

# High-Reliability FPGA-Based Systems: Space, High-Energy Physics, and Beyond

Michael Wirthlin

NSF Center for High-Performance Reconfigurable Computing (CHREC)

Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602

## ABSTRACT

Field-Programmable Gate Arrays (FPGAs) have been shown to provide high computational density and efficiency for many computing applications by allowing circuits to be customized to any application of interest. FPGAs also support programmability by allowing the circuit to be changed at a later time through reconfiguration. There is great interest in exploiting these benefits in space and other radiation environments. FPGAs, however, are very sensitive to radiation and great care must be taken to properly address the effects of radiation in FPGA-based systems. This paper will highlight the effects of radiation on FPGA-based systems and summarize the challenges in deploying FPGAs in such environments. Several well-known mitigation methods will be described and the unique ability of FPGAs to customize the system for improved reliability will be discussed. Finally, two case studies summarizing successful deployment of FPGAs in radiation environments will be presented.

## I. INTRODUCTION

The introduction of field programmable gate arrays (FPGAs) and other forms reconfigurable logic has led to the development of many novel forms of programmable computing architectures. These computing approaches rely on the reconfigurability of these devices to *customize* the logic structures, computing units, communication networks, and memory structure to an application-specific computation [1]. By customizing the computing architecture to the application, the computation can be performed with higher efficiency than using traditional general-purpose processors. When sufficient reconfigurable resources are available, this efficiency can provide much higher performance than other general-purpose architectures. Many examples have demonstrated the increased performance and efficiency of reconfigurable systems over traditional programmable processor approaches [2].

Reconfigurability is not free and comes at a high cost – reconfigurable systems require a large amount of silicon to support in-field reconfigurability. A large amount of static memory is needed to store the configuration of the FPGA. This configuration memory is used to define the function of the programmable logic units, routing switches, internal memory and other special-purpose structures. In large FPGAs, the configuration data used to define the reconfigurable circuit can exceed 100 Mb. The configurable logic memory cells

needed to implement a custom function increase the silicon area needed to implement the function when compared to an application-specific integrated circuit (ASIC). When compared to ASIC alternatives, this reconfigurability can increase the silicon area of a computation by as much  $35\times$  and increase the critical path delay of the circuit by  $3\text{--}4\times$  [3].

Like all static memory cells, SRAM-based configuration cells used in FPGAs are susceptible to data corruption from high-energy radiation. Because of the growing number of configuration cells in FPGAs, upsets within the configuration memory are increasing. Upsets within the configuration memory of reconfigurable systems are especially troublesome since these cells specify the operation of the reconfigurable fabric. These upsets can change the operation of the logic, routing, and other functions of the device.

There is growing interest in using FPGAs in space and other extreme environments where high-energy radiation is more common than on earth [4]. The ability to reconfigure logic resources is extremely valuable in a spacecraft as such reconfiguration supports the ability to upgrade satellite electronics, exploit in-system reconfiguration, or create a design that avoids permanent failures in a device. There is also great interest in using FPGAs within high-energy physics (HEP) experiments for readout electronics or high-bandwidth data transfer [5]. Further, there is growing interest in using FPGAs in high-altitude environments (which are more susceptible to radiation) and in high-reliability terrestrial systems [6].

This interest has motivated the investigation of design techniques and system integration strategies for using FPGAs reliably in these environments. The success of these techniques has facilitated the adoption of FPGAs in a variety of radiation environments. Many satellites have successfully integrated FPGAs into the electronics payload and are currently operational in space. FPGAs are reliably used within several high-energy physics experiments and are being adopted in a number of high-altitude and high-reliability situations. It is likely that FPGAs will be used more often as these techniques become more widespread, tools are developed to simplify their implementation, and FPGA devices are improved to better address soft-errors.

This paper will summarize the effects of radiation on FPGAs and discuss the implications of these effects in complex systems. The most successful mitigation methods to address these issues will be presented along with methods unique to FPGAs and reconfigurable systems. Examples of the use of FPGAs in harsh environments will be discussed including space systems and high-energy physics.

This work was supported by the IUCRC Program of the National Science Foundation under Grant No. 1265957.

Digital Object Identifier: 10.1109/JPROC.2017.2404212

## II. EFFECTS OF RADIATION ON FPGAS

For the purposes of this discussion, radiation is the transmission of energy through atomic and sub-atomic particles with very high kinetic energy. Radiation is a natural phenomena and is generated from materials on earth, the sun, and other cosmic sources. Fortunately, most radiation directed to the earth from cosmic sources is filtered by the atmosphere limiting the energy and flux of radiation we experience on the ground. Radiation has a negative impact on semiconductor circuits and high levels of radiation can damage semiconductor systems. This section will discuss the effects of radiation on integrated circuits and describe on how these radiation effects apply to modern FPGAs.

### A. Effects of Radiation on Semiconductor Circuits

Radiation has a long-term, damaging effect on electronic components. 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. In addition, high energy particles can damage semiconductor materials by displacing atoms in the lattice. Such displacement damage also changes the electrical parameters of the device. Ultimately, radiation will cause functional failures within the device. The amount of radiation dose that a device can tolerate before failing to meet published parameter specifications is called total ionizing dose (TID) [7].

In addition to long-term effects, the radiation from individual high-energy particles can cause immediate effects within the device that are collectively called “single-event effects” (SEE) [8]. There are a variety of single-event effects that must be considered before using a device in a radiation environment. A brief summary of the most relevant effects for FPGAs are described below.

a) *Single-Event Latchup (SEL)*: Single-event latch-up is a potentially destructive condition in which a single charged particle induces a parasitic p-n-p-n structure (equivalent to a silicon controlled rectifier or SCR). This structure produces a low-impedance path between power and ground resulting in large currents flowing through the parasitic bi-polar transistors. In many cases, this current is high enough to destroy the device. Once the parasitic transistors enter a latchup state, the positive feedback of the device will hold the device in latchup until power is removed or the device is destroyed. Any device considered for high radiation environments must be tested for latchup as a latchup event is a potentially catastrophic failure.

b) *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 transferred from one node to another. If the charge is greater than a device-specific critical charge,  $Q_{crit}$ , this charge transfer can change the voltage level of critical nodes within the memory cell such that the modified voltage level reflects the opposite state of the cell (i.e., changing a logic '1' to a logic '0' or a logic '0' to a logic '1'). The feedback nature of static latches will preserve this new

value and the original value will be lost. Unlike SEL, single-event upsets do not cause any permanent damage within the device.

c) *Single-Event Transient (SET)*: A single-event transient occurs when a radiation induced transient voltage pulse is generated in a digital circuit. SETs cause unwanted glitches that propagate through combinational circuitry. If the temporary glitch is latched into a sequential circuit (flip-flop or latch), it will appear as a single-event upset. Like SEUs, SETs do not cause any permanent damage and introduce unwanted transient behavior into a circuit.

d) *Single-Event Functional Interrupt (SEFI)*: A single-event functional interrupt is a broad term referring to a single-event that causes a significant change in the functional operation of a device. SEFIs are usually caused by changing the internal state of important control registers within a device that control device-level functionality. Examples of SEFIs include device-level reset, lock-up, initiation of power-on reset, initiation of unique operating mode (brown out, sleep mode, etc.), and device shutdown. Fortunately, the circuit area devoted to these structures is very small and the probability of causing such SEFIs is thus very low. SEFIs are not permanent or destructive and can be resolved by repowering the device and placing it in its initial state.

### B. Single-Event Effects and FPGAs

FPGAs suffer the same problems with respect to radiation as other semiconductor devices. The effects of radiation on an FPGA depend in a large part on the mechanism used by the FPGA to store the configuration data. For the purposes of this discussion, FPGAs will be classified into three different categories based on the technology used to store the configuration. These categories include antifuse, flash, and SRAM-based FPGAs.

1) *Antifuse FPGAs*: Antifuse FPGAs use one-time programmable fuses to permanently set the state of each FPGA configuration bit. A device programmer is used to program all configuration fuses from a configuration file. These FPGAs are non-volatile and the state of the configuration memory is retained even when the device is not powered. The advantage of antifuse FPGAs include the smaller size of the configuration memory cells and the relative immunity to single-event effects. The primary disadvantage of antifuse FPGA is that the configuration data cannot be changed once the fuses have been programmed. This prevents the user from updating the device in the field or using them in reconfigurable computing applications.

From a single-event effects perspective, antifuse FPGAs are generally the most reliable type of FPGA [9]. Because the configuration cells are made from passive, pre-programmed fuses, they generally immune to single-event effects. The primary radiation concern for antifuse FPGAs are SEUs and SETs within the user flip-flops. From a radiation effects perspective, antifuse FPGAs look much like application-specific integrated circuits (ASIC) whose circuit configuration are static and do not change in the presence of ionizing radiation. A number of reliable antifuse FPGAs are available for high

radiation environments and have been successfully deployed in a number of space systems.

2) *Flash FPGAs*: As the name implies, flash FPGAs use flash memory cells to set the state of the FPGA configuration memory. Flash memory cells use an electrically isolated floating gate to store the state of the memory cell (electrons can be trapped in or removed from the floating gate to set the cell state). Flash-based FPGAs can be programmed in-system and are non-volatile. Although flash FPGAs are in-system reprogrammable, there is a limit in the number of times they can be reprogrammed which may not be suitable for reconfigurable systems requiring frequent reconfiguration.

The flash cell is generally immune to SEUs and thus the configuration memory of a flash-based FPGA is protected from SEUs [10]. Like antifuse FPGAs, the primary concern for single-event effects are SETs and SEUs within the user flip-flops and block memories. Flash memory, however, is more sensitive to total ionizing dose due to damage of the internal charge pump (needed for high-voltage erasing and writing) and for degradation in the threshold voltage ( $V_t$ ). Thus flash-based FPGAs have a lower TID threshold than conventional SRAM based FPGAs and can only be used in radiation environments for a limited amount of time. A new family of radiation tolerant FLASH-based FPGAs has been developed with a higher TID [11].

3) *SRAM FPGAs*: SRAM-based FPGAs use static memory cells to store the internal FPGA configuration. These static memory cells require power to store the configuration state and must be programmed from external memory after the FPGA has powered up. SRAM-based FPGAs are volatile and lose their configuration when power is removed. Although SRAM cells consume more silicon area and require more power than antifuse or flash cells, they can be reprogrammed an unlimited number of times to facilitate frequent reconfiguration and adaptation.

The primary reliability concern for SRAM FPGAs operating in a radiation environment are SEUs within the configuration memory. Because the configuration memory cells are made using standard static memory techniques, these cells are susceptible to radiation-induced upsets. Since the vast majority of system state within an SRAM FPGA is configuration memory, upsets in the configuration memory are the most common failure mechanism of these FPGAs in a radiation environment.

Although all three types of FPGAs are used in radiation environments, this paper will focus on SRAM FPGAs and the unique mitigation methods needed to operate in a radiation environment. Many of the techniques described apply to all FPGA types.

### III. RADIATION ENVIRONMENTS

With increased interest in exploiting programmable logic in radiation environments, researchers have investigated the suitability of commercially available FPGAs in radiation environments such as space [12], [13]. This section will review several important radiation environments in which FPGAs are particularly useful including space environments, high-energy physics experiments, and even terrestrial environments.

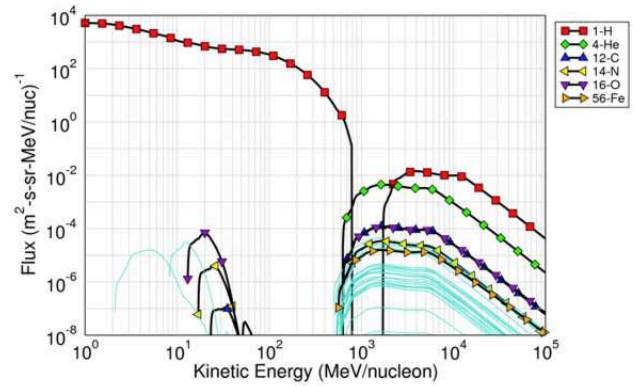


Fig. 1. Estimated Flux as a Function of Ion and Energy for Cibola Flight Experiment Orbit (Figure 8 in [15]).

#### A. Space Environments

Reprogrammable FPGAs have been used in space environments for many years [14]. The use of FPGAs within modern spacecraft is motivated by the growing computational needs associated with modern sensors used in spacecraft. Because of the tremendous amount of data generated by modern sensors, it is no longer possible to send all sensor data back to earth for processing. Today, much of that processing (including data compression) must be done on the spacecraft.

FPGAs provide an attractive solution for many of the computationally intensive on-board processing tasks used by modern spacecraft. Modern FPGAs contain a tremendous amount of computational resources due to the large number of logic elements, internal memory, and dedicated digital signal processors (DSP). In addition, modern FPGAs contain multi-gigabit serial I/O links to facilitate the transfer of real-time sensor data within the spacecraft. FPGAs can perform many of these computations more efficiently than programmable processors since the datapath and control of these algorithms can be customized in the form of dedicated circuits (i.e., custom, reconfigurable computing). Alternatives to FPGAs include radiation hardened ASICs (which are not reprogrammable), commercial GPUs or radiation hardened programmable processors (which are orders of magnitude slower than commercially available processors).

The primary challenge for using FPGAs in spacecraft is addressing the effects of radiation on FPGA operation. The radiation experienced by satellite electronics in space is generated from several different sources including protons and heavy ions emitted by the sun (i.e., solar particles), galactic cosmic rays, and particles trapped in the earth's magnetic field. The radiation environment in space can be quantified by plotting the flux ( $particle/cm^2/s$ ) of a particular particle as a function of particle energy ( $MeV$ ). This environment is complex and includes a large spectrum of particles of different mass each with a different energy spectrum. Figure 1 demonstrates the radiation environment for a low-earth orbit in "Solar Quiet Conditions" [15]. Dedicated software is often used to perform complex calculations to estimate the rate at which electronic circuits will upset for specific space environments.

The radiation environment in space is well characterized and varies considerably based on the altitude, inclination, and eccentricity of the satellite orbit. This environment also depends on the location of the spacecraft within the orbit – some locations such as the South Atlantic Anomaly (SAA) have a very high concentration of trapped protons and can generate upsets that are an order of magnitude higher than other regions within the orbit. The radiation environment is also heavily dependent on the transient solar weather and the eleven year solar cycle. Large solar events like solar flares can increase the instantaneous flux of radiation by several orders of magnitude. One of the challenges of using FPGAs in space is estimating the upset rate of the device in the particular space environment and anticipating and handling infrequent but harsh space weather events like solar flares and coronal mass ejections (CME).

### B. High-Energy Physics Experiments

FPGAs are increasingly being used within high-energy physics experiments (HEP) within the readout electronics. High-energy physics experiments study the properties and interactions of the fundamental particles of nature (quarks, leptons, muons, etc.). HEP experiments use particle accelerators to accelerate charged particles to very high speeds (and thus high energy) and then directing two beams of high speed particles to form a particle collision. This collision generates a number of byproducts that are studied to learn more about the subatomic structure and fundamental laws of nature.

An important part of HEP experiments are the detectors that measure the byproducts of high energy particle collisions. A variety of particle detectors have been developed over the years and are can be used to measure the energy, direction, spin, charge, etc. of a variety of particles. High speed electronics are used within detectors to capture the particle data and send this data to external computer systems for post-processing (readout electronics). FPGAs are often used in the detectors of HEP experiments for interfacing with sensors (usually ADCs), performing simple calculations, measuring sub-nano second time differences, and streaming the data outside of the experiment (high-speed serial I/O). Thousands of FPGAs can be used within large HEP experiments, such as the ATLAS, CMS, ALICE, and LHCb experiments that operate within the Large Hadron Collider (LHC).

An intense radiation field is generated from the high-energy particle collisions within HEP experiments. The actual radiation environment depends heavily on the experiment itself and on the location within the experiment (in general, the radiation field is higher as you move closer to the particle collision). At some locations within the experiment such as the inner detector, the radiation field is very high and FPGAs could not be used. For many locations, however, the radiation environment is modest and FPGAs are appropriate with proper SEU mitigation methods. Figure 2 demonstrates the estimated radiation environment within the ATLAS Liquid Argon Calorimeter. As seen in this figure, the environment contains a variety of atomic and sub-atomic particles at a wide range of energy values.

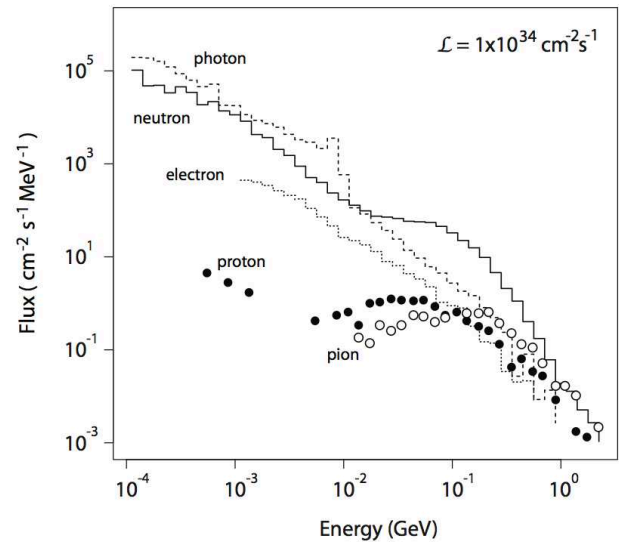


Fig. 2. Estimated Radiation Field for ATLAS Liquid Argon Crate (simulation).

Unlike space environments, the radiation flux within HEP experiments is relatively constant. This makes it easier to predict FPGA upsets and create appropriate mitigation methods. In addition, HEP experiments typically revolve around periodic pulses inherent to the particle accelerator. This periodic nature can be exploited by the application-specific SEU mitigation methods. Unlike the space environment, these experiments undergo relatively frequent shutdowns where the electronics can be inspected, replaced, and upgraded.

### C. Terrestrial Environments

Another important radiation environment for FPGAs is the terrestrial environment here on earth. The terrestrial earth environment is usually not considered a “harsh” radiation environment like space or HEP experiments. Electronic circuits operating in terrestrial environments, however, are exposed to radiation that can negatively impact their operation. While upsets within FPGAs due to terrestrial radiation are rare, they do occur and are easily detectable using conventional error detection techniques. These impact of these upsets are especially important in high-reliable applications or systems in which there are a large number of FPGAs [6].

The radiation received on earth comes a variety of sources including cosmic radiation and terrestrial sources. Cosmic radiation is derived from sources outside the solar system and interacts with atoms in the atmosphere. These interactions produce secondary radiation (mostly high-energy neutrons) that interact with electronic systems. Figure 3 shows the flux of cosmic-ray induced neutrons at sea-level. Terrestrial sources of radiation include naturally occurring materials in the earth such as soil, rocks, water, and the air. Most naturally occurring terrestrial radiation is relatively low energy and has little impact on electronic systems. The exception to this is the radiation sometimes found within the packaging materials used to manufacture semiconductor systems. With proper

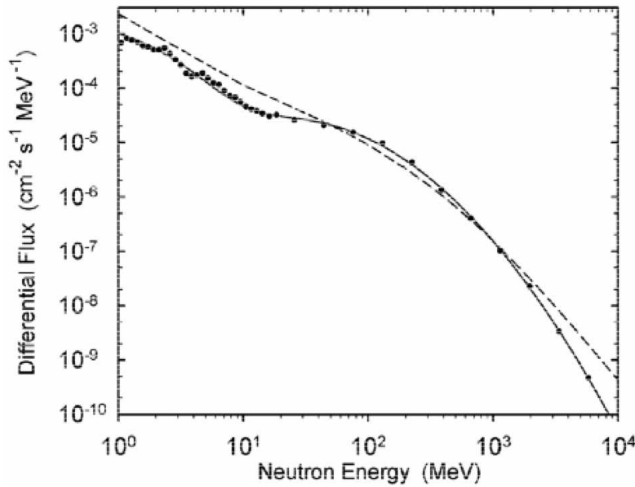


Fig. 3. Cosmic-Ray Induced Neutron Flux as a Function of Neutron Energy (Sea Level) [16].

manufacturing techniques, the effects of naturally occurring radiation on electronic systems can be eliminated.

The dose rate of cosmic radiation varies throughout the world and depends on the magnetic field and altitude of the location. The higher the altitude of the system, the higher the terrestrial soft error rate. High altitude applications of FPGAs (including avionics) and high reliable systems such as communication systems, power systems, medical systems, automotive, and industrial applications must carefully estimate the effects of SEUs and provide proper error detection and correction capabilities. Industrial standards have been created for measuring and reporting these errors in semiconductor devices [16].

#### IV. FPGA ARCHITECTURE VULNERABILITIES

Modern FPGAs are very complex devices that contain a wide variety of heterogeneous resources. Modern FPGAs include programmable look-up tables for logic, user flip-flops for storing internal state, internal block memory, programmable processors, I/O resources including high-speed multi-gigabit transceivers, digital signal processing (DSP) blocks, analog to digital converters, PCIe interfaces, clock management resources, etc. The effects of radiation on each of these internal resources varies and an understanding of these effects can only be understood with directed radiation testing [17], [18].

This section will summarize the primary radiation effects on several of the most important architectural components of the FPGA. Specifically, this section will discuss the effects of radiation on the configuration memory, block memory, user flip-flops, and internal proprietary state. To aid this discussion, the Xilinx 7 Series Kintex 325-T FPGA will be used to highlight these issues [19]. Table I summarizes the most important user-accessible internal state of this device. As shown in this table, this device contains almost 92 million bits of known state that is susceptible to radiation-induced upsets. The impact of upsets within each of these memory types will be described below.

Memory Type	# Bits	%
Configuration	72,868,672	79.3%
Block RAM	18,661,568	20.3%
Distributed RAM†	4,096,000	4.5%
User Flip-Flops	407,600	.44%
Total	91,937,840	100%

† Distributed RAM are a subset of the configuration memory.

TABLE I  
MEMORY BITS WITHIN THE KINTEX-7 325T DEVICE

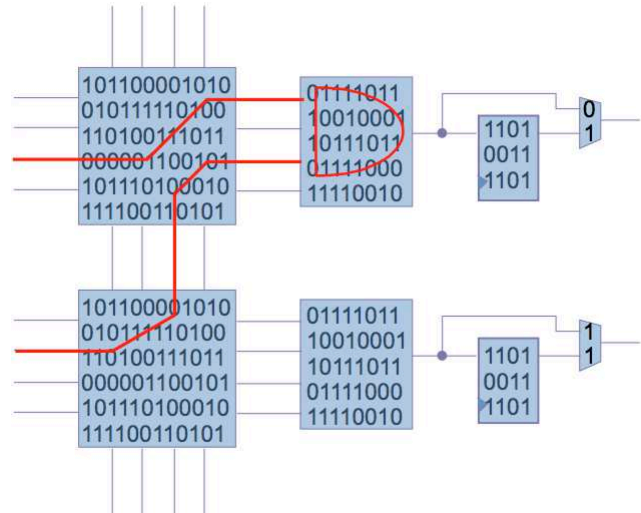


Fig. 4. Configuration Bits Used to Specify Logic and Routing.

#### A. Configuration Memory

As shown in Table I, 79% of the memory cells within the Kintex 325T device are devoted to configuration memory. As described earlier, 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. Figure 4 depicts the relationship between the configuration memory and the circuit configured on the device. The data in the routing blocks and look-up tables of this figure form the configuration to implement a two-input “AND” gate. Because these configuration cells are implemented as static memory, a different logic function and routing organization can be implemented on the device by reconfiguring the 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. 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. Figure 5 demonstrates what may happen to the two-input “AND” gate in Figure 4 when upsets occur in the configuration memory. The first configuration upset is a change in the routing configuration data and disconnects one input of the “AND” gate (i.e., an open). The second configuration upset is a change in the look-up table contents

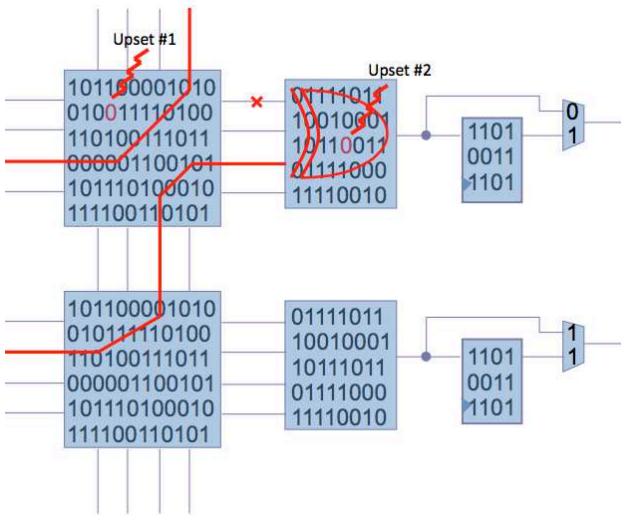


Fig. 5. Upset Configuration Bits Change the Logic and Routing.

of the “AND” gate and modifies the operation logic function (it no longer performs the “AND” function). In both cases, upsets in the configuration memory change the behavior of the circuit so that the circuit no longer performs the function intended by the circuit designer.

Not all upsets to the configuration memory will cause the design to deviate from its intended function. Many of the configuration bits associated with logic and routing are “unused” by the circuit and thus upsets in these bits have no effect on the design. Configuration bits that cause a design to fail when upset are called “sensitive” configuration bits and those that do not impact the design are called “insensitive”. Some have suggested that, at most, 10% of the configuration bits are sensitive for a given design. The actual sensitivity rate varies from design to design and fault-injection is frequently performed to estimate this sensitivity [20].

### B. Block Memory

All modern FPGAs provide access to a large number of internal memories to support a variety of operations and computations. The distributed nature of this memory provides a large amount of internal memory bandwidth to support high-performance computing and memory buffering. As seen in Table I, internal Block RAM memory makes up the second largest component of internal FPGA state. Modern FPGAs provide many blocks of internal memory to perform traditional random access memory functions such as data storage, buffering, FIFO, etc. The 7 Series family of FPGAs include dual-ported Block RAM memories that provide 36 Kbits of randomly accessible memory that can be configured in a variety of ways.

Dense static memory such as the Block RAM memory is also susceptible to radiation-induced SEUs. Upsets in the Block memories will introduce data errors into the FPGA circuit. The impact of such data errors on the system behavior will depend on how the data is being used. If the upset occurs within an unused memory word, the upset will have no impact on the system. Most FPGA designs do not use all of the

available Block memories and many designs do not use the full memory space within a single block RAM. If the upset occurs within a memory word that is used by the system, errors can propagate throughout the system and cause a variety of undesirable effects.

### C. User Flip-Flops

An important architectural component of all FPGAs is the user programmable flip-flops. Most FPGA designs use many flip-flops to implement common sequential logic circuits such as registers, state machines, counters, delay lines, synchronizers, and small memory buffers. In non-SRAM based FPGAs (i.e., antifuse and Flash FPGAs), it is possible to experience “upsets” within the flip-flop due to single-event transients (SET) [21]. Transient glitches in intermediate signals and logic gates due to single-event effects may be latched into a flip-flop as an incorrect value (it can be difficult to distinguish between a direct single-event upset in the flip-flop and an upset caused by an SET).

Although flip-flops represent the smallest portion of internal state for SRAM FPGAs (see Table I), flip-flops typically contain the most important state of the circuit and upsets in the flip-flops often cause significant disruption to the circuit’s operation. For example, upsets within the flip-flops of a state machine will cause the state machine to enter a new state. Even worse, such upsets may send the state machine into a non-existent deadlock state. Other problems that may occur with flip-flop upsets include changes in the internal counter values and upsets within the data elements of registers or delay elements.

### D. Internal Proprietary State

All FPGAs contain internal state to manage the internal operation and configuration of the FPGA. Much of this state is not visible to the user but plays a very important function in the operation of the FPGA. This state is quite small and thus not very sensitive to radiation when compared to block memory or user flip flops. However, upsets in this internal state often lead to very troublesome results for the FPGA. These type of failures are often grouped together into a single category called Single-Event Functional Interrupts (SEFI). SEFI events cause global, system-wide functional interruptions and cannot be resolved with common localized mitigation methods. For example, test results from radiation testing of an FPGA found the “power-on reset” SEFI that caused the power-on reset circuitry to operate [22].

## V. MITIGATION OF FPGA SINGLE-EVENT EFFECTS

The strong interest in using FPGAs in radiation environments like space and high-energy physics has motivated the study of techniques that mitigate the effects of soft-errors on SRAM FPGAs. Some techniques involve hardware changes to the FPGA fabric or device architecture. For example, the Xilinx V5QV FPGA is a modified version of the commercial Virtex 5 family of FPGAs that incorporates radiation-hardened by design (RHBD) techniques to protect the configuration

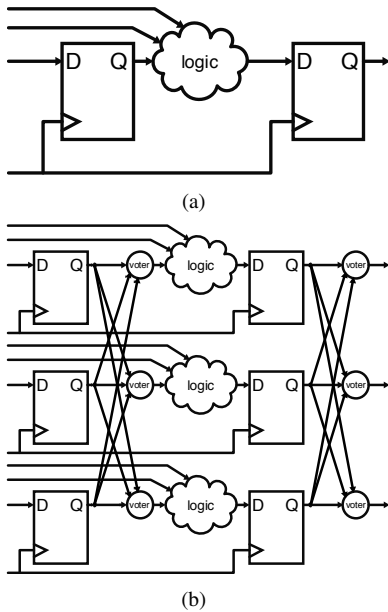


Fig. 6. (a) A typical circuit with flip-flops and logic and (b) the same system operating with triple modular redundancy.

memory cell from single-event upsets [23]. Other techniques can be implemented by the FPGA designer to detect and provide appropriate response to these single-event effects. This section will summarize a number of techniques used by FPGA designers and system integrators to address single-event effects within commercial-grade FPGAs. Many FPGA-based systems have been deployed in radiation environments using these and other SEU mitigation techniques (Section VI will highlight two of these systems).

#### A. Hardware Redundancy

To mask the effects of upsets in the FPGA configuration memory, temporal or structural redundancy can be applied to the system. Temporal redundancy involves the replication of a computation or logic function in time to mitigate failures that occur during one of the redundant computations. Structural redundancy involves the replication of selected circuit structures to remove single-point failures. Failures in the circuit can be masked by performing the logic or computing function in more than one circuit location.

The most common form of structural redundancy is to apply triple-modular redundancy (TMR) [24]. As shown in Figure 6, TMR involves the triplication of all circuit resources and the addition of majority voters at the appropriate circuit outputs. In principle, TMR allows a circuit to operate with any single fault – the presence of a single fault will not cause any operational problem as there are two working copies of the circuit that operate correctly and selected by the majority voter. A variety of FPGA-specific TMR implementation methods have been introduced to address these issues [25]. Artificial injection of faults into the configuration memory can be used to verify the effectiveness of TMR and related techniques.

In addition to TMR, there are a variety of other methods that can be used to mitigate the effects of FPGA SEUs [26].

These techniques include state machine encoding [27], special-purpose placement and routing [28], design diversity redundancy [29], reduced precision redundancy [30], duplication with compare [31], temporal redundancy, alternative logic systems, dynamic redundancy, etc. Many of these techniques take advantage of application-specific properties of the circuit operating within the FPGA to achieve more efficient SEU mitigation than TMR. There is an active research community investigating new techniques to provide efficient SEU mitigation.

#### B. Configuration Scrubbing

To prevent the build-up of upsets within the configuration memory, the upsets in the configuration memory can be repaired through scrubbing. Configuration scrubbing is the process of repairing upsets within the configuration memory by writing the correct configuration data back into the configuration memory [32]. Configuration scrubbing is much like scrubbing used in conventional memory systems to preserve the integrity of the memory. Scrubbing involves a continuous process of reading the current state of the configuration memory and writing correct results back into the memory. There are many methods of configuration scrubbing including blind scrubbing, readback scrubbing, and hybrid scrubbing [33].

Configuration scrubbing is not sufficient to prevent configuration upsets from temporarily changing the behavior of an FPGA circuit. Even with high-speed configuration scrubbing there is a finite period of time between the occurrence of an upset in the configuration memory and the repair of the upset through scrubbing. During this time period the circuit may be operating incorrectly, entering into incorrect states, or produce incorrect results. For the highest reliability, scrubbing needs to be coupled with a structural redundancy technique like TMR. The reliability of TMR with scrubbing can be modeled as a Markov model of a redundant system with repair. These models suggest significant improvements in reliability over either technique applied by itself [34]. Scrubbing significantly reduces the probability that multiple upsets will occur between scrubbing and “break” TMR. Although expensive, the most successful SEU mitigation approach for FPGAs operating in a radiation environment includes a combination of TMR and frequent configuration scrubbing [35].

#### C. Error Correction Coding

Most FPGA families provide built-in error correction coding (ECC) to support the ability for detecting and correcting errors within Block RAM memories. Additional parity or check bits are provided within the block memories to facilitate the detection and/or correction of individual words. For the Xilinx 7 Series family of FPGAs, this ECC support provides single-bit correction and double error detection (SECDDED) when the memory is configured to operate with a 72-bit data width (64 data bits + 8 check bits). When individual 72-bit words are read from memory, the ECC logic will correct single-bit errors and detect double-bit errors. ECC protection is often used in terrestrial applications when data integrity is essential.

While built-in error correction protects individual memory words from single upsets during a read operation, it does not correct the internal state of the memory. To avoid the build-up of upsets within the same word, some form of memory scrubbing should be employed for data that resides in internal block memory for an extended period of time. For radiation environments like space and HEP, block memories are sometimes triplicated to avoid single-point failures in the memory (such as stuck-at upsets on the write-enable line). Triplication of memories along with memory scrubbing also provides a robust memory protection strategy.

#### D. Flip-Flop Mitigation

For all FPGA types, the state of the flip-flops can be protected from SEUs by exploiting redundancy such as TMR [36]. For SRAM-based FPGAs, the input forming logic associated with the flip-flops should be protected with TMR as the configuration bits associated with the input forming logic are much more likely to be upset than the flip-flops themselves. For non-SRAM based FPGAs, it is essential that voters are applied within a feedback loop of the flip-flops to support resynchronization of the flip-flops after an upset [14]. Some radiation hardened FPGAs implement individual flip-flops as three internal flip-flops with dedicated voters to avoid the need for user-based SEU mitigation [37].

SET filters can be manually added to a design to reduce the occurrence of SET induced upsets. Some high reliability FPGAs provide the ability to automatically adding SET filters on selected signal paths. Other forms of mitigation for flip-flops include safe state machine encoding, alternative logic systems, self-checking circuitry, and masking logic.

#### E. System-Level Mitigation

A variety of system-level mitigation techniques can also be used to improve the reliability of FPGA-based systems in radiation environments. For example, some systems apply TMR at the system-level by triplicating three identical FPGAs that implement the same function. External voters are used to determine the correct FPGA output. While costly, this approach will tolerate any device-level failures like a SEFI or inadvertent reconfiguration. Other system-level mitigation approaches include the use of external watch-dog timers, radiation-hardened system monitors, and system checkpointing.

For many FPGA-based systems, SEFI detection is necessary. SEFI detection involves the monitoring (including self-monitoring) of key health parameters of the FPGA and reporting any anomalous behavior. SEFI mitigation is specific to the particular FPGA family and often requires accelerated radiation testing to identify. SEFI mitigation methods include special-purpose scrubbing modes, periodic self-reset, or scrubbing of internal configuration registers.

## VI. HIGH RELIABLE FPGA SYSTEMS

Many FPGA-based systems have been successfully deployed in radiation environments by exploiting mitigation

techniques such as the ones described earlier. This section will highlight two such systems and summarize important lessons learned.

#### A. Space Systems

FPGAs have been successfully deployed in many spacecraft using a variety of SEU mitigation techniques including those described in this paper. Unfortunately, most of the FPGA-based spacecraft are either classified or proprietary making it difficult to learn how FPGAs operate in a real space environment. This section will highlight an FPGA-based platform that has been deployed in space and has published reports on the effectiveness of the SEU mitigation techniques.

The Los Alamos National Laboratory (LANL) Cibola Flight Experiment is one of the first satellites to successfully deploy and use SRAM-based FPGAs in a radiation environment [15]. Nine reconfigurable Xilinx Virtex FPGAs are included in the payload to process sampled radio spectrum data for ionospheric and lightning studies. Reconfiguration is used to apply different algorithms based on the sensitivity of the system. The reconfigurable FPGAs provide significant higher processing power than was available at the time using radiation-hardened technology.

The reconfigurable computing (RCC) platform includes a radhard processor, radiation tolerant antifuse FPGA, flash memory, and three Virtex FPGAs. Configuration readback was performed using the antifuse FPGA and flash memory to identify and log configuration upsets. Upsets were repaired using configuration scrubbing. TMR was applied to a number of user designs to protect against configuration upsets and “half-latches”<sup>1</sup> were removed using a custom tool (RadDRC).

The satellite was launched in March of 2007 and has been operating ever since (six years of operation). The payload has experienced 2,649 configuration SEUs since launch for an average of .78 SEUs/device/day during solar quiet conditions and .47 SEUs/device/day for active solar conditions. The vast majority of upsets occur within the South Atlantic Anomaly (SAA). Since launch, CFE has experienced 11 MBUs and no SEFIs. The satellite continues to collect data and is an excellent example on how SRAM FPGAs can mitigate the effects of SEUs in space to perform high-performance computations.

#### B. High Energy Physics

One of the most successful deployment of reconfigurable FPGAs within a high-energy physics experiment is the ALICE TPC readout system [5]. The ALICE experiment is one of four experiments operating on the Large Hadron Collider at CERN. The purpose of this experiment is to study quark-gluon plasma which is believed to have existed soon after the Big Bang. The main tracking detector within ALICE is the Time Projection Chamber (TPC) and consists of several cylindrical shaped field cages. The TPC contains readout electronics to sample the 150 GBytes/s data generated by the detector and transfer the data out to the counting room.

<sup>1</sup>Half-latches are weak pull up circuits in the Virtex architecture that are susceptible to SEUs. Half-latches do not appear to be present in more recent FPGA architectures.



The ALICE TPC readout system contains 216 readout control units that each contain a Xilinx Virtex-II pro FPGAs. This FPGA-based system performs many important functions including compression, signal shaping, digital signal processing, and data buffering. The FPGA designs are highly utilized so TMR is not possible. In this system, upsets in the FPGA are monitored and logged but no mitigation is provided. If the SEU causes a crash on the readout control unit (RCU), it is reconfigured. Fault injection tests performed before deploying system suggested that only one out of 93 upsets in the configuration memory cause a system crash.

The ALICE TPC continually reads the configuration memory of the FPGA to identify configuration upsets. This read-back system takes 150 ms or 6.7 Hz. All SEUs were logged and correlated with the location of the detector providing a real-time upset detection throughout the system. Over 1600 upsets were logged during 2011 and a linear relationship between beam luminosity and SEU upset rates was measured. The FPGA proved very successful within the RCU and based on the success of this experiment, FPGAs will continue to be used in future ALICE TPC upgrades.

## VII. CONCLUSION

The in-system reconfigurability, high logic density, and high-speed I/O of modern FPGAs make them ideal for spacecraft and high-energy physics experiments. FPGAs, however, are susceptible to radiation-induced single-event effects. A variety of well-known and proven SEU mitigation techniques have been applied to FPGA-based systems and successfully demonstrated in radiation environments. The success of these techniques has facilitated FPGAs being used space applications, high-energy physics experiments, and high-reliable terrestrial applications.

As the density of FPGAs continues to increase and as more system-level circuitry is integrated within FPGA-based devices, there will be greater interest in deploying FPGAs in radiation environments. Additional research will need to investigate new methods for reliably integrating these complex FPGA-based system-on-chip (SOC) systems in radiation environments. Based on the success of using commercial SRAM-based FPGAs in radiation environments and the positive results in fault-tolerant processor design, it is likely that the these hybrid SOC devices can also be deployed reliably in radiation environments.

## REFERENCES

- [1] Scout Hauck and André DeHon. *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*. Morgan Kaufmann, 2007.
- [2] Katherine Compton and Scott Hauck. Reconfigurable computing: A survey of systems and software. *ACM Comput. Surv.*, 34(2):171–210, June 2002.
- [3] Ian Kuon and J. Rose. Measuring the gap between FPGAs and ASICs. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(2):203–215, Feb 2007.
- [4] 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 Algorithms (ERSA)*, pages 49–53. CSREA Press, June 2002.
- [5] Ketil Red. *Single Event Upsets in SRAM FPGA based readout electronics for the Time Projection Chamber in the ALICE experiment*. PhD thesis, The University of Bergen, 2009.

- [6] H. Quinn and P. Graham. Terrestrial-based radiation upsets: a cautionary tale. In *Field-Programmable Custom Computing Machines, 2005. FCCM 2005. 13th Annual IEEE Symposium on*, pages 193–202, April 2005.
- [7] H.J. Barnaby. Total-ionizing-dose effects in modern CMOS technologies. *Nuclear Science, IEEE Transactions on*, 53(6):3103–3121, Dec 2006.
- [8] P.E. Dodd and L.W. Massengill. Basic mechanisms and modeling of single-event upset in digital microelectronics. *Nuclear Science, IEEE Transactions on*, 50(3):583–602, June 2003.
- [9] J. McCollum. ASIC versus antifuse FPGA reliability. In *Aerospace conference, 2009 IEEE*, pages 1–11, March 2009.
- [10] John McCollum. Radiation tolerant flash FPGA. US Patent (6324102), November 2001.
- [11] Ken O’Neill. Technology roadmap for digital space flight products. Technical report, Microsemi Corporation, November 2013.
- [12] 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.
- [13] 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.
- [14] R. Katz, R. Barto, P. McKerracher, B. Carkhuff, and B. Koga. SEU Hardening of field programmable gate arrays (FPGAs) for space applications and device characterization. *IEEE Transactions on Nuclear Science*, 41(6):2179–2186, 1994.
- [15] H. Quinn, D. Roussel-Dupre, M. Caffrey, P. Graham, M. Wirthlin, K. Morgan, T. Salazar, T. Nelson, E. Johnson, J. Johnson, B. Bratt, N. Rollins, and J. Krone. The Cibola flight experiment. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 2014. To be published.
- [16] *Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices*, October 2006. JESD89.
- [17] Gary Swift and Gregory Allen. Virtex-5QV Static SEU Characterization Summary. Technical report, NASA Jet Propulsion Laboratory (JPL), May 17 2013. <http://parts.jpl.nasa.gov/organization/group-5144/radiation-effects-in-fpgas/xilinx>.
- [18] Gary Swift, Carl Carmichael, Gregory Allen, Roberto Monreal, George Madias, and Eric Miller. Virtex-5QV Architectural Features SEU Characterization Summary. Technical report, NASA Jet Propulsion Laboratory (JPL) and Xilinx, August 23 2013. <http://parts.jpl.nasa.gov/organization/group-5144/radiation-effects-in-fpgas/xilinx>.
- [19] 7 series FPGAs overview. Technical report, Xilinx, February 2014. DS180 (v1.15).
- [20] 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.
- [21] M. Berg, J. J. Wang, R. Ladbury, S. Buchner, H. Kim, J. Howard, K. Label, A. Phan, T. Irwin, and M. Friedlich. An analysis of single event upset dependencies on high frequency and architectural implementations within Actel RTAX-S family field programmable gate arrays. *IEEE TNS*, 53(6):3569–3574, December 2006.
- [22] C.C. Yui, G.M. Swift, C. Carmichael, R. Koga, and J.S. George. SEU mitigation testing of Xilinx Virtex II FPGAs. In *Radiation Effects Data Workshop, 2003. IEEE*, pages 92–97, July 2003.
- [23] Radiation-Hardened, Space-Grade Virtex-5QV Family Overview, March 2012. DS192 (v1.3).
- [24] Carl Carmichael. Triple module redundancy design techniques for Virtex FPGAs. Technical report, Xilinx Corporation, November 1, 2001. XAPP197 (v1.0).
- [25] Jonathan M. Johnson. Synchronization voter insertion algorithms for FPGA designs using triple modular redundancy. Master’s thesis, Brigham Young University, 2010.
- [26] K.S. Morgan, D.L. McMurtrey, B.H. Pratt, and M.J. Wirthlin. A comparison of TMR with alternative fault-tolerant design techniques for FPGAs. *Nuclear Science, IEEE Transactions on*, 54(6):2065–2072, Dec 2007.
- [27] S. Niranjan and J.F. Frenzel. A comparison of fault-tolerant state machine architectures for space-borne electronics. *Reliability, IEEE Transactions on*, 45(1):109–113, Mar 1996.

- [28] L. Sterpone and M. Violante. A new reliability-oriented place and route algorithm for SRAM-based FPGAs. *Computers, IEEE Transactions on*, 55(6):732–744, June 2006.
- [29] L.A. Tambara, F.L. Kastensmidt, J.R. Azambuja, E. Chielle, F. Almeida, G. Nazar, P. Rech, C. Frost, and M.S. Lubaszewski. Evaluating the effectiveness of a diversity TMR scheme under neutrons. In *Radiation and Its Effects on Components and Systems (RADECS), 2013 14th European Conference on*, pages 1–5, Sept 2013.
- [30] Joshua Snodgrass. *Low-power fault tolerance for spacecraft FPGA-based numerical computing*. PhD thesis, Naval Postgraduate School, 2006.
- [31] J. Johnson, W. Howes, M. Wirthlin, D.L. McMurtrey, M. Caffrey, P. Graham, and K. Morgan. Using duplication with compare for on-line error detection in FPGA-based designs. In *Aerospace Conference, 2008 IEEE*, pages 1–11, March 2008.
- [32] 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).
- [33] Melanie Berg, C Poivey, D Petrick, D Espinosa, Austin Lesea, KA LaBel, M Friendlich, H Kim, and Anthony Phan. Effectiveness of internal versus external SEU scrubbing mitigation strategies in a Xilinx FPGA: Design, test, and analysis. *Nuclear Science, IEEE Transactions on*, 55(4):2259–2266, 2008.
- [34] Daniel McMurtrey, Keith Morgan, Brian Pratt, and Michael Wirthlin. Estimating TMR reliability on FPGAs using Markov models. Technical Report 149, Brigham Young University, December 2008. <http://scholarsarchive.byu.edu/facpub/149>.
- [35] Nathaniel Rollins, Michael Wirthlin, Michael Caffrey, and Paul Graham. Evaluating TMR techniques in the presence of single event upsets. In *Proceedings of the 6th Annual International Conference on Military and Aerospace Programmable Logic Devices (MAPLD)*, page P63, Washington, D.C., September 2003. NASA Office of Logic Design, AIAA.
- [36] 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 Systems (RADECS 2001)*, 2001.
- [37] J. McCollum, R. Lambertson, J. Ranweera, J. Moriarta, J.J. Want, F. Hawley, and A. Kundu. Reliability of antifuse-based field programmable gate arrays for military and aerospace applications. In *MAPLD*. Actel Corporation, 2001.