

NOISE IMPACT OF SINGLE-EVENT UPSETS ON AN FPGA-BASED DIGITAL FILTER

Brian H. Pratt, Michael J. Wirthlin

NSF Center for High Performance
Reconfigurable Computing (CHREC)
Dept. of Elec. & Comp. Engineering
Brigham Young University
Provo, UT 84604 USA
brianpratt@byu.net, wirthlin@ee.byu.edu

Michael Caffrey, Paul Graham, Keith Morgan

ISR-3 Space Data Systems
Los Alamos National Laboratory
Los Alamos, NM 87545 USA
mpc, grahamp, morgank@lanl.gov

ABSTRACT

Field-programmable gate arrays are well-suited to DSP and digital communications applications. SRAM-based FPGAs, however, are susceptible to radiation-induced single-event upsets (SEUs) when deployed in space environments. These effects are often handled with the area and power-intensive TMR mitigation technique. This paper evaluates the effects of SEUs in the FPGA configuration memory as noise in a digital filter, showing that many SEUs in a digital communications system cause effects that could be considered *noise* rather than circuit failure. Since DSP and digital communications applications are designed to withstand certain types of noise, SEU mitigation techniques that are less costly than TMR may be applicable. This could result in large savings in area and power when implementing a reliable system. Our experiments show that, of the SEUs that affected the digital filter with a 20 dB SNR input signal, less than 14% caused an SNR loss of more than 1 dB at the output.

1. INTRODUCTION

Field-programmable gate arrays (FPGAs) are known for their high processing throughput, reconfigurability, and often lower total cost as compared to application-specific integrated circuits (ASICs). These characteristics make them desirable for use in digital signal processing (DSP) and digital communications systems. DSP systems are heavily used in space systems and, because of these favorable characteristics, FPGAs are often considered for these applications [1].

SRAM-based FPGAs, however, are susceptible to the effects of radiation-induced single-event upsets (SEUs). The SRAM memory elements which hold the configuration memory of the FPGA can be corrupted by high-energy par-

ticles common in space environments. Since the configuration memory determines the circuit implemented by the FPGA, the actual function of the FPGA can be altered in addition to the data in the user memory. In general, the configuration memory is the more critical set since it makes up the majority (>90%) of the on-chip memory [2].

These SEU-induced faults can be repaired by restoring the original configuration of the FPGA. A method called *configuration scrubbing* is used to periodically refresh the configuration memory to repair these faults soon after the faults are discovered. The errors induced by the faults, however, cannot be repaired by scrubbing and must be handled separately.

Mitigating the effects of SEUs is important for FPGA-based systems operating in radiation environments. The most popular mitigation technique is triple-modular redundancy (TMR). TMR has been singled out because of its relatively simple architecture and its general applicability. TMR involves triplicating the circuit modules that need to be protected from SEUs and adding majority voters at their outputs to decide what the correct output is from the three replicates. Assuming only one fault at a time, TMR is very effective at correcting errors introduced by SEUs [3].

Despite its effectiveness, TMR is very expensive and costs at least 3x in terms of hardware [4]. Further, systems that employ TMR are slower than their non-redundant counterparts. Because of this high area and timing cost, many people are investigating other methods of SEU mitigation [5]. Some of these methods, like TMR, are generic techniques, while other methods benefit from using knowledge of the circuit design in question to focus on reliability issues specific to that system.

DSP and digital communications applications are candidates for alternative mitigation strategies. Some systems in these categories may not need full TMR protection since they are designed to tolerate certain types of errors and noise in the first place. A demodulator in a digital communications

This work was supported by the I/UCRC Program of the National Science Foundation under Grant No. 0801876. Approved for public release by Los Alamos National Laboratory under LA-UR-09-03658; distribution is unlimited.

system, for example, must be designed to tolerate a certain amount of thermal noise added to the received signal. This noise decreases the signal-to-noise ratio (SNR) of the received signal and thus the performance of the demodulator. These built-in error-correcting abilities may also be able to correct errors introduced by SEUs in an FPGA-based DSP system.

This paper will evaluate the effects of SEUs on a simple digital communications system implemented in an FPGA. We have performed fault injection experiments to determine the effect of SEUs on the matched filter of a simple demodulator system. This paper will show that the effects of SEUs may often be viewed as noise introduced into the system. Due to the nature of the system, some of these errors may be handled gracefully without any explicit SEU mitigation techniques. Others may be protected against using mitigation techniques less costly than TMR, reducing the area and power overhead required for a reliable system.

2. RELATED WORK

In searching for alternatives to TMR, various authors have noted that reduced-cost mitigation techniques might be obtained by using knowledge of the system in question. These approaches, primarily targeting ASIC-based systems, have been called *algorithm-based fault tolerance* (ABFT) [6], *algorithmic soft error tolerance* (ASET) [8], and *system knowledge* [7].

Some authors have shown that the effects of soft errors in a DSP system can sometimes be viewed as noise. Several papers have examined soft errors produced in ASICs by deep-submicron (DSM) noise as well as those produced by using voltage overscaling (VOS) to reduce power [8]. Although we will make a similar analysis, the causes of the soft errors are distinct from those which are of main concern for SRAM FPGA systems. For example, the errors introduced by the VOS technique tend to be located in the most significant bits (MSBs) of a computation rather than the uniformly-distributed errors expected from radiation-induced upsets.

Others have published papers dealing with the effects of radiation-induced SEUs in ASIC-based DSP systems [7]. These papers focus on errors caused by upsets only in the memory elements of the systems, which is the dominant issue in ASIC technologies. In contrast, this paper will consider the effects of SEUs in any part of the FPGA configuration memory, which specifies the logic implemented in addition to the user memory.

3. NOISE IN DIGITAL COMMUNICATIONS SYSTEMS

As mentioned in Section 1, digital communications systems are designed to operate under adverse conditions. In order to

predict the performance of a communications system, mathematical models are developed that represent the most important characteristics of the signal transmission medium. The simplest and most common model used to represent a communications channel is the additive noise channel, illustrated in Fig. 1.

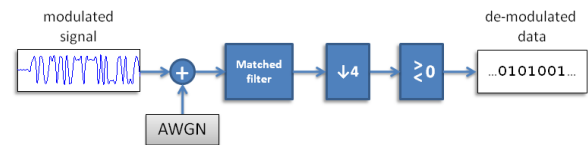


Fig. 1. Diagram of a simple binary phase-shift keying (BPSK) receiver circuit with additive white Gaussian noise (AWGN).

Digital communications systems are often optimized mathematically to correct errors introduced by additive Gaussian noise in order to combat thermal noise inherent in these systems. These built-in error-correction properties are critical to the performance of a communications system. These properties are those which we would like to exploit when considering an application-specific mitigation technique for digital communications circuits on FPGAs.

When designing an application-specific mitigation scheme, the measure of performance is critical. The traditional method of measuring the error-handling performance of a digital communications receiver is bit error rate (BER). BER is simply the ratio of incorrectly-decoded data bits to the total number of bits received. With a higher amount of Gaussian noise, the BER of the system increases. Fig. 2 shows the relationship between amount of Gaussian noise in a receiver system in terms of SNR and the BER at the output of the system. The plot shows that a lower SNR, corresponding to a stronger noise signal, results in a higher BER.

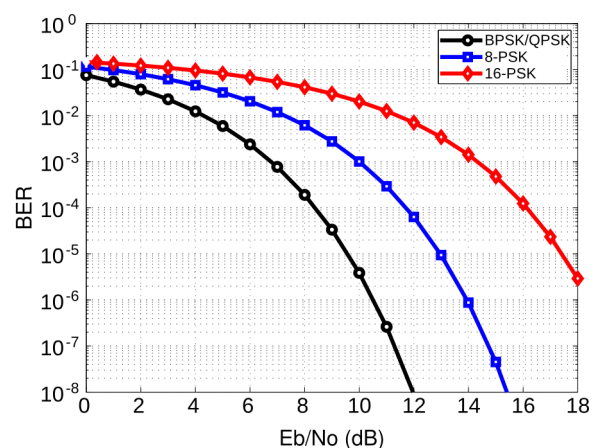


Fig. 2. Bit error rate (BER) curves for several M-PSK modulation schemes.

4. SEU-INDUCED NOISE

In addition to the thermal noise inherent in any digital communications system, FPGA-based systems in space environments must also deal with errors caused by SEUs. As mentioned in Section 1, SEUs in SRAM-based FPGAs corrupt both the user memory and the configuration memory of the device. This means that both the data being processed and the hardware doing the processing are vulnerable to these upsets. Whether it is the hardware or the data itself that is affected, the result is incorrect data being produced by the affected module.

In some sense, this incorrect data can be viewed as *noisy* data. The SEU that caused the fault in the design results in a corrupt data signal leaving the design. For a DSP system, we will call this type of corruption *SEU-induced noise*.

The dynamics of this type of “noise” are most likely distinct from the thermal noise most often dealt with in DSP and digital communications systems. As mentioned in Section 3, this noise is modeled as Gaussian noise, that is, the noise follows a Gaussian probability distribution. Since the mechanics of SEU-induced errors are decidedly different than those causing thermal noise, we do not expect SEU-induced noise to follow a strict Gaussian distribution. For example, upsets in the routing of the design, which controls how logic components are connected, are not expected to cause errors which strongly resemble Gaussian noise.

Although the SEU-induced noise in an FPGA system may not perfectly resemble Gaussian noise, the noise compensation characteristics of a DSP system may still be able to process the data affected by SEU-induced noise. If the effects of SEUs are similar, in some way, to Gaussian noise, the system may react with a similar increase in BER. We refer to these types of effects as *Gaussian-like*, since the circuitry designed to filter Gaussian noise is able to filter this type of SEU-induced noise as well.

If a significant percentage of SEUs cause Gaussian-like noise in an FPGA-based DSP application, we may be able to mitigate many of the adverse effects of SEUs at a much lower cost than TMR. Since this type of error will be compensated for by the DSP application, it may not be necessary to explicitly mitigate their effects. Instead, the mitigation scheme can focus on the more critical errors that are not handled by the built-in noise compensation of the application. These errors would include those that are not Gaussian-like as well as Gaussian-like, high-magnitude errors that are beyond the noise filtering capabilities of the DSP application in question.

5. EXPERIMENT METHODOLOGY

To evaluate the effects of SEUs on an FPGA-based DSP system, we performed fault injection on a simple DSP module to emulate radiation-induced upsets and observed the ef-

fects. The experiments considered only the configuration memory cells of the FPGA since these make up the vast majority (>90%) of the on-chip memory and thus the majority of the fault locations [2]. We did not investigate the effects of upsets within block memories, which were not used by our test design, nor those within the user flip-flops.

The module examined in these experiments is the matched filter of a binary phase-shift keying (BPSK) digital communications receiver. A simple BPSK receiver consists of a downsampler, a matched filter, and a decision block as shown in Fig. 1. The matched filter is the most complex of these components and is responsible for the majority of the error-handling of the system. The filter used for this experiment has the following properties:

- 49-tap FIR filter
- Square-root raised cosine (SRRC) pulse shape with 50% roll-off
- 16-bit fixed-point input (Q2.14 format)
- 18-bit fixed-point output (Q4.14 format)
- 15% of slices occupied on Virtex 1000 FPGA

By injecting faults into the configuration bitstream and running modulated data through the filter, we were able to measure the impact of each configuration bit in the FPGA being upset.

The BYU-LANL fault injection tool used has been described in detail in previous papers [9]. The fault injection tool’s hardware consists of three Virtex 1000 FPGAs: one for the design under test (DUT), one for the golden design, and one for data generation and comparison. In the standard configuration, a design is implemented on both the DUT and golden FPGAs. The third FPGA generates random data using an LFSR and passes it as inputs to the DUT and golden designs. The outputs of these two FPGAs are constantly compared, looking for differences. The fault injection tool’s software injects faults in the DUT FPGA by reconfiguring individual configuration bits. If a difference in output is observed after one of these injected faults, that configuration bit is marked as *sensitive*. The tool analyzes all of the 5,810,024 configuration bits, one by one, in a total of approximately 25 minutes.

Fig. 3 shows the steps involved in the experiments performed. The matched filter design was implemented on a Virtex 1000 FPGA. A 10,000-sample long sequence of random data was generated and fed through a modulator system in Matlab. We will refer to this input signal as the signal x . Gaussian noise was added to the modulated data, creating the signal $x + n_G$, which was then passed through the FPGA-based matched filter, H_{gold} . The output was recorded and stored as $y = H_{gold}(x + n_G)$, as shown in Fig. 3(a). The output from the noiseless case ($y_0 = H_{gold}(x)$), where

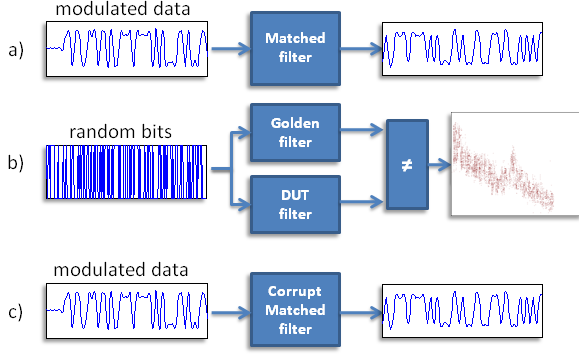


Fig. 3. Flow of the matched filter experiment.

$n_G = 0$) was the *golden* output that was compared against the data from step three.

Next, the fault injection tool was used to completely characterize the FPGA design by determining which configuration bits were *sensitive* to SEUs. This is illustrated in Fig. 3(b), where the diagram at the right illustrates the physical locations of the sensitive bits within the FPGA. It is important to note that the input to each version of the filter in this step was the default for the fault injection tool: a pseudorandom sequence of bits for each of the 16 input bits. In other words, the input sequence was not a modulated signal, but white noise, which is likely to stress the filter more than the modulated signal that it is designed to receive. This provides better coverage of the design.

This process discovered that the matched filter design in question utilizes 149,696 configuration bits (out of the total 5,810,024 available in the Virtex 1000 FPGA). Fig. 6(b) shows a plot of the physical location of each of these configuration bits in this FPGA. This is referred to as the *dynamic cross section* of this specific FPGA design. Only these sensitive bits were considered for the rest of the experiment.

The third step of the experiment determined the impact of sensitive SEUs on the filter as shown in Fig. 3(c). Each sensitive configuration bit, as discovered in step (b), was upset and the output recorded as y'_i , i referring to which bit was upset. The input to the corrupt filter was $x + n_G$, the same modulated signal from step (a). Each corrupt output signal could then be subtracted from the golden output signal to obtain the “noise” signal at the output of this particular corrupt filter, referred to as $n_i = y'_i - y$. In this way, we obtained a sample of the SEU-induced noise for every sensitive configuration bit in the matched filter design.

In order to obtain results for different types of noise environments, noise was added to the input signal and steps 1 and 3 were repeated for three cases: 20 dB, 10 dB, and 5 dB SNR at the input to the filter.

To analyze the results from this experiment, we have taken the individual corrupt output signals for each upset bit and calculated the loss in signal-to-noise ratio caused by

each upset. First, the SNR at the output of the golden filter was calculated as follows:

$$n_{\text{gold}} = y - y_0, \quad (1)$$

$$\text{gold_SNR_dB} = 10 * \log_{10} \left(\frac{\text{power}(y_0)}{\text{power}(n_{\text{gold}})} \right). \quad (2)$$

Then the SNR at the output of each corrupt filter (one for each configuration bit) was calculated:

$$n_i = y'_i - y, \quad (3)$$

$$\text{corr_SNR_dB}_i = 10 * \log_{10} \left(\frac{\text{power}(y_0)}{\text{power}(n'_i)} \right). \quad (4)$$

Finally, the difference of the two SNR values was taken for each corrupt output to show the difference in SNR caused by the injected fault:

$$\text{SNR_loss}_i = \text{gold_SNR_dB} - \text{corr_SNR_dB}_i. \quad (5)$$

In a standard communications system model with an AWGN channel, the loss in SNR can be used to estimate the change in BER of the system. Using Fig. 2, we can estimate the impact of a certain loss of SNR on BER. A loss in SNR corresponds to sliding up and left along one of the BER curves, resulting in an increase in BER.

6. EXPERIMENTAL RESULTS

The calculations described in the previous section resulted in a large number of SNR loss values: one for each *sensitive* configuration bit in the FIR filter design. Fig. 4 shows three cumulative distribution functions (CDF) of the loss in SNR due to each single configuration bit upset. The plot displays the results of the three experiments, each with a different amount of noise added to the input signal. The CDFs were created from 149,696 entries: one for each of the sensitive configuration bits (i.e. FPGA configuration bits actually utilized by the filter design).

As an example, for the 20 dB input SNR case, the plot shows that 95% of the sensitive configuration upsets caused less than 15 dB of SNR loss at the output of the filter. Conversely, only 5% of sensitive upsets caused an SNR loss of 15 dB or more. Virtually all upsets caused an SNR loss of less than 40 dB. It is clear from the plot, then, that most of the configuration upsets caused little loss in SNR. In fact, for the 20 dB input SNR case, only 12.7% of the upsets caused an SNR loss of 1 dB or more.

It is interesting to note that in the cases with more severe noise, even more of the configuration upsets caused little loss in SNR. Table 1 shows numerical results for all three experiments. The results show that, as the noise at the input to the filter increased, the impact of SEUs on the SNR at the output of filter was reduced. In the 5 dB input SNR case, only 7.1% of the upsets caused an SNR loss of 1 dB or more.

Table 1. Percentages of upsets of sensitive configuration bits causing negative effects falling under certain thresholds.

Input SNR	Less than 0.1dB loss in SNR	Less than 1dB loss in SNR	Less than 3dB loss in SNR	Less than 6dB loss in SNR
20 dB	121,493 trials (81.2%)	129,217 trials (86.3%)	133,337 trials (89.1%)	136,229 trials (91.0%)
10 dB	128,725 trials (86.0%)	135,982 trials (90.8%)	139,586 trials (93.2%)	142,133 trials (94.9%)
5 dB	132,449 trials (88.5%)	139,124 trials (92.9%)	142,231 trials (95.0%)	143,824 trials (96.1%)

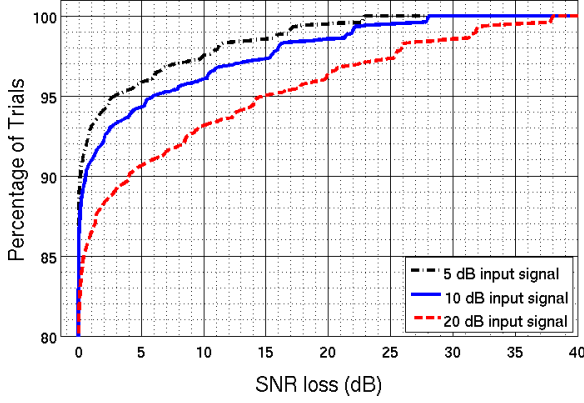


Fig. 4. Cumulative distribution functions (CDF) showing the loss in SNR due to a single *sensitive* configuration bit with various SNR levels at the input to the filter.

It appears that the errors caused by the configuration upsets were absorbed into the noise already present in the system.

As mentioned in Section 5, these SNR results can be linked to the desired BER metric, but this is only truly valid if the configuration upsets cause Gaussian-like noise. Unfortunately, the amount of time required to measure the BER for each configuration upset at each SNR level made it difficult to obtain direct BER measurements.

Initial investigation into whether the SNR measurements are a good indicator of BER are encouraging. Fig. 5 shows the BER curves for the golden filter as well as five different sample configuration bits. The plot shows that some of the upsets do cause a general degradation of BER as the SNR worsens. Others cause more dramatic effects, such as bit 1000574, as labeled in the plot. When this bit was upset, a BER *floor* was observed, meaning that the BER never improved past a value of about 0.005, or 1 bit error in every 200 bits sent. Effects such as these are certainly not Gaussian-like, but many of the bits we have examined do seem to have this property. Only a more thorough investigation will tell how good the assumption of Gaussian-like noise is. This will be pursued in future work.

The results of the experiments presented here are encouraging. They imply that it is indeed possible for a DSP or digital communications system to handle many of the errors caused by SEUs and that further investigation is warranted. The impact of this knowledge could lead to a dramatic re-

duction in the redundancy applied to FPGA-based DSP systems intended for radiation environments. Just as the dynamic cross section of the FPGA design specifies the configuration bits that affect the operation of a particular design, we may define an *application-specific* cross section that describes the set of configuration bits that are not inherently protected by higher-level algorithms.

For example, Fig. 6(c) is a plot of the configuration bits of the matched filter design which, with an input SNR of 20 dB, cause greater than 1 dB loss in SNR at the output of the filter. This is the application-specific cross section for an application where 1 dB or less SNR loss is acceptable for this receiver system. This cross section only contains 20,479 configuration bits (0.35% of the total 5,810,024 configuration bits), whereas the dynamic cross section contained 149,696 bits (2.5% of the total). This is nearly a 10x reduction in the number of configuration bits that need additional protection using TMR or some other technique.

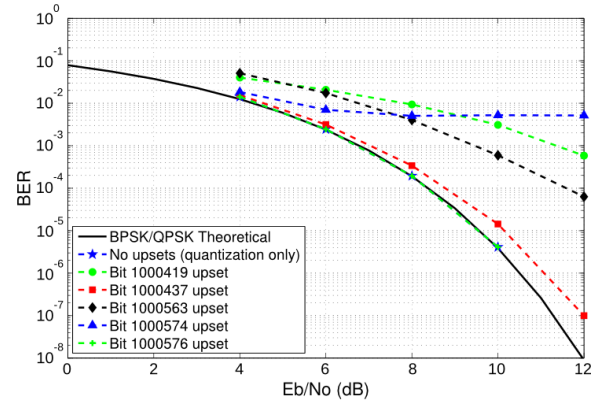


Fig. 5. The bit error rate (BER) curves for five selected configuration upsets compared to the theoretical curve for a BPSK/QPSK system.

7. FUTURE WORK

In future work, we will further analyze the direct BER impact of SEU effects on the matched filter and other components of a digital communications receiver. With these experiments, we will be better able to determine which upsets are handled naturally by the error-correction of the receiver, i.e. those that cause Gaussian-like noise.

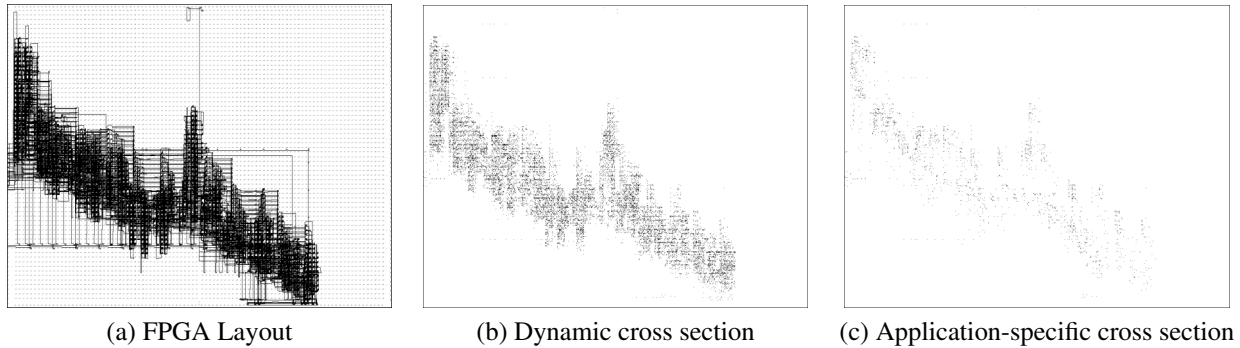


Fig. 6. (a) Screen capture of the physical layout of the matched filter circuit. (b) The dynamic cross section of the design as recorded by the BYU-LANL fault injection tool. (c) The subset of configuration bits that, when upset, cause an SNR loss greater than 1 dB with an input signal SNR of 20 dB.

We will also investigate reduced-cost mitigation strategies suitable for SEU-induced errors in FPGA-based DSP systems. These strategies will focus on mitigating the effects of the faults which cause high-magnitude noise. The faults which produce low-magnitude noise could be left unprotected and left to the inherent error-handling of the DSP algorithm. With this targeted approach, mitigation overhead may be drastically reduced compared to a full-design solution such as TMR.

8. CONCLUSION

The experiment presented shows that, in a real world FPGA-based system, many SEU-induced errors may be able to be safely ignored. Combined with traditional configuration scrubbing, a communications system like the one described in this paper is much less sensitive to configuration SEUs than expected. In the experiments presented, the majority of SEUs affecting the FPGA-based digital filter caused less than 1 dB loss in SNR at the filter's output. These results should be considered when designing reliable digital communications systems in order to avoid unnecessary over-design. A reduced-cost mitigation technique could be utilized in place of TMR, resulting in significant savings in terms of circuit area and power.

9. REFERENCES

- [1] M. Caffrey, "A space-based reconfigurable radio," in *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, T. P. Plaks and P. M. Athanas, Eds. CSREA Press, June 2002, pp. 49–53.
- [2] P. Graham, M. Caffrey, M. Wirthlin, D. E. Johnson, and N. Rollins, "Reconfigurable computing in space: From current technology to reconfigurable systems-on-a-chip," in *Proceedings of the 2003 IEEE Aerospace Conference*. Big Sky, MT: IEEE, March 2003, pp. T07.0603.1–12.
- [3] C. Carmichael, E. Fuller, J. Fabula, and F. D. Lima, "Proton testing of SEU mitigation methods for the Virtex FPGA," in *Proceedings of the IEEE Microelectronics Reliability and Qualification Workshop*, Pasadena, CA, December 2001.
- [4] M. Wirthlin, N. Rollins, M. Caffrey, and P. Graham, "Hardness by design techniques for field-programmable gate arrays," in *Proceedings of the 11th Annual NASA Symposium on VLSI design*, Coeur d'Alene, ID, May 2003, pp. WA11.1–WA11.6.
- [5] K. Morgan, D. McMurtrey, B. Pratt, and M. Wirthlin, "A comparison of TMR with alternative Fault-Tolerant design techniques for FPGAs," *Nuclear Science, IEEE Transactions on*, vol. 54, no. 6, pp. 2065–2072, 2007.
- [6] P. Reddy, A.L.N.; Banerjee, "Algorithm-based fault detection for signal processing applications," *Transactions on Computers*, vol. 39, no. 10, pp. 1304–1308, Oct 1990.
- [7] P. Reyes, P. Reviriego, J. Maestro, and O. Ruano, "A new protection technique for finite impulse response (FIR) filters in the presence of soft errors," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, 2007, pp. 3328–3333.
- [8] B. Shim and N. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 336–348, 2006.
- [9] E. Johnson, M. Caffrey, P. Graham, N. Rollins, and M. Wirthlin, "Accelerator validation of an FPGA SEU simulator," *IEEE Transactions on Nuclear Science*, vol. 50, no. 6, pp. 2147–2157, December 2003.