THINK INDIA Journal

ISSN:0971-1260
Vol-22-Issue-16-August-2019

Design and Analysis of CCALBO - A Novel BIST Structure

Abhishek Kumar
School of Electronics and Electrical Engineering
Lovely Professional University
Punjab, India
abkvjti@gmail.com

Abstract— This paper proposes an original approach for BILBO-based BISTs architectures coined (Concurrent Built-In Logic Block Observer) CCALBO. This technique is a combined inspiration from the designs of CBILBO and CALBO. We give the detail of designing the CCALBO cell and its fault-masking probability with inclusion in a combination and sequential CUT i.e., Vedic Multiplier and multiply and Accumulate Unit. CCALBO-based BIST is then compared with the most competitive technique CBILBO-based BIST. We choose major parameters such as power, area, and delay with additional parameters such as functionality, fault coverage. When compared to the CBILBO technique, a total dynamic power reduction of 66% for CCALBO based BIST using the MAC unit. It was found that area reduced as well since CCALBO uses CUTs sequential components to generate additional test logic. CCALBO-based BIST was also found to beat CBILBO in terms of speed and test time, requiring a lesser test time to achieve fault coverage of 100%. Overall, CCALBO triumphs over its counterpart, CBILBO.

Keywords— BILBO, BIST, CALBO, CBILBO, CAR, CCALBO, MAC, Vedic Multiplier.

1. INTRODUCTION

There To improve the testability of Circuit under test (CUT), designers often add extra logic to the existing test structure. This approach, to an extent, is helpful when it comes to fault coverage and speed-up the testing process, but also increases the test area. [1] Observed this drawback and proposed the Built-In Logic Block Observation (BILBO) test architecture for BIST circuits. Theoretically, since this structure edits parts of CUT itself to build a test structure, BILBO takes zero test logic area requirements. The idea is to partition CUT into combinational and sequential blocks. The sequential blocks are then modified to work as test circuits (BILBO blocks) without hampering the original functionality using mode-based logic. BILBO performs as a test pattern generator, Multiple Input Signature Register (MISR), scan register, and normal D-Flip Flops. Researchers observed that test scheduling can be a cumbersome process for BILBO-based BIST architectures [2]. Figure 1 shows the test scheduling process in BILBO/CBILBO/CALBO/CCALBO-based BIST. Concurrent Built-In Logic Block Observer (CBILBO) was proposed in [2] to overcome problems of test session scheduling and register self-adjacency. BILBO uses a 2n set of registers to include the same test logic as BILBO. Since BILBO and CCALBO test methods are based on Linear Feedback Shift Register (LFSR) techniques for test pattern generation, use of global feedback, making routing interconnects longer and in-turn increasing the area. [3] Proposed Cellular Automata Logic Block Observer

ISSN:0971-1260
Vol-22-Issue-16-August-2019

(CALBO) which used Cellular Automata Registers (CAR) cells with rules 90 and 150 alternatively. CAR cells have been proved [4] to be more random in terms of generating test patterns as compared to any other test pattern techniques. Unlike LFSR (global feedback), CAR cells only communicate with neighboring cells, which helps in reducing the area overhead caused by the earlier structure. The proposed design, CCALBO is a unique architecture that combines advantages of both CBILBO and CALBO. Due to concurrency, test scheduling is avoided and in turn, reduces the test time (CBILBO) and routing interconnect issues are solved by using CAR cells (CALBO).

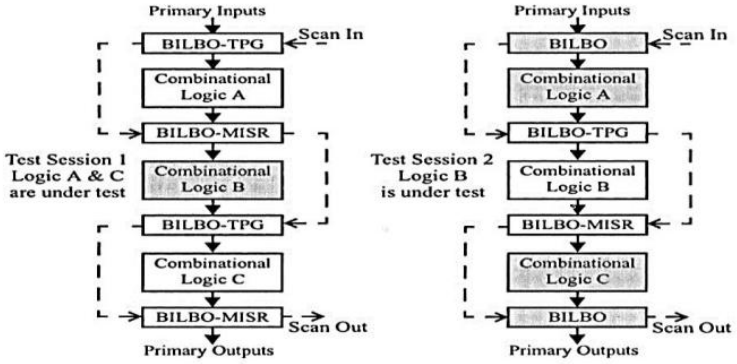


Fig. 1 BILBO test scheduling example [1].

The work is reported as follows; Section II describes the proposed structure in detail and gives and generic design to be used in BIST applications. Section III illustrates the CUT structure and inclusion of CCALBO to design BIST. Section IV discusses the results and other parameters to compare CCALBO with other designs. Section V concludes the work and briefs some of the extensions

2. **CCALBO**

As stated earlier, CCALBO is a combination of two architectures CALBO and CBILBO. CALBO structure comprises of CAR cells to increase the randomness in the patterns and reduce area.

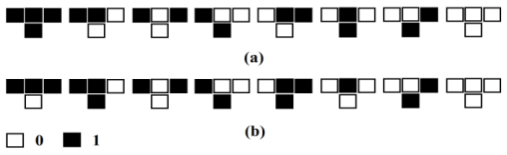


Fig. 2. State transition diagram for CA rules; (a) Rule 150; (b) Rule 90.

While LFSR and BS-CFSR requires global feedback, CAR cells only communicate with neighboring cells only. This architecture has a collection of flip flops (cells) with only immediate forward and backward routings. The format of connections is defined by rules, where each rule suggests the next state of a cell based on the state of its neighbors and/or the current cell. The equations (1) and (2) are obtained by the 8-bit output value obtained from figure 2 (a) and (b) using K-map with variables xj-1 (t), xj (t), and xj+1 (t) for previous, present, and next cell respectively. Rule 150 produces output value 10010110 and rule 90 gives 01011010.

ISSN:0971-1260
Vol-22-Issue-16-August-2019

$$Rule\ 90: x_j(t+1) = x_{j-1}(t) + x_{j+1}(t) \quad .1$$

$$Rule\ 150: x_j(t+1) = x_{j-1}(t) + x_j(t) + x_{j+1}(t) \quad .2$$

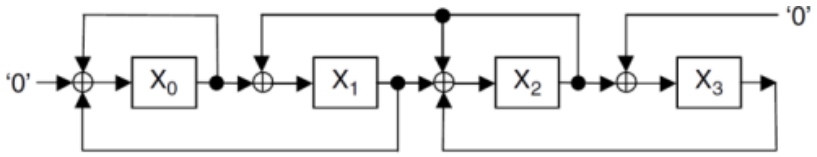


Fig3. Example of 4-stage TPG based on CAR using 90 and 150 rules [4].

Figure 4 shows an arrangement where these CAR cells of rules 90 and 150 are alternatively placed making an-bit CALBO.

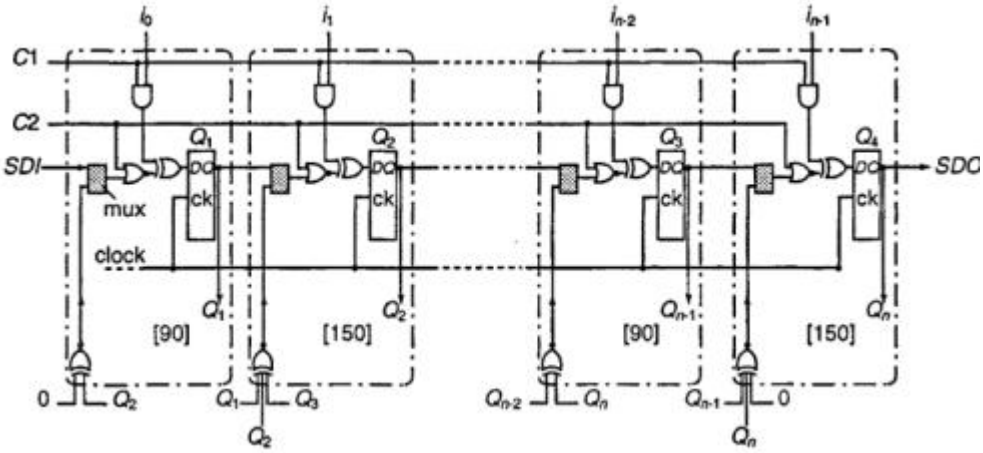
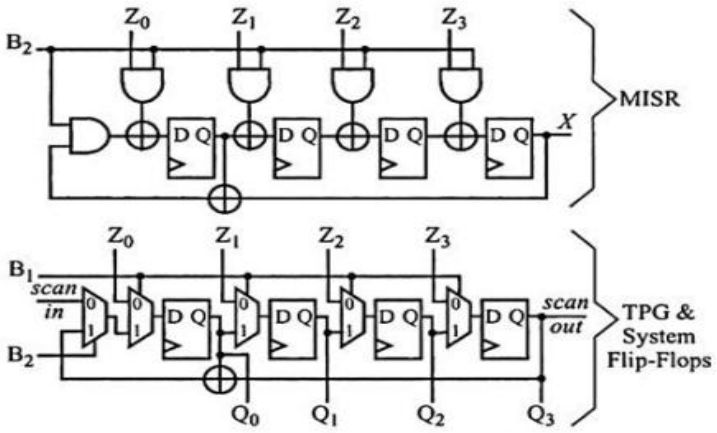


Fig4. n-bit CALBO based on CAR cell rules 90 and 150 [3].

CBILBO's design helps to remove test scheduling and register self-adjacency with two sets of register chains to produce test patterns and signature simultaneously. The idea is to create register pairs from either CUT or adding new registers (if required) to achieve this concurrency. Figure 5 shows the 4-bit CBILBO structure.



ISSN:0971-1260
Vol-22-Issue-16-August-2019

Fig5. 4-bit CBILBO architecture[2].

In the light of these two structures, a new improved architecture named CCALBO combining the advantages of both techniques is coined. Figure 6 shows the proposed 1-bit CCALBO cell structure. According to the CUT, one can add multiple CCALBO cells and make a scan chain by modifying the existing registers. One cell uses two sets of register to achieve concurrency and operates on four modes with two control signals. In figure 6, the control signal C1 and C2 decide the modes of the cell. Controlling information of basic gates (NOR is 1 and AND is 0) was used to change between modes of the cell. Scan in or Xi-1 is the inputs corresponding to the beginning of the scan chain or previous cell output respectively. Similarly, scan out or Xi is outputs corresponding to end of scan chain or current cell output respectively and i corresponds inputs from CUT. Depending on the rule selected Ex-OR gate will take inputs from neighboring cells (equations 1 and 2). The best method is to use alternate cells of rules 90 and 150 for getting the most randomness of test patterns. The four modes depending on control signals C1 and C2 are as follows;

1. *Cl = 1, C2 = 1,* normal operation;
2. *Cl = 0, C2 = 0,* scan mode;
3. *Cl = 1, C2 = 0,* TPG and MISR simultaneously;
4. *Cl = 0, C2 = 1,* reset with all Q = 0;

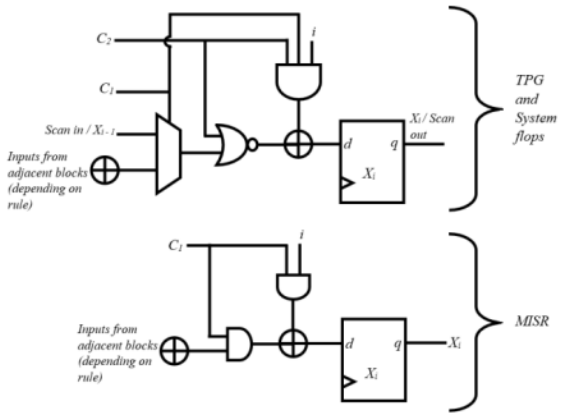


Fig6. Proposed CCALBO cell.

A theoretical consideration of the probability of fault masking or aliasing probability (producing fault-free signature even when a fault has occurred) can be made. Let the multiple inputs to MISR in CCALBO be m and the number of stages in MISR be n, where n<m, then the exhaustive sequence of possible multiple inputs the pattern becomes 2^m, however, the number of possible different signatures in MISR is only 2^n. Therefore, by considering all possible faults in the input bitstreams to be equally possible, one correct signature and a fault-free bitstream in MISR will be 2^m–1 to 1 faulty bitstreams of which 2^{m+n}-1 - 1 give the same signature as the fault-free input. The aliasing probability is therefore given by:

$$P_{CCALBO} = \frac{2^{m-1} - 1}{2^{m+n-1} - 1} \times 100\,\% \quad [3]$$

ISSN:0971-1260
Vol-22-Issue-16-August-2019

When n and m are sufficiently large, one fault-free response can be neglected and equation 3 changes to:

$$P_{CCALBO} = \frac{2^{m-1}}{2^{m+n-1}} \times 100\,\% = \frac{1}{2^n} \times 100\,\% \quad [4]$$

3. CIRCUIT UNDER TEST

To implement BIST using CCALBO structure, we use a combinational and the sequential structure as a CUT. Vedic Multiplier (VM) and multiply and Accumulate unit (MAC) processor were chosen as CUT for CCALBO-based BIST. We use the "Vertical and Crosswise" algorithm and architecture of VM used in [5-8] shown in Figures 7 and 8 respectively.

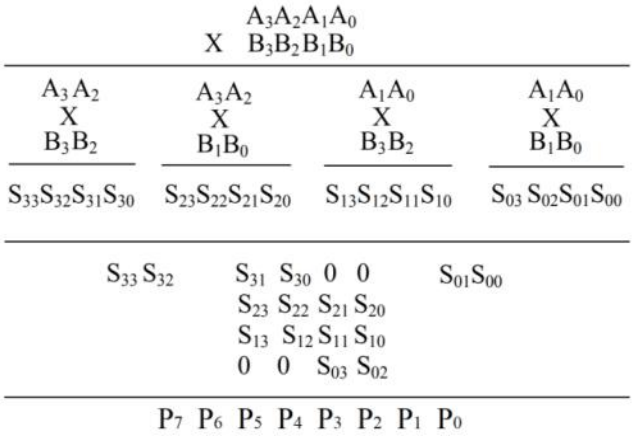
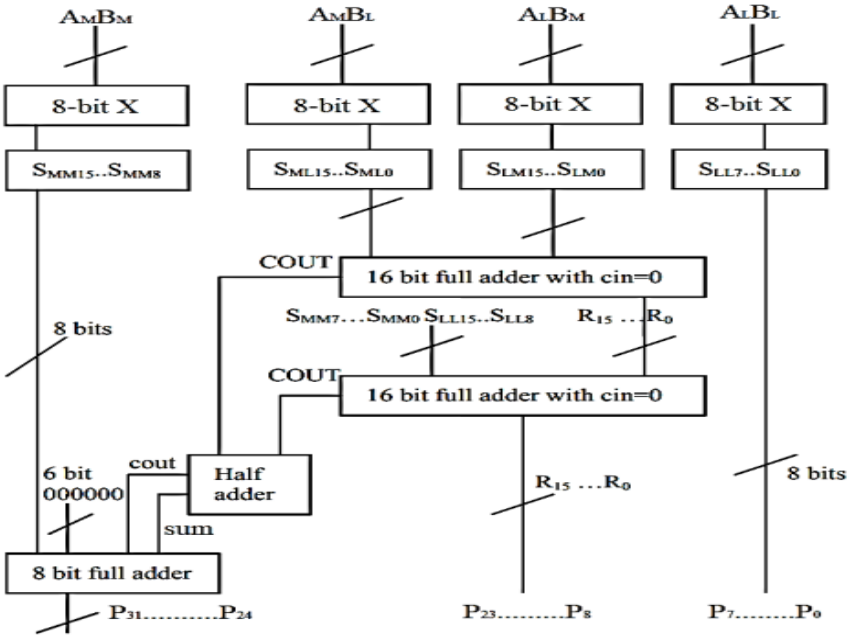


Fig. 7. Vertical and Crosswise algorithm for 4-bit Vedic Multiplier[5].



ISSN:0971-1260
Vol-22-Issue-16-August-2019

Fig8. Block diagram of 16-bit VM [9].

Fig. 8 describes a simplified MAC processor. Since MAC unit uses Parallel-In Parallel-Out (PIPO) registers, making it a sequential circuit is shown in figure 9. When designing CCALBO-based BIST, PIPO registers were replaced by CCALBO cells. Multiplier and Addition units used were 16-bit VM and 32-bit Carry Skip Adder (CSA) respectively.

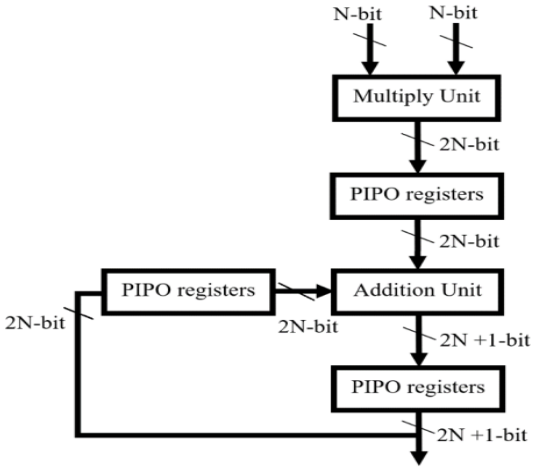
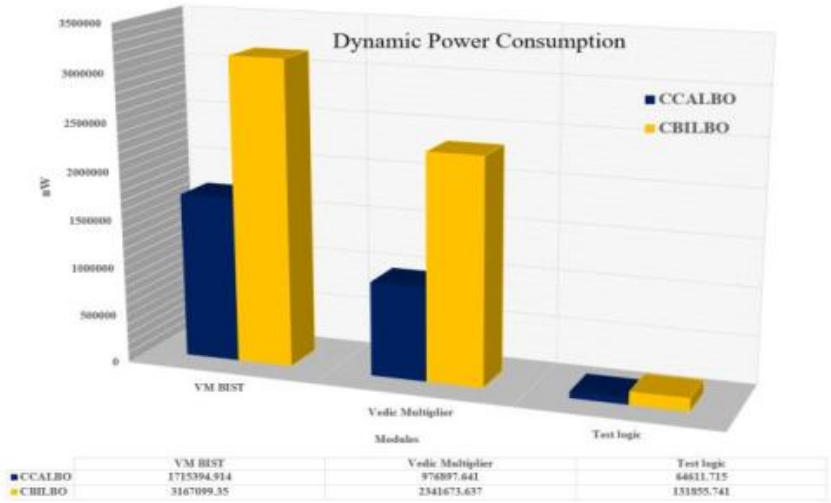


Fig9. Simplified structure of the MAC unit [9].

4. RESULT & DISUSSION

The coding and RTL analysis were carried out on *Xilinx Vivado 2018.1 Artix 7 FPGA*. We used *Cadence RTL compiler 14.26* for power, area, and delay analysis. The analysis carried out as follows:

4.1. Power Consumption Analysis



	VM BIST	Vedic Multiplier	Test logic
CCALBO	1715394.914	976897.641	64611.715
CBILBO	3167099.35	2341673.637	133855.741

ISSN:0971-1260
Vol-22-Issue-16-August-2019

Fig10. Power consumption of CCALBO-based BIST compared with CBILBO-based BIST using VM as CUT in nW.

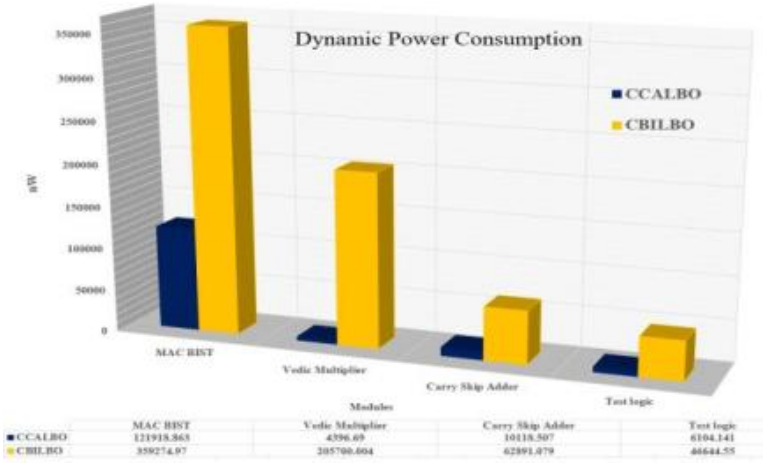
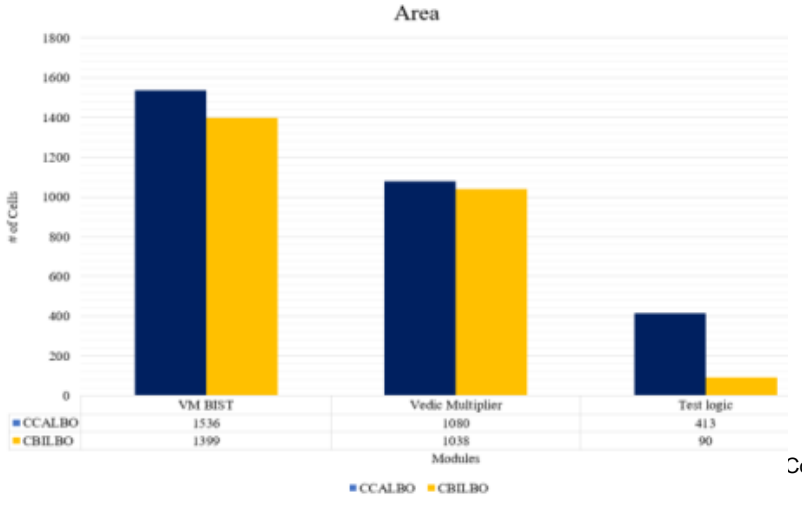


Fig11. Power consumption of CCALBO-based BIST compared with CBILBO-based BIST using MAC unit as CUT in nW.

Dynamic power takes up a major contribution in total power consumption. Figures 10 and 11 shows that CCALBO-based BIST takes lesser power than CBILBO-based BIST for both VM and MAC CUTs. We see a reduction of 46% for VM CUT based BIST and 66% for MAC CUT based BIST.

4.2. Area Analysis

Figure 12 shows the number of cells used (area) for CCLABO vs. CBILBO for VM as CUT. A sudden spike is seen in the number of cells for the proposed design; the reason is VM being a complete combinational circuit. Test logic to be added for combinational circuits increase twice the requirement. New registers to make scan chains were newly introduced, thus increasing the area. However, MAC is a sequential circuit and PIPO registers can be used to make scan chains and thus the area reduces, as shown in figure 13.



ISSN:0971-1260
Vol-22-Issue-16-August-2019

Fig12. Graph showing the numbers of cells used in CCALBO vs. CBILBO for VM as CUT.

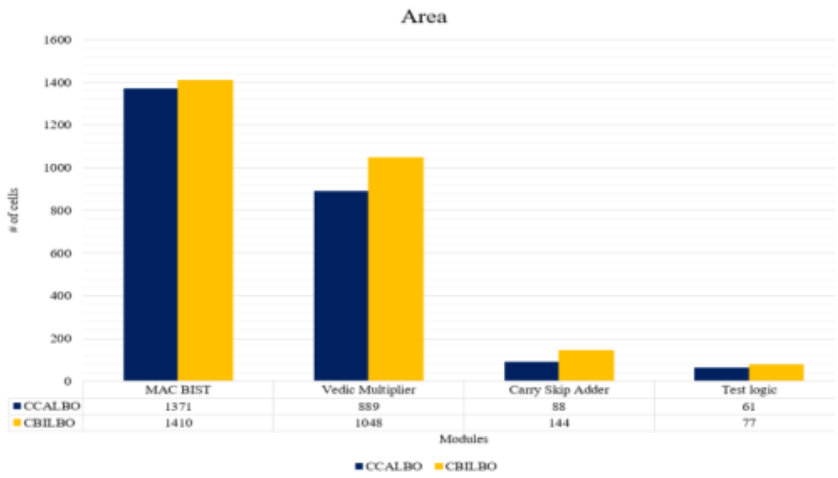


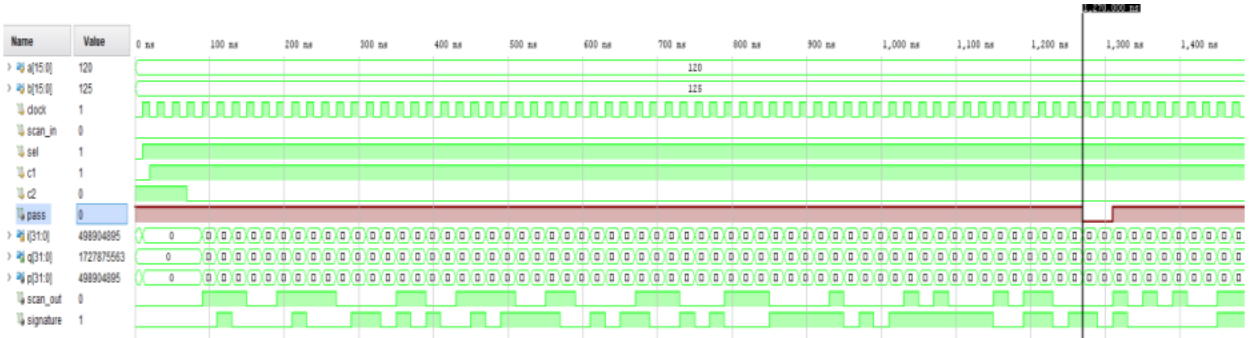
Fig13. Graph showing numbers of cells used in CCALBO vs. CBILBO for MAC as CUT**.**

4.4. Delay and Worst path analysis

During the test mode, the measure of suggesting how fast and accurate BIST has been identifying the fault/s can be analyzed by observing the delay parameters. CCALBO based circuits have worst-case delay of 6518 p sec for VM BIST and 6602 p sec for MAC BIST. Similarly, CBILBO-based BIST has worst-case delay of 6081 p sec for VM BIST and 6668 p sec for MAC BIST. No timing violation on worst-case delays of these circuits were observed. CCALBO-based BIST proves to be faster than CBILBO-based BIST.

4.5. Functionality analysis

Functionality can be analyzed through timing responses for both test and normal modes of operation. We start locking the test session to the fault-free response test session time. For VM-based BIST presented in figure 14 is 1270 ns and MAC-based BIST in figure 15 gives 1530 ns. These times will be used for comparison with the faulty test sessions. One of the examples is given in figure 16 and 17 for VM and MAC-based BIST. Deliberately a single stuck-at fault was introduced in the circuit which



ISSN:0971-1260
Vol-22-Issue-16-August-2019

produced a different test session end time, i.e., 1210 ns for VM-based BIST and 1770 ns for MAC-based BIST. This proves the fault can be detected by observing the variations in the test sessions.

Fig 14. Fault free response of CCALBO-based BIST using VM as CUT.

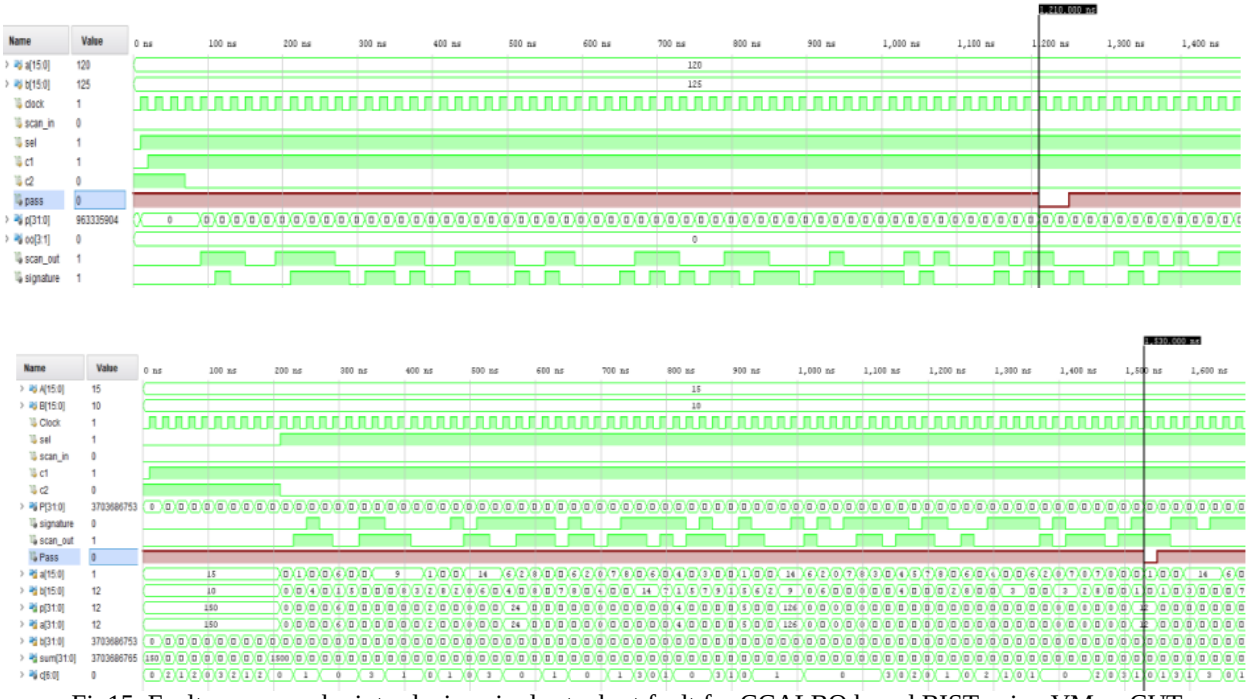


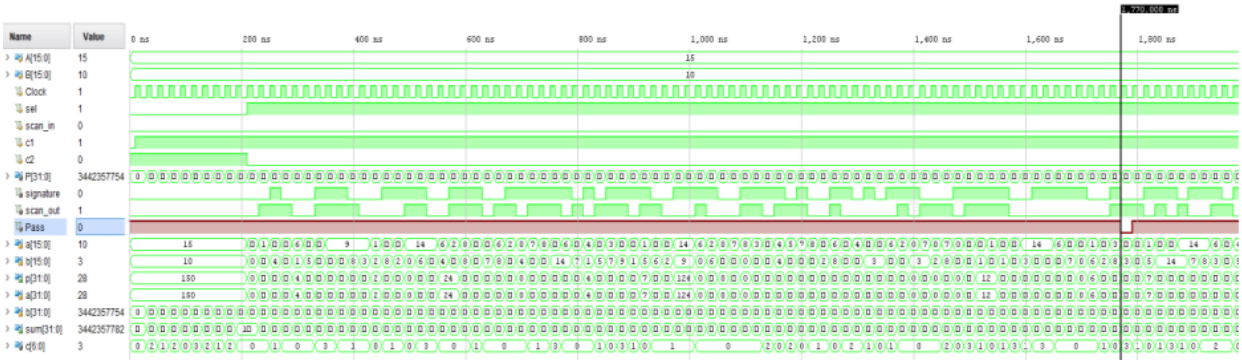
Fig15. Faulty response by introducing single stuck-at fault for CCALBO based BIST using VM as CUT

Fig16. Fault free response of CCALBO-based BIST using MAC as CUT.

Fig17 Faulty response by introducing a single stuck-at fault for CCALBO-based BIST using MAC as CUT.

4.6. Fault Coverage

Single stuck-at and bridging faults were introduced to estimate the fault coverage of every BIST techniques. We assume one test session period for all the circuits to reach 100% fault coverage. We observed CBILBO-based circuits can cover 100% faults requiring larger test vectors with more than twice the test session period as compared to CCALBO. Though VM-based BIST is a combinational circuit, it performs well due to randomness in the CCLABO when compared to CBILBO-based BIST counterpart.



ISSN:0971-1260
Vol-22-Issue-16-August-2019

5. Conclusion and Future Work

The proposed design CCALBO-based BIST was compared with CBILBO-based BIST with parameters such as power, area, delay, etc. We mathematically proved that fault masking probability reduces as the number of stages off CCLABO increases and most suitable for today's complex architecture. CCALBO architecture showed higher randomness due to CAR rule-based cells. Also, while analyzing power consumption, CCALBO system showed 66% reduction than CBILBO. The proposed designed also showed lesser area requirement than BILBO-based BIST architecture. We analyzed functionality with respect to a single test session and their respective fault coverages, where CCALBO achieves 100% coverage in shorter time and lesser essential test vectors. Future work includes implementing the same architecture for other more complicated benchmark circuits and complex processors and analyze the results for the same. As the technology move towards T Hz frequency speed, new testing techniques and architectures are required to sustain.

References

[1]B. Konemann, J. Mucha, and G. Zwiehof, "Built-in logic block observation techniques", Proc. Int. Test Conf., pp. 37–41, 1979.

[2] L.-T. Wang and E. J. McCluskey, "Concurrent built-in logic block observer (CBILBO)", Proc. Int. Symposium on Circuits and Systems, vol. 3, pp. 1054–1057, 1986.

[3] Shaik Mohammed Waseem and Afroz Fatima, "Cellular Automata Logic Block Observer-Based Testing for Network-on-Chip Architecture", Intelligent Computing and Information and Communication, pp. 19-26, Jan. 2018.

[4] Afroz Fatima and Shaik Mohammed Waseem, "Cellular Automata-based Built-In-Self Test implementation for Star Topology NoC", 11th International Conference on Intelligent Systems and Control, pp. 45-48, Jan. 2017.

[5] Shamim Akhter, "VHDL implementation of fast N X N multiplier based on Vedic mathematic". 18th European Conference on Circuit Theory and Design, pp. 472 – 475, Aug. 2007.

[6]Swati Malik and Sangeeta Dhall, "Implementation of MAC unit using booth multiplier & ripple carry adder". International Journal of Applied Engineering Research, vol. 7, no.11, pp. 358-363, 2012.

[7]L. Gao, Y. Zhang and J. Zhao, "BIST using Cellular Automata as test pattern generator and response compaction" Consumer Electronics, Communications and Networks (CECNet). 2nd International Conference on, Yichang, pp. 200–203, 2012

ISSN:0971-1260
Vol-22-Issue-16-August-2019

[8] Bradley, Daryl W., and Andrew M. Tyrrell. "Immunotronics-novel finite-state-machine architectures with built-in self-test using self-nonself differentiation." IEEE Trans. on Evol. Comput, vol 6, no. 3, pp. 227-238, 2002.

[9] P. D. Hortensius, R. D. McLeod and H. C. Card, “Cellular automata-based signature analysis for built-in self-test”, IEEE Trans. on Computers, vol. 39, no. 10, pp. 1273–1283, Oct. 1990.