INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

IMPROVED ERROR DETECTION APPROACH TO RECONFIGURABLE FPGA DEVICE

SURESH PALAKA [1], Dr. K. SRINIVASA RAO [2]

[1] Research Scholar, Jawaharlal Nehru Technological University, Hyderabad, India
palaka.suresh@gmail.com

[2] Principal & Professor, Department of ECE, TRRCE, Hyderabad
principaltrr@gmail.com

Abstract: *-. An FPGA-based reconfigurable system may contain boards of FPGAs which are reconfigured for different applications and must work without any faults. In order to minimize system downtime and increase availability, a fault location technique with very short detecting latency is desired. In this paper, a fast technique for run-time FPGA fault location that can be used for high-availability reconfigurable systems is proposed. By integrating FPGA fault tolerance and online fault Detection (OFD) techniques, the proposed approach can achieve significant availability improvement by minimizing the number of reconfigurations required for FPGA fault location and recovery. The area overhead of our approach is studied and illustrated using applications implemented in FPGAs.*

1. INTRODUCTION

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing, hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). FPGAs can be used to implement any logical function that an ASIC could perform. One powerful and exciting application of FPGAs is in constructing reconfigurable systems (or custom computing machines). The hardware in such systems can be reconfigured on the fly to adapt to different computing requirements for different applications. Reconfigurable systems offer higher computational density and higher throughput for many applications compared with conventional fixed hardware systems. Reconfigurable computing is an active area of research. An important and challenging issue for reconfigurable systems is reliability. Reconfigurable hardware is inherently less reliable than conventional hardware because of the large amount of additional circuitry needed to support the reconfigurability.

Testing reconfigurable hardware to ensure that it is defect-free is substantially more difficult than testing conventional hardware. There are an exponential number of different ways that the programmable hardware can be configured. There is no way to test that all possible configurations are fault-free. Previous work on FPGA, configuration independent fault location can be found in [1-6]. However, these configuration independent [7-10] techniques generally require significant numbers of configurations for loading special test configurations in the FPGA. This results in a major component of the system downtime for FPGA-based configurable systems. Therefore, there is a need for fast FPGA fault location that reduces the number of configurations for detection and integrates well with the subsequent FPGA fault tolerance scheme for reducing the overall downtime. In this paper, a fast run-time FPGA fault location approach is proposed. This approach is based on the column-based precompiled configuration schemes for FPGA fault tolerance. Instead of designing special test configurations for fault location, a online fault Detection (OFD) scheme [11-14] in the application circuitry to find a precompiled configuration that avoids the fault in the FPGA is used. By partitioning the entire system into several sub-circuits in which

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

local OFD schemes are used, it can localize suspect faulty columns and reduce the number of configuration attempts. The organization of this paper is as follows. In Sec. 2, the model of FPGA used in this paper is described briefly. In Sec. 3, previous work related to this paper is discussed. Section 4 gives the details about the proposed method for fast fault detection and location based on dependant configuration. Section 5 deals with the results of proposed method, conclusions are given in section 6.

2. FPGA MODEL

Figure 1 shows the model of the programmable logic core of the FPGA. In this model, the programmable logic core of the FPGA consists of an array of three basic elements: Configurable Logic Blocks (CLBs), Connection Boxes (CBs), and Switch Boxes (SBs). A CLB is the basic building block in the two-dimensional programmable logic core of an FPGA. It contains several lookup tables (LUTs) used to store user-defined combinational logic functions. It also contains flip-flops, multiplexers, and dedicated circuitry for optimizing the performance of user applications.

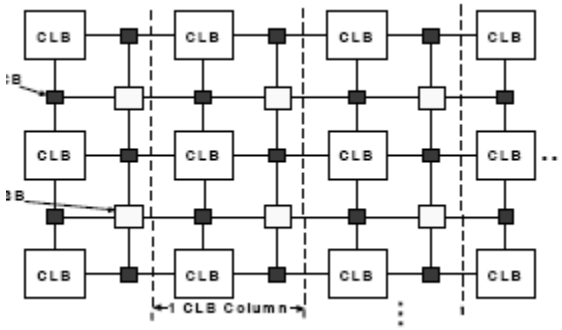

Figure 1: Architecture of the programmable logic core in FPGAs.

CLBs are connected through horizontal and vertical wiring channels between neighboring rows and columns. Signals on the wires are directed among CLBs and wiring channels by two types of routing matrices, CBs and SBs. CBs connect the inputs and outputs of a CLB to the adjacent wiring channels. SBs route horizontal and vertical wiring channels other than the I/Os of CLBs. Both CBs and SBs are matrices of Programmable Interconnect Points (PIPs). The states of the PIPs in these switch matrices are controlled by SRAM cells, which are configured according to the desired functionality.

The next section gives the details about fault detection.

3. FAULT DIAGNOSIS

In this section, FPGA configurations used for fault diagnosis in previous method is discussed.

3.1 Scan Paths

To SCAN a '1' through each net, the CLBs are configured to define pseudo scan paths that run from the primary inputs to the primary outputs. The FPGA is reconfigured such that the place and route information of the original design is used as it is; only the logic performed by the CLBs is changed. The CLBs are essentially used as routers to construct scan paths. Since the inter-connections used are the same as the inter-connections in the original design, every CLB in the configuration has the same set of fanins as the original circuit. The output of each CLB is taken through the internal flip-flop in the CLB so that all segments of the scan path are clocked. Any CLB on a scan path can be controlled and the CLB output is observable since it is uniquely routed to a primary output; its value can be scanned out.

3.2 Fault detection and location

The overall procedure for fault detection is given below.

Step 1. Load the FPGA configuration that SCANs a '1' through the circuit. This configuration (ALLIGN-1) detects all stuck-at-1 and wired-AND bridging faults.

There are two types of FPGA configurations used for fault detection. Scan paths are constructed in both configurations as described in Section 3.1. The logic implemented by the CLBs in the configuration differs. In ALLIGN-1 the reset value of each flip-flop is '0'. The output of a CLB is '1' when there is exactly one '1' in its fanins **and** this '1' is on a net labeled WORKING-REGION.

Step 2. Apply the test case that detects all stuck-at-1 faults on WORKING-REGIONs. This test case (BURST-0) is a sequence of test vectors applied at the primary inputs, where each test vector is the vector of all '0's. The length of the test sequence i.e. the number of times the design is clocked is equal to the length of the longest path in the design. On applying BURST-0 to ALLIGN-1, the fault free response is vectors of all '0's.

Step 3. Apply the test case that detects all stuck-at-1 faults on INERT-REGIONs and all wired-AND bridging faults. This test case (SCAN-1) is a sequence of test vectors applied at the primary inputs of ALLIGN-1 that SCAN a '1' through each of the scan paths. This can be done by applying a test vector with one of the primary inputs set to '1' followed by vectors of all '0's. The '1' should appear at the

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

primary output at the end of the scan path when the number of clocks applied is equal to the length of the scan path. The number of all '0' vectors applied i.e. the number of times the design is clocked after setting one of the primary inputs to '1' is twice the length of the longest path in the design.

Step 4. Load ALLIGN-0. This configuration is used to detect all stuck-at-0 and wired-OR bridging faults.

Step 5. BURST-1. This test case is used to detect all stuck-at-0 faults on WORKING-REGIONs.

Step 6. SCAN-0. This test case is used to detect all stuck-at-0 faults on INERT-REGIONs and all wired-OR bridging faults. ALLIGN-0, BURST-1 and SCAN-0 are the exact duals of ALLIGN-1, BURST-0 and SCAN-1 respectively. If a fault is detected at the above steps, fault location is done by applying the appropriate test vectors.

However these techniques generally require significant numbers of reconfigurations for loading special test configurations in the FPGA. This results in a major component of the system downtime for FPGA-based reconfigurable systems. Therefore, there is a need, discussed in next section, for fast FPGA fault location that reduces the number of reconfigurations for diagnosis and integrates well with the subsequent FPGA fault tolerance scheme for reducing the overall downtime.

4. ENHANOFD FAULT LOCATION SCHEME

4.1 Distributed OFD Checkers

There are different levels of granularity at which OFD can be performed. For reconfigurable systems, since it is possible to repair the system rather than replacing the faulty chip or the faulty board, it is reasonable to implement fine-grained, *distributed OFD* schemes in which each sub-circuit in the system has its own local OFD checker. For example, Fig. 2 shows the architecture of a duplex OFD scheme with checkers in each stage of a pipelined system. Other system-level architectures with such fine-grained OFD have been described in.

In previous approaches, because pipeline registers are used to separate different stages in the system, errors signals from distributed OFD checkers can localize faults within part of the system. If the OFD checker that checks the *i*-th stage outputs does not signal an error in a certain cycle, correctness of computations in the *i*-th stage in this cycle can be assumed. In this way, we can reduce the number of suspect faulty columns in the FPGA and the number of configuration attempts.

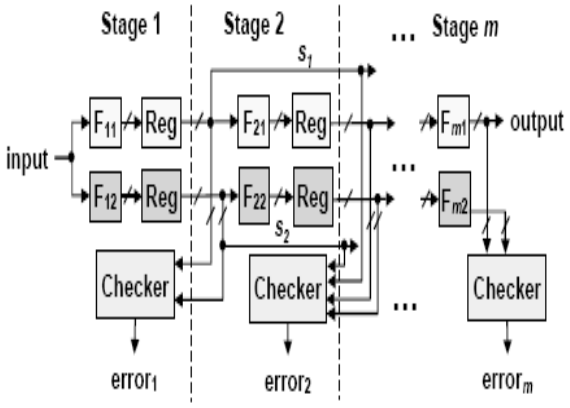

Figure 2: Duplex OFD with distributed checkers in a pipelined system.

Signals that propagate across more than two sub-circuits (e.g., signal *s*1 and *s*2 in Fig. 2) are also checked in intermediate sub-circuits if they are routed by PIPs in such intermediate sub circuits. This is used for localizing the fault that is caused by an open in a long, global signal. The outputs of the distributed checkers in each sub-circuit can be stored in flip-flops, which can be connected in a scan chain. The idea of using a scan chain to scan out the checker outputs is also used in the IBM mainframes to locate the faulty chip or board. When the system signals an error, the contents of the flip-flops storing the checker outputs can be scanned out for further processing by the controller in the other FPGA of the dual-FPGA architecture. The scanned data can be read out through a dedicated Scan Out pin or through the boundary scan port generally available in FPGA chips. If the number of checkers is low, dedicated I/O pins can be used to directly observe the checker outputs.

The controller stores information about the CLB columns that are occupied by each sub circuit in the system. Such information is directly available from CAD tools, such as Xilinx Alliance software, at the design phase through the floor plan of each sub-circuit and checker in the FPGA. After the controller scans out the checker data, it can execute the following simple routine to find the set of suspect faulty columns:

Volume 1, Issue I, Feb 2013
ISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

```
Suspect = ∅
For each checker output do
    If the checker produces an error signal
        Suspect = Suspect ∪ {Columns occupied by the sub-circuit that is
            checked by the checker}
    Endif
Endfor
```

Here the union operation is performed for finding the suspect set because a fault in a global signal that connects multiple sub-circuits causes error reports in multiple checkers. In this case, every sub-circuit that uses or propagates this signal can be suspect. If the suspect set contains only one CLB column, the configuration to be loaded is the one that does not use the suspect column. If the suspect set contains multiple CLB columns, the configurations to be loaded are restricted to those that do not use at least one of the suspect CLB columns. Suppose that there are *m* sub-circuits in the system, each of which has a localized, distributed OFD checker. For this distributed checker scheme, the worst-case number of configuration attempts for avoiding single fault that does not occur on wires across multiple sub-circuits is *max(k1, k2, …, km)*, where *ki* is the number of CLB columns occupied by the *i*-th sub-circuit. The worst-case number of reconfigurations for avoiding single fault that occurs on a wire across multiple sub-circuits is the summation of *ki*'s in every sub-circuit along the path of the wire. Note that since the faulty column suspects are specified by distributed checkers, the proposed technique can also be integrated with the roving STAR approach in order to find the precise fault location. The only difference is that, instead of roving the STAR across the entire FPGA, it only needs to rove the STAR across the suspect columns. The resulting number of reconfigurations required in the roving STAR approach is thus reduOFD when integrated with the distributed OFD checkers in this approach.

5. RESULTS

The developed system is defined using VHDL definition language and the timing results and the implementation on a targeted FPGA device is observed as:

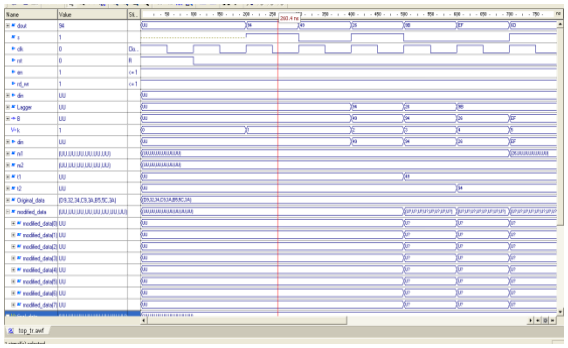

Figure 3: Simulation plot showing the observations made for the developed Lagger algorithm.

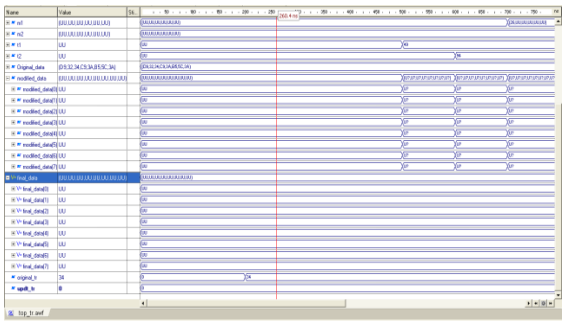

Figure 4: Figure illustrating the original data the modified data regenerated after coding.

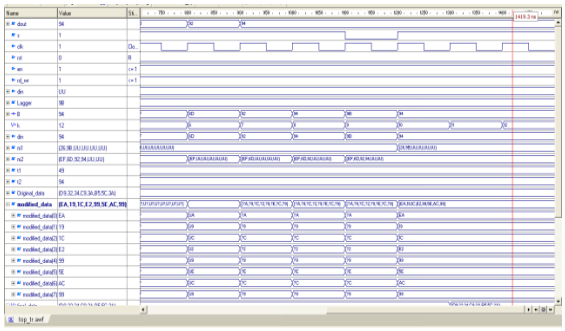

Figure 5: Figure illustrating the bus, Lagger and input data line for the designed encoder and decoder unit.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

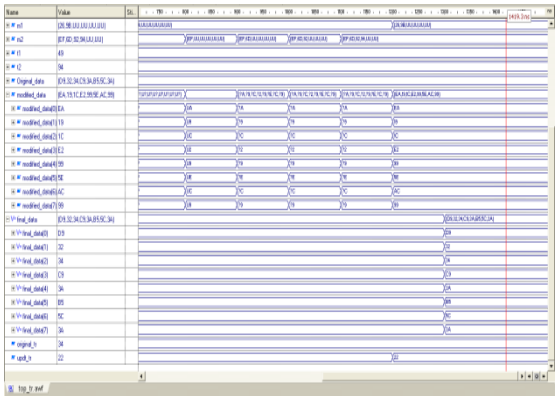

Figure 6: Figure illustrating the content of the memory buffered data for regeneration and the reconstructed data from it.

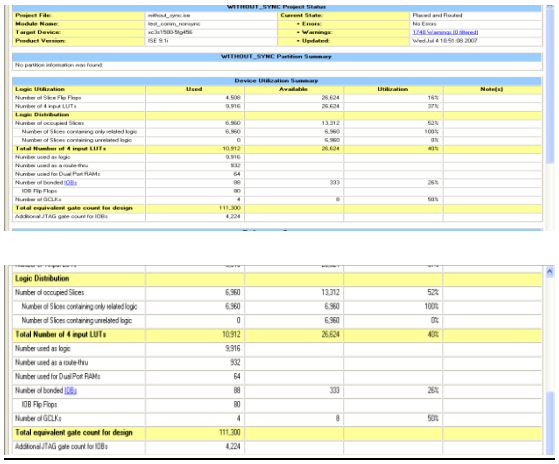

Figure 7: Logical report for the implementation of the suggested approach.

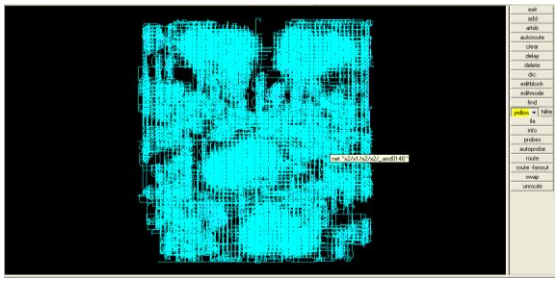

Figure 8: Logical placement of the proposed system.

Default fault location = ck1 (under asynchronous mode communication)
Total number of communicating nodes = 4
Total amount of data generated per node = 3 bytes
Total amount of expected data in processing = 12 bytes
Total transition taken = 5580 (under non synchronous round based comm.)
(Total time = processing time + comm. Time)

6. CONCLUSION

In this paper a new technique is proposed, which integrates OFD schemes with the column-based precompiled configuration approach used for FPGA fault tolerance, for solving run-time FPGA fault location and recovery problem rapidly. Using distributed OFD checkers in each floor planned sub-circuit of a system, faulty column suspects in the FPGA can be obtained, and the number of configuration attempts for fault location and recovery can be reduOFD.

References

[1] Jordan, C., and W.P. Marnane, "Incoming Inspection of FPGAs," Euro. Test Conf., pp. 371-377, 2003.
[2] Lombardi, F., D. Ashen, X. Chen, and W.K. Huang, "Diagnosing Programmable Interconnect Systems for FPGAs," Proc. Int'l Symp. FPGAs, pp. 100-106, 2006.
[3] Stroud, C., E. Lee, and M. Abramovici, "BIST-Based Diagnostics of FPGA Logic Blocks," Proc. Int'l Test Conf., pp. 539-547, 2007.
[4] Stroud, C., S. Wijesuriya, C. Hamilton, and M. Abramovici, "Built-In Self-Test of FPGA Interconnect," Proc. Int'l Test Conf., pp. 404-411, 2008.
[5] Renovell, M., J.M. Portal, J. Figueras, and Y. Zorian, "Testing the Interconnect of RAM-Based FPGAs," IEEE Design & Test of Computers, pp. 45-50, Jan.-Mar. 2008.
[6] Mitra, S., P. P. Shirvani, and E.J. McCluskey, "Fault Location in FPGA-Based Reconfigurable Systems," IEEE Intl. High Level Design Validation and Test Workshop (http://crc.stanford.edu), 2008.
[7] Liu, T., F. Lombardi, and J. Salinas, "Diagnosis of Interconnects and FPICs Using a Structured SCANing-1 Approach," Proc. IEEE VLSI Test Symposium, 2005.
[8] Huang, W.K., X. T. Chen, and F. Lombardi, "On the Diagnosis of Programmable Interconnect Systems: Theory and Applications," Proc. IEEE VLSI Test Symposium, 2006, pp. 204-209.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

[9] Renovell, M., J. Figueras, Y. Zorian, "Test of RAM-Based FPGA: Methodology and Application to the Interconnect," Proc. IEEE VLSI Test Symposium, 2007.
[10] Zhao, L., D. M. H. SCANer, F. Lombardi, "Bridging Fault Detetction in FPGA Interconnects Using IDDQ," Proc. ACM Int. Symp. On FPGAs," 2008.
[11] Mitra, S. and E.J. McCluskey, "Which Concurrent Error Detection Scheme to Choose?," Proc. Int'l Test Conf.
[12] Saxena, N. R., S. Fernandez-Gomez, W.-J. Huang, S. Mitra, S.-Y Yu, and E. J. McCluskey, "Dependable Computing and On-Line Testing in Adaptive and Configurable Systems," IEEE Design and Test, Vol.17, No. 1, pp. 29-41, 2000.
[13] Touba, N. A. and E. J. McCluskey, "Logic Synthesis of Multilevel Circuits with Concurrent Error Detection," IEEE Trans. CAD, Vol. 16, pp. 783-789, July 2007.
[14] Zeng, C., N. R. Saxena and E. J. McCluskey, "Finite State Machine Synthesis with Concurrent Error Detection," Proc. Intl. Test Conf., pp. 672-680, 2009.