

DRAFT: An On-Line Fault Detection Method for Dynamic and Partially Reconfigurable FPGAs

Manuel G. Gericota, Gustavo R. Alves
Department of Electrical Engineering
ISEP
{mgg, galves}@dee.isep.ipp.pt

Miguel L. Silva, José M. Ferreira
Dep. of Computers and Electrical Engineering
FEUP
{mlms, jmf}@fe.up.pt

Abstract

Reconfigurable systems have benefited of the novel partial dynamic reconfiguration features of recent FPGA devices. Enabling the concurrent reconfiguration without disturbing system operation, this technology has raised a new test challenge: to assure a continuously fault free operation, independently of the circuit present after many reconfiguration processes, testing the FPGA without disturbing the whole system operation.

Re-using the IEEE 1149.1 infrastructure, already widely used for In-System Programming, and exploiting the same dynamic and partially reconfigurable features underlying this test challenge, this paper develops a new structural concurrent test approach able to detect faults and introduce fault tolerance features, without disturbing system operation, in the field and throughout its lifetime.

1. Introduction

The advantages of the use of Field Programmable Gate Arrays (FPGAs) were considerably reinforced with the new dynamic and partially reconfigurable SRAM-based FPGAs (e. g. Xilinx's Virtex family), capable of implementing fast run-time partial reconfiguration, enabling the dynamic customization of hardware functions concurrently with system operation.

Unfortunately, current technology tends to make FPGAs less reliable, because smaller submicron scales increase the threat of electromigration, due to higher electronic current density in metal traces. Larger FPGA dies is another factor that increases the probability of failure [1]. Certain defects related to manufacturing

imperfections are not large enough to influence initial testing, but they become exposed after large periods of operation, emerging as either stuck-at faults or transient faults [2].

A higher reliability level can therefore only be achieved through the continuous test of all FPGA blocks and the introduction of fault tolerance features. In this paper we propose a structural concurrent test method, the DRAFT method (Dynamically Rotate And Free for Test), which uses the dynamic and partially reconfigurable features introduced by these devices, and the IEEE 1149.1 Boundary Scan Test (BST) infrastructure [3] for FPGA reconfiguration, vector test application and response capturing, thus presenting a very low test overhead at chip and board level.

2. The DRAFT method

In the vast majority of applications, only a part of the entire FPGA resources is used to implement a given functional specification (the desired functionality). Even when independent hardware blocks dynamically share the same FPGA device (in the case of a dynamically reconfigurable hardware system), 100% usage of its resources is hardly ever achieved, so a few blocks will always be free. Therefore, it is possible to consider a strategy to test temporarily unused blocks, without disturbing system operation, taking advantage of the dynamic and partially reconfigurable features offered by new FPGAs.

Using a dynamic rotation mechanism, each Configurable Logic Block (CLB) currently being used by a given application can have their functionality replicated in one of the CLBs already tested. Both CLBs must remain active with the same state, inputs, outputs, and functionality, for at least one clock cycle, in order to avoid output glitches.

If the current CLB function is purely combinational, a simple read-modify-write configuration procedure is

* This work is supported by the Portuguese Foundation for Science and Technology (FCT), under contract POCTI/33842/ESE/2000

sufficient to accomplish the replication process. However, and in the case of a CLB implementing a sequential function, the internal state information has to be preserved during the replication process. In FPGA devices belonging to the Virtex FPGA family, it is possible to read the value of a register, but not to perform a direct write operation. Therefore, a temporary transfer path should be established between the registers in the two CLBs, to allow state information to be copied between them, and at least one clock pulse applied to both, as described in [4]. This solution guarantees that the whole FPGA can be tested, without disturbing the system operation, provided that at least one unused CLB is available in the current implementation.

3. The dynamic rotation process

The rotation mechanism used in order to free CLBs for test should have a minimum influence in the system operation, as well as a reduced overhead in terms of reconfiguration cost. This cost depends on the number of reconfiguration frames needed to replicate and free each CLB, since a great number of frames would imply a longer test time. The impact of this process in the overall system operation is mainly due to variations on circuit timing, because of the changes in routing. Thus, if the re-routing procedure originates a path delay higher than the previous maximum, the maximum frequency of operation is reduced, leading to an undesirable impact in the system operation.

Three possibilities were considered for establishing a rule for the rotation of the free CLB, among the entire CLB array: random, horizontal and vertical rotation.

The random strategy was rejected for several reasons. If the placement algorithm (in an attempt to reduce path delays) concentrated in the same area the logic needed to implement the components of a given application, it would be unwise to disperse the blocks: firstly, it would generate longer paths (and hence, an increase in path delays); secondly, it would put too much stress in the limited routing resources. Furthermore, a random rotation strategy would imply an unpredictable defect coverage latency, which is not acceptable.

The second strategy, horizontal rotation, is illustrated in figure 2-a. The free CLB would rotate along an horizontal path that would cover all the CLBs in the array. The replication process would take place between neighbouring CLBs, due to scarcity of routing resources and to prevent higher path delays. The same rule applies to the vertical rotation strategy illustrated in figure 2-b, where the free CLB is rotated along a vertical path.

Simulations performed with the last two strategies using Xilinx's Virtex FPGAs, with results presented in [5], have shown that the vertical rotation strategy achieves

lower costs. The size of the reconfiguration files obtained by the application of both strategies to the same circuit implementation was relatively close (approximately 20% higher when using the horizontal strategy), but the influence of each one in the maximum frequency of operation was substantially different, mainly due to a pair of dedicated paths per CLB that propagate carry signals vertically to adjacent CLBs. When the rotation process breaks a dedicated carry path, due to the insertion of the free CLB, the propagation of this carry signal between the nearest adjacent CLBs (above and below) is re-established through generic routing resources, increasing the path delay. When long counters or shift registers are implemented, the horizontal rotation would break all the carry nets, increasing path delays, while the vertical rotation would break only those in the top or bottom of the CLB columns. Considering both costs, the reduction on the maximum frequency and the size of the reconfiguration files, the vertical strategy is preferable to the horizontal one.

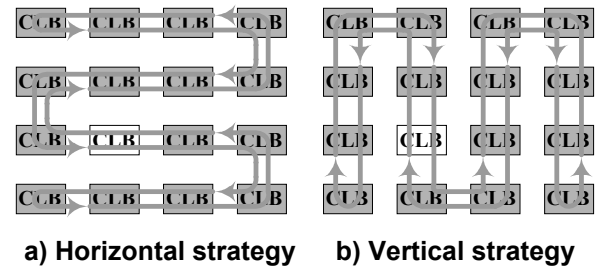


Figure 2. Dynamic rotation of the free CLB

This back and forth dynamic free-CLB rotation across the chip implies a variable test latency. The time to again reach a given CLB alternates between a maximum and a minimum value (according to the rotation direction), depending on the size of the device.

The maximum fault detection latency is given by

$$\tau_{scan_MAX} = ((\#CLB_{rows} \times \#CLB_{columns}) - 2) \times 2 \times (t_{reconf} + t_{test})$$

The minimum fault detection latency is in turn given by

$$\tau_{scan_min} = 2 \times (t_{reconf} + t_{test})$$

where:

t_{reconf} : time needed to complete a CLB replication

t_{test} : time needed to test a free CLB

After a complete rotation, the initial routing is restored.

4. The test session

Each Virtex CLB comprises two exactly equal slices. One of them, representing the test model, is shown in figure 3. In total, the CLB test model has 13 inputs (test

vectors are applied to both slices of each CLB simultaneously) and 12 outputs (6 from each slice).

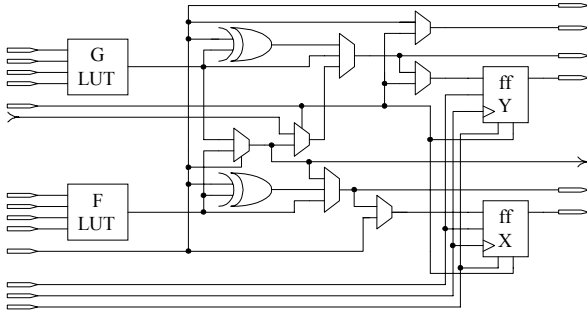


Figure 3. Test model of one Virtex slice structure

The BST infrastructure is used to apply test vectors and to capture test responses, with the outputs of the CLB under test being routed to unused BST register cells (BST register cells associated to output or tri-state lines in IOBs configured as inputs, or BST register cells associated to inputs lines in IOBs configured as tri-state outputs). It is not possible to apply the test vectors through the BST register without affecting the values present at each FPGA input, so an alternative User Test Register must be used (the Virtex family enables the definition of two user registers controlled through the BST infrastructure). This User Test Register comprises 13 cells, corresponding to the required number of CLB test configuration inputs. The seven CLBs occupied by this register and the CLB needed to perform the rotation make up for the 0,7% test overhead, calculated for the CLB resources of a medium size XCV200 Virtex device. Figure 4 illustrates the implementation of our test procedure.

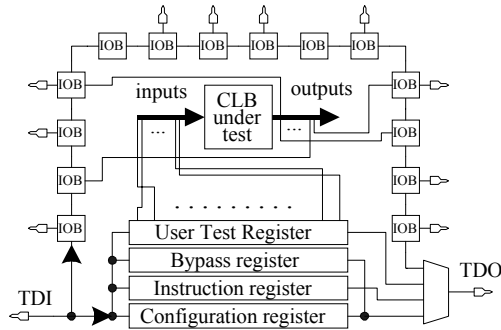


Figure 4. Test of a CLB

Test vector shifting through the User Test Register is very fast, in view of its reduced length. Shifting the response test vector depends on the length of the BST register (device size). Since this user register is part of the CLB array, the CLBs where it is implemented are also tested through the same process. This means that all the

hardware resources used to implement the test procedure are self-tested.

As the result of our analysis of the Virtex CLB test model structure, we concluded that four test phases are enough to exercise all possible configurations in the CLB. As we did not know the implementation structure of the CLBs multiplexers and flip-flops, we considered a hybrid fault model [6]. Table 1 summarises our experimental results.

Table 1. Experimental test results

Test session	
1 st test phase	18 test applications
2 nd test phase	3 test applications
3 rd test phase	2 test applications
4 th test phase	16 test applications

This procedure accounts for 100% fault coverage under the considered fault model.

5. Conclusion

The solution proposed in this paper enables the implementation of a concurrent test method that reuses the standard BST infrastructure and the novel partial dynamic reconfiguration features of recent FPGA devices, in order to improve the reliability of reconfigurable hardware systems, with minimal test overhead and in a way that is completely transparent to the system operation.

Our current work focuses on the extension of the proposed methodology to other FPGA resources and on the development of computational tools to introduce a higher degree of automation in the whole process.

6. References

- [1] Lach, J., Mangione-Smith, W. H., Potkonjak, M., "Low Overhead Fault-Tolerant FPGA Systems", *IEEE Trans. on VLSI Systems*, Vol. 6, N° 2, pp. 212-221, June 1998.
- [2] Shnidman, N. R., Mangione-Smith, H., Potkonjak, M., "On-Line Fault Detection for Bus-Based Field Programmable Gate Arrays", *IEEE Trans. on VLSI Systems*, Vol. 6, N° 4, pp. 656-666, December 1998.
- [3] *IEEE Standard Test Access Port and Boundary Scan Architecture* (IEEE Std 1149.1), IEEE Standards Board, October 1993.
- [4] Abramovici, M., Stroud, M., Wijesuriya, S., Hamilton, C., Verma, V., "On-Line Testing and Diagnosis of FPGAs with Roving STARS", *Proc. of the 5th IEEE International On-Line Testing Workshop*, pp. 2-7, July 1999.
- [5] Gericota, M. G., Alves, G. R., Ferreira, J. M., "Dynamically Rotate And Free for Test: The Path for FPGA Concurrent Test", *2nd IEEE Latin-American Test Workshop Digest of Papers*, pp. 180-185, Feb. 2001.
- [6] Huang, W. K., Meyer, F. J., Chen, X., Lombardi, F., "Testing Configurable LUT-Based FPGAs", *IEEE Trans. on VLSI Systems*, Vol. 6, N° 2, pp. 276-283, June 1998.