SANDIA REPORT
SAND2012-0454
Unlimited Release
Printed January 2012

Evaluation of the Leon3 Soft-Core Processor Within a Xilinx Radiation-Hardened Field-Programmable Gate Array

Mark W. Learn

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2012-0454
Unlimited Release
Printed January 2012

Evaluation of the Leon3 Soft-Core Processor Within a Xilinx Radiation-Hardened Field-Programmable Gate Array

Mark W. Learn
Sensors & Embedded Systems 02623
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0661

Abstract

The purpose of this document is to summarize the work done to evaluate the performance of the Leon3 soft-core processor in a radiation environment while instantiated in a radiation-hardened static random-access memory based field-programmable gate array. This evaluation will look at the differences between two soft-core processors: the open-source Leon3 core and the fault-tolerant Leon3 core. Radiation testing of these two cores was conducted at the Texas A&M University Cyclotron facility and Lawrence Berkeley National Laboratory. The results of these tests are included within the report along with designs intended to improve the mitigation of the open-source Leon3. The test setup used for evaluating both versions of the Leon3 is also included within this document.

ACKNOWLEDGMENTS

Aeroflex-Gaisler
Xilinx
Xilinx Radiation Test Consortium

CONTENTS

FIGURES

TABLES

NOMENCLATURE

AHB	AMBA high-speed bus
APB	AMBA peripheral bus
AMBA	advanced microcontroller bus architecture
BRAM	block random-access memory
CRC	cyclic redundancy check
DCM	digital clock manager
DRAM	dynamic random-access memory
DSU	debug support unit
DTLB	dual translation look-aside buffer
DUT	device under test
ECC	error-correcting code
FPGA	field-programmable gate array
FPU	floating-point unit
FT	fault tolerant
GCR	Galactic Cosmic Rays
GPIO	general purpose input/output
GUI	graphical user interface
IRAM	intelligent random-access memory
ITLB	instruction translation look-aside buffer
IU	integer unit
JTAG	Joint Test Action Group
LBNL	Lawrence Berkeley National Laboratory
LET	linear energy transfer
LUT	look-up table
MISSE	Materials International Space Station Experimental
MMU	memory management unit
NBA	node-based architecture
OS	operating system
OTP	one-time programmable
PLL	phased lock loops
RAM	random-access memory
ROM	read-only memory
RTEMS	Real-Time Executive for Multiprocessor Systems
SDP	simple dual port
SEU	single-event upset
SRAM	static random-access memory
TAMU	Texas A&M University
UART	universal asynchronous receiver/transmitter
XRTC	Xilinx Radiation Test Consortium

EXECUTIVE SUMMARY

Node-based architecture (NBA) designs for future satellite projects hold the promise of decreasing system development time and costs, size, weight, and power and positioning the laboratory to address other emerging mission opportunities quickly.

Reconfigurable field-programmable gate array (FPGA)-based modules will comprise the core of several of the NBA nodes. Microprocessing capabilities will be necessary with varying degrees of mission-specific performance requirements on these nodes. To enable the flexibility of these reconfigurable nodes, it is advantageous to incorporate the microprocessor into the FPGA itself.

Soft-core processors are implemented in the fabric of FPGA devices. When used in static random-access memory (SRAM)-based FPGAs, these soft-core processors are susceptible to single-event upsets (SEUs) from radiation sources.

An SEU is a transient event, and though they are nondestructive, SEUs can cause a change in the state of a logic circuit. In particular, user memory such as the cache, instruction/data registers, and even floating-point units (FPUs) can cause system lockup when an SEU occurs.

Utilization of radiation-hardened-by-design FPGAs can help reduce SEUs to the configuration memory of the soft-core processor, but not all features of these FPGAs are necessarily hardened and could still be susceptible to on-orbit upsets. Aeroflex-Gaisler offers a fault-tolerant version of the Leon3 soft-core processor targeted to a one-time programmable FPGA such as the Actel RTAX. A similar version of this fault-tolerant core is also provided for use in an SRAM-based FPGA, primarily for prototyping in a reconfigurable device. Until now, no testing has been done to assess radiation hardness of this core in a radiation-hardened SRAM-based FPGA.

Since the fault-tolerant Leon3 was designed for anti-fusing technology, the purpose of this study is to evaluate how well the fault-tolerant Leon3 performs in a radiation-hardened, SRAM-based FPGA within a radiation environment and how it differs from the unmitigated, open-source Leon3 core. This evaluation will look at the differences between the unmitigated, open-source Leon3 and the fault-tolerant Leon3. The evaluation also looks at the results in providing some basic mitigation to the open-source Leon3 to see if there was any improvement over the fault-tolerant Leon3. This included removing the on-chip RAM from the design and using only LUT-RAM, which is inherently protected by the radiation-hardened FPGA. The second mitigation design looked at using the error correcting code block RAM (BRAM) available on the Virtex-5 FPGA.

Overall, the fault-tolerant Leon3 performed much better than the unmitigated Leon3 design. Using certain software such as the integer unit test, the Leon3FT would see about one reset every 391 years in a geostationary earth orbit while the unmitigated open-source Leon3 would see a reset every 1.6 years. Though the mitigated open-source Leon3 designs provided some improvement over the unmitigated design, the fault-tolerant Leon3 still provided much more mitigation than the two designs.

Further improvements to these designs could help to reduce the error rates even more. As seen during testing, traps were the dominant issue seen during software testing. The large increase in traps was mainly due to multi-bit errors that occurred while utilizing such a high flux.

1 GENERAL

1.1 Scope

The purpose of this document is to summarize the work done to evaluate the radiation environment performance of the Leon3 soft-core processor in a radiation-hardened static random-access memory (SRAM)-based field-programmable gate array (FPGA).

1.2 Document Structure

This document consists of the following major sections, with a brief description of each given below:

1. **General Information** – Outlines the purpose, summary, background, and contents of this study.

2. **Leon3 Background** – Provides a brief description of the processor used in this study.

3. **Test Setup** – Describes the hardware and software test setup used for evaluating the soft-core processors.

4. **Results** – Provides the results of the evaluations of the soft-core processors.

5. **Conclusion** – Provides an overview of the document.

1.3 References

1.3.1 Project References[1]

The references identified in Table 1 were used in preparation of this document.

Table 1. External Reference Documents.

Document Number	Document Title
1.0.21	GRLIB IP Core User's Manual
1.0.21	GRLIB FT-FPGA User's Manual
1.0.21	GRLIB IP Library User's Manual
1.1.36	GRMON User's Manual
SAV080S19308	The SPARC Architecture Manual, Version 8
	www.gaisler.com
DS512	LogiCORE IP Block Memory Generator v4.2
	Single Event Effects Qualification Summary for the UT699 Leon3FT Processor

[1] Documents used are the latest version, unless otherwise specified.

2 LEON3 BACKGROUND

Two versions of the Leon3 soft-core processor were used in this study: the open-source processor and the licensed processor. Both of these processors are described in the following sections.

2.1 Open-Source Leon3

The open-source Leon3 soft-core processor is a 32-bit processor core conforming to the IEEE-1754 (SPARC V8) architecture. The block diagram of the Leon3 is shown in Figure 1.

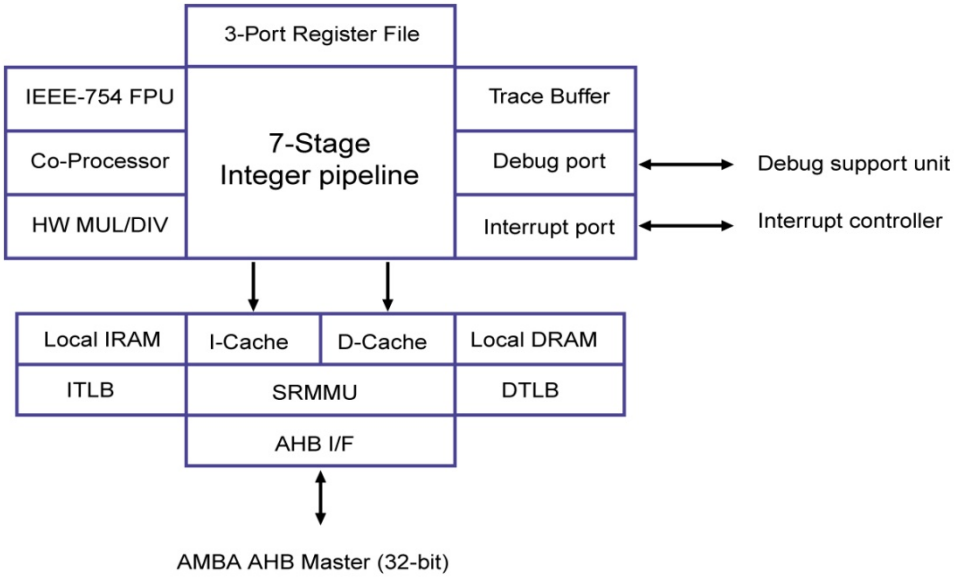

Figure 1. Leon3 core block diagram.

The Leon3 is customizable to generate a smaller or faster implementation and the full source code is available under the GNU General Public License. The Leon3 is also capable of supporting floating-point operations by using either a high-performance floating-point unit (FPU), which uses more resources, or a lighter FPU, which uses fewer resources and has a lower performance. Both versions of these FPUs support double-precision floating-point operations.

2.2 Licensed Leon3

The licensed core is also referenced as the Leon3FT because it was designed to be fault tolerant (FT) within an Actel RTAX FPGA. The licensed Leon3 will be referenced as Leon3FT throughout the remainder of this paper. The Leon3FT used within this evaluation was not intended to be fault tolerant in a Xilinx device, but to be used as a prototype and developed with an equivalent netlist on a reconfigurable platform rather than with the complications of developing on a one-time programmable (OTP) platform.

The Leon3FT core is similar to the open-source version, but is built to reduce the number of single-event upsets (SEUs) within the processor when implemented in an Actel RTAX. The Leon3FT core focuses on protection of on-chip RAM used for the integer unit (IU), FPU register file, and the cache memory. Configuring the Leon3FT is not as flexible as the open-source version since a netlist is provided only by Gaisler and the source code is not available. Any changes that are required of this core must be requested of Gaisler.

Generally, the IU is configurable and can be protected by four different implementations in an Actel RTAX:

- Hardened flip-flops with no error checking.

- 4-bit checksum per 32-bit word. Detects and corrects 1 bit per byte (4 bits per word). The pipeline is restarted on correction.

- 8-bit checksum per 32-bit word. Detects and corrects 1 bit per byte (4 bits per word). Correction is “on the fly” without pipeline restart.

- 7-bit BCH checksum per 32-bit word. Detects 2 bits and corrects 1 bit per word. The pipeline is restarted on correction.

The IU is not configurable with the netlist provided by Gaisler. The IU is configured so that it uses 8-bit parity protection.

The type of protection used on the FPU is not configurable and is always protected with an 8-bit parity without pipeline restart. If the light FPU is used, then 4-bit parity is implemented.

The cache protection is not configurable and is protected using 4-bit parity for the tag and data. If there is an error during a cache access, then the cache line will get flushed and the failed instruction will be re-executed.

Note that within the Xilinx radiation-hardened-by-design XQR5V FX130T FPGA (refered to as V5QV for the remainder of the paper), flip-flops are also hardened, as they are in the Actel RTAX.

3 TEST SETUP

The Xilinx V5QV radiation-hardened-by-design FPGA was used while testing the Leon3. The V5QV was used to protect the configuration bits, flip-flops, and look-up tables (LUTs). Resources not protected by the V5QV were removed when possible from the Leon design such as the digital signal processors (DSP48s), digital clock managers (DCMs), and phased lock loops (PLLs) to reduce the number of possible SEUs and capture the most accurate processor cross section possible.

The following sections describe the configurations used for both the open-source Leon3 and the Leon3FT on the V5QV.

3.1 Device Under Test

In order to test the radiation susceptibility of the Leon3 processor, the hardware configuration in Figure 2 was used.

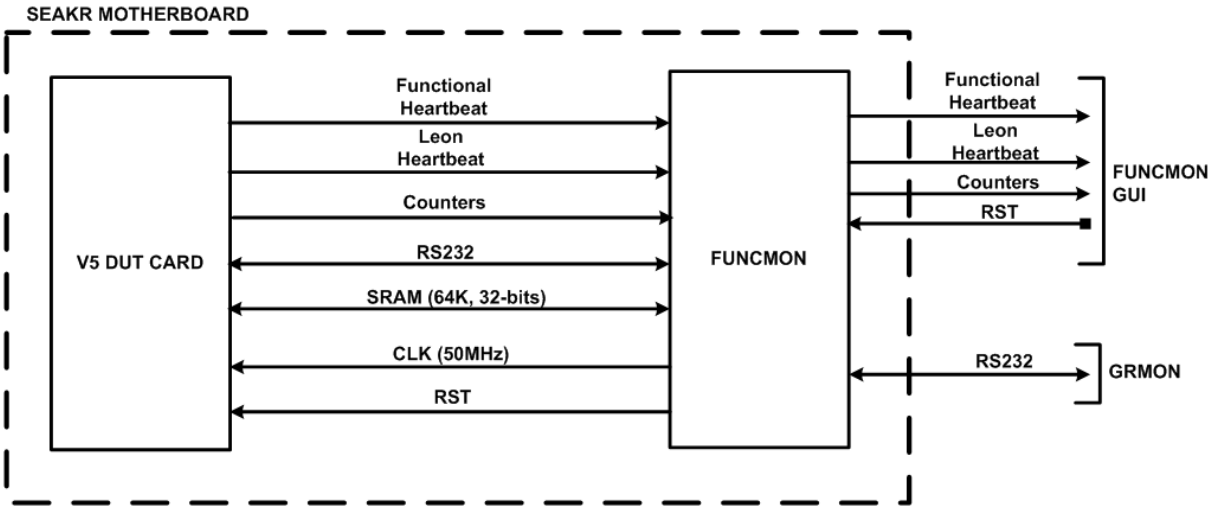

Figure 2. Device under test block diagram.

The V5QV device under test (DUT) card is used to test the open-source Leon3 and the fault-tolerant Leon3. The DUT interacts with a functional monitor (FUNCMON) located on the SEAKR motherboard. FUNCMON is a Virtex 2 Pro FPGA. The DUT card is a custom card built to sit on two Teradyne connectors and interface with FUNCMON. An external configuration scrubber (CONFIGMON) is also located on the motherboard, but is not shown in the figure. CONFIGMON's purpose is to configure both FUNCMON and the DUT and to provide configuration scrubbing on the V5QV FPGA.

As seen in Figure 2, the DUT card sends various status signals to FUNCMON. Each signal that is passed from the DUT to FUNCMON is also outputted to a graphical user interface (GUI) on a separate PC to monitor the state of the Leon3. Table 2 lists each signal used and provides a brief description of the signal.

Table 2. Signal Interface Description.

Signal Name	Description
Functional Heartbeat	Pulses on each CLK cycle to indicate that the Leon3 has been configured.
Leon Heartbeat	Pulses to indicate that the Leon3 software is running.
Counters	Various counters that indicate when a certain error has occurred.
RS232	UART used to provide communication between the debugger and the Leon3. Used to download executable code and execute code.
SRAM	64K x 32-bit memory used to store executable code. FUNCMON used as pseudo-SRAM to the Leon3 processor.
CLK	50-MHz clock provided by FUNCMON to the Leon3 processor.
RST	Reset signal that can be issued from the GUI to the Leon3 processor.

3.1.1 Open-Source Leon3 Configuration

The configuration of the open-source Leon3 is represented in Figure 3.

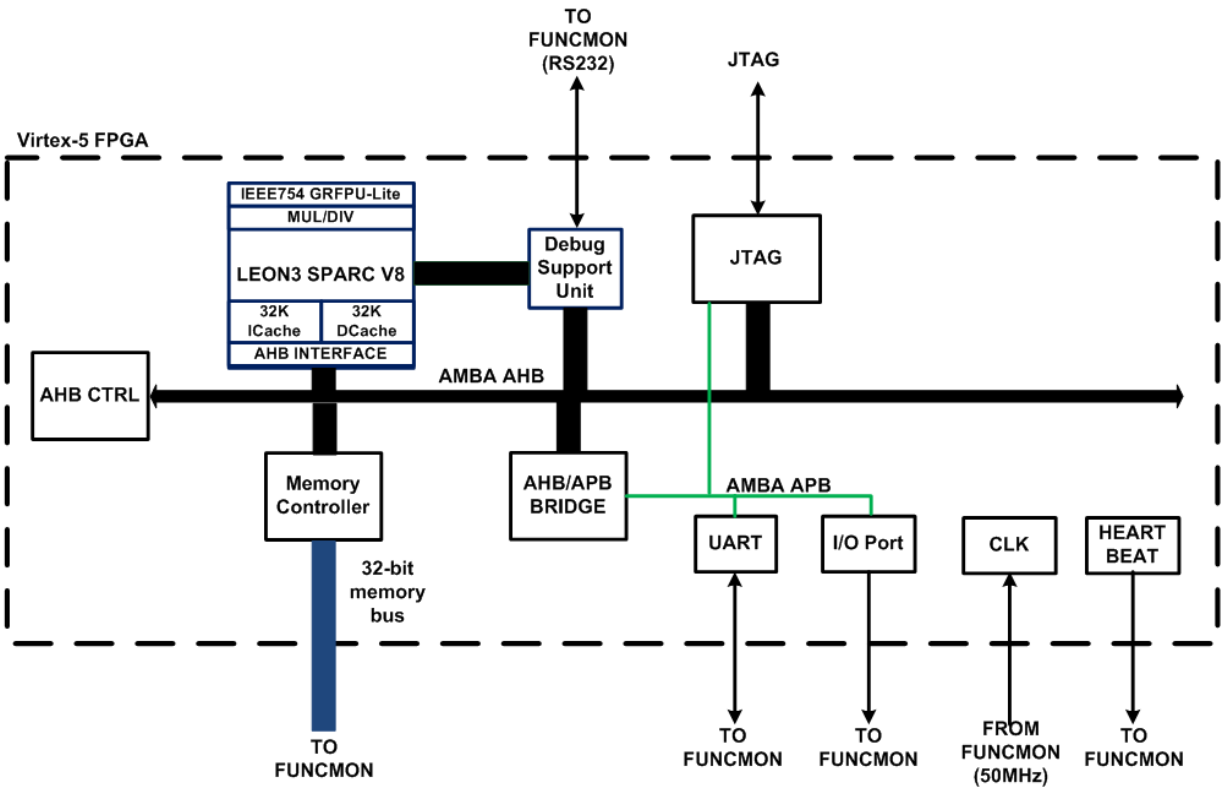

Figure 3. Open-source Leon3 configuration.

The open-source Leon3 design is quite simple. A design was chosen with a minimum set of peripherals in order to test just the core elements of the Leon3 processor and to reduce the number of SEUs that could cause a failure within the open-source Leon3. As seen in the figure, the Leon3 is configured with the GRFPU-Lite, the hardware multiply and divide IU, and a 32-KB cache.

The advanced microcontroller bus architecture (AMBA) high-speed bus (AHB) is utilized in order to access the components seen in the figure. An AMBA peripheral bus (APB) is also utilized to access registers from certain components. Debugging of the Leon3 processor is performed either through the universal asynchronous receiver/transmitter (UART) link connected to the debug support unit (DSU) or the through the Joint Test Action Group (JTAG) link. During radiation testing the UART link was utilized since the JTAG link was inoperable during scrubbing. The general-purpose input/output (GPIO) is used to transfer status information and the Leon heartbeat out to FUNCMON.

3.1.2 Licensed Leon3 Configuration

The configuration of the Leon3FT is shown in Figure 4 and is very similar to the open-source design, but now includes fault-tolerant components.

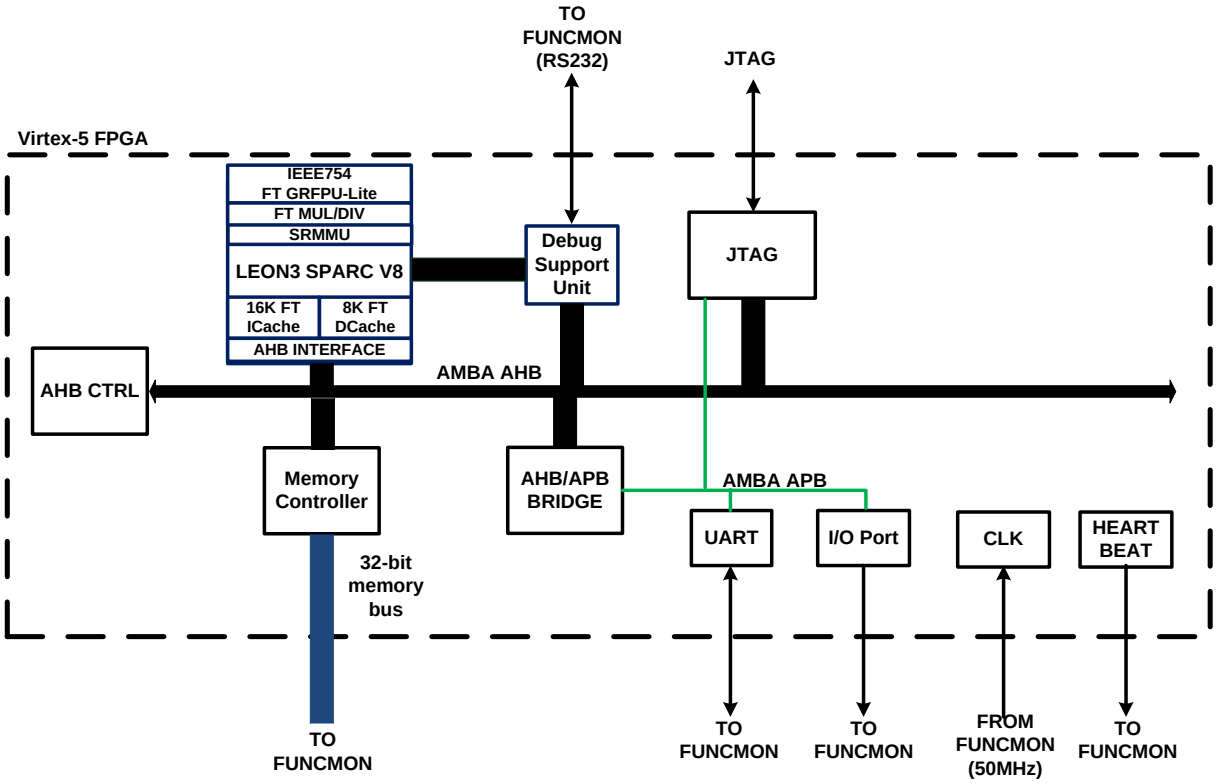

Figure 4. Licensed Leon3 configuration.

The fault-tolerant Leon3 provides mitigation to the user memory, which utilizes on-chip RAM blocks. This includes the IU and FPU register files and the cache memory. Since the fault-tolerant Leon3 is closed-source, Aeroflex-Gaisler provides a preconfigured netlist. The current preconfigured netlist used during testing includes a memory management unit (MMU), a 16-KB instruction cache, and an 8-KB data cache. A request can be made to Gaisler to change the cache size if needed.

3.2 Software Test Setup

To test the functionality of the Leon3 within a radiation environment, various software routines were created to exercise each part of the processor. For example, in order to test the IU and cache, a software routine was created to perform multiply, divide, and unsigned divide instructions in a continuous loop. Within this loop the cache was continuously written to and read from. The basic flow of this software routine is seen in Figure 5.

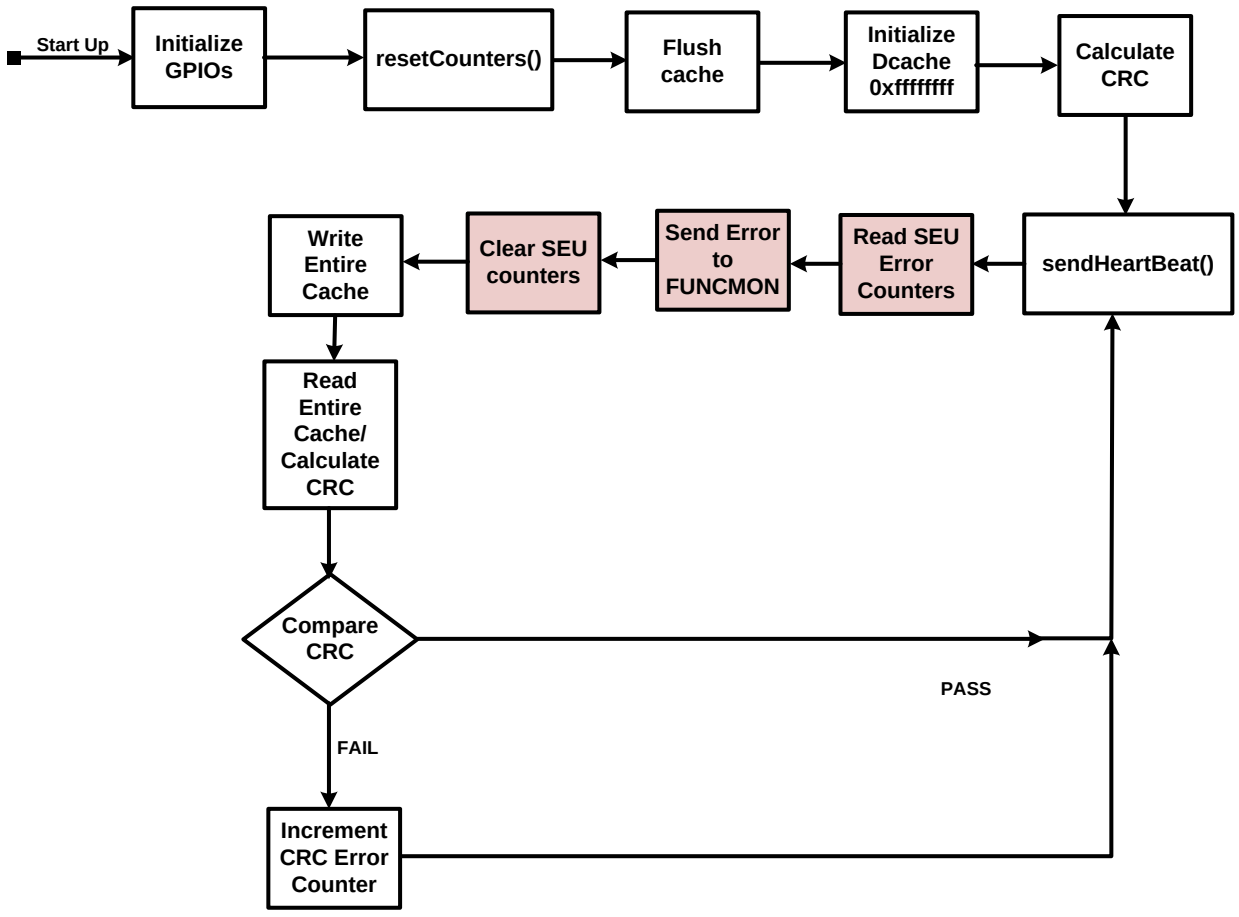

Figure 5. Register file/cache testing flow diagram.

The highlighted blocks in red shown in Figure 5 were used only during the fault-tolerant Leon3 testing. The fault-tolerant Leon3 provides SEU registers that monitor when an error was detected and corrected. A counter value is stored within these registers, which are read during testing and output to FUNCMON to display.

In order to avoid executable corruption within the DUT's on-chip memory, the executable for each software routine was stored in memory within FUNCMON's on-chip memory. This on-chip FUNCMON memory acts as pseudo-SRAM to the Leon3. The executable was downloaded, run, and reset with the use of the GRMON debugging tool. This tool uses the UART to connect to the Leon3 processor and provide debugging capabilities.

As radiation testing progressed, a software scrubber was added to the software. The software scrubber is a recursive function used to spill out all register windows to the stack (external memory), which corrects them in the process. This helps to avoid error accumulation due to the high flux used during testing. The scrubber is necessary in the designs since no operating system (OS) is used and these applications are single-threaded. The reason that this is needed is because an OS would natively access the register file whereas a raw software application will not unless that capability is added. This scrubber also adds an easy way to test the register window since the scrubber is performing a Fibonacci eight levels deep.

Figure 6 shows the addition of the software scrubber to register file/cache test.

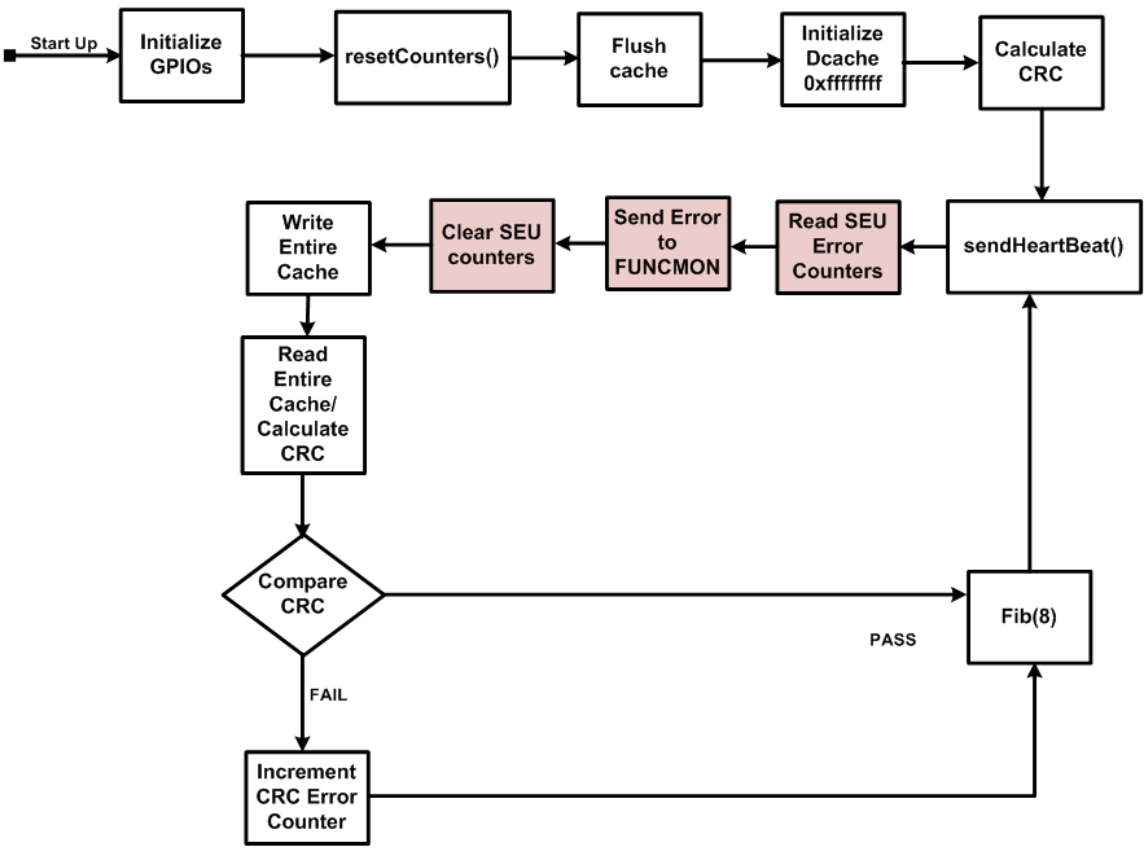

Figure 6. Software scrubber flow diagram.

4 RESULTS

Testing of the Leon3 was conducted at the Texas A&M University (TAMU) cyclotron facility and at the Lawrence Berkeley National Laboratory (LBNL) from September 2010 to July 2011. Testing used the Xilinx Radiation Test Consortium (XRTC) motherboard developed by SEAKR and the daughter card developed by Sandia National Laboratories in conjunction with the XRTC. The FPGA, motherboard, and daughter card are all shown in Figure 7.

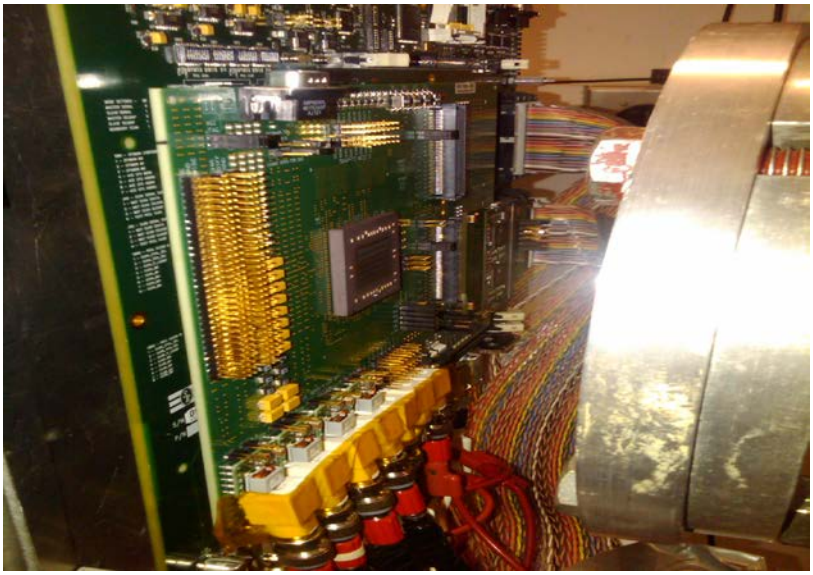

Figure 7. Test setup.

4.1 Experiments Tested

During testing, the criteria for a successful test or a failure was a basic binary pass/fail. A pass is when the Leon3 can recover either through an SEU mitigation scheme or with a reset. A fail is when the Leon3 stops responding.

To test this criteria, three various Leon3 designs were tested at the radiation facility. These included the following:

- Unmitigated open-source Leon3 – use the results as a baseline when comparing licensed and mitigated open-source designs.
- Licensed Leon3 – assess radiation hardness of this core.
- Mitigated open-source Leon3 – design attempts to provide basic mitigation to the open-source Leon3.

The mitigated open-source Leon3 consisted of two designs. The first design used the error-correcting code (ECC) on-chip memory available on the V5QV. The ECC memory is only available as a 64-bit simple dual port (SDP) memory. Since the Leon3 design utilizes only 32 bits of the 64-bit memory, the remaining unused bits could create invalid error detections in

the design. Unused bits can still accumulate errors that could leave to false detections if the actual user data is uncorrupted. A few issues with this is that time could be spent correcting unused bits and in determining whether or not the actual user data was corrupted and corrected for reporting during testing.

The second mitigated design replaced block RAM (BRAM) with LUT-RAM. LUT-RAM is constructed from the CLB LUTS and is therefore part of the radiation-hardened portion of the V5QV. Therefore, cache and the FPU and IU registers should be protected. Unfortunately, this design is not very practical due to the large increase in LUT-RAM utilized. As can be seen by the resource utilization shown in Figure 8, 71% of the V5QV's LUT-RAMs are utilized when using this design as opposed to 9% BRAM utilized in the open-source Leon3 and 7% BRAM used in the Leon3FT. The smaller use of BRAM in the Leon3FT compared to the open-source Leon3 is due to the smaller cache. There is also a slight increase in BRAM utilization within the ECC-BRAM design due to the use of the 64-bit SDP memory.

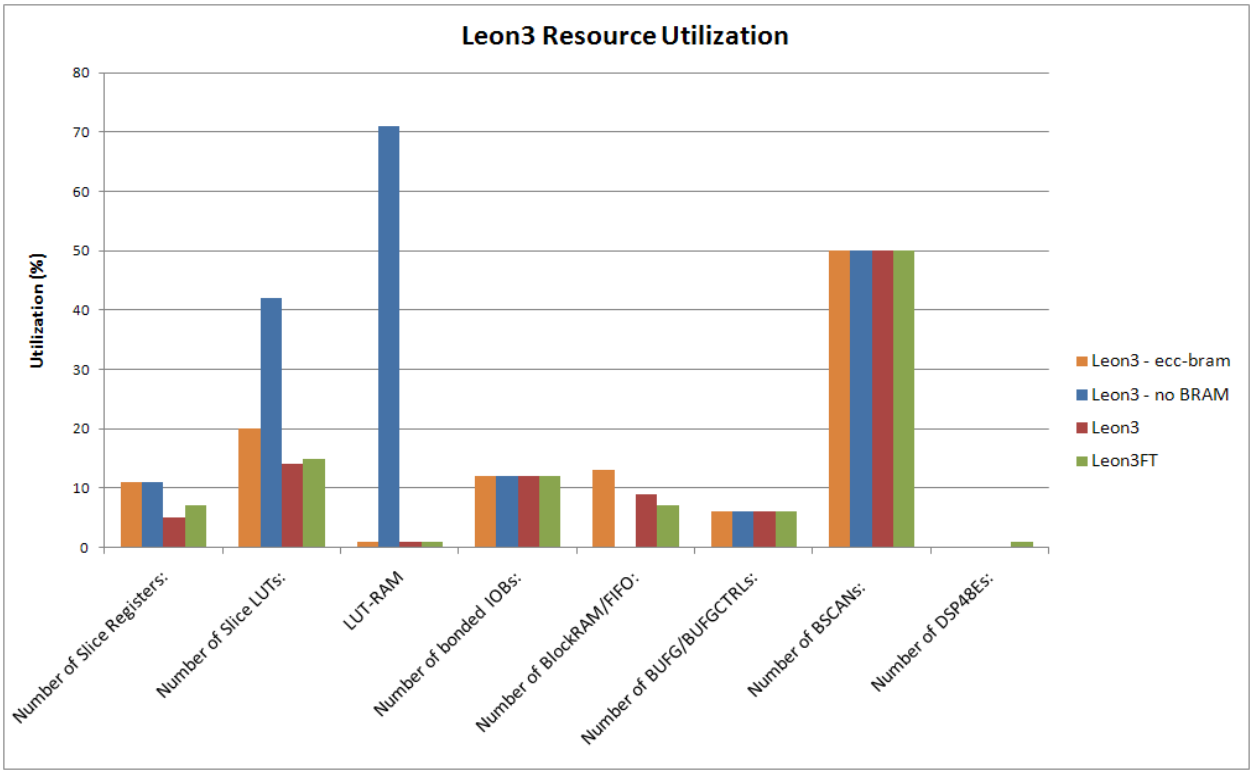

Figure 8. Resource utilization.

As seen in Figure 8, a small amount of DSP48Es are utilized within the Leon3FT. Due to the netlist provided, the DSP48E utilization could not be removed from the design as it was with the open-source Leon3 designs.

4.2 Radiation Results

Initial testing of the Leon3 processor consisted of enabling and disabling certain functionality of the Leon3 processor. For example, during an IU test, the cache and FPU were disabled. The purpose of this testing was to determine which functionality of the Leon3 provided the most upsets and which would require the most mitigation in a radiation environment. Also, no software scrubber was present during this testing.

The cross section for this initial testing conducted in February 2010 and May 2010 is shown in Figure 9.

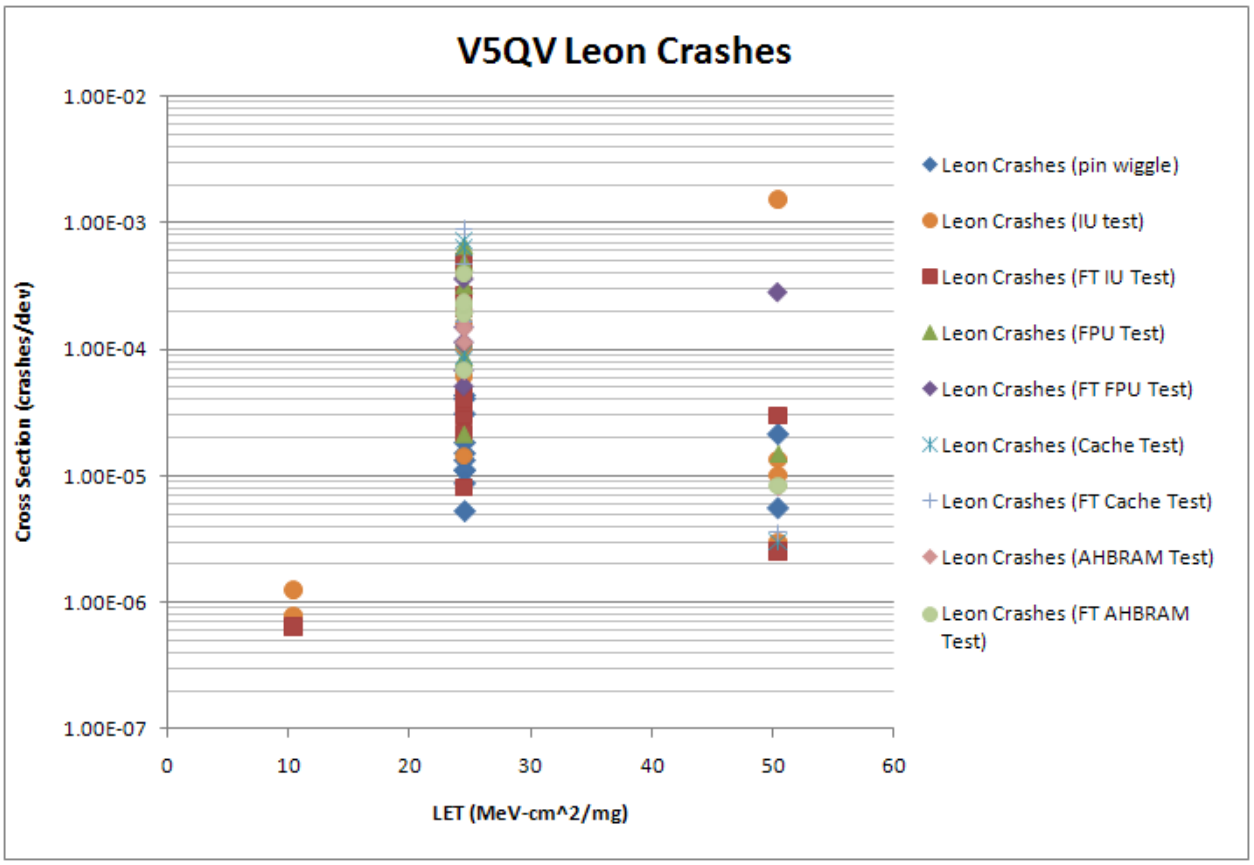

Figure 9. Initial testing cross section.

Each data point within the figure is for each run. Each individual run is presented in this figure to illustrate that the cross section between the fault-tolerant Leon3 and the unmitigated Leon3 varied only slightly.

The cross section for each data point was calculated based off the number of failures seen over the total fluence. In this case the number of failures was one, which is the number of crashes from the Leon3. A crash is considered as the Leon3 stops responding and will not recover with a reset.

Note that at a linear energy transfer (LET) of 24.5, the majority of the data points for each run are stacked on top of each other. The reason for this is because this experiment consisted of storing the software routines within read-only memory (ROM) on the DUT. Due to the large cross section of the ROM, the Leon3 quickly stopped responding, providing a poor cross section. The next two experiments at LET 10 and 50.4 moved the executable into external memory. Note that these data points are also stacked on top of each other. This is because these runs do not include the software scrubber mentioned previously.

The results in Figure 9 show that the unmitigated Leon3 and the fault-tolerant Leon3 perform about the same with little to no improvement when utilizing the fault-tolerant core. There are even some cases where the fault-tolerant core performs worse than the unmitigated core. This performance issue between the fault-tolerant core and the unmitigated core is due to errors accumulating within the register windows. To mitigate this issue, the software scrubber was included in all future experiments.

The addition of the software scrubber improved the reliability of the fault-tolerant Leon3 and is demonstrated in Figure 10. Unlike Figure 9, Figure 10 shows the averaged cross section for each run and Weibull curve for testing conducted in September 2010, October 2010, and December 2010. An averaged cross-section is an average of each experiment's run errors and fluence at the current LET. A Weibull curve is used to calculate estimated error rates. Also included within the figure are error bars to indicate the level of uncertainty of the measurement. Note that at an LET of 10, there is a large error bar that stretches from 1e-8 to about 3e-5. This is because no errors were detected during this experiment, causing a zero point and large error bars.

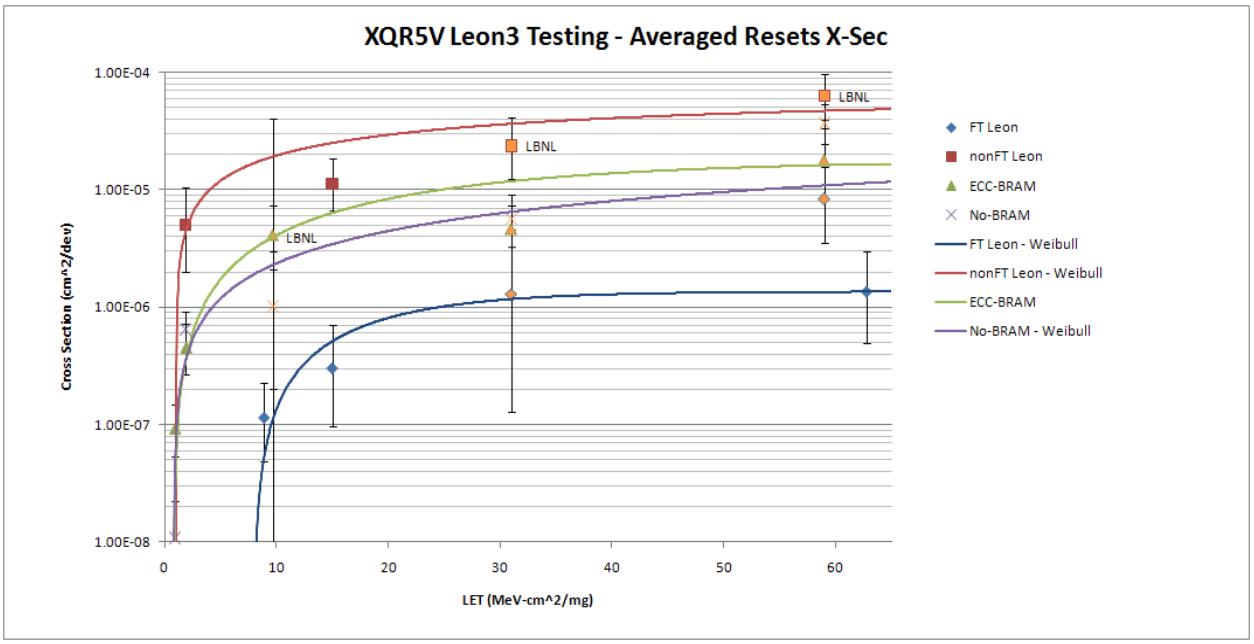

Figure 10. Leon3 software scrubber testing cross section.

Figure 10 shows that the fault-tolerant Leon3 has a much lower cross section than the open-source Leon3. The averaged cross section was calculated based off the total number of resets seen over the total fluence.

Note that these experiments include results from both the TAMU and LBNL radiation facilities. Data points highlighted in orange are from LBNL while all other colors are from TAMU. The software used during this testing was the IU/cache software test. Unlike the previous testing conducted, these experiments did not disable certain functionality of the Leon3; the FPU, cache, and IU were always enabled. The rationale for doing this was from another suggestion from Aeroflex-Gaisler that the fault-tolerant Leon3 would perform similar to the unmitigated Leon3 unless all functionality of the fault-tolerant Leon3 was enabled.

Though the mitigated open-source Leon3 designs had a lower cross section than the unmitigated open-source Leon3 design, as seen in Figure 10, the fault-tolerant Leon3 still performed much better than the other designs. For the ECC-BRAM, part of this reason that the fault-tolerant Leon3 peformed much better is that even though the ECC-BRAM was used more BRAM is used. BRAM has a large cross section, and the more BRAM that is used, the more susceptible it is to SEUs. Further study needs to be conducted to determine the cross section of the ECC-SDP BRAM to determine exactly how susceptible it is to SEUs and how accurate it is in detecting and correcting errors. The LUT-RAM design, on the other hand, relies on the V5QV to protect it. Since there is such a large use of LUT-RAM, the susceptibility to errors could be much higher. As with the ECC-BRAM, further study into the cross section of the LUT-RAM needs to be determined in order to see how susceptible LUT-RAM is.

Finally, to determine how well the Leon3 performed running various tests in the beam, two additional software tests were used. These included a real-world application and a floating-point application. The real-world application was originally implemented on the Xilinx MicoBlaze soft-core processor for the Materials International Space Station Experiment (MISSE) and includes floating-point applications. The second software test was a dedicated floating-point application to exercise the FPU within the Leon3.

These results are shown in Figure 11.

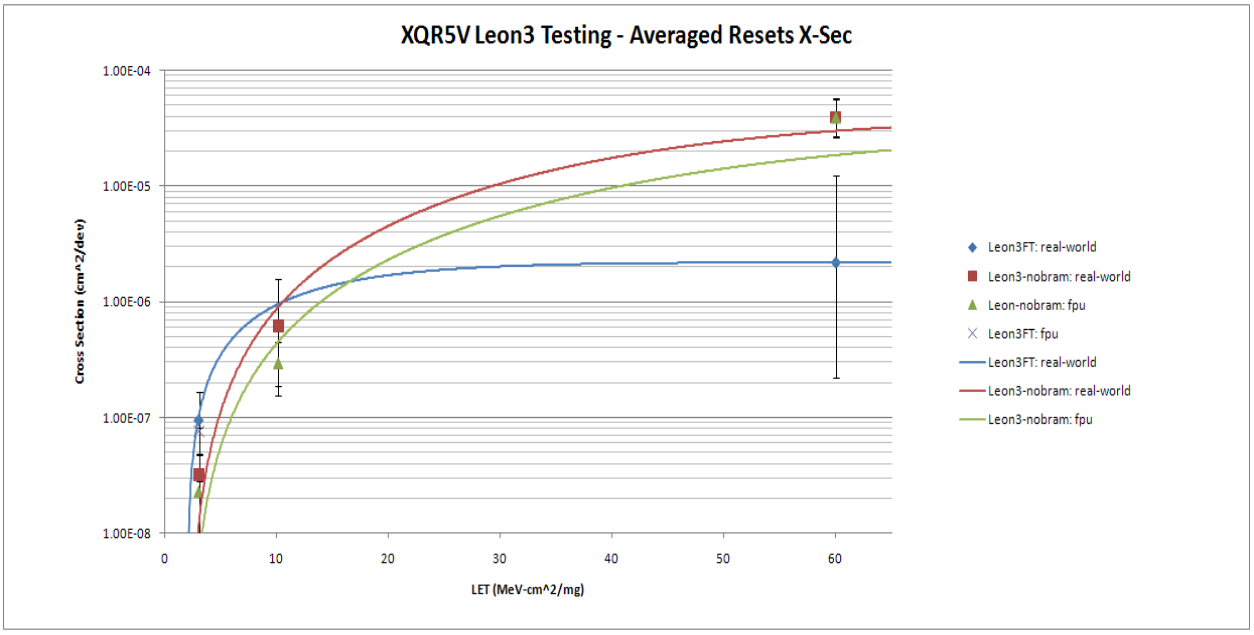

Figure 11. Additional software testing cross section.

Due to limited time available for testing during the March 2011 TAMU testing, the software was only executed on the Leon3FT and the LUT-RAM only designs. Further data points are required in order to get an accurate cross section. Initially, though, the Leon3FT still appears to have a lower cross section than the other design.

During each experiment for all the results shown, the dominant software issue seen was traps. This was mainly due to multi-bit errors that occurred with utilizing such a high flux. Further improvement in trap handling needs to be implemented in order to avoid the amount of resets seen during testing. With the addition of improvements in trap handling, the error rates seen could be reduced.

4.2.1 Error Rates

Based off the cross-section data and Weibull curves, an estimate on the error rates could be calculated based of the software used. The CREME96 software was used with the following parameters to calculate the necessary rates:

- Solar-Quiet Conditions – Galactic Cosmic Rays (GCR) Solar-min.
- Spacecraft Location – Near-Earth Interplanetary/Geosynchronous Orbit
- Shielding – 100 mils, Aluminum

Based off these conditions and Weibull curves, the error rate calculations are shown in Table 3.

Table 3. Error Rates Using Current Software.

Test	Upset Rate (resets/day)	Upset Rate (years/reset)
FT Leon3	IU/CACHE: 7.01218e-6 RW: 3.609e-5	IU/CACHE: 391 RW: 75.9
Unmitigated Leon3	IU/CACHE: 1.69414e-3	IU/CACHE: 1.6
Mitigated Leon3: ECC-BRAM	IU/CACHE: 2.54481e-4	IU/CACHE: 11
Mitigated Leon3: NO-BRAM	IU/CACHE: 1.89166e-4 RW: 8.277e-5 FPU: 4.6366e-5	IU/CACHE: 14.4 RW: 33 FPU: 59
Aeroflex-Gaisler: UT699[2] **(errors/device-day)**	IUTEST: 1.7e-4 errors/device-day RW: 1e-4 errors/device-day Benchmark: 5.1e-5 errors/device-day	IUTEST: 16 years/error RW: 27 years/error Benchmark: 54 years/error

Note that these results are based on the current software used during testing and could change based on the software used. For example, utilizing the IU/CACHE test, the Leon3FT needs to be reset every 391 years, whereas it would take about 76 years before having to issue a reset using the real-world application. The addition of an OS could also change these error rates, which is why further testing needs to be conducted utilizing an OS. Also, improvements to traps could also help to reduce the error rates seen.

Though these results are based off the current software used, these results give a comparison on how well the register files and caches handle SEUs. These results also give a good indication of what portion of the Leon3 is most affected by SEUs. For instance, the error rates for the unmitigated design using the IU/CACHE software test indicate that the cache is seeing more SEUs than the register file because the error rate is only 1.6 years before a reset may occur. If the register file was seeing more errors, this error rate would be a lot worse. Instead the cache is consistently replacing instructions and data due to a cache miss. Therefore, the chances are that the cache is inherently being scrubbed, removing invalid instructions and data during cache misses. The register files are also less likely to be exposed to SEUs because the cache is much larger, utilizing more BRAM than the register files. The benefit of the fault-tolerant Leon3 is that the cache is always flushed if an error is detected and thus corrects the problem, thus resulting in a lower error rate.

The table also lists the results for the UT699 ASIC Leon3 processor. Note that these results are not apples-to-apples comparisons, but are used only to demonstrate the results seen on the UT699. Each software test run on the UT699 and the test setup used are completely different than what was run on the soft-core processor.

[2] *Single Event Effects Qualification Summary for the UT699 Leon3FT Processor*, Graig Hafer.

5 CONCLUSION

Aeroflex Gaisler designed a fault-tolerant version of the Leon3 for use within an OTP FPGA. The preferred target technology is the Actel RTAX. The purpose of this study was to evaluate how well the fault-tolerant Leon3 would perform outside of its intended platform. This evaluation would then give us some insight on potential mitigation benefits in an SRAM-based FPGA by the core that was designed to be fault-tolerant on OTP FPGAs.

The fault-tolerant Leon3 performed much better than the unmitigated design and even than the two mitigated open-source designs used in this testing. As the results demonstrated, using the current software, the Leon3FT would require one reset every 391 years in a geostationary earth orbit if running the IU/cache test, or it would only need one reset every 76 years running a real-world application. The unmitigated Leon3 would require a reset every 1.6 years running only the IU/cache test. Finally, though there was a slight improvement with providing some basic mitigation techniques to the open-source Leon3, both designs still had higher error rates than the fault-tolerant Leon3.

During software testing, the dominant software error seen was traps. This was mainly due to multi-bit errors that occurred with utilizing such a high flux. Traps need to be correctly handled in order to avoid the amount of resets seen during testing. With the addition of improvements in trap handling, the error rates seen could be reduced.

Also, further characterization on the Leon3FT needs to be done in order to see how well the processor performs in a radiation environment while running an OS such as Real-Time Executive for Multiprocessor Systems (RTEMS). This will not only verify the need of the software scrubber in bare-metal applications, but will characterize how well the fault-tolerant Leon3 performs while executing a more complex design. Further software tests are also needed in order to get an accurate cross section on the various software tests used. In addition, further improvements to the mitigated open-source Leon3 designs should be considered, such as ways to flush the cache on the detection of an error. This would also help to reduce the error rates seen in these two designs. Finally, further effort needs to be made to the software applications in order to determine if the software routines are indeed performing their correct operations; for example, if the calculations being performed are returning correct results.

Based off the current results, the Leon3FT appeared to operate quite well on the SRAM-based FPGA, though, as indicated previously, further study is required.

DISTRIBUTION

1	MS0503	James W. Daniels	5337
1	MS0503	Dominic A. Perea	5337
1	MS0503	Mythi M. To	5337
1	MS0503	Christopher K. Wojahn	5337
1	MS0503	Donald E. Tolsch	5339
1	MS0513	Richard D. Hunt	5336
1	MS0661	Daniel E. Gallegos	2623
1	MS0661	Mark W. Learn	2623
1	MS0661	Aaron D. Niese	2623
1	MS0672	Brian P. Van Leeuwen	5628
1	MS0971	Ethan L. Blansett	5735
1	MS0980	Jay F. Jakubczak	5710
1	MS0980	Matt P. Napier	5571
1	MS0982	Jaime Gomez	5732
1	MS0982	Dan J. Kral	5732
1	MS0986	David M. Bullington	2664
1	MS0986	Jonathon W. Donaldson	2664
1	MS0986	Justin W. Enderle	2664
1	MS0986	David Heine	2664
1	MS0986	Jeffrey L. Kalb	2664
1	MS0986	David S. Lee	2664
1	MS0986	J. (Heidi) Ruffner	2664
1	MS1027	John M. Eldridge	5632
1	MS0359	D. Chavez, LDRD Office	1911
1	MS0899	RIM-Reports Management	9532 *(electronic copy)*


Sandia National Laboratories