

BIST SCHEME FOR EMBEDDED MULTI-PORT RAMS

V. CASTRO ALVES, M. S. LUBASZEWSKI, M. NICOLAIDIS
and B. COURTOIS

Key words: integrated circuits, VLSI, application-specific integrated circuits, computer storage devices, circuit testing, embedded random access storage, built-in self test, fault models, boundary scan, test algorithms, self-checking circuits

Abstract: In this paper we present a general BIST scheme for the test of memories (single and multi-port) embedded in very complex ASICs. A simple BIST circuit driven by the IEEE standard for the Boundary Scan is shared by all the memories that are tested simultaneously. The area overhead is greatly compensated by the test development time reduction and the link with boundary.

Vgrajen samotest vloženega pomnilnika z večkratnim dostopom

Ključne besede: vezja integrirana, VLSI vezje, ASIC vezje, pomnilniki računalniški, preskušanje vezij, RAM vstavljeni, BIST testiranje, modeli napak, skeniranje robno, algoritmi testni, vezja samokontrolna

Povzetek: V članku je prikazan splošen pristop k realizaciji vgrajenega samotesta pomnilnikov z enojnim ali večkratnim dostopom vloženih v zelo kompleksna ASIC vezja. Preprosto vezje za vgrajen samotest, ki ga krmili IEEE standard za robno skeniranje, je primerno za testiranje vseh spominov hkrati. Povečana površina je kompenzirana s skrajšanjem razvoja testnega programa in izboljšanjem povezave z robom vezja.

1. INTRODUCTION

Testing VLSI circuits is one of the great challenges that IC designers and test engineers are facing nowadays. As RAM generators are being introduced in most of the existing ASIC libraries, many IC designers use embedded memories as macrocells. Moreover, multi-port RAMs are now widely used as embedded memories in telecommunication ICs and multi-processor systems. The testing of these memories is getting more important since they are often embedded in high gate-count ASICs requiring short test development time due to time to market delay requirements. A complete test of an embedded RAM can bring two problems: increase the test length as well as the test development time. This is due to the fact that it may be difficult to reach data, address and control signals from external pins and that it could also be hard to run the test of the embedded RAM concurrently with the test of another block of the circuit. If one considers that in practice, an ASIC design may have more than one embedded memory, and that some can be multi-port RAMs, it is easy to understand that powerful DFT and BIST techniques are needed in such cases. Many authors have proposed built-in self-test schemes for embedded single or dual port RAM testing (SUN 84) (SAL 87) (NAD 90) (SAS 90) (RIT 91), and recently some authors have proposed other schemes that include some interface with a boundary scan chain (KRI 91) (RIJ 91). In this paper, we present an efficient BIST scheme for the test of embedded single and multi-port RAMs. A simple BIST circuit driven by the IEEE

standard for the Boundary Scan (BS) (MAU 90) is shared by all the memories that are tested simultaneously. Although the application of such a circuit is exemplified herein for a memory set, it is also suitable for the go/no-go test of any functional blocks set whose checking scheme is based on signature analysis. Provided that the proposed scheme supplies compact response for the BIST of the entire circuit, its use will lead to a more efficient way of carrying out the testing of the individual ICs on the board.

II. RAM FAULT MODEL

Some recent work have introduced new specific fault models for multi-port RAM testing in order to achieve a high quality test. They are:

- shorts between word lines and bit lines (resp.) of different ports (NAD 90)
- complex coupling (CAS 91), where the simultaneous, transition which is possible in multi-port RAMs, of two or more cells (say cells C_1, C_2, \dots, C_n) can influence a cell C_0 in the memory array, even if the single transitions of cells C_1, C_2, \dots, C_n do not influence C_0 .

Fault models containing complex coupling faults ensure a very high quality test at the expense of an increase of the test complexity. In the following, we will assume that the fault model, as they were defined for single port

RAMs in (MAR 82) or (SUK 81), are sufficient to ensure the detection of most of the faults occurring in multi-port RAMs. We will see later that some topological restrictions must be considered in order to use properly this kind of model. Marinescu, for example, uses the following fault model:

- memory cell stuck-at
- transition fault
- decoder and read/write logic faults
- idempotent coupling faults

Initially, coupling fault models were used for the test of single port 1 bit wide RAMs and they have often been adapted to the test word wide single (RIT 91) (DEK 88) and dual port (NAD 90) RAMs. However, as it has been pointed out in (CAS 91), some restrictions concerning the memory array topology, as well as the use of more than one port in parallel for test application, must be carefully analysed.

Let us first examine the case in which we use more than one port of a multi-port RAM to apply the test vectors in order to speed up the test. If multiple coupling faults are present, they may occur simultaneously if the concerned cells are written at the same time. To illustrate this case let us assume we have a dual port RAM together with the following fault configuration:

- $C_i \uparrow \Rightarrow C_k \downarrow$ (i.e. the positive transition of cell C_i forces cell C_k to make a **negative** transition)
- $C_j \uparrow \Rightarrow C_k \uparrow$ (i.e. the positive transition of cell C_j forces cell C_k to make a **positive** transition)

In this case we are not able to determine the state of cell C_k if simultaneous transitions of cells C_i and C_j are performed. Unless we use an appropriate fault and test algorithm, it is not recommended to use more than one port at a time when we are testing multi-port RAMs.

Another problem arises when you have to test word organized memories, which is common for embedded RAMs. In the general case, there can be coupling faults between any two cells of the memory array, and thus we must consider influences between bits of the same word. Note that when a write access is performed to a given word, we are implicitly performing simultaneous write access to different cells of the array. Thus, as explained previously, conventional coupling fault model cannot predict effect of simultaneous multiple coupling faults.

It seems evident that in order to test such faults, one must use complex coupling fault models that have already been defined in (CAS 91). However, on the one hand, we should use the same algorithm for all types of memories and on the other hand, the number of bits per word can be quite high (there is no real limit in practice), thus pushing up the complexity of the coupling fault model. This is due to the fact that the more bits per word are used, the more we have simultaneous transitions occurring in the memory. This can lead to unacceptably long tests sequences because the test complexity grows

with the number of simultaneous write operations. In order to simplify the problem (and thus the algorithm complexity), we shall use in the following some topological restrictions. They are:

- (1) coupling faults exist only between adjacent cells. This assumption is quite realistic and is in fact a simplified dynamic Pattern Sensitive Fault model

The above restriction is necessary but not sufficient to avoid **concurrent** coupling faults testing (CAS 91) in word organized RAMs. We will investigate also, the effect of some topological restrictions of the memory array structure.

- (2) the use of column multiplexing (see figure II.1) may avoid that cells belonging to the same word (i.e. simultaneously written) are adjacent.

This technique is commonly used for word organized random access memories so that we can say that it is

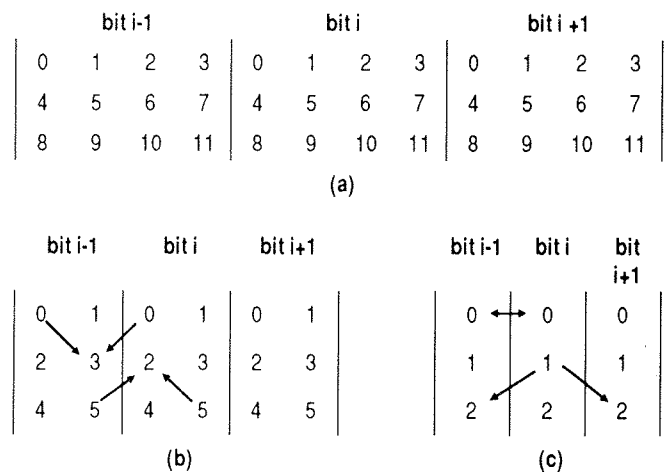


FIGURE II.1 - Column multiplexing

also a quite realistic assumption. Figure II.1 details various possible cases.

In figure II.1.a, the column multiplexing m is equal to 4. We note that cells belonging to the same word are separated by $m-1$ cells, so that if condition (1) is true, we avoid concurrent coupling faults. We also note that no cell can be neighbour of two other cells belonging to the same word. This eliminates the possibility of having complex couplings when a single write operation is performed. Figures II.1.b and II.1.c depict the configurations for $m=2$ and $m=1$ respectively. In these two cases, complex couplings can exist when a single write operation is performed (see arrows). For $m=1$ we can also have concurrent couplings between cells of the same word since cells of bit i has as its neighbour cells of bit $i-1$ and $i+1$.

We can conclude saying that with a column multiplexing greater than four, the memory designer can obtain RAMs with no adjacent cells belonging to the same word. This can greatly simplify the algorithm used to test single and multi-port word wide random access memories.

III - TEST ALGORITHM

The test algorithm must ensure the detection of all faults of the model. Besides of that, the algorithms may satisfy some other conditions:

- short test length (the best known algorithms for testing simplex coupling faults range between $9n$ (DEK 88) and $17n$ (MAR 82))
- simple test pattern sequences (which means simpler hardware and less area overhead)
- perform the test of word organized RAMs (note that most of the test algorithms are valid only for 1bit word wide RAMs)

The goal to be reached is to test simultaneously all the embedded memories of a circuit. Provided that the restrictions presented in section II are respected, it is possible to apply simultaneously the test patterns to all the bits of the selected word. In the case where memories of different word size are tested, the same test pattern is applied in each test cycle to each selected word. The same principle can be applied in the selection of words belonging to different memories. The scheme proposed in the following is valid for any test algorithm provided that the words are addressed sequentially either in an incrementing order or in a decrementing order.

Let us suppose we have to test two embedded memories: memory A with n words and memory B with m words (and $n < m$), using a march test (where words are addressed sequentially according to the definition (GOO 90)) like MARCH C (MAR 82). This algorithm consists of 2 incrementing addressing sequences and two decrementing addressing sequences. For the first

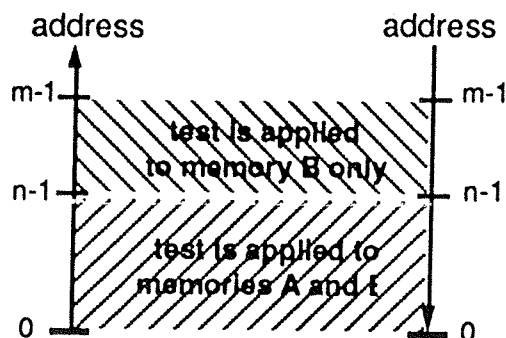


FIGURE III.1 - Testing memories of different sizes

type of sequence, from @ 0 to @ $n-1$, test patterns are applied to both memories. From @ n to @ $m-1$ the test patterns are applied to memory B and the content of memory A is not modified. The same procedure is used in decreasing addressing sequences (see figure III.1). In this way, the test length is determined by the longest memory (i.e. the one that has the greatest number of words). It is obvious that the test algorithm must be independent of the topology of the memory and the address the memory cells to be tested, otherwise it would be impossible to share the RAM BIST circuit with memories of different sizes.

IV - RAM BIST SCHEME

Most of the existing RAM BIST circuits (SAL 87) (YOU 84) (MAZ 87) have been developed for the test of big dynamic RAM chips where the main targets were test speed and area overhead. Complex ASIC testing requires also a short test development time because of time to market delay requirements. Some recent work (NAD 90) (DEK 88) deal with the problem of embedded memories although there is a lack of an efficient general scheme for a BIST circuit because many designs use a variety of RAMs with different sizes, different word sizes and different number of ports. It is not uncommon to find RAMs with up to four ports and system designers are asking for a growing number of ports. A recent work on BIST of RAMs (FRA 90) seems to indicate that approaches based on test architectures rather than on test algorithms are more versatile and will predominate in the future. In the following we will assume that the circuit may have any number of single port and multi port RAMs (with any number of ports), that these memories may have different sizes and that they may have different number of bits per word. BIST circuit sequencing will depend both on the chosen algorithm and on the characteristics of the RAMs to be tested. They are: number of RAMs, number of ports and memory size. Let us use the example of figure IV.1 to explain the BIST scheme.

In this configuration, RAM A has 2^n words of a bits and two read/write ports; RAM B has 2^m words of b bits and RAM C has 2^k words of c bits and three read/write ports. Note that $c < a < b$ and that $k < n < m$. The word size of each memory defines the test pattern bus width, in the same way that the size of the RAMs (m, n, k) define the address bus width.

We know that some errors of the RAM address decoder are mapped into coupling faults in the memory array (THA 78). It is thus essential to fully test each port of each embedded memory. The test sequencing is thus splitted in three phases.

*phase 1: test RAM B
 test RAM A through port 1
 test RAM C through port 1

*phase 2: test RAM A through port 2
 test RAM C through port 2

*phase 3: test RAM C through port 3

The number of phases is equal to the greatest number of ports to be tested among all the memories. In each phase, the test length depends only on the size of the greatest memory to be tested. Once all the ports of a memory are tested, it will not be tested in the next phase. The test ends when all ports of all memories have been tested. In the example shown above the test length is proportional to:

$$2^m + 2^n + 2^k$$

and obviously depends on the complexity of the test algorithm that is implemented.

Note that read-only ports can be associated to a read/write port and they can be checked simultaneously by reading from both ports at the same time. Write-only ports are tested by using another port: either read-only or read/write.

Single port RAMs have one port of order one, dual port RAMs have one port of order one and one port of order two etc... Note that, only ports of the same order are tested simultaneously. Read-only ports are associated to a read/write port of the same memory and receive the same order, so that they are tested at the same time.

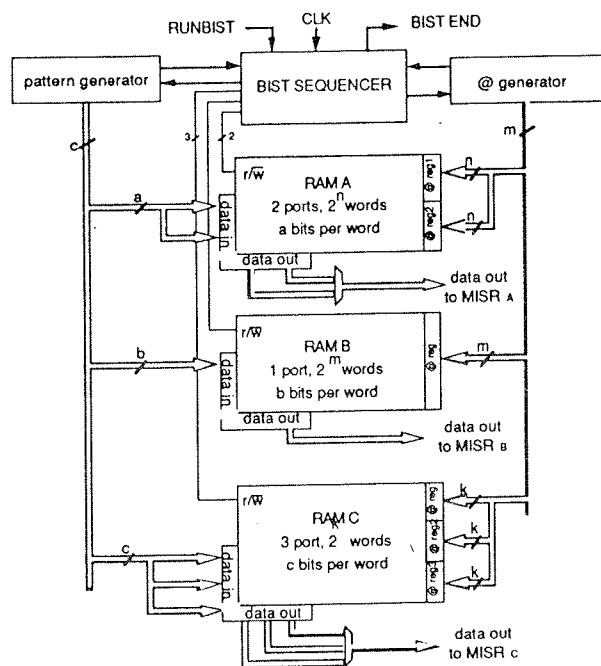


FIGURE IV.1 - RAM BIST scheme

Test verification is done by using one signature analyser per memory. The size of each SA depends on the word size of each memory but the lower limit is given by calculating the aliasing. The advantage of using signature analysis is detailed in section V.

In fact, the goal is to design a programmable BIST generator that is provided with all the necessary information about the memories to be tested. The proposed scheme is composed of three main functional blocks: the address generator, the test pattern generator and the sequencer. The address generator is implemented as a reconfigurable counter because the addressing space might vary during the test phases defined above. The test pattern generator depends only on the chosen test algorithm and is controlled by the sequencer. It can thus be implemented as a non-programmable part of the BIST circuit. Finally, the sequencer is a programmable Finite State Machine because the sequencing of the BIST depends on the characteristics of the memories. As shown in figure IV.2, we use PLA based FSM in order to obtain a greater flexibility for the hardware implementation of the expected sequencing. Moreover, it is quite easy to write a programmable generator for such a structure given that the tools are available in most of the commercial CAD tools.

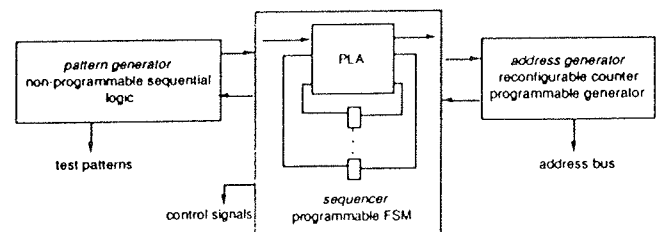


FIGURE IV.2 - BIST implementation

FIGURE IV.2 - BIST implementation

V - COMBINING BIST AND BOUNDARY SCAN

The IEEE standard implementing the boundary scan technique defines a logic to be designed into chips, which is mainly aimed at testing their internal blocks and at checking board interconnections. Its architecture consists of a 4-pin test access port (TAP), a 16-state controller, an instruction register, a boundary scan register, a bypass register and optional user-defined test data registers. A test instruction set is also defined by the IEEE standard. This set is composed by instructions achieving device identification check, circuit bypass, single-step test of circuit internal logic, board interconnection test, circuit interface real-time snapshots and circuit BIST execution.

In the context of this paper, we are interested in the BIST capabilities embedded in the IEEE standard for the boundary scan, e.g. the means of serially reading and

writing into user-defined test data registers within devices and the means of starting circuits BIST process by loading instruction registers with the RUNBIST test instruction. For more details concerning to this standard, readers are referred to (MAU 90).

The availability of a complete BIST process for integrated circuits can greatly reduce the board test time and the test controller memory requirements, since single-step test execution is avoided. Such a complete process is supposed to achieve the test of each entity without the need of loading different instructions for checking different functional blocks, and without the need of neither scanning-in test pattern generation seeds nor scanning-out long signatures. When feasible, this goal should be reached by combining BIST and BS through a proper implementation of the RUNBIST test instruction.

To reach such a goal, first of all the test of all functional blocks must be started by RUNBIST and performed concurrently on the circuit during the Run-Test/Idle TAP controller state activation. Following the BIST completion, all test signatures must be verified inside the circuit, in such a way that a compressed test response for the entire functional block set is provided. Finally, the compressed test responses must be shifted outside circuits in the Shift-DR TAP controller state and analysed by a board test controller circuit. A finite state machine (FSM) for controlling the presented BIST process at the circuit level is shown in figure V.1.

Our BIST scheme for testing single and multiport RAMs was already described in previous sections. BIST schemes for testing other classes of functional blocks can be found in the literature. To complete hardware requirements for testing, we present in the following a built-in test response verification approach, which is applicable whenever all test schemes employed in a circuit are based on signature analysis.

The built-in approach we propose for verifying test responses is based on the combination of the signature verification scheme presented in (NIC 88) and the error memorization capability found in the fail-safe system theory (NIC 89). It consists of an internal scan chain of signature registers, a flip-flop implementing a very

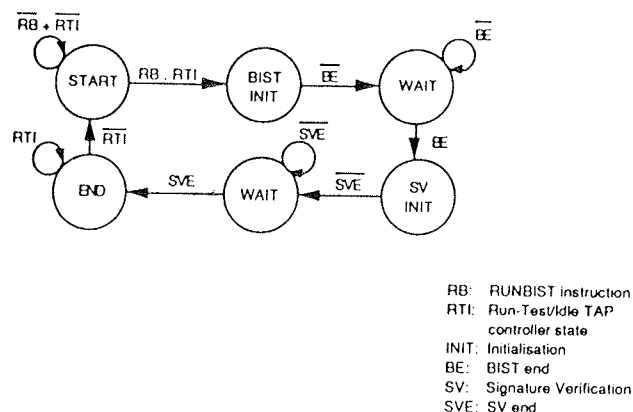


FIGURE V.1 - The BIST controller.

simple state machine, two double-rail checkers and an error indication register providing both parallel and serial loading modes (figure V.2).

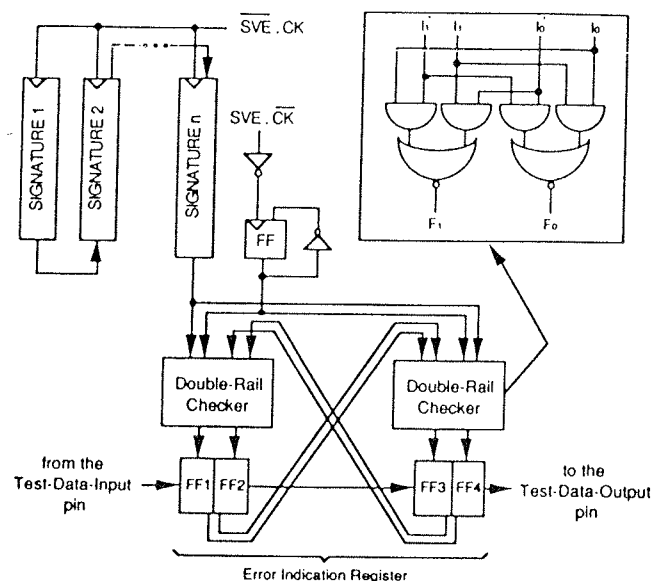


FIGURE V.2 - Signature verification and error memorization.

The checker employed in our approach is a very simple circuit which verifies whether its inputs are double-rail encoded ($i_0 i_0' i_1 i_1'$, where $i_0 i_0'$ and $i_1 i_1' \in \{ "01", "10" \}$) or not ($i_0 i_0' i_1 i_1'$, where $i_0 i_0'$ or $i_1 i_1'$ or both $\in \{ "00", "11" \}$). It delivers at its outputs an encoded word in the former case and a noncoded one in the latter one (CAR 68). This kind of checker is largely used in self-checking circuits (AND 71) (NIC 84) for verifying functional block encoded outputs. One possible logic network implementing it is presented at figure V.2.

The checker feedback configuration depicted at figure V.2 ensures that noncoded inputs coming either from the signatures chain and FF output, or from checker outputs due to their own faults, provoke the memorization of a noncoded word in the error indication register.

The functional principle of our approach is to initialise signature analysers in such a way that, after applying the whole set of test vectors, we have as signatures sequences of the type 0101... (McA 86) presents a simple procedure for computing the signature analyser initial state so that the final state is a constant, regardless of the circuitry being tested. After circuit BIST completion, signature registers are connected serially and shifted into an input of the error memorization circuit. Meanwhile, a flip-flop changing of state at each shift pulse provides the complementary sequence 1010... to

the other error memorization circuit input. Since the referred pair of signals is sent to double-rail checker inputs, the faulty-free condition (a double-rail encoded word) will be memorized when both generated sequences are properly encoded. Whenever an erroneous signature is verified, an error indication (a non-double-rail encoded word) will be memorized by the error indication register.

As we can easily realize, this approach offers a very low surface overhead, whose size does not depend on the number of functional blocks being tested in a circuit. Furthermore, the test hardware duplication employed herein ensures a high coverage of single faults affecting the approach itself, what implies in a high safety.

Concerning to the signature verification time, it is obvious that a parallel approach would be more performant than the one we propose. On the other hand, if the combination of all-zero final signatures and an OR-tree for checking their correctness (as proposed in (McA 86)) is employed, the implied surface overhead will be greatly increased. Beside this, only hardware duplication (the inclusion of an AND-tree in this case) or a test phase checking the OR-tree hardware would provide sufficient safety for such a built-in comparison scheme.

The main goal achieved by our approach in relation to other BIST-BS combination propositions ((KRI 91) (RIJ 91)...) can be summarized by the fact that signatures are verified inside circuits by a very small self-checking logic, avoiding so several time-consuming board scan operations for analysing test results.

VI. CONCLUSION

In this paper we present an efficient general BIST scheme suitable for the test of embedded memories. It can be used for any number of different RAMs, each of which having different word sizes and any number of ports, provided that a minor restriction on memory array topology is respected. Various classes of test algorithms can be implemented on the proposed structure, like march algorithms that are among of the most performing tests actually used. The great flexibility of the BIST scheme here proposed is mainly based on the implementation of a programmable module generator. The techniques here used are based on a previous experience of a BIST generator design (CAS 91). We are now developing a BIST generator that implements MARCH C Marinescu's algorithm (MAR 82), using the L language available under GDT* CAD tool.

The area overhead is not yet quantified but similar to other RAM BIST schemes proposed in the literature. The penalty is greatly compensated for three reasons:

- the flexibility of the BIST circuit allows the test of any kind of embedded memory,

- the proposed scheme allows an efficient sharing of the BIST since the memories are tested simultaneously
- the test development time is greatly reduced

In the proposed scheme, BIST circuit is linked to boundary scan thus providing at-speed high fault coverage capabilities to board test. We also propose a self-checking interface to the boundary scan chain. The aim is two fold:

- perform internally the signature analysis verification, thus avoiding several time-consuming board scan operations for analysing test results
- maintain a high safety for test result compaction

The area overhead resulting from self-checking interface implementation is very small compared to BIST or BS circuitry.

REFERENCES

- (AND 71) ANDERSON D. A. "Design of self-checking digital networks using coding techniques". CSL Univ. of Illinois, Urbana, sept. 1971. Report 527.
- (CAR 68) CARTER W. and SCHNEIDER P. "Design of dynamically checked computers". IFIP Congress, Edinburgh, 1968. Inf Processing 68. Amsterdam, North Holland, 1969. V.2, pp.878-83
- (CAS 91) CASTRO ALVES V., NICOLAIDIS M. LESTRAT P. COURTOIS B. - "Built-in self-test for multi-port RAMs" - to be published in the proceedings of the ICCAD - Santa Clara - November 1991.
- (DEK 88) DEKKER R., BEENKER F., THIJSEN L. - "Fault modeling and test algorithm development for static RAMs" - ITC 1988.
- (FRA 90) FRANKLIN M., SALUJA K. K. - "Built-in self-testing of RAMs" - IEEE Computer - October 1990.
- (GOO 90) VAN DE GOOR A. J., VERRUIJT C. A. - "An overview of deterministic functional RAM chip testing" - ACM Computing Surveys, Vol. 22 N° 1, March 1990.
- (KRI 91) KRITTER S., RAHAGA T. - "Boundary and BIST compatible IEEE 1149.1: VHDL & autosynthesis of a SRAM tester macrocell and chip" - European Test Conference - April 1991.
- (MAR 82) MARINESCU M. - "Simple and efficient algorithms for functional RAM testing" - IEEE International Test Conference 1982.
- (MAU 90) MAUNDER C. M. and TULLOSS R. E. "The Test Access Port and Boundary-Scan Architecture". IEEE Computer Society Press, 1990.
- (McA 86) McANNEY W. H. and SAVIR J. "Built-in Checking of the Correct Self-Test Signature". IEEE International Test Conference, 1986. pp. 54-8.
- (MAZ 87) MAZUNDER P., PATEL J. H. - "An efficient built-in self-testing for random access memory" - ITC 1987.
- (NAD 90) NADEAU-DOSTIE B., SILBURT A., AGERWAL V. K. - "Serial Interfacing for Embedded-memory Testing" - IEEE Design & Test of computers - April 1990.
- (NIC 84) NICOLAIDIS M., JANSCH I. and COURTOIS B. "Strongly Code Disjoint Checkers". 14th Fault Tolerant Computer Symposium, Kisseme, USA, 1984.

(NIC 88) NICOLAIDIS M. "A Unified Built-In Self-Test Scheme: UBIST". 18th Fault tolerant Computer Symposium, June 1988. pp. 157-63.

(NIC 89) NICOLAIDIS M., NORAZ S. and COURTOIS B. "A Generalized Theory of Fail-Safe Systems". 19th Fault Tolerant computer Symposium, June 1989. pp. 398-406.

(RAT 90) RATIU I.M., BAKOGLU H. B., - "Pseudorandom built-in self-test methodology and implementation for the IBM RISC System/6000 processor" - IBM Journal Res. Develop. Vol 34 N°1 January 1990.

(RIJ 91) RIJK R. H. A., SCHWAIR T.M. - "A universal test algorithm for the self-test of parameterizable RAMs" - European Test Conference - April 1991

(RIT 91) RITTER H.C., DEKKER R., KERKHOFF H. G. - "Self-test of a 256Kx4 bit stand alone static RAM" - European Test Conference - April 1991

(SAL 87) SALUJA K.K., SNG S.H., KINOSHITA K. - "Built-in self-testing RAM: a practical alternative" - IEEE Design&Test - February 1987

(SAS 90) van SAS J., CATTHOOR F., INZE L., De MANH. - "Testability strategy and test pattern generation for register files and cus-

tomized memories" - Microprocessors and Microsystems - Vol 14 No 7 September 1990.

(SUN 84) SUN Z., WANG L. - "Self-testing of embedded RAMs" - International Test Conference - 1984

(SUK 81) SUK D., REDDY S.M. - "A march test for functional faults in semiconductor random access memories" - IEEE Transactions on Computers vol. C-30 n° 12 December 1981.

(THA 78) THATTE S. M., ABRAHAM J. A. - "Testing of semiconductor random access memories" - IEEE Transactions on Computers vol. C- 28 June 1978.

(YOU 84) YOU Y., HAYES J. P. - "A self-testing dynamic RAM chip" - Conference on advanced research in VLSI, M.I.T. - 1984-9

V. CASTRO ALVES*, M. S. LUBASZEWSKI, M.
NICOLAIDIS* and B. COURTOIS
IMAG/TIM3 LABORATORY
*Reliable Integrated System Group
46, av. Felix Viallet 38031 GRENOBLE CEDEX -
FRANCE

Prispelo: 15.10.92

Sprejeto: 24.11.92