

BIST to Diagnosis Delay Fault in the LUT of Cluster Based FPGA

Nachiketa Das, Hafizur Rahaman, and Indrajit Banerjee

Abstract—This work reports a novel scheme for testing and diagnosis of a delay fault in LUT of a cluster based FPGA. The solution is based on implementing a BIST structure to diagnosis the delay fault of the LUT. The BIST is implemented by chaining k-number of Look-Up Tables (LUT) in specific way. The ORA used a polling scheme to determine the most suitable result and an ATPG will generate the optimum test pattern that will have full test coverage. The entire scheme was implemented and simulated for Virtex-II FPGA. Here the intention was to overcome the drawbacks of previously used method. A design example using the proposed method shows better result. The entire testing scheme can also be applied in On-Line testing environment by using Xilinx Jbits 3.0 API (Application Program Interface) for Xilinx Virtex-II FPGAs.

Index Terms—Delay fault, FPGA, JBits, Look-up table (LUT), Testing.

I. INTRODUCTION

Nowadays due to low development cost and inherent functional flexibility, the programmable logic in the form of Field Programmable Gate Array has become a widely accepted design approach for low and medium computing application. Manufacturer provides FPGA in various architectures, such as EPROM switch-based on FAMOS on RAM in order to store configuration information, and fuse or antifuse. This work is focused on testing SRAM-Based FPGA [1].

Testing such FPGA from both the manufacturer test point of view and the application-oriented test point of view has been increasingly receiving attention of the testing engineers. Several methods to test logic cells and interconnect faults such as “stuck_at” or “bridging faults” are presented in papers [2], [3]-[17]. Various types of faults may be formed in FPGA while manufacturing due to variation of different parameters. These faults can be classified as static fault that includes delay faults. Another type of fault known as dynamic fault, that includes certain type of bridging fault, which may arise due to repeated reconfiguration of the FPGAs.

For decades, Build-In-Self-Test (BIST) [2]-[4], [5] has become very popular for testing and diagnosis of various faults. Traditionally logic BIST has performed in context of system, burn-in test and gate level test where diagnostic resolutions are usually not required.

But with recent advancement of technology the logic

BIST is regaining its popularity as alternative test compression technique.

Reference [5] presents a 1- and 2- diagnosable BIST design that makes up Roving Tester (ROTE). The proposed BIST can perform diagnosis without compromising fault coverage by avoiding time- intensive adaptive diagnosis. Here, the results show that they have the highest coverage in 1- diagnosable functional –test based BIST with a three PLB TPG. [6] presents a programmable approach for scan based logic BIST. The proposed approach combines the techniques of reseeding and weight random test pattern test.

Paper [7] analyzes the timing behavior of Look-Up Tables (LUT) containing FPGA both in faulty and fault free conditions. Analysis of the result shows that the LUT delay fault is not independent of the realized functions.

In [8] a testing method is presented in manufacturer-oriented context, for testing delay fault in LUT. In this paper a test configuration is presented where LUTs are chained in a specific way and test patterns are applied, to test large and small delay faults (i.e StR and StF delay faults). The objective of [9] was to present a BIST structure to detect delay fault in LUT of a SRAM-Based FPGA. The testing configurations were same as that used in paper [8] but an added ORA is used. [10] presented an on line and off-line BIST based testing scheme to detect delay fault in FPGA. It uses a roving star architecture. Their scheme is implemented in Xilinx sparten FPGA.

The main objective of this work is to detect the delay fault in the Look-Up Tables (LUT) of the cluster based FPGA from the manufacturer-oriented testing point of view. The proposed testing scheme will overcome the drawbacks of the method reported in [8]-[9]. The entire testing scheme could be applied in On-Line testing environment by using *Xilinx Jbits 3.0* [11] API (Application Program Interface) for Xilinx Virtex-II FPGAs. Same analysis of delay fault model is considered as described in paper [7] and also Xilinx Virtex-II FPGA Architecture is used to describe the proposed method. In paper [12]-[13] presents a BIST architecture for testing of stuck_at fault, delay fault and bridging fault in FPGA interconnect. The area overhead of the proposed scheme is 0.5% (used in Xilinx FPGA)

The arrangement of the paper is given in the following lines. In section 2 a popular cluster based FPGA architecture and the LUT timing analysis is described in brief. In section 3 the proposed scheme and the relevant analysis is presented. Lastly the simulation results are shown.

II. BACKGROUND

A. Architecture of FPGA

The architecture of Virtex-II [1], which is the target device, is shown in Fig.1. This FPGA consists of the CLB, IOB, Block

Manuscript received January 28, 2012; revised March 3, 2012.

The authors are with School of VLSI Technology, Bengal Engg. & Science University, Shibpur, India (e-mail: nachiketad@gmail.com, rahaman_h@it.becs.ac.in, e-mail: ibanerjee@it.becs.ac.in).

select SRAM, Multiplier and DCM elements.

All the elements use the same interconnect scheme. The Virtex-II FPGA consists of two-dimensional array of CLBs as shown in Fig.1. Each CLB contains four slice and two three-stage buffers. Each slice has two four input LUTs, two D flip-flops and Fast carry look-ahead chains, etc. All elements like CLB, IOB and Block RAM etc are connected to an identical switch matrix for accessing the global routing resource as shown in Fig.1. Signals in Virtex-II are routed using global routing resources, which are located in horizontal and vertical routing channel between each switch matrix. The hierarchical routing resources are shown in Fig.2. It consists of twenty-four bidirectional lines, which distribute signals across the device. Vertical and horizontal long lines span the full height and width of the device. The 120 hex lines route signals to every third or sixth block away in all four directions. Organized in a staggered pattern, hex lines can only be driven from one end. Hex-line signals can be accessed either at the endpoints or at the midpoint (three blocks from the source). Forty double lines route signals to every first or second block away in all four directions. Organized in a staggered pattern, double lines can be driven only at their endpoints. Double-line signals can be accessed either at the endpoints or at the midpoint (one block from the source). The direct connect lines route signals to neighboring blocks: vertically, horizontally, and diagonally. The fast connect lines are the internal CLB local interconnections from LUT outputs to LUT inputs. In addition to the global and local routing resources, dedicated signals are also available.

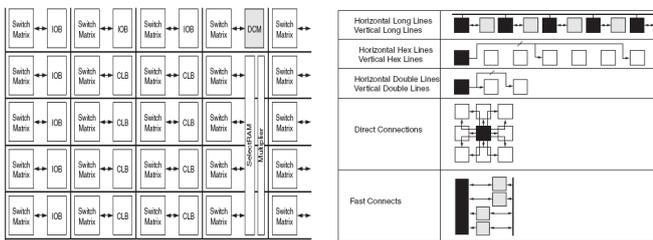


Fig. 1. Routing Resources [1]. Fig. 2. Hierarchical Routing Resources [1].

B. Timing Behavior and Delay Fault Analysis of LUT

From [8]-[7] an n input LUT can be represented as n cascaded stage of SRAM cell as shown in Fig. 3.

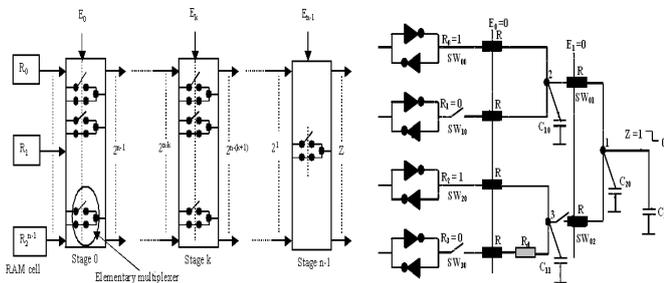


Fig. 3. n-input LUT [7]. Fig. 4. 2-input LUT with resistive open [7].

where E_0, E_1, \dots, E_{n-1} are LUT input and $R_0, R_1, \dots, R_2^{(n-1)}$ are corresponding values of the implemented functions in the SRAM cells. Z is output of the last stage of LUT output. Every stage is a one-dimensional array of vertical multiplexer made of two data input and one select line. A path connects one SRAM cell on the left to the output Z if the all switches on that path are ON. So for P_i to be active the entire switch SW_{kn} should be ON. All the paths are

associated with a unique input configuration I_i Where,

$$I_i = (E_0, E_1 \dots E_{n-1}) \Leftrightarrow P_i$$

The dynamic behavior of LUT can be explained by modifying the model of Fig.3 with RC component [7] as shown in Fig.4 where C_L is load capacitor.

To describe the switching behavior of the active path we have to consider the initial stage of the capacitor C_L and C_{kx} and the final pattern (value) in response to the input I_i . According to [7] “the largest propagation delay is obtained when input pattern generates transition on the input which is close to SRAM cell” (input is E_0). Let for 2 input LUT as shown in Fig.4 has initial output is ‘1’ with initial input pattern (0,0). The capacitor C_{20} at node 1 and C_{10} at node 2 will be set to V_{dd} . Say if next input to LUT is (1,0) then both the capacitor C_{20} and C_{10} will be set to GND. High resistance R_d may get induced in the switching path, because of resistive open in drain or source of the transistor. The time constant of the capacitor C_L and C_{kx} will change, -- hence it will add delay in the path, when complementary signal passes through that path which will in turn produce incorrect values due to switching time difference. This may be modeled as bridging fault or an open circuit that exists for a short duration of time. For 2 input LUT shown in Fig.4, let initially (E_0, E_1) was (0,0) and changed to (1,1). Due to difference in switching speed it will change as follows [00->01->11] or [00->10->11]. Hence it will produce intermediate Bridging fault or open fault at node-1, node-2, and node-3 associated with respective branch B_{ky} . Similarly for all other changes in input possible fault are summarized in Fig.5. From the above discussion it may be concluded that, slow-to-rise (StR), slow-to-fall (StF) and small delay fault in a branch B_{ky} can be determined by applying input pattern I_i such that it will produce complementary output.

INITIAL	E0	E1	FINAL	E0	E1	Type of Fault in Node		
						1	2	3
0	0	0	0	0	No change			
0	0	0	1	B/O	X	X	X	
0	0	1	0	X	B/O	X		
0	0	1	1	B/O	B/O	B/O		
0	1	0	0	B/O	X	X		
0	1	0	1	B/O	B/O	B/O		
0	1	1	0	No change				
0	1	1	1	X	X	B/O		
+1	0	0	0	0	X	B/O	X	
+1	0	0	1	B/O	B/O	B/O	B/O	
+1	0	1	0	No change				
+1	0	1	1	B/O	X	X		
+1	1	0	0	B/O	B/O	B/O		
+1	1	0	1	X	X	B/O		
+1	1	1	0	B/O	X	X		
+1	1	1	1	1	1	No change		

No fault = X Bridge fault/Open fault = B/O Largest propagation delay = *

Fig. 5. Possible Bridging fault in 2-input LUT (with respect to Fig. 4.).

C. Methods used in paper [8][9]

Two methods were discussed in [8]-[9] to detect delay fault of LUT. In one test configuration scheme k-number of LUT is connected in chain. Output of first stage is connected to the a_0 th input of the next stage and so on. Each LUT was configured with function $f(E_0, E_1 \dots E_{n-1}) = E_0$. Though this system can detect delay fault but it has few disadvantages. Those are delay between input pad and the output pad will deteriorates detection capacity and testing frequency. It cannot locate the faulty area.

Inserting a D-flip-flop between each stage paved way for the second test configuration from first testing configuration. To detect the small delay fault, StR and StF faults, LUT was configured with functions $f(E_0, E_1 \dots E_{n-1}) = E_0$ and $f(E_0, E_1 \dots E_{n-1}) = \bar{E}_0$. This method also suffers from few drawbacks. First if any of the flip-flops is faulty then its delay

will be added with the total path delay, which will lead to wrong conclusion. Secondly the time delay of LUT is very difficult to know, so it will be very difficult to latch the faulty value. Moreover, the long wire used to transmit clock may also have some delay. Since latching the faulty value is the critical part of the proposed testing technique, hence it is bound to make the detection of delay fault difficult.

III. PROPOSED TESTING CONFIGURATION

A. Block Under Test (BUT) Architecture

In order to overcome the drawbacks discussed in section-2.C a new method to diagnose the delay fault is proposed. The BUT similar to that as used in [8]-[9] is configured, but with necessary modification. As long wires and local wires will be used by the compiler to connect from TPG to BUT and within BUT i.e. from LUT to LUT. Hence it may get affected by the delay, which exists between long wires and the local wires. To diagnose the cause of the defect, the effect of one fault (LUT /long wire/ local wire delay fault) was quarantined from affecting another. In order to do that a new scheme is proposed as shown in Fig.6. Here there are k-number of LUT connected in chain. Output of the leftmost LUT is connected to the input pin a_0 of the next stage and so on. A D flip-flop is inserted between first two LUT from the left. As we have discussed above the long wire and short wire may have different time delay, so to isolate this delay from affecting the LUT delay a D flip-flop is inserted. As a result the left most LUT will become an extended part of TPG, hence it will be non-testable. All LUTs will be configured with function $f(E_0, E_1 \dots E_{n-1}) = E_0$. The output of first LUT will ripple through all LUTs. If any delay occurs in the path it will be reflected in the output of the last LUT. The delay will be determined by comparing the output of two BUT in ORA. The time period of the clock of D flip-flop will be greater than maximum time required for a signal to reach the last LUT by long wire from TPG. This scheme can detect slow-to-rise (StR), slow-to-fall (StF) and delay fault in LUT. And to detect short delay fault between long and the local wire the BUT will be the same as in [8] and as shown in Fig.7. All LUTs will be configured with function $f(E_0, E_1 \dots E_{n-1}) = E_0$.

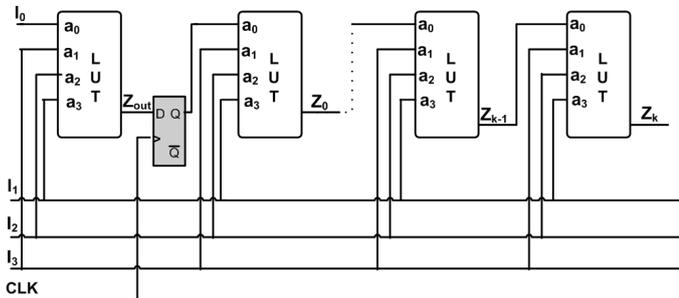


Fig. 6. Proposed configuration of BUT

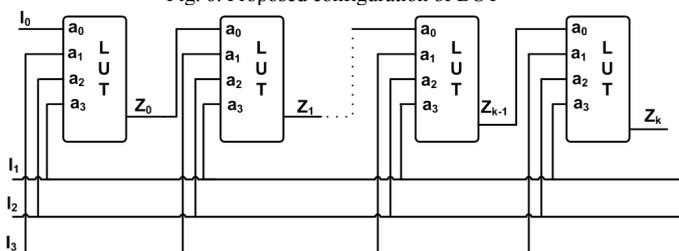


Fig. 7. Configuration of BUT to detect short delay fault [8].

If true output is received while testing using configuration of Fig.6 and then false output will result when BUT is configured as shown in Fig.7, then it can be concluded that the delay fault is due to delay between long wire and short wire.

B. Output Result Analyzer and Test pattern Generator

To compare and analyze the output of two BUT the proposed ORA structure is shown in Fig.8. A two input XOR gate will compare the inputs from two BUTs. From Fig.8 when there is no delay the XOR gate will produce a '0'. When a small delay occurs the XOR gate will produce two transitions as shown in Fig.8 and the T flip-flop will produce square wave whose duration is same as that of input wave. But when slow-to-rise (StR) or slow-to-fall (StF) event occurs T flip-flop will produce square wave whose time duration is twice the time duration of input wave. While connecting two BUT output to ORA it may so happen that compiler may use two different wire types with unequal time delay. In these circumstances ORA may give false result. To avoid this polling is used. The decision of the majority vote will be declared as the final result. The modified ORA and its decision table is shown in Fig.9 where T_0, T_1 and T_2 are output of XOR gate.

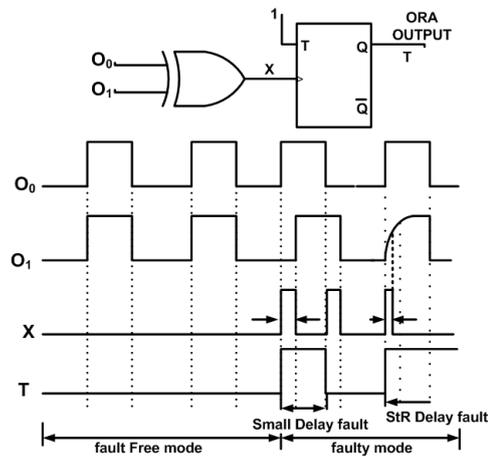
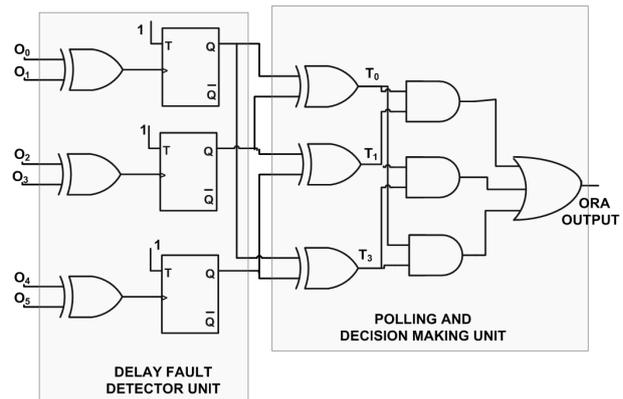


Fig. 8. ORA scheme and its response to input pattern



(a)

T_0	T_1	T_2	ORA OUTPUT
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

F = 1 = False T = 0 = True

(b)

Fig. 9. a) Modified ORA scheme. b) Decision table.

TPG is a FSM which will generate 2^n test patterns for n-input LUT of length n bit ,say E_0, E_1, \dots, E_{n-1} . The output E_0 will only go to the a_0 input of first LUT from left (refer to Fig. 6) and rest of the bits E_1, \dots, E_{n-1} will go to all input ($a_1 \dots a_n$) of LUTs. An additional pulse generator will be required for TPG used in configuration of Fig. 6.

IV. IMPLEMENTATION OF BIST

For accurately testing the delay fault the BISTer architecture is shown in Fig 10 and Fig.11. There are two BISTer structures slightly different from one another; For example BISTer-1 that will do test-1 and BISTer-2 that will perform test-2. At first the FPGA is configured as BISTer-1 to perform test-1 and then the FPGA will be configured in BISTer-2 to perform test-2. BISTer-1 has a TPG that will produce 2^n test sequence of n-bit length where n is number of inputs in LUT. The TPG will generate a pulse known as “test start signal” for the D-flip-flop. The ORA is same as that shown in Fig.9. Two BUTs configuration is shown in Fig.6. After performing test-1 the FPGA will be configured as BISTer-2 whose structure is similar to that of BISTer-1 except that-- TPG does not have to generate “test start signal” and the BUT is same as shown in Fig.7. The time period of TPG clock is greater than the time delay of a signal to propagate from input of first LUT to output of last LUT in order to avoid the overlapping of two consecutive test patterns.

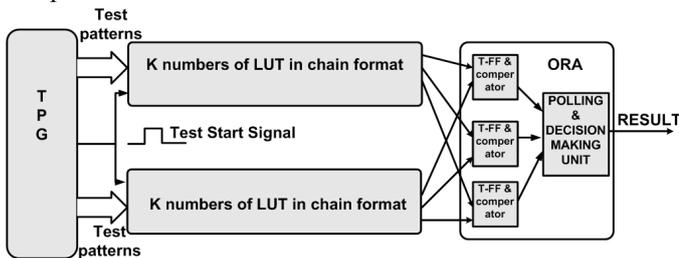


Fig. 10. BISTer-1 structure

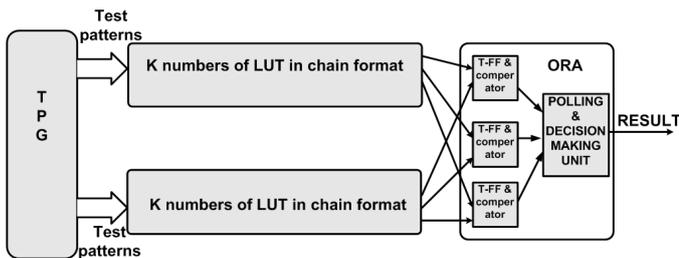


Fig. 11. BISTer-2 structures

By analyzing the result of the two schemes conclusion can be drawn in following way as shown in table-1. If Test-1 did not detect any time delay and Test-2 detects small delay fault then the fault was between long wire and local wire. Next if Test-1 detects StR/StF fault and test-2 detect small delay fault then it was concluded that the fault is in LUT and between long wire and the local wire. And lastly if Test-1 and Test-2 both detects StR/StF fault then it can be concluded that the delay fault is present in LUT.

Systematically removing LUT one by one and then performing the test till no fault occurs can determine the location of the fault. An example of this testing scheme is shown in Fig.12. Different sections of the scheme such as TPG, ORA and BUT are shown in shaded boxes. Dark lines

represent the long wires and thin lines represent the local wires. The signal from BUT to ORA can be transmitted by either long wires or local wires depending on the position of BUT and ORA input section.

TABLE I: FAULT TYPE

Test-1	Test-2	Conclusion
No time delay	Small delay	Fault between long and local wire
StR/ StF	Small delay	Fault is in LUT and between long and local wire
StR/ StF	StR/ StF	Fault is in LUT

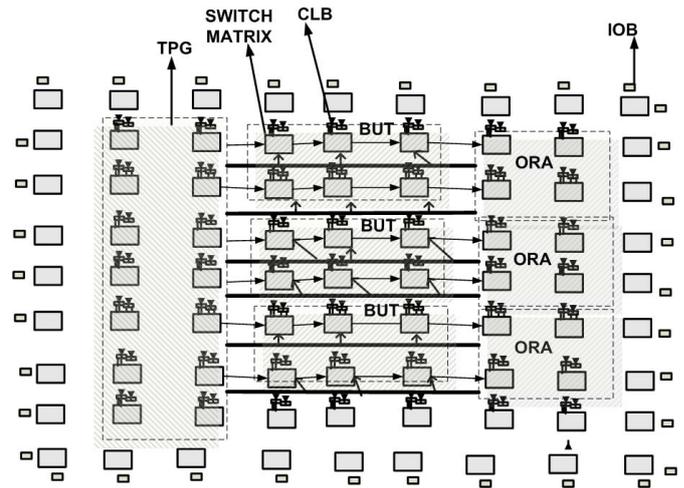


Fig. 12. BIST configuration of the scheme

To validate our testing scheme we have implemented it in Xilinx [1] Virtex-II xc2v1000 Architecture. Four LUTs are used to make BUT. “FPGA editor Tool” was used for coding, placing and routing. Modelsim XE III 6.0 was used for simulation. To implement the ORA shown in Fig.9a three flip-flops and four LUTs were used. First, fault free mode is simulated and then a delay was injected in a_2 path of third LUT. The simulation result is shown in Fig.13. The simulation started with resetting the TPG shown by ‘reset_BIST’ signal line. Four TPG outputs are shown and the response of BUTs with respect to test patterns are represented by signal line ‘but1’ and ‘but2’. When simulated in fault-free condition but1 and but2 signal lines have same patterns, so the ORA out was ‘0’ shown by signal line ‘result’. At fault mode the ‘but2’ first high pulse of signals were stretched by an amount that equals the slow-to-fall (StF) time delay. Then the ‘result’ signal line produces a transition indicating the presence of delay fault as shown in Fig.13.

The test time required for proposed testing scheme is independent of the diagnosis resolution. As the testing of the blocks was done in a parallel manner, so that the overall test time will be represented by the highest test time of the individual test set. One test set consists of TPG, BUTs and the ORA. If processing delay of TPG and ORA is T_1 and T_2 , and the over-all transport delay of two BUTs are T_{31} and T_{32} then test time for a single test sequence will be $T_1 + T_2 + T_{31}(T_{32})$. Therefore total test time for ‘n’ number of test sequence is $n*[T_1 + T_2 + T_{31}(T_{32})]$. Here $T_{31}(T_{32})$ means if T_{31} and T_{32} are the delay of BUT-1 or BUT-2 whichever is greater will be taken into account.

Run-time environment for on line implementation is

JBits3.0 API which is used to configure individual test set in those areas of FPGA that are currently not in use.

V. CONCLUSIONS

In this paper a novel technique to detect delay fault in LUT and the interconnects of LUT in cluster based FPGA is presented. The testing scheme was simulated with Modelsim

XE III 6.0 for Xilinx Virtex-II xc2v1000. The proposed technique is found to overcome the drawbacks of the previously used methods. The LUTs were connected in a chain format where output of first LUT was connected to a_0^{th} input of next LUT. TPG was a FSM connected to both the BUTs. An ORA was constructed with T-flip-flop which produces transition whenever a delay fault occurs. Polling technique is used to determine the exact result.

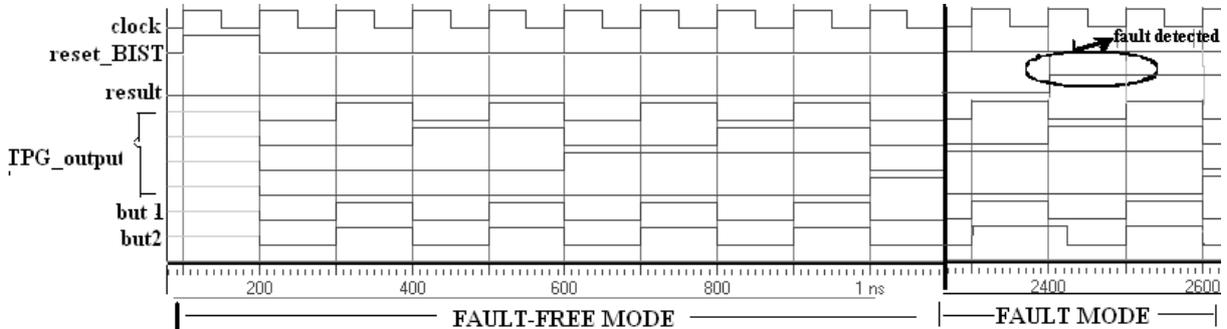


Fig. 13. BIST simulation result

Clock- applied clock signal to the system, **reset_BIST-** signal to reset the test configuration, **result-** out put of ORA (out put result analyzer),
TPG_output- out of TPG(test pattern generator) **but 1, but 2-** out put of block under test (but). **FAULT-FREE MODE-** when both the BUT has same time delay.
FAULTY MODE – when the BUTs have different time delay.

REFERENCES

[1] Xilinx Corp. [Online]. Available: <http://www.xilinx.com>
 [2] C. Stroud, S. Wijesuriya, C. Hamilton, "Built-In Self Test of FPGA Interconnect," *International Test Conference*, pp. 404-411, 1998.
 [3] I. G Harris, P R Menon, R Tessier "BIST-Based Delay Path Testing in FPGA Architectures," in *Proc. Of IEEE International Test Conference*, 2001, pp. 932-938.
 [4] M. Abramovic, M. A. Breuer, A. D. Friedman, *Digital System Testing and Testable Design*, Wiley-Addison, 1994.
 [5] S. Dutt, V. Verma, and V. Suthar, "Build-In-Self-Test of FPGA with Provable Diagnosabilities and High Diagnostic Coverage With Application to Online Testing," *IEEE Transactions on computer aided design of intregated circuits and system*, vol. 27 pp. 309-326, 2008.
 [6] L. Lai, W T Cheng, and T Rinderknecht, "Programmable scan based logic build-in-self-test," in *Proc. 16th IEEE Asian Test Symposium* , 2007, pp.371- 377.
 [7] P. Girad, O. Heron, S. Pravossoudovitch, and M. Renovell, "Defect Analysis for Delay-Fault in FPGAs," in *Proceeding of the 9th IEEE IOLTS*, 2003, pp- 1-5.
 [8] P. Girad, O. Heron, S. Pravossoudovitch, and M. Renovell, "Delay Fault Testing of Look-Up Table in SRAM-Based FPGA," *Journal of Electronic Testing: Theory and Application*, vol. 21, pp. 43-55, 2005.
 [9] P. Girad, O. Heron, S. Pravossoudovitch, and M. Renovell, "An Efficient BIST Architecture for Delay Fault in the Logic Cell of Symmetrical ARAM-Based FPGAs," *Journal of Electronic Testing: Theory and Application*, vol. 22, pp.161-172, 2006.
 [10] J. Smith, T. Xia and C. Stroud, "An Automated BIST Architecture for Testing and Diagnosing FPGA Interconnect," *Journal of electronic testing*, vol. 22, pp. 239-253, 2006.
 [11] S. A. Guiccione and D. Levi," XBI: A java based interface to FPGA Hardware," Configurable computing Technology and its Uses in High Performance Computing, DSP and System engineering, Proc. SPIE Photonics East, J.Schewel (Ed.), SPIE. The International Society for Optical Engineering, Bellingham, WA, November, 1998.
 [12] N. Das, S. Ghosh, and H. Rahaman, "Detection of Single Stuck-at and Bridging Faults in Cluster-based FPGA Architectures," in *Proc. of IEEE VDAT* , 2007, pp. 205 – 212.
 [13] N. Das, P. Roy, and H. Rahaman, "On Line Testing of Single Feedback Bridging Fault in Cluster Based FPGA by using Asynchronous element," in *Proc. of IEEE IOLTS* ,2008, pp.190-191.
 [14] I. G. Haris and R. Tessier "Testing and Diagnosis of Interconnect Faults in Cluster-based FPGA Architecture," *IEEE Transaction on*

Computer-Aided Design of Integrated Circuits and Systems, vol.21, no. 11, pp. 1337-1343, Nov 2002.
 [15] F. Lima, C. Carcamichael, J. Fabula, R. Padovani, and R. Reis, "A Fault Injection Analysis of Virtex FPGA TMR Design Methodology," in *Proc. of the Radiation Effects on Components and Systems Conference*, 2001.
 [16] M. Renovell, J. M. Portal, J. Figueras, and Y. Zorian, "Testing the Interconnect of RAM-Based FPGAs," *IEEE Design and Test of Computers*, vol. 15, no. 1, 1998, pp 45-50.
 [17] M. Abramovici and C. E. Stroud, "BIST based delay fault testing in FPGA," *Journal of electronic testing*, vol.19, pp 549-558, 2003.



Nachiketa Das is a research scholar in School of VLSI Technology at Bengal Engineering and Science University, Shibpur, India. He got the bachelor degree in electronics and communication engineering from Institute of Electronics and Telecommunication Engineers, India. He received his masters in Digital electronics and advanced communication in 2005. His main area of research is design and testing of Field Programmable Gate Array.



Hafizur Rahaman received the B.E. degree in electrical engineering from Bengal Engineering College, Calcutta University, Calcutta, India, in 1982 and the M.E. degree in electrical engineering and Ph.D. degree in computer science and engineering from Jadavpur University, Calcutta, in 1988 and 2003, respectively. During 2006–2007, he visited the Department of Computer Science, Bristol University, Bristol, U.K., as Postdoctoral Research Fellow. He is currently chairing the Department of Information Technology, Bengal Engineering and Science University, Howrah, India. His research interests include logic synthesis and testing of VLSI circuits, fault-tolerant computing, Galois-field-based arithmetic circuits, and quantum computing. Dr. Rahaman has served on the Organizing Committees of the International Conference on VLSI Design in 2000 and 2005 and as the Registration Chair for the 2005 Asian Test Symposium, which was held in Calcutta.



Indrajit Banerjee is an assistant professor in the School of VLSI technology at Bengal Engineering and Science University, Shibpur, India. He got the bachelor degree in mechanical engineering from Institute of Engineers, India. He received his masters in Information Technology from Bengal Engineering and Science University in 2004. He is currently pursuing his Ph. D. in Information Technology in Bengal Engineering & Science University. His main research interests are cellular automata, wireless ad hoc and sensor network.