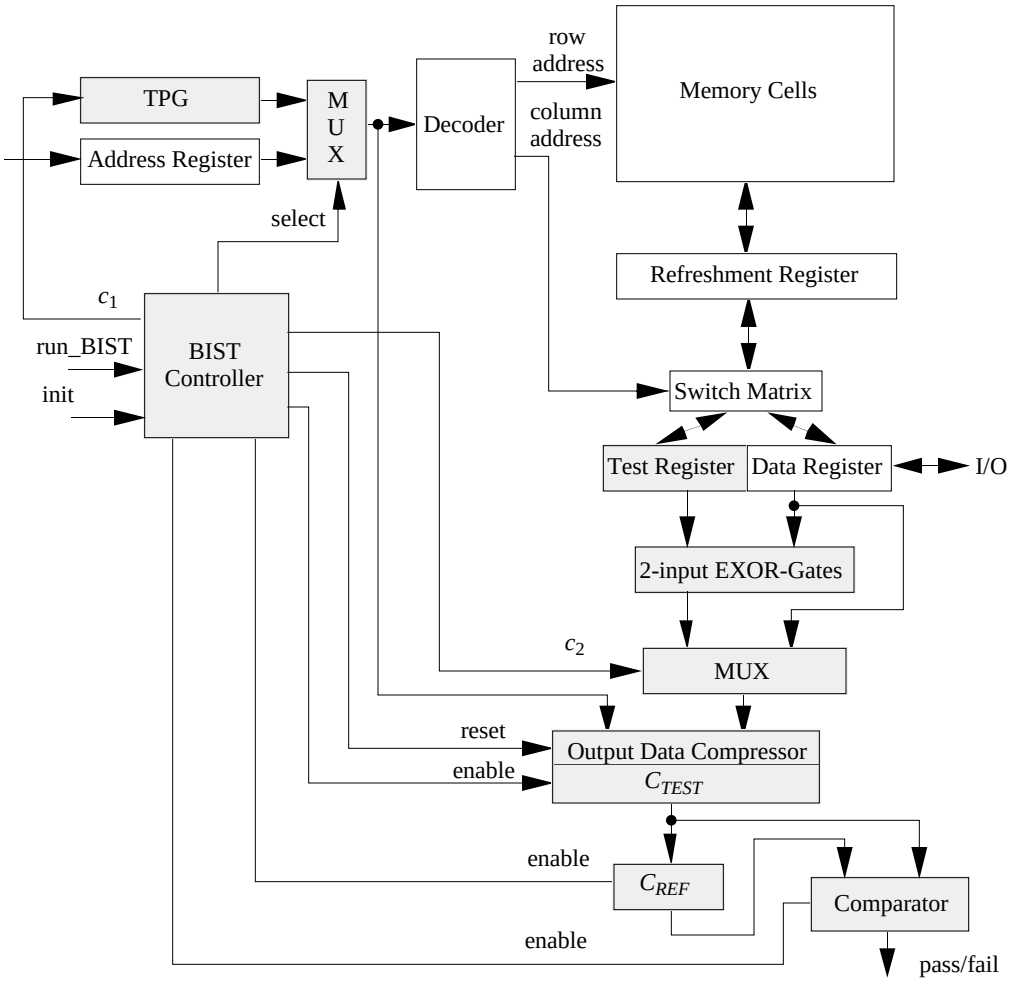

Figure 8: DRAM with BIST.

at the end of this phase the obtained reference characteristic in the flip-flops of the output data compressor is written to the register C_{REF}. If appropriate test patterns have been written into the memory, C_{REF} also represents the outcome of a start-up test.

2) During system operation C_{REF} has to be updated after every write operation. As explained is Section 2.2 this implies that for each write operation the „difference“, i.e. the bitwise EXOR of the old and the new memory entry has to be determined. Therefore, a key issue in implementing the proposed BIST approach is to ensure a backup of the old memory entry at a low hardware and performance penalty. The best method to achieve this goal, of course, depends on the memory organization. Here it is assumed, that a refreshment algorithm for the dynamic RAM is used which writes the complete row containing the word targeted by a write request to the refreshment register before loading the new word from data register. In this case the old memory entry can be transferred from the refreshment register to the test register via the switching matrix. The updated reference characteristic now contained in the flip-flops of the output data compressor is also written to the register C_{REF} .
3) During BIST, the test pattern generator produces all possible addresses, and the test characteristic C_{TEST} is determined in the same way as the reference characteristic C_{REF} during the initialization phase. However, during BIST the enable signal for register C_{REF} is kept zero, such that finally both characteristics C_{REF} and C_{TEST} can be compared to provide the test result at the output of the comparator.

The BIST controller has to realize two loops, one for initialization and one for BIST, and has to control the write operation within two clock cycles. One counter of

length $\lceil \log_2 m \rceil$, two flip-flops, and some combinational logic are sufficient for implementing these three modes, one of which has two states. In addition to this, we need four n-bit registers C_{REF}, C_{TEST}, TPG, and Test Register, and some combinational logic. Prototype implementations show an area requirement of the entire BIST equipment in the order of 10 n-bit registers nearly independent of the memory size.

The BIST hardware is not placed within the datapath and the address logic, and there is no time penalty for read accesses. In a straightforward implementation of write accesses there may be one additional clock latency, as the old contents of the refreshment register must be written into the test register before the new value of the data register can be captured by the register. This latency is avoided, if the refreshment register on the one side and the test and data registers on the other side work on different clock phases. They may even work on the same phase, if the refreshment register has two ports and the switch matrix is doubled.

4 Conclusions

In this paper we have proposed a new method for the built-in test of memory consistency. In contrast to methods known so far, a reference signature is not computed by scanning the entire memory periodically, but by adjusting it dynamically. The reference value of the fault free contents of the RAM is computed as „modulo-2 address characteristic“, and a new technique for output data compression has been presented based on this characterization. It has been shown that the same test quality is achieved as by a conventional approach based on signature analysis. But compared to conventional signature analysis schemes an enormous gain in time is achieved at only slightly higher hardware costs.

5 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 a Telecommunication System; Proceedings IEEE International Test Conference, 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 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.

5 B. Cockburn, Y.-F. N. Sat: Synthesized Transparent BIST for Detecting Scrambled Pattern-Sensitive Faults in RAMs; Proceedings IEEE International Test Conference, Washington, DC, Oct. 1995, pp. 23-32.

6 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

7 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.

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

9 O. Kebichi, M. Nicolaidis, V. N. Yarmolik: Exact Aliasing Computation for RAM BIST; Proceedings IEEE Int. Test Conference, Washington, DC, Oct. 1995, pp. 13-22.

10 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.

11 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, 1986, pp. 830-839.

12 B. Nadeau-Dostie, A. Silburt, and V. K. Agarwal: Serial Interfacing for Embedded-Memory Testing; IEEE Design & Test of Computers, Vol. 7, No. 2, April 1990, pp. 52-64.

13 M. Nicolaidis: Transparent BIST for RAMs; Proceedings IEEE International Test Conference, Baltimore, MD, Oct. 1992, pp. 598-607.

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

15 T. R. N. Rao, E. Fujiwara: Error-Control Coding for Computer Systems; Englewood Cliffs, NJ: Prenctice Hall, Inc., 1989.

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

17 V. N. Yarmolik: Analysis of Signature Testability of Digital Circuits; Automation and Remote Control, March 1990, pp. 1437-1443.

18 Y. You, J. P. Hayes: A self-testing dynamic RAM chip; IEEE JSSC, February 1985, pp. 428-435.