UDK 004.33

DOI: 10.15222/TKEA2018.5-6.03

S. S. GRITCOV, Ph.D. G. F. SOROKIN, Ph.D. T. V. SHESTACOVA

Moldova, Chisinau, Technical University of Moldova
E-mail: gritscov@gmail.com

PSEUDO-RING TESTS RESOLUTION FOR DYNAMIC SINGLE FAULTS IN WORD-ORIENTED MEMORY

This paper presents single dynamic faults and methods for their detection. Such dynamic faults as dRDF, dDRDF and dIRF are considered in detail. Also, pseudo-ring testing and the principles of single dynamic faults detecting by pseudo-ring tests are considered. The paper presents the resolution determination results for pseudo-ring tests in relation to these faults in the word-oriented memory. Also, a comparative analysis of the pseudo-ring tests with known March tests is done. The results show that pseudo-ring tests with an algorithmic complexity of (30—60)N, where N is the number of all memory cells, can cover from 75 to 100% of all single dynamic faults. This advantage allows using pseudo-ring tests as an alternative to existing classical and March tests.

Keywords: dynamic single faults, pseudo-ring testing, iteration.

A lot of projects based on digital processing systems use memory devices for data storing. Using a memory device, we must consider the possibility of faults appearing in this memory. These faults appear because of various defects that occur during the memory device operation [1]. Dynamic faults are a type of faults which can appear in the memory device [2]. To detect them, it is necessary to develop specific tests which are presented in different publications [3, 4]. These tests are March tests. The main idea of March testing is to write a value in every memory cell, read these cells and compare with written value. If there are no faults in memory, read and written values must coincide. A March test represent a sequence of write and read operations of 0 and 1 values in memory cells [5]. For example, March test MATS represent write 0 and read 0 from all memory cells and write 1 and read 1 from all memory cells. As we can see March tests are very simple. A feature of the March tests is the low algorithmic complexity, and as the result — the highest speed of the test execution. Also, March tests have good fault coverage. These advantages account for March tests being used nowadays [6]. On the other hand, March tests have a high hardware complexity.

An alternative testing method is pseudo-ring testing [7]. The main idea of pseudo-ring testing consists in LFSR (Linear Feedback Shift Register) passing through all memory cells and comparing final LFSR state with the expected one, which is described in this paper in more detail. For March test implementation we need to use a data generator for writing 0 and 1 in memory cells and result analyzer for reading memory cells. In pseudo-ring testing the data generator and result analyzer are merged in one LFSR, which provides a significant hardware resources reducing. Also, pseudo-ring tests can detect dynamic faults in bit-oriented memory [8]. Modern memory is word-oriented and when moving from bit-oriented memory to word-oriented memory, pseudo-ring testing needs changes in LFSR structure and LFSR initial states [9]. It means that new pseudo-ring tests must be determined for word-oriented memory. Only a few examples of pseudo-ring tests for a few static single and coupled faults in word-oriented memory are presented in literature [9, 10], which means that pseudo-ring testing is poorly understood for word-oriented memory and needs more research in this field.

The main purpose of this paper is to determine pseudo-rings tests for word-oriented memory and to determine resolution of these tests for single dynamic faults. Also, it is very important to compare the proposed pseudo-rings tests with the widely used March tests and to analyze algorithmic complexity of these tests and to analyze their resolution, also known as fault coverage.

Single dynamic faults are presented in detail in section 1 of this paper. In section 2 the main idea of the pseudo-ring methods is considered. In section 3 a few pseudo-ring tests for word-oriented memory are proposed, their resolution with respect to single dynamic faults is determined and a comparative analysis of pseudo-ring tests with March tests is carried out.

1. Dynamic memory faults

There are large number of faults occurring in memory. The most typical classification of known faults is presented in Fig. 1 [11].

In **Fig. 1** dynamic faults are highlighted. They are usually divided into single dynamic and coupled dynamic faults. Let's consider the single dynamic faults [12].

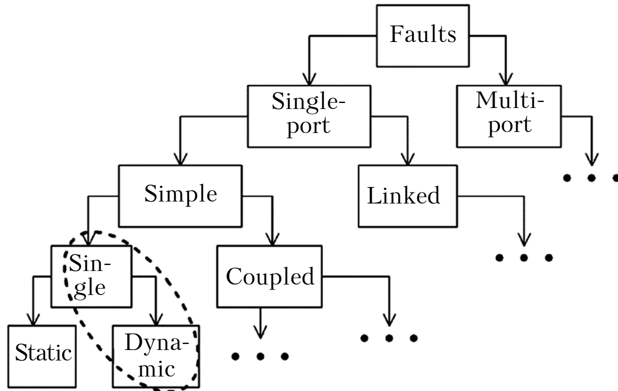

Fig. 1. Typical digital memory faults classification

Dynamic Read Disturb Faults (**dRDF**). A cell has dRDF if a read operation immediately followed by a write operation ($w0$, $r0$ or $w1$, $r1$) in this cell cause an incorrect logical state at the output. There are four types of dRDF ($0w0r0$, $0w1r1$, $1w0r0$, $1w1r1$).

Dynamic Deceptive Read Disturb Faults (**dDRDF**). A cell has dDRDF if a read operation immediately followed by a write operation ($w0$, $r0$ or $w1$, $r1$) in this cell returns the correct logical value, but the content of the cell changes. There are four types of dDRDF.

Dynamic Incorrect Read Fault (**dIRF**). A cell has dIRF if a read operation immediately followed by a write operation ($w0$, $r0$ or $w1$, $r1$) in this cell returns the incorrect logical value, but the content of the cell remains correct. There are four types of dIRF.

To detect these faults, it is necessary to perform certain operations, in this case, perform a read operation, which follows immediately after the write operation to a specific cell. This sequence of operations is performed by default in pseudo-ring testing. Let us consider this type of testing in more detail.

2. Pseudo-ring testing and detecting principle for dynamic single faults

Pseudo-ring testing is based on the LFSR (Linear Feedback Shift Register) passage through memory cells. LFSR is a test data generator and result analyzer. In the process of the LFSR passing, the memory cells are overwritten and read, which ensures the fault detection. To provide higher resolution, unlike pseudo-random testing, the LFSR repeatedly passes through memory cells a certain number of times, which is called iteration. The structure of the LFSR is defined by the structure of an irreducible polynomial, and the iterations differ in that the LFSR has different initial values at the beginning of each iteration. The structure of the LFSR, the initial states of the LFSR and the direction of the test determine the three main parameters that define the pseudo-ring test [13]. An example of iterative pseudo-ring testing is presented in **Fig. 2**.

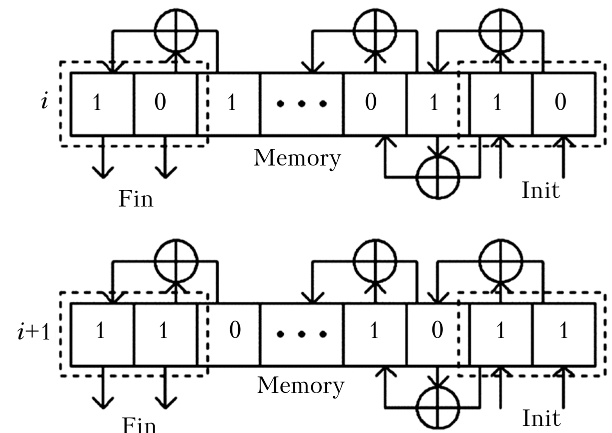

Fig. 2. An example of iterative pseudo-ring testing

In Fig. 2 '*i*' means the iteration number, 'Init' stands for the initial state of the LFSR, and 'Fin' is the final state of the LFSR. The final state of the LFSR is compared with the expected one [13]. If these two states are different, then a fault has been detected. Also, the final state of the LFSR based on a certain irreducible polynomial will coincide with the initial one if no faults were detected, provided that the number of memory cells is equal to the irreducible polynomial period.

For word-oriented memory tester, the LFSR structure is given by the structure of an irreducible polynomial over extended Galois fields [14]. This study deals with 4-bit memory testing. In this case, we can apply the irreducible polynomial $g(z)=1+2z+2z^2$ over the extended Galois field $(2^2)^4$ with the generating polynomial $p(x)=1+x+x^4$. The period of the polynomial will be equal to

$$T = (2^n)^m - 1, \quad (1)$$

where n is the bit width of the memory cells, m is the number of digits in the LFSR [14].

In our case, the period will be equal to

$$T = (2^4)^2 - 1 = 255.$$

Thus, for 4-bit memory, the value of the memory cells with addresses 0 and 1 should be equal to memory cells with addresses 255 and 256.

Let's consider in detail the principle of single dynamic faults detection by pseudo-ring tests.

3. Pseudo-ring tests resolution determining

Pseudo-ring tests resolution is determined practically, i. e., by simulating the pseudo-ring testing in the faulty memory, since it is rather difficult to analytically determine the resolution. For simulating pseudo-ring testing in the faulty memory, we can apply the algorithm described in detail in [15]. This simulation system can be implemented in Python 3. This programming language is the easiest to learn and has many auxiliary libraries, which greatly simplifies and speeds up the development process of any system.

Let us consider a test with the following parameters:

— the structure of LFSR is determined by the irreducible polynomial $g(z) = 1+2z+2z^2$ over the extended Galois field $(22)^4$ with the generating polynomial $p(x) = 1+x+x^4$;

— the initial state of LFSR:

- 0000 0000
- 0000 0001
- 0000 0010
- . . .
- 1000 0000

— test direction: ascending addresses.

We shall call this test PS01 (pseudo-ring 01), where 0 is the first iteration, and 1 is all iterations in which there is 1 in the initial state. This test can also be written as PS [*w*0 *w*1], where *w*0 provides writing to all cells of logical 0 (the first iteration of the test), and *w*1 provides writing to all cells of logical 1 (all remaining iterations of the test). The logical 1 with the displacement in the initial states of the LFSR, passing through all the bits of the LFSR (all test iterations except the first), provides writing to all memory cells of the logical 1, which is well described in the source [9].

When determining the resolution of the PS01 test with respect to dRDF, dDRDF and dIRF, it is important to use the pseudo-ring tester architecture with external loading of the initial state and unloading of the final state of the LFSR.

PS01 test based on the architecture shown in **Fig. 3** can be called PS01e (pseudo-ring 01 external). Also, this test can be written as PS[*w*0 *w*1]e. This tester architecture provides the condition that each tested memory cell will be read two times. This feature will be important for detecting single dynamic faults. Let us proceed to the consideration of the PS01e resolution definition in relation to these faults.

The example of PS01e executing in memory with dRDF is presented in **Fig. 4**.

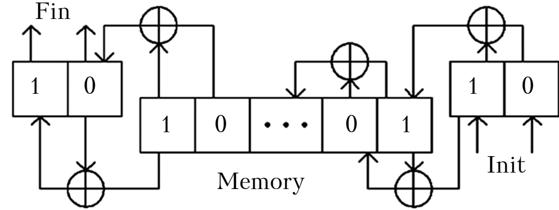

Fig. 3. Pseudo-ring tester architecture for dynamic faults detecting

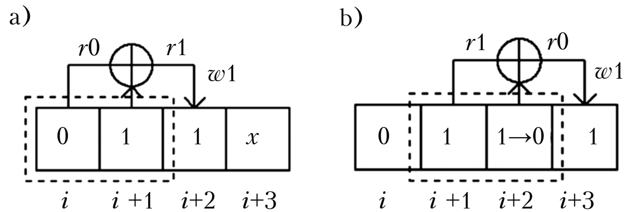

Fig. 4. Example of dRDF detecting by PS01e

Fig. 4, *a* shows an example when, during the testing process, the LFSR is at the memory cells with addresses *i*, *i*+1. At the next step of testing, the LFSR is at the memory cells with addresses *i*+1, *i*+2 (Fig. 4, *b*). In this example, a dRDF fault occurs in the cell *i*+2. The value 1 should be recorded in this cell. Fig. 4, *b* shows that when reading cell *i*+2 the contents of this cell is inverted and instead of reading 1, 0 is read. The value written to the cell *i*+3 will be 1. This change in the cell *i*+2 also leads to the fact that the final LFSR state does not coincide with the expected one and, as a result, the fault will be detected.

The result of PS01e simulating in memory with dRDF is presented in **Fig. 5**.

In Fig. 5, *a,* the executed iterations are numbered 0, 1, 2, ... 8; 51.76 means that 51.76% of possible faulty cells with dRDF00 where detected by iteration 0; 76.08 means that 76.08% of all possible faulty cells with dRDF00 where detected by iterations 0 and 1; 93.24 means that 93.24% of all possible faulty cells with dRDF00 where detected by all 9 iterations of the test PS01e.

Fig. 5, *b, c, d* present PS01e fault coverage for dRDF10, dRDF01 and dRDF11 respectively.

Also, Fig. 5 shows that PS01e detects all dRDF01, since in this case the next condition in each memory cell is provided: 0*w*1*r*1, that is, the reading 1 after the transition from 0 to 1. The remaining types of dRDF cannot be fully detected by this test. The algorithmic complexity of the PS01e test is $3{\cdot}9N = 27N$, where N is the number of the memory cells. To detect dRDF00, we need to duplicate the operation *w*0. To detect dRDF10, we can add operation *w*0 at the end of the PS01e test. Thus, the PS[*w*0 *w*0 *w*1 *w*0]e or PS0010e test can be applied to detect all dRDF except dRDF11.

a)		b)		c)		d)	
0:	51.76	0:	50.39	0:	0.0	0:	0.0
1:	76.08	1:	50.39	1:	50.2	1:	0.0
2:	76.08	2:	62.84	2:	75.29	2:	25.1
3:	81.86	3:	75.98	3:	87.84	3:	37.65
4:	84.61	4:	85.29	4:	94.12	4:	50.2
5:	86.96	5:	91.27	5:	97.25	5:	59.61
6:	89.31	6:	95.2	6:	98.82	6:	67.45
7:	91.27	7:	96.96	7:	99.61	7:	73.73
8:	93.24	8:	98.33	8:	100.0	8:	78.82

Fig. 5. PS01e resolution with respect to dRDF00 (*a*), dRDF10 (*b*), dRDF01 (*c*) and dRDF11 (*d*)

0000 0001
0000 0001
0000 0010
0000 0010
. . .
1000 0000
1000 0000

Fig. 6. Initial LFSR states with double operation *w*1

Algorithmic complexity of this test will be equal to 3·11*N* = 33*N*. To detect dRDF11, it is necessary to duplicate the operation *w*1. The initial LFSR states for *w*1 will look as is shown in **Fig. 6**.

In this case, the algorithmic complexity of the test will increase significantly: 3·19*N* = 57*N*, but all dRDF will be detected. This test could be written as PS[*w*0 *w*0 2*w*1 *w*0]e or $PS001_20e$.

Let us consider PS01e resolution in relation to other single dynamic faults. The example of PS01e executing in memory with dDRDF is presented in **Fig. 7**.

Fig. 7, *a* shows an example when, during the testing process, the LFSR is at the memory cells with addresses *i*+1, *i*+2. At the next step of testing, the LFSR is at the memory cells with addresses *i*+2, *i*+3 (Fig. 7, *b*). In this example, a dDRDF fault occurs in cell *i*+2. The value 1 should be recorded in this cell.

Fig. 7, *b* shows that during the reading of *i*+2 cell, the content of this cell is inverted, but the correct state 1 is read. The written value to the *i*+3 cell will be 0. When the *i*+2 cell will be read again, the read value will not be 0 and as a result, not the expected value 1, but the value 0 will be written to the *i*+4 cell. This change will lead to the fact that the final LFSR state does not coincide with the expected one and, as a result, the fault will be detected.

The result of PS01e simulating in memory with dDRDF is presented in **Fig. 8**.

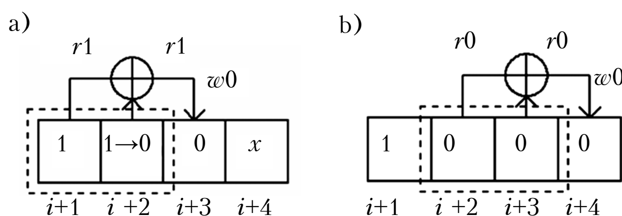

Fig. 7. Example of dDRDF detecting by PS01e

In Fig. 8, *a,* the executed iterations are numbered 0, 1, 2, ... 8; values in the right part mean how many of possible faulty cells with dDRDF00 where detected. For example: 89.9 means that 89.9% of all possible faulty cells with dDRDF00 where detected by 7 iterations of the PS01e. Fig. 8, *b, c, d* present PS01e fault coverage for dDRDF10, dDRDF01 and dDRDF11 respectively.

Fig. 8 shows that PS01e detects all dDRDF01. We can use the PS[*w*0 *w*0 *w*1 *w*0]e or PS0010e test to detect all dDRDF except dDRDF11, and we need to use the $PS001_20e$ test to detect all dDRDF.

a)		b)		c)		d)	
0:	49.12	0:	48.03	0:	0.0	0:	0.0
1:	75.29	1:	48.03	1:	50.3	1:	0.0
2:	75.29	2:	60.63	2:	75.39	2:	25.1
3:	81.27	3:	73.43	3:	87.89	3:	37.65
4:	84.51	4:	84.06	4:	94.09	4:	50.2
5:	87.84	5:	91.04	5:	97.24	5:	59.61
6:	89.9	6:	94.98	6:	98.82	6:	67.45
7:	91.57	7:	97.24	7:	99.61	7:	73.73
8:	92.75	8:	98.62	8:	100.0	8:	78.82

Fig. 8. PS01e resolution with respect to dDRDF00 (*a*), dDRDF10 (*b*), dDRDF01 (*c*) and dDRDF11(*d*)

ISSN 2225-5818

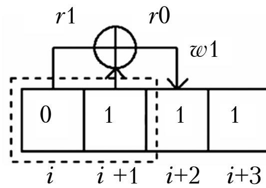

Fig. 9. Example of dIRF detecting by PS01e

	a)	b)	c)	d)
0:	49.9	50.2	0.0	0.0
1:	74.71	50.2	50.2	0.0
2:	74.71	62.35	75.29	25.1
3:	81.57	75.69	87.84	37.65
4:	84.41	85.59	94.12	50.2
5:	87.45	91.76	97.25	59.61
6:	89.8	95.69	98.82	67.45
7:	91.27	98.43	99.61	73.73
8:	93.24	99.41	100.0	78.82

Fig. 10. PS01e resolution with respect to dIRF00 (*a*), dIRF10 (*b*), dIRF01 (*c*) and dIRF11(*d*)

Let us consider PS01e resolution in relation to dIRF. The example of PS01e executing in memory with dIRF is presented in **Fig. 9**.

Fig. 9 shows an example when, during the testing process, the LFSR is at the memory cells with addresses *i*+1, *i*+2. In this example, a dIRF fault occurs in the *i*+2 cell. The value 1 should be recorded in this cell. During the *i*+2 cell reading, the contents of this cell remains correct, but the incorrect state 0 is read. The value 1 will be written to the *i*+3 cell and the final state of the LFSR will not coincide with the expected one, and as a result the fault will be detected.

The result of PS01e simulating in memory with dIRF is presented in **Fig. 10**.

In Fig. 10, *a,* the executed iterations are numbered 0, 1, 2, … 8; values in the right part mean how many of possible faulty cells with dIRF00 where detected. For example: 81.57 means that 81.57% of all possible faulty cells with dIRF00 where detected by 4 iterations of the PS01e. Fig. 10, *b, c, d* present PS01e fault coverage for dIRF10, dIRF01 and dIRF11 respectively.

Fig. 10 shows that PS01e detects all dIRF01. To detect all dIRF except dIRF11, we can use test PS[*w*0 *w*0 *w*1 *w*0]e or PS0010e, and to detect all dIRF, we must apply the $PS001_20e$ test.

Results of single dynamic faults detection by different tests are presented in the **Table**.

Test resolution for single dynamic faults

Test	dRDF	dDRDF	dIRF	Algorithmic complexity
PS01e	1/4	1/4	1/4	27*N*
PS0010e	3/4	3/4	3/4	33*N*
$PS001_20e$	4/4	4/4	4/4	57*N*
March U	2/4	0/4	2/4	13*N*
March LA	2/4	2/4	2/4	22*N*

The Table presents the resolution and algorithmic complexity of the pseudo-ring tests and March tests for 4-bit memory [11, 12]. For example, '2/4' means that a test fully detects two subtypes of faults from the four possible ones, *N* indicates the number of memory cells. According to the results presented in the Table, pseudo-ring tests have higher algorithmic complexity, but at the same time they detect single dynamic faults well.

Conclusion

According to the research results presented in the paper, it can be argued that the pseudo-ring tests have a high resolution with respect to single dynamic faults in the word-oriented memory. The PS0010e test presented in this paper detects 75% of all dynamic single faults with an algorithmic complexity of 33*N*, where *N* is the number of all memory cells. By comparison, the March LA test detects only 50% of all dynamic single faults with algorithmic complexity 22*N*. Thus, the algorithmic complexity of the pseudo-ring tests is higher than that of the March tests, but the detecting ability of the pseudo-ring tests is high, and their algorithmic complexity remains linear in comparison with classical testing methods.

REFERENCES

1. Borri S., Hage-Hassan M., Girard P., Pravossoudovitch S., Virazel A. Defect-oriented dynamic fault models for embedded-SRAMs. *The Eighth IEEE European Test Workshop*, 2003, pp. 23–28. https://doi.org/10.1109/ETW.2003.1231664

2. Van de Goor A. J., Al-Ars Z. Functional memory faults: a formal notation and a taxonomy. *18th IEEE VLSI Test Symposium*, Montreal, Que, 2000, pp. 281–289. https://doi.org/10.1109/VTEST.2000.843856

3. Dilillo L., Girard P., Pravossoudovitch S. et al. Dynamic read destructive fault in embedded-SRAMs: analysis and march test solution. *ETS* 2004, pp. 140–145. https://doi.org/10.1109/ETSYM.2004.1347645

4. Dilillo L., Girard P., Pravossoudovitch S. et al. Efficient test of dynamic read destructive faults in SRAM memories. *6th IEEE Latin American Test Workshop*, 2005, pp. 40–45.

5. Yarmolik S., Zankovici A., Ivaniuk A. *Marshevye Testy Dlja Samotestirovanja OZU* [Marching Tests for RAM Self-Testing]. Minsk: Izdatelskiy tsentr BGU, 2009, 270 p. (Rus)

6. Kusuma L. Performance analysis of bist algorithm for RAMs. *International Journal for Technological Research in Engineering*, 2015, vol. 2, iss. 7, pp. 674–678.

7. Om. S. Al-Mushayt. Concept of pseudo-ring self-test of the RAM. *Science International*, 2014, vol. 26(2), pp. 755–758.

8. Bodean G. C. Compact testing of RAM: Schemes, tools and results. *Automation, Quality and Testing, Robotics*, Cluj-Napoca, Romania, 2008, pp. 227–232.

9. Grițcov S. Algorithmic complexity of pseudo-ring testing for stuck-at faults. *The 5th International Conference «Telecommunications, Electronics and Informatics»*, UTM, Chisinau, 2015, pp. 75–76.

10. Grițcov S. П-testirovanie s primeneniem LFSR na osnove privodimykh polinomov, *The 5th International Conference «Telecommunications, Electronics and Informatics»*, UTM, Chisinau, 2015, pp. 130–131.

11. Hamdioui S., Al-Ars Z., van de Goor A. J. et al. Dynamic faults in random-access-memories: concept, fault models and tests. *Journal of Electronic Testing: Theory and Applications*, 2003, vol. 19, iss. 2, pp. 195–205. http://dx.doi.org/10.1023/A:1022802010738

12. Hamdioui S., Al-Ars Z., van de Goor A. J. Testing static and dynamic faults in random access memories. *Proceedings of the 20th IEEE VLSI Test Symposium.* 2002, pp. 395–400. – https://doi.org/10.1109/VTS.2002.1011170

13. Grițcov S., Sorochin Gh., Șestacova T. Pseudo-ring testing of the FPGA memory using software Nios processor. *The 6th International Conference «Telecommunications, Electronics and Informatics»*, UTM, Chisinau, 2018, pp. 328–331.

14. Bodean Gh. *Coduri nonbinare corectoare de erori* [Non-binary codes for error correction]. Chisinau, Editura «Tehnica-UTM», 2010, 550 p. (Md)

15. Grițcov S., Lazar D., Sorochin Gh. [Development of a simulation system for determining-tests resolution]. *The 5th International Conference «Telecommunications, Electronics and Informatics»*, UTM, Chisinau, 2015, pp. 132–133. (Rus)

Received 16.11 2018

DOI: 10.15222/TKEA2018.5-6.03
УДК 004.33

С. С. ГРІЦКОВ, Г. Ф. СОРОКІН, Т. В. ШЕСТАКОВА

Молдова, м. Кишинів,
Технічний Університет Молдови
E-mail: gritscov@gmail.com

ЗДАТНІСТЬ ПСЕВДОКІЛЬЦЕВИХ ТЕСТІВ ВИЯВЛЯТИ ДИНАМІЧНІ ОДИНОЧНІ НЕСПРАВНОСТІ У СЛОВООРІЄНТОВАНІЙ ПАМ'ЯТІ

У даній роботі представлені одиночні динамічні несправності цифрової пам'яті і методи їхнього виявлення. Детально розглянуто такі динамічні несправності, як динамічне руйнувальне читання (Dynamic Read Destructive Fault — dRDF), динамічне уявне руйнуюче читання (dynamic Deceptive Read Destructive Fault — dDRDF) і динамічне некоректне читання (dynamic Incorrect Read Fault — dIRF).

Існують різні методики виявлення даних несправностей. Найбільш популярною методикою є маршові тести, оскільки таким тестам властива лінійна алгоритмічна складність, яка визначає число операцій, що проводяться над кожною коміркою пам'яті в процесі тестування. Дана особливість визначає мінімальний час виконання тесту.

Альтернативою маршовим тестам є псевдокільцеві тести. Їхньою перевагою над іншими існуючими методами тестування є низька апаратна складність. Псевдокільцеві тести мають лінійно алгоритмічну складність. Вони непогано досліджені відносно класичної однобітної пам'яті, але що стосується словоорієнтованої пам'яті, тут дослідження псевдокільцевого тестування практично відсутні.

У даній роботі розглядаються псевдокільцеві тести по відношенню до одиночних динамічних несправностей на прикладі чотирьохбітної пам'яті. Представлена загальна ідея ітеративного псевдокільцевого тестування. Детально розглянуто принцип виявлення dRDF, dDRDF і dIRF, а також представлено здатність псевдокільцевих тестів виявляти дані несправності.

Показано, що псевдокільцеві тести мають не гіршу здатність виявляти вказані несправності, ніж маршові тести. За результатами роботи видно, що псевдокільцеві тести з алгоритмічною складністю (30—60)N, де N — кількість всіх осередків пам'яті, дозволяють покрити від 75 до 100% всіх одиночних динамічних несправностей, що вказує на можливість їхнього використання як альтернативи існуючим тестам.

Ключові слова: динамічні одиночні несправності, псевдокільцеве тестування, ітеративність.

С. С. ГРИЦКОВ, Г. Ф. СОРОКИН, Т. В. ШЕСТАКОВА

Молдова, г. Кишинев, Технический Университет Молдовы
E-mail: gritscov@gmail.com

СПОСОБНОСТЬ ПСЕВДОКОЛЬЦЕВЫХ ТЕСТОВ ОБНАРУЖИВАТЬ ДИНАМИЧЕСКИЕ ОДИНОЧНЫЕ НЕИСПРАВНОСТИ В СЛОВООРИЕНТИРОВАННОЙ ПАМЯТИ

В данной работе представлены одиночные динамические неисправности цифровой памяти и методы их обнаружения. Детально рассмотрены такие динамические неисправности, как динамическое разрушающее чтение (dynamic Read Destructive Fault — dRDF), динамическое мнимое разрушающее чтение (dynamic Deceptive Read Destructive Fault — dDRDF) и динамическое некорректное чтение (dynamic Incorrect Read Fault — dIRF).

Существуют различные методики обнаружения данных неисправностей. Наиболее популярной методикой являются маршевые тесты, поскольку они обладают линейной алгоритмической сложностью, определяющей число операций, проводимых над каждой ячейкой памяти в процессе тестирования. Данная особенность определяет минимальное время выполнения теста.

Альтернативой маршевым тестам являются псевдокольцевые тесты. Их преимуществом перед другими существующими методами тестирования является низкая аппаратная сложность. Псевдокольцевые тесты неплохо изучены по отношению к классической однобитной памяти, однако их исследования касательно словоориентированной памяти практически отсутствуют.

В данной работе рассматриваются псевдокольцевые тесты по отношению к одиночным динамическим неисправностям на примере четырехбитной памяти. Представлена общая идея итеративного псевдокольцевого тестирования. Подробно рассмотрен принцип обнаружения dRDF, dDRDF и dIRF и представлена способность псевдокольцевых тестов обнаруживать данные неисправности.

Показано, что псевдокольцевые тесты обладают не худшей обнаруживающей способностью, чем маршевые тесты. По результатам работы видно, что псевдокольцевые тесты с алгоритмической сложностью (30—60)N, где N — количество всех ячеек памяти, позволяют покрыть от 75 до 100% всех одиночных динамических неисправностей, что указывает на возможность их использования как альтернативы существующим тестам.

Ключевые слова: динамические одиночные неисправности, псевдокольцевое тестирование, итеративность.

Cite the article as:
Gritcov S. S., Sorokin G. F., Shestacova T. V. Pseudo-ring tests resolution for dynamic single faults in word-oriented memory. Tekhnologiya i Konstruirovanie v Elektronnoi Apparature, 2018, no. 5–6, pp. 3–9. http://dx.doi.org/10.15222/TKEA2018.5-6.03