Abstract

This paper presents a new technique for detecting resistive open defects in FPGAs. This technique is based on the reconfigurability feature of FPGAs. Using this technique, the delay of a defective path is increased several times more than the delay of the fault-free path, resulting in a higher resolution in detectability of resistive open defects in FPGAs, even at lower tester speed. Various detailed SPICE simulations are performed to validate this method. Also, a test configuration generation scheme is presented for the entire FPGA.

1. Introduction

In deep sub-micron technology, opens are the major type of defect - it is suspected that 58% of costumers’ returned parts have open defects [11]. An open defect is a break in the connection between two circuit nodes that should be completely connected. A resistive open is a failure mode in which the connection between two circuit nodes is resistive, increasing circuit delay and introducing timing defects [10]. For large values of the defect resistance, they even cause hard failures.

Unlike an ASIC, most of the area of an FPGA is dedicated to the routing resources and wires of different lengths; hence, open defects are more probable. Most of the costumer-returned FPGAs are detected as having open defects.

Detection and diagnosis of resistive open defects in ASICs are discussed in [10]: the effect of speed, supply voltage, temperature, and test patterns are evaluated. Some other research addresses the effect of test conditions for ASICs [1] [5] [9] [11] [12].

There is research being done on the testing of delay faults in FPGAs focusing on logic block and application-dependent testing of FPGAs using self-test schemes [7] [8] [9] [12]. Other research focuses on static timing analysis and 9-valued logic simulation for detection and diagnosis of delay faults [3][4][6].

In this paper, we address the problem of resistive open defects in FPGAs and present new techniques to improve their detectability, which can be exploited in both application-independent (manufacturing) testing of FPGAs, and application-dependent testing as well. Manufacturing test of FPGAs consists of a sufficient number of test configurations and test pattern applications to completely test the resources with respect to the fault model. Due to the regular structure of the FPGA, a portion of the configuration for one block can easily be repeated over the whole FPGA to obtain the complete test configuration. In user dependent testing, the presented technique can be used for only some specific paths, such as critical paths, in order to reduce the total test time. Also, a test configuration generation scheme for resistive opens is presented for manufacturing testing, by just modifying the test configuration already generated for conventional boolean testing.

The rest of this paper is organized as follows. In Sec. 2 a theoretical analysis of resistive opens and the transistor-level circuit model for this defect is presented. The simplified transistor-level model of interconnect configuration in the FPGA is discussed. In Sec. 3, the test technique and simulation results are shown. In Sec. 4, path delay analysis for this technique and the simulation results for different cases are addressed. In Sec. 5, a test configuration generation scheme is presented. Finally, Sec. 6 concludes the paper.

2. Theoretical Analysis

2.1. Modeling Resistive Opens

Figure 1 shows a three-stage inverter chain with a resistive open defect in the middle stage. The delay of the middle stage can be estimated by the following equation [10],

\[ delay \equiv (R_s (V_{DD}) + R_{def}) \times C \]

where \( R_s \) is the transistor turn-on resistance as a function of the supply voltage, \( V_{DD} \), \( C \) is the total capacitance, including wire capacitances and load capacitance of the next stage, seen at this stage. The load capacitance is proportional to the number of fanouts of this stage and input capacitance of each fanout branch.

The delay ratio is used as the detectability metric [10], which is the delay of the defective circuit over the delay of the good circuit. Delay ratio can be viewed as a signal to noise ratio - a larger value means higher resolution in detecting the delay fault.

![Figure 1: A three-stage inverter chain](image)

2.2. Modeling in FPGA

2.2.1 FPGA Architecture

The FPGA model we used is a two dimensional array of configurable logic blocks (CLBs) consisting of logic blocks and switch matrices. There are logic blocks in each CLB connected to the switch matrix through input and output MUXes (IMUX and OMUX). Switch matrices provide the connectivity to different CLBs, while logic blocks contain the combinational and sequential programmable logic. CLBs are connected through horizontal and vertical wiring channels of different lengths, called line segments. Inside each switch matrix are programmable interconnect points (PIPs); a pass transistor controllable by a user-programmable SRAM cell. These PIPs provides selective connectivity between pairs of line segments connected to the switch matrix [15].

The PIP acts as a switch. If the switch is closed, the connection between two lines is established. The line segments can be uni-directional or bi-directional, buffered or unbuffered.
[2] depending on whether they contain a driver or buffer, and so. Therefore these PIPs can be modeled as uni-directional (buffered) or bi-directional wires.

### 2.2.2. Transistor-Level Model in FPGA

![Switch Matrix and Logic Block Diagram](image)

**Figure 2**: A simple routing path in an FPGA

The connection between these two neighboring logic blocks is established through a closed PIP in the switch matrix. Since there are some buffers in the output stage of each logic block (and also for the output stage of LUTs), the resistive defect in this portion of the configuration can be modeled by the circuit shown in Fig. 3. The output stage of the first logic block and also the input load of the second logic block are modeled by inverters. Because the line segments and PIPs in the switch matrix where the defect is located are of main interest, the circuit model of the elements inside the logic blocks are not addressed.

![Circuit Model for FPGA Inverter Chain](image)

**Figure 3**: Circuit model for FPGA inverter chain

As there are buffers in input and output stages of each logic block, the other stages not considered are isolated from this circuit, i.e. the load capacitances of those stage are not seen by this stage of logic.

The closed PIP in the path can be modeled by a pass transistor whose gate is connected to logic value “1”, supply voltage, memory cell, as shown in Fig. 3. The defect in the line segment can be modeled by a resistance whose value is proportional to the size of the defect. The parameters in this model, such as \( R_{tr} \), \( R_{def} \), and \( C \) are similar to those in Fig.1.

In testing for resistive opens in ASICs, test conditions like supply voltage and test temperature are major parameters that affect the test result [1][11][14]; the effect on the delay ratio of changing these parameters is small. Li shows that the effect of these parameters on the delay ratio is less than 10% [10].

In ASICs, the load capacitance is almost constant and is invariant to test conditions when testing for resistive opens. The test conditions affect other parameters such as transistor turn-on resistance, wire resistance and supply voltage. On the other hand, a circuit in an FPGA is reconfigurable, thus the load of the circuit can be adjusted to alter the value of delay, and hence the delay ratio. This programmability is not only in the logic, but also in the interconnects. The abundance of routing resources allows us to use a plurality of routing resources for test purposes without changing the connectivity of the nodes in the original circuit, even under complete logic utilization. Although the same techniques used in ASICs to detect resistive open defects can be exploited in FPGAs, the reconfigurability of FPGAs provides an advantage, yielding much better results (by orders of magnitude).

### 3. The Branch Adding Technique

#### 3.1. Idea and Modeling

Adding more fanouts to a path increases its delay. By adding more fanouts, the load capacitance is increased and hence, total delay increases. The fanouts are added to increase the delay of a defective circuit to be more than that for a defect-free circuit, thereby increasing the delay ratio. This is done by turning on an additional PIP that is connected to a PIP in the original path, as shown in Fig. 4. In this figure, the original path is from the IN block to the OUT block, passing through PIP (A,B). Since there are lots of unused PIPs available in an FPGA, it is not difficult to add additional fanout paths, in this example path (A,C). The corresponding circuit model is shown in Fig. 5.

![FPGA Model with Additional Fanout](image)

**Figure 4**: FPGA model with additional fanout

This additional branch is not isolated from the original path, therefore, the drain-to-bulk capacitance, \( C_{db} \), of this additional transistor can contribute to the total capacitance seen by the original path, increasing its total delay. It should be note that these PIPs in the FPGA have no buffer on their input side.

![Inverter Chain in FPGA with Additional Fanout](image)

**Figure 5**: Inverter chain in FPGA with additional fanout

This technique is scalable, in the sense that more fanout branches can be added to this path, up to the number of unused PIPs connected to any point in the original path. In this example, this number is equal to the number of unused PIPs connected to point A in Fig. 4. The increase in delay of the original path is almost proportional to the number of fanout branches added to the circuit.

#### 3.2. Simulation Results

To verify this technique, SPICE simulations similar to those in the previous section are performed for the circuit model of Fig. 5. The simulations for one and two additional branches are performed to examine the scalability of the technique. Figure 6 shows the delay ratio of the circuit with original configuration, one additional branch, and two additional branches for various
The delay ratio almost doubles by adding a new fanout to the circuit, showing the proportional relationship between delay ratio and additional fanout paths as well as the scalability of this technique.

4. Path Delay Analysis

The effect of path length on the delay ratio is addressed by considering routing paths of different lengths (different numbers of PIPs in the path) for two different cases, buffered PIPs and unbuffered PIPs. In both cases, the worst case analysis is used when considering the effect of path length on delay ratio, meaning only one fanout branch is added to the first PIP in the path.

4.1. Unbuffered PIPs

An unbuffered PIPs is modeled as a pass transistor, therefore, the path of PIPs is modeled by a path of pass transistors. Figure 7 shows the transistor-level circuit of the model. Only a fanout branch consisting of a pass transistor (as a PIP) is added to the first stage of the path.

Simulation results show that the delay ratio decreases as the length of the path is increased, simply because the portion of the delay of one stage to the total path delay decreases. But the delay ratio of the original design, i.e. without using additional fanouts, also decreases. Therefore, to show the impact of path length, the delay ratio of the circuit with fanout is compared with that of original circuit for every value of defect resistance and path length.

Figure 8 shows the improvement ratio gained as a function of path length. This improvement ratio is defined as:

\[
\text{Improvement Ratio} = \frac{\text{Delay Ratio of Circuit with Fanout}}{\text{Delay Ratio of Original Design}}
\]

4.2. Buffered PIPs

In this case, a buffered PIP is modeled by a pass transistor followed by an inverter. Figure 9 shows the transistor-level circuit model for a path of buffered PIPs. Again, the worst case analysis is performed, where only one fanout branch was added only to the first stage.

Simulation results are similar to those for unbuffered PIPs. The results, in terms of improvement ratio, are shown in Fig. 10. Each stage in the path is isolated because they are separated by buffers, yielding a slight reduction in improvement ratio as the path length increases compared to that for unbuffered PIPs. The conclusion is that the technique is still almost insensitive to path length, even for isolated stages.

5. Test Configuration Generation

The test configurations for interconnects consists of a set of wires under test (WUTs) to cover the entire FPGA. Each WUT consists of line segments and PIPs connecting those lines, which extends from the output of a logic block, LBOUT, to input of another logic block, LBOUT. These logic blocks are configured to
implement transparent logic, identity function followed by flip-flop. This configuration of WUTs can be considered as several parallel scan chains covering entire FPGAs, each from an IOB to another IOB. There are $m$ CLBs between $L_D$ and $L_B$. The total number of configurations depends on $m$, greater values of $m$ allows more WUTs passing through the same CLBs in the same direction, reducing the total number of configurations. As presented in [13], three sets of configurations are necessary to cover all the routing resources in the FPGA, each set targeting PIPs in the same direction.

In order to modify the test configurations presented in [13] to test for resistive opens, an extra fanout branch is added to each WUT in the test configuration. Note that at least one fanout must be added to each PIP in the WUT. By applying 0-to-1 or 1-to-0 transition test vectors at appropriate speed, the WUT having resistive open defect does not make the transition and the faulty value will be captured in the flip-flop in $L_B$. The content of the flip-flops can be read out either using readback [15] or by scanning out the WUT contents. Note that the first failing clock cycle corresponds to the faulty $L_B$, which precisely specifies the faulty WUT. Hence, this technique offers a high resolution diagnosis at no extra cost.

If the direction of PIPs used in the WUTs of a given test configuration is $d$, the PIPs used for the additional branches are chosen from another direction $d'$. In this way, the branch-adding technique does not reduce the number of WUTs that can be tested in the same test configuration. As a result, the number of test configurations for resistive opens is exactly the same as the number of test configurations targeting boolean faults in interconnects, as presented in [13]. An example of a test configuration is shown in Fig. 11. In this example, $m = 2$, i.e. each WUT passes through two switch matrices. An extra fanout is added to each PIP in every WUT. This test configuration consists of WUTs in the horizontal direction, i.e. only horizontal PIPs are used in WUTs. The PIPs used for additional fanouts are in diagonal direction, as a result there is no impact on the number of configurations.

![Image of a test configuration for resistive opens](image)

**Figure 11:** A portion of a test configuration for resistive opens

---

6. Summary and Conclusion

In this paper, new techniques to detect resistive open defects are presented, based on the programmability of FPGAs and abundance of unused routing resources. The load capacitance of the path under test is increased by activating additional fanout branches. Different techniques for buffered and unbuffered PIPs that can be combined to increase the effectiveness of this method. These techniques are quite scalable in the sense that the increase in the delay ratio is proportional to the number of added fanout branches. Detailed SPICE simulations are provided to support the technique, showing an increase of several times in delay ratio compared to the results of similar methods applied to ASICs. Simulations for path delay analysis of both buffered and unbuffered PIPs show that improvement gained is almost insensitive to length of the path.

Also, a test configuration generation method is presented for targeting resistive open defects in an entire FPGA for manufacturing test. The number of test configuration is the same as that for conventional Boolean testing, hence all the routing resources in the FPGA can be tested for resistive opens as well as conventional boolean faults without any extra test configurations.

**Acknowledgment**

This work was supported by Xilinx Inc. under Contract No. 2DSA907. The author would like to specially thank Professor Edward J. McCluskey for his contributions to this work and supervision of the project. Also, I would like to thank Dr. James Li for his valuable discussions and feedback, and Erik Chimelar for his comments.

**References**


