LA-UR-03-2750

Approved for public release;
distribution is unlimited.

Title: THE RELIABILITY OF FPGA CIRCUIT DESIGNS IN THE PRESENCE OF RADIATION INDUCED CONFIGURATION UPSETS

Author(s): Michael Joseph Wirthlin, Brigham Young University
Darrel Eric Johnson, Brigham Young University
Nathaniel Hatley Rollins, Brigham Young University
Michael Paul Caffrey, 111180, NIS-3
Paul Stanley Graham, 181856, NIS-3

Submitted to: IEEE Symposium on Field-Programmable Custom Computing Machines, April 8-11, 2003, Napa, CA, USA

Los Alamos
NATIONAL LABORATORY

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.

Form 836 (8/00)

The Reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets*

Michael Wirthlin, Eric Johnson, and Nathan Rollins
Department of Electrical and Computer Engineering
Brigham Young University, Provo, UT
Michael Caffrey and Paul Graham
Los Alamos National Laboratory, Los Alamos, NM

Abstract

FPGAs are an appealing solution for space-based remote sensing applications. However, in a low-earth orbit, FPGAs are susceptible to Single-Event Upsets (SEUs). In an effort to understand the effects of SEUs, an SEU simulator based on the SLAAC-1V computing board has been developed. This simulator artifically upsets the configuration memory of an FPGA and measures its impact on FPGA designs. The accuracy of this simulation environment has been verified using ground-based radiation testing. This simulation tool is being used to characterize the reliability of SEU mitigation techniques for FPGAs.

1 Introduction

There is increasing interest in the use of reconfigurable computing in space-based applications such as remote sensing[1]. The use of reconfigurable Field Programmable Gate Arrays (FPGAs) within a spacecraft allows the use of application-specific hardware in place of programmable processors. 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 the use of traditional programmable processors.

In addition to improved computational efficiency, the use of SRAM-based FPGAs within a spacecraft allows the programmable hardware 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. Errors in an FPGA design can be resolved by fixing the incorrect design and reconfiguring the FPGA with an updated configuration bitstream. Further, custom circuit designs can be created to avoid FPGA resources that have failed during the course of the spacecraft mission.

While the use of FPGAs for remote sensing offers several advantages over conventional computing methods, SRAM-based 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)[2]. Single-event upsets in the FPGA affect the user design flip-flops, the FPGA configuration bitstream, and any hidden FPGA registers, latches, or internal state. 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 reliability of FPGA designs in the presence of configuration SEUs. An important component of this work is a fault simulator that was created to manually insert SEUs into the configuration bitstream[3]. Based on the SLAAC-1V FPGA computing board, this testbed reconfigures FPGA resources to simulate SEUs in the configuration bitstream. A number of experiments were conducted on this simulator to analyze the susceptibility of FPGA designs to configuration upsets. The results of this fault simulator were verified using a ground radiation source. The results of both the simulator and radiation testing will be presented.

This paper will begin with an overview of radiation effects on FPGAs. Next, the process of simulating configuration SEUs will be discussed along with a

*This work is supported by the Deployable Adaptive Processing Systems project (DAPS) at the Los Alamos National Laboratory, United States Department of Energy.

detailed description of the SLAAC-1V SEU simulator. To demonstrate the effectiveness of this simulator, results from several SEU tests will be presented. After presenting the simulator results, this paper will discuss the radiation testing experiments that were completed to verify the fault simulator. The paper will conclude by summarizing the results of the experiments and discussing future work.

2 Effects of Radiation on FPGAs

Electronic circuits operating outside the earth's atmosphere are exposed to a radiation environment much different than the radiation 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 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[4, 5]. This section will discuss the effects of radiation on integrated circuits and present current results on how these radiation effects apply to modern SRAM-based FPGAs.

2.1 Effects of Radiation on Integrated Circuits

An important function of the earth's atmosphere is to filter the ionizing radiation found in space. Without the atmosphere, the earth would be subject to the high energy radiation found in space. The radiation found in most earth orbits is caused by protons and heavy ions emitted by the sun (i.e. solar particles), galactic cosmic rays, and particles trapped in the earth's magnetic field.

Space radiation has both long-term and single particle effects on electronic components. Long-term effects include total ionizing dose (TID). Single-event effects include single-event latchup (SEL) and single-event upset (SEU). Each of these effects must be considered before using a device in a space application.

Total Ionizing Dose (TID) Total ionizing dose is the long term ionizing damage to a semiconductor device caused by high energy protons and electrons. Exposure to high-energy ionizing radiation generates electron-hole pairs within the oxide of a MOS device. These generated carriers cause a buildup of charge within the oxide. This buildup of charge will change the threshold voltage, increase the leakage current, and modify the timing of the MOS transistors. Ultimately, ionizing radiation will cause functional failures within the device.

Single-Event Latchup (SEL) Single-event latchup is a potentially destructive condition in which a single charged particle induces latchup within a CMOS device. With enough energy, a charged particle may trigger the parasitic npn-pnp circuit found within CMOS circuits. Once in latchup, high currents will flow through the parasitic bi-polar transistors and destroy the device.

Single-Event Upsets (SEU) A single-event upset is the change in state of a digital memory element caused by an ionizing particle. As the ionizing particle passes through the device, charge can be transfered from one node to another. This charge transfer can lower the voltage of a memory cell and change its internal state. These single-event upsets are soft errors that do not cause any permanent damage within the device.

2.2 Radiation and FPGAs

SRAM-based FPGAs suffer the same challenges with respect to radiation as other semiconductor devices. Users of FPGA devices must consider these radiation effects before including an FPGA within a space application. To address the need for radiation tolerant FPGAs, Xilinx has introduced the QPRO^{tm} line of high-reliability FPGA family[6]. This radiation tolerant FPGA is manufactured on a thin-epitaxial 0.22 μmm CMOS process and based on the commercially available Virtex FPGA architecture.

The QPRO^{tm} high-reliability Virtex FPGA has been tested extensively for radiation tolerance and has been shown to tolerate a total dose in range of 80 to 100 krads(Si). This total dose tolerance is acceptable for many space applications. In addition, this device is immune to latchup up to an LET of 125 MeV=cm^2/mg.

While the QPRO^{tm} Virtex FPGA is immune to latch-up and has an acceptable total-dose tolerance, it is sensitive to single-event upsets (SEUs). Single-event upsets are changes in the state of a flip-flop, latch, or register caused by heavy ion collisions. The worst-case upset sensitivity of the XQV V1000 was calculated for the Cibola flight experiment orbit. As shown in Table 1, memory cells are anticipated to upset at a rate of .13 upsets/hour (3.2 upsets/day) in a quite sun environment and upset at a rate of 4.2 upsets/hour during the peak upset rate.

Devices that contain dense arrays of memory cells are especially sensitive to such SEUs due to the large

Radiation Environment	Upset Rate
Quite Sun (Orbit Avg.)	.13/hour
Flare Enhanced (Orbit Avg.)	1.1/hour
Peak Rate (SAA Region)	4.2/hour

Table 1. SEU Upset Rates for the V1000 FPGA

amount of memory state within a relatively small amount of circuit area. Much like SRAM and DRAM, SRAM-based FPGAs contain large amounts of memory cells within a device and are especially sensitive to radiation induced SEUs. As suggested in Table 2, the Virtex V1000 FPGA contains almost six-million bits of internal state. This known internal state is used for the following important purposes:

User Flip-Flops An important architectural component of all FPGAs are user programmable flip-flops. User designs exploit these flip-flops to implement common sequential logic circuits such as state machines, counters, and registers. User flip-flops in most digital technologies are susceptible to radiation-induced single-event upsets. Many digital circuits operating in a radiation environment exploit redundancy (i.e. multiple flip-flops) to mitigate against such single-event effects [7].

Memory Type	# Bits	%
Configuration	5,810,048	97.4%
LUT Bits†	393,216	6.7%
Block RAM	131,072	2.2%
User Flip-Flops	26,112	0.4%
Total	5,967,232	100%

† LUT Bits are a subset of the configuration memory.

Table 2. Memory Bits Within the Virtex XCV1000

User Memory Modern FPGAs provide blocks of internal memory larger than the typical look-up table. This block memory is used for traditional random access memory functions such as data storage, buffering, FIFO, etc. The Virtex family includes a set of internal dual-ported BlockRAM memories that provide 4096-bits of randomly accessible memory. Dense static memory such as the BlockRAM is especially susceptible to radiation-induced SEUs. Well-known error-correction coding techniques are often used within a user design to detect and correct such upsets[8].

Configuration Memory A large amount of memory cells are required to define the operation of the user-designed FPGA circuit. These memory cells define the operation of the configurable logic blocks, routing resources, input/output blocks, and other programmable FPGA resources. The use of static memory cells for configuration storage allows the device to be reprogrammed as often as necessary by reloading a new configuration memory. Like other static memory cells, configuration memory is susceptible to single-event upsets. Upsets within the configuration memory are especially troublesome as they may *change* the operation of the circuit. Several techniques have been proposed for detecting and mitigating such upsets.

Half-Latches Another form of internal state found within the Virtex FPGA is the "half-latch" structure. Half-latch structures are used to generate many of the constant "0" and "1" logic values used throughout a user FPGA design. For example, the half-latch in Figure 1 generates a constant "1" for a clock-enable signal of a user flip-flop. Unlike other internal state, half-latches are not visible to the circuit or the user. Because of this lack of visibility, upsets within a half-latch cannot be detected. To prevent undetectable half-latch upsets from occurring, half-latch structures must be removed from a design[9].

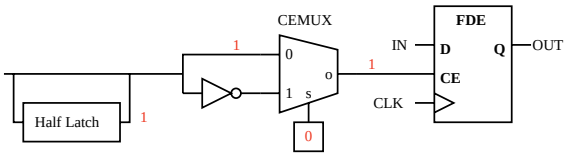


Figure 1. Half-Latch Generating a Constant "1"

As shown in Table 2, 97% of the memory cells within the Virtex V1000 device are devoted to configuration memory. Because of the large amount of configuration memory within the device, the configuration memory is especially susceptible to single-event upsets (SEUs). While upsets in the user flip-flops or memory may alter the state and output of the circuit, upsets within the configuration memory actually *change* the user design. Upsets within the configuration memory may alter the function of the configurable

logic blocks, upset the routing network, or modify the operation of the input/output blocks. Any spacecraft that utilizes SRAM-based FPGAs must anticipate and mitigate against upsets within the device configuration memory.

Several techniques have been proposed and tested for mitigating SEUs in FPGAs. Many techniques use hardware redundancy to reduce the probability of failure[10]. 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 "scrub" the configuration bitstream[11]. 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 Configuration Fault Simulator

Ground-based radiation sources are typically applied to electronic circuits to simulate the radiation within a natural space environment (i.e. solar and cosmic radiation). Electron linear accelerators and proton cyclotrons are commonly used to test both the total-dose response and proton-induced single-event effects of electronic devices[12, 13]. This form of radiation testing is essential for the characterization and qualification of radiation hardened electronic devices used within a spacecraft.

While this form of radiation testing is important, there are several drawbacks of using ground-based radiation sources to test the behavior of upsets within the FPGA configuration memory. First, radiation testing is relatively expensive. Second, the number of radiation tests is limited by the availability of the testing facility and the need to travel. Third, ground-based radiation tests insert high-energy particles into a device in a random, undirected manner. While such random radiation is similar to the radiation occurring in space, it does not allow the ability to create targeted tests that evaluate the behavior of specific FPGA resources.

To facilitate the frequent testing of upsets within an FPGA, a configuration fault simulator was developed at Brigham Young University (BYU) in conjunction with Los Alamos National Laboratory (LANL) [3]. This system simulates upsets within the configuration memory by *artificially* inserting faulty bits within the configuration bitstream. The goal of this simulator is to test the operation of FPGA designs in the presence of configuration upsets *without* the need of ground-based radiation testing.

3.1 Fault Simulator Architecture

This fault simulator is based on the architecture shown in Figure 2. Two FPGAs are configured with equivalent designs and are run with identical clock and circuit inputs. Under normal operating circumstances, the two FPGAs produce identical results. During fault simulation, one of the FPGAs is subjected to artificial modifications in the configuration bitstream. The fault simulator will monitor the behavior of the FPGA design under test by comparing the circuit output with that of the golden FPGA design. If discrepancies are found, the fault condition is recorded and the bitstream is repaired.

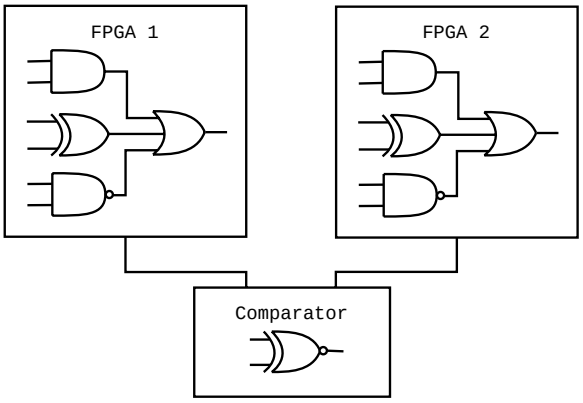


Figure 2. SEU Simulator Architecture

The fault simulator was developed using the SLAAC-1V configurable computing board at USC-ISI East[14]. This board provides three Virtex V1000 FPGAs, a crossbar interconnect, external SRAM memory, and a PCI bus interface. The fault simulator architecture maps nicely to the SLAAC-1V board – the X1 FPGA is for the control FPGA design, the X2 FPGA is used for the design under test and the X0 FPGA is used to compare the results and provide a stimulus to the two FPGA designs.

An important goal of this simulator is to inject artificial configuration faults into the design under test as fast as possible. This simulator achieves rapid fault injection by exploiting the high-speed PCI configuration mode of the SLAAC-1V and the partial configuration SelectMap configuration interface provided on the Virtex FPGA. Using these high-speed configuration techniques allows the fault simulator to corrupt a single

configuration bit in 100 micro-seconds.

3.2 Simulator Execution Sequence and Timing

The fault simulator follows a simple sequence to test the impact of a configuration single-event upset on the design under test. The simulator begins by reconfiguring the design under test with a modified version of the original bitstream. Specifically, a single bit within the original bitstream is toggled to simulate a single-bit configuration memory upset. The simulator cycles the FPGAs to simulate the operation of the circuit in the presence of a single configuration bit upset.

While the simulator cycles the two FPGAs, the comparator circuit operating in X0 monitors the output of the FPGAs to detect any discrepancies in the circuit behavior. If a discrepancy is found between the two circuits, the bit location of the corrupted configuration bit is recorded and archived for later analysis. After this execution test has been completed, the corrupted configuration bit is repaired through a final reconfiguration step. The inner loop for this corruption process is shown in Figure 3.

```
01: do {
02:     corrupt configuration bit
03:     test for design discrepancies
04:     if discrepancies exist
05:        record bit location
06:     repair configuration bit
07: } until all bits have been tested

  Total time: 214 usec/loop
              20 minutes 44 sec/bitstream
```

Figure 3. Configuration Inner-Loop

To completely characterize the behavior of a design in a radiation environment, *each* of the Virtex V1000 configuration bits must be tested using this process. The inner loop shown in Figure 3 is repeated for each of the 5,810,048 configuration bits required by the Virtex V1000 device. With iteration of the fault simulator requiring 214μs, testing of the entire bitstream takes 20 minutes.

4 Testbed Results

The configuration upset simulator has been used to test the effects of upsets within the configuration memory on several FPGA designs. The first set of designs include a number of pipelined array multipliers. These multiplier designs are used to evaluate the sensitivity of datapath circuitry to configuration SEUs. The second set of designs include several linear feedback shift registers (LFSR). Unlike the multiplier designs, the LFSR designs consume relatively few logic resources while requiring a large number of user flop-flops. The LFSR also contains feedback and will highlight the effects of configuration upsets on circuits with feedback.

Several designs were created with varying amounts of logic for both the multiplier and the LFSR. Using a variety of design sizes will identify the impact of logic density on configuration SEU sensitivity. It is expected that larger designs requiring more logic will be more sensitive to configuration upsets than smaller designs.

The test procedure outlined in Figure 3 is applied to each of the multiplier and LFSR designs. During this test sequence, *each* of the almost six-million configuration bits are independently upset within the design. Configuration upsets that cause a failure in the design are recorded and stored for analysis. The following information will be reported for each design test:

Logic Slices The size of a design is specified by the number of logic slices used by the design. A logic slice within the Virtex architecture contains two 4-input look-up tables, two flip-flops and is roughly equivalent to 30 logic gates.

Flip-Flops The size of a design is also specified by the number of user flip-flops consumed by the design. This parameter is useful in identifying the density of flip-flops found within the design.

SEU Design Failures The total number of upsets within the configuration memory that cause an operational failure are identified as SEU design failures.

Failure Rate The failure rate is computed by dividing the number of SEU design failures in a test by the total number of configuration bits in the bitstream (i.e. 5,810,048). This value indicates the probability that an upset within the configuration memory will disrupt the operation of the circuit under test. Note that this result is applicable only to the design under consideration.

Failures per Logic Slice The number of SEU failures can be normalized to the design size by dividing the number of failures by the number of slices consumed by the design. It is expected that the normalized failure count will be relatively constant for designs of similar composition.

Normalized Failure Rate The failure rate can also be normalized to the design size by dividing the number of failures per logic slice by the average number of configuration bits required for a slice. With 5,810,048 bits necessary to configure 12,288 logic slices, an average of 473 bits are required to configure each logic slice. The normalized failure rate estimates the percentage of configuration bits *used* by a design that are sensitive to single event upsets.

The results of the SEU simulation for each of the designs are summarized in Table 3. The following two sections will describe each of these designs in detail and discuss the results from the fault simulator.

4.1 Multiplier Test Designs

The multiplier designs are arranged in a multiply-accumulate (MAC) configuration, as shown in Figure 4. The MAC design is a feed-forward datapath circuit that is representative of computing kernels used in many RF signal processing algorithms. This circuit contains a two stage pipeline and performs the arithmetic function $O = (A \times B + A \times B) + A \times B$.

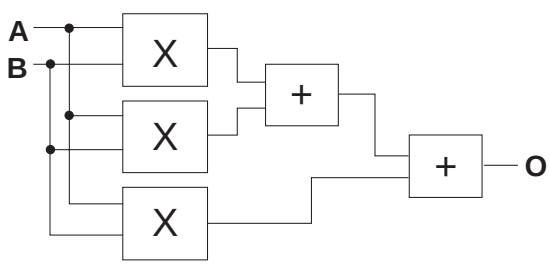


Figure 4. Multiplier-Accumulate Test Design.

Two different styles of multiplication were used for these tests. The first multiplier style uses simple logic primitives (i.e. AND/OR/INVERT) to implement the MAC circuit. Four different sizes of this circuit were tested. The size of the design is changed by varying the width of the arithmetic operators. The datapath width of these designs are 12, 24, 36, and 48-bits. The second multiplication style exploits architectural features specific to the Virtex FPGA. The use of Virtex-specific primitives allows the construction of multipliers that are smaller and faster than the generic logic multiplier. Four sizes of the Virtex multiplier were also tested. The datapath width of the Virtex multiplier designs are 18, 36, 48, and 72-bits.

The sensitivity of these datapath circuits to configuration upsets is summarized in Table 3. The first fact to note from these results is that the multiplier designs are relatively insensitive to configuration upsets. The largest Logic multiplier design had a failure rate of 3.8% – only one configuration upset in 26 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.

Another interesting fact to note from these tables 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. The number of failures per slice is relatively constant for each of the two classes of multipliers. 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.

It is also interesting to compare the failure rate of the Logic multiplier with that of the Virtex-specific multiplier. As seen in Table 3, the Virtex-specific multipliers are smaller than their generic logic counterparts. Because these multipliers are smaller, fewer configuration upsets will alter the operation of the multiplier. However, the number of SEU failures per logic slice is higher for the Virtex multiplier than that of the generic multipliers. This suggests that the Virtex multipliers use FPGA resources more efficiently than the generic multipliers and are thus more sensitive to SEUs on a per-slice basis.

4.2 Linear Feedback Shift Register (LFSR) Designs

The configuration fault simulator has also been used to test several linear feedback shift register(LFSR) designs. LFSRs are frequently used for high-speed counters, pseudo-random number generators, and encryption/decryption algorithms. LFSRs sequence through a series of $2^N - 1$ states where N is the number of registers in the LFSR. LFSRs are constructed by creating a linear shift register and adding feedback by performing an exclusive or (XOR) on predefined bits within the register[15]. The LFSR shown in Figure 5 demonstrates an 8-bit counter that implements the polynomial $g(x) = 1 + x^2 + x^3 + x^4 + x^8$.

Four LFSR designs were created with varying amounts of logic. The output widths of these LFSR designs are 18, 36, 64, and 72 bits, respectively. Each output bit is the results of an XOR function applied to the most significant bit output of eight separate LFSRs. The LFSRs are all 12 bits wide, a width small enough to allow for reasonable coverage of all possible

Design	Logic Slices	Flip Flops	SEU Design Failures	Failures Rate	Failure /Slice	Normalized Failure Rate
12-bit Logic	144	144	13263	0.23%	92.1	19.5%
24-bit Logic	561	612	52454	0.90%	93.5	19.8%
36-bit Logic	1249	1404	122657	2.11%	98.2	20.8%
48-bit Logic	2205	2520	220197	3.79%	99.9	21.1%
18-bit Virtex Multiplier	583	1000	60929	1.05%	104.5	22.1%
36-bit Virtex Multiplier	2206	3744	232239	4.00%	105.3	22.3%
54-bit Virtex Multiplier	4781	8848	520747	8.96%	108.9	23.0%
72-bit Virtex Multiplier	8308	15264	856802	14.75%	103.1	21.9%
18-bit LFSR	2178	2160	66797	1.15%	30.7	6.5%
36-bit LFSR	4356	4320	137861	2.37%	31.6	6.7%
54-bit LFSR	6534	6480	208536	3.59%	31.9	6.7%
72-bit LFSR	8712	8640	279450	4.81%	32.1	6.8%

Table 3. Configuration SEU Failure Rate for Three FPGA Design Types.

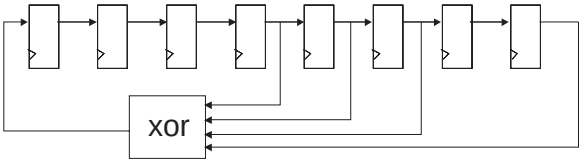


Figure 5. Linear Feedback Shift Register.

outputs ($2^{12} = 4096$), while still large enough to create a design which uses a significant portion of available resources on the FPGA. As a result of the nature of the LFSR, the major constraint for these designs is the amount of available FPGA logic resources, whereas the major constraining factor for the multiplier designs was the amount of available routing within the FPGA.

The results of the SEU simulation for these four LFSR designs are summarized in Table 3. The first fact to note from these results is that the failure rate for LFSRs is linear with respect to design size. Like the multiplier designs, the normalized failure rate is relatively constant for the four design sizes.

The second fact to note from these results is that the LFSR design is less sensitive to configuration upsets than the multiplier design. The normalized failure rate of the LFSR designs is less than half that of the multiplier designs. This outcome can be explained by noting that the LFSR design uses far less combinational logic and routing than the multiplier design. Most logic slices are used to hold flip-flops and contain little, if any, combinational logic. Because the logic density of the LFSR is lower than that of the multipliers, there are fewer sensitive configuration bits than that of the multiplier

The results in Table 3 are specific to the LFSR and multiplier designs and do not necessarily apply to other FPGA designs. The configuration fault 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.

4.3 Location of Sensitive Configuration Bits

It is clear from Table 3 that the failure rate of a design is dependent upon both the size of the design and the type of resources used by the design. This suggests that modification of the *allocated* FPGA resources has an impact on design behavior – the upset of configuration bits associated with *unused* FPGA resources has no impact on the operation of the design. This suggestion can be verified by plotting the location of the sensitive configuration bits and correlating them with the layout of the allocated FPGA resources of a design.

A plot of the sensitive configuration bits can be made by determining the `x,y` location of each sensitive configuration bit determined through the fault simulation process. The `x,y` location of a configuration bit can be made by following the guidelines found in the Xilinx Application Note 151[16]. Figure 6(a) shows the location of sensitive configuration bits of a 48-bit multiplier-accumulate design.

The corresponding layout of this design is shown in Figure 6(b). This image was created by taking a screen capture of the FPGA Editor layout tool provided with the Xilinx tools. Darkened regions of this image indicate routing and logic resources allocated by this design. The location of sensitive configuration bits corre-

lates closely with the regions of the FPGA allocated for this design. This correlation is quite encouraging, as it supports the validity of the fault injection simulator.

5 Radiation Testing

The fault injection simulator provides a convenient and low-cost facility for analyzing the reliability of FPGA designs in the presence of configuration upsets. This simulator is able to rapidly analyze the behavior of a design when *each* of the almost six million configuration bits are upset. Because the simulator will be used to test many FPGA designs, it is important to validate the simulator using a ground-based radiation source.

To validate the fault simulator and the results obtained in Table 3, the SLAAC-1V fault simulation environment was tested at the Crocker Nuclear Laboratory, University of California, Davis. Rather than artificially inserting configuration upsets as described in Section 3, this test was organized to upset the configuration memory (and other internal state) using a medium-energy proton beam. The device under test (X1) is placed in front of the proton beam and exposed to the appropriate flux of radiation. The control device (X2), is shielded from the proton beam and will operate in parallel with the X1 circuit as described earlier. The comparator chip (X0) is used to monitor the operation of both circuits and signal to the host a design failure in X1. The accelerator test configuration containing the SLAAC-1V board is shown in Figure 7.

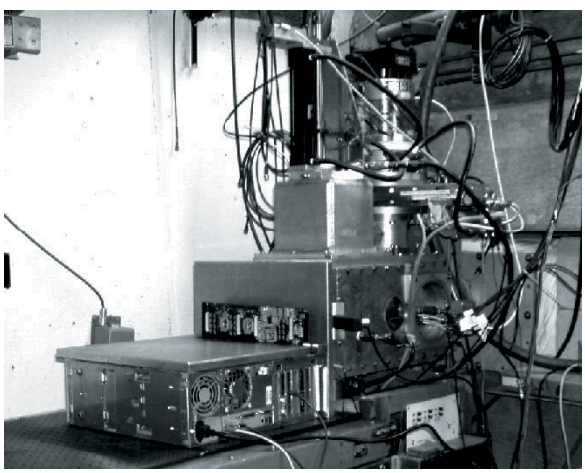


Figure 7. Proton Accelerator Test.

The radiation testing procedure is similar to the artificial fault injection simulator with the exception that configuration bits are not artificially injected. The sequence shown in Figure 8 is repeated until the appropriate amount of testing time is complete. This sequence begins by querying X0 for design output errors. If errors are found, it is recorded with a time-stamp. Next, the host performs a device readback on the X1 FPGA to obtain the current state of the configuration bits. This configuration state is compared against a known correct copy of the bitstream to determine the presence of configuration memory upsets. If a configuration bit has been upset by a proton, the configuration bit is recorded and the device is reconfigured to its original state. Finally, the board is reset to re-synchronize the two designs as necessary.

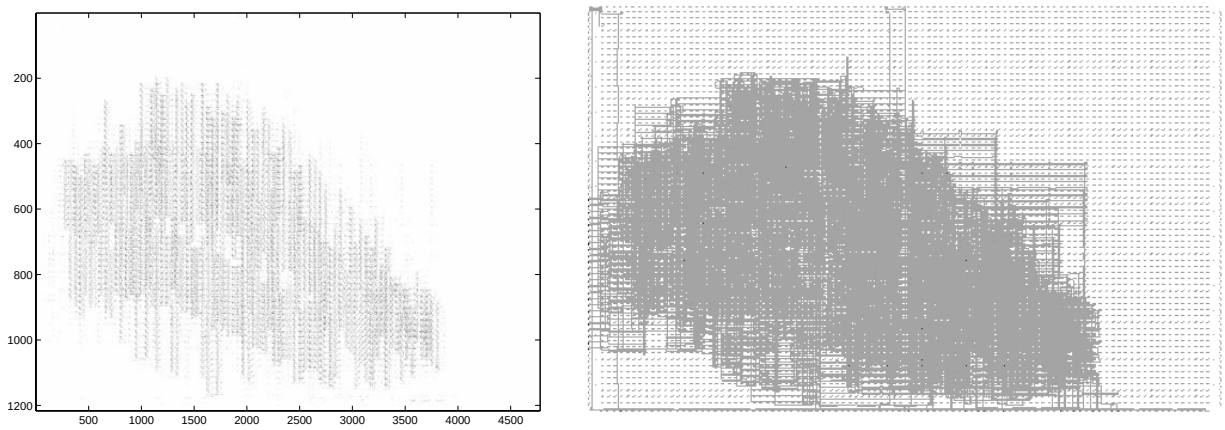
```
do {
    check for design failures
    if errors exist
       set flag & record time stamp
    perform device readback
    examine configuration bitstream for errors
    if errors exist
       record bit error location(s)
       repair configuration bits
    if flag from output error condition
       reset board
} until testing time complete
```

Figure 8. Configuration Inner-Loop

The testing sequence described above operates continuously throughout the test with each iteration of this sequence taking 430 milli-seconds. The speed of this sequence is limited by the time required to perform a configuration readback operation on the device. Once a sufficient number of configuration memory upsets are recorded, the test is stopped.

Over 60 radiation tests were conducted over a 16 hour period. The radiation source was configured to provide roughly one proton-induced configuration upset every second. During this time, over 50,000 configuration upsets were recorded. The preliminary results for three of the tests are shown in Table 4. This table lists the number of configuration upsets and design failures observed during the test. In addition, this table provides the failure rate of both the radiation test and the artificial fault simulator.

The accelerator results shown in Figure 4 suggest that the fault injection simulator closely matches the results seen in the accelerator. This similarity suggests that the fault-injection simulator is indeed a valid technique for simulating the behavior of FPGA circuits in the presence of radiation induced single-event upsets. It is important to note that the failure rates of the accelerator tests are slightly higher than that of the simulator. This difference can be attributed to the fact that



(a) Location of Sensitive Configuration Bits

(b) Design Layout and Routing

Figure 6. Relation of Sensitive Configuration Bits to Design Layout.

Design	Test Time (seconds)	Configuration Upsets	Design Failures	Failure Rate	Simulator Failure Rate
36-bit Virtex Multiplier	3707	3002	148	4.9%	4.0%
72-bit Virtex Multiplier	6314	5753	992	17.2%	14.75%
72-bit LFSR	1033	1067	53	5.0%	4.81%

Table 4. Proton Accelerator Test Results.

the accelerator will upset *all* state within the FPGA device and not just the configuration memory. As upsets occur within the user flip-flops and other device state, additional output errors will be seen.

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.

This simulator will also 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] Michael Caffrey. A space-based reconfigurable radio. In Toomas P. Plaks and Peter M. Athanas, editors, *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Al-*

gorithms (ERSA), pages 49–53. CSREA Press, June 2002.

[2] 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.

[3] Eric Johnson, Michael J. Wirthlin, and Michael Caffrey. Single-event upset simulation on an FPGA. In Toomas P. Plaks and Peter M. Athanas, editors, *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, pages 68–73. CSREA Press, June 2002.

[4] R. Katz, K. LaBel, J.J. Wang, B. Cronquist, R. Koga, S. Penzin, and G. Swift. Radiation effects on current field programmable technologies. *IEEE Transactions on Nuclear Science*, 44(6):1945–1956, December 1997.

[5] 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.

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

[7] 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.

[8] John Baylis. *Error Correcting Codes: A Mathematical Introduction*. CRC Press, 1997.

[9] Paul Graham, Michael Caffrey, Michael Wirthlin, Eric Johnson, and Nathan Rollins. Reconfigurable computing in space: From current technology to reconfigurable systems-on-a-chip. In *24th Anual IEEE Aerospace Conference*, 2003. To be published.

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

[11] 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).

[12] J. M. Benedetto, Jr. H.E. Boesch, and T.R. Oldham. Measurement of low-engergy dose-enhancement in MOS devices with metal silicon gates. *IEEE Transactions on Nuclear Science*, 34(6):1540–1543, December 1987.

[13] M. R. Shaneyfelt, K. L. Hughes, J. R. Schwank, F. W. Sexton, D. M. Fleetwood, and P. S. Winokur. Wafer-level radiation testing for hardness assurance. *IEEE Transactions on Nuclear Science*, 38(6):1586–1605, December 1991.

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

[15] S. W. Golomb. *Shift Register Sequences*. Holden-Day, 1967.

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