A&QT-R 2002 (THETA 13)
International Conference on Automation, Quality and Testing, Robotics
May 23 - 25, 2002, Cluj-Napoca, Romania

# PRT: PSEUDO-RING TESTING - A METHOD FOR SELF-TESTING RAM

**Ghenadie BODEAN**

*Technical University of Moldova,*
*MD-2012, R. Moldova, Kishinau, St. cel Mare, 168*
gbodean@mail.md

***Abstract*— Pseudo-ring testing it's a scan-method for random access memory (RAM). But in opposite to the well-known embedded self-testing methods this technique does not require the reconfiguration of the unit under test. Moreover, thousandth shares of percent of overhead for it implementation is needed. The RAM is tested on the working frequencies.**

**For quality test evaluation the strongly mathematical model is introduced. The stuck-at and the combination of stuck-at faults for RAM cells array are analyzed. It is proved that the pseudo-ring testing provides the 100% detection of all declared faults. The complexity of the method is the order *O*(9*N*), *N* - capacity of RAM.**

**Pseudo-ring testing can be applied as for RAM with one-bit organization of cell array and for word-oriented RAM.**

***Key words*— RAM self-testing, cell-faults, low redundancy.**

## I.Introduction

The semiconductor memory is an indispensable, inherent and most important part of any control system. The reliable functioning of memory modules in much predestines the reliable functioning of the whole system. At the last time many efforts are undertaken as manufacturers of digital components for example, AMD, Atmel, Intel, TI and others, and the diverse computer society to apply the economical test methods and tools. Visible success in this sense was achieved in the field of elaboration and mastering of compact built-in self-test (BIST) systems.

The technology of embedded test system has succeeded in a several steps: from the compact separately realization of stimuli generator and responds (outputs) analyzer of unit under test (UUT), to compact joint (in common) realization of generator and analyzer. The BILBO (Built-in Logic Observator) tools may be a good example. Now the stage in the evolution of BIST tools, namely the compact realization of the generator and the analyzer based on the UUT itself is born. It's expected that this will allow to reduce the complexity of test equipment hardware, because appears the possibility to transform the UUT into a generator, i.e. linear automaton, by connecting the inputs and outputs [1]. However not each unit can be presented as a linear automaton. And the application, for example, of Level-Sensitive Scan Design technology needs more "overhead" expenses - about 20%.

In this work will be showed the way to realize the generator and the analyzer on the base of the unit itself that is the Random Access Memory (RAM). The typical model of RAM is presented as array of cells and the peripheral equipment such as address decoder, read-write logic, in-out amplifiers, internal buss and others. If to accept the hypothesis that the peripheral equipment function normally, i.e. according to the specification, then the efficiency of test system will be estimated by the quality of array memory testing.

It needs to remind that the number of faults detected evaluates the test quality. Single-cell faults, such as stuck-at faults - stuck-at logic 0 (s@0) and stuck-at logic 1 (s@1), are the most typical faults of the memory array [2,3]. Among classical faults, frequently are used the other models such as coupling, inter-influence, transition, speed-related and other kind of faults. More "exploited" algorithm for RAM testing is the March and its modifications [4]. The order of complexity of March and March-like tests is close to $O(N)$, with $N$= capacity of memory.

As the chip area increases and the structure sizes of semiconductor devices decrease, the designers of high-end processors and system-on-chip search for a method and an algorithm for efficient embedded memories testing. A suitable test algorithm is chosen and implemented as a BIST. At the same time the low area overhead is one of the major criterion for acceptance of the additional, supplementary modules, blocks.

In this work a new approach, conventionally named ***pseudo-ring testing*** (PRT), to built-in self test that, in authors opinion, embodies the ideas and principles of the design of a compact generator and analyzer on the base of the components of tested random access memory.

## II. The idea and designing of pseudo-ring RAM testing

Pseudo-ring testing is based on the existence of some "likeness" between the abstract representation of RAM and Linear Feedback Shift Register (LFSR), i.e. linear automaton.

The linear automaton are the essence of almost all generators, like pseudo-random pattern generator, and of the signature analyzer of the UUT outputs. In the algorithm of ring testing the conclusion about fault-free is made on comparing the starting (initial) **Init** and the final **Fin** states of automaton, and the goal of design and analysis of ring system is reduced out to linearization of UUT [1]. These features are most attractive for economical testable design!

Suppose that the register ***r***-staged ***Rg*** is the part, for example the first ***r***, of the cells of RAM. Make an analysis as follows. At the each clock of time the contents of register ***Rg*** is shifted left, i.e. the data is rewritten from one position of register to other, from the left to the right:

$Rg[\ i]$ **ß** $Rg[\ i+1]$, $i=1, \ldots, r-1$,
$Rg[\ r]$ **ß** "the sum modulo 2".

In this procedure the part of data of register ***Rg***, which is contained (stocked) in the ***r***-1 position, is remains intact. New unit of data is added to this part. So, at the each clock of time, during of some period ***T***, the frame of register ***Rg*** seems to shift left by one position together with sum forms a new state of ***virtual*** linear automaton. If imagine the sequential transition of this virtual automaton in the space of cells of RAM array, then it will result something like the picture in fig.1.

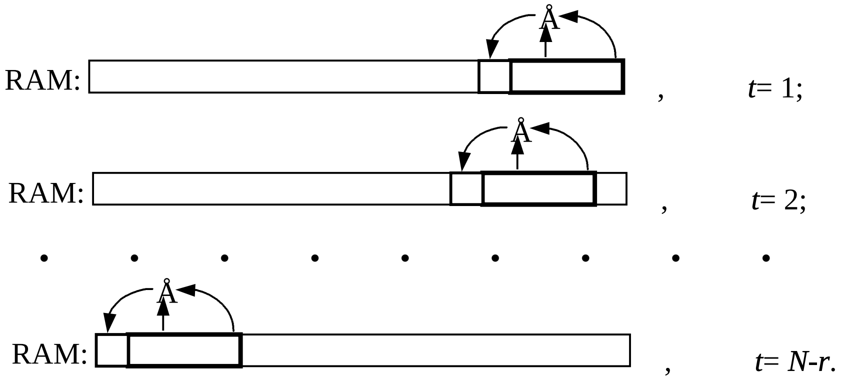


Fig.1. The dynamics of the virtual automaton transition; *t* - clocks.

The functions of the generator and the analyzer of the ring system *seem to be taken place by* the components of UUT. That's why the proposed technique is named *pseudo-ring testing* of RAM.

Now, from the idea go to the practice. In the beginning we'll consider the task of adaptation of the technique to the available RAM. Unfortunately the modern circuits, block of memory don't allow to read simultaneously some memory cells. This restriction dictates the architecture of PRT system.

Let's consider some particularities of pseudo-ring testing itself. At the start moment ($t$= 0) the value, corresponding to the initial state of the LFSR is written in the (first) $r$ cells of RAM. Within the time $T$ the virtual automaton should cross through the memory cells. Let's name this transition - *test iteration.* At each clock of test iteration is carried out a set of operations, which will be named *test subiteration.*

Let the *test experiment* be the sequence of test iterations, needed to achieve the required level of PRT quality. The tasks, which are necessary to put and to solve in the framework of PRT RAM, are: (1) determination of the structure of the linear automaton LFSR; (2) establish the number of test iterations necessary to achieve the maximal ffect, i.e. high resolution; (3) what should be the initial states of LFSR in each iteration of test experiment?

The initial dates for PRT designing are the parameters of RAM: 1) capacity, evaluated from the address input $A$; 2) the number of cell bit position, called *binarity*, $m$, as a rule is multiple to 2 - 1 bit, 8 bit (= 1 byte) etc, and 3) the control lines of read-write, usually, combined in *R/`W*. The task of the design is the elaboration of the scheme of the virtual LFSR and the overhead of hardware, which implement the algorithm of PRT.

Let's consider a trivial example. A very simple scheme of LFSR is that which contains a 2-stage register. Such a scheme is described by the irreducible polynomial $\varphi(x)= 1+x+x^2$ over Galois field **GF** (2). Let's apply this scheme to implement PRT RAM with 1 bit cell. The typical circuit RAM has address input $A$, read-write input *R/`W*, a data input $D$ and an output. A $j$-th test subiteration involves 3 steps:

$$\textbf{subiteration j}\begin{cases}\text{clock }1:\text{read cell }i,\\ \text{clock }2:\text{read cell }i+1,\\ \text{clock }3:\text{write cell }i+2,\end{cases}\ \textbf{subiteration j}+\mathbf{1}\begin{cases}\text{clock }1:\text{read cell }i+1,\\ \text{clock }2:\text{read cell }i+2,\\ \text{clock }3:\text{write cell }i+3,\end{cases}\ \textbf{etc}.$$

The presented bottom sub iterations allow reconstructing the scheme of the additional equipment, needed to implement the algorithm of PRT. It's obvious that it needs supplementary the UP-DOWN counter, two flip-flops, adder modulo 2 (XOR) and a register to clocks the test sub iterations. Then, for the example analyzed before, the scheme of PRT system will be represented as it's illustrated in fig.2.

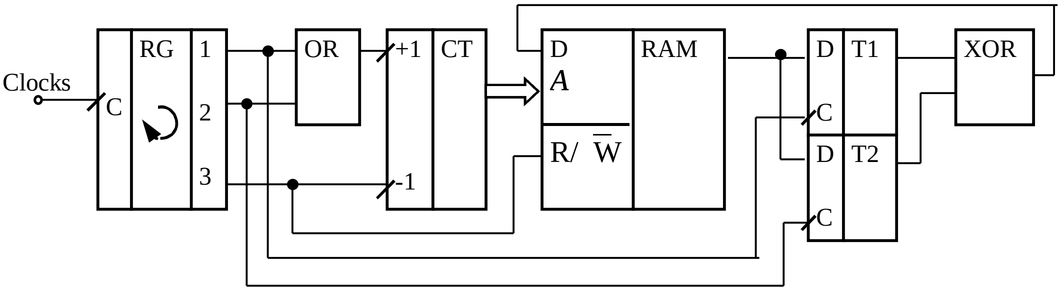


Fig.2. Scheme of PRT system and the flow diagrams for the case $\varphi(x)= 1+x+x^2$.

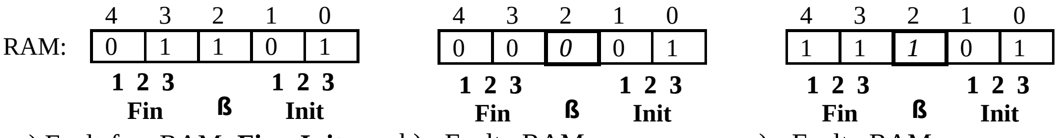
The scheme in fig.2, where: RG - rotate shift register, log.0 on the background of log.1, CT – up (+1) and down (-1) counter, T1 and T2 - D-flip-flops; can be used as a prototype to implement a real PRT system

The analyzed example of the pseudo-ring system design will be a start point to analyze the quantitative measure of the method.

### III. Pseudo-ring test quality evaluation

The goal of PRT is to verify the absence of RAM cell-faults of known type. Test iteration is the basis of PRT. Because a coincidence of any state of virtual LFSR with a faulty cell under test iteration may take place, then exists a risk to accept the faulty RAM as a faulty-free one. On the other hand, the fault *undetected* in test iteration can be *detected* in other one. Remind that the test iterations are distinguished by their initial state of virtual automaton.

Let's consider an example. Take pseudo-ring experiment for a trivial case, *r*=2. The essence of the test experiment is to carry out the test iterations with the goal to achieve the expected level of PRT quality. Let the capacity of RAM be *N*+*r*. Now, model the test experiment, where the test iteration will be performed for all single stuck-at cell-faults. Each test experiment starts with own initial state of virtual automaton. For the fixed initial state the cell of RAM will be set in one of the faulty state - s@0 or s@1.



| | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| RAM: | 0 | 1 | 1 | 0 | 1 |

| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| 0 | 0 | *0* | 0 | 1 |

| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| 1 | 1 | *1* | 0 | 1 |

a) Fault-free RAM: **Fin**= **Init**

b) Faulty RAM - cell 2 stuck at 0: **Fin**≠ **Init**

a) Faulty RAM - cell 2 stuck at 1: **Fin**≠ **Init**

Fig.3. The example of "filling" the RAM cell after the transition of virtual automaton; initial state of automaton is **Init**= 01.

The criterion of fault detection is the inequality of initial **Init** and final **Fin** states of the simulated LFSR. The example of test iteration is show in fig.3: in fig.3, b) is an example when the fault is *detected* and in fig.3, c) - the fault is *undetected*.

The test iterations are carried out until don't remain the undetected faults. In the table 1 the results of test iteration are illustrated.

Table 1

The results of test experiment in the case φ($x$)= 1+$x$+$x^2$.

| Addr. cell \ Init | Type of fault | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Stuck at 0 | | | | | Stuck at 1 | | | | |
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| 01 | ▨ | | ▨ | ▨ | | | ▨ | | | ▨ |
| 10 | | ▨ | ▨ | | ▨ | ▨ | | | ▨ | |
| 11 | ▨ | ▨ | | ▨ | ▨ | | | ▨ | | |

▨ - Detect; □ - Undetected.

From the table 1 it can be seen that test iterations detects the equal number of faults. This results in that the table 1 can be identified as a Boolean lattice of the test iteration results partition. This combinatorial model shows that after the first test iteration will be detected 4/7 of the faults list, after second - 2/7 and after third - 1/7 from remained. The study of such correspondences for diverse values of $r$ show that under certain initial conditions there are an optimal (minimal) set of test iterations which cover all single stuck-at cell-faults. It has been established that *the minimal number of test iterations equals to* $r$+1. In this conditions the value of test quality-*resolution R*, will follow the next low:

$$R(r) = \frac{1}{\left|\mathbf{BL}_{r+1}\right| - 1} \sum_{a \in \mathbf{BL}_{r+1}} F(a) \quad \text{or} \quad R(r) = \frac{1}{2^{r+1} - 1} \sum_{i=1}^{r+1} 2^{r-i+1}. \qquad (1)$$

Moreover, the analogous low was deduced for the general case too, when the binarity of cell is $m$>1. In this important for practice case the resolution of PRT will be determined by the expression:

$$R(r,m) = \frac{2^m - 1}{2^{m(r+1)} - 1} \sum_{i=0}^{r} \left(2^m\right)^i. \qquad (2)$$

Remark that the expressions (1) and (2) take place *just* for the optimal set of test iterations, i.e. for $r$+1 iterations with predefined initial states of virtual automaton. Also, was established that *the optimal set of test iterations detect any combination of RAM cell-faults.*

The further analysis of the detection proprieties of PRT has proved that *to achieve the maximal value of resolution (=1) it is sufficient to use the 2-stage LFSR.* This is the basis for the assertion that *the complexity of pseudo-ring testing is the order O(9N),* i.e. *just* it's needed 3 test iteration in each of them the 3-steped subiteration literately is performed.

So, the scheme represented in fig.2 is the *basis* for designing the PRT RAM system with 1 bit cell. It remains invariant for all one-bit cell RAM: the change of array capacity just leads to corresponded change of counter binarity, which depends *logarithmically* on the RAM capacity. Supposing, that 1 bit position of register *Rg* and so one of counter *CT*, and flip-plops T1 and T2 too, is equivalent to one cell, then the overhead, for example, for 1 Megabit RAM will be about 25 units ($<2^5$), will make up about 0.004 % (percent) of array memory capacity!

## IV.Conclusion

In this work a new technique, named Pseudo-Ring Testing, to design the built-in self of RAM tools is considered. PRT follows the tradition of BIST technologies. The main characteristic of the proposed technique is the application of inherent parts of unit to organize the procedure of RAM testing. In the same time the hardware overhead necessary to implement the method makes up thousandth part of percent from the capacity of cell array.

The scheme of PRT system is invariant with reference to the type of RAM and it is applicable for testing the array with cell of any binarity. It's needed to remark that the testing is performed at the working frequencies of UUT. It's just needed 3 test iteration to provide the full pseudo-ring test. The test iteration is based on emulation of linear automaton by components of RAM and transition of it through the cell memory space.

The complexity of method is order $O(9N)$, where $N$= capacity RAM array. If to embed some inherent part of PRT architecture and to supplement at least two RAM cell with the performance to simultaneous read, then the complexity can be reduced to order $O(3N)$.

The mathematical model to estimate the PRT quality was elaborated. On the basis of this model was founded the necessary conditions to make the test iterations to achieve full (100%) fault coverage, i.e. to cover all single stuck-at cell-faults. Also, it was proved that PRT allows to testing any combination of stuck-at cell-faults.

Thanks to existence of the virtual feedback between the cells, caused by the structure of the simulated linear automaton, there is the reason to assume that the coverage of the inter-influence faults is high too, including the coupling faults, at least, for the length *r* - the capacity of the automaton memory.

From the framework of this article more tasks and problems remain unconsidered, in particular, those that concern the technical features of design the PRT system for word-oriented RAM. The author hopes this material will find proper attention and comprehension among the designers of ATE, associations and the organizations (committees) for standardization and applications of the diagnostic technologies, methods and tools.

## Bibliography

[1] Litikov I.P. (1986), "Ring-like testing of Digital circuits" in *Measurement*, vol.4, No.**1**, pp.2-6.

[2] R. David, A. Fuentes and B.Courtois (1989), "Random Pattern Testing Versus Deterministic Testing of RAM's" in *IEEE Transactions on Computer*, Vol.38, No. 5, pp.637-650.

[3] A.K. Sharma and R.Rajsuman (1997*), Semiconductor Memories - Technology, Testing, and Reliability*, IEEE Press, 596 P.

[4] A.J. van de Goor (1991), *Testing Semiconductor Memories - Theory and Practice*, John Wiley & Sons, 536 P.