International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 23 Issue 8 –NOVEMBER 2016.

FPGA Implementation of Sharing Logic for Built in Generation using 3x3 Circuit Under test

MVV Sankar Prasad[#1] and Vijendra Dandimudi[*2]

[#]*M.Tech, Electronics and Communication Engineering. Godavari Institute of Engineering & Technology (GIET), Rajahmundry, Andhra Pradesh, India*

***Abstract*— In this paper we discussed about the built in generation for logic blocks. The advantage of this technique is identifying the similar characteristics of the each logical block. Functional broadside tests are important for addressing over testing of delay faults as well as avoiding excessive power dissipation during test application. Finally to analysis the performance of the built in generation technique and compare to existing system.**

***Index Terms*— BIST, area efficient, logic blocks.**

I. INTRODUCTION

In the automotive, aviation and all electronics devices in biomedical filed the reliability is very important one. And also the repair cycle times low is needed. Testing the reliability of the devices in external sources then the area is increased. In built in self test to reduce the test data volume is reduced.

VLSI chips have reached a complexity and still their density twice every 2 years [1]. This makes it impossible to rule out faults during production and design, even with the best design tools and fabrication processes available. If testing facilitates of rapid diagnosis and thus are provides a means to avoid fatal error in production delays resulting from excessive debug time or sales the defective products, it is worth the additional cost.

This work considers the on-chip generation of functional broadside tests. On-chip test generation facilitates at-speed test application and decreases the test data volume. The on-chip test generation method from [2] applies PFTG (pseudo functional test generation) based on LFSR. The tests that are needed for achieving this more fault coverage are also ones that can cause over testing.

The gates are used for modifying the sequence in random order to avoid cases where the sequence takes repeatedly till the circuit reachable same or similar states. This is referred to as repeated synchronization [3].

In [4], Reduce the power consumption in scan-based built-in generation methods (BIGMs) is by using scan chain ordering techniques. These reduce the average power consumption when scanning in scanning out captured responses test vectors. Although this algorithms aim to reduce average-power consumption.

The rest of this paper to introduce the related work for the paper is discussed in section II. Then, in section III, the proposed system of BIST technique is present. Section IV presents the simulation result of the paper. Finally Section V presents the conclusion of the paper.

II. RELATED WORK

The case where the design-under-test is partitioned into logic blocks, and there is a choice with respect to the placement of the built-in test generation logic for these blocks. Fig. 1 illustrates a design that consists of four logic blocks B0, B1, B2 and B3. Horizontal lines stand for scan chains, and vertical lines stand for primary inputs. The built-in test generation method from produces values only for primary inputs, and not for scan chains. The scan chains are used initially for bringing the circuit into a reachable state, and for observing output responses. After initialization, the primary input sequences generated by the built-in test generation logic take the circuit through reachable states. Some of these states are used as initial states for functional broadside tests.

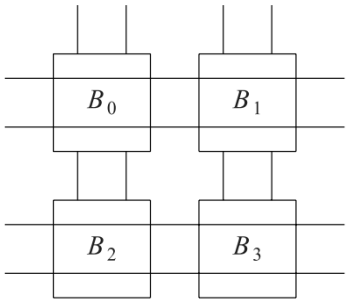


Figure 1: Example logic blocks.

The resulting configuration is illustrated by Fig. 2. In Fig. 2, B0 has n0 = 4 primary inputs, B1 has n1 = 2 primary inputs, B2 has n2 = 3 primary inputs, and B3 has n3 = 2 primary inputs. With n = 4, the LF SR has 4d bits, and four outputs. The gates corresponding to c_G are not shown in Fig. 2. Output 0 of the test generation logic drives primary input 0 of B0, B1, B2 and B3. Output 1 of the test generation logic drives

primary input 1 of B0, B1, B2 and B3. Output 2 of the test generation logic drives primary input 2 of B0 and B2. Output 3 of the test generation logic drives primary input 3 of B0.

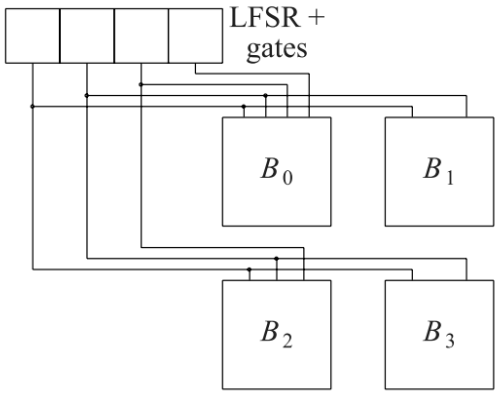


Figure 2: Test generation logic for a group.

III. PROPOSED SYSTEM

In proposed system to use the 9 block for testing of BIST technique. Overall, the built-in test generation method from requires a d*n-bit LFSR, a modulo-L counter, and at most n+1 gates. The initial state s_{init} as well as the seeds are assumed to be scanned in before the application of each primary input sequence. Circular shift requires scan chains of equal lengths. This can be achieved by adding dummy flip-flops to the shorter scan chains.

A. Linear Feedback Shift Registers

The key distribution problem for One-Time Pad suggests that one might use an algorithm to generate the random sequence needed as the key (transfer of only a short seed would then be needed).

However, no algorithm using a finite state machine can produce a truly random sequence, since the finiteness forces the sequence to be periodic. The best we can do is use very long period sequences, called *pseudo-random sequences.*

1) Golomb's Principles

- G1: The number of zeros and ones should be as equal as possible per period.
- G2: Half the runs in a period have length 1, one-quarter have length 2, ... , $1/2^i$ have length i. Moreover, for any length, half the runs are blocks and the other half gaps. (A *block* is a subsequence of the form ...011110... and a *gap* is one of the form ...10000001...., either type is called a *run*.)
- G3: The out-of-phase autocorrelation AC(k) has the same value for all k.

AC(k) = (Agreements - Disagreements)/p where we are comparing a sequence of period p and its shift by k places. The autocorrelation is out-of-phase if p does not divide k.

Furthermore, to be of practical use for cryptologists we would require:

- C1: The period should be very long (~ 10^{50} at a minimum).
- C2: The sequence should be easy to generate (for fast encryption).
- C3: The cryptosystem based on the sequence should be cryptographically secure against chosen plaintext attack. (minimum level of security for modern cryptosystems)

B. Feedback Shift Registers

Feedback Shift Registers are a commonly used method of producing pseudo-random sequences. The block diagram of the FSR is shown in figure 3.

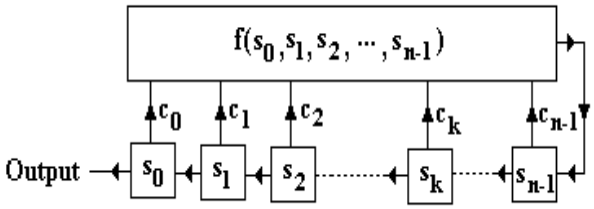


Figure 3: FSR

We first consider the case that f is a linear function, i.e.,

$$f(s) = \sum_{i=0}^{n-1} c_i s_i \qquad (1)$$

The output of this LFSR is determined by the initial values s_0, s_1... s_{n-1} and the linear recursion relationship:

$$s_{k+n} = \sum_{i=0}^{n-1} c_i s_{k+i}, \quad k \geq 0 \qquad (2)$$

Or equivalently

$$\sum_{i=0}^{n} c_i s_{k+i} = 0, \quad k \geq 0 \qquad (3)$$

Where $c_n = 1$ by definition.

A sequence produced by a length n LFSR which has period 2^n-1 is called a PN-sequence (or a pseudo-noise sequence). We can characterize the LFSR's that produce PN-sequences. We define the *characteristic polynomial* of an LFSR as the polynomial,

$$f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} + x^n = \sum_{i=0}^{n} c_i x^i \qquad (4)$$

where $c_n = 1$ by definition and $c_0 = 1$ by assumption.

C. Some Facts and Definitions From Algebra

1. Every polynomial f(x) with coefficients in GF (2) having f(0) = 1 divides $x^m + 1$ for some m. The smallest m for which this is true is called the *period* of f(x).
2. An *irreducible* (cannot be factored) polynomial of degree n has a period which divides $2^n - 1$.

An irreducible polynomial of degree n whose period is $2^n - 1$ is called a *primitive polynomial*.

The primary input sequence A is generated by an LFSR whose states are used as pseudo-random vectors. The LFSR sequence is modified in order to avoid an effect called repeated synchronization, where certain primary input values cause certain state variables to assume the same values repeatedly. The logic for generating the primary input sequence A is illustrated by Fig. 4. For a parameter denoted by d, a distinct set of d bits of the LFSR is used for determining the sequence applied to every primary input. For a parameter denoted by mod, up to mod of the d bits dedicated to each primary input are used for avoiding repeated synchronization. If the value 0 on a primary input synchronizes fewer state variables than the value 1, then the value 0 is preferred. In this case, a mod-input AND gate is used for ensuring that a 0 appears more often than a 1 on this primary input. A mod-input OR gate is used for the primary input if the value 1 synchronizes fewer state variables, and it is thus the preferred value for the primary input. No gate is used if both values synchronize the same number of state variables.

For a circuit with n primary inputs, this method requires an LFSR with d*n bits, and at most one mod-input gate for every primary input. The preferred values of the primary inputs are captured in a primary input cube denoted by c. For a primary input j, c (j) indicates its preferred value, which may be 0, 1 or x. Several primary input sequences are applied by using several different seeds for initializing the LFSR. Each additional sequence results in a different set of functional broadside tests, and helps increase the fault coverage. All the sequences use the same values of the parameters L, d and mod. Consequently, the same logic is used for generating all the tests.

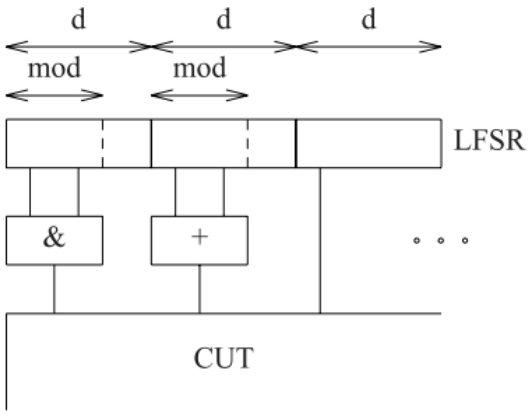


Figure 4: Test generation logic.

IV. SIMULATION AND RESULT

The simulate the proposed system architecture in Modelsim and to analysis the area, power, and delay of the proposed system in Spartan 6 by using Xilinx software. The simulation result for the existing system and proposed system is shows in figure 5 and 6. The synthesis report of the proposed system is shown in figure 7 and figure 8. Finally the comparison of the proposed system is detailed in table I.

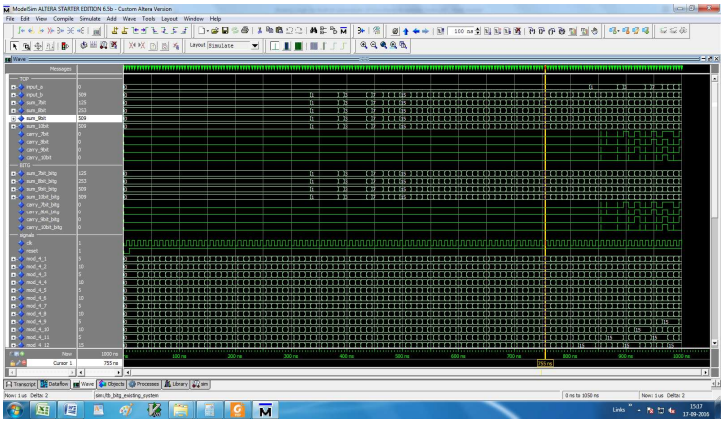


Fig.5: simulation result for existing system

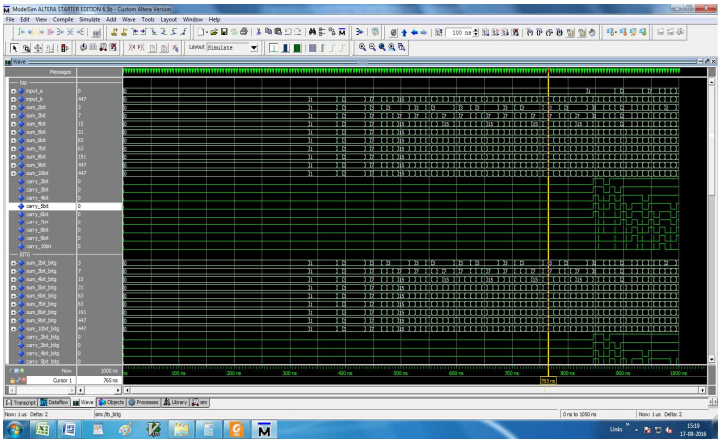


Fig.6: simulation result for proposed system

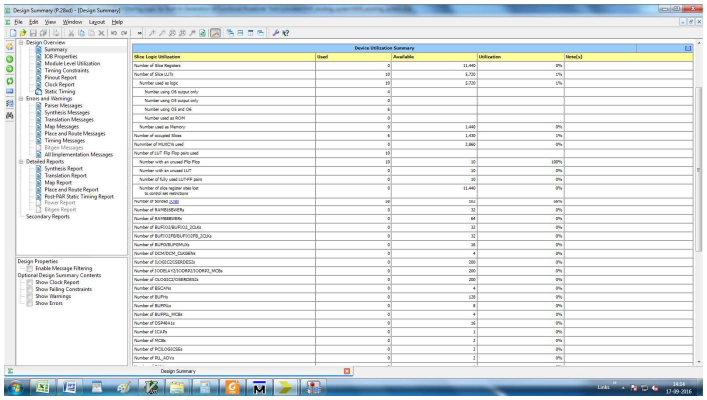


Fig.7: synthesis report for existing system

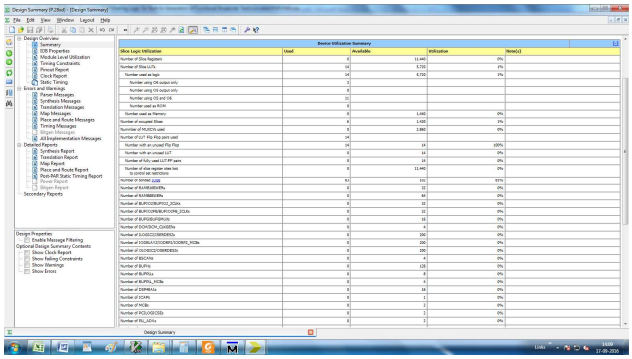
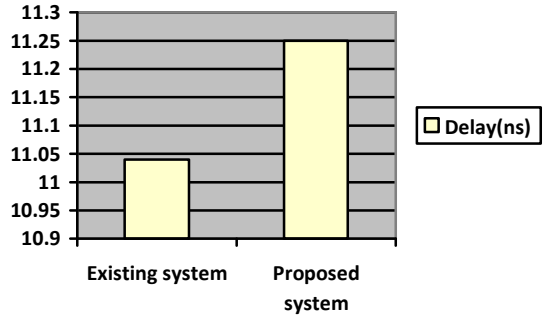
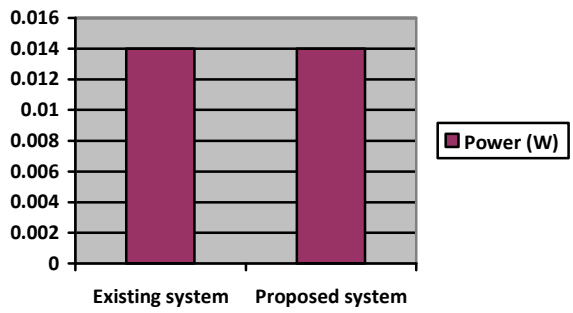


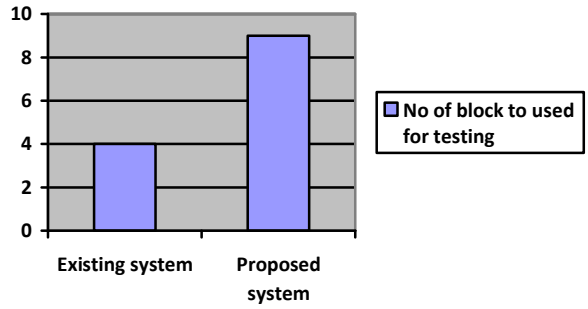
Fig.8: synthesis report for proposed system

Table 1: comparison

	Existing system	Proposed system
No of block to used for testing	**4**	**9**
Power (W)	**0.014**	**0.014**
Delay(ns)	11.04	**11.25**







V. CONCLUSION

We are discussed about the built in generation for logic blocks. The advantage of this technique is identifying the similar characteristics of the each logical block. Functional broadside tests are important for addressing over testing of delay faults as well as avoiding excessive power dissipation during test application. Finally to analysis the delay of the built in generation technique is 11.24ns.

REFERENCES

[1]. X. Sun, M. Serra, "Merging Concurrent Checking and Off-line BIST", Proc. International Test Conference 1992, Sept. 20- 24 1992, Baltimore, MD, IEEE CS Press, pp. 958-967.

[2]. Y.-C. Lin, F. Lu, and K.-T. Cheng, "Pseudofunctional testing," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., pp. 1535–1546, 2006.

[3]. I. Pomeranz and S. M. Reddy, "On reset based functional broadside tests," in Proc. Design Autom. Test Euro. Conf., 2010, pp. 1438–1443.

[4]. J. Rearick, "Too much delay fault coverage is a bad thing," in Proc. Int. Test Conf., 2001, pp. 624–633.

[5]. R L.-T. Wang, X. Wen, S. Wu, H. Furukawa, H.-J. Chao, B. Sheu, J. Guo, and W.-B. Jone, "Using Launch-on-Capture for Testing BIST Designs Containing Synchronous and Asynchronous Clock Domains," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 2, pp. 299-312, Feb. 2010.

[6]. M.E. Amyeen, A. Jayalakshmi, S. Venkataraman, S.V. Pathy, and E.C. Tan, "Logic BIST Silicon Debug and Volume Diagnosis Methodology," Proc. Int'l Test Conf., pp. 1-10, 2011.

[7]. M. Abramovici, M.A. Breuer, and A.D. Friedman, Digital Systems Testing and Testable Design. IEEE Press, 1995.

[8]. P. Girard, "Survey of Low-Power Testing of VLSI Circuits," IEEE Design and Test of Computers, vol. 19, no. 3, pp. 80-90, May/June 2002.

[9]. P. Rosinger, B.M. Al-Hashimi, and N. Nicolici, "Dual Multiple-Polynomial LFSR for Low-Power Mixed-Mode BIST," IEE Proc.—Computers and Digital Techniques, vol. 150, no. 4, pp. 209-217, 2003.

[10]. J. Lee and N.-A. Touba, “Low Power BIST Based on Scan Partitioning,” Proc. Symp. Defect and Fault Tolerance in VLSI Systems, pp. 33-41, 2005.

[11]. S. Wang, “A BIST TPG for Low Power Dissipation and High Fault Coverage,” IEEE Trans. VLSI Systems, vol. 15, no. 7, pp. 777-789, July 2007.

[12]. H. Lee, I. Pomeranz, and S.M. Reddy, “Scan BIST Targeting Transition Faults Using a Markov Source,” Proc. Fifth Int’l Symp. Quality Electronic Design, pp. 497-502, 2004.

[13]. N. Tendolkar, D. Belete, A. Razdan, H. Reyes, B. Schwarz, and M. Sullivan, “Test Methodology for Freescale’s High Performance e600 Core Based on PowerPC(R) Instruction Set Architecture,” Proc. Int’l Test Conf., pp. 1-9, 2005.