BUILT-IN SELF-TEST DESIGN FOR FAULT DETECTION AND FAULT DIAGNOSIS IN SRAM-BASED FPGA

By

S.KRISHNADEVI

Reg. No. 1020106011

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University of Technology, Coimbatore)

COIMBATORE – 641 049

A MINI PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF ENGINEERING

IN

APPLIED ELECTRONICS

MAY 2011

BONAFIDE CERTIFICATE

Certified that this project report entitled "**BUILT-IN SELF-TEST DESIGN FOR FAULT DETECTION AND FAULT DIAGNOSIS IN SRAM-BASED FPGA**" is the bonafide work of **S.KRISHNADEVI** [Reg. no. 1020106011] who carried out the miniproject under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

K. Thilagavathi 18/5/11
Project Guide

Ms.K.Thilagavathi M.E.,

Head of the Department

Dr. Ms. Rajeswari Mariappan

The candidate with university Register No.**1020106011** was examined by us in the project viva-voce examination held on 23.05.2011.....

Internal Examiner
23/5/11

External Examiner
23/5/11

ACKNOWLEDGEMENT

A project of this nature needs co-operation and support from many for successful completion. In this regards, I would like to express my thanks and appreciation to the many people who have contributed to the successful completion of this project. Firstly I thank **Dr.J.Shanmugam Ph.D**, Director, for providing me an opportunity to carry out this project work.

I would like to thank **Dr. S. Ramachandran Ph.D,** Principal, who gave his continual support and opportunity for completing the project work successfully.

I would like to thank **Dr. Rajeswari Mariappan Ph.D,** Prof of Head, Department of Electronics and Communication Engineering, who gave her continual support for me throughout the course of this project.

I would like to thank **Ms.K.Thilagavathi M.E** Assistant Professor(SRG), for her technical guidance, constructive criticism and many valuable suggestions provided throughout the project work.

My heartfelt thanks to **Ms. R.Hemalatha M.E** Assistant Professor(SRG),Project Coordinator, for her contribution and innovative ideas at various stages of the project and for her help to successful completion of this project work.

I express my sincere gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering department for their support throughout the course of my project.

ABSTRACT

BIST is a design technique that allows a circuit to test itself. The technique can provide shorter test time compared to an externally applied test and allows the use of low-cost test equipment during all stages of production. The use of field-programmable gate arrays (FPGAs) to implement complex logic functions in digital applications has become increasingly common. FPGAs are regular structures of logic modules that communicate through an interconnected architecture of lines and switches. The logic modules and the interconnect structures are programmed to select a particular function of each logic module and specific interconnect paths to realize the global function of the FPGA.

This proposed work presents a built-in self-test (BIST) design for fault detection and fault diagnosis of Static-RAM (SRAM)-based field-programmable gate arrays (FPGAs). The proposed FPGA BIST structure can test both the interconnect resources [wire channels and Programmable Switches (PSs)] and Look Up Tables (LUTs) in the Configurable Logic Blocks (CLBs). The test pattern generator and output response analyzer are configured by existing CLBs in FPGAs; thus, no extra area overhead is needed for the proposed BIST structure. The target fault detection/diagnosis of the proposed BIST structure are open/short faults in the wire channels, stuck on/off faults in PSs, and stuck-at-0/1 faults in LUTs.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BIST:

Bist is the technique which allows the circuit to test itself. In integrated circuits, BIST is used to make faster, less-expensive manufacturing tests. The IC has a function that verifies all or a portion of the internal functionality of the IC. In some cases, this is valuable to customers, as well. For example, a BIST mechanism is provided in advanced field bus systems to verify functionality. At a high level this can be viewed similar to the PC BIOS's Power-On Self-Test (POST) that performs a self-test of the RAM and buses on power-up.

The aim and purpose of BIST is to reduce the complexity, and thereby decrease the cost and reduce reliance upon external (pattern-programmed) test equipment. BIST reduces cost in two ways: (1) reduces test-cycle duration; and (2) reduces the complexity of the test/probe setup, by reducing the number of I/O signals that must be driven/examined under tester control. Both lead to a reduction in hourly charges for automated test equipment (ATE) service. The testing process is performed by configuring the TPG, ORA, and BUT in each test block. Additionally, the TPG and ORA of the proposed FPGA BIST structure are designed using the existing CLBs to reduce the extra area and test cost needed.

1.2 TEST PATTERN GENERATOR:

The TPG is an address generator, which consists of eight MUXs, eight LUTs, and eight DFFs to continuously generate the addresses (0–15) in the LUTs to produce the

algorithm is needed to write different test patterns into LUTs, read the contents of LUTs, an check whether the read data are correct or not. The TPG in Fig. 4 is an address generato which consists of eight MUXs, eight LUTs, and eight DFFs to continuously generate th addresses (0–15) in the LUTs to produce the corresponding test patterns. Significantly, t ensure the accuracy of LUTs in the TPG, an algorithm is needed to write different tes patterns into LUTs, read the contents of LUTs, and check whether the read data are correct o not In other words, according to the MUX selections, the LUTs' addresses can be figured ou in sequence, and the test patterns can be generated by repeatedly checking the contents.

1.3 OUTPUT RESPONSE ANALYZER:

The ORA design also uses the existing CLBs in an FPGA. The two-input XOR gat and three-input OR gate in the ORA are used to test the global interconnect open and shor faults, respectively. Otherwise, the faults presented in the LUTs and local interconnects ar detected by using the four-input parity checker. The logic diagram of a simple parity checke is composed of a 4-b XOR and a unit delay. Significantly, the parity checker can b implemented with the internal logic gates and a DFF of a CLB. Moreover, the P/F checke (functioning in the same way as a simple OR gate) in the ORA circuit is designed to generate a P/F signal to trigger the fault pass circuit to deliver the faults from the primary outputs of a BUT to the fault dictionary for fault diagnosis.

1.4 FAULT DIAGNOSIS:

Fault detection is an important step in guaranteeing the quality of the product. Moreover, fault diagnosis is required, particularly if the manufacturer plans to enhance the yield or if the user intends to tolerate the faults. The target diagnosis in this project is arrived

SRAM-based FPGA is diagnosed here in two steps: 1) BUT diagnosis and 2) fault-typ diagnosis. In step 1)the faulty BUTs can be located by directly observing the output signals i.e., P/Fs, of ORAs. The fault-type diagnosis in step 2) is performed by comparing the fault between the outputs of the faulty BUT and the fault dictionary.

1.5 SOFTWARE USED:

- ModelSim XE III 6.2g

ORGANIZATION OF THE REPORT:

- **Chapter 2** discusses about the SRAM-based FPGA Architecture.
- **Chapter 3** discusses about testing considerations made in SRAM-based FPGA Architecture.
- **Chapter 4** discusses the proposed FPGA BIST design.
- **Chapter 5** discusses the fault models in the proposed FPGA BIST design.
- **Chapter 6** discusses the test pattern generation and fault diagnosis.
- **Chapter 7** discusses the simulation results.
- **Chapter 8** shows the conclusion of the project.

CHAPTER 2

SRAM- BASED FPGA ARCHITECTURE:

SRAM-based FPGAs are of special interest due to their compatibility with mainstream integrated circuit fabrication technology, as well as because of their wide use in practical applications. The SRAM-based FPGA (Fig. 1) consists of an array of n × n CLBs and local/global interconnect re-sources. The CLBs can be programmed with configuration cell data to generate logical functions. The set of all configuration cell data makes up an FPGA configuration. The basic internal architecture of a CLB is made up of three components: 1) look-up tables (LUTs); 2) multiplexers (MUXs); and 3) D-type flip-flops (DFFs).

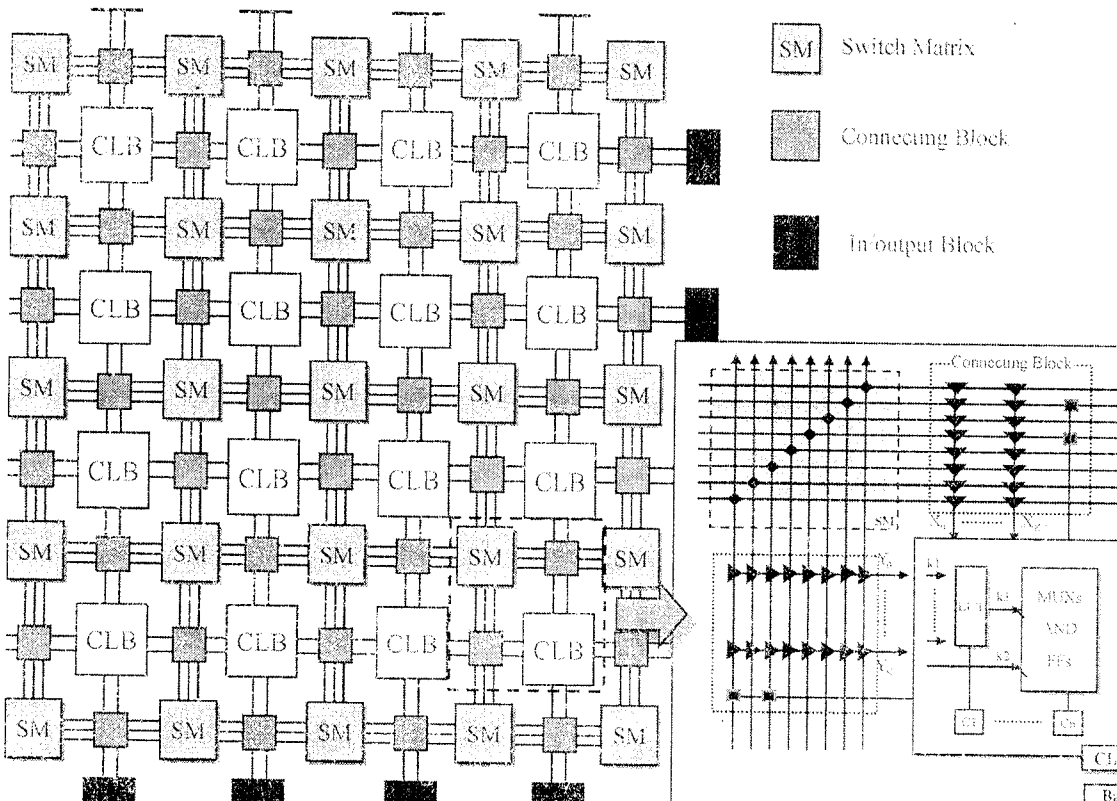

Fig. 1. SRAM-based FPGA architecture.

In Fig. 1, the LUT's implement either any logical function with k1 inputs or RAM. Every CLB is configured with configuration cells C1 , . . . , Cn . The number of outputs of LUTs is assumed to be k3, and the inputs k2 directly drive some MUXs or DFFs, producing the CLB outputs. Note that parameters k1, k2, and k3 depend on the FPGA type.

The local interconnects are associated with CLBs, including wire segments and connecting blocks. Note that a connecting block contains some programmable-interconnect-point PSs(PIP-PSs) and multiplexer PSs (MUX-PSs) to bring signals into and out of CLBs. On the other hand, wire segments and programmable cross-point PSs (PCP-PSs) within the switch matrix (SM) in global interconnects form horizontal and vertical routing channels that connect signals between CLBs. The input–output signals can be transmitted into or out of the FPGA using the input–output blocks (IOBs). An SM is a programmable connecting element that receives k wires on each side.The wires connect SM pins identified as north, east, south, and west. Some pairs of pins in an SM cannot be connected; these are called nonconnectable pins. Pairs that can be connected are called connectable pins.

CHAPTER 3

TESTING CONSIDERATIONS:

The FPGA test procedure is a two-part process. Each TC must be first loaded into the FPGA device, and then, test vectors must be applied. This process is repeated for a number of TCs. The PIP-PSs and MUX-PSs in the connecting block and the PCP-PSs in the SM are the essential components for programming TCs under FPGA testing [1]. Fig. 2(a) shows a PIP-PS (square shaped), which contains a pass transistor and an SRAM cell. The SRAM cell can be programmed to open or close the pass transistor. An MUX-PS, illustrated as a set of uni-directional triangles over the horizontal and vertical wire segments, functions as a many-to-one MUX. An MUX-PS allows one of the inputs to be routed to the output for certain selection signals. The selection logic can only be set through the configuration bits during the configuration stage.

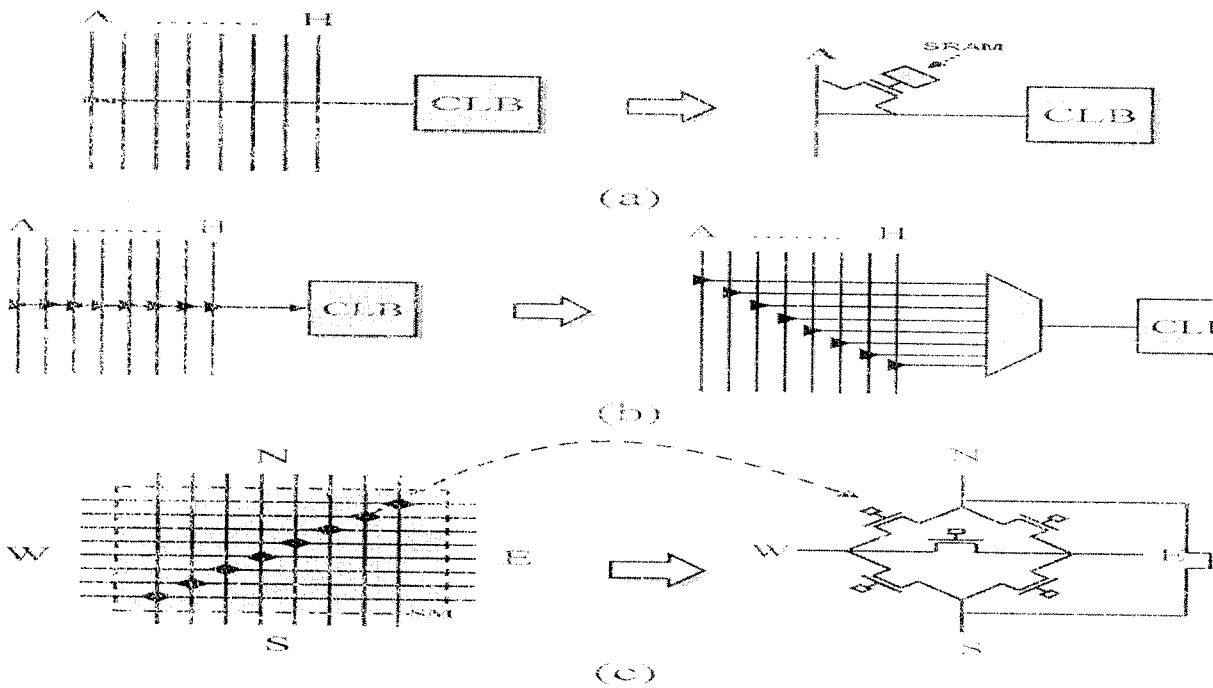

Fig. 2. Three types of PS. (a) PIP-PS. (b) MUX-PS. (c) PCP-PS.

Fig. 2(b) illustrates the symbolic diagram of an eight-to-one MUX-PS and its

shaped box within an SM, which is comprised of six PIP-PSs that can be programmed t connect the horizontal and vertical wire segments in six directions. In other words, a PCP-P connects the wire segments in the west–east (W–E), north–south (N–S), north–west (N–W south–west (S–W), north–east (N–E), and south–east (S–E) directions.

Generally, the TCs are fault graded. If the desired fault coverage is not achieved, tes engineers must develop additional test vectors and configurations to increase the faul coverage [2]. In other words, the testing of an FPGA chip poses a challenge to test engineers Different configurations (programmings) of the FPGA are required, and so, certain question arise. Which TC and corresponding vector test sequence (TS) is needed to cover the faults o a certain structural fault model for the FPGA? When are TCs and test vectors generated for a given FPGA? It is extremely important to bear in mind the high cost of changing the configuration due to reprogramming costs. The configuration loading process is performed through a serial access, and so, this process is really time consuming. Consequently, when test generation for an FPGA is performed, the main objective is to minimize the number o TCs.

CHAPTER 4

PROPOSED FPGA BIST DESIGN:

Bist is the technique which allows the circuit to test itself. In integrated circuits, BIST is used to make faster, less-expensive manufacturing tests. The IC has a function that verifies all or a portion of the internal functionality of the IC. In some cases, this is valuable to customers, as well. For example, a BIST mechanism is provided in advanced field bus systems to verify functionality

Although BIST techniques in general are associated with high performance, they are also associated with high area overhead incurred by on-chip test hardware. However, the BIST overhead is not an issue for FPGA BIST because the test hardware is easily reconfigured by inserting and removing test pattern generators (TPGs) and ORAs. This is particularly important for the testing of FPGAs. The testing strategy of the proposed FPGA BIST structure is to configure groups of ten CLBs into a test block, as illustrated in Fig. 3. In each test block, four CLBs are configured as a TPG to generate the addresses for test patterns. Additionally, two CLBs are configured as an ORA for comparison with each output of the block under test (BUT) to observe the test results.

The global/local interconnect resources and CLBs in a BUT, which are configured by four CLBs in a test block, are then sequentially tested. To guarantee the testing of all global/local interconnect resources and CLBs, the FPGA has to be reconfigured to shift the test blocks for testing. The test processes of the proposed FPGA BIST structure are simultaneously performed by a BIST controller, which repeatedly reconfigures the test blocks for testing.

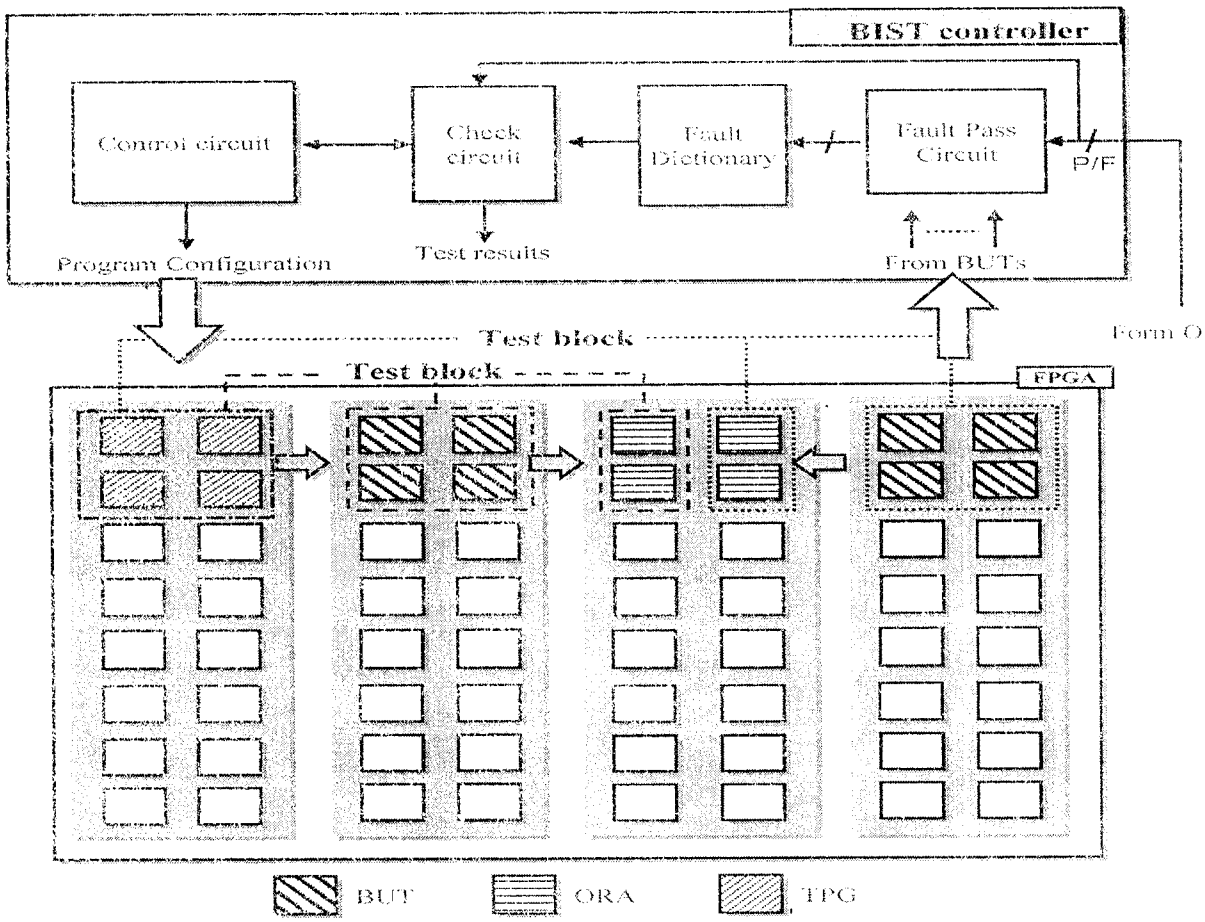

Fig. 3. Proposed FPGA BIST structure.

Briefly, the testing processes can be summarized in the following steps.

1) Reconfigure the FPGA to create test blocks.

2) Program the TCs.

3) Initiate the TS for global/local interconnect resources and CLBs.

4) Generate the test vectors.

5) Analyze the test results.

In other words, the test blocks are first (re)configured by the BIST controller. Second, the TCs should be reconfigured for global/local interconnect resource and CLB testing. Then, the LUT-based method is used to configure the TPG and ORA to generate the test vectors. Finally, the test results are analyzed.

CHAPTER 5

FAULT MODELS:

The aim of this project is to design a BIST structure for both CLB and interconnect resource testing in SRAM-based FPGAs. The faults in this paper can be categorized into four major groups, namely, open/short, stuck-on/off, stuck-at-0/1, and interconnect delay faults. Fig. 5 illustrates all the fault models. The stuck-on/off faults appear in the pass transistor of PIP-PSs or MUX-PSs in local interconnects, while open/short faults occur on PCP-PSs or wire segments in global interconnects. Significantly, the delay fault is presented with a path under test (PUT). On the other hand, the stuck-at-0/1 faults can be found in the LUTs of CLBs. Note that it is assumed that IOBs were previously tested. This assumption is proper since the IOBs can be tested by the boundary scan [3].Fig. 4(a) shows cases of PIP-PSs and MUX-PSs with stuck-on/off faults in the local interconnects.

Note that a stuck-on/off fault causes the pass transistor in PIP-PSs or MUX-PSs to be permanently on/off, regardless of the value of the SRAM cell controlling the pass transistor in PIP-PSs or MUX-PSs [4].An open fault in the global interconnect is a disconnection of any wires, while a short fault indicates a bridging between two wires. Fig. 4(b) illustrates cases of wire open/short faults.Furthermore, Fig.4(b) shows cases of PCP-PS open/short faults that occur when there is a stuck-off fault and a stuck-on fault in the PIP-PS of the connectable and non connectable directions of wire segments, respectively. Moreover, since the TCs for interconnect resource and CLB testing are the same and the TPG and ORA of the proposed BIST design are built by using the LUTs, the faults in a CLB only consider the stuck-at-0/1 faults on every RAM cell in LUTs [see Fig. 4(c)].

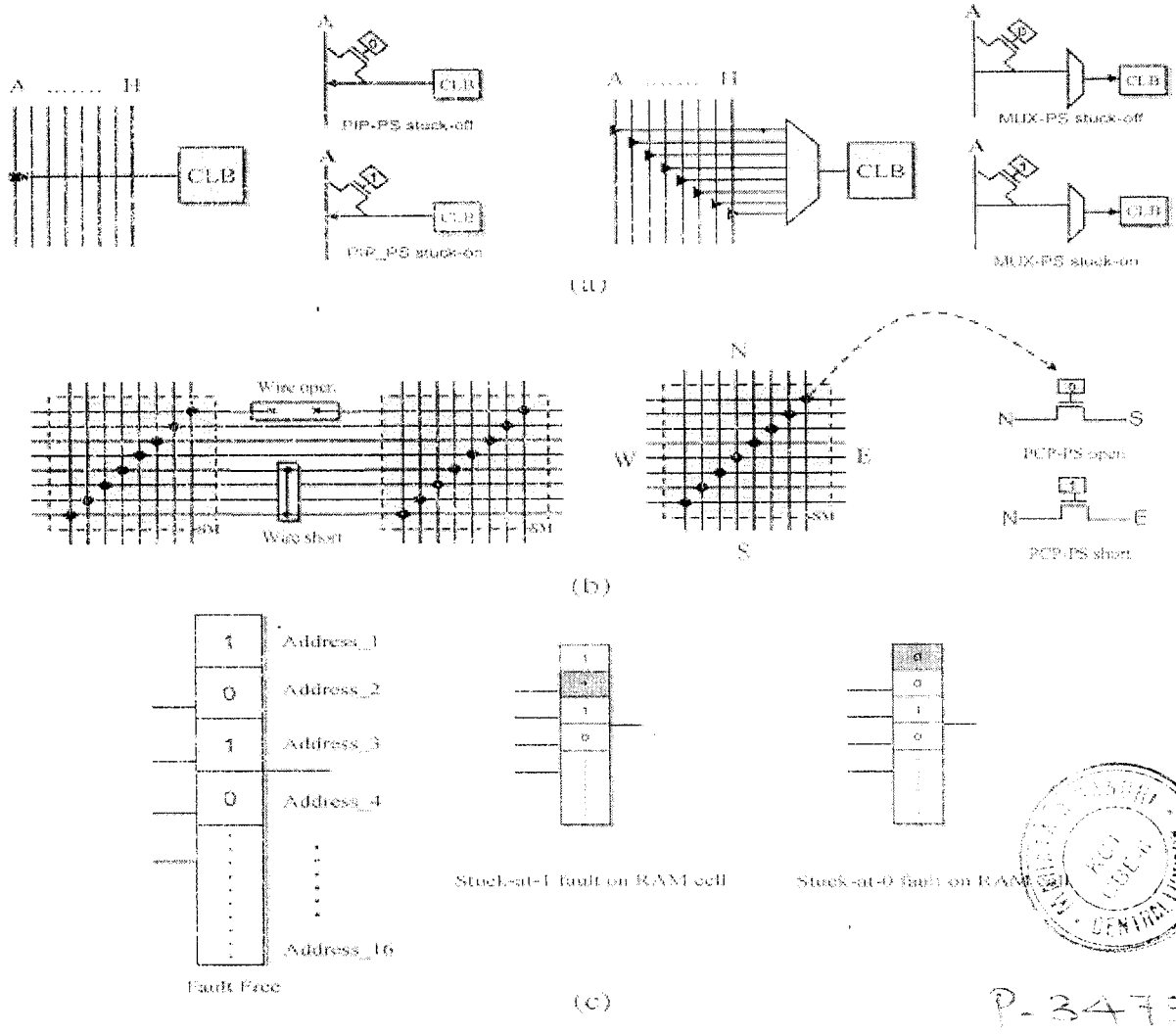

Fig.4. Fault models. (a) Stuck-on/off faults in the local interconnects (b) Open/short faults in the global interconnects. (c) Stuck-at-0/1 faults on a RAM cell in an LUT.

In other words, for an LUT, the fault can occur in any one of the memory cells making it incapable of storing the correct logic value(an LUT has a single-bit output, and therefore, this value is either 0 or 1). Thus, the stuck-at-1 or the stuck-at-0 fault may occur a a memory cell. Note that there is no need to separately consider the stuck-at fault model i interconnects, since these faults can be modeled as short circuits to power supply and ground lines. On the other hand, by setting the clock period to the specification time and generating transition to go through the PUT, we can be determine whether the PUT is fault free or not

Fig. 5(d) shows an example of testing a PUT. If the two DFFs initially store logic 0 and generate a rising-transition propagation from A to B, then logic 1 in DFF2 can be obtained in the fault-free situation. Otherwise, the logic value of DFF2 will remain 0 after a specified time if the PUT is faulty.

Briefly, the fault models in this project are summed up as follows.

1) **PIP-PS stuck-on fault:** The pass transistor of a non connectable PIP-PS is turned on in the local interconnect.

2) **PIP-PS stuck-off fault:** The pass transistor of a connectable PIP-PS is turned off in the local interconnect.

3) **MUX-PS stuck-on fault:** One of the pass transistors of a non selective MUX-PS is turned on in the local interconnect.

4) **MUX-PS stuck-off fault:** One of the pass transistors of a selective MUX-PS is turned off in the local interconnect.

5) **Wire open fault:** A disconnection occurs on any wires in the global interconnect.

6) **Wires short fault:** A bridge occurs between two wires in the global interconnect.

7) **PCP-PS short fault:** A PIP-PS stuck-on fault occurs in the nonconnectable direction in a PCP-PS of wire segments.

8) **PCP-PS open fault:** A PIP-PS stuck-off fault occurs in the connectable direction in a PCP-PS of wire segments.

9) **LUT stuck-at-0 fault:** The RAM cell value of an LUT in the CLB is always 0.

10) **LUT stuck-at-1 fault:** The RAM cell value of an LUT in the CLB is always 1.

CHAPTER 6

6.1 TEST PATTERN GENERATION:

According to the MUX selections, the LUTs' addresses can be figured out in sequence, and the test patterns can be generated by repeatedly checking the contents in Table I. It should be noted that the addresses do not need to be sequentially generated, but they have to cover the completed address space of the LUTs from 0 to 15 during testing. The data in the TPG LUTs are important because on one hand, the combinations of data read from four LUTs have to generate all addresses ranging from 0 to 15, and on the other hand, they have to be memory test patterns in memory cells C1–C8, as shown in Table I. Fig. 9 clearly indicates that outputs 1–4 and 5–8 of the LUTs are configured according to patterns C1, C3, C5, C7 and C2, C4, C6, C8 in Table I. For example, if select lines S1 –S4 and S5 –S8 of the MUXs are set to "0010" and "0011," then the addresses of LUTs 1–4 and LUTs 5–8 are also indicated as "0010" and "0011," respectively. Thus, the corresponding cell values "0010" and "1100" in Table I from C1, C3, C5, C7 and C2, C4, C6, C8 can be captured at the LUTs outputs. Finally, test pattern "00101100" can be generated when the DFFs are enabled. In other words, the different addresses of the LUTs can be obtained by carefully changing the signals of select lines S1 –S4 and S5 –S8 of the MUXs. Then, the series of test patterns can be sequentially captured when the DFFs are triggered at specific times. The ORA design in Fig. 5 also uses the existing CLBs in an FPGA. In Fig. 5, the two-input XOR gate and three-input OR gate in the ORA are used to test the global interconnect open and short faults respectively. Otherwise, the faults presented in the LUTs and local interconnects are detected by using the four-input parity checker. The logic diagram of a simple parity checker is composed of a 4-b XOR and a unit delay [5].

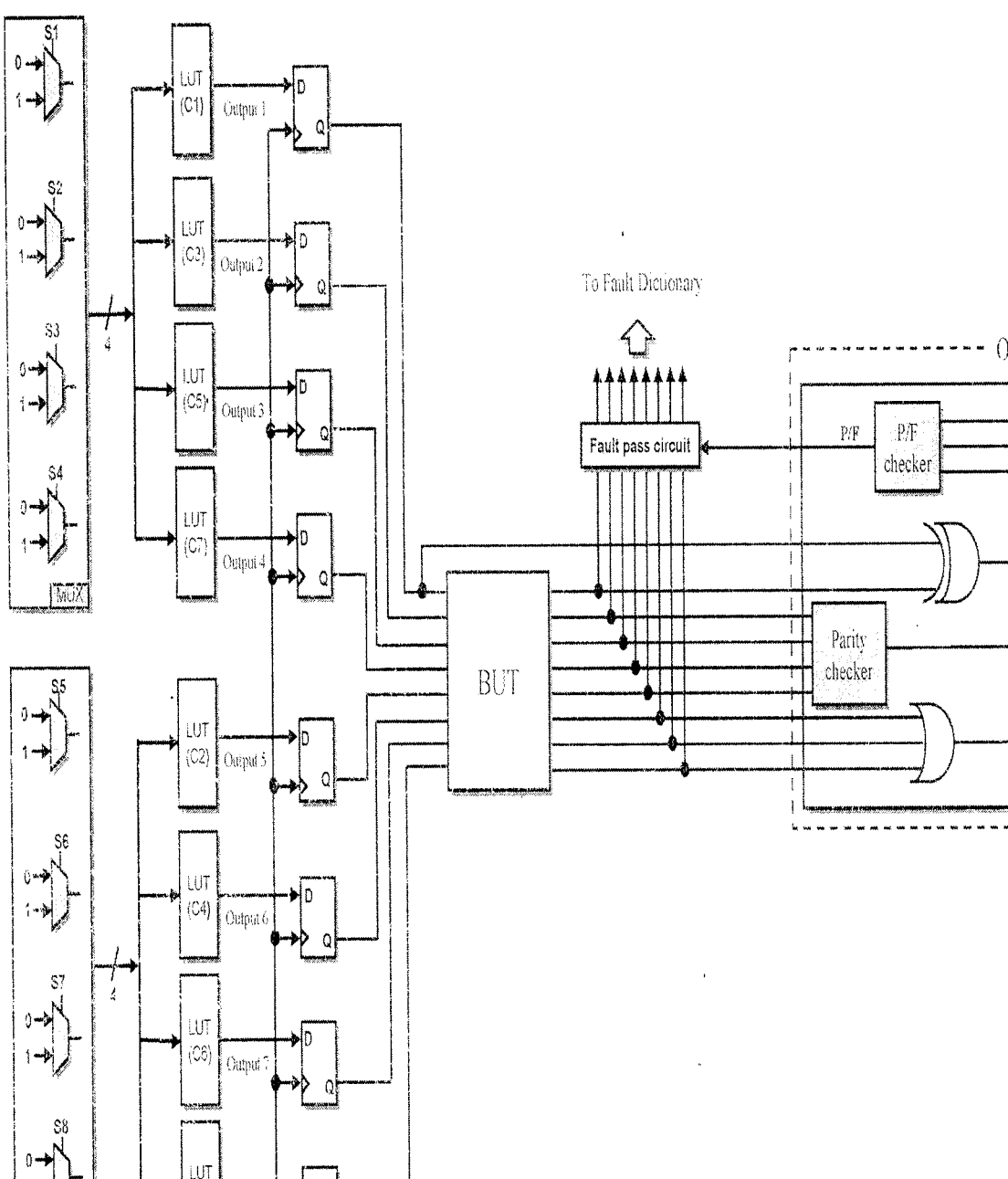

Fig .5.Schema of a BIST structure in a test block.

Significantly, the parity checker can be implemented with the internal logic gates and a DFF of a CLB. Moreover, the P/F checker (functioning in the same way as a simple OR gate) in the ORA circuit is designed to generate a P/F signal to trigger the fault pass circuit to deliver the faults from the primary outputs of a BUT to the fault dictionary for fault

fault diagnosis if the following Boolean equation of an ORA is true:

P/F = (b0_t−1 ⊕ b0_t) □ (b1 ⊕ b2 ⊕ b3 ⊕ b4 ⊕ bxor_4)□(b5 + b6 + b7) where *bi*(*i* = 0, 1, 2, . . . , 7) are the primary outputs of a BUT.*b*0_t−1 and b0_t indicate the primary input and output of the first bit of a BUT, respectively. Additionally, bxor_4 represents the output of the parity checker.

6.2 FAULT DIAGNOSIS:

Fault detection is an important step in guaranteeing the quality of the product. Moreover, fault diagnosis is required, particularly if the manufacturer plans to enhance the yield or if the user intends to tolerate the faults.The target diagnosis in this paper is arrived at by means of locating the faulty BUT and its corresponding fault type. In other words, the SRAM-based FPGA is diagnosed here in two steps: 1) BUT diagnosis and 2) fault-type diagnosis. In step 1),the faulty BUTs can be located by directly observing the output signals, i.e., P/Fs, of ORAs. The fault-type diagnosis in step2) is performed by comparing the faults between the outputs of the faulty BUT and the fault dictionary.

As shown in Fig.6, the 8-b signals from a faulty BUTi will be stored in register i (RGSi) when pass control circuit i (PCCi) is enabled by a faulty signal "1" from ORAi . Notably, all the faulty signals from BUTs in a specific TC can be simultaneously delivered and stored at their corresponding registers. Moreover, the registers are sequentially triggered by the timing circuitry.In other words, the faulty signals from the registers will be transmitted into the fault dictionary in sequence to indicate the fault type.

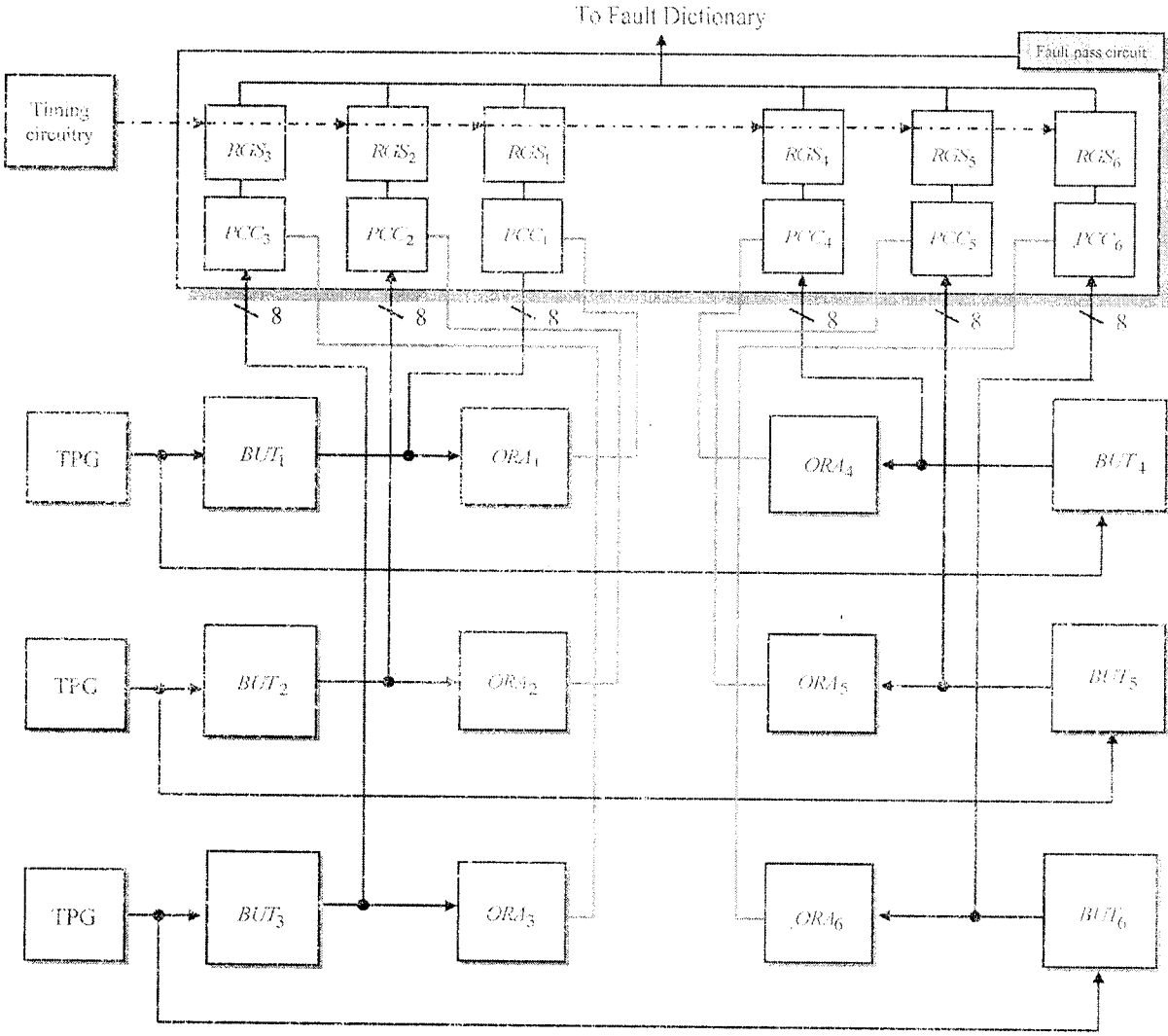

Fig.6. Schema of fault diagnosis

Based on the self-test processes,the CLBs in FPGA can be checked offline to confirm whether the elements in the CLBs are fault free or not. After the self-test processes and according to the proposed FPGA BIST structure a BIST scheme for a test block to detect and diagnose the faults on both interconnect resources and LUTs in CLBs online. The testing process is performed by configuring the TPG, ORA, and BUT in each test block. Additionally, the TPG and ORA of the proposed FPGA BIST structure are designed using the existing CLBs to reduce the extra area and test cost needed.

CHAPTER 7

SIMULATION RESULTS

7.1 Stuck-off fault in the local interconnects.

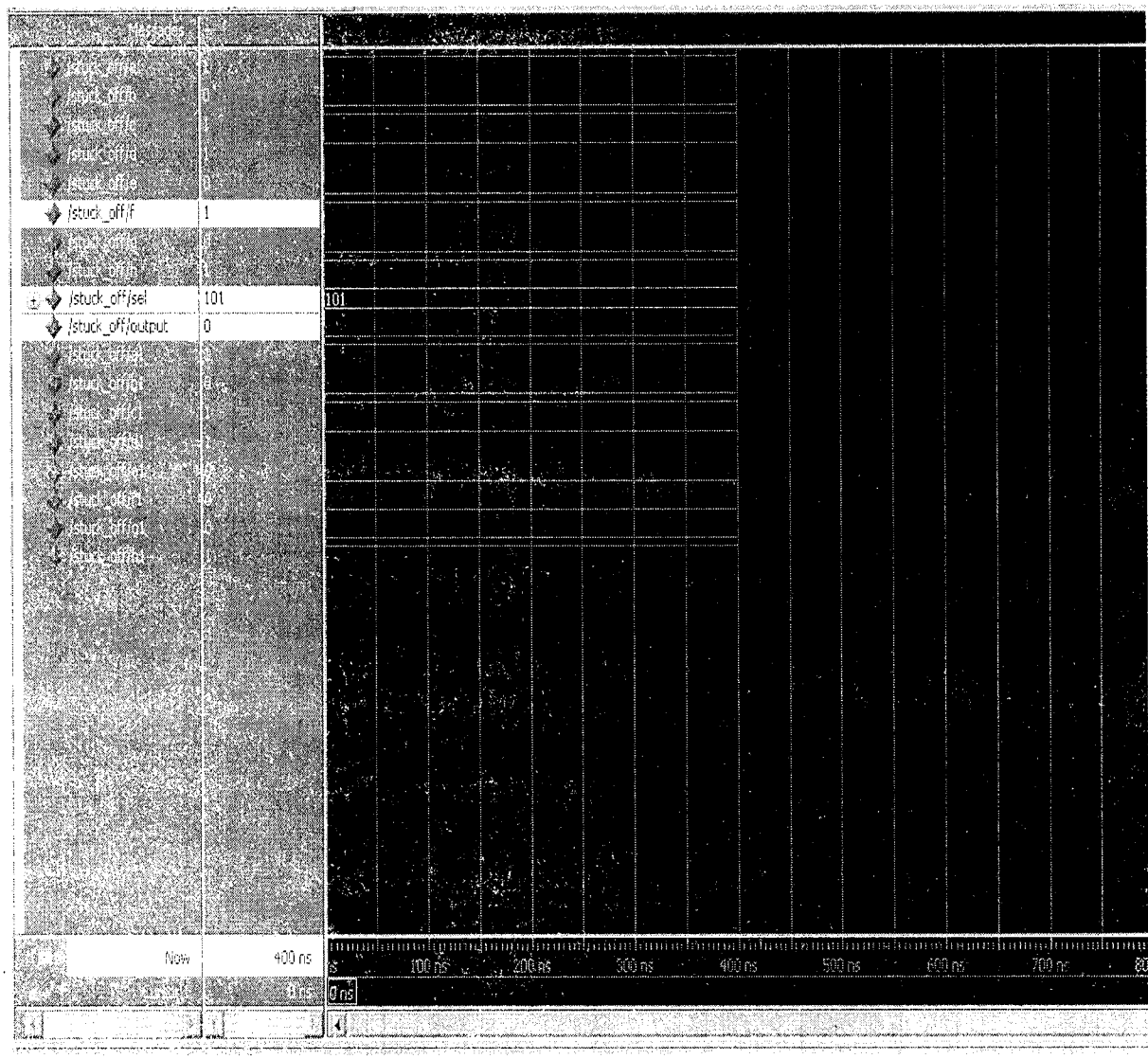

Fig .7. Simulation result of Stuck-off fault in the local interconnects.

7.2 Stuck-on fault in the local interconnects.

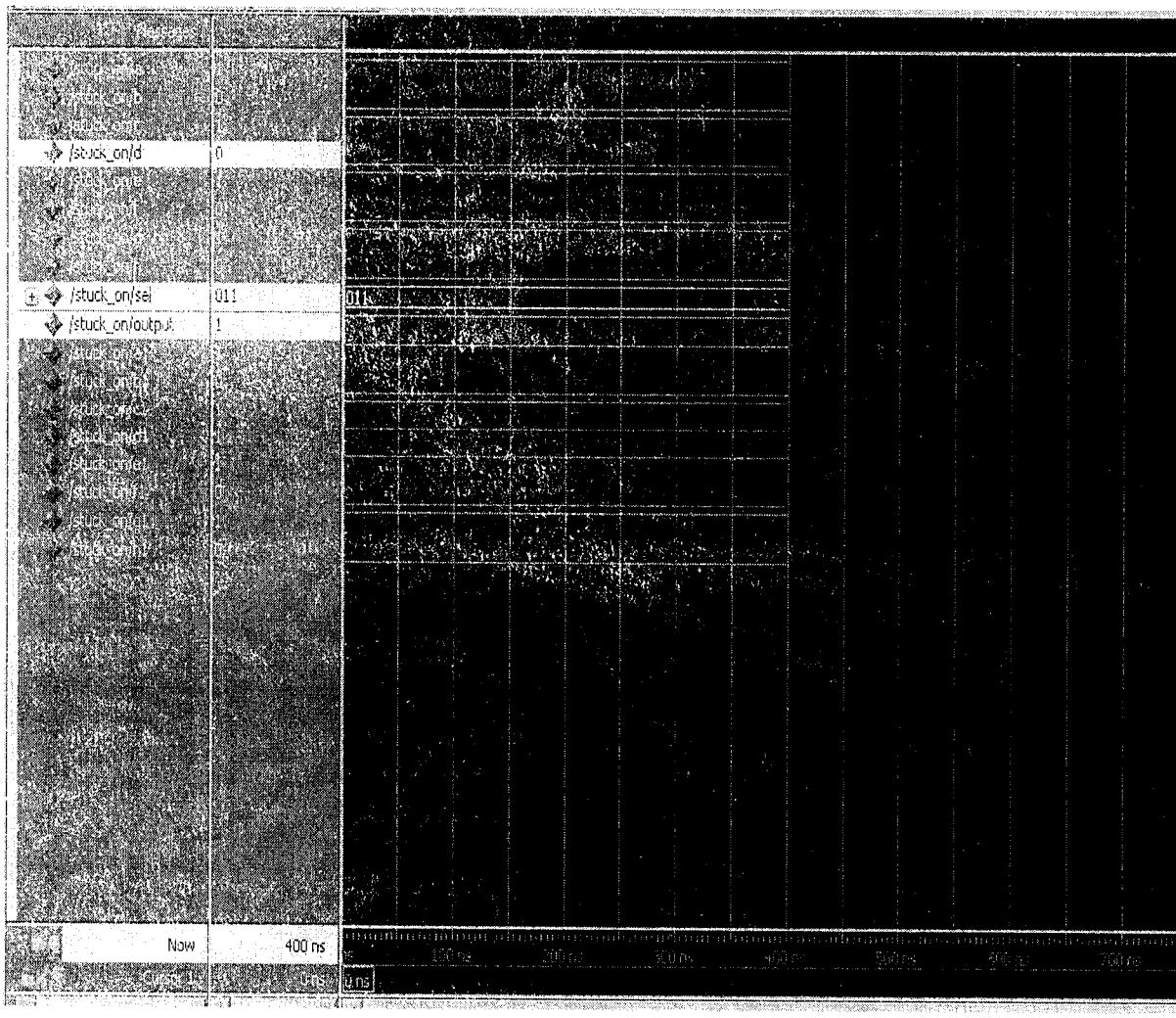

Fig.8. Simulation result of Stuck-on fault in the local interconnects.

7.3 Stuck-at-1 fault on a RAM cell in an LUT.

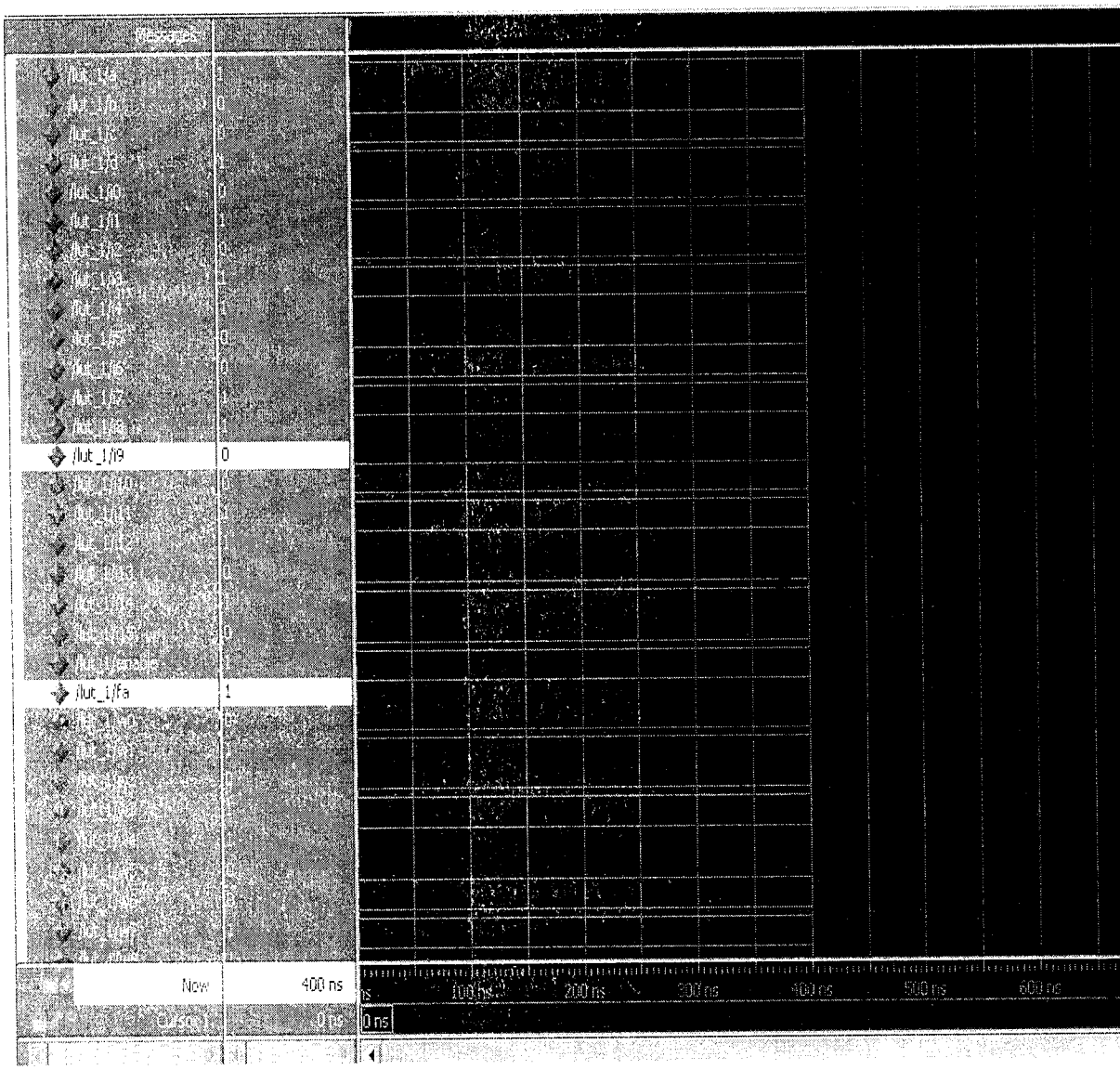

Fig.9. Simulation result of Stuck-at-1 fault on a RAM cell in an LUT.

7.4 Stuck-at-0 fault on a RAM cell in an LUT.

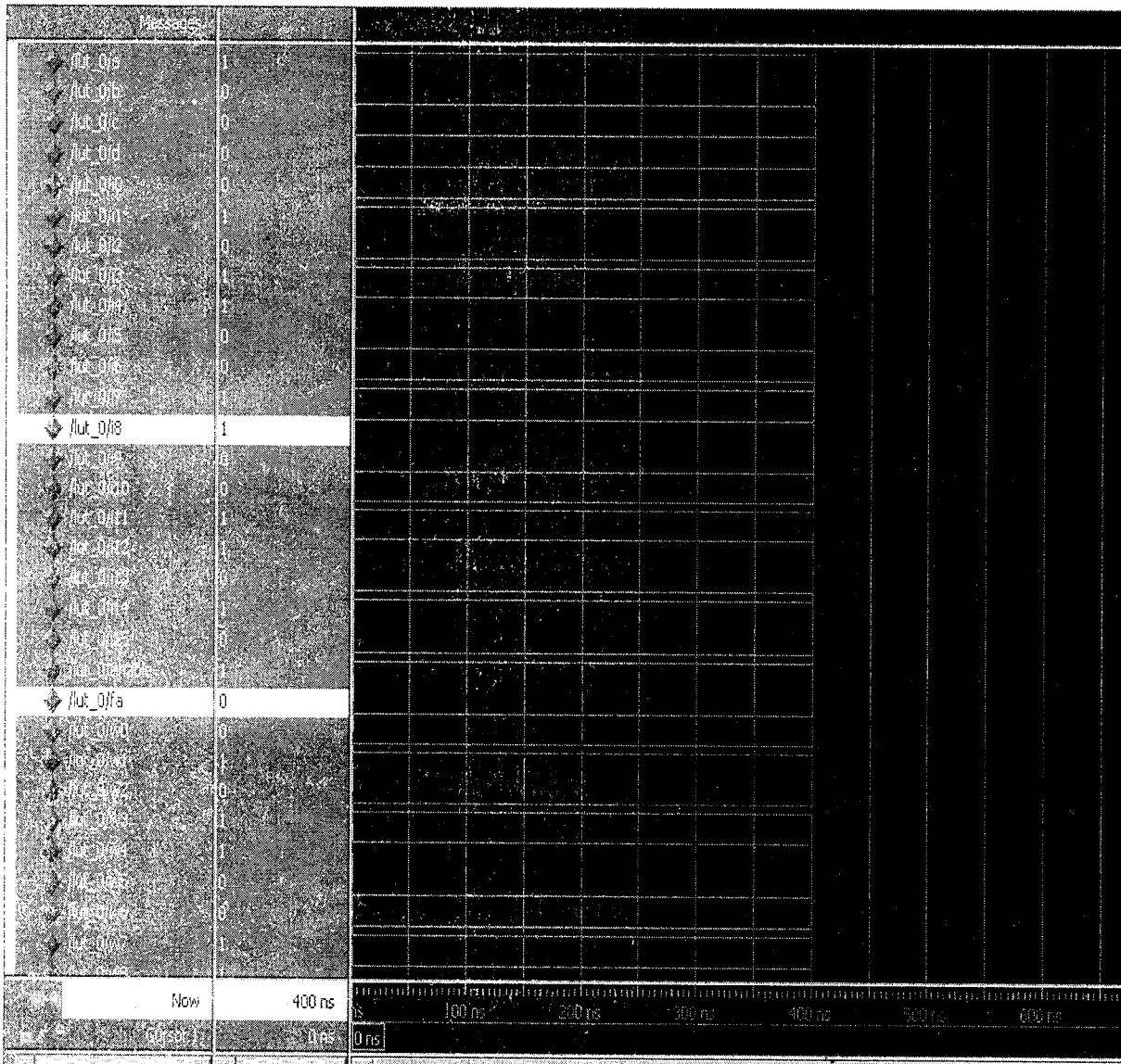

Fig.10. Simulation result of Stuck-at-0 fault on a RAM cell in an LUT.

7.5 Open fault in the global interconnect:

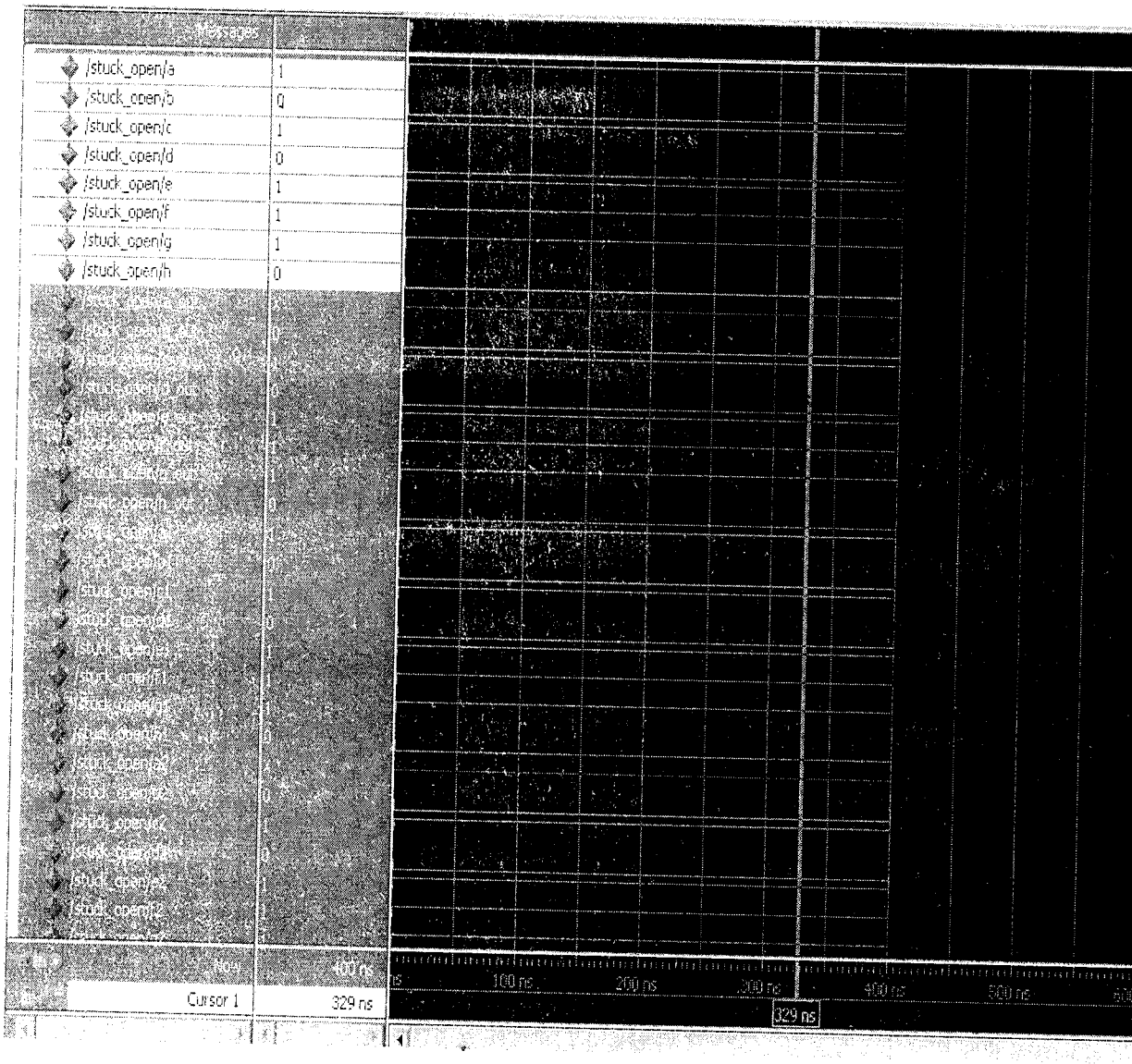

Fig.11. Simulation result of Open fault in the global interconnect.

7.6 Short fault in the global interconnect:

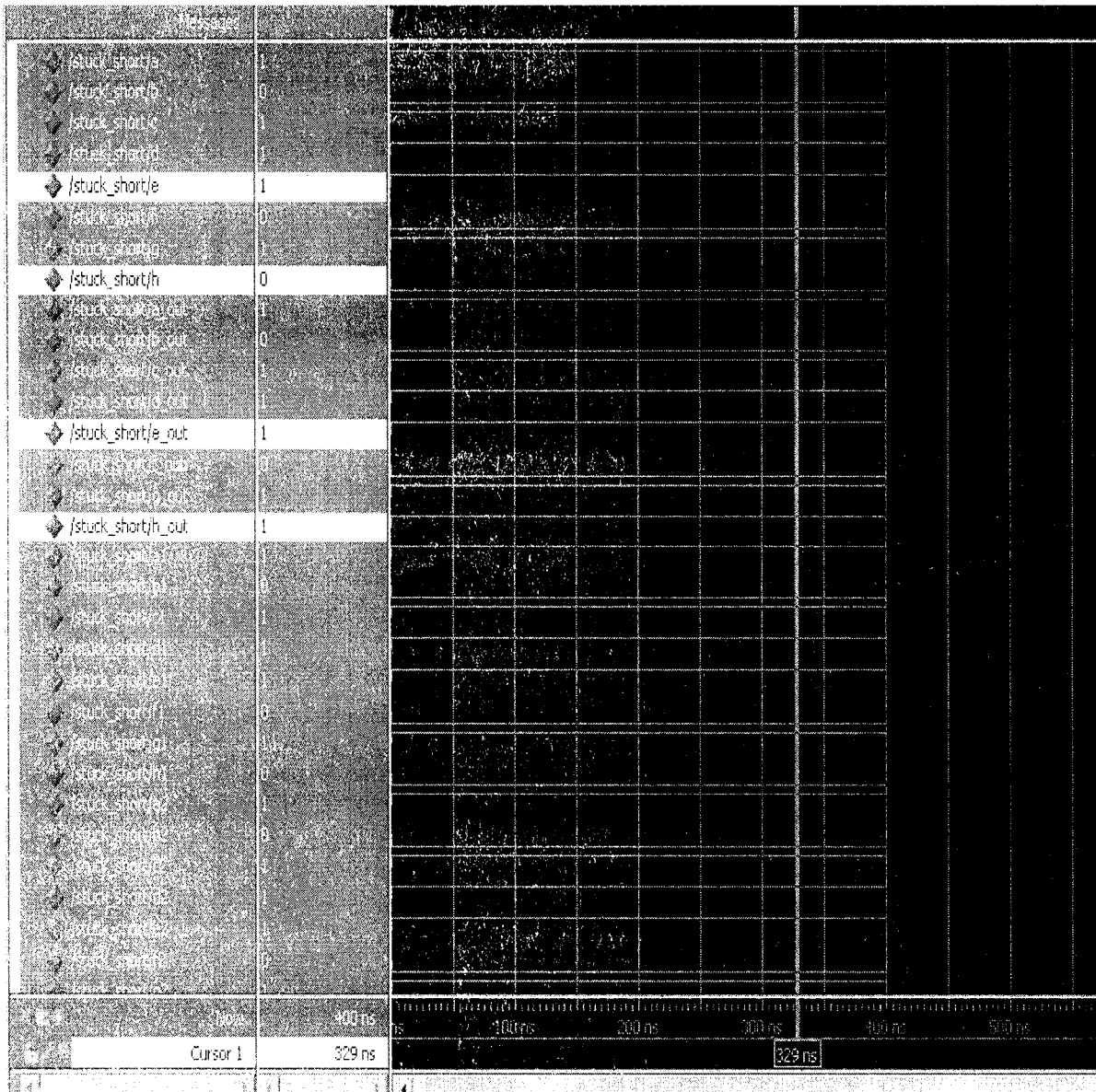

Fig.12.Simulation result of short fault in the global interconnect.

7.7 Proposed fault diagnosis:

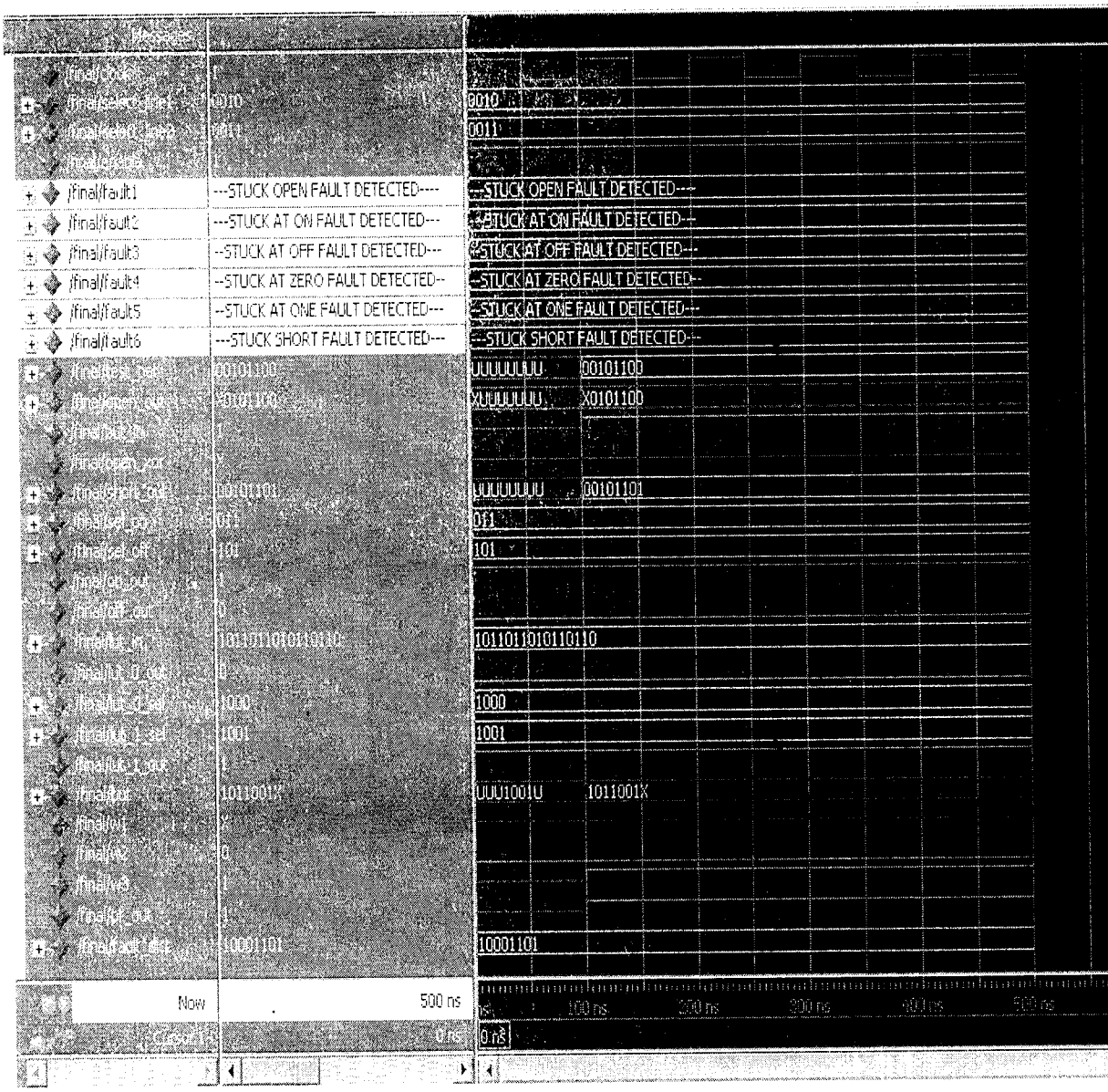

Fig .13. Simulation result of proposed fault diagnosis.

CHAPTER 8

CONCLUSION:

A new BIST approach for fault detection and fault diagnosis of SRAM-based FPGAs has been proposed in this project. The proposed FPGA BIST structure has high fault coverage on the modeled intereonnect and CLB faults, including short/open in wire channels, stuck on/off faults in PSs, and stuck-at-0/1 faults in LUTs. This proposed BIST structure possesses the ability to simultaneously detect and diagnose faults on both interconnect resources and CLBs.

BIBLIOGRAPHY

[1] The Programmable Logic Data Book, Xilinx Inc., San Jose, CA, 2000.

[2] S. Toutounchi and A. Lai, "FPGA test and coverage," in Proc. Int. Test Conf., Baltimore MD, 2002, pp. 599–607.

[3] S. Harrison, G. Noeninckx, P. Horwood, and P. Collins, "Hierarchi- cal boundary-scan: A scan chip-set solution," in Proc. Int. Test Conf., Nov. 2001, pp. 480–486.

[4] W. K. Huang, X. T. Cheng, and F. Lombardi, "On the diagnosis of pro-grammable interconnect systems: Theory and application," in Proc. IEEEVLSI Test Symp., Princeton, NJ, 1996, pp. 204–209.

[5] M. Abramovici, C. Stroud, and M. E. John, "Online BIST and BIST-based diagnosis of FPGA logic blocks," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 12, pp. 1284–1294, Dec. 2004.

[6] M. Y. Niamat, R. Nambiar, and M. M. Jamali, "A BIST scheme for testing the interconnect of SRAM-based FGPAs," in Proc. MWSCAS, Aug. 2002, pp. 480–486.

[7] C. Stroud, J. Nall, M. Lashinsky, and M. Abramovici, "BIST-based diagnosis of FPGA interconnect," in Proc. Int. Test Conf., Oct. 2002, pp. 618–627.

[8] J. Liu and S. Simmons, "BIST-diagnosis of interconnect fault locations in FPGAs," in Proc. IEEE Elect. Comput. Eng., May 2003, pp. 207–210.

[9] J. Smith, T. Xia, and C. Stroud, "An automated BIST architecture for testing and diagnosing FPGA interconnect faults," J. Electron. Test., Theory Appl., vol. 22, no. 3, pp. 239–253, Jun. 2006.

[10] P. K. Lala and A. L. Burress, "Self-checking logic design for FPGA implementation," IEEE Trans. Instrum. Meas., vol. 52, no. 5, pp. 1391–1398, Oct. 2003.