Multiple Test Set Generation Method for LFSR-Based BIST

Youhua Shi
Ohtsuki & Yanagisawa Lab
Waseda University, Japan
shi@yanagi.comm.waseda.ac.jp

Zhe Zhang
National ASIC System Engineering Research Center
Southeast University, China
zhang@seu.edu.cn

Abstract -- In this paper we propose a new reseeding method for LFSR-based test pattern generation suitable for circuits with random pattern resistant faults. The character of our method is that the proposed test pattern generator (TPG) can work both in normal LFSR mode, to generate pseudorandom test vectors, and in jumping mode to make the TPG jump from a state to the required state (seed of next group). Experimental results indicate that its superiority against other known reseeding techniques with respect to the length of the test sequence and the required area overhead.

1. Introduction

With growing complexity of integrated circuits and systems, the cost of testing has become ever more significant. BIST (Built-In Self Test) is increasingly being applied as an effective means to reduce the cost of testing [1].

The main components of a BIST scheme are Test Pattern Generator (TPG), which produces the test patterns applied to the circuit under test (CUT), and output response analyzer (ORA), which compacts the response of the CUT to a single pattern called signature and compares it with the signature of the fault-free circuit [2]. Usually for any successful BIST scheme, complete fault coverage (FC), minimal test application time, area overhead, and test data storage as well as minimal performance degradation and at-speed testing should be taken into consideration.

In BIST scheme, linear feedback shifter register (LFSR) is commonly used as a TPG because LFSR can generate sequences with good random property with little area overhead. The main objective that has to be satisfied in classical BIST is to get the highest fault coverage with the shortest test sequence length. However, in circuits with random pattern resistant faults, high fault coverage cannot be achieved with an acceptable test length [3]. In this case reseeding is a technique that has been proposed to solve this problem.

In [4] a test-per-clock scheme based on a modified design of an LSSD-based LFSR is described. The proposed scheme is capable of changing seeds by applying a pair of clock pulses at the time of change. The seeds cannot be predetermined, they are randomly selected and they have the property of being uniformly distributed over the entire LFSR pattern space. In [5] a test-per-scan technique is presented where an LFSR is used to generate pseudorandom and deterministic patterns that are encoded as seeds. In [6] a scheme using a shift register driven by an LFSR (LFSR/SR) is proposed. And recently, a test-per-clock technique is presented in [7] that, based on Genetic Algorithms, computes the initial values for several general functional modules and LFSRs, so that they are able to produce test patterns with complete fault coverage.

However all of these techniques suffer from the same problem. In the case of circuits with many random pattern resistant faults, a large number of seeds must be used, and then the overall area overhead may be very large.

In this paper we present a new reseeding technique for LFSR-based test pattern generation, which can generate effective vectors in normal mode and produce the seed of the next group in jumping mode. The proposed technique achieves complete fault coverage with shorter test length and less area overhead.

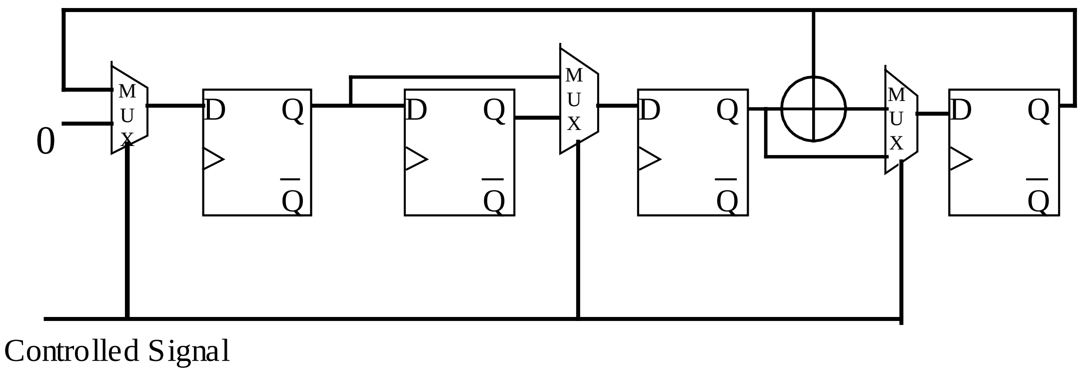

Fig. 1. Example of Proposed TPG Structure

The rest of this paper is organized as follows: section 2 and 3 present the structure and the reseeding algorithm for the proposed TPG respectively. Then in section 4 the effectiveness of the proposed technique is evaluated with experimental results and comparisons with previous works. At last conclusions are given in section 5.

2. Proposed Structure

The proposed TPG structure is presented in Fig. 1. The proposed TPG can be called as mixed-mode LFSR, which composes of a normal LFSR and a number of additional multiplexers distributed among the stages of the register.

One characteristic of this TPG structure is that it can generate pseudorandom test vectors as general LFSR in normal mode, while at the same time it can jump from one vector to another in jumping mode.

The TPG structure is originated from a kind of general LFSR, called as a Shift Division Circuit (SDC), which is shown as in Fig. 2.

Due to the fact that, for each LFSR, the corresponding transition matrix can be used to represent the behavior of the LFSR, therefore we can implement the transition logic in our proposed TPG by altering the structure of SDC-LFSR.

In this case, the corresponding transition matrix T_c is of the following form:

$$T_c = \begin{bmatrix} 0 & 1 & 0 & \Lambda & 0 \\ 0 & 0 & 1 & \Lambda & 0 \\ \Lambda & \Lambda & \Lambda & \Lambda & \Lambda \\ 0 & 0 & 0 & \Lambda & 1 \\ c_0 & c_1 & c_2 & \Lambda & c_{n-1} \end{bmatrix}$$

Where c_0 , c_1 … c_{n-1} are the coefficients of the corresponding characteristic polynomial SDC-LFSR. Thereby the test vectors generated by SDC-LFSR satisfy the following equation:

$$V_{T_c,t+1} = V_{T_c,t} \times T_c$$

Where $V_{T_c,t}$ is the current state of the LFSR and $V_{T_c,t+1}$ the next state. In [8] it has shown that if matrix T and T_c are similar matrix, in other words $T = B^{-1} \times T_c \times B$, given the matrix T and T_c, the solution of the equation above is the same as the equation below:

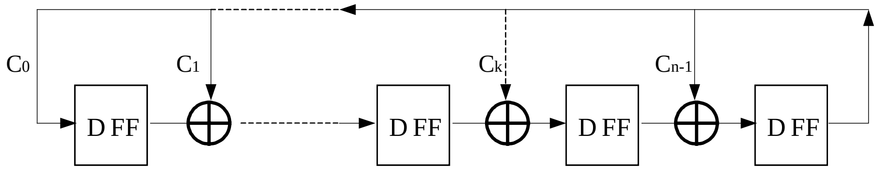

Fig. 2. Shift Division Circuit - a canonical form LFSR

$$K \times B_i = 0 \qquad (1)$$

Where matrix B_i is the column vector of B, which can be expressed following:

$$B_i = [b_{1,1}, b_{1,2}, \ldots b_{1,n}, b_{2,1}, \ldots b_{2,n}, \ldots, b_{n,1}, \ldots b_{n,n}]^T$$

So matrix can be denoted as:

$$K = \begin{bmatrix} T^t & I_n & & \\ & T^t & I_n & \\ & & \ddots & I_n \\ I_{c,0} & I_{c,1} & \ldots & I_{c,n-1} + T^t \end{bmatrix}$$

The null space of equation (1) can be represented as:

$$N = \begin{bmatrix} \left([T^t]^{-1}\right)^{n-1} \\ \left([T^t]^{-1}\right)^{n-2} \\ \vdots \\ \left([T^t]^{-1}\right) \\ I_n \end{bmatrix}$$

Any matrix B_i that verifies any linear combination of the column vectors of matrix *N* will satisfy the equation (1). A solution for B_i will satisfy the following equation:

$$B_i = \begin{bmatrix} \left([T^t]^{-1}\right)^{n-1} \\ \left([T^t]^{-1}\right)^{n-2} \\ \vdots \\ \left([T^t]^{-1}\right) \\ I_n \end{bmatrix} \times C$$

Where C is a column vector whose elements are not all zero. Then we can get the equation as below:

$$\begin{bmatrix} V_{T,0} \\ V_{T,2} \\ \vdots \\ V_{T,2m} \end{bmatrix}_{(m\times n)} \times T_{(n\times n)} = \begin{bmatrix} V_{T,1} \\ V_{T,3} \\ \vdots \\ V_{T,2m+1} \end{bmatrix}_{(m\times n)} \qquad (2)$$

Where $V_{T,2i}$ is the end vector of the i^{th} group, while $V_{T,2i+1}$ is the first vector of the next group. Therefore according to Equation (2), we can work out the transition matrix *T*, and make the LFSR turn from the current group to the next group at ease.

For example, to a test pattern generate by a 4-tuples LFSR with the transition matrix as $T_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$. If three groups are divided and we want the test patterns to jump from $V_1 = [1 \quad 0 \quad 1 \quad 0]$ to $V_2 = [0 \quad 1 \quad 1 \quad 1]$, and from $V_3 = [0 \quad 1 \quad 1 \quad 0]$ to $V_4 = [0 \quad 0 \quad 0 \quad 1]$, according to Equation (2), we can get the transition matrix *T* as $T = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ and the TPG structure is shown in Fig. 1. Here note that in the transition matrix *T*, '1' means XOR gate is desired in the corresponding site of the jumping structure. Therefore we can combine the normal mode and jumping mode into a single LFSR through control the multiplexers to change the transition matrix through input controlled signal. Also if we can make the transition matrix

T more similar to T_c, some multiplexers can be saved. In general, if we keep the group number into a limited amount, we have more extent to select the transition matrix.

In the following section, we will discuss how to select the test groups and reseeding.

3. Reseeding Algorithm

In the test sequence produced by LFSR, not all the test vectors contribute to the fault coverage. So in order to remove the non-detecting vectors from the test sequence, we group the test vectors into several groups, and then in each group it is comprised of successive vectors produced by LFSR.

Obviously, it is impossible to seek the optimum groups in all possible cases for computational complexity, however because that annealing algorithm can seek the global optimization, the simulated annealing algorithm is applied in this paper.

Supposed that the number of detecting vectors, which is contributive to the fault coverage, is *n*, the optimum groups of *n* detecting vectors are sought based on the following algorithm. By selecting different number of detecting vectors, the optimum solution of number of vectors and groups can be achieved while the required fault coverage is guaranteed.

(1) Configurations: $\Omega = (G_1, G_2 \ldots G_k)$, $G_1 \ldots G_k$ are k possible groups of *n* detecting vectors.

(2) The cost function:

$$f(V_1 \ldots V_k) = \sum_k L_k^{1/2} + (FC_d - FC_s) \times P$$

Where L_k is the shortest distance between the k^{th} and its neighboring group, FC_d the attainted fault coverage, FC_s the fault coverage the present vectors can detect, P is the penalty coefficient with a high value that prevents the emergence of undesirable solution that the present test vectors cannot satisfy the required fault coverage.

(3) Move set: Select the “boundary” of detecting vectors (the first or the last vector in a random group of detecting vectors) to be non-detecting vector, and add a new detecting vector adjacent directly to the boundary of another random group of detecting vectors, thus generating new groups $G_1 \ldots G_m$ and new cost function $f(G_1 \ldots G_k)$.

(4) Cooling schedule: $t_{k+1} = \alpha t_k, \alpha < 1$, where the initial starting temperature t_0 and cooling rate α are determined empirically to give good results.

(5) Stopping criterion: the solution of M successive temperatures does not change, or the temperature t reaches the terminative temperature t_f.

4. Experimental Results

In order to evaluate the effectiveness of the proposed TPG structure, we performed the simulations on ISCAS ’85 and the combinational part of ISCAS’89 benchmarks circuits.

In Tables 1 and 2 we compare the proposed method against the methods presented in [7] and [9]. Among the results given in Table 1, a dash (-) in the comparison tables means that no results has been provided by the authors of the referenced paper for the corresponding circuit. Also because there is not enough information in [7], therefore the area overhead of the control logic cannot be calculated, and is represented with letter ‘N’ in Table 2.

In Table 1 we compare the three techniques with respect to the number of seeds and test vectors they require for fully testing of the CUT. From Table 1 we can see that, in the majority of cases, the proposed technique requires less seeds and shorter test sequences than those of [7] and [9], which is because of the efficiency of the simulated annealing algorithm.

The area overhead comparisons are given in Table 2. Here we do not consider the cost of modifying a register to a shift register and the hardware overhead is given in terms of gate equivalents, assuming that 1 gate equivalent corresponds to a 2-input NAND gate and each memory cell of a ROM equivalent to 0.25 gate. From Fig. 1, we can observe that compared with normal LFSR, the proposed TPG is only added some multiplexers and XOR gates, so its area overhead is little.

Table 1. Seeds and test vectors comparisons for complete fault coverage

Circuits	Proposed		Twisted-ring Counters [9]			LFSR-based TPG of [7]		
	Seeds	Test Vectors	Seeds	Test Vectors	Vectors Reduction (%)	Seeds	Test Vectors	Vectors Reduction (%)
c2670	12	3712	70	58930	93.7	34	10206	63.6
c7552	26	8437	107	76447	89	-	-	-
s420	16	2568	8	10816	76.3	10	10843	76.3
s641	10	1302	9	11458	88.6	7	2430	46.4
s713	9	1979	8	11296	82.5	8	2759	28.3
s820	12	647	-	-	-	35	527	-22.8
s838	24	1435	29	15742	90.9	44	9273	84.5
s953	7	3114	6	10810	71.2	5	4834	35.6
s1196	10	6372	12	11152	42.9	5	18776	66.1
s1238	8	5918	9	10864	45.5	6	7713	23.3
s1423	13	929	-	-	-	5	1308	30
s5378	11	4763	1	10642	55.2	-	-	-

Table 2. Hardware overhead comparisons

Circuits	Number of Primary Inputs	Proposed Technique (gate equiv.)	Twisted-ring Counter of [9]		LFSR-based TPG of [7]	
			ROM bits (gate equiv.)	Control logic (gate equiv.)	ROM bits (gate equiv.)	Control logic
c2670	233	508	4078	65	1981	N
c7552	207	462	5537	65	-	-
s420	34	132	68	42	85	N
s641	54	67	122	46	95	N
s713	54	71	108	42	108	N
s820	23	132	-	-	201	N
s838	66	376	479	54	726	N
s953	45	83	68	42	56	N
s1196	32	37	96	42	40	N
s1238	32	62	72	42	48	N
s1423	91	58	-	-	114	N
s5378	214	97	54	41	-	-

From Table 1 and 2, we can observe that the area overhead mainly lies on the increased amount the XOR gates and multiplexers in LFSR. So it is very important to select appropriate amount of groups of test patterns and make ‘1’ in transition matrix *T* as few as possible.

We should finally mention that the CPU time of the simulated annealing algorithm is not very long. Although the simulated annealing algorithm is a global optimized algorithm, our algorithm did not need excessive time to work out the ideal results. For large circuits we just need a few hours and for small circuits less than one hour. The simulated annealing algorithm is implemented in C++ language. And all simulations are performed on a Sun SPARC Ultra 60 workstation. However there are no results represented in [7] and [9], we did not make any comparison in this paper.

5. Conclusion

Reseeding has been proposed as an effective technique for testing circuits with random pattern resistant faults, since it can achieve complete fault coverage with an acceptable number of test vectors. In this paper a new reseeding technique for LFSR-based test pattern generation was proposed. The character of the proposed TPG is that we integrate general LFSR and jumping structure into the same LFSR. Experimental results on ISCAS’85 and part of ISCAS’89 benchmark circuits have shown the efficiency of the proposed TPG structure, with respect to the number of seeds and test vectors as well as the area overhead, against other previous methods.

Acknowledgements

We would like to thank Prof. Tatsuo Ohtsuki, Prof. Masao Yanagisawa, Prof. Shinji Kimura and Prof. Nozomu Togawa of Waseda University for their comments and suggestions. We also wish to thank Prof. Longxing Shi and Prof. Shengli Lu of Southeast University for their technical help.

REFERENCE

[1] H.J.Wunderlich, “BIST for systems-on-a-chip,” *Integration, The VLSI Journal*, Vol. 26, (No.1-2), Elsevier, pp. 55-78, December 1998.

[2] M.Abramovici, M.A.Breuer, and A.D.Friedman, Digital Systems Testing and Testable Design, Computer Science Press, New York, NY, 1990.

[3] E.Kalligeros, X.Kavousianos, D.Bakalis and D.Nikolos, “A New Reseeding Technique for LFSR-based Test Pattern Generation,” *International Test Conference*, pp.80-86, 2001.

[4] J.Savir and W. McAnney, “A Multiple Seed Linear Feedback Shift Register,” *IEEE Trans. on Computers*, Vol.41, No.2, pp. 250-252, February 1992.

[5] B. Koenemann, “LFSR-Coded Test Patterns for Scan Design,” *Proc. of European Test Conference*, Munich, Germany, pp. 237-242, April 1991.

[6] S.K.Mukund, E.J.McCluskey and T.R.N.Rao, “An Apparatus for Pseudo-Deterministic Testing,” *Proc. of 13th VLSI Test Symposium*, pp.125-131, April-May 1995.

[7] S. Chiusano, P.Printto and H.J.Wunderlich, “Non-Intrusive BIST for Systems-on-a-chip,” *Proc. of International Test Conference*, pp.644-651, Oct.2000.

[8] R.Li, Ch.Hu, J.Yang, Zh.Zhang, Y.Shi and L.Shi, “A new Low Power BIST Methodology by Altering the Structure of Linear Shift Registers,” IEEE Press, *The 4th International Conference on ASIC 2001*, pp.646-649, 2001.

[9] K.Chakrabarty, B.T.Murray and V.Iyengar, “Built-in Test Pattern Generation For High-Performance Circuits Using Twisted-Ring Counters,” *Proc. of 17th IEEE VLSI Test Symposium*, pp.22-27, April 1999.