An Introduction to Radiation-Induced Failure Modes and Related Mitigation Methods For Xilinx SRAM FPGAs

Heather Quinn, Paul Graham, Keith Morgan, Jim Krone, Michael Caffrey, and Michael Wirthlin

Abstract— **Over the past decade, reconfigurable, static random-access memory (SRAM) based field-programmable gate arrays (FPGAs) have made inroads into space-based computation tasks. As these devices are well-suited to the digital signal processing tasks that are often the focus of space-based processing, these devices could provide orders of magnitudes speedup over traditional radiation-hardened processors at a fraction of the cost. Unfortunately, all commercially available SRAM-based devices have problems with the harsh radiation environment in space. This paper will provide an introduction to the potential radiation-induced faults and possible mitigation methods**

I. Introduction

Over the past decade, several organizations have started using static random-access memory (SRAM) based field-programmable gate arrays (FPGAs) in space-based computational platforms. SRAM-based FPGAs can provide orders of magnitude speedup over traditional radiation-hardened microprocessors without the cost of manufacturing application-specific integrated circuits (ASICs). The reprogrammable nature of these devices allows them to be updated while on mission, which could extend the usable lifetime of space-based electronics. Finally, these devices are well suited to digital signal processing applications that are often the focus of space-based computational platforms.

The space environment has a very rich radiation environment of electrons, protons and heavy ions. Each orbit is characterized by the flux and energy of these radiation sources. Unfortunately, all commercial SRAM devices are affected by these radiation environments, and SRAM-based FPGAs are no exception. The two most common radiation-induced faults are single-event upsets (SEUs) and single-event functional interrupts (SEFIs). In SRAM-based FPGAs, SEUs occur when radiation induces a memory cell to change its value (also called a *bit flips*). Likewise, a SEFI is an SEU that has ab effect on device control related logic that results in an interruption in how the device normally functions globally or regionally. An SEU, for example, occurs when the state of a user flip-flop is inverted, while a SEFI occurs when an SEU affects the programming logic for the FPGA. can cause the device to become unresponsive.

While SEUs and SEFIs make reliable processing more challenging than ground-based systems, we have found many traditional fault-tolerance algorithms can be used to mask the effects of radiation-induced faults. To this end, triple-modular redundancy (TMR) can be used to mask a single error in the system in a fully redundant system. Furthermore, using the on-line reconfiguration ability of these devices can remove SEUs from the device so they do not accumulate.

In this paper, we will present a taxonomy of possible failure modes for Xilinx SRAM-based FPGAs in harsh radiation environments, as well as a number of fault tolerance methods that can be used to mitigate their effects. This paper is an update of our 2003 paper [1] and a shortened version of our Design Guide available through the FPGA Mission Assurance Center [2]. For a more in-depth discussion of radiation effects, the authors suggest "The Radiation Effects Handbook" [3]. Section II provides background information on the important radiation effects for SRAM-based FPGAs devices. Section III covers the radiation-induced failure modes, and Section IV covers mitigation methods.

Document release number: LA-UR-08-0843. This work was funded by the Department of Energy through the Deployable Adaptive Processing Systems and Sensor-Oriented Processing and Networking projects, the Cibola Flight Experiment project at Los Alamos National Laboratory, NASA through the Reconfigurable Hardware in Orbit project under AIST contract #NAG5-13516, and the Air Force Research Laboratory under the FPGA Mission Assurance Center.

H. Quinn, K. Morgan, P. Graham, J. Krone, and M. Caffrey are with ISR-3 Space Data Systems, Los Alamos National Laboratory, Los Alamos, NM, 87545 USA (phone: 505-665-7041, e-mail: hquinn@lanl.gov)

M. Wirthlin is with Brigham Young University, Provo, UT, 84602 USA

Category: Regular paper

Technical Session: Reconfigurable Computing for Space Based Applications

Keywords: FPGAs, failure modes, redundancy, scrubbing

II. Basic Radiation Effects on Semiconductor Devices

For hardware to be used reliably on orbit the devices must be able withstand the radiation environment and still process reliably over the usable lifetime of the spacecraft. While there are number of radiation-induced faults that could beset space-based hardware, most designers are concerned about total ionizing dose (TID) and single-event effects (SEE, also know as single-event phenomena). Single-event phenomena can take many forms, such as single-event latchup (SEL), single-event transients (SET), SEUs, and SEFIs. These effects are discussed in greater detail below.

A. Total Ionizing Dose

While deployed, the voltage and switching characteristics of transistors can change gradually with long-term exposure to protons and electrons [4]. Space-bound electronics are tested for the maximum amount of radiation the device can accumulate before it cannot be used reliably. The amount of radiation a deployed system will endure is dependent on the orbit and the mission duration. For a low earth orbit (LEO) 100 kRads should be sufficient for several years of reliable operation, which is adequate for Virtex family devices.

B. Single-Event Effects

There are four primary forms of SEE that SRAM-based devices are concerned about: SEL, SET, SEU, and SEFI. While there are a handful of SEE types that can damage to a device, SEL is the predominant concern. The remaining three discussed in this paper are not destructive, but can make fault-tolerant computation challenging. These phenomena are discussed below.

1) Single-Event Latchup: Single-event latchup or *latchup* is a primary concern for many spacecraft designers. SEL is a radiation-induced form of latchup that occurs in transistor devices due to naturally occurring parasitic transistors that occur between the wells and the substrate. Devices that latchup are often not used in space, because even non-latchup, high current events on devices can cause latent damage [5]. For this reason, Altera's product lines have been avoided, since they latchup when exposed to heavy ions [6]. Xilinx products do not latchup in either protons or heavy ions.

2) Single-Event Transients: Single-event transients or *transients* are common in many semiconductor circuits. With this phenomena the ionizing particle causes a transient current state and could change an intermediate processing value, if registered. For SRAM-based FPGAs the current understanding is that SETs are possible, but are either not observable due to SEUs or are indistinguishable from SEUs in the user flip-flops.

3) Single-Event Upsets and Single-Event Functional Interrupts: As stated above, SEUs (or *upsets*) and SEFIs are the primary phenomena for these devices. While they should not actively hurt the device, they do complicate fault-tolerant computing. For space-based applications, devices are tested for the SEU response to heavy ions and protons. This response is called the *bit cross-section*, which is a measure of the per-bit sensitive area to the particular radiation source and has the units of cm^2/bit. SEFIs have a similar cross-section, but are usually reported on a per device basis. SEU and SEFI cross-sections have an *onset threshold* and a *saturation cross-section*. The onset threshold indicates the lowest energy or energy equivalent needed to cause an SEU or a SEFI. The saturation cross-section indicates the maximum sensitivity to the radiation source. With modern technology, though, many devices have an below 1 MeV-cm^2/mg for heavy ions. The on-set threshold for protons is low enough to make testing difficult. The SEU saturation cross-section often does not saturate in modern devices due to the presence of multiple-bit upsets (MBUs) [7].

SEU and SEFI cross-sections are found experimentally using particle accelerators. Table I has a list of SEU bit cross-sections and SEFI device cross-sections for 63.3 or 65 MeV protons and Figure 1 shows the SEU bit cross-sections for heavy ions for Virtex family devices. Note that the SEFI device cross-sections from Table I appear to be on the same scale as the SEU bit cross-sections, but in reality when the SEU cross-sections are scaled to device cross-sections the SEU cross-sections are several orders of magnitude larger than the SEFI device cross-sections. While the SEU bit cross-section is very small, each device has millions of bits. The SEFI device cross-section reflects the accurate reasoning that only 10-1000 bits are usually responsible for each SEFI state. These cross-section values are used with orbit prediction tools, such as CREME96 [8], to determine the expected on-orbit error rate for a given device.

Device	Energy (MeV)	σ_{bit} (cm^2/bit)	σ_{SEFI} (cm^2/device)
XCV1000	63.3	$1.32x10^{-14}$	$\approx 7.1x10^{-13}$ (config SEFI)
XC2V1000	63.3	$2.10x10^{-14}$	$9.46x10^{-13}$
XC4VLX25	63.3	$1.08x10^{-14}$	Unknown
XC5VLX50	65.0	$7.56x10^{-14}$	Unknown

TABLE I
Bit Cross-Section for SEUs and Device Saturation Cross-section for SEFIs for Protons for Several Xilinx FPGAs [7]

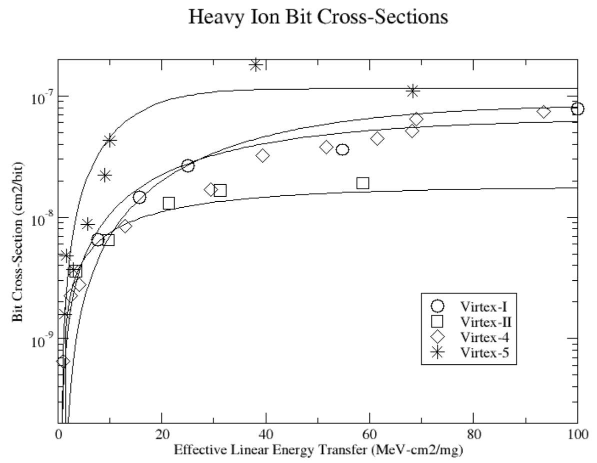


Fig. 1. Heavy Ion Bit Cross Sections for Virtex Family Devices [9].

III. Failure modes from SEUs and SEFIs

FPGAs have many SEU-induced failure modes that conventional ASICs circuits do not have. For example, by changing one configuration bit, a LUT resource may no longer operate as a simple LUT, a wire might not connect the same two endpoints, or an input may suddenly be coming from somewhere else. There are many possible ways to categorize the failure modes. One could do a broad classification of routing errors, logic errors, user memory errors, and architectural errors. For this paper, eleven specific failure modes are discussed: mux select, PIP short, PIP open, buffer off, buffer on, LUT value change, control bit change, user flip-flop, BlockRAM, half-latches, and the power network. Under this classification, SEFIs are classified as architectural errors and a number of SEFI modes are discussed in that section.

A. Routing Errors

In Virtex family FPGAs, the routing network largely consists of multiplexers (muxes), programmable interconnect points (PIPs), and buffers. All circuit inputs and outputs are multiplexed. In the older devices the programmable interconnect used PIPs to connect routes, whereas the newer devices use muxes. Finally, buffers are use through out the device to connect wires. The rest of this section will be devoted to discussing the failure modes for muxes, PIPs and buffers.

1) Multiplexers: Multiplexers used for routing are very sensitive to SEUs because any change in their select lines will cause a different routing configuration. An example mux select failure is shown in Figure 2.

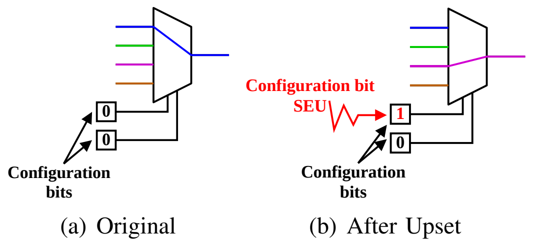


Fig. 2. Mux Select Failure Example

2) Programmable Interconnect Points: In older Virtex technology the routing network used PIPs. A PIP is a pass transistor between two wires that can either be on or off. Thus, the wires are either connected or not connected. PIPs can cause a few different kinds of SEU-induced failures. The first, shown in Figure 3(b), is called a PIP short failure, where two wires with different functions in the design are shorted together. A PIP short can produce contention, causing output errors and increased power consumption. The second type of PIP failure is shown in Figure 3(d) and is called a PIP open. This occurs when a PIP, normally turned on in the design, effectively breaks a wire into two pieces. A PIP open causes an interruption in the flow of information from one part of the design to another.

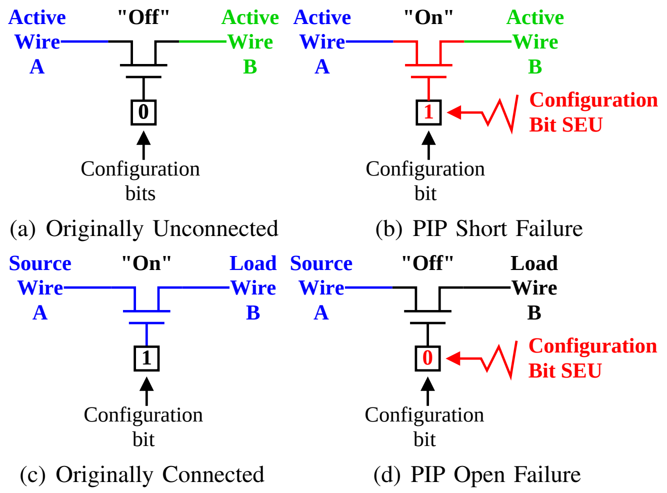
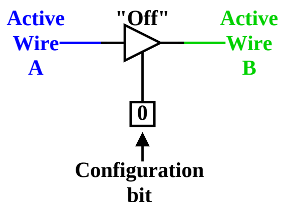
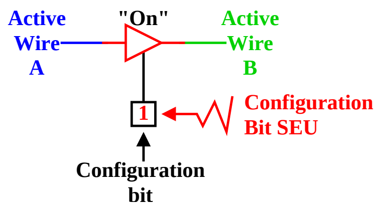


Fig. 3. PIP Failure Mode Examples

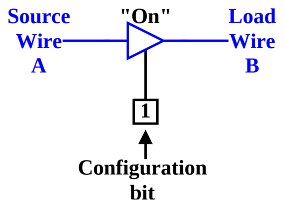
3) Buffers: The final component of the Virtex routing network considered here is the buffer. A buffer in this context is a driver which can either be turned on or off. As shown in Figure 4, the buffer has two failure modes associated with it and they are very similar to the PIP failures. The main difference here is that instead of a pass transistor, the failure is being caused by an active driver and, therefore, is unidirectional, in a sense. With a PIP failure, it is quite possible that errors can be caused on both sides of the PIP, but with a buffer failure only the output is affected. Buffers usually are placed on the outputs of some multiplexers and on bi-directional wires.



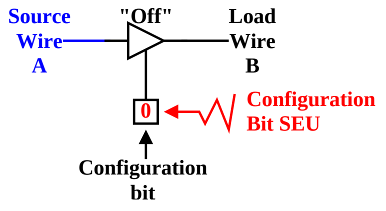
(a) Originally Unconnected



(b) Buffer Short Failure



(c) Original Connected

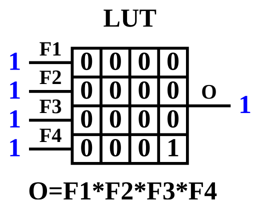


(d) Buffer Open Failure

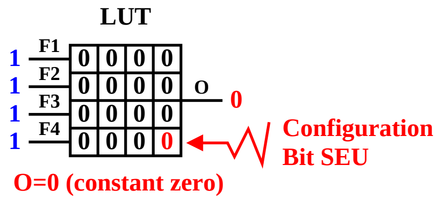
Fig. 4. Buffer Failure Mode Examples

B. *Logic Errors*

There are two types of logic errors: LUT value changes and control bit changes. The Virtex family FPGAs use lookup tables to generate most logic functions, so a change in the values stored in a LUT would impact the logic function implemented therein. This failure mode could cause constant or intermittent output errors depending on the inputs to the circuit and which part of the logic function is impacted. An example is shown in Figure 5. Here the LUT implements a 4-input AND function. If the one bit that defines the "true" condition is upset, the result is a constant-zero function. For most inputs, the output of the function would still be correct, however, there is the one case that would cause problems.



(a) Original 4-input AND Function



(b) Upset LUT Function (Constant "0")

Fig. 5. LUT Upset Example

In contrast to LUT value changes, control bit changes generally cause errors for all, or almost all, possible circuit inputs. The CLBs and IOBs use quite a few control bits to determine miscellaneous functionality. Figure 6 shows a partial schematic of a CLB. Bits *V*, *E*, *F*, and *G* are called programmable inversion bits. An upset to one of these will cause the value carried on that particular wire to be inverted, likely resulting in a circuit error when the value is used. The *T* bits, on the other hand, determine whether the LUT in this CLB performs as a LUT, a 16x1 dual-ported RAM, a part of a 32x1 RAM, or as a programmable shift register. If a LUT suddenly turns into a shift register, output errors will likely result. Other control bits determine such things as the electrical standard used in off-chip I/O and whether a storage element is a flip-flop or a latch.

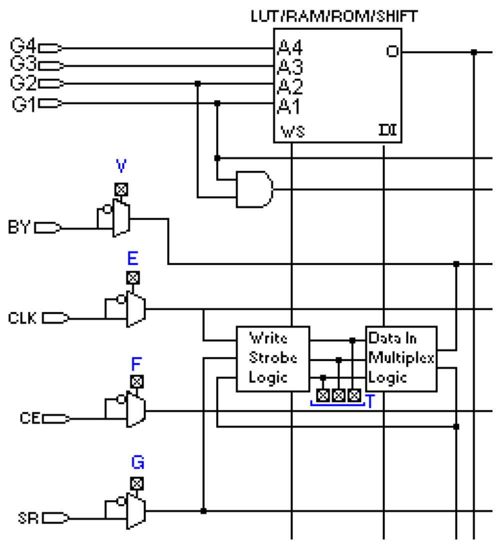


Fig. 6. Control Bit Examples [10]

C. *User Memory Errors*

SEUs can also affect user memory, such as flip-flops or user SRAM. Upsets that occur in flip-flops are not detectable as changes in the bitstream. Upsets in user SRAM bight be discernible through changes in the bitstream, but are not usually detected in this manner since it is hard to predict the value of a memory whose values are dynamic. Further, it is not generally possible to read the contents of the memory while it is actively being used in a circuit without the possibility of affecting its content. Upsets in these resources can introduce bad data into a circuit.

D. *Architectural Errors*

Architectural errors are those upsets that occur in the control elements of the FPGA, such as the configuration circuit, the JTAG TAP controller, and reset control.

Architectural errors are not directly observable and can usually only be measured indirectly through an upset "signature". For example, an SEU in the configuration control circuit changes many of the configuration bits at once, creating a very noticeable effect. Below is a discussion of three different architectural SEUs: half-latches errors, power network errors and SEFIs.

1) Half-Latch Errors: Xilinx Virtex and later FPGAs use half-latches (see Figure 7) to generate constant zero and one logic values used internally by FPGA designs [11]. Half-latches are widely used in designs to drive inputs to I/O, logic, RAM, clocking, and other resources. By using half-latches to create constant values, more expensive logic resources, such as look-up tables (LUTs), are not needed. Unfortunately, half-latches are susceptible to SEUs (see Figure 8). Even worse, half-latches are not directly initialized or controlled with programming data, which makes their state hard to observe and modify. Furthermore, only full reconfiguration re-initializes half-latches and processes that use on-line reconfiguration for repair, such as scrubbing, are unable to reset them. In Virtex-I, half-latches can be upset and often would retain this state for many seconds or even longer. In Virtex-II, the half-latches would generally return to their intended state in much less than a few seconds.

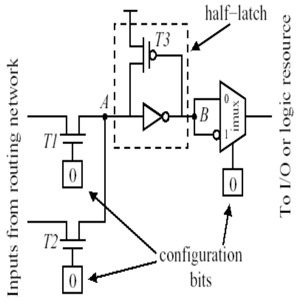


Fig. 7. Xilinx half-latch [11].

2) Single-Event Functional Interrupts: Internal to the device there are a handful of configuration and control registers that can be affected by SEUs. SEUs in these registers can affect the ability of the device to function properly, and these SEUs are classified as SEFIs. Examples of SEFIs include JTAG TAP controller upsets, SelectMAP controller upsets, and configuration control logic upsets. Below, a few of these SEFIs are discussed.

A SEFI in the JTAG controller, whether it is being actively used or not, can move the device into an undesirable state. Compounding the problem, the Virtex device does not make the reset pin (TRST) available to the user, denying designers the ability to hold the JTAG controller in reset while it is deployed.

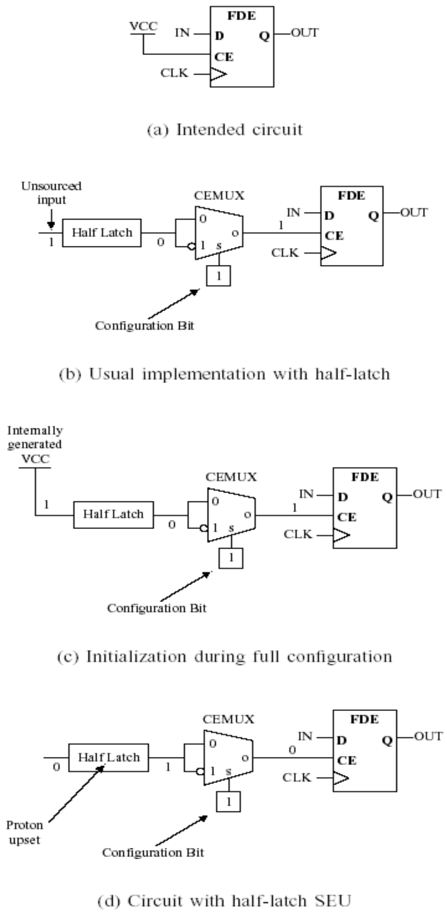


Fig. 8. Results of a Xilinx half-latch SEU [11].

SEFIs within the programmable SelectMAP interface configuration pins will result in a malfunctioning interface. Since the interface can no longer be accessed reliably, reading and writing to the device should be suspended until fixed. The corrupted interface could return bad values on a read and corrupt the configuration on a write. Additionally, the control registers and internal state of the configuration interface can be affected by SEUs causing problematic side-effects if not addressed. Sometimes these particular SEFIs will require a complete reconfiguration of the part to obtain a good state once again.

The configuration control state machine of a Virtex family device is vulnerable to the power-on reset (POR) SEFI. In this case, the device behaves as if $\overline{PROGRAM}$ has been asserted — its configuration is cleared and the $DONE$ pin is driven low.

IV. Mitigation and Repair Methods

The discussion of mitigation will concentrate on SEUs. For a discussion of SEFI mitigation, this paper [12] will be useful. Two predominant methods are used to handle radiation-induced upsets: TMR to mask errors and on-line reconfiguration (*scrubbing*) to correct errors. While both TMR and scrubbing seem simple in theory, in practice doing both correct is difficult. Therefore, the authors suggest engaging Xilinx's help with both tasks, especially scrubbing.

A. Triple Modular Redundancy (TMR)

Conventional TMR in its simplest form involves triplicating logic modules and using voters to arbitrate between them. In principle, when any of the redundant modules fail, the other two modules continue to operate correctly, and the voter outputs the correct value. The recommended approach to implement TMR on SRAM-based FPGAs is to fully triplicate all logic (modules and voters) and signals (input, output, clock, and reset). Since internal voters are implemented in LUTs, they need to be triplicated, otherwise they are a single points of failure. Recent work [13] has shown that protecting the clock and reset trees is of the utmost priority, since these trees will have to grow accordingly to support the triplicated flip flops. If these guidelines are followed all designs are guaranteed to withstand one single-bit upset in the routing, logic or user memory resources without noticeable output errors. TMR will not necessarily protect the user design from problems with half-latches, the power network or SEFIs. The first two problems are addressed below. There is no solution for internally mitigating SEFIs for single chip TMR implementations. If SEFIs are expected to become an availability issue, then triplicating the design across three devices is necessary.

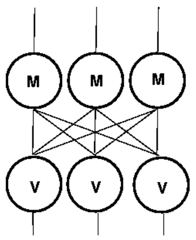


Fig. 9. The recommended implementation of TMR functional with triplicated modules (M) and voters (V).

Applying TMR techniques to user's circuits is not simple and can be error-prone, leading to unprotected cross-section in the resulting circuits. LANL's Scalable Tool for Reliable Circuits (STARC) can be used to help designers find unprotected cross-section in their designs [13]. The potential issues with implementing TMR on these devices are three-fold: problems with the circuit design, design constraints, and architectural influences on the circuit design. These issues are presented below.

The first issue is the design of the TMR-protected circuit. Since much of the design process depends on synthesis tools that are attempting to minimize the area and maximize the speed of the user circuit, even the most careful descriptions of TMR-protected circuits are often undermined by synthesis tools removing all or some of the redundant logic. To circumvent issues with the synthesis tools, the recommended approach for applying TMR to circuit descriptions is to use one of the two tools (BLTMR [14] and XTMR [15]) that automatically apply TMR to post-synthesis circuit representations of FPGA designs.

The second issue regards design constraints. Since these devices can be pin- and area-constrained, designers are sometimes unable to implement a fully triplicated design. In particular, not being able to triplicate input, output, clock or reset signals is common and SEUs in the input/output blocks, routing, global clock network, and flip-flops could cause errors to manifest across all three logic modules. While it possible to triplicate some of these signals internally on the device, an unprotected cross-section still exists.

Designers might also find themselves constrained by the device's size, and are unable to fully triplicate the circuit logic. The BLTMR tool addresses this problem by balancing the need to protect the most essential parts of the design and meeting area constraints by applying TMR partially to the circuit. BLTMR gives highest priority to sub-circuits that may reach a persistent error state due to feedback, since error recovery may require external intervention [14]. In cases where TMR has only been partially applied to the circuit, unprotected cross-section exists.

The third issue is the implementation of the circuit on the architecture. As discussed earlier, half-latches are susceptible to SEUs and must be removed from the design. In the Virtex-II, the power network replaces some of the role of half-latches in providing logical constants to the design. Since the power network is load balanced by the tools, redundant logic in TMR-protected designs could share the same power network, introducing potential single points of failure into the design. Further complicating the issue, the power network is implemented with LUTs and the routing network, making it susceptible to all of those failure modes. Both BLTMR and XTMR tools address issues with logical constants by using input/output pins to provide these constant logic values in a TMR-aware manner. LANL's RadDRC tool can be used on Virtex-I devices to remove half-latches from designs as well. Placement of the design on the device can also cause problems. Redundant logic can be placed in close proximity, making TMR vulnerable to

MBUs [9].

B. Scrubbing

Scrubbing uses on-line reconfiguration to reload the FPGAs configuration bitstream, removing any SEUs that may have accumulated in the bitstream between scrubs. Once the SEU is removed from the device, the three TMR modules should be able to resynchronize. In the past, so called "blind" scrubbing (i.e., without readbacks) was done more frequently. Over the years, though, the control logic and registers necessary for scrubbing have grown larger and SEUs in these areas during scrubbing has been observed to cause high current states [12].

The algorithm outlined below is one recommended scrubbing algorithm. Readback the configuration data. While the readback is being performed, check each frame against it's "golden" (un-irradiated) CRC value. If the CRC value does not match, scrub the frame. Since the device is not scrubbed end-to-end, SEUs in the configuration circuitry should only have a localized effect.

For the Virtex-I and Virtex-II bitstreams, the portion scrubbed includes all of the configuration data for the global clock (GCLK), BRAM interconnect, IOB, and CLB configuration data. In these devices, LUT RAM, SRL16s and BRAM could not be scrubbed effectively, since either reading back their stored data would interrupt circuit operation or predicting the correct value to scrub into memories with dynamic values is difficult. In the Virtex-4 and Virtex-5, there is logic on the devices that can be used to skip LUT RAM resources when the programming data is being read or written, making it possible to use the scrubbing approach mentioned above in the presence of user LUT RAM and SRL16s. In cases where the BRAM is being used as a ROM, scrubbing is necessary and a BRAM scrubber is available from Xilinx [16]. In general, implementing a good scrubber is of the utmost priority to successful space missions. While Xilinx provides some guidance [17], Xilinx must be engaged to guarantee success.

V. Conclusion

In summary, this paper presented a number of possible radiation-induced faults in SRAM-based FPGAs. These faults could affect the routing, logic, and user memory resources in a user design implemented on the device. We have also presented how errors in the underlying architecture could cause faults in the user designs or make the device inoperable until reset. Finally, we presented how TMR could be used to mask SEUs in the user design and how on-line reconfiguration could be used to remove SEUs from the device. Used in conjunction these two methods should provide fault-tolerant computing for space-based applications.

References

[1] P. Graham, M. Caffrey, M. Wirthlin, E. Johnson, and N. Rollins, "Consequences and categories of SRAM FPGA configuration SEUs," in *Proceeding of the Military and Aerospace Programmable Logic Devices International Conference(MAPLD)*, Washington, DC, September 2003, submitted.

[2] P. Graham, H. Quinn, and J. Moore, *Xilinx Virtex FPGA Design Guide for Space*. on web at www.fpgamac.com, 2008.

[3] A. Holmes-Siedle and L. Adams, *Handbook of Radiation Effects*. Oxford University Press, 2002.

[4] H. Barnaby, *Evolving Issues for the Application of Microelectronics in Space*, ser. Short Course Notebook for the Nuclear and Radiation Effects Conference. IEEE, 2005, ch. Total Dose Effects in Modern Integrated Circuit Technology.

[5] H. N. Becker, T. F. Miyahira, and A. H. Johnston, "Latent damage in cmos devices from single-event latchup." *IEEE transactions on nuclear science*, vol. 49, no. 6, pp. 3009 – 3015, 2002.

[6] S. L. Clark, K. Avery, and R. Parker, "Tid and see testing results of altera cyclone field programmable gate array." *IEEE Radiation Effects Data Workshop*, pp. 88 – 90, 2004.

[7] H. Quinn, P. Graham, J. Krone, M. Caffrey, and S. Rezgui, "Radiation-induced multi-bit upsets in SRAM-based FPGAs," *IEEE Transactions on Nuclear Science*, vol. 52, no. 6, pp. 2455 – 2461, December 2005.

[8] "Cosmic ray effects on micro-electronics (1996 revision)," on web https://creme96.nrl.navy.mil/.

[9] H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey, and K. Lundgreen, "Domain crossing errors: Limitations on single device triple-modular redundancy circuits in Xilinx FPGAs," *IEEE Transactions on Nuclear Science*, vol. 54, no. 6, pp. 2037 – 43, 2007.

[10] "http://www.xilinx.com/labs/projects/jbits/."

[11] P. Graham, M. Caffrey, M. Wirthlin, D. E. Johnson, and N. Rollins, "SEU mitigation for half-latches in xilinx virtex fpgas," in *Proceedings of the IEEE 2003 Nuclear and Space Radiation Effects Conference*, IEEE. Monterey, CA: IEEE, July 2003, p. TBA, accepted.

[12] C. W. Tseng, C. Carmichael, and G. Swift, "Optimizing configuration management for SEU mitigation in xilinx virte-4 FPGA and self-scrubbing," http://nepp.nasa.gov/mafa/talks/MAFA07_20_Allen.pdf.

[13] H. Quinn, P. Graham, and B. Pratt, "An automated approach to estimating hardness assurance issues in triple-modular redundancy circuits in xilinx FPGAs," Los Alamos National Laboratory, Tech. Rep., 2008.

[14] K. Morgan, M. Caffrey, P. Graham, E. Johnson, B. Pratt, and M. Wirthlin, "SEU-induced persistent error propagation in FPGAs," *IEEE Transactions on Nuclear Science*, vol. 52, no. 6, pp. 2438 – 45, 2005.

[15] "Xilinx TMRTool user guide," on web: http://www.xilinx.com/products/milaero/ug156.pdf.

[16] G. Miller, C. Carmichael, and Jet Propulsion Labs, "Single-event upset mitigation for xilinx FPGA block memories: Application note 962," on web: http://www.xilinx.com, 2004.

[17] C. Carmichael, M. Caffrey, and A. Salazar, "Correcting single-event upsets through virtex partial configuration: Application note 216," on web: http://www.xilinx.com, 2000.