Structure Chart of Pseudo-Ring Testing and Evaluation of its Algorithmic and Hardware Complexity

Serghei Grițcov
Technical University of Moldova
Chisinau, Moldova
gritscov@gmail.com

***Abstract* — This paper presents different modifications of structure of pseudo-ring testers and determination of its algorithmic an hardware complexity, as well as the types of faults the testers allow to detect.**

***Key Words* — Longitudinal pseudo-ring testing, transversal pseudo-ring testing, faults.**

I. Introduction

The increase of size of digital memory used in the technical devices leads to the necessity to form tests that would allow to reduce time of testing [1]. Otherwise, testing a memory, which size may be units or dozens of gigabytes, can take days, weeks or even months [2]. Another important parameter of any tester is its hardware complexity, reduction of which allows to reduce complexity of implementation of the tester, as well as to save hardware resources. Known march tests allowed to reduce algorithmic complexity to a minimum in comparison to the classical methods of testing. However, the hardware complexity of march tests remains rather high.

Pseudo-ring (π-) testing is based on passing through memory cells of a LFSR (linear feedback shift register), which combines a test generator and a result analyzer. It is possible to implement a LFSR using the resources of the memory device [3-5]. These features allow to reduce several times the hardware complexity of the tester in comparison with march tests.

The paper considers possible variations of the implementation of π-testers, of which algorithmic and hardware complexity for detection of stuck-at faults was determined.

II. Implementation Of Pseudo-Ring Testing

Implementation of pseudo-ring testing is based on passing of a LFSR through the memory. The LFSR can be implemented using resources of the digital memory. In this case, the process of testing can be described by steps shown in Fig. 1.

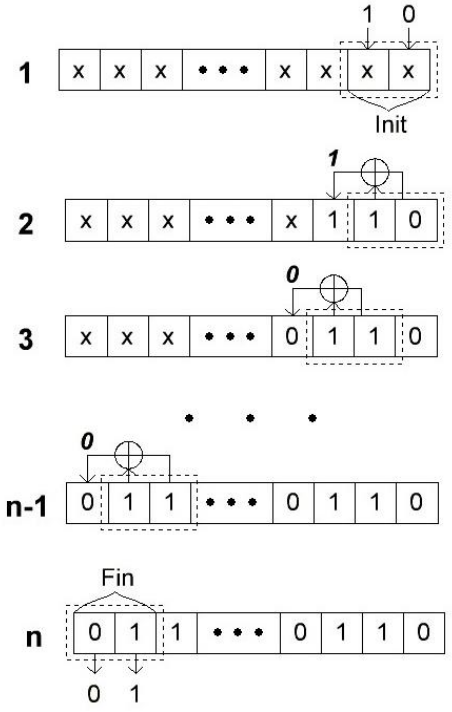

Fig. 1. Iterations of π-testing.

In the first step, the initial value is written into the first two memory cells (step 1). Then, the values of the first two cells are read, the new value is calculated and it is written into the third memory cell (step 2). After that, LFSR is shifted to the position of the second and third memory sell and the actions performed in the second step are repeated (step 3). Thus, LFSR passes through the memory cells. In the last step (step n) the values of LFSR are read and compared with expected value, which can be calculated beforehand [6]. As a result, the algorithmic complexity of the tester will be equal to 3n (n - number of memory cells): $\Updownarrow\{r_i, r_{i+1}, w_{i+2}(r_i \oplus r_{i+1})\}$. Implementation of the LFSR is based on the memory cells and will only require an additional logic gate XOR. This method is called longitudinal π-testing.

It is possible to reduce the time of testing by reducing the algorithmic complexity by changing the principle of passing of the LFSR through the digital memory cells (Fig. 2).

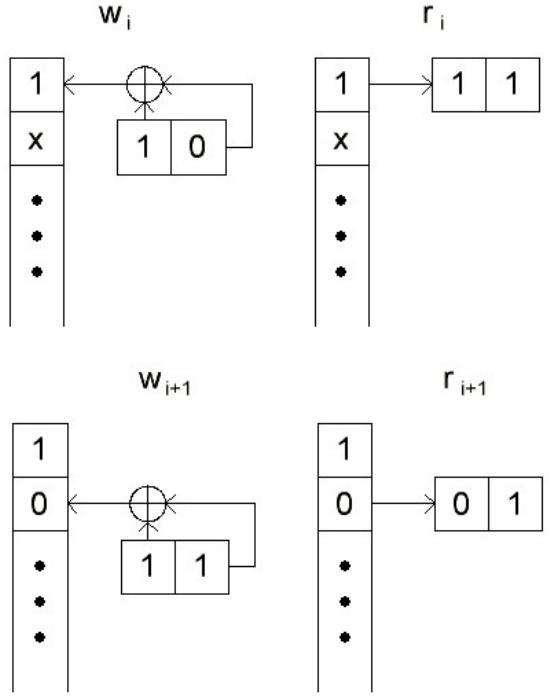

Fig. 2. Iterations of transversal π-testing.

This method is called transversal π-testing. In this case, after the initial state of the LFSR is written, reading of value of the LFSR is performed, a new value is calculated, which is written in the first memory cell. In the next step the value of the first cell is read and written in the LFSR, in which the previous data is shifted and the new value written. Further operations are repeated until the LFSR passes through all the cells of the digital memory. This method of testing is presenter in Fig. 3.

The continuous line in Fig. 3 shows the first write operation into the memory, and the dashed line indicates subsequent operation of reading this value from the memory. In this case, the algorithmic complexity will be equal to 2n (n - number of memory cells): $\Updownarrow\{w_i, r_i\}$.

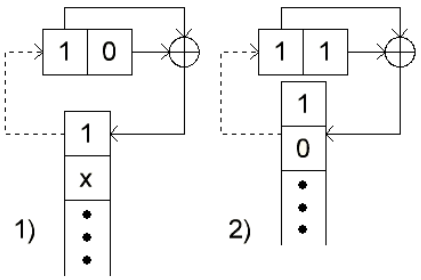

Fig. 3. Iterations of transversal π-testing.

Implementation of the LFSR will require additional register and logic gate XOR. Thus, the presence of extra register and some modifications in the method allowed to reduce of the algorithmic complexity of the testing by a factor of 1.5.

As shown in [6] the methods of π-testing detect 100% of stuck-at faults. Algorithmic complexity of transversal π-test is lower, which makes it more effective in testing the digital memory.

CONCLUSIONS

Transversal π-test detects all stuck-at faults, and its algorithmic complexity is 1.5 times lower than the longitudinal π-testing. To detect coupled faults the use of the transversal π-testing becomes ineffective. To detect coupled faults the value of the victim cell must be read after a value into the aggressor cell was written. The longitudinal π-testing performs reading of a cell after the neighbor cell is written, which makes it suitable for detection coupled faults as well as stuck-at faults.

REFERENCES

[1] Van de Goor, A., Gaydadjiev, G., Hamdioui, S., Memory testing with a RISC microcontroller, ComTex, Gouda, Netherlands, IEEE Int. Test Conf., pp. 214-219, 2010.

[2] С. Грицков, Г. Сорокин, "ОЦЕНКА КАЧЕСТВА ПСЕВДО-КОЛЬЦЕВОГО ТЕСТИРОВАНИЯ УСТРОЙСТВ ОПЕРАТИВНОЙ ПАМЯТИ," CONFERINŢA STUDENŢILOR, MASTERANZILOR, DOCTORANZILOR ŞI COLABORATORILOR, FRT, CHIŞINĂU, 2012.

[3] D. Bodean, Gh. Bodean, "Pseudo-Ring Testing Schemes and Algorithms of RAM Built-In and Embedded Self-Testing," Jun 21 2011 cs.AR arXiv:1106.3677v1

[4] Gh. Bodean, D. Bodean, A. Labunetz, "New Schemes for Self-Testing RAM," Design, Automation and Test in Europe, Munich, Germany. March 7, 2005. pp: 858-859

[5] G. Bodean, "PRT: Pseudo-Ring Testing – A Method for SelfTesting RAM", IEEE-TTTC Int. Conf. On Automation, Quality and Testing, Robotics: AQTR 2002 (THETA 13), Tome 1, Cluj-Napoca, Romania, May 2002, pp. 295-300.

[6] S. Grițcov, A. Ghincul, Gh. Bodean "AUTOTESTAREA PSEUDOINELARĂ A MICROCONTROLERELOR NANOSATELITULUI SATUM," ICTEI-2012, vol. 2, 2012, pp. 260–267.

[7] Mikitjuk, V.G., Yarmolik, V.N., van de Goor, A.J., RAM testing algorithms for detection multiple linked faults, Byelorussian State Univ., Minsk, IEEE Int. Test Conf., pp. 435 – 439, 1996.

[8] Park, Youngkyu, Park, Jaeseok, Han, Taewoo, Kang, Sungho, An Effective Programmable Memory BIST for Embedded Memory, IEICE Transactions on Information and Systems, Volume E92.D, Issue 12, 2009, pp.2508-2511.

[9] D. Bodean, Gh. Bodean, Wajeb Gharibi, "Pseudo-Ring Testing Schemes and Algorithms of RAM Built-In and Embedded Self-Testing," 2011.

[10] Gh. Bodean, Wajeb Gharibi. Analyzing of Pseudo-Ring Memory Self- Testing Schemes with Algorithms, International Journal of Distributed and Parallel Systems 2012.