

LA-UR-03-6855

Approved for public release;
distribution is unlimited.

<i>Title:</i>	Paper: Testing FPGA Devices in a Space Environment
<i>Author(s):</i>	Prasanna Sundararajan, Bob Patrie, Xilinx Inc., San Jose CA Michael Caffrey, Paul Graham, Los Alamos National Laboratory, Los Alamos NM
<i>Submitted to:</i>	Military and Aerospace Programmable Logic Devices International Conference, Washington DC 9/9-9/11/2003



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Testing FPGA Devices in Space Environment

Prasanna Sundararajan¹, Rob Wells¹, Bob Patrie¹, Michael Caffrey², and Paul Graham²

{Prasanna.Sundararajan, Rob.Wells, Bob.Patrie}@xilinx.com

{mpc, graham}@lanl.gov

¹Xilinx Inc, 2100 Logic Drive, San Jose, CA 95124

²Los Alamos National Laboratory, Los Alamos, NM

Abstract

SRAM-based FPGAs are a relatively new technology for space systems and harsh environments. Assurances that the devices are working properly in deployed systems are important. This paper describes coverage-optimized test patterns and strategies to test FPGA devices deployed in space under different scenarios. This paper will demonstrate that the number of test patterns required to test a majority of the FPGA logic resources is relatively small.

I. INTRODUCTION

SRAM-based FPGAs offer reconfigurability, time-to-market, cost, and performance advantages to space-based systems. The recent introduction of these devices to the space market means that the end-of-life behavior is not well characterized. While no radiation testing has exposed hard failures within the device, it may be that they develop at the end of life or through some unforeseen condition on-orbit. This paper discusses how a system incorporating reconfiguration-based built-in-self-test can diagnose hard faults in the FPGA silicon die. These tests can help a deployed system degrade more gracefully. Periodic testing and reporting allows an application to be remapped around a fault and re-deployed using reconfiguration. This can extend system life and reduce costs.

One experimental system, the Los Alamos National Laboratory Cibola Flight Experiment (CFE) [1], will demonstrate the utility of Xilinx Virtex FPGAs in a RF remote sensing LEO signal processor. CFE uses 9 Virtex XQVR1000 devices to process 2.4 Gbps continuously; one experiment objective is to reconfigure the mission algorithms to demonstrate that reconfigurability magnifies the capabilities of the satellite. Figure 1 shows the Virtex-based reconfigurable system on CFE. One feature of CFE is that it uses the reconfigurability of the Virtex to perform in-situ system level testing for failures such as opens or shorts in connectivity between modules or devices. The Virtex configuration bitstreams can be uploaded from the ground station as well as readback from devices and analyzed in place or downlinked.

We wish to achieve the most coverage and isolation possible of hard silicon faults with a few configurations. Each configuration upload will require approximately one pass over the ground station. Many configurations can be stored on-board and swapped into the FPGA as needed, but this storage space must be shared with mission algorithms. The flexibility of the CFE system makes it an excellent demonstration platform for silicon die tests. The silicon testing proposed in this paper uses a two-pronged approach: first, a variety of test configurations identify the presence of a fault (fault detection), then other configuration(s) are used to perform fault isolation. Once the fault is isolated, applications may be remapped and avoid the fault. In the sections that follow, coverage-optimized patterns developed to test the silicon on-board are described. The envisioned procedure to detect and isolate faults is detailed along with the estimates of coverage and size of the test patterns.

II. FPGA Test Techniques

One primary goal in any test project is to maximize test coverage in each pattern and reduce the overall test time. In a manufacturing environment, shorter test times provide cost benefits as well as enable the company to meet the production deadlines. In the case of in-situ testing on-board CFE, shorter test time means shorter system downtime. For CFE there is an added advantage in maximizing the coverage. Because the satellite is in a LEO orbit, only a few 15-minute passes per day are available for uplinking new configurations, so coverage efficiency is important.

Papers [2], [3], [4] describe techniques to test various FPGA resources. These techniques cannot be directly applied to in-situ testing as they rely on the input and output pins to stimulate and gather test results. Built-In Self-Test (BIST)[5] [6] solves this problem by including on-chip stimulus and result accumulator circuitry.

The technique developed to test the logic resources of the FPGAs on CFE is a novel BIST-based technique. The wire resources are tested using the technique proposed in [7]. Both the logic and wire test patterns exploit Virtex architectural features to optimize test coverage.

III. LOGIC TESTING

The Xilinx Reconfigurable Self-Test (XRST) pattern provides three test patterns, two to test the Configurable Logic Block (CLB) resources and one to test the Block RAM (BRAM) resources. The FPGA device-level structure of XRST is shown in Figure 2. Figure 3 shows the FPGA floorplans of the test structures.

A. CLB Resource Testing

The slice logic tests are constructed of a number of identical 34-bit linear feedback shift registers (LFSRs) each constructed from a 32x1 LUT-RAM and two flip-flops (FFs). Figure 4 illustrates the basic building block of XRST.

The LFSRs are designed to shift on the rising edge of the test clock. The address, chip select and write-enable signals needed to sequence the 34-bit LFSRs are generated by a companion 6-bit LFSR counter, which shifts on the falling edge of the test clock. The output of one of the 34-bit LFSRs is fed into a CRC-32 generator. Comparisons of each LFSR-34 are also made with each of its neighbors on each rising edge of the test clock. Mismatches between neighboring LFSR outputs ripple through the error detection logic, eventually arriving at the CRC-32 generator where the mismatch is latched. The latched mismatch signal causes an inversion of the data input to the CRC-32 generator, thus forcing a faulty signature. To test the error detection logic, the test sequence is executed a second time with the feedback data to the first LFSR-34 being inverted. This inversion will result in a mismatch with its neighbor that will ripple through the error logic eventually forcing an inversion of the input data to the CRC-32. At the end of the second pass through the test sequence, the CRC-32 generator is disabled and its contents are externally compared with the expected CRC value to determine the test result.

A second slice logic test pattern swaps the resources used to implement the LFSR-34s and the corresponding 6-bit LFSR counters in the first pattern. This enables 100% coverage of the following CLB resources:

- LUT RAM and flip flops (FF)
- X, Y, XQ, and YQ outputs
- F1-F4, G1-G4 BX, BY, CE, SR, and CLK inputs.

Coverage of additional resources like carry chains, tri-state buffers, input/output blocks, etc. requires extra test patterns.

B. BRAM Logic Testing

In the BRAM test, each Block RAM is configured as a 256x16 RAM with one write-only and one read-only port. Each BRAM is initialized such that each address location contains a copy of its own address in both its upper and lower bytes. During execution of the test, 8-bit parallel input CRC-32 generators monitor the activity on the upper and lower bytes of one of the BRAMs' outputs. Combinatorial logic performs a bit-wise comparison of all of the BRAM outputs producing and latching separate error signals if any mismatch is detected between corresponding bits of any BRAM. An additional error signal is latched if a mismatch is simultaneously detected on all output bits. The ANY-error signal inverts the inputs to one of the CRC-32 generators, while the ALL-error signal inverts the inputs to the second CRC-32 generator. Since the Any-error condition is a subset of the All-error condition, the All-error condition actually inverts the inputs to both CRC-32 generators. The read and write port addresses as well as the write port data and read port expected data are generated by an 8-bit LFSR counter, with the write port address being delayed by two clock cycles from the read port address.

The test sequence is as follows:

1. All 256 RAM locations are read to confirm their initial values.
2. Each memory address is then written so that both bytes of each location contain the complement of its address. During the write cycle the read addresses continue to cycle, but since the read port is disabled, the outputs remain frozen at the last read value.
3. The write port is disabled and all 256-memory locations are read and verified.
4. The write port is enabled, the read port is disabled, and all 256 locations are written back to their initial values.
5. The write port is disabled, the read port is enabled, and all memory locations are read and verified.
6. Repeating steps 1 through 5 then tests the error detection logic. In this second pass, the write data to one BRAM is inverted forcing a mismatch between all RAM output bits on a subsequent read. This event will be reflected in the eventual CRC-32 calculations.
7. The CRC-32 generators are disabled and their values are externally compared to the expected signature value.

These XRST test patterns can also be re-orchestrated to zoom into a specific faulty location. More on fault detection and isolation is discussed in section IV.

III. WIRE TESTING

A study was conducted to estimate the configurations required to test all the single-length wires of the Virtex architecture. There are total of 96 single wires available in a CLB with 24 each direction. Twenty of the 24 wires, with a total of 80 in a CLB, are associated with an output multiplexer. The wire-test was designed to test these 80 wires using partial reconfiguration and readback. One test session involves one configuration and two readbacks. Four single wires per CLB can be tested in a test session. Since, in one session, the test is parallelised to test all the CLBs; a total of 20 configurations and 40 readbacks are required to test these 80 wires for all of the CLBs.

A variation of the technique proposed in [7] was implemented. Figure 5 provides the details of the wire test. The sequence to test the single-length wires is:

1. Configure initial test data: column 0 as a buffer and other columns as inverters with all FFs initialized to zero. Chain the CLBs together with the wire under test.
2. Step the clock once.
3. Send a readback command.
4. Readback the entire CLB data and check for stuck-at-one faults.
5. Step the clock once.
6. Send a readback command.
7. Readback the entire CLB data and check for stuck-at-zero faults.
8. Partially reconfigure the device with a new wire-under-test and iterate from step 2.

The four single wires without an output multiplexer need separate test patterns to test them.

While the technique described above can be used to test all of the FPGA routing resources, high fault coverage of the interconnect used within a particular user design can be achieved by deriving two test designs from the original design. Both test designs are instrumented to preserve all of the original design's interconnect. One replaces the contents of each LUT and BRAM element so that they perform a logical AND function of its inputs, while the second replaces the LUT and BRAM contents so that they perform a logical OR function of its inputs. By clocking ones from each of the primary inputs to the primary outputs of the first design, the interconnect network is verified to be free of stuck-at-zero faults. Similarly, clocking zeroes from the primary inputs to the primary outputs of the second design, the interconnect network is verified to be free of stuck-at-one faults. The study that is related to application-specific wire testing is a subject for future work.

IV. FAULT DETECTION AND ISOLATION

In order to facilitate a graceful degradation of FPGAs on-board it is required to remap mission algorithms around faults. This means faults need to be isolated upon detection.

The XRST patterns that are used to test the logic resources of the FPGA are designed to generate a 32-bit CRC test signature value that can be used to determine a "pass/fail" at a device level. But these XRST patterns can be re-orchestrated to locate and isolate the faults. Rather than reading back the 32-bit CRC signature at the end of a test run, if the output of the LFSRs are readback after every clock cycle the location of the fault can be identified at a CLB granularity level. The present implementation of the wire test enables fault isolation and detection as the test results are readback after every clock step.

In order to reduce the system downtime for testing, it is initially planned to run the XRST test patterns and determine a "pass/fail". If a test fails, the re-orchestrated XRST patterns would be uploaded and executed to zoom into the fault location.

V. TEST CONFIGURATION COMPRESSION

A preliminary study was conducted to compress the test configuration bitstreams that need to be transported over the communication link between the ground station and the satellite. A size reduction of as much as a factor of 99 has been observed. This dramatically reduces the time required to transport the bitstreams. The data was compressed using the GZIP utilities provided by Java development kit.

Table 1: Compression Results for Wire Test Pattern

Wire Test	Uncompressed (bytes)	Compressed (bytes)	Compression Factor
Full Configuration	766012	7734	99.04
Partial Configuration	494036	5809	85.05

CLB Readback	745524	7660	97.33
--------------	--------	------	-------

VI. CONCLUSIONS AND FUTURE WORK

This paper has shown two example test approaches that detect hard faults in a Virtex FPGA. Faults in CLB, BRAM and singles routing resources are covered. These fault detection tests have been demonstrated on the CFE engineering hardware to prove that the system has the ability to perform in-situ tests using reconfiguration. Feedback from periodic testing should facilitate a graceful degradation of the CFE satellite as it nears its end of life. If hard-faults are detected and can be mitigated with new logic designs that avoid the failed resource, we can extend the life of the spacecraft.

The XRST test patterns would be re-orchestrated such that the result of the test can be readback after each clock step. In order to maximize the test coverage of the wires, a design-specific routing test would be explored as a future work.

REFERENCES

1. M. Caffrey, et al, "A Space-Based Reconfigurable Radio," In *Proceedings of the 2002 Military and Aerospace Applications of Programmable Logic Devices*, Sep 2002, Laurel MD.
2. M. Renovell, et al, "A specific test methodology for symmetric SRAM-based FPGAs. In *Proceedings of tenth International Workshop on Field-Programmable Logic*, FPL 2000, pages 300-311, Springer-Verlag, Berlin. Aug 2000. Lecture Notes in Computer Science 1896.
3. M. Renovell, et al, "RAM-based FPGAs: A test approach for the configurable logic", In *Proceedings of the 1998 Design Automation and Test in Europe (DATE '98)*, pages 82{88, Los Alamitos, CA, February 1998. IEEE Computer Society Press.
4. M. Renovell, et al, "Testing the interconnect of RAM-based FPGAs", In *IEEE Design and Test of Computers*, pages45-50, Jan-Mar 1998.
5. Charles E. Stroud, Eric Lee, Miron Abramovici, "BIST-Based Diagnostics of FPGA Logic Blocks", In *Proceedings of IEEE International Test Conference*, pages539-547, 1997
6. Charles Stroud, Sajitha Wijesuriya, Carter Hamilton, and Miron Abramovici. Build-in self test of FPGA interconnect. In *Proceedings of IEEE International Test Conference*, pages404-411, 1998.
7. Prasanna Sundararajan, Scott McMillan, and Steven Guccione, "Testing FPGA Devices Using JBits", In *Proceedings of the 2002 Military and Aerospace Applications of Programmable Logic Devices*, Sep 2001, Laurel MD.

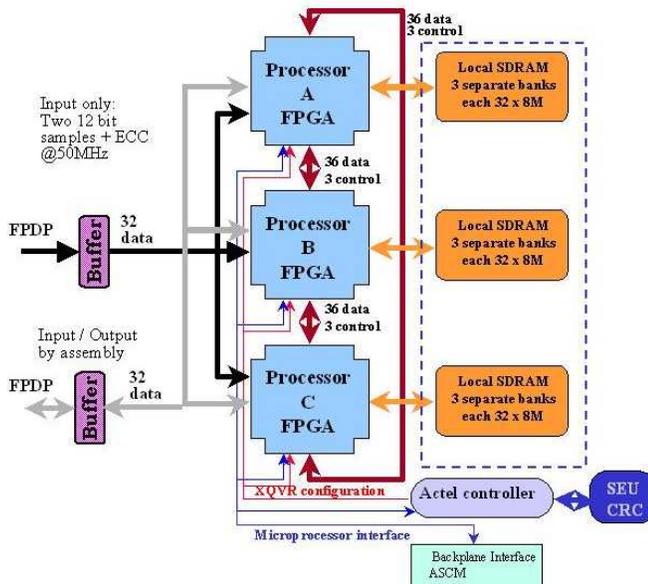


Figure 1. Reconfigurable Virtex FPGA System on Cibola

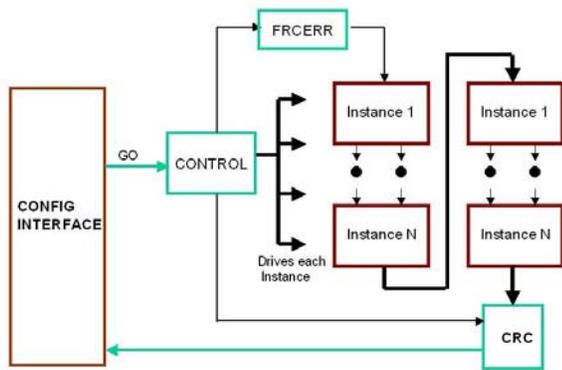


Figure 2. FPGA Device Level Structure of XRST Test

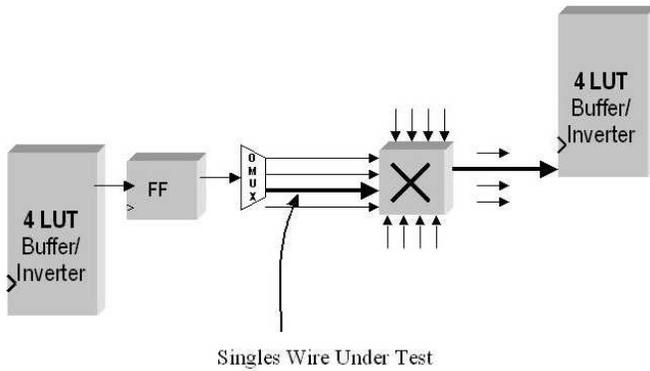


Figure 5. Singles Wire Test

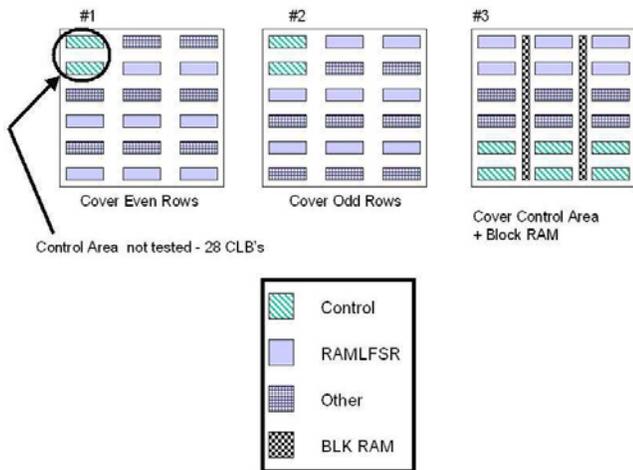


Figure 3. FPGA Floorplans of Test Structures

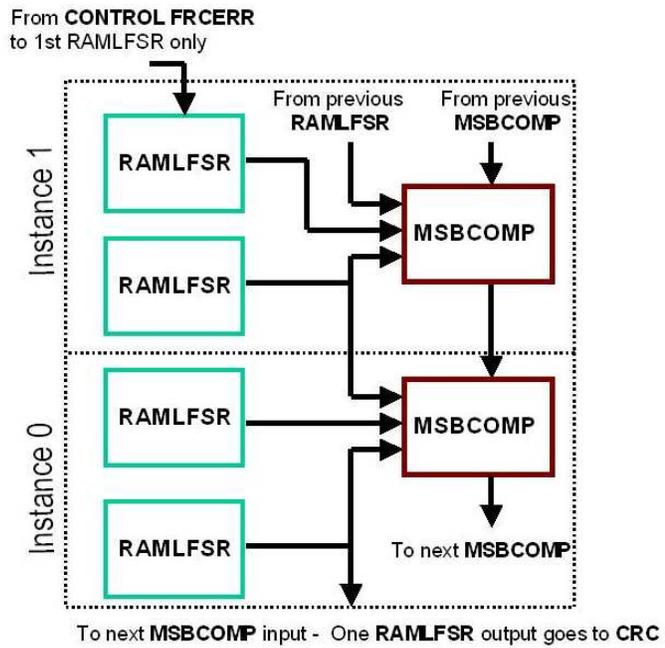


Figure 4. Basic Building Block of XRST Logic Test