XAPP987 (v1.0) March 18, 2008

Single-Event Upset Mitigation Selection Guide

Author: Brendan Bridgford, Carl Carmichael, and Chen Wei Tseng

Summary

This application note discusses different aspects of single-event upsets and recommends appropriate mitigation schemes under each circumstance. The concept is not constrained by device family. Furthermore, this document provides guidelines to choose the most cost-effective mitigation scheme.

Introduction

When designing in-orbit, space-based, or extra-terrestrial applications in hostile radiation environments, users must consider the effects of charged particles such as heavy ions or protons. As these charged particles travel through the FPGA (Figure 1), they can alter the logic state of any static memory element, resulting in single-event upsets (SEUs). An SEU in the configuration memory array can have adverse effect on expected FPGA functionality.

Note: Static upset in the configuration memory is not synonymous with a functional error. Depending on the application requirement and cost constraints, different mitigation should be deployed.

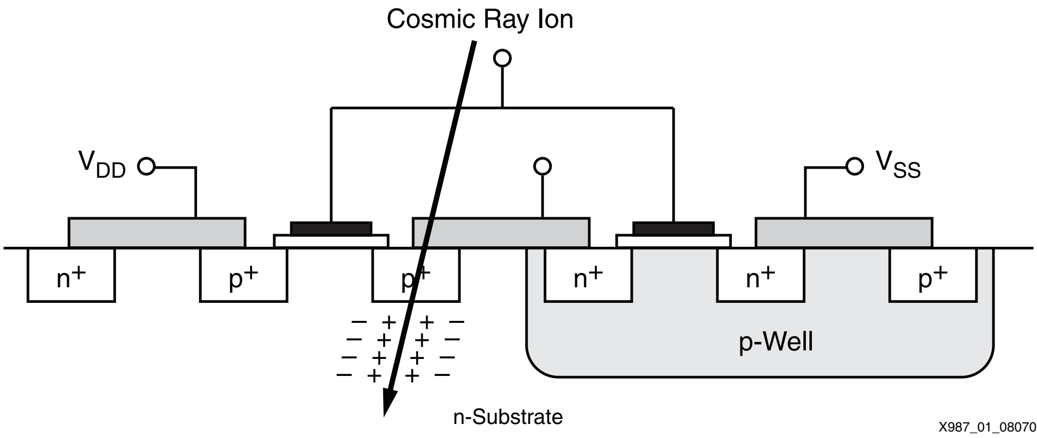


Figure 1: **SEU in FPGA**

Glossary

Some common terminology used throughout this document are defined in this section.

Cross Section

A statistical representation of the sensitivity of a certain single-event effect represented as a relative area (cm^2).

Device

A single integrated circuit.

Failure

An unrecoverable error.

Functional Error

A logic error in the user function.

Functional Interrupt

A disruption in device operation requiring system level intervention to regain normal functionality. Typically causes the loss of user or system data.

Functional Disturbance

A self-recovering disruption in device operation that causes the loss of user or system data.

Logic Error

Incorrect state or signal transition.

Multiple-Bit Upset (MBU)

An SEU that results in more than one adjacent bits flipping due to an oblique angle strike. MBU probability steadily increases as geometries shrink. Use of maximum MBU distance observed is useful to determine block RAM interleaving required so that even MBUs are able to be corrected by the ECC.

Single-Bit Upset (SBU)

Same as SEU.

Scrubbing

The process of correcting any configuration cell upsets through FPGA partial reconfiguration. Scrubbing does not interrupt user design function.

Single-Event Effect (SEE)

The resulting electrical disturbances caused by the direct ionization of a silicon lattice by an energetic charged subatomic particle.

Single-Event Functional Interrupt (SEFI)

An SEE that results in the interference of the normal operation of a complex digital circuit. SEFI is typically used to indicate a failure in a support circuit, such as loss of configuration capability, power on reset, JTAG functionality, a region of configuration memory, or the entire configuration.

Single-Event Transient (SET)

A signal transition caused by a SEE. Often observed as a glitch.

Single-Event Upset (SEU)

A state change (or *flip*) of a single data bit storage or memory cell caused by an SEE. An SEU can affect the configuration memory cell states, the block RAM contents, a CLB DFF, a LUTRAM, or SRL16 memory cell (which are also configuration memory cells, directly accessible to the user).

System

An integration of multiple devices and circuit boards or modular sub-systems.

User Function

User-specified operational functions defined by the data stored in device configuration memory.

Application Analysis and Requirement

Designing a robust system against SEUs can be costly. Application and radiation-tolerance requirements can dramatically impact mitigation methods. If the FPGA is to be deployed in mission-critical applications in a harsh environment, the mitigation scheme should involve redundant FPGAs with configuration management. On the contrary, if the application, such as image capture, is fault tolerant while the operating window is small, no mitigation might be necessary.

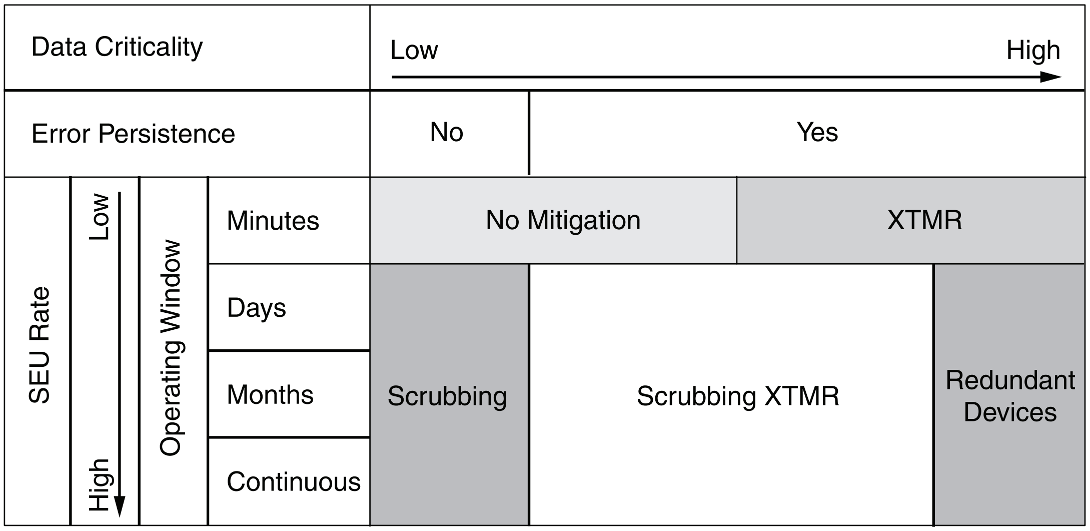
Before selecting a mitigation scheme, it is important to answer the following questions:

- Is the application error tolerant?
- What is the expected operating time window?
- What is the FPGA performance requirement?
- What is the FPGA power requirement?
- How cost sensitive is the application?
- Can the application withstand any possible FPGA downtime? If so, does the SEFI cross section specified in the data sheet meet the application requirement?

Trade-offs between these application requirements dictate the mitigation scheme. Figure 2 provides an overview for mitigation scheme selection based on the application and environmental needs.

Note: The matrix in Figure 2 does not account for power consumption or design performance analysis.

A more detailed discussion on each mitigation scheme follows in “Mitigation Schemes.”



X987_02_051007

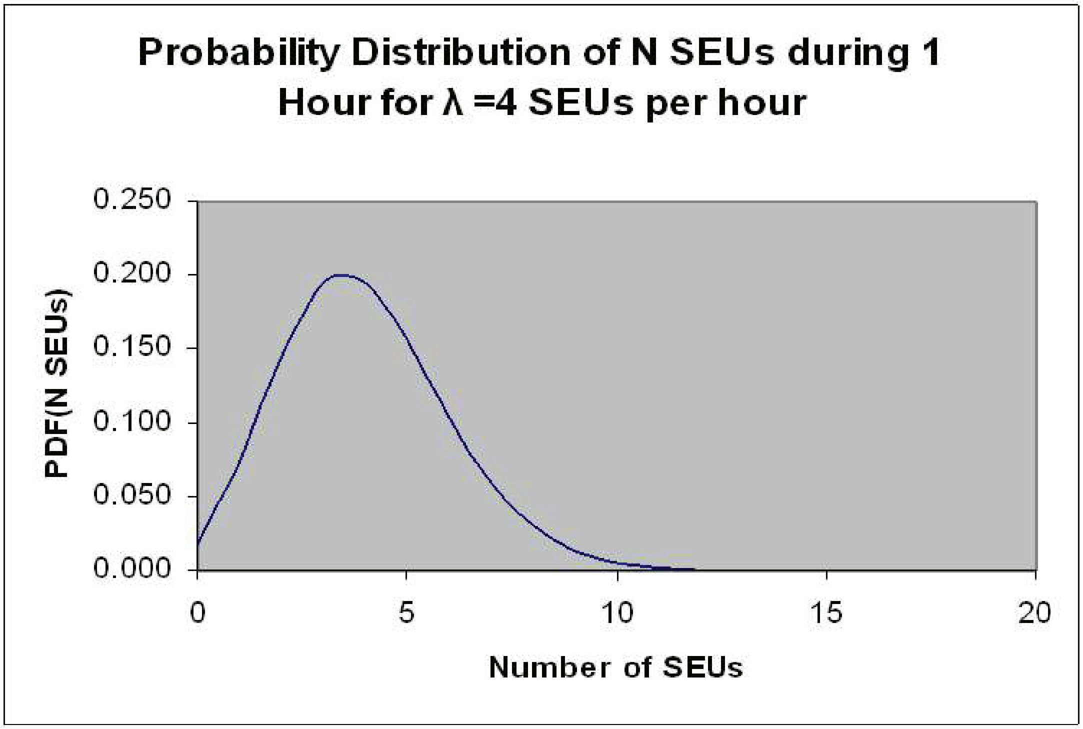
Figure 2: **Mitigation Scheme Matrix**

Table 1 shows example error rates for a XQR2V6000CF1144V in a geosynchronous orbit (Solar Quiet conditions) for the configuration memory, block memory, and SEFIs.

Table 1: **GEO Error Rates for XQR2V6000(CREME96)**

XRR2V6000 – 36,000 km	Mean Time to Error	Units
Configuration Memory	1.8	Hours
Block Memory	11.8	Hours
POR-SEFI	221	Years
SMAP-SEFI	181	Years

SEU error rates are modeled as a Poisson process (Figure 3). For example, if the expected upset rate is four events per hour, it is possible that the FPGA experiences no upsets (1.8% chance), or 7 upsets (6% chance) per hour. Therefore, if the configuration mitigation rate is at least ten times the expected upset rate, the probability of upset accumulation is minimized.



X987_03_031208

Figure 3: **SEU Rate Modeled as a Poisson Process**

Mitigation Schemes

Power-Cycling Only

Any time the FPGA is power-cycled, all its contents are refreshed. Therefore, power-cycling is essentially the simplest form of scrubbing. Fault-tolerant applications in low radiation environments should consider this scheme. Bear in mind that approximately one out of every eight configuration bits is routing bits. In a fully utilized FPGA, less than 40% of the routing bits are used. Therefore, even if an FPGA experiences an SEU, it is likely to have no impact on the design.

Configuration Management (Advanced Scrubbing)

Upsets in Xilinx FPGAs can be removed by reconfiguring the affected configuration bits. This process of correcting SEUs is referred to as scrubbing. Configuration management (or advanced scrubbing) serves the purpose of detecting and correcting SEUs in the FPGA. Configuration management can be hosted in a radiation-hardened FGPA, CPU, ASIC, or the FPGA itself. In most cases, configuration management is implemented in conjunction with Xilinx Triple Module Redundancy (XTMR) or redundant FPGA mitigation schemes.

Configuration management can detect and correct design alterations caused by SEUs; however, it cannot mitigate the effects caused by SEUs. To mitigate these effects on the design, implementation of triplicated logic or devices must be deployed.

Many flavors of configuration management exist. Different algorithms are implemented for performance or resource utilization reasons. The main functions of the configuration management include SEU correction, SEFI detection, and SEFI mitigation.

For more details on Virtex™ and Virtex-E configuration management, refer to XAPP216, *Correcting Single-Event Upsets Through Virtex Partial Configuration*.

For more details on Virtex-2 configuration management, refer to XAPP779, *Correcting Single-Event Upsets in Virtex-II Platform FPGA Configuration Memory.*

For more details on Virtex-4 configuration management, refer to XAPP988, *Correcting Single-Event Upsets in Virtex-4 Platform FPGA Configuration Memory* (available under non-disclosure agreement).

Although configuration management is typically hosted on a radiation-hardened device, it is possible to shift most of these functions back to the FPGA, leaving the radiation-hardened device for other functions. This option has the advantage of guaranteeing implementation of the latest Xilinx-endorsed configuration management algorithm without the need to respin the board.

For more details on self-hosting configuration management, refer to XAPP989, *Correcting Single-Event Upsets in Xilinx FPGAs with a Self-Hosting Configuration Management Core*.

Xilinx Triple Module Redundancy (XTMR)

Xilinx provides a software tool, TMRTool, to simplify the task of design triplication. TMRTool can partially or fully triplicate a design, insert voters, synchronize feedback path loops, and allow customized user-triplicated module insertion. A triplicated design mitigates SEU impacts on the user design.

However, a TMR design in a single device is still vulnerable to SEFI. A TMR design also consumes more power and can suffer a design performance hit. For example, system clock frequencies over 100 MHz in a Virtex-II device might be difficult to achieve. Some other design considerations such as board layout complexity, signal integrity analysis, and asynchronous applications are documented in the TMRTool user guide which can be obtained with the evaluation TMRTool.

For more information regarding TMRTool, refer to the TMRTool product page:

http://www.xilinx.com/ise/optional_prod/tmrtool.htm

Redundant Devices

Implementation of configuration management and duplicating the design on multiple FPGAs with voting on the outputs of the FPGAs provides the most robust mitigation scheme. A wide variation of the implementation exists, ranging from two, three or more FPGAs.

In a dual FPGA implementation (Figure 4), each FPGA should host two copies of the same design. The outputs can then be voted on a radiation hardened device. If one output differs from the other three, the voter should ignore all outputs from the failing FPGA as it re-synchronizes with the other FPGA. If the remaining FPGA has mismatched outputs, all data should then be discarded as the system resets both FPGAs.

For three-plus FPGAs implementations (Figure 5), each FPGA holds same design and has the outputs voted on a hardened device.

For Four or more FPGA implementations, voting can be done on three FPGA outputs while the other FPGAs stand in reserve. If any of the three primary FPGAs are down, the reserve replaces the failing FPGA while the failing FPGA comes back online and in sync with the other FPGAs. This scheme can work well as long as there are still two matching FPGA outputs.

Designing applications with redundant-device mitigation can be challenging. If one FPGA is temporary dysfunctional, bringing the failing device back in sync with the other FPGAs might require some thoughtful design considerations. One approach is to designate a reset stage in the design. The application returns to the reset state either routinely or when it detects one of the FPGA is out due to SEFI.

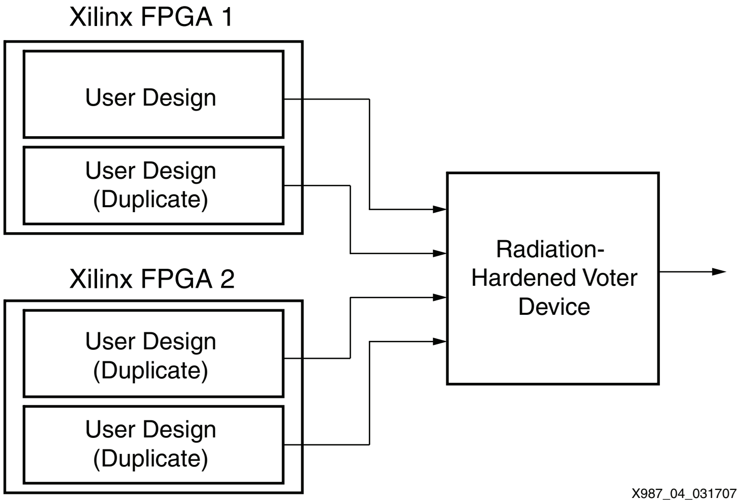


Figure 4: **Dual FPGAs implementation**

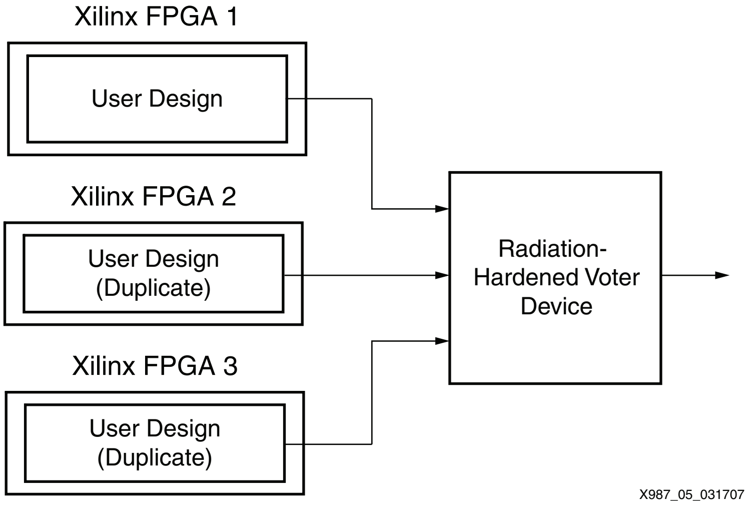


Figure 5: **Three FPGA Implementation**

It is also feasible to employ a board-level hard-wire voting scheme. In the hard-wire voting scheme, the traces of the outputs converge on the PCB board to create a voting scheme. In this scheme, the output standard and strength must be carefully selected to ensure two outputs can override the disagreeing output while the downstream device can correctly recognize the logic level. This scheme requires intensive IBIS simulation and board layout planning such as trace length, termination resistors, etc., to eliminate the reflection and protect signal integrity.

Selecting a Mitigation Schemes

As previously discussed, some mitigation schemes compliment each other. For example, configuration management is usually an addition to XTMR or redundant-device schemes. Configuration management ensures design integrity while XTMR or redundant devices mitigate design impacts. A successful implementation of XTMR with scrubbing lowers the design's failure cross-section to its SEFI cross-section.

Table 2: **Performance Overview of Mitigation Schemes**

Mitigation Scheme	Mitigation Strength	Board Layout Complexity	Ease in Meeting Timing Constraints	Power Consumption	Component Cost
No mitigation (power cycling)	Weak	Low	Normal	Typical	Low
XTMR	Medium	High	Reduced	~3X typical	Low
Configuration management (scrubbing)	Medium	Low	Normal	Typical	Medium
XTMR + Configuration management	Strong	High	Reduced	~3X Typical	Medium
Redundant devices + configuration management	Strongest	Medium	Normal	2~4X typical	High

Conclusion

Although FPGA can be vulnerable to SEUs in hostile environments, Xilinx provides various software tools and documentations on mitigation techniques. Numerous mitigation schemes are available; each with its strengths and weakness. Selecting the most appropriate mitigation scheme ensures mission success while minimizing system cost.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/18/08	1.0	Initial Xilinx release.

Notice of Disclaimer

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.