Random-Pattern Coverage Enhancement for BIST

Ondrej Novak
TU Liberec, Hálkova 6, 461 17 Liberec 1, Czech Republic
e-mail: ondrej.novak@vslib.cz

Abstract
The paper presents a design method for Built-In Self Test (BIST) that uses a cellular automaton (CA) for test pattern generation. Similarly to the test pattern generators with linear feedback shift registers (LFSR) the CA can generate pseudorandom test patterns. The patterns correspond to code words or to linear combination of different code words of codes with non primitive irreducible polynomials and with higher minimal code distance of its dual code. The CA is formed by T flip-flops and does not contain any XOR in the feedback. We proposed a new scheme of BIST where the CA is formed by a modified scan chain. A number of experiments were done with ISCAS 85 and 89 benchmark circuits. The achieved fault coverage is better than that obtained with the help of patterns generated by LFSRs.

Introduction

The hardware test pattern generation is usually done by mixed mode pattern generation **[1].** It consists of generation of a given number of pseudorandom patterns and after it in exercising the CUT with deterministic test vectors. The deterministic vectors have to detect random resistant faults. We can evaluate the quality of pseudorandom testing phase by the number of undetected faults. Pseudorandom patterns are generated usually in an LFSR whose serial output feeds the scan chain of designed circuit.
In this paper we introduce a new scheme of pseudorandom pattern generation. Instead of the external LFSR we modify the scan chain in such a way that it forms a CA. We have verified that the quality of test patterns generated in the CA is better than the quality of the patterns generated in the LFSR with primitive polynomial.

Linear Code Based Pattern Generation

Let us suppose, that we have a binary linear code (n,k). Let us suppose that the code word bits are contained in a feedback shift register with local feedback loops as it is given in Fig.1. From the theory of linear codes follows that if we add mod 2 two different code words we obtain another code word. As the codes are cyclic, another code word could be obtained by shifting a code word to the right or to the left. The scheme given in Fig. 1 performs simultaneously adding mod 2 of two different code words (One code word corresponds to the present state of the shift register, the second corresponds to the code word shifted to the right, these two code words are added with the help of XORs). If the characteristic polynomial of the (n,k) code is non primitive we obtain the code word length shorter than 2^{n-k}-1. If the characteristic polynomial is irreducible there must exist a primitive element of the corresponding field. If the polynomial x+1 is a primitive element of the corresponding field the automaton from Fig.1 generates after seeding with one code word all code words of the (n,k) code.

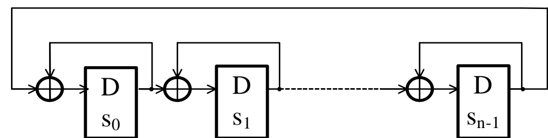

Fig. 1. *Cellular automaton created by D flip-flops performing multiplication of the polynomials corresponding to code words by the polynomial x+1.*

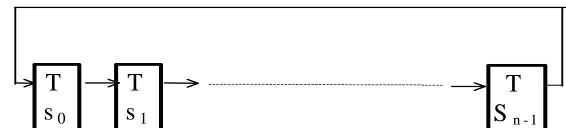

Fig. 2. *Cellular automaton created from T flip-flops performing multiplication of the polynomials corresponding to code words by the polynomial x+1.*

The automaton is a cyclic additive cellular automaton (CA) with the rule 60 for each cell **[2]**, having a regular linear structure.This CA can be further simplified. Instead of the D flip-flops and local feedback taps with XORs we can use T flip-flops because a T flip-flop performs in each clock period a XOR function of its own state and the input signal, the result is stored as a new internal state . Thus the CA given in Fig. 2 has the same function as the CA in Fig. 1, the hardware realization is substantially simpler. The proposed CA is universal in the sense that it can generate all possible code sequences of the given length without any hardware modification, we can also generate non code patterns which correspond to

XOR of two or more code words of different codes.

We can use the CA as a TPG in several ways. One possibility is shown in Fig. 3. Let us suppose that the CUT is designed in accordance with Scan Design methodology with modified Boundary Scan cells in such a way that the D flip-flops in the scan chain are replaced by T flip/flops . We separate the input and output part of the chain and we complete the CA with the global feedback. We suppose to use the CA with the number of flip-flops equal to or greater then the number of CUT inputs. In order to keep the period of the CA maximal it is necessary to choose *n* equal to the length of some code with irreducible polynomial and with the primitive element $x+1$. We have chosen the codes with lengths 17, 23, 25, 41, 47, 55, 69, 267, 765, 1687, ... The testing starts after reset and seeding the CA from the memory. At each clock cycle the CA generates a new pattern, and the CUT responds are compacted in the output part of the chain. After performing a given number of test steps the deterministic test patterns are shifted into the chain and the CUT responses are compacted in the output part of the chain. We have to keep in mind that the seeds which are stored in the memory have to be modified for both parts of testing in such a way that after shifting into the scan chain the desired test cube would be present in the chain.

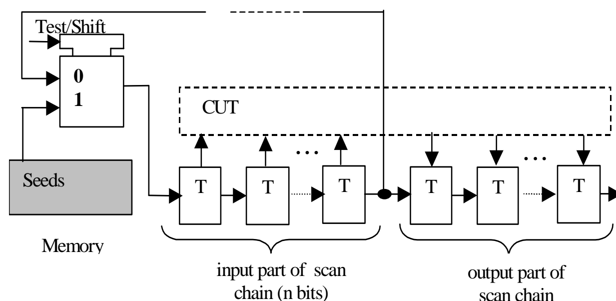

Fig. 3. *Conceptual diagram of the BIST based on the modified scan chain into a CA.*

Results Obtained

We have studied an influence of using the proposed CAs instead of using LFSRs with primitive polynomials on the first part of mixed mode pattern generation i.e. on the efficiency of pseudorandom testing. We have chosen ISCAS 85 and ISCAS 89 benchmark circuits with a significant number of random pattern resistant faults. The internal flip-flops were considered to be CUT inputs. We checked the number of non detected faults both for a 32 bit LFSR with a primitive polynomial and CA. The polynomial of the 32 bit LFSR was randomly chosen from the list of primitive polynomials, the seed of the CA was randomly chosen from the possible seeds. We simulated the number of undetected faults for at least 4 initial conditions. The best results are given in Tab. 1.

circuit	test cube length	32 bit LFSR : undetected faults	CA: undetected faults
s 9234	247	648	616
s 5378	214	38	27
s1196	32	17	11
s 15850	611	605	565
s 13207	700	624	472
c 7552	206	324	231

Table 1. *Comparison of the numbers of faults which remain undetected after 10 000 test patterns. In the case of the LFSR we have chosen 32 bit LFSRs with randomly chosen primitive polynomial. The length of CA is equal or higher than the number of circuit inputs.*

Conclusion

A linear CA with a simple structure which can be used as a TPG for BIST was designed. The simplicity of the structure is due to the existence of only one feedback with no XOR. The CA can be implemented in the CUT by replacing the D flip flops in the scan chain with the T flip/flops and by adding a feedback from the last input flip-flop to the first one.Comparing with a LFSR the CA has the following advantages: no hardware overhead (the whole TPG can be built by converting the existing scan chain), the fault coverage for ISCAS benchmark circuits is better than it is for an external LFSR. Disadvantages: All flip-flops have to be reset-able, a seed for the CA has to be calculated in more complex manner than for the LFSR, the patterns are generated in the scan chain and thus the CUT responses must not influence the flip/flops in the input part of the chain during the test.The TPG can be used advantageously in mixed mode testing where we generate a given number of pseudorandom patterns and after it we exercise the circuit with deterministic test patterns which are stored in a memory. The technique of deterministic pattern compaction as it was described in **[1]** can be used too. The research was in part supported by the research grants of the Czech Grant Agency GACR 102/98/1003 and of the Ministry of Education, Youth and Sport VS 96006.

References:

[1] KOENEMANN, B.: LFSR – coded test patterns for scan designs. Proc. Europ. Test Conf., Munich , Germany, 1991, pp. 237-242

[2] CHAUDHURI, P.P. et al.: Additive Cellular Automata Theory and Applications Volume I. IEEE Computer Society Press, 1997 .