computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

FAULT TOLERANT MEMORY SYSTEM WITH ACTIVE REDUNDANCY FOR CRITICAL APPLICATIONS

Petru Caşcaval[1)], Doina Caşcaval[2)]

[1)] Department of Computer Science, E-mail: cascaval@cs.tuiasi.ro
[2)] Department of Textile Technologies, E-mail: cascaval@tex.tuiasi.ro
"Gh. Asachi" Technical University of Iaşi, 53A D. Mangeron Blvd, 700050, Iaşi, ROMANIA

Abstract: *This paper proposes a fault tolerant RAM memory architecture with active redundancy dedicated to critical applications. To assure high reliability, additional modules are provided in the system. To reach a high safety, memory system is composed by built-in self-testing RAM devices (BIST–RAM). Memory modules are tested online one by one, in a sequential order, during the normal operation. A primary module under testing is temporary replaced by another one. If one module is diagnosed as being failed it is eliminated from the system automatically and replaced with a spare module. The memory control unit responsible for performing the reconfiguration of the memory system has a cellular structure and introduces an acceptable extra delay for the memory access.*

Keywords: *BIST-RAM, Online Testing, Fault Tolerance, Reliability, Safety, Reconfigurable System*.

1. INTRODUCTION

Digital systems for critical applications have very high reliability and safety requirements. To assure system integrity, control systems must be tested continuously during the normal operation. In other words, we need online testing to assure system integrity. On the other hand, a high reliability implies fault tolerance. In this work the fault tolerance is based on active redundancy.

In order to increase safety of RAM memory two complementary techniques are available [1]:

a) to add additional circuits for detecting and correcting errors;
b) to perform online testing for detecting immediately the faults occurred in the system.

This paper deals with the online memory testing. The goal of online testing is to detect faults and take appropriate corrective action. An additional module is necessary to perform online testing because by testing data in RAM is lost. More exactly, an additional module is necessary to keep data of the module under testing.

The online testing can be peformed by microprocessor while normal operation is temporarily suspended. This technique is called non-concurrent online testing and its goal is to detect permanent faults [2], [3]. Non-concurrent online testing cannot detect transient or intermittent faults whose effects disappear quickly. For critical or high available systems, the testing must be performed in parallel with the normal operation using additional circuits. This technique is called concurrent online testing [4].

In this paper a memory system able to assure a concurrent online testing is discussed. More exactly, a memory system composed by built-in self-testing RAM devices (BIST–RAM) is considered. The synthesis of a BIST–RAM implementing the march test reported in [5] is discussed in [6].

The remainder of this paper is organised as follows. Section 2 presents the structure of the reconfigurable memory system we propose and describes how this memory system works. Section 3 presents a possible implementation of the memory control unit which is responsible for performing the reconfiguration of the memory system. Section 4 presents an online testing procedure that allows memory modules to be tested one by one in a sequential order. In section 5, the improvement in memory reliability provided by spare modules is discussed. Finally, some conclusions are drawn regarding this work.

2. RECONFIGURABLE MEMORY SYSTEM

Consider a RAM memory system with active redundancy illustrated in Fig. 1 [7]. The memory is

logically organized into m equal partitions so that the memory system comprises m identical primary modules. To assure a high reliability, s additional memory modules are provided in the system. Thus, the memory system is composed by $m+s$ identical modules (M_1, …, M_{m+s}), from which m modules are used for normal operations in the memory, s-1 modules are in stand-by and one module is under testing. The module under testing is temporary replaced by another module. Memory modules are tested periodically, one by one in a sequential order, under the microprocessor control. If one of the modules is diagnosed as being failed then it is

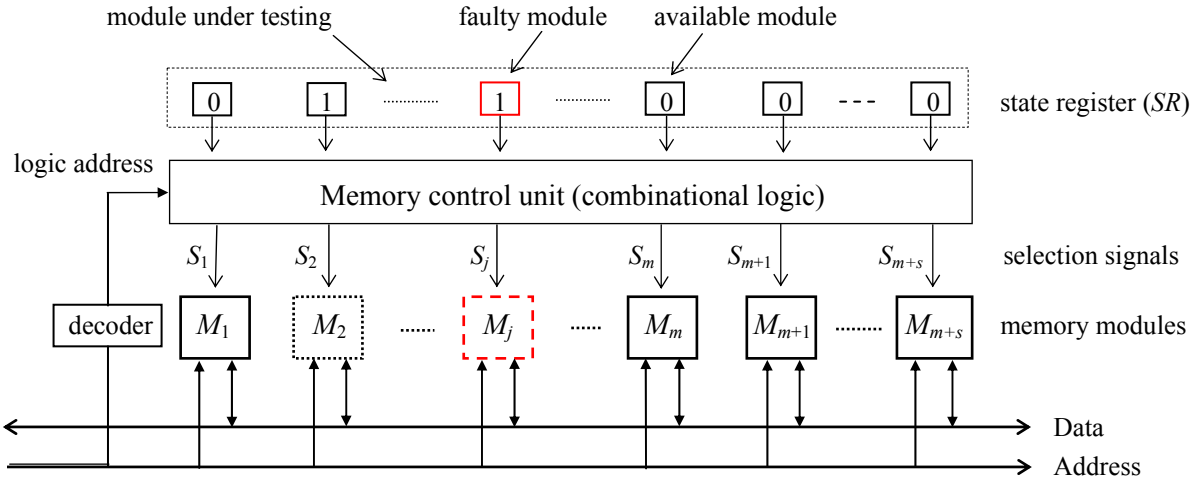

Fig.1 - Memory system with active redundancy

marked as inoperable in a state register (SR), and replaced with the first available spare module. Each spare module can substitute any of the m primary modules. Consequently, the memory system can tolerate up to s module failures before the memory becomes inoperable. The module under testing is also inoperable.

The memory control unit is a combinational logic responsible for the normal operations of the memory. The memory control unit is also responsible for performing the reconfiguration of the memory system. Control unit selects the first m available modules for use in the memory, which are marked with *0* in the state register SR. The next section is dedicated to the synthesis of this memory control unit.

3. SYNTHESIS OF MEMORY CONTROL UNIT

Consider a memory system with m primary and two additional modules (s=2). Let A_i be the logic address, (i=1, 2, ..., m) and S_j the signal to select module j (j=1, 2, ..., m+2). Only one logic address can be 1 at a time. Signals S_1, ..., S_{m+2} are generated depending on the logic addresses, A_1, ..., A_m, and the state values, SR_1, ..., SR_{m+2}. The memory control unit selects the first available module when A_1=1, the second available module when A_2=1, and so on, the mth available module when A_m=1. To locate the ith available module, memory modules are countered in increasing order, jumping over the inoperable modules, which are marked with 1 in the state register SR. Let BS_j be the logic block which generates signal S_j to select module M_j (j =1, 2, …, m+2). Of course, signal S_j can be activated only if M_j is available (SR_j=0). Since there are two additional modules, one or two inoperable modules are tolerated by this system with active redundancy.

As follows we present the conditions in which memory modules M_1, M_2, …, M_{m+2} are selected.

1) M_1 is selected when it is operable and A_1=1.
2) M_2 is selected in two cases:
   - M_1 and M_2 are operable, and A_2=1, or
   - M_1 is inoperable and A_1=1.
3) Module M_j (j=3,…, m) is selected in three cases:
   - all modules M_1, …, M_j are operable, and A_j=1, or
   - one of the modules M_1, …, M_j is inoperable, and A_{j-1}=1, or
   - two of the modules M_1, …, M_j are inoperable, and A_{j-2}=1.
4) M_{m+1} is selected in two cases:
   - one of the modules M_1, …, M_m is inoperable, and A_m=1;
   - two of the modules M_1, …, M_m are inoperable, and A_{m-1}=1.
5) M_{m+2} is selected when two of the modules M_1, …, M_{m+1} are inoperable, and A_m=1.

Taking into considerations conditions 1–5 previously defined, we propose a memory control unit with the cellular structure illustrated in Fig. 2.

Logic variable x_1 reflects the state of module M_1, and logic variables x_j^1 and x_j^2 ($j = 2,\ldots, m+2$) reflect the state of memory modules $M_1,\ldots, M_j$, as follows:

- $x_1 = 0$ when module M_1 is available, and $x_1 = 1$, otherwise ($x_1 = SR_1$);
- $x_2^1 = 0$ and $x_2^2 = 0$ when modules M_1 and M_2 are available;
- $x_2^1 = 1$ and $x_2^2 = 0$ when one of modules M_1 and M_2 is inoperable;
- $x_2^1 = 0$ and $x_2^2 = 1$ when both modules M_1 and M_2 are inoperable.

Note that $x_2^1 = 1$ and $x_2^2 = 1$ is an invalid combination.

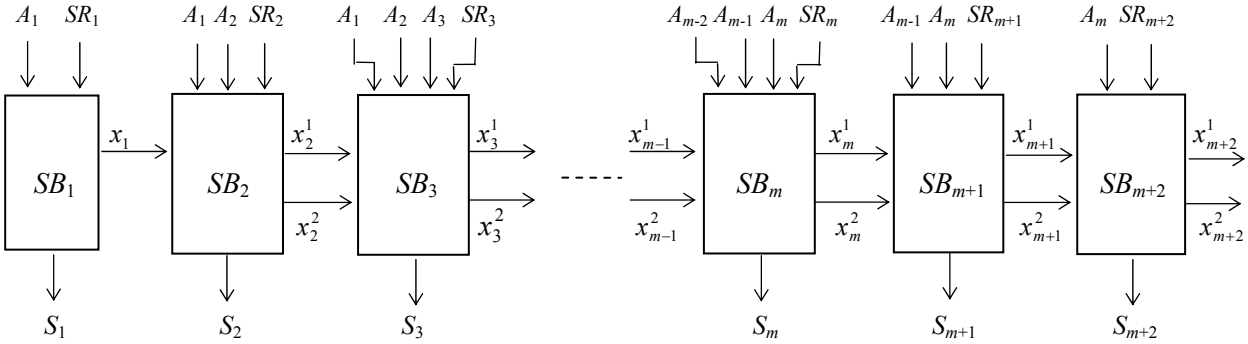

Fig. 2. - Cellular structure of control unit

- $x_j^1 = 0$ and $x_j^2 = 0$, $j = 3,\ldots, m+2$ when all modules $M_1,\ldots, M_j$ are available;
- $x_j^1 = 1$ and $x_j^2 = 0$, $j = 3,\ldots, m+2$ when one module of the set $M_1,\ldots, M_j$ is inoperable;
- $x_j^1 = 0$ and $x_j^2 = 1$, $j = 3,\ldots, m+2$ when two modules of the set $M_1,\ldots, M_j$ are inoperable;
- $x_j^1 = 1$ and $x_j^2 = 1$, $j = 3,\ldots, m+2$ when more than two modules of the set $M_1,\ldots, M_j$ are inoperable.

Taking into account conditions 1–5, and the signification of the logic variables previously defined, logic variables S_1, S_2, S_j (j=3,…, m), S_{m+1}, S_{m+2} are given by the following equations.

$$S_1 = \overline{SR_1} A_1 \tag{1}$$

$$S_2 = \overline{SR_2} A_2 \overline{x_1} + \overline{SR_2} A_1 x_1 \tag{2}$$

$$S_j = \overline{SR_j} A_j \overline{x_{j-1}^1}\,\overline{x_{j-1}^2} + \overline{SR_j} A_{j-1} \overline{x_{j-1}^1} x_{j-1}^2 + \overline{SR_j} A_{j-2} x_{j-1}^1 \overline{x_{j-1}^2},\ j = 3,\ldots, m. \tag{3}$$

$$S_{m+1} = \overline{SR_{m+1}} A_m x_m^1 \overline{x_m^2} + \overline{SR_{m+1}} A_{m-1} \overline{x_m^1} x_m^2 \tag{4}$$

$$S_{m+2} = \overline{SR_{m+2}} A_m \overline{x_{m+1}^1} x_{m+1}^2 \tag{5}$$

Logic variables x_2^1, x_2^2 and x_j^1, x_j^2 (j=3,…, m+2) are given by Eqs. (6) - (9).

$$x_2^1 = x_1 \overline{SR_2} + \overline{x_1} SR_2 \tag{6}$$

$$x_2^2 = x_1 SR_2 \tag{7}$$

$$x_j^1 = \overline{x_{j-1}^1}\,\overline{x_{j-1}^2} SR_j + \overline{x_{j-1}^1} x_{j-1}^2 SR_j + x_{j-1}^1 x_{j-1}^2 = \overline{x_{j-1}^1} SR_j + x_{j-1}^1 x_{j-1}^2 \tag{8}$$

$$x_j^2 = x_{j-1}^2 + x_{j-1}^1 \overline{x_{j-1}^2} SR_j \tag{9}$$

Based on Eqs. (1)–(9), the selection blocks SB_1, SB_2, SB_j (j=3,…, m), SB_{m+1} and SB_{m+2}, that compose the control unit, can be implemented with NAND logic gates as shown in Figs. 3–7. Note that selection blocks $SB_3,\ldots, SB_m$ are identical. We highlighted in Figs. 3–7 the logic gates that affect the memory access time. Note that the memory control unit introduces an acceptable extra delay in the memory access time equal to the propagation time through two logic gates.

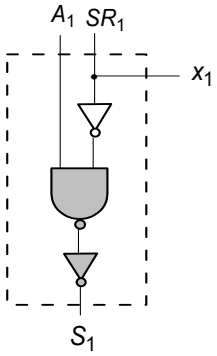

Fig. 3 – Selection block SB_1

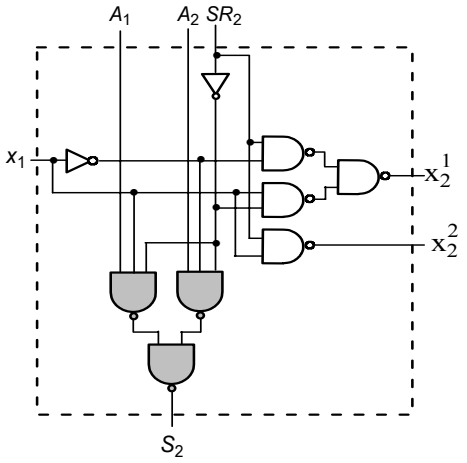

Fig. 4 – Selection block SB_2

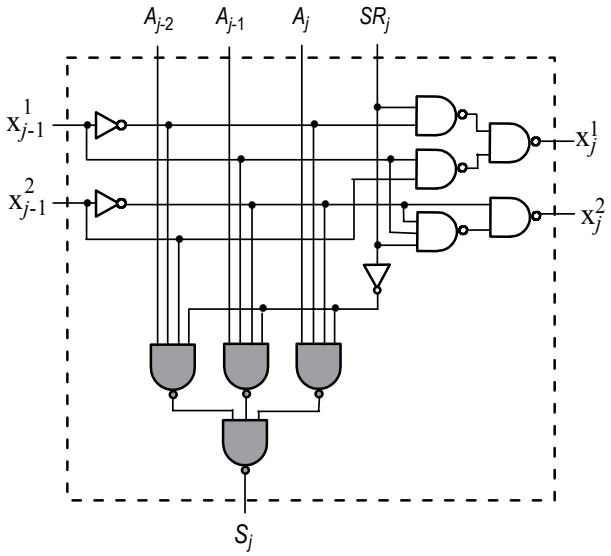

Fig.5 – Selection block SB_j (j=3, ..., m)

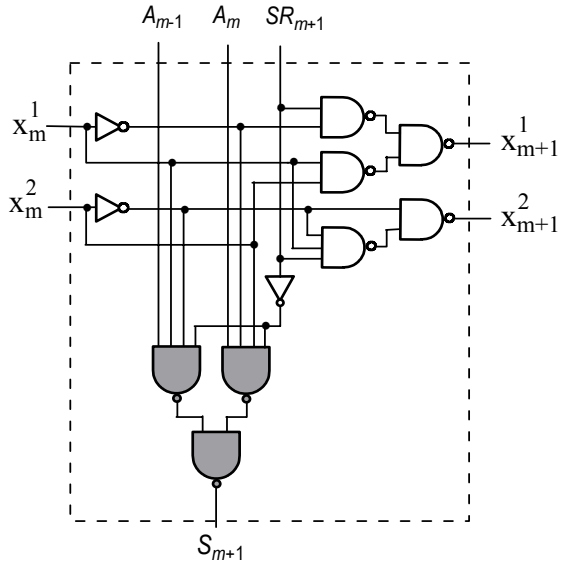

Fig. 6 – Selection block SB_{m+1}

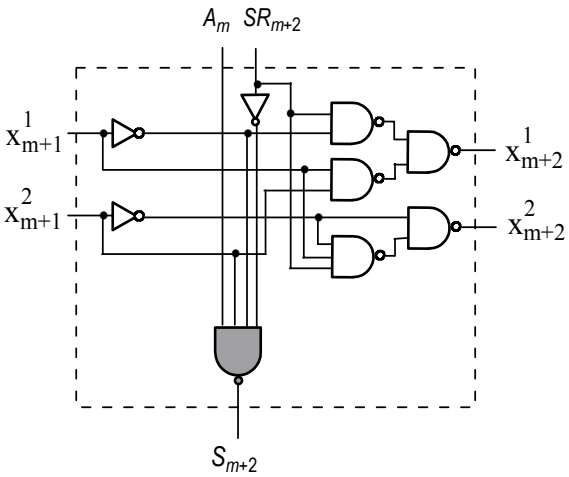

Fig. 7 – Selection block SB_{m+2}

Logic variables x^1_{m+2} and x^2_{m+2} can be used by microprocessor to test the memory control unit. The values of these logic variables must be in accordance with the number of inoperable memory modules that are marked with 1 in the state register *SR*.

4. ONLINE MEMORY TESTING

During a cycle of memory testing all BIST-RAM devices are tested two times. The RAM modules are tested one by one, first in decreasing order and then in increasing order [7], as shown in Fig. 8, where symbol * denotes the module under testing.

The preliminary operations needed for testing module M_j, driving the test from right to left (+) or from left to right (−), are presented as follows:

1) Module $M_{j\pm1}$ is put on normal operation mode and marked as available in the state register

($SR_{j\pm1}$=0);

2) Data from M_j is copied to $M_{j\pm1}$;

3) Module M_j is put on self-testing mode and marked as inoperable in the state register ($SR_{j\pm1}$=1).

Comparing with the procedure of testing modules by rotation reported in [3], this procedure that alternates decreasing and increasing order of testing needs a half number of operations for data transfer. As disadvantage, the memory modules are tested at different intervals of time. For example, consider four primary and one additional modules which makes possible the online testing. Let T be the period of time at which the microprocessor starts the testing process for a new module.

Fig. 9 describes a testing cycle of the memory array. Memory modules are numbered from 1 to 5.

1	2	3	.	j–1	j	j+1	.	m–1	m	*
1	2	3	.	j–1	j	j+1	.	m–1	*	m
1	2	3	.	j–1	j	j+1	.	*	m–1	m
.	.	.	.	.	.	.	.	.	.	.
1	2	3	.	j–1	j	*	.	m–2	m–1	m
1	2	3	.	j–1	*	j	.	m–2	m–1	m
1	2	3	.	*	j–1	j	.	m–2	m–1	m
.	.	.	.	.	.	.	.	.	.	.
1	2	*	.	j–2	j–1	j	.	m–2	m–1	m
1	*	2	.	j–2	j–1	j	.	m–2	m–1	m
*	1	2	.	j–2	j–1	j	.	m–2	m–1	m
1	*	2	.	j–2	j–1	j	.	m–2	m–1	m
1	2	*	.	j–2	j–1	j	.	m–2	m–1	m
.	.	.	.	.	.	.	.	.	.	.
1	2	3	.	*	j–1	j	.	m–2	m–1	m
1	2	3	.	j–1	*	j	.	m–2	m–1	m
1	2	3	.	j–1	j	*	.	m–2	m–1	m
.	.	.	.	.	.	.	.	.	.	.
1	2	3	.	j–1	j	j+1	.	*	m–1	m
1	2	3	.	j–1	j	j+1	.	m–1	*	m
1	2	3	.	j–1	j	j+1	.	m–1	m	*

Fig. 8 – Test cycle of memory array (logic modules are numbered from 1 to *m*)

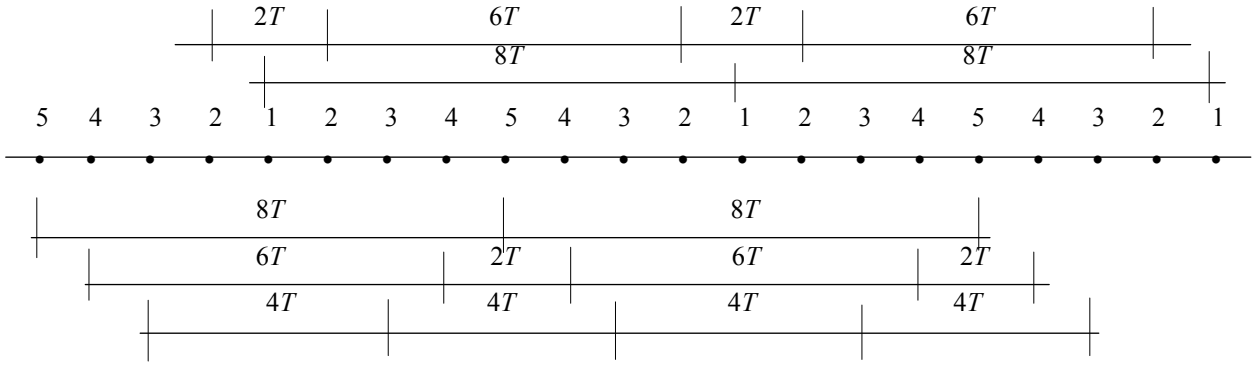

Fig. 9 – Test cycle for each memory module (*m*=4, *s*=1)

For a memory system with m primary modules, the modules are tested periodically, at intervals of time given by the following relationships:

$2mT$ for modules M_1 and M_{m+1} (10)

$2(j-1)T$ and $2(m-j+1)T$, alternatively, for modules M_j and M_{m+2-j}, where j=2, 3, ... (11)

The online testing does not affect significantly the normal operation. The microprocessor interrupts

the normal operation only for saving data before starting the test for the next module.

5. RELIABILITY IMPROVEMENTS

The improvement in memory reliability provided by s spare modules can be expressed as the ratio of the probability of single failure in a memory system with m modules to the probability of $s+1$ failures in a memory system with $m+s$ modules.

Let $R=exp(-\lambda t)$ be the probability of a single memory device operating correctly, where λ is the failure rate of single memory device. The probability of the device failing is $F=1-R=1-exp(-\lambda t)$.

Of concern the non–redundant system, the probability of all m devices operating correctly is R^m, and the probability of one failure is

$$P_1 = mR^{m-1}F\,. \tag{12}$$

The probability of $s+1$ failures in a memory system with $m+s$ modules is

$$P_{s+1} = C_{m+s}^{s+1}(1-R)^{s+1}R^{m-1}\,. \tag{13}$$

Consider $m=8$, $s=2$, $\lambda=10^{-7}h^{-1}$. The improvement in memory reliability depends on the period of time for which we compute the reliability. Table 1 presents the values of ratio $r=\frac{P_1}{P_{s+1}}$, for different periods of time (T), in which P_1 and P_{s+1} are given by Eqs. (12) and (13).

Table 1. The improvement in memory reliability

(m=8, s=2, $\lambda=10^{-7}h^{-1}$)

T [years]	1	2	5	7	10
r	86952	21757	3490	1784	876

6. CONCLUSIONS

This paper focuses on microprocessor based system dedicated to critical applications and proposes a RAM memory system with active redundancy, able to assure high reliability and safety. Taking into account safety requirements, built-in self–testing RAM devices have been considered. Memory modules are tested periodically, one by one, in parallel with normal operations. If one of the modules is diagnosed as being failed then it is automatically replaced with a spare module. In this way memory system can reach very high reliability.

Comparing with the procedure reported in [3], where modules are tested by rotation, in this implementation the procedure of testing alternates the decreasing order with the increasing one. Thus, the number of operations for data transfer before testing is two times smaller in comparison with the classic procedure. The memory control unit responsible for performing the reconfiguration of the memory system has a cellular structure and introduces an acceptable extra delay for the memory access. This reconfigurable memory architecture can beconsidered for both sequential and parallel computing systems.

7. REFERENCES

[1] D. Pradhan, *Fault-Tolerant Computer System Design*, Prentice Hall, New York (1996).

[2] H. Al–Asaad, B. Murray, J. Hayes, Online BIST for Embedded Systems, *IEEE Design & Test of Computers*, 15 (4), (1998), pp.17-24.

[3] S. Barbagallo, et al., Integrating Online and Offline Testing of a Switching Memory, *IEEE Design & Test of Computers*, 15 (1) (1998), pp. 63-70.

[4] M. Lobetti – Bodoni, A. Pricco, et al., An On-Line Bisted SRAM IP Core, *Digest of Papers, Int'l Test Conference*, Washington, (1999), pp. 993-999.

[5] P. Caşcaval, S. Bennett, Efficient March Test for 3-Coupling Faults in Random Access Memories, *J. of Microprocessors and Microsystems*, Elsevier, 24 (10) (2001), pp. 501-509.

[6] P. Caşcaval , V. Onofrei., Built-in Self-Testing for Coupling Faults in Random Access Memories, *Bul. Inst. Polit.Iaşi, XLVI* (*L*)*, 1-4, Autom. Control & Computer Science*., (2000), pp. 93-101.

[7] P. Caşcaval, New results in testing of microprocessor based systems, *PhD Diss., Tech. Univ."Gh.Asachi"*, Iassy, (2001), pp. 162-170.

Petru CAŞCAVAL *is currently Associate Professor in the Department of Computer Science at "Gh.Asachi" Technical University of Iaşi, Romania. Petru Caşcaval received a BEng degree in Computer Science, and a PhD*

degree in Systems Engineering, from "Gh.Asachi" Technical University of Iaşi. His research interests include test generation and testable design of digital systems, modeling and simulation, real-time systems, reliability and fault-tolerance.

Doina CAŞCAVAL *is currently Associate Professor in the Department of Textile Technologies at "Gh.Asachi" Technical University of Iaşi. Doina Caşcaval received a BEeng degree in Computer Science, and a PhD degree in Textile Engineering, from "Gh.Asachi" Technical University of Iaşi. Her research interests include computer programming, real–time monitoring and control in textile processes, discrete–event systems, modeling and simulation, reliability and fault–tolerance.*