International Journal of Innovative Research in Electronics and Communications (IJIREC)
Volume 2, Issue 5, July 2015, PP 1-7
ISSN 2349-4042 (Print) & ISSN 2349-4050 (Online)
www.arcjournals.org

An Application Specific Reconfigurable Architecture Diagnosis Fault in the LUT of Cluster Based FPGA

Sake Partha Saradhi[1], P.Rajesh[2]

[1]PG Scholar, Electronics and Communication Engineering, CRIT Engineering College, AP, India
[2]Assistant Professor, Electronics and Communication Engineering, CRIT Engineering College, AP, India
parthu.only1@gmail.com

Abstract: *The increased circuit complexity of field programmable gate array (FPGA) poses a major challenge in the testing of FPGAs. One of the test challenges is to detect the delay faults in high-speed circuits. Built-in-self-test (BIST) Technique is an ease solution compared with expensive automatic test equipment. In this work, a BIST structure is proposed to detect the delay faults in the various resources of the FPGA such as multiplier, digital signal processing (DSP) block, look-up tables etc. and interconnects of FPGA. The authors have also proposed a full-diagnosable BISTer structure that improves the testing efficiency of the logic BIST. The proposed BISTer structure can diagnose the faulty configurable logic block (CLB), when all the CLBs in the 2 × 3 BIST are faulty. The proposed scheme has been simulated in Xilinx Vertex FPGA, using ISE tool, Jbits3.0 API and XHWI (Xilinx HardWare Interface) and MATLAB7.0. The result shows significant improvement compared with earlier BIST methods.*

Keywords: *BIST,CLB,FPGA.*

1. Introduction

Field programmable gate array (FPGA) has become widely accepted design approach for low- and medium-range application because of functional flexibility and low development cost. Unique re configurability property of FPGA enables it to achieve function and features that may not be available in application specific integrated circuit (ASIC). The current FPGA runtime testing techniques are realised by reconfiguring FPGA with multiple test phases in one small portion of FPGA hardware, whereas other major portion may run normal applications simultaneously. It is possible to test various components of configurable logic block (CLB) along with interconnects [3–6] because approximately 80% of FPGA area are dedicated to interconnects. The testing technique must be able to detect the latent defects. For decade, built-in-self-test BIST [7–10] techniques have been very popular for testing and diagnosis of various faults in the FPGA. The technique proposed in [9] presents a one- and two-diagnosable BIST er design that makes up roving tester (ROTE). The proposed BISTer avoids time-intensive adaptive diagnosis without compromising fault coverage. The technique achieves highest coverage in one-diagnosable functional – test based BISTer with a three programmable logic block (PLB) test pattern generator (TPG). The method in [11] proposed programmable approaches for scan-based logic BIST. The proposed approach combines the techniques of reseeding and weight random test pattern test. The work in [12] analyses the timing behaviour of look-up tables (LUT) in FPGA (in faulty and fault-free conditions). The author had shown that the LUT delay faults are not independent of the realised functions. The method in [6] has been presented for detecting delay fault in LUT. The test configuration is constructed by chaining LUT in a specific manner and the test patterns are applied in order to test large and small delay faults. A BISTer structure has been proposed in [13] to detect delay fault in LUT of a static random access memory (SRAM)-based FPGA. The test architecture is same as that proposed in [6] but an additional output response analyser (ORA) is used. The technique in [14] has proposed an on-line and off-line BIST-based testing scheme to detect delay faults in FPGA using a roving self-testing-area (STAR) approach. The method in [15] presents BIST architecture for testing of stuck-at-faults, delay faults and bridging faults in FPGA interconnect. The scheme [2, 16] proposes the diagnosis of delay fault for most of the resources of FPGAs. The dynamic delay model of LUT in FPGA has been explained using resistor–capacitor (RC) model [12]. In this model, an input LUT can be represented as n cascaded stages of SRAM cell, where every stage is

one-dimensional array of vertical (2 : 1) multiplexers [6, 12, 13]. The delay fault testability of LUT proposed in [6, 13] suffers from few drawbacks. One of the drawbacks is the addition of delay produced by the faulty flip-flop with LUT chain delay. This will lead to the wrong decision. As this consideration of the faulty value is the critical issue of the testing technique, it makes the detection of delay fault to be difficult. It is also very difficult to predict the time delay of LUT.

2. 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, lock 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.

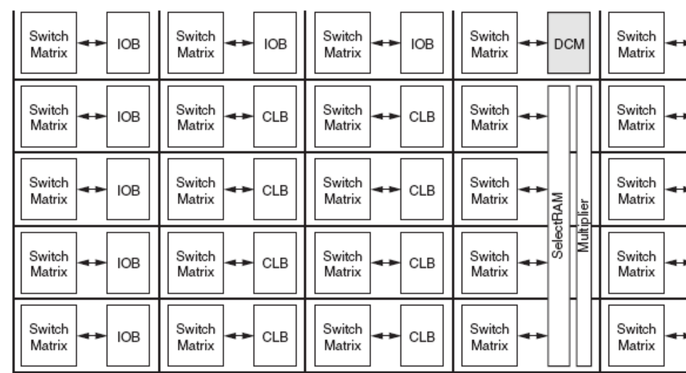


Fig. 1. *Routing Resources*

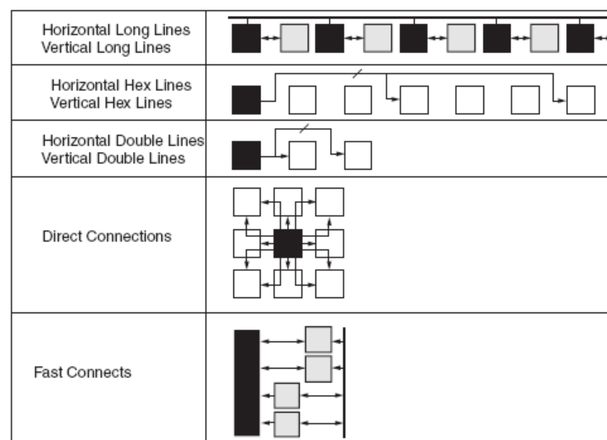


Fig. 2. *Hierarchical Routing Resources*

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.\

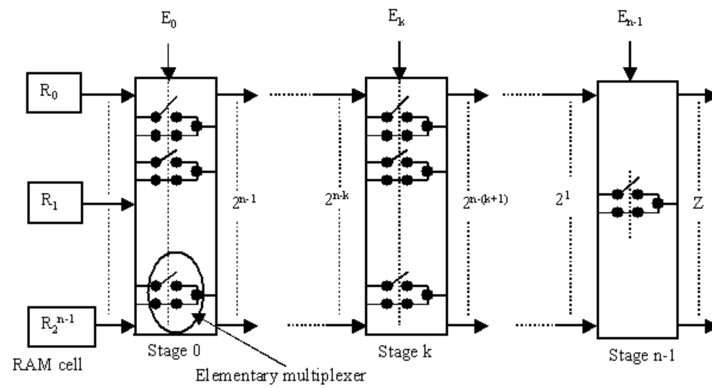


Fig. 3. *n-input LUT*

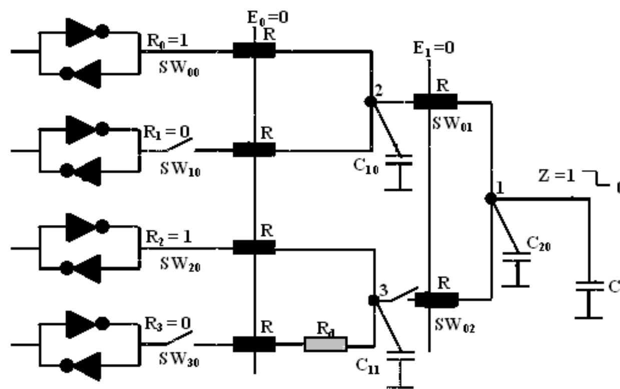


Fig. 4. *2-input LUT with resistive open*

where *E*0, *E*1,… , *En*-1 are LUT input and *R*0, *R*1, ….. *R*2 (*n*-1) are corresponding values of the implemented functions in he 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 Pi to be active the entire switch SWkn should be ON. All the paths are associated with a unique input configuration *Ii* Where,

Ii = (*E*0, *E*1 ….. *En*-1) *Pi*

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 CL is load capacitor.

To describe the switching behavior of the active path we have to consider the initial stage of the capacitor CL and Ckx and the final pattern (value) in response to the input Ii. 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 E0). Let for 2 input LUT as shown in Fig.4 has initial output is '1' with initial input pattern (0,0). The capacitor C20 at node 1 and C10 at node 2 will be set to Vdd. Say if next input to LUT is (1,0) then both the capacitor C20 and C10 will be set to GND. High resistance Rd may get induced in the switching path, because of resistive open in drain or source of the transistor. The time constant of the capacitor CL and Ckx 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 Bky. 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 Bky can be determined by applying input pattern Ii such that it will produce complementary output.

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

INITIAL E0	INITIAL E1	FINAL E0	FINAL E1	Type of Fault in Node 1	2	3
1	0	0	0	X	B/O	X
*1	0	0	1	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	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 a0th input of the next stage and so on. Each LUT was configured with function $f(E0, E1 \ldots\ldots En\text{-}1) = E0$. 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

$f(E0, E1 \ldots\ldots En\text{-}1) = \mathrm{E0}$ and $f(E0,E1 \ldots\ldots En\text{-}1) = 0\ E$.

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.

3. PROPOSED TESTING CONFIGURATION

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 a0 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 a extended part of TPG, hence it will be non-testable. All LUTs will be configured with function $f(E0, E1 \ldots\ldots En\text{-}1) = E0$. 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 flipflop 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

$f(E0, E1 \ldots\ldots En\text{-}1) = E0.$

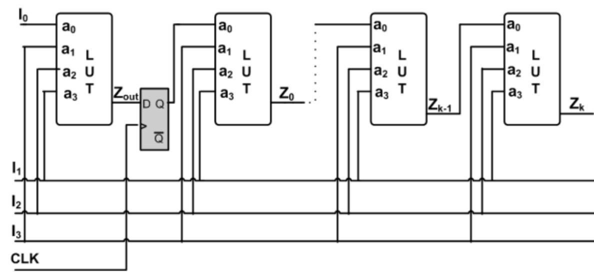


Fig. 6. *Proposed configuration of BUT*

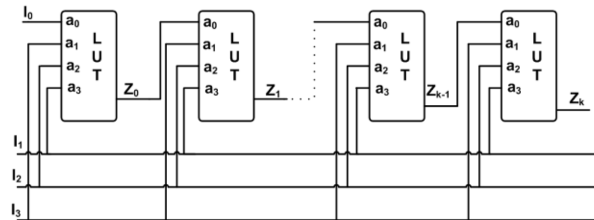


Fig. 7. *Configuration of BUT to detect short delay fault*

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 I shown in Fig.9 where T0, T1 and T2 are output of XOR gate

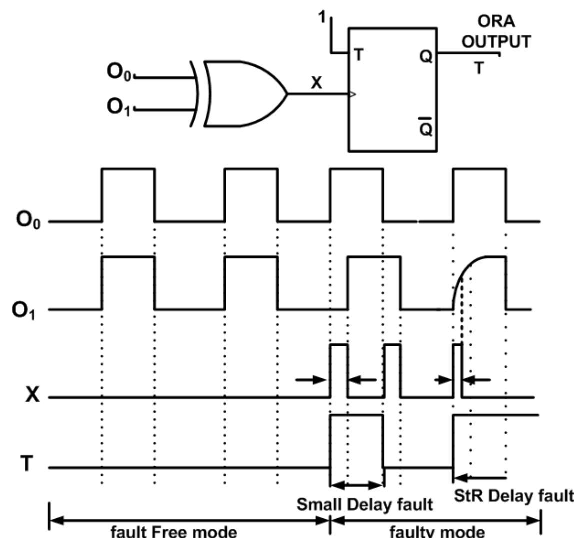
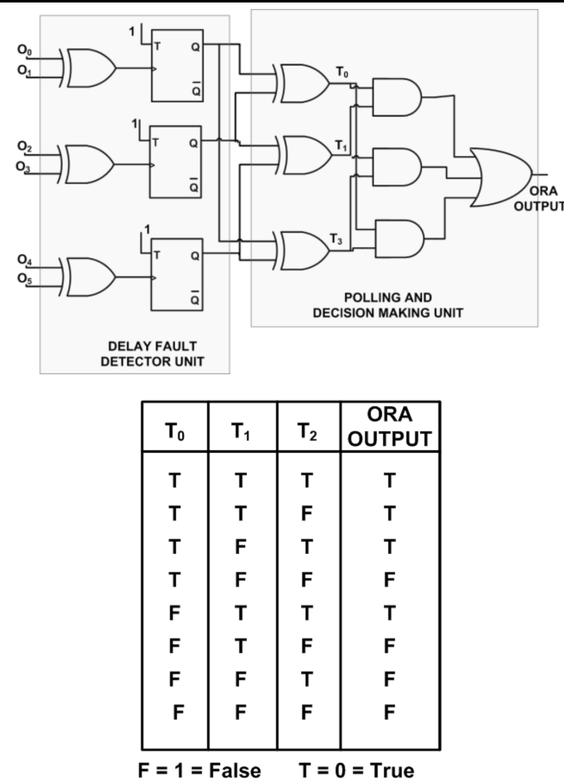


Fig. 8. *ORA scheme and its response to input pattern*

TPG is a FSM which will generate 2*n* test patterns for n input LUT of length n bit ,say *E*0, *E*1 ….. *En*-1. The output *E*0 will only go to the a0 input of first LUT from left (refer to Fig. 6) and rest of the bits *E*1 ….. *En*-1 will go to all input (*a*1…*an*) of LUTs. An additional pulse generator will be required for TPG used in configuration of Fig. 6



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

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

4. Simulation Results

The **DIAGNOSIS FAULT IN THE LUT OF CLUSTER BASED FPGA** implemented in verilog, compiled and simulation using Xilinx ise . The circuit simulated and synthesized. The simulated result for Fault

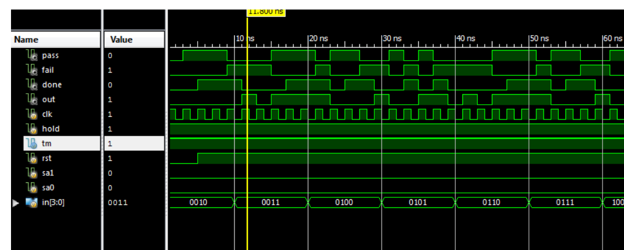


Fig. 10 *Simulation Result.*

5. Conclusion

This work presented a technique for testing delay-fault in various recourses of the cluster based FPGAs. The proposed testing schemes are applicable for both on-line and off-line testing using roving STAR approach. The proposed BIST methods can test all the resources (such as LUT, flip-flop, arithmetic units, multiplier, multiplexer and DSP units) available in a modern FPGA without additional overhead. The proposed new full-diagnosable BISTer structure improves the testing efficiency of the logic BIST. The 2-diagnosable BISTer-2 structure proposed in earlier works, detects the presence of two faulty CLBs in the BIST configuration. The proposed BISTer structure has 100% diagnostic resolution at 100% fault density compared with previously proposed method, which has only 86% diagnostics resolution at 30% fault density. We have achieved significant improvement over previously proposed BIST method. We have emulated the delay faults by using longer chain in case of faulty condition compared with fault-free condition.

References

[1]. 1 Das, N., Roy, P., Rahaman, H.: ‘Build-in-self-test of FPGA for diagnosis of delay fault’. IEEE ASQED, 2011, pp. 4–61

[2]. 2 Das, N., Roy, P., Rahaman, H.: ‘New technique for testing of delay fault in cluster based FPGA’. IEEE MWSCAS, 2011, pp. 1–4

[3]. 3 Harris, I.G., Menon, P.R., Tessier, R.: 'BIST-based delay path testing in FPGA architectures'. IEEE ITC, 2001, pp. 932–938

[4]. 4 Das, N., Ghosh, S., Rahaman, H.: 'Detection of single stuck-at and bridging faults in cluster-based FPGA Architectures'. Proc. VDAT, 2007, pp. 205–212

[5]. 5 Das, N., Roy, P., Rahaman, H.: 'On line testing of single feedback bridging fault in cluster based FPGA by using Asynchronous element'. IEEE IOLTS, 2008

[6]. 6 Girard, P., Heron, O., Pravossoudovitch, S., Renovell, M.: 'Delay fault testing of look-up table in SRAM-based FPGA', J. Electron. Test., Theory Appl., 2005, 21, pp. 43–55

[7]. 7 Abramovic, M., Breuer, M.A., Friedman, A.D.: 'Digital system testing and testable design' (Wiley-Addison, 1994)

[8]. 8 Stroud, C., Wijesuriya, S., Hamilton, C.: 'Built-in self test of FPGA interconnect'. Int. Test Conf., 1998, pp. 404–411

[9]. 9 Dutt, S., Verma, V., Suthar, V.: 'Build-in-self-test of FPGA with provable diagnosabilities and high diagnostic coverage with application to online testing'. IEEE TCAD, 2008, vol. 27, pp. 309–326

[10]. 10 Abramovici, M., Stroud, C.E.: 'BIST based delay fault testing in FPGA', J. Electron. Test., 2003, 19, pp. 549–558