A Fault Propagation Model for FPGA Netlist with Un-reconvergence Paths

Jian Wang, *IEEE member*, Zhe Chen

University of Electronic Science and Technology of China
Chengdu, China
wangjian3630@uestc.edu.cn

Jinzhi Lu

KTH-Royal Institute of Technology
Stockholm, Sweden
jinzhl@kth.se

Abstract—In this paper, we propose an analysis method of fault propagation based on the FPGA netlist with un-reconvergence path. First of all, we analyze the characteristics of common components in the netlist, calculate the fault propagation probability of these components, and then establish a netlist-level fault propagation model based on the interconnection topological structure of the components in the netlist. When a fault happens at input ports of the netlist, we analyze the fault propagation characteristics in the whole netlist, calculate the probability of the fault propagating from input to output ports, and find out the input port which is most likely to cause the fault. Finally, we verify the model presented in this paper by experiments. The experimental results show that the estimated error of fault propagation probability is negligible. It is less than 0.1% compared to the corresponding testing results.

Keywords- FPGA; Netlist; Fault Propagation;

I. INTRODUCTION

FPGA (Field Programmable Gate array) is originated from the development of VLSI and CAD techniques. It has the advantages in configurable and high integration and low power consumption, etc., meeting the needs in most of the electronic design fields. Therefore, FPGA is gradually replacing ASIC and PDSP, deeply affecting our production and widely used in communications, defense, aerospace, medicine and industrial automation, etc [1][2].

However, the reliability of FPGA suffers from challenges. Today, the fixed-type faults frequently appearance in digital circuits, especially in PCBs which are aging and corroded due to time, resulting in open and short circuits and affecting the function of PCB [3]. In addition, some factors, such as the instability of clock and voltage in the PCB board, may lead to intermittent faults of the input signal of FPGA. If the input signal of FPGA circuit has intermittent fault, the fault will greatly threat the performance and security of FPGA system. As mentioned in [4], the occurrence rate of intermittent faults is usually ranged from 50% to 70%, in terms of the total fault in the aviation electronics and computer industries. Therefore, the fault propagation of FPGA attracts attentions.

Figure. 1 shows the 6 faults encountered on the complex PCBs and the related probability occurrence rate [5]. We can find that due to corrosion and aging factors such as the discrete device fault and due to solder joint leakage welding

Figure 1. Different faults on complex PCBs and their probability of occurrence [5]

or corrosion caused by the fault, the proportion of all faults in the highest proportions. In this paper, we present an analysis method to analyze the FPGA fault caused by these two common PCB faults.

The organizational structure of this paper is as follows. In section II, we discuss the current research situations of logic circuit fault propagation at home and abroad. In section III, we analyze the fault propagation probability of each component in FPGA, and propose a fault propagation model for FPGA circuit. Then, we perform extensive experiments to verify our model in section IV. Finally, we summarize our work in Section V.

II. LITERATURE REIEWS

Fault propagation analysis is beneficial for both the early fault detections and the prediction of fault development trend.

A lot of work puts effort on fault propagation, but they mainly focus on the combinatorial logic circuits. For example, in [6], the **Probability Transfer Matrix (PTM)** and **Ideal Transfer Matrix (ITM)** were proposed to evaluate the logic characteristic and the fault propagation in circuits. In [7] and [8], the PTM is used to promote the operation of matrices and to capture the correlation of internal signals. In [9], a probability fault model based on **Bayesian networks** was proposed to estimate the fault probability of the expected output, and the conditional probability table was used to capture all internal signal correlations for each gate. In [10], the author proposed a model to calculate the fault transmission probability in gate-level. The model took the

reconvergence effect into considerations, such that the accuracy is higher than the previous work. However, the scalability of the method is weak. The author in [11] estimated the probability of fault propagation based on signal probability. In this paper, the fault propagation of the logic shielding was evaluated by using the fault propagation rules. In [12], Chen *et al* proposed a new analytical method in which the fault propagation was formalized by using Boolean functions.

These methods have been used to analyze and evaluate the fault propagation of combinational logic circuits. However, when a FPGA netlist touches upon sequential logic circuits, they cannot work. To address this issue, a cube based analysis algorithm was proposed in [13] to estimate the fault propagation probability of the FPGA netlist. In [14], a software error propagation analysis technique is proposed to evaluate the soft error rate of FPGA. However, these two methods cannot tackle the FPGA faults caused by the external board level faults.

In [15], a graph model was introduced to formalize the physical configuration of FPGA, in which the lines between logical unit, slice and IOB are regarded as the nodes of the direction graph, and the PIPs (Programmable interconnect node) are exactly the edges of the graph. In this paper, we present the entry and exit degrees of the direction graph corresponding to the layout and wiring in FPGA, and makes a comparison of the node's degree under different conditions. A fault injection method is accomplished by artificially modifying the relatively large .xdl file, reflecting the short circuit and open circuit faults.

III. FAULT PROPAGATION MODEL

In this paper, we build a fault propagation model, including all components in the FPGA netlist. Besides, un-reconvergence paths in FPGA netlist are all taken into considerations.

A. Preliminary

We define the following assumptions before establish our models :

- Each fault (level reversal) duration is more than or equal to a clock cycle, which is in accordance with the actual FPGA in most conditions.
- We assume that only one input could be failure at each time.
- Since LUT is essentially a RAM, so we ignore the signal propagation delays in the circuit.

We define the notations globally used throughout the paper in TABLE 1, and we describe the fault propagation process in Figure 2.

Firstly, based on the structure of the FPGA netlist circuit, we analyze the probability of the fault propagation of the components in the circuit, which mainly includes LUT, trigger, multiplexer and latch.

TABLE 1. The Symbols and Their Corresponding Meanings

Symbols	The meaning of symbols
P_k	The fault propagation probability at the kth gate in LUT
P_{LUT}	The fault propagation probability of LUT
T	The length of one clock cycle
t_D	The duration of the fault in the trigger
P_{DFF}	The fault propagation probability of trigger
t_{Latch}	The duration of the fault in the latch
P_{Latch}	The fault propagation probability of latch
P_{MUX}	The fault propagation probability of Multiplexer
P_{N_i, M_j}	The fault propagation probability from netlist input N_i to netlist output M_j

Then, we propose a calculation method for the fault propagation probability of the netlist. The method utilizes the algorithm of fault propagation probability of each component integrating with the circuit topology information in the netlist to analyze the fault propagation process from the specified input to the related output, to obtain the probability of fault in the path. Finally, the method obtains the input port which is most likely to cause the output fault when the output fault occurs, thus realizes the accurate quantification and the evaluation to the fault propagation in the netlist circuit.

Figure 2. Fault propagation process

B. Modeling

Now, we establish the fault propagation model of FPGA with the un-reconvergence path. First, we present a description of the fault propagation of the netlist elements, and then the fault propagation of the netlist.

1) Commonly used fault propagation model of netlist component

The Display look-up-table in the FPGA netlist is referred to as one LUT, which is essentially a ram. It writes data to ram beforehand, and whenever a signal is entered, it is equivalent to entering an address to check the table,

finding the content of the address, and outputting the content.

Each LUT has its expression in the netlist, and we can read its expression directly from the netlist. In order to analyze the fault propagation of LUT, we transform the expressions to the equivalent gate-level circuit structure, and then we calculate the probability of the LUT input fault propagating to the LUT output port by the following calculation method, according to the structure of gate-level circuit.

In an LUT corresponding gate-level circuit, we assume that its input n has failed, the fault propagated to its output m , the path from n to m is $l_{n,m}$. So the process of calculating the fault propagation probability P_{LUT} is as follows.

First, for the output signal S_k of the Kth gate in the LUT equivalent circuit, we define the probability value $P_{s_k}(0)$ and $P_{s_k}(1)$ for 0 and 1 probability of the signal S_k respectively. At the same time, we define P_k as the probability that the fault signal propagates from the input of the Kth gate in the path $l_{n,m}$ to the output of the Kth gate.

Then we calculate the value $P_{s_k}(0)$ and the value $P_{s_k}(1)$ of the output signal S_k of each gate from the input of the LUT equivalent gate circuit to the output according to a breadth-first approach. Based on the value $P_{s_k}(0)$ and value $P_{s_k}(1)$, we then use the calculation rules in table 1 to find the value P_k of each gate in the path $l_{n,m}$ from n to m , which is the probability of the fault propagation at each gate in the path $l_{n,m}$. Finally, we multiply all the values P_k in the path $l_{n,m}$, and the result is the probability P_{LUT} that the fault signal propagates from the LUT input n to the output m , as shown in the formula (1).

$$P_{LUT} = \prod_{k \in l_{n,m}} P_k \quad (1)$$

As an initial condition, the $P_{s_k}(0)$ and $P_{s_k}(1)$ values of the input terminals of the circuit are predefined as determined values.

TABLE 2. The Law of Calculation of Different Gates

Gate Type	$P_{out}(0)$	$P_{out}(1)$	$P_{k(in_i)}$
AND	$1 - P_{out}(1)$	$\prod_i P_{in_i}(1)$	$\prod_{i \neq j} P_{in_j}(1)$
OR	$\prod_i P_{in_i}(0)$	$1 - P_{out}(0)$	$\prod_{i \neq j} P_{in_j}(0)$
XOR	$1 - P_{out}(1)$	$P_{in_0}(0) \times P_{in_1}(1) + P_{in_0}(1) \times P_{in_1}(0)$	1

Table 2 shows the law of calculation for P_k of AND gates, OR gates and XOR gates. $P_{out}(0)$ and $P_{out}(1)$ respectively indicate the probability that the output signal of the gate is 0 and 1; $P_{in_i}(0)$ and $P_{in_i}(1)$ respectively denote the probability that the No.i input signal of the gate is 0 and 1; $P_{k(in_i)}$ indicates the probability that the signal at the No.i

input of the gate affects the output of the gate. When the No.i input is a fault signal, $P_{k(in_i)}$ is the P_k at that place.

For the calculation rules of $P_{k(in_i)}$ in Table 1, we give an example. As shown in Figure 3, the figure shows an AND gate:

Figure 3. Schematic of AND Gate

If the input is F1 (F2), the output is F3. When the value of F2 is changed to F2', the output is F3', and the truth table is as follows:

TABLE 3. Truth Table

F1	F2/F2'	F3/F3'
0	0/1	0/0
0	1/0	0/0
1	0/1	0/1
1	1/0	1/0

It can be seen that if the F3 fault caused by the fault of input F2, then the value of F1 at this point must be 1. So, we can get the probability of fault propagation for the AND gate.

$$P_{k(in_i)} = \prod_{i \neq j} P_{in_j}(1) \quad (2)$$

For the OR gate and XOR gate, the same rules can be analyzed by the related truth table. By applying the calculation rules in table 1, we can calculate the values of $P_{s_k}(0)$ and $P_{s_k}(1)$ at any position of the gate circuit from the initial conditions, and then compute the P_k by $P_{s_k}(0)$ and $P_{s_k}(1)$. Finally, according to the formula (1), the fault propagation probability P_{LUT} of the LUT equivalent gate-level circuit is obtained.

Because the D trigger only samples the input signal at the time the clock is rising. We assume that the time length of a clock cycle is T , the input has a level reversal (fault), the length of the rollover is t_D . As we consider $t_D \geq T$ in this article, then t_D must contain a clock rise along, so this level reversal (fault) certainly affects the D trigger output, as shown in Fig. 4.

Figure 4. Trigger Fault Propagation Analysis Diagram

At this point, fault propagation probability of D Trigger is as follow.

$$P_{DFF} = 1 \quad (3)$$

Therefore, based on the assumption given in the previous section, the fault propagation probability of the trigger in the FPGA netlist is 1. And, at the output of D Trigger, the time of bit flip is still more than or equal to a clock cycle, if the input signal of the next trigger is fault, and the probability of fault propagation is still 1.

The latch only changes the output value when the clock is high, and the latch remains on the last state when the clock is low. If there is a level reversal (fault) in the latch input, and the time of Flip is t_{Latch} , the fault duration will be more than a clock period, that is, $t_{Latch} \geq T$. The fault at this time must affect the output of the latch, as shown in Fig. 5.

Figure 5. Latch Fault Propagation Analysis Diagram

Therefore in this case, the fault propagation probability of the latch is $P_{Latch} = 1$. Based on the assumption given in this paper, the fault propagation probability of the latch in the FPGA netlist is 1. And, at the output of the latch, the time of bit flip is still more than or equal to a clock cycle, that is, if the input signal of the rear latch is fault, and the probability of the fault propagation is still 1.

If the multiplexer does not have the control signal line and the configuration bit determines that the circuit uses only one multiplexer input port, at this time multiplexer is equivalent to a buffer, so the fault propagation probability is 100%; if the multiplexer has a control signal line, if the control signal is considered not wrong, whenever a multiplexer's input port fails, this fault must be propagated to its output. So the fault propagation probability of multiplexer is still 1. So there

$$P_{MUX} = 1 \quad (4)$$

2) Fault propagation model of FPGA with un-reconvergence netlist

For the fault propagation of FPGA netlist, we can analyze the whole netlist circuit of FPGA by using the fault propagation analysis model of FPGA above.

First, we define one input port N_i of the netlist circuit, assuming N_i fails, and select an output port M_j of the circuit. In the case of an un-reconvergence path, we can determine a unique path l_{ij} from N_i to M_j . Then, according to the fault propagation model of the netlist elements, we find out the probability of fault propagation of all LUT, triggers and latches on the path l_{ij} . Finally, the fault propagation probability of all components on the path from N_i to M_j is multiplied, and the result P_{N_i, M_j} is the probability that the output port M_j will fail when the fault occurs at the input N_i . As shown in the formula (5):

$$P_{N_i, M_j} = \prod_{l_{ij}} P_{LUT} \times \prod_{l_{ij}} P_{DFF} \times \prod_{l_{ij}} P_{Latch} \times \prod_{l_{ij}} P_{MUX} \quad (5)$$

Then we go through all the input port N_i , find out the fault propagation probability of each input port to M_j respectively, and get the maximum fault propagation probability value P_{N_{max}, M_j} , the corresponding input port is

N_{max} . In turn, in the case of knowing output M_j fault, refers to how much probability each input will cause M_j fault, and we can deduce the most easily cause M_j fault of the input port, which is N_{max} . Finally, we go through all the output M_j and get the input port which is most likely to cause fault.

IV. EXPERIMENTAL RESULT

In order to evaluate the above algorithm, we test it experimentally. Based on a given FPGA netlist logic, we first calculate the fault propagation probability value between each input port and output port by using the above algorithm and get the input port which is most likely to cause the output fault. Then, we verify the result of the algorithm on a hardware platform, using the test method based on the fault injection to inject the fault into each input port, and finally calculate the fault propagation probability of each input port of the measured circuit. Then the two kinds of results are compared and finally the verification of the proposed algorithm is realized.

Experimental results:

We present the experimental conditions and results of two experiments as follows:

Experiment one: We validated a simple circuit structure. The circuit has only one LUT. A LUT expression from the FPGA netlist is $N22 = (N1 \& (N7 \& (N2 \& (N3 \& N6))))$, and the probability of the input signal being 0 and 1 is set to be: N1: 0.5/0.5, N2: 0.5/0.5, N3: 0.5/0.5, N6: 0.6/0.4, N7: 0.3/0.7. Based on the algorithm presented in the paper, the fault propagation probabilities of each input are shown in the second row in table 2.

According to table 2 we can find that the input port that is most likely to cause the fault of the output terminal is N6 when the N22 fault occurs.

Then we carried on the experiment verification on the FPGA. We inject the fault in the original circuit, then under the same test datum, respectively obtains output of the original circuit and the fault injection circuit, and compares two circuit's output result to be consistent or not. Finally,

according to the proportion of the fault number of the output mismatch in the total injection fault number, the possibility of the system fault caused by the fault of the tested circuit can be deduced. In this experiment, we set each input port to fail, the number of total injection faults per input is 100,000, and then we count the number of mismatch faults in the output, dividing the numbers by the total number of faults, and finally calculating the approximate probability of fault propagation for each port, and the result is shown in the third row in table 2.

Table 4. Case I: Comparison Between Analysis Results and Test Results

	N1	N2	N3	N6	N7
Analyzed Results	0.07	0.07	0.07	0.0875	0.05
Testing Results	0.07	0.07	0.07	0.0875	0.05

From table 4 , We can see that the results obtained from this calculation are consistent with the actual results. From the experimental results , we can also deduce the input port N6 is most likely to cause the output port failure.

Experiment two: This experiment verifies the experiment result with trigger, LUT expression is $\sim ((A6\&A1) | (A2\&A4))$, the output signal is connected to a trigger after LUT, and the trigger is attached to the output port. The probability of the input signal being 0 and 1 is preset to: A6:0.5/0.5; A1:0.5/0.5; A2:0.6/0.4; A4:0.3/0.7. During the test, we still let each input port fault, the total injection fault number of each port is 100,000, the fault last for a clock cycle. The results are shown in table 3.

Table 5. Case II: Comparison Between Analysis Results and Test Result

	A6	A1	A2	A4
Analyzed Results	0.36	0.36	0.525	0.3
Testing Results	0.36	0.36	0.524	0.3

From the results in table 5, we can see that the fault propagation probability reported by our model is almost the same as the testing result, except for the input port A2 which has error, being less than 1%. Therefore, the method in the paper can calculate the LUT fault probability of LUT with the RAM structure in the netlist.

V. CONCLUSION

In this paper, we present a fault propagation analysis method based on the FPGA netlist with un-reconvergence path. The fault propagation probability of the netlist is calculated by analyzing the fault propagation process of the netlist. Meanwhile, given an output port with error, we can deduce the input port which is most likely to cause this error based on our method through experiments, we can claim that the results based on our model are in accordance with the experimental results. Therefore, we can conclude the

proposed method is effective for analyzing the fault propagation based on FPGA netlist.

VI. ACKNOWLEDGE

The author would like to thanks the supporting from National Science Foundation of China [61671110].

REFERENCES

- [1] L. Sterpone and M. Violante. "A New Partial Reconfiguration-Based Fault-Injection System to Evaluate SEU Effects in SRAM-Based FPGAs," *IEEE Transactions on Nuclear Science*, 2007, 54(4), pp. 965-970.
- [2] P. J. McNulty, G. E. Farrell and R. C. Wyatt, *et al.* "Upset Phenomena Induced by Energetic Protons and Electrons," *IEEE Transactions on Nuclear Science*, 1980, 27(6), pp. 1516-1522.
- [3] C. C. Bech, K. H. Arya and C. Radke, *et al.* "Do stuck fault models reflect manufacturing defects," In *Proc.Int. Test Conference*, 1982, pp. 35-42.
- [4] Q. Haiyu, S. Ganesan, and M. Pecht. "No-fault-found and intermittent failures in electronic products," *Microelectronics Reliability* 48.5, 2008, pp. 663-674.
- [5] A. Insaf, M. Baig and Z. Nayyar. "Techniques to Identify and Test PCB Faults with Proposed Solution," *Journal of Basic & Applied Sciences* 10, 2014, pp. 532.
- [6] D.T. Franco, M.C. Vasconcelos and L. Naviner, *et al.* "Signal probability for reliability evaluation of logic circuits," *Microelectronics Reliability*, vol. 48, no. 8-9, 2008, pp. 1586-1591.
- [7] S. Krishnaswamy, G. F. Viamontes and I. L. Markov, *et al.* "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 2008, 13(1), pp. 8.
- [8] S. Krishnaswamy, G. Viamontes and I. Markov I, *et al.* "Accurate reliability evaluation and enhancement via probabilistic transfer matrices," In *Proceedings of design, automation and test in Europe*, 2005, pp. 282-287.
- [9] T. Rejimon, K. Lingasubramanian and S. Bhanja. "Probabilistic error modeling for nano-domain logic circuits," *IEEE Tran on Very Large Scale Integration Systems*, 2009, 17(1), pp. 55-65.
- [10] N. Mohyuddin, E. Pakbaznia and M. Pedram. "Probabilistic error propagation in logic circuits using the Boolean difference calculus," *IEEE International conference on computer design*, 2008, pp. 7-13.
- [11] H. Asadi and M.B. Tahoori. "Soft Error Derating Computaion in Sequential Circuits," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, 2006, pp. 497-501.
- [12] L. Chen, M. Ebrahimi and M.B. Tahoori, "CEP: Correlated Error Propagation for Hierarchical Soft Error Analysis," *Journal of Electronic Testing*, vol. 29, no. 2, 2013, pp. 143-158.
- [13] K. Huang, Y. Hu and X. Li. "Reliability-Oriented Placement and Routing Algorithm for SRAM-Based FPGAs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, Feb. 2014, pp. 256-269.
- [14] Y. Kimi, G. Matsukawa and S. Yoshida, *et al.* "An accurate soft error propagation analysis technique considering temporal masking disablement," *2015 IEEE 21st International On-Line Testing Symposium (IOLTS)*, Halkidiki, 2015, pp. 23-25.
- [15] X. Wang, C. Gao and J. Yao *et al.*, "A fault propagation analysis method of SRAM-based FPGA based on directed graph," *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*, Chengdu, 2016, pp. 1-5.