LA-UR-02-2907

Title: Single-Event Upset Simulation on an FPGA

Author(s): Eric Johnson, BYU, dej23@ee.byu.edu
Michael Wirthlin, BYU, wirthlin@ee.byu.edu,
Michael Caffrey, LANL, mpc@lanl.gov

Submitted to: ENGINEERING OF RECONFIGURABLE SYSTEMS AND ALGORITHMS (ERSA)

June 24-27, 2002, Las Vegas, Nevada, USA

Single-Event Upset Simulation on an FPGA

Eric Johnson[1], Michael J. Wirthlin[1], and Michael Caffrey[2]
dej23@ee.byu.edu, *wirthlin@ee.byu.edu*, and *mpc@lanl.gov*
[1]Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT
[2]Los Alamos National Laboratory, Los Alamos, NM

Abstract *FPGAs have become an appealing solution for space-based remote sensing applications. However, in a low-earth orbit, FPGAs are susceptible to Single Event Upsets (SEUs). In order to overcome their effects, an SEU simulator based on the SLAAC-1V computing board has been developed. With this tool, work has begun on characterizing the reliability of FPGA designs in the hopes of developing techniques to mitigate the effects of SEUs.*

Keywords: configurable computing, FPGA, single-event upset

1 Introduction

Field Programmable Gate Arrays (FPGAs) are an attractive alternative to programmable processors used for space-based remote sensing. The ability to customize the datapath within an FPGA to an application-specific computation allows the FPGA to perform many operations faster and more efficiently than a programmable processor. FPGA circuits have been created for a variety of remote sensing operations such as multi-spectral filtering, data compression, and frequency transformations.

In addition to improved computational efficiency, FPGAs are reprogrammable and can be configured to perform any user-specified operation. Unlike application-specific integrated circuits (ASICs), FPGAs can be configured *after* the spacecraft has been launched. This flexibility allows the same FPGA resources to be used for multiple instruments, missions, or changing spacecraft objectives. Further, errors in an FPGA design can be resolved by fixing the incorrect design and reconfiguring the FPGA with an updated configuration bitstream.

While the use of FPGAs for remote sensing offers several advantages over conventional computing methods, FPGAs are sensitive to radiation effects in a low earth orbit. FPGAs are sensitive to both heavy ion and proton induced single event upsets (SEUs) to flip-flop state[1]. Upsets in the FPGA affect the user design flip-flops, the FPGA configuration bitstream, and any hidden FPGA registers or state machines.

Configuration bitstream upsets are especially important because such upsets affect both the state and operation of the design. Configuration upsets may perturb the routing resources and logic functions in a way that *changes* the operation of the circuit.

The purpose of this work is to study the effects of configuration SEUs on the operation of FPGA circuit designs. A simulator testbed was created to manually insert SEUs into the configuration bitstream and analyze the effects of the SEU on the circuit design. Based on the SLAAC-1V FPGA computing board, this testbed reconfigures FPGA resources to simulate SEUs in the configuration bitstream. By exploiting the reconfiguration speed of this board, the effects of SEUs upon each individual configuration bit can be rapidly tested and analyzed.

This paper will begin by reviewing the effects of radiation on modern FPGAs. Next, the process of simulating configuration SEUs will be discussed along with a detailed description of the SLAAC-1V SEU simulator. To demon-

strate the effectiveness of this simulator, results from several SEU tests will be presented. The paper will conclude by summarizing the effectiveness of the simulator and discussing future work.

2 Radiation Effects on FPGAs

Electronic circuits operating outside the earth's atmosphere are exposed to levels of radiation much higher than the radiation level found on Earth. High-levels of radiation may damage or upset the operation of a conventional semiconductor device. Electronic circuits can be designed to tolerate high levels of radiation through custom and often expensive manufacturing techniques.

With increased interest in exploiting programmable logic in space related applications, several researchers have investigated the suitability of commercially available FPGAs in radiation environments[2]. The XQVR Virtex FPGA has been tested extensively for radiation tolerance [3]. Based on a thin-epitaxial 0.22 μmm CMOS process, this device tolerates a total dose in range of 80 to 100 krads(Si). This total dose tolerance is acceptable for many applications. Further, there is no risk of latch-up occurring in orbit.

While the XQVR Virtex FPGA is immune to latch-up and has an acceptable total-dose tolerance, it is sensitive to single-event upsets. Single-event upsets are changes in the state of a flip-flop or register caused by heavy ion collisions. SEUs will alter the state of the device and cause the device to produce incorrect results and unspecified behavior. FPGA devices operating in space must anticipate SEUs and incorporate appropriate SEU mitigating techniques.

As shown in Table 1, most of the memory cells within the Virtex XCV1000 device are devoted to configuration memory (over 97%). Configuration memory is used to specify the interconnection, logic, and operating modes of the device. While upsets in the user design may alter the behavior and output of the circuit, upsets within the configuration memory actually *change* the user design. Such upsets may change the circuit in a way that produces incorrect results.

Configuration Bits	5,962,944	97.4%
User Flip-Flops	27,648	0.5%
User BlockRAM Bits	131,072	2.1%

Table 1: Memory Bits within the Virtex XCV1000

Several techniques have been proposed and tested for mitigating SEUs in FPGAs. Many techniques use hardware redundancy to reduce the probability of failure [4]. By replicating the desired circuitry and comparing the results, faults in the configuration can be detected and reported. Other techniques rely on device reconfiguration to continually "clean" the configuration bitstream[5]. By repeatedly configuring the device, SEUs occurring within the configuration bitstream are replaced by the correct value.

The purpose of this work is to create a low-cost testbed for evaluating SEU mitigation techniques on the Virtex configuration bitstream. This testbed relies on commercially available FPGAs and does not require the expense of traditional testing techniques such as a proton accelerator. The configuration SEU testbed will be described in detail in the following section.

3 SLAAC-1V Testbed

An SEU insertion testbed was developed by researchers at the Los Alamos National Laboratory[6]. This testbed is based on a Xilinx Virtex FPGA and the Xilinx Multilinx cable. A number of designs were evaluated on this testbed including a circuit using Triple Modular Redundancy (TMR). Researchers at Xilinx Corporation used this testbed to verify that any single-bit change in the configuration bitstream would not modify the behavior of the

TMR design under test.

The SLAAC-1V testbed described in this paper will build upon these techniques to create a general-purpose testbed that can be used to rapidly simulate configuration SEUs on a variety of designs. The primary goal of this testbed is to improve the time required to simulate configuration SEUs. By rapidly simulating configuration SEUs, it will be possible to perform many tests on a large number of designs.

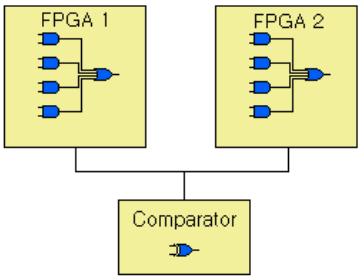

Figure 1: Ideal SEU Simulator Setup

This testbed simulates the effects of SEUs by using two FPGAs each operating with identical designs, as shown in Figure 1. Both FPGAs are sent identical clock and circuit inputs, and should produce identical results. Circuit problems caused by configuration SEUs can be identified by identifying discrepancies in the circuit outputs.

Configuration SEUs can be modeled by intentionally corrupting the configuration bitstream of the FPGA. This is due to the fact that an SEU effectively alters the user design by toggling the state of a flip-flop inside of the FPGA. In order to simulate SEUs, we will alter the configuration bitstream of an FPGA design using partial reconfiguration. This bitstream corruption occurs in only one of the two FPGAs shown in Figure 1.

Once an SEU has been inserted into the configuration bitstream of an FPGA design, both designs are cycled. The design outputs are checked to determine the effect, if any, that the bit change has had upon the behavior of the design. Any discrepancy between the outputs indicates that an SEU occurring at the given location has effectively altered the behavior of the user design.

After determining if the SEU insertion has affected the circuit behavior, the configuration bitstream is repaired and the test continues. Every bit in the entire configuration bitstream is subjected to SEU insertion. Because we are testing each bit in the bitstream, the ability to quickly reconfigure the FPGA is a necessity.

3.1 SLAAC-1V

Developed at USC-ISI East, the SLAAC-1V is a powerful FPGA computing board. As shown in Figure 2, the board contains three Virtex XCV1000 FPGAs, namely X0, X1 and X2. The board also has ten 256Kx36 ZBT SRAMs, a three port crossbar connecting all three FPGAs, and a PCI bus interface[7].

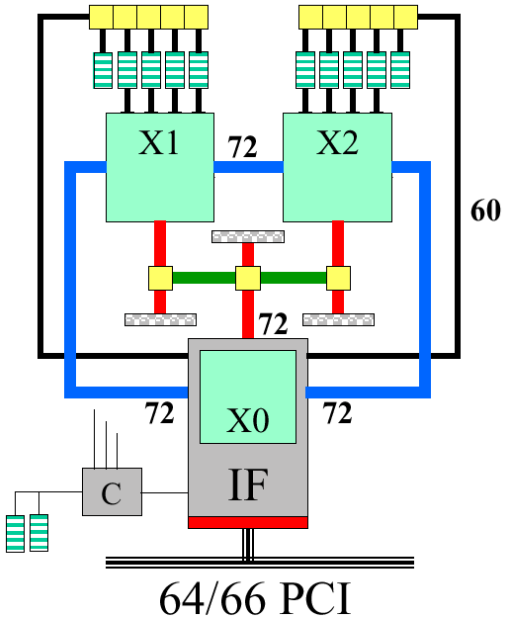

Figure 2: SLAAC-1V

The CC block in Figure 2 is the configuration controller for the computing board. Though this Virtex XCV100 FPGA is not available to the user, it is dedicated to providing reconfiguration of any FPGA on the board. One of the main goals of the SLAAC-1V architecture is to provide an experimental platform for fast partial runtime reconfiguration.

This enables us to perform SEU insertions very quickly, making the SLAAC-1V an attractive choice for use as an SEU simulator.

3.2 Single Bit Corruption

As has already been discussed, the effects of an SEU can be modeled by toggling a bit in the configuration bitstream of an FPGA design. This process is illustrated in Figure 3. This figure assumes that the portion of the bitstream being subjected to SEU insertion represents a Look Up Table(LUT) in X1. A LUT can implement a 4-input logic function. The LUT shown represents a two-input XOR gate.

The configuration bitstream starts out in its initial, uncorrupted state, as shown in Figure 3a. In this state, the output of the design in X1 should match up identically with output of X2.

The configuration bitstream is then altered, as shown in Figure 3b. Only one bit in the bitstream is modified. As can be seen in the figure, this one-bit modification can have a great impact on the overall circuit configuration. The single-bit change in Figure 3b transformed the LUT from a 2-input XOR gate into a 4-input multiple gate logic function. While in this state, it is possible that the outputs of the two FPGAs will not match up, indicating that the SEU has caused a circuit behavior change.

After the configuration bitstream has been altered, it is subsequently repaired, as shown in Figure 3c. At this point, the behavior of the designs in X1 and X2 should again match identically.

4 Testbed Execution and Timing

Every bit in the configuration bitstream of the FPGA must be subjected to SEU insertion. For the Virtex XCV1000, this translates to 5,962,944 bits[8]. After considering that every bit in the configuration bitstream must be toggled, it is quite clear that a fast SEU simulator is a necessity.

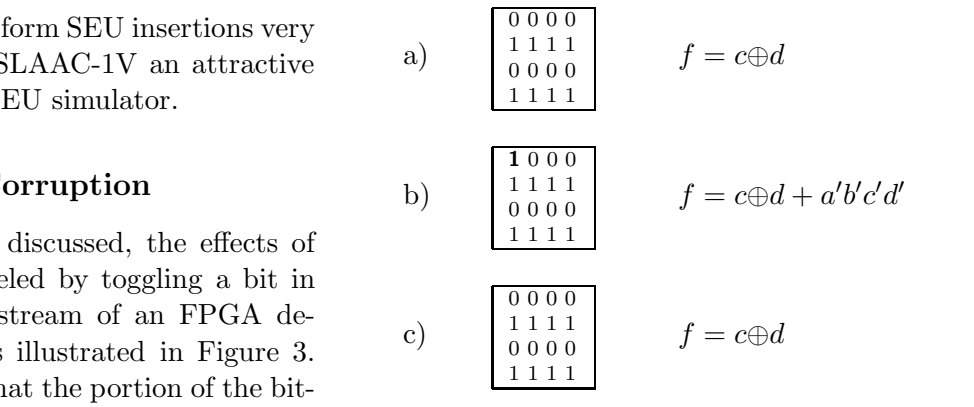

Figure 3: Illustration of the effects of SEU insertion on a LUT

The single-bit corruption described in the previous section is implemented as shown in Figure 4. This inner loop must be executed once for each SEU insertion, or 5,962,944 times. The test begins as follows: the first bit in the configuration bitstream is toggled and the frame in which it is located is reconfigured to FPGA X1 (line 2). All FPGAs on the board are cycled (line 3), and the outputs from X1 and X2 are compared in X0. The result from this comparison is read by the host machine (line 4). If any discrepancies occur, the bit location of the inserted SEU is recorded (line 6), and all of the flip-flops in the X1 and X2 are reset. The altered bit is then repaired and X1 is reconfigured to its original state (line 7). The simulation continues until every bit in the configuration bitstream has been tested.

```
01: do {
02:     corrupt configuration bit -  128us
03:     cycle circuit             -    8us
04:     test for discrepancies.   -    2us
05:     if present
06:        record bit location    -    1us
07:     repair configuration bit  -  128us
08: } until all bits have been tested
                             total  -  267us
```

Figure 4: Configuration Inner-loop

Once the bitstream corruption test has completed for a given circuit design, the locations of SEU insertions which affected the behavior

of the circuit are written to a file on the host machine. In order to verify deterministic results, the procedure of SEU simulation can be repeated an arbitrary number of times.

The SEU insertion inner loop can be performed at a remarkable speed. The execution time for the individual parts of the loop are shown in Figure 4. The entire loop requires only 267μs, which means that performing SEU insertion on the entire configuration bitstream would require just under 27 minutes. The speed with which the test can be executed makes it possible to simulate the effects of SEUs on the configuration bitstream for multiple designs.

5 Testbed Results

The SLAAC-1V SEU simulator has been used to test the sensitivity of a linear feedback shift register (LFSR) design to configuration SEUs. Four LFSR designs were created with varying amounts of logic to determine the impact logic density on SEU sensitivity. The LFSR designs consisted of several 168-bit wide LFSRs in parallel. The most significant bit of each LFSR is connected to one output pin. The output width of these LFSR designs are 18, 36, 54, and 72-bits respectively.

The results of the SEU simulation for these four LFSR designs are summarized in Table 2. This table provides the size of the design (in logic slices), the number of failures, the failure rate, and the normalized failure rate per logic slice. The failure rate is computed by dividing the number of circuit failures by the total number of configuration bits in the bitstream.

The first fact to note from these results is that the LFSR design is relatively insensitive to configuration upsets. The largest LFSR design had a failure rate of .079% – only one configuration upset in 1272 will cause a failure in the design operation. The overall probability of a design failure due to a configuration SEU is the probability of a configuration SEU multiplied by the design failure rate. The low sensitivity of the design to configuration SEUs suggests that the overall failure rate of a design will be much lower than the probability of a configuration SEU.

Another interesting fact to note from this table is that the SEU failure rate is a function of the logic density – larger designs that consume more logic slices are more sensitive to configuration SEUs than smaller designs. Since larger designs use more logic and routing resources, a larger portion of the configuration bitstream will be used to define the circuit functionality. Smaller designs that use fewer resources contain more "don't care" configuration bits within the bitstream and can tolerate more configuration upsets.

	Logic Slices	Failures	Failure Rate†	Flip Flops	Failure /Slice‡
18	3404	1022	1.713	5910	5.04
36	6320	2123	3.560	11724	5.63
54	9236	3456	5.796	17538	6.28
72	12152	4689	7.864	25352	6.47

$^{\dagger}10^{-4}$ failures per configuration upset
$^{\ddagger}10^{-8}$ failures per upset per logic slice

Table 2: Configuration SEU Failure Rate for Linear Feedback Shift Register

While the results in Table 2 suggest that the LFSR is relatively insensitive to configuration SEUs, the results in this table are specific to the LFSR and do not necessarily apply to other FPGA designs. The SLAAC-1V SEU simulator will be used to test a variety of other FPGA designs to better understand the sensitivity of FPGA circuits to configuration SEUs. As more designs and design styles are tested in this SEU simulator, accurate models of configuration SEUs can be created.

6 Conclusions

The SEU simulator described above has been used to successfully test the sensitivity of configuration SEUs on a number of FPGA designs. This simulator computes the failure rate of an FPGA design by testing the behavior of the

design while configuration bit upsets are introduced. To fully characterize the failure rate, each bit within the bitstream is corrupted. Because there are so many configuration bits in the Virtex V1000 bitstream, configuration time is essential for this simulator. The high-speed configuration modes of the SLAAC-1V are used to maximize simulation time.

This work will continue by testing many more designs including those that will be placed on a spacecraft sensor. The simulator will also be used to characterize the effectiveness of design techniques used to improve circuit reliability. Triple module redundancy, circuit checksums, and other redundant hardware techniques can be tested and characterized to determine relative effectiveness of any redundant hardware.

Finally, this simulator will be used to characterize the reliability of specific architectural components of the Virtex FPGA. Specifically, the Input Output Buffers (IOBs), Block RAM, SelectRAM, and routing blocks will be tested to determine local sensitivity to configuration SEUs. By understanding the reliability of FPGAs in the presence of single-event upsets, design techniques can be created to improve reliability and encourage the use of FPGAs for remote-sensing and other space-based applications.

References

[1] E. Fuller, M. Caffrey, P. Blain, C. Carmichael, N. Khalsa, and A. Salazar. Radiation test results of the Virtex FPGA and ZBT SRAM for space based reconfigurable computing. In *MAPLD Proceedings*, September 1999.

[2] E. Fuller, M. Caffrey, A. Salazar, C. Carmichael, and J. Fabula. Radiation testing update, seu mitigation, and availability analysis of the virtex fpga for space reconfigurable computing. In *4th Annual Conference on Military and Aerospace Programmable Logic Devices (MAPLD)*, page P30, 2000.

[3] Phil Brinkley and Carl Carmichael. SEU mitigation design techniques for the XQR4000XL. Technical report, Xilinx Corporation, March 15, 2000. XAPP181 (v1.0).

[4] Carl Carmichael. Triple module redundancy design techniques for virtex FPGAs. Technical report, Xilinx Corporation, November 1, 2001. XAPP197 (v1.0).

[5] Carl Carmichael, Michael Caffrey, and Anthony Salazar. Correcting single-event upsets through virtex partial configuration. Technical report, Xilinx Corporation, June 1, 2000. XAPP216 (v1.0).

[6] F. Lima, C. Carmichael, J. Fabula, R. Padovani, and R. Reis. A fault injection analysis of Virtex FPGA TMR design methodology. In *Proceedings of the 6th European Conference on Radiation and its Effects on Components and Sysemts (RADECS 2001)*, 2001.

[7] USC-ISI East. *SLAAC-1V User VHDL Guide*, October 1, 2000. Release 0.3.1.

[8] Xilinx, Inc. *Virtex Series Configuration Architecture User Guide, Xilinx Application Notes 151, v1.5*, September 2000.