ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, Vol 8(S2), 207–211, January 2015
DOI:.10.17485/ijst/2015/v8iS2/60479

Design of Reconfigurable 2-D Linear Feedback Shift Register for Built-In-Self-Testing of Multiple System-on-Chip Cores

Md. Shabaz*, Anand Patel, Suraj Iyer, S. Ravi and Harish M. Kittur

VLSI Division, SENSE, VIT University, Vellore, Tamil Nadu, India; mdshabaz256@gmail.com

Abstract

Due to the continual downscaling of technology, System on Chip (SoC) is becoming denser and denser with multiple IP cores within. As the number of cores within a SoC increase, so does the number of faults within the chip. Along with the designing aspect of a chip, design for testability too is a major area of concern. Testing methods like Built-In-Self-Test (BIST) allow the chip to test itself without the need for external testing equipment. Test patterns for BIST are generated using Linear Feedback Shift Register (LFSR) which produces test vectors in a pseudo random manner. This paper concentrates on improving the hardware in terms of area and number of logical gates in the 2-D LFSR used for testing an SoC with multiple IP cores so that vectors in various patterns can be generated using a single reconfigurable 2 Dimensional LFSR. The proposed technique is much more useful for testing System on a Chip with large number of cores as the same configuration network is used to test different SoC cores.

Keywords: Built-In-Self-Test (BIST), Linear Feedback Shift Register (LFSR), Reconfigurable 2-D LFSR, System-on-Chip (SoC)

1. Introduction

A VLSI chip undergoes through various processes that involve chemical, optical and metallurgical processes before it hits the market for the customers. But while going through such processes during manufacturing, there is a possibility of an error occurring in the process. This may cause the functionality of the chip to be affected and it may malfunction. Therefore each chip has to be tested before it can be shipped to the consumers. The task of detecting the faults in the chip is fulfilled by testing and the role of finding out the fault is fulfilled by diagnosis faults[1]. This makes it possible to avoid the same errors can in future productions. Each chip goes through two types of test which are parametric test and functional tests. Parametric Tests are technology dependent tests and check the current-voltage-power parameters in a circuit. They also include propagation delay tests, setup and hold time tests, access tests, rise and fall time tests. Functional Tests check the operation of a design by testing the internal nodes in the chip[2]. Functional testing is usually done by applying input test vectors and verifying the response with the expected response. Design for Testability is a technique in which extra features are added to the hardware so that the chip can test itself without the need for external stimulus to be applied at the input. Various DFT techniques like Scan Design, BIST (Built-In-Self-Test); Boundary Scan testing can be used to verify the functional authenticity of the chip.

In this paper we are using the Built in Self Test method for the functional testing of SoCs. System on Chip or SoC is an integrated chip in which multiple components (analog, digital and mixed signal) may be integrated on a single substrate[3]. As complex as this may sound, the testing of these SoCs is equally complex. Each core in these SoC may need to be tested using different test vectors to cover 100% faults. Accessing each core in a SoC is proving to be a challenge with the continuous decrease in transistor sizes as more and more cores get packed on a single SoC. The cores used in a SoC can be designed either by

*Author for correspondence

the designer or if it already exists, it can be purchased from an IP core provider. However, most IP core providers may not provide the source code to the end user. Instead a set of test vectors are provided that can be used to test that core.

Similarly, at the end of design of a SoC, we get a chip with a number of cores embedded on it and a series of test vectors for the testing of the individual cores[4]. The test vectors can be stored on a ROM in the BIST circuitry, but considering the number of test vectors required to test the whole chip, a large amount of space would be needed to store those vectors and therefore this idea is rejected. Another way to generate these test vectors is by using a Linear Feedback Shift Register (LFSR). LFSRs usually generate test patterns in a pseudorandom manner depending on the way the hardware is configured. Now if the first pattern is seeded into the LFSR, the succeeding patterns are automatically generated. This is a more efficient way to generate test patterns than storing in a ROM. Another version of the LFSR is known as the 2D LFSR[5]. This configuration has been proven to generate test patterns in a more pseudorandom manner and thereby increase the fault coverage of a core.

2. 2-D Linear Feedback Shift Register (LFSR)

In the general Linear Feedback Shift Register (LFSR), the input given to a LFSR is a linear function of previously generated output. This linear functionality of single bit input can be achieved through XOR and XNOR gates[6]. Therefore by providing proper feedback mechanism, a conventional LFSR can be implemented by using basic XOR and XNOR gates with flip-flops, which will generate pseudorandom patterns. The maximum length of the periodic sequence which will be generated is given by 2^M-1 where M is the number of stages of flip-flop used in circuitry. The randomness of generated pattern depends on initial value of flip-flops, which is called as seed, and the feedback polynomial used in design[7]. Here an LFSR goes through a finite number of possible states and after that it enters into repeating cycle. Once its initial states and generating function are fixed then the pattern generated are also determined. It cannot generate recurrent deterministic patterns which is required to detect a number of faults existing in some logic structures, which is referred to as random pattern resistance faults. A two dimensional LFSR can generate better pseudorandom patterns than the conventional LFSR and can detect random pattern resistance faults[8].

The architecture of 2-D LFSR is shown in Figure 1, in which, to optimize the circuit the test sequences are divided into group of subsequences, which will generate both the predetermined sequence of patterns and pseudorandom patterns.

The architecture of a 2-D LFSR contains the Configuration Networks (CN), Control Unit (CU), Multiplexer & Demultiplexer block and Flip Flop Array (FFA) of size NxM, where N indicates the number of input to Circuit Under Test (CUT) and M indicates the number of stages of Flip-Flops.

To decrease the hardware overhead the M should be as small as possible. Here each configuration network is designed with a particular set of XOR gates[9]. The function of MUX is to select one of the outputs of CN and then forwards further to the FFA. The output of FFA is further passed to the DEMUX and also used as the feedback to all the configuration blocks. DEMUX output is given to the respective CUT. The function of control unit is to generate the signals that allow MUX to choose the CN and as well as controlling signal for DEMUX to select the specified CUT. The internal structure of CN1 and CN2 blocks are shown in the Figure 2 and 3 respectively. To generate the test sequences with the minimum number of flip flop, the following equation is used:

$$V_i = \sum_{j=1}^{N}\sum_{k=1}^{M} a_{ijk} V_j D^k, i = 1 \rightarrow N \qquad (1)$$

Here V_j represents an N-bit vector and this circuit consist of N shift registers in which each has M stages. V_jD^k reprents the kth delay of vector V_j. Now if a_{ijk}=1,

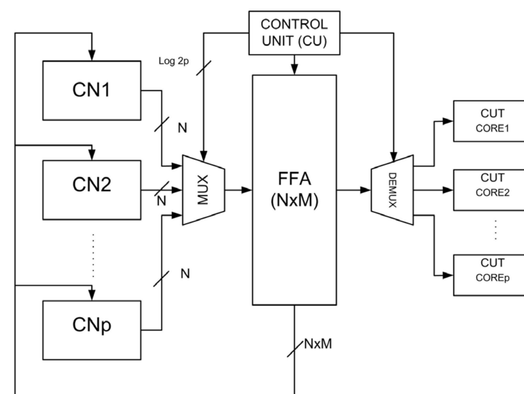

Figure 1. Architecture of a 2-D LFSR.

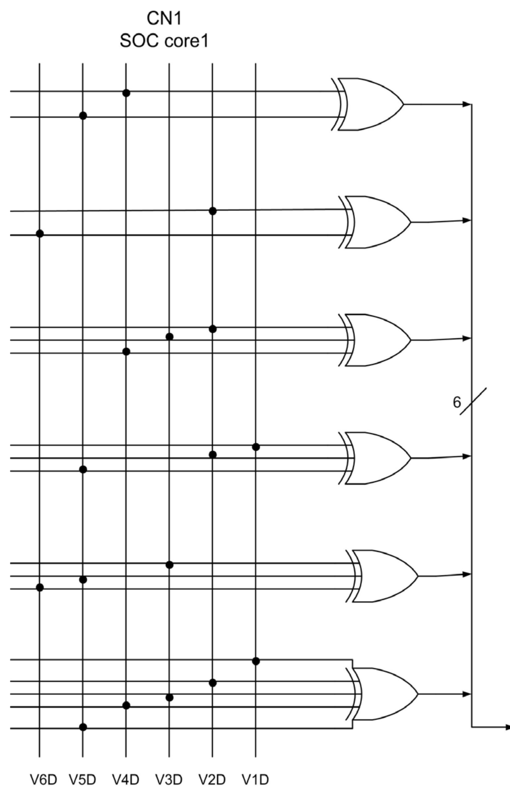

Figure 2. Structure of CN1 (SoC Core1)

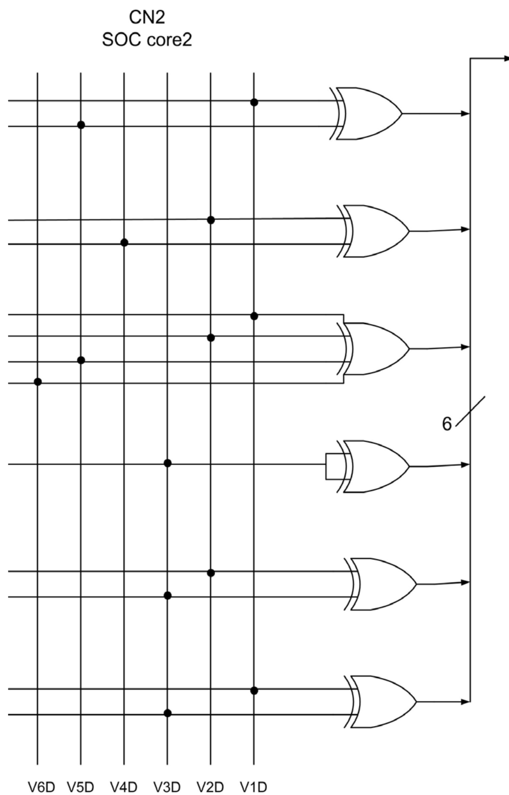

Figure 3. Structure of CN2 (SoC Core2).

then the V_jD^k will be connected to the xor gate and it will generate the V_j, otherwise the connection does not exist.

The internal structure of the CN1 and CN2 blocks are shown in the Figures 2 and 3 respectively.

3. Reconfigurable 2-D Linear Feedback Shift Register (LFSR)

The main objective here is to achieve the same goals as the 2D LFSR using lesser hardware. Assume a SoC with n IP cores. Each IP core needs its own series of deterministic test patterns to achieve 100% fault coverage. The deterministic pattern of each core may vary from that of the other. To generate these patterns it will become necessary that there be a 2D LFSR for each core, thereby a total of n cores in total. The area needed to house this configuration network will be very high. In this paper, a very basic technique is used to reduce the number of configurable 2D LFSRs from n to just 1.

The steps used to implement BIST using reconfigurable 2-D LFSR are shown in the following Figure 4.

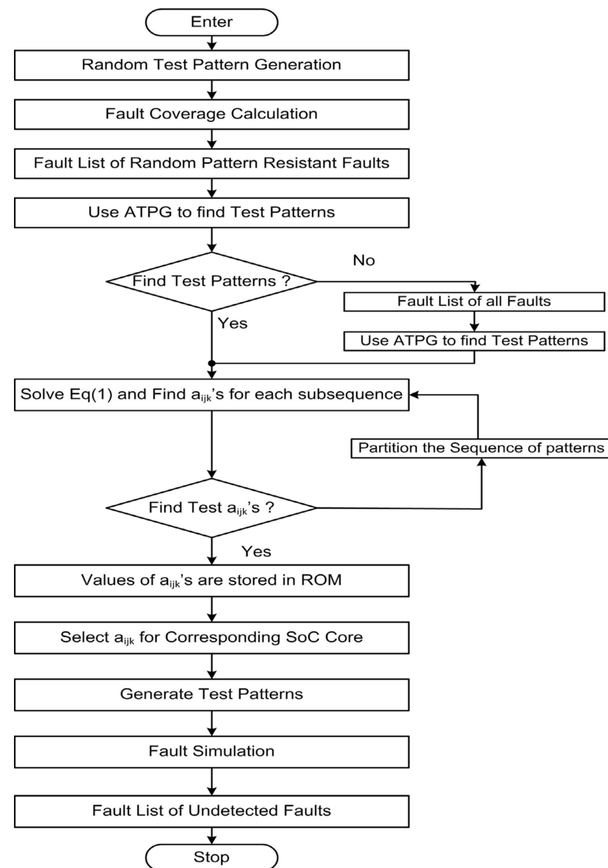

Figure 4. Step for Synthesis of Reconfigurable 2-D LFSR.

The connections in each configurable 2D LFSR is done by calculating the a_{ijk} value for each bit according to the series of deterministic test patterns and using those values of a_{ijk}, we decide whether to connect a particular wire to the XOR gate or not[4]. If a_{ijk} = 1, wire is connected to input of XOR gate. If a_{ijk} = 0, wire is not connected to input of XOR gate. The connections in a 2D LFSR cannot be changed once it is made so each CN will generate the same deterministic series of test patterns each time. The fact that logic '0' given to any input of an n input XOR gate makes it behave like an n-1 input XOR gate is exploited in making the XOR gate connections inputs reconfigurable. Using this fact, the number of configurable LFSR networks can be shrunken down from many to one, greatly reducing the area. As shown in the Figure 5, to produce 5 bit test vectors, we need 5 values of a_i which will be 5 bits each. Thus by using five 5 bit values, test vectors can be generated in any order we want. Considering the same configuration for SoCs with more number of cores with more number of inputs, the design of the reconfigurable 2D LFSR is greatly simplified as only values of a_i has to be changed for each core.

The value of a_{ijk} is calculated using the above equation (1) using binary linear programming and this can be successfully used to create test vectors in an order which covers maximum faults in the circuits. The values of a_i are stored in an on-chip ROM. The internal structure of the XOR Configuration Network is shown in the following Figure 6. The control of random pattern generation depends upon the values of a_i. Thus, this reconfigurable circuit can be used in any SoC that requires similar number of input bits. The order of the test patterns can be changed by changing the values of the a_i vector. For an SoC with n number of cores which have m inputs each, the reconfigurable LFSR will consist of 'm^2 '2-input AND gate and 'm' number of m-input XOR gates.

Therefore we can see that even if the number of cores increases, the hardware for the reconfigurable 2D LFSR remains the same. The only thing that need to be added are the corresponding a_i values that have been calculated for the generation of the test patterns required for testing the extra cores. This way, we see that the number of gates remain the same for any number of cores in the SoC. Hence, this technique can be very powerful if used in an SoC which has a lot of cores and most of them have the same number of primary inputs as the number of XOR gates and AND gates remain the same reducing the hardware by a large percentage.

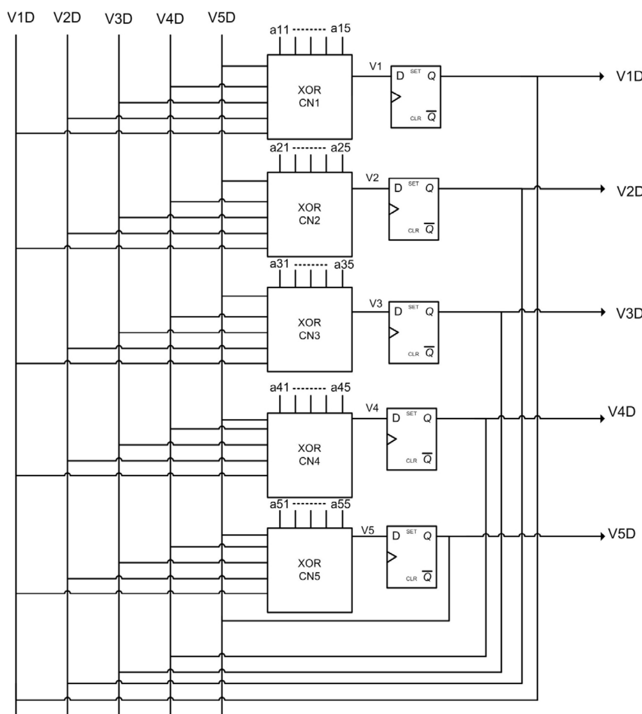

Figure 5. Architecture of a Reconfigurable 2-D LFSR.

4. Result and Analysis

The test patterns for two SoC cores i.e benchmark circuits' mul16 and div16 has been generated and synthesized by 2D LFSR and Reconfigurable 2D LFSR. Mul16 is a 16 bit 2's complement multiplier benchmark circuit which uses basic shift and Add multiplication algorithm. Div16 is 16 bit divider benchmark circuit which performs division operation by means of repeated subtraction. Table 1 shows the gate count, transistor count and fault coverage comparison of 2D LFSR and Reconfigurable 2D LFSR for different bench mark circuits. There is only one

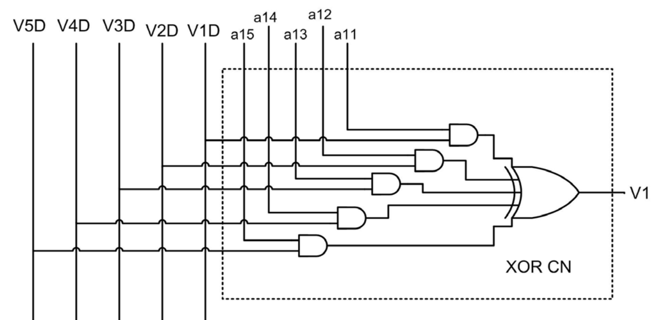

Figure 6. Internal Structure of XOR CN Block.

Table 1. Comparison of Gate Count, Fault Coverage and Transistor Count

Circuit	Gate Count				Faults	Detected Faults	Fault Coverage	Transistor Count	
	2D LFSR		Reconfigurable 2D LFSR					2D LFSR	Reconfigurable 2D LFSR
	Xor	Inv	Xor	And					
Mul16	217	21	240	256	1708	1605	94%	2782	2944
Div16	188	33			2147	1719	80%	2712	
Total	405	54	240	256	3855	3324	86%	5494	2944

Configuration network in Reconfigurable 2D LFSR which is used for testing different benchmark circuits whereas 2D LFSR requires different configuration networks for testing different SoC cores. The Fault coverage for mul16 and div16 is 94% and 80% respectively. The no of transistors used in Reconfigurable 2D LFSR are less compared to 2D LFSR as less number of XOR gates are used in Reconfigurable 2D LFSR. The Fault coverage is same in both the techniques but there is significant reduction in hardware when multiple SoC cores have to be tested.

5. Conclusion

This paper presents a basic approach to design Reconfigurable 2D LFSR to generate test patterns for testing SoC embedded cores. The main idea behind this technique is to reduce the hardware architecture for testing multiple cores. There is a drastic reduction by 46.4% in transistor count for Reconfigurable 2D LFSR compared to 2D LFSR. The proposed technique is much more useful for testing System on a Chip with large number of cores as same configuration network is used to test different SoC cores whereas 2D LFSR uses different configuration networks for testing different SoC cores.

6. References

1. Bushnell ML, Agarwal VD. Essentials of Electronic Testing for Digital, Memory and Mixed Signal VLSI Circuits. New York: Kluwer Academic Publishers; 2002.
2. Chen C-IH, George K. Configurable two-dimensional Linear Feedback Shifter Registers for parallel and serial Built-In Self-Test. IEEE Trans Instrum Meas. 2004 Aug; 53(7):1005–14.
3. Ahmed N, Tehranipoor M, Nourani M. Low Transition LFSR for BIST Based Applications. Proceedings Asian Test Symposium; 2005 Dec. p. 138–43.
4. Patel JH, Hamzaoglu I. Reducing Test Application Time for Built In-Self-Test Test Pattern Generators. Proceedings VLSI Test Symposium; 2000 Apr. p. 369–75.
5. Psarakis M, Kranitis N, Gizopoulos D, Paschalis A, Zorian Y. An Effective Deterministic BIST Scheme for Shifter/Accumulator Pairs in Data Paths. Proceedings Int Symp Quality Electron; 2001 Mar. p. 343–9.
6. Touba NA, Krishna CV. Hybrid BIST Using an Incrementally Guided LFSR. Proceedings Int Symp Defect and Fault Tolerance in VLSI systems; 2003 Nov. p. 217–24.
7. Chen C-IH, Chakravarthy A, Zhang X. Structure Design and Optimization of 2-D LFSR-Based Multisequence Test Generator for Built-In-Self-Test. IEEE Trans Instrum Meas. 2008 Mar; 53(3):651–63.
8. Neophytou S, Tragoudas S, Michael MK. Efficient Deterministic test generation for BIST schemes with LFSR Reseeding. Proceedings Int Symp On-Line Testing; 2006 Jul. p. 16–21.
9. Al-Hashimi BM, Rosinger P, Nicolici N. Dual Multiple polynomial LFSR for Low-Power Mixed-Mode BIST. Proceedings Comput Digital Techniques; 2003 Jul; 4:209–17.