Algorithmic Complexity of Pseudo-Ring Testing for Stuck-at Faults

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

***Abstract* — This paper describes calculation of algorithmic complexity of pseudo-ring testing obtained by simulation of stuck-at faults and realization of pseudo-ring testing of 4-bit and 8-bit memory. The method is designed for both bit- and word-oriented memory and it is invariant to the width of data bus of the memory.**

***Key Words* — Pseudo-ring testing, algorithmic complexity, stuck-at faults, bit width invariance, generator polynomial**

I. Introduction

In today's electronic devices more and more frequently 16 and 32-bit digital memory is used. The use of digital memory with a bigger address and data bus, especially in computers, allows to solve more complex problems and solve them much (hundred times or more) faster than when using a 4 or 8-bit memory. Increase of data width of digital memory leads to certain difficulties in testing this memory [1].

Known methods of testing (ring tests, march tests etc.) are designed to test bit-oriented memory. The use of classical march tests designed to test bit-oriented memory does not allow to detect a certain number of faults. This feature makes it necessary to elaborate additional tests that take into account architecture of memory devices. As a result, the transition from bit-oriented memory to word-oriented leads to an increase in algorithmic complexity of these tests in units and tens of times, which is unacceptable, given the volume of modern storage devices (from one to dozens of gigabytes).

Described in this paper method of pseudo-ring (π-) testing is a further development of ring tests [2]. This method is designed for testing both bit-oriented and word-oriented memory and it is invariant for the width of memory cells. As a result, the switch from bit-oriented memory to a word-oriented memory or the change of data bus width does not require the need to recompose tests.

II. Π-Testing for Stuck-at Faults

The π-testing intended for bit-oriented memory an LFSR (linear feedback shift register) is used, which is based on an irreducible polynomial. Along with the choice of polynomial the two other parameters of the test are the direction of iterations and values in the LFSR in the beginning of each iteration [2]. Invariance of the π-test regarding to the data bus width consists in the fact, that it is necessary to change only the structure of the LFSR when switching from one data bus width to another. Therefore, when switching to a word-oriented memory, as the basis of an LFSR, a polynomial over extended Galois fields is used. The direction of iterations and LFSR initial values remain the same as in the bit-oriented memory test. For detection stuck-at fault s@1 it is enough to set the initial value of LFSR, equal to zero. Then into all the memory bits will be written '0' in case if no faults are present. To detect the stuck-at fault s@0 into all memory bits must to be written '1'. This operation is not possible, as the LFSR based on an irreducible polynomial whose values will always change and cannot constantly contain '1' in all the bits. Therefore, it is necessary to compose a test for detection of faults s@0.

Below there is an example of a test for 4- and 8-bit memory. For 4-bit memory LFSR is constructed on the bases of polynomial $g(z)=1+2z+2z^2$ over extended Galois field $GF(2^4)$ with a generator polynomial $p(x)=1+x+x^4$. Direction of iterations of π-testing – in ascending addresses and initial states – running unit. To determine the resolution of the test the system shown in Fig. 1 was implement.

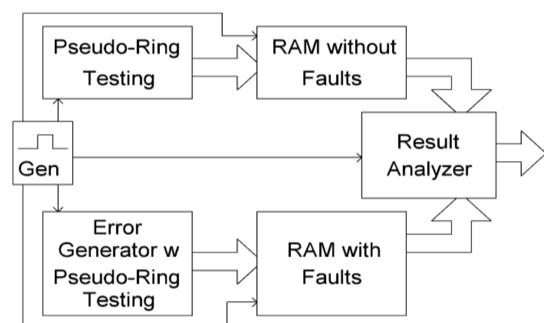

Fig. 1. Block diagram for determination of resolution of π-test for stuck-at faults.

The blocks shown in Fig. 1, will determine the resolution π-test for detection of stuck-at faults. Block "Pseudo-Ring Testing" represents means necessary for the implementation of the pseudo-ring testing: registers, counters and combinational circuits carrying out the load of the original value of the test, memory addressing, writing values into memory and reading from the memory, determining the direction of iterations, as well as the number of test combinations. Block "RAM without Faults" – a unit of digital memory. Block "Error Generator with Pseudo-Ring Testing" represents the means necessary for the implementation of the pseudo-ring testing and generation of faults; this block should allow to simulate faults sequentially in

every memory cell. "RAM with Faults" – a unit of digital memory, in which will be imitated some faults.

"Result Analyzer" compares the final values obtained after each iteration of the memory with and without particular type of fault. "Control Logic" – block that synchronizes the work of all other units of the device.

Testing of 4-bit memory with imitation of stuck-at zero faults gave results presented in Tab. 1.

TABLE I. RESULTS OF SIMULATION OF S@0 FAULTS FOR 4-BIT MEMORY

№	Init	s@0, %
1	0000 0001	50,05
2	0000 0010	(+25) 75,05
3	0000 0100	(+12,6) 87,65
4	0000 1000	(+6,3) 93,95
5	0001 0000	(+3,17) 97,12
6	0010 0000	(+1,61) 98,73
7	0100 0000	(+0,83) 99,56
8	1000 0000	(+0,44)100

The results presented in Tab. 1 display pattern: each test iteration finds half of the number of faults detected in the previous iteration of the test. To confirm the statement above the formula of the Boolean lattice model can be applied [3]:

$$F(a) = \sum_{i=1}^{r+1} \frac{2^{r-i+1}}{2^{r+1}-1} . \qquad (1)$$

If r (the amount of iterations) =8, (i - № of iter.) we obtain:

$$\frac{2^8}{2^9-1}+\frac{2^7}{2^9-1}+\frac{2^6}{2^9-1}+\frac{2^5}{2^9-1}+\frac{2^4}{2^9-1}+\frac{2^3}{2^9-1}+\frac{2^2}{2^9-1}+\frac{2^1}{2^9-1}+\frac{2^0}{2^9-1}=$$

$$=\frac{1}{2^9-1}(2^8+2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0)=1$$

Algorithmic complexity of such a test is (8n*3) (8 – the number of iterations, n – the number of memory cells, 3 – the number of operations performed over one cell: $\Updownarrow\{r_i, r_{i+1}, w_{i+2}(r_i \oplus r_{i+1})\}$).

For detection of stuck-at faults s@1 is sufficient to carry out one iteration of the test with all initial values of the LFSR equal to 0. Then algorithmic complexity of the test for detection of all stuck-at faults equals (24n+3n): 24n – for detection of s@0 and 3n – for detection of s@1.

To test an 8-bit memory LFSR is constructed on the bases of polynomial $g(z)=1+z+2z^2$ over extended Galois field $GF(2^8)$ with a generator polynomial $p(x)=1+x+x^6+x^7+x^8$. Direction of iterations of π-testing – in ascending addresses and initial values – running unit.

Simulation can be realized on the basis of the flowchart shown in Fig. 1, same as in the example of bit-oriented memory. Testing of 4-bit memory with imitation of stuck-at zero faults gave results presented in Table 2. Results in Table 2 satisfy the conditions in equation 1. The algorithmic complexity of such a test is (16n*3) (16 – number of iterations, n - the number of memory cells).

TABLE II. RESULTS OF SIMULATION OF S@0 FAULTS FOR 8-BIT MEMORY

№	Init	s@0, %
1	00000000 00000001	48,23
2	00000000 00000010	(+23,93) 72,16
3	00000000 00000100	(+12,94) 85,1
4	00000000 00001000	(+6,2) 91,3
5	00000000 00010000	(+2,24) 93,54
6	00000000 00100000	(+1,81) 95,35
7	00000000 01000000	(+1,33) 96,68
8	00000000 10000000	(+1,07) 97,75
9	00000001 00000000	(+0,82) 98,57
10	00000010 00000000	(+0,51) 99,08
11	00000100 00000000	(+0,32) 99,4
12	00001000 00000000	(+0,24) 99,64
13	00010000 00000000	(+0,16) 99,80
14	00100000 00000000	(+0,1) 99,90
15	01000000 00000000	(+0,06) 99,97
16	10000000 00000000	(+0,03)100

For the detection of stuck-at faults s@1 it is sufficient to carry out one iteration of the test with all initial values of the LFSR equal to 0. Then algorithmic complexity of the test for detection of all stuck-at faults equals (54n+3n): 54n – for detection of s@0 and 3n – for detection of s@1.

CONCLUSIONS

For detection of stuck-at fault s@1 it is enough to set the initial values of LFSR, equal to zero. In determining the resolution of π-test of 4 or 8-bit memory for detection of stuck-at zero fault s@0 was revealed the following pattern: one of the shortest tests for detection stuck-at zero faults is running one. Algorithmic complexity of this test depends linearly on the number of bits in the LFSR, which depends on the number of bits in one memory cells. And the main feature of the π-test is that the transition from 4 to 8-bit memory requires only the change in structure of the LFSR, which confirms the invariance the π-testing for the bit width of memory cells.

REFERENCES

[1] A.J. van de Goor, I.B.S. Tlili, "March tests for word-oriented memories," Design, Automation and Test in Europe, 1998, pp. 501–508.

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

[3] O. Eriomenco, Pseudo-Ring Testing of Random Access Memory, Acta Academia, Chisinau, 2001.