Melanie Berg

Using Classical Reliability Models and Single Event Upset (SEU) Data to Determine Optimum Implementation Schemes for Triple Modular Redundancy (TMR) in SRAM-based Field Programmable Gate Array (FPGA) Devices

M. Berg[1], *Member IEEE,* H. Kim[1], A. Phan[1], C. Seidleck[1], K. LaBel[2], *Member IEEE*, J. Pellish[2] *Member IEEE*, M. Campola[2] *Member IEEE*

1. AS&D in support of NASA Goddard Space Flight Center, Laurel, MD 20707
2. NASA Goddard Space Flight Center, Code 561.4, Greenbelt, MD 20771

National Aeronautics and Space Administration

Abstract: Space applications are complex systems that require intricate trade analyses for optimum implementations. We focus on a subset of the trade process, using classical reliability theory and SEU data, to illustrate appropriate TMR scheme selection.

Introduction

This study investigates mitigation performance and risk analysis for a variety of mitigation design strategies. The intention is to provide a means for optimum mitigation integration for critical applications. Risk is measured by analyzing reliability across time using classical reliability models and measured single event upset (SEU) data.

In this study, reliability is also analyzed across particle fluence by transforming classical reliability models [1] from the time domain into the fluence domain. As a benefit, analyzing mitigation in the fluence domain enhances the evaluation process by providing the ability to make direct comparisons to accelerated radiation test data (SEU data). Design implementation is targeted to a Sequential random access memory (SRAM)-based field programmable gate array (FPGA) (Xilinx Kintex-7) [2]. SEU test data was obtained by performing heavy-ion testing at Texas A&M Cyclotron Facility.

Triple Modular Redundancy (TMR)

TMR schemes [3-5] are defined by which portion of the circuit is triplicated and where the voters are placed.

- The strongest TMR implementation will triplicate all data-paths and apply separate voters to each data-path.
- However, this can be costly: area, power, and complexity.
- **A trade is performed to determine the TMR scheme that requires the least amount of effort and circuitry and while meeting project requirements.**
- Scope of mitigation for this study is:
  Block TMR(BTMR), Local TMR (LTMR), and Distributed TMR (DTMR).

Block TMR(BTMR)

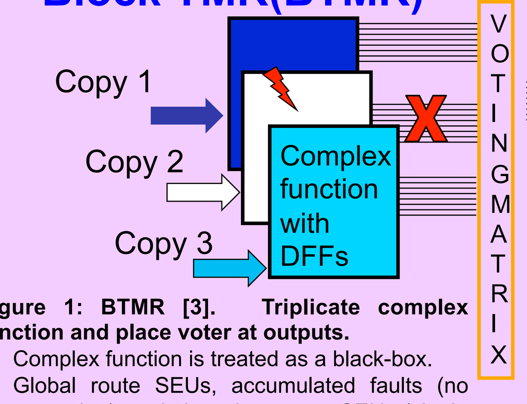

Figure 1: BTMR [3]. Triplicate complex function and place voter at outputs.
- Complex function is treated as a black-box.
- Global route SEUs, accumulated faults (no correction), and shared resource SEUs (single points of failure) can break the mitigation scheme.

BTMR: Voting is only performed at the outputs of complex blocks.
- Does not correct errors only masks them.
- **If blocks are not regularly flushed (e.g. reset), errors might accumulate – may not be an effective technique.**

Local TMR (LTMR)

DFF: Flip flop
CL: combinatorial logic

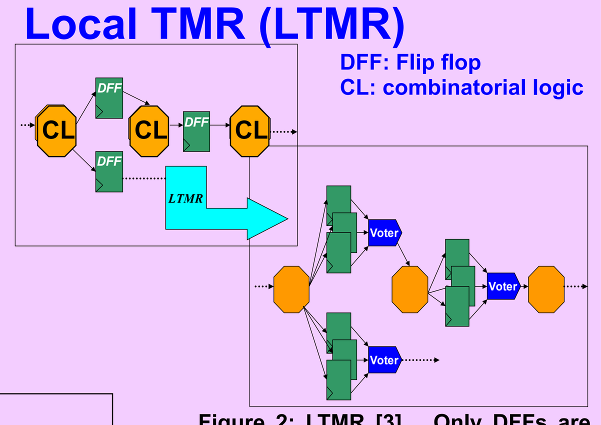

Figure 2: LTMR [3]. Only DFFs are triplicated. Data-paths stay singular.
- LTMR masks upsets from DFFs and corrects DFF upsets if voter feedback is used.
- SEUs in the data path, global route SEUs, and shared resource SEUs (single points of failure) can break the mitigation scheme.

Distributed TMR (DTMR)

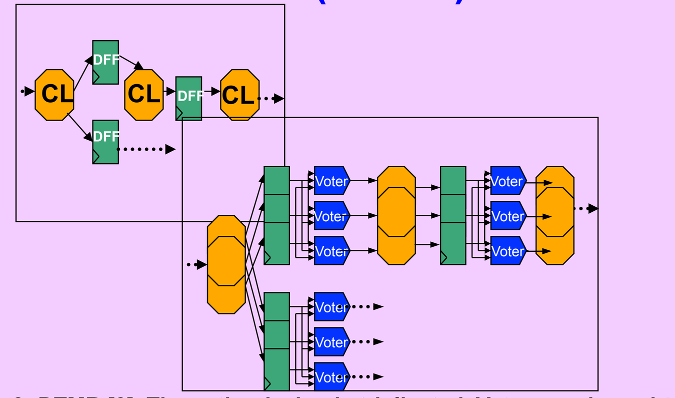

Figure 3: DTMR [3]. The entire design is triplicated. Voters are brought inside of the design and are placed after the DFFs.
- Most SEUs are masked.
- Errors can be corrected if voter feedback to DFFs are used.
- Global route SEUs, and shared resource SEUs (single points of failure) can break the mitigation scheme.

SEU Errors and Fault Correction

SEU cross-section (σ_{SEU}) DFF: Flip flop CL: combinatorial logic Single event functional interrupt: SEFI

$$P(fs)_{error} \propto P_{Configuration} + P(fs)_{functionalLogic} + P_{SEFI} \qquad (1)$$

Design σ_{SEU} — Configuration σ_{SEU} — Functional logic σ_{SEU} (DFF + CL) — SEFI σ_{SEU} (Global routes and hidden Logic)

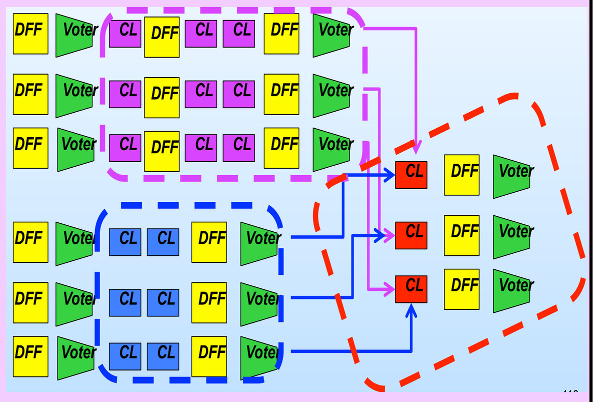

Figure 4: Mitigation window (MW) example.

Equation (1) describes the four categories of SEUs in FPGAs [3]: Configuration, functional logic, global routes, and hidden logic. Figure 4 illustrates how a design can be modularized into mitigation windows (MWs). The effects of SEUs on MWs can be classified as follows:

- The SEU in associated with an unused resource or disabled logic and consequently does not affect system operation.
- The SEU fault only affects one of the TMR triplet copies within the same mitigation window (MW). If only one triplet copy has a fault, then the MW will either (a) mask the fault or (b) mask and correct the fault.
- The SEU fault affects multiple triplet copies within the same MW. If more than one of the triplet copies are affected by an SEU, then the MW will malfunction and the fault will not be masked from the system. This is a common error signature of global route SEUs or configuration SEUs that control shared resources.

SRAM-Based FPGA TMR Strategies: Strengths and Weaknesses

SRAM-Based FPGAs and LTMR:

Because configuration SEUs are the dominant component faults in unhardened SRAM-based FPGAs [3]; and their affects to system operation are prolific, a **strong TMR** scheme (that can mask a variety of configuration SEUs) is required **(BTMR or DTMR)** [3-5]. Consequently, **do not use LTMR** with SRAM-based FPGAs as illustrated in Figure 5.

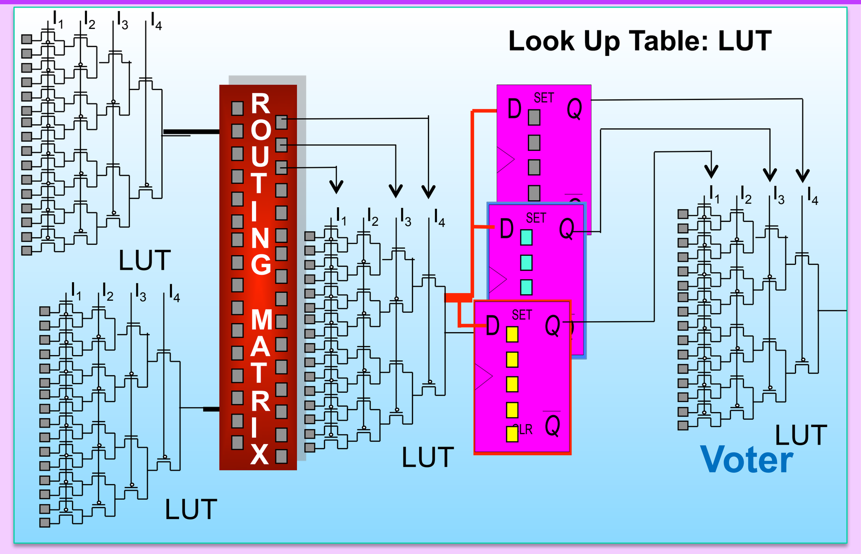

Figure 5: Block diagram of LTMR in an SRAM-based FPGA. With LTMR, there exists too many other configuration bits + logic (beyond the protected DFFs) that can be corrupted by an SEU. Applied mitigation needs to be stronger for SRAM-based FPGA devices.

SRAM-Based FPGAs and BTMR:

BTMR is a common approach to mitigation for two primary reasons:
- It requires the least number of voters (area savings) and
- It can be applied to black-box logic (e.g., intellectual property (IP) cores).

BTMR is a fairly strong mitigation strategy for flushable designs; i.e., feed-forward or highly-resettable circuits (as illustrated in Figure 6). It is also considered a strong application when requirements dictate short time intervals of correct operation.

Table 1 lists the classical reliability and Mean time to failure (MTTF) models [1] for an unmitigated design versus the corresponding BTMR implementation. If a mission's requirements dictate that BTMR shall reliably operate for a length of time near the time of failure for one unmitigated block, then BTMR becomes a weak mitigation scheme. This is because over time, the reliability of the BTMR scheme drops off faster than the unmitigated. **Table 1 shows: $MTTF_{block} > MTTF_{BTMR}$.**

Table 1: Classical reliability models over time.

λ:error rate	Classical Reliability Equation
Reliability for one block	$e^{-\lambda t}$
Reliability for BTMR	$3e^{-2\lambda t} - 2e^{-3\lambda t}$
Mean time to failure ($MTTF_{block}$)	$1/\lambda$
Mean time to failure ($MTTF_{BTMR}$)	$(5/6\lambda) = 0.833/\lambda$

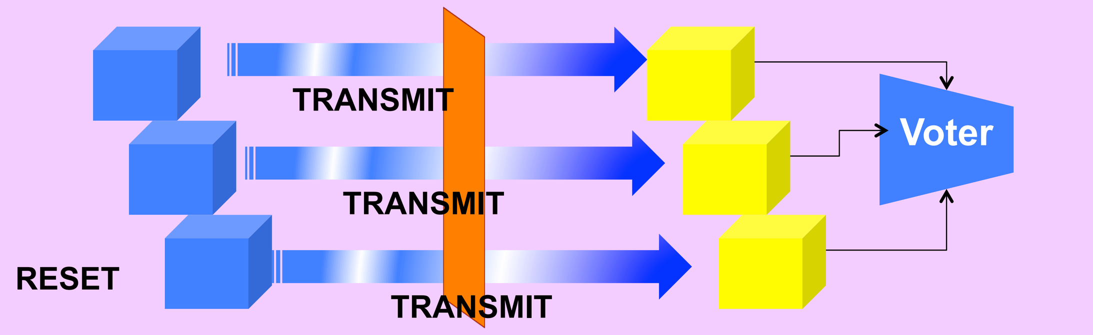

Figure 6: Flushable BTMR system.

SEU Test Methodology

The primary design under investigation (DUI) was the Counter Array [2] as illustrated in Figure 7. Variations of the DUI were created and implemented in the Xilinx Kintex-7 FPAG device (XC7K325T-1FBG900). Heavy-ion testing was performed at Texas A&M University Cyclotron Facility (TAMU).

Partitioning: SRAM-based FPGAs contain a significant number of shared resources that can become single points of failure [7]. Consequently, designs were partitioned such that no resources were shared across TMR domains. A partitioned design is illustrated in Figure 8. The Xilinx floor-planner tool was used for partitioning.

Mitigation: Evaluating mitigation strength was the primary goal of this investigation. The following is a list of DUI (counter array) variations that were manually developed: (1) no-mitigation (pure counters), (2) BTMR with partitioning, (3) DTMR with partitioning, and (4) DTMR without partitioning. One additional DUI was created by using the Synopsys' "Synplify Premier" synthesis tool [9]. Referring to Figure 7, DTMR (without voter feedback) was applied to the counter-array and LTMR was applied to its snap-shop array. We refer to this scheme as partial TMR (PTMR).

Scrubbing: During testing, an external configuration memory scrubber [3] was applied to the DUT's configuration. The configuration scrubber was verified for full operation during testing by performing a read-back of the DUT's configuration memory. Read-back should indicate approximately zero errors after each heavy-ion test.

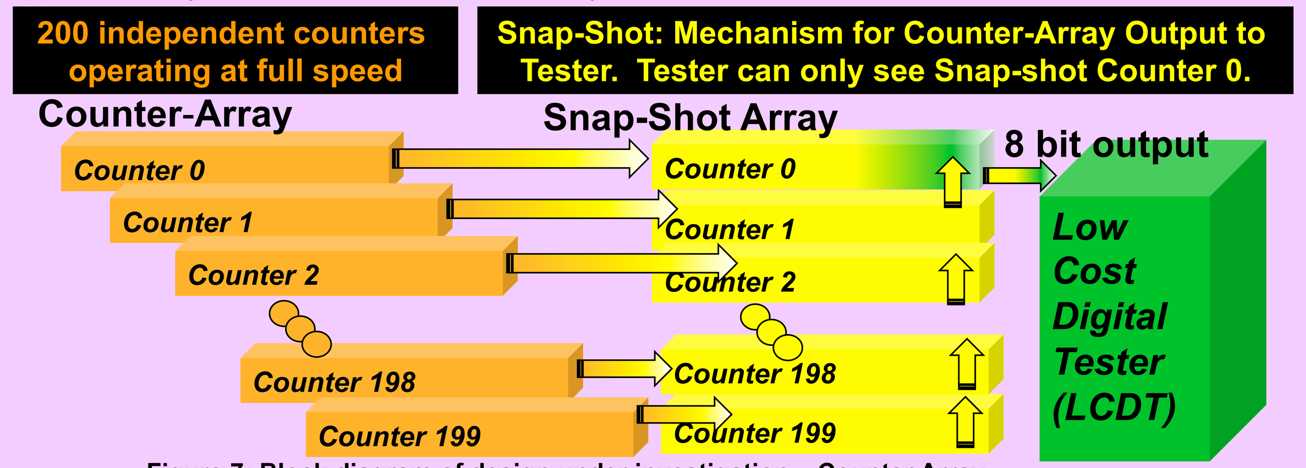

Figure 7: Block diagram of design under investigation – Counter Array

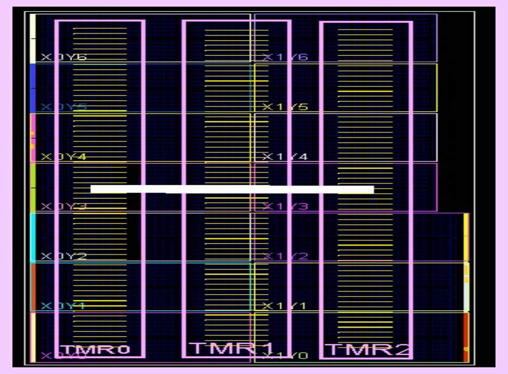

Figure 8: Physical layout of a partitioned TMR design (the same partitions were used for BTMR, DTMR, and PTMR).

SEU Test Results

The traditional approach to SEU data analysis and characterization is to convert σ_{SEU} data to error-rates. However, during the conversion process (from the fluence (Φ) domain to the time (t) domain), important information in the lower LET region is lost. Referring to Figure 9, there is a significant difference of particle flux when comparing lower LET particles to higher. Subsequently, in order to perform a comparison of mitigation efficiency, it is important to investigate behavior at lower LET without loss of information. For this reason, we replace error-rates (λ) with σ_{SEU} data ($\lambda \rightarrow \sigma_{SEU}$) and time with fluence ($t \rightarrow \Phi$) to convert classical reliability models from the t-domain to the Φ-domain. σ_{SEU} data from Figure 10 can be used as the variables in the Table 1 equations to obtain the reliability graphs shown in Figure 11 and Figure 12.

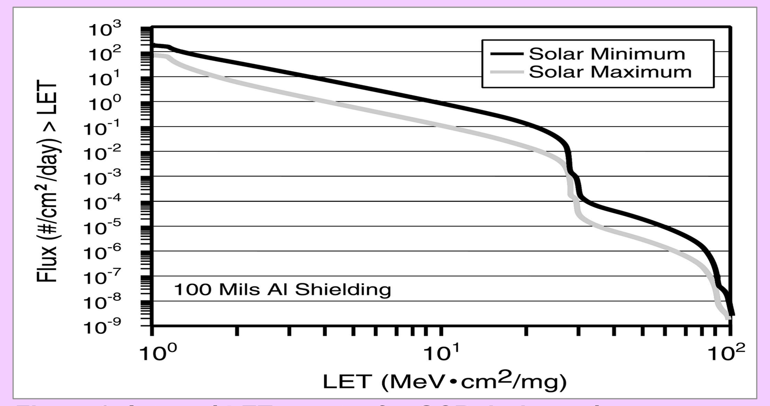

Figure 9: Integral LET spectra for GCR during solar maximum and solar minimum [8].

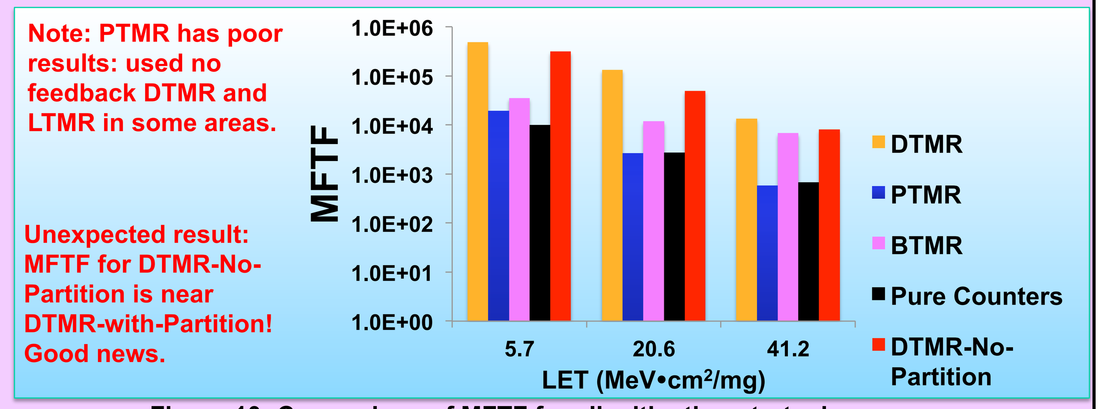

Figure 10: Comparison of MFTF for all mitigation strategies.

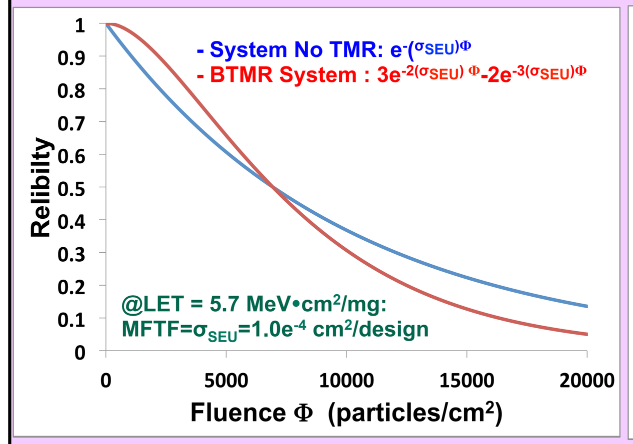

Figure 11: Reliability across particle fluence: BTMR versus System with No Mitigation.

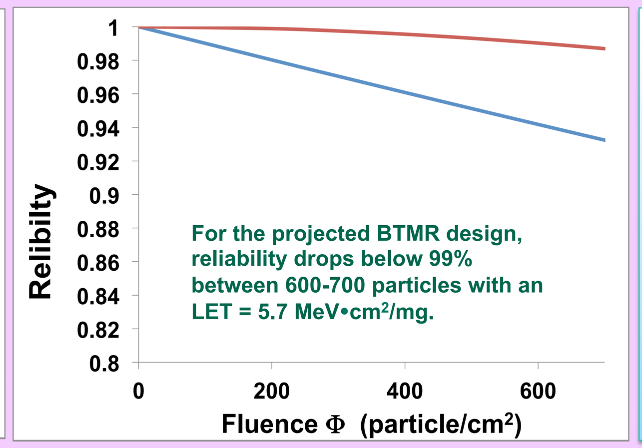

Figure 12: Reliability across particle fluence: Zoomed-in low fluence region of Figure 11.

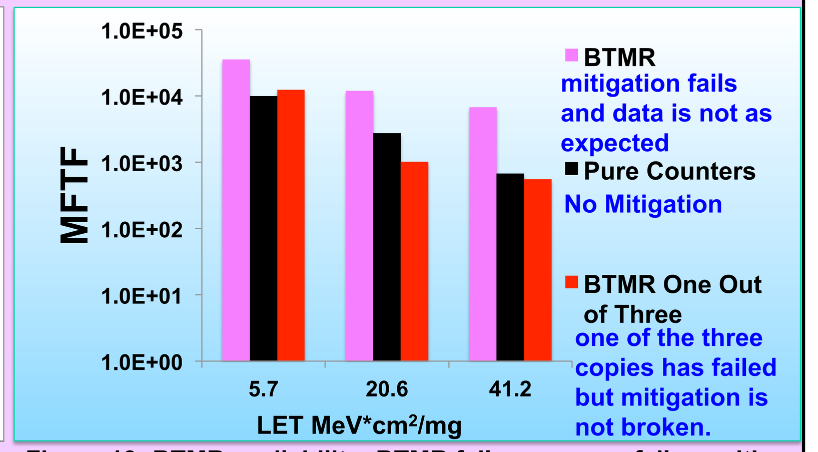

Figure 13: BTMR availability: BTMR failure; versus failure with no mitigation; versus any one failure out of the BTMR copies.

Analysis of SEU Test Results

Mitigation Strength

Given the counter-array DUI and based on σ_{SEU} data from Figure 10, DTMR with partitioning is the strongest mitigation strategy. This result is as expected. At the lower LETs, DTMR σ_{SEU} data has greater than one decade of improvement versus BTMR and over two decades versus no-mitigation.

As LET increases, the mitigation strategies start to converge in performance. This is because of the dominance of global route SEUs at higher LETs. Global route SEUs are a common factor for failure in all three strategies.

It was interesting to observe that there was not a significant difference between DTMR with partitioning and DTMR without partitioning. Potential explanations for the insignificant difference in σ_{SEU} data are the following: there may be hidden shared resources beyond the control of the floor-planner (partitioning tool), global routes may have a strong significance, and the DUI's isolated independent modules may play a role. As future work, this will be further investigated using a variety of DUIs and fault injection.

PTMR proved to be weaker than the system with no mitigation as LET increased. This is because LTMR was applied to the snap-shot register. **As previously mentioned, LTMR should not be used with SRAM-based FPGAs.**

BTMR and Reliability Models

In this investigation (counter-array DUI), BTMR performed better than expected; i.e., it's MFTF was higher (in all tested LETs) than the unmitigated design. This can be attributed to the fact that the DUI was made of 200 small independent modules (counters) and one large flushable structure (snap-shot array). Results closer to the reliability model predictions are expected to occur with DUIs that have one large MW with strong co-dependencies between modules. However, better results are expected with DUIs that are purely flushable.

BTMR and Availability

Regarding Figure 13, while the BTMR σ_{SEU} data can be used to characterize mitigation failure, the one-out-of-three can be used to assess availability. The premise is that when one copy fails, another is assumed to fail soon. Subsequently, **BTMR schemes require the system to halt or shut down upon one-out-of-three failure**. During this time, the system either flushes or the failed copied is serviced. This affects availability and is of **critical concern for satisfying mission requirements**. Further discussion is in the paper.

Conclusion

The conversion process from σ_{SEU} data to error-rates tends to lose valuable information regarding data trends across low LET. Hence, an analysis in the fluence domain was performed by transforming classical reliability models from the time domain to the fluence domain. The conversion is as follows: replace error-rates (λ) with heavy-ion accelerated testing σ_{SEU} data ($\lambda \rightarrow \sigma_{SEU}$) and time with fluence ($t \rightarrow \Phi$). This transformation was performed to improve lower-LET analysis of mitigation strategies.

As expected, DTMR was the strongest mitigation scheme. However, there is interesting data that show **DTMR without partitioning performed almost as well as DTMR with partitioning**. This will be further investigated with altering DUI MW size, creating more co-dependent internal-MW modules, and fault injection.

σ_{SEU} data and reliability models illustrated the strengths and weaknesses of BTMR. Data show that it is important to take into account the mission's required operational time and availability prior to selecting BTMR as the system's mitigation strategy.

An important result was observed with **PTMR**. The data showed that a **poor choice of mitigation application can cause the system to be more susceptible than a system with no mitigation.**

Acknowledgements

This work was supported in part by the NASA Electronic Parts and Packaging (NEPP) Program and the Defense Threat Reduction Agency.

References

See paper.

To be presented by Melanie Berg at the Institute of Electrical and Electronics Engineers (IEEE) Nuclear and Space Radiation Effects Conference (NSREC), Boston, Massachusetts, July 13-17, 2015.