National Aeronautics and Space Administration

Microsemi PolarFire® Field Programmable Gate Array Single Event Effects Test Report

Melanie Berg – Principal Investigator: SSAI in support of NASA/GSFC
HAK Kim: SSAI in support of NASA/GSFC
Michael Campola: NASA/GSFC
Jonathan Pellish: NASA/GSFC

Test Date: 11/18/2019 Lawrence Berkeley National Laboratory (LBNL) 88-inch Cyclotron
Report Date: 12/15/2019

Revision Number	Submission Date	Comment
20191212 – SEE_Test_Report_PolarFire - Rev1		

Table of Contents

Table of Figures:

Table of Tables:

1. INTRODUCTION

The purpose of this study was to investigate Microsemi PolarFire® field programmable gate array (FPGA) single event effect (SEE) responses while the devices were exposed to heavy-ion particles. This document references the Microsemi PolarFire® as the device under test (DUT). DUT susceptibilities were determined by monitoring the DUT's electrical parametrics, internal components, and dynamic function for single-event transient (SET), single-event upset (SEU), single-event functional interrupt (SEFI), single-event latchup (SEL), and single-event failure (SEF) induced faults. General guidelines for FPGA SEE testing [1][2] were incorporated throughout the evaluation process.

It is important to note that the study was limited and was intended to be a first-look into DUT SEE susceptibility.

2. DEVICE SPECIFICS

Table 1: Microsemi PolarFire® Logic Elements per Device Type

Resources	MPF100	MPF200	MPF300	MPF500
Logic elements	71,736	127,896	198,744	319,992
Interface logic	36,864	64,512	100,800	161,280
Total Logic	108,600	192,408	299,544	481,272
LSRAM blocks (20 Kb each)	352	616	952	1,520
Total LSRAM bits (Mb)	6.87	12.03	18.59	29.69
µSRAM blocks (768 bits each)	1,008	1,764	2,772	4,440
Total µSRAM bits (Mb)	0.74	1.29	2.03	3.25
Total RAM (Mb)	7.6	13.32	20.62	32.94
Math blocks (18 × 18 MACC)	336	588	924	1,480
µPROM (Kb)	297	297	459	513

PolarFire® FPGA devices are fabricated with 28 nm technology. The FPGA configuration is built using flash memory (reprogrammable and non-volatile configuration). The programmable user fabric consists of the following resources (specific numbers are listed in Table 1):

- **Logic elements**—these are basic building blocks in the PolarFire® FPGA.
- **Embedded memory blocks**—these include large SRAM (LSRAM), microSRAM (µSRAM), microPROM (µPROM), and secure non-volatile memory (sNVM).
- **Math blocks (MACC)**—these have a built-in multiplier, pre-adder, and adder.

There are no embedded triple modular redundant (TMR) components. However, the embedded memory blocks have the option to use embedded single error correct double error detect (SECDED) circuitry. Figure 1 is a block-level illustration of the programmable user fabric.

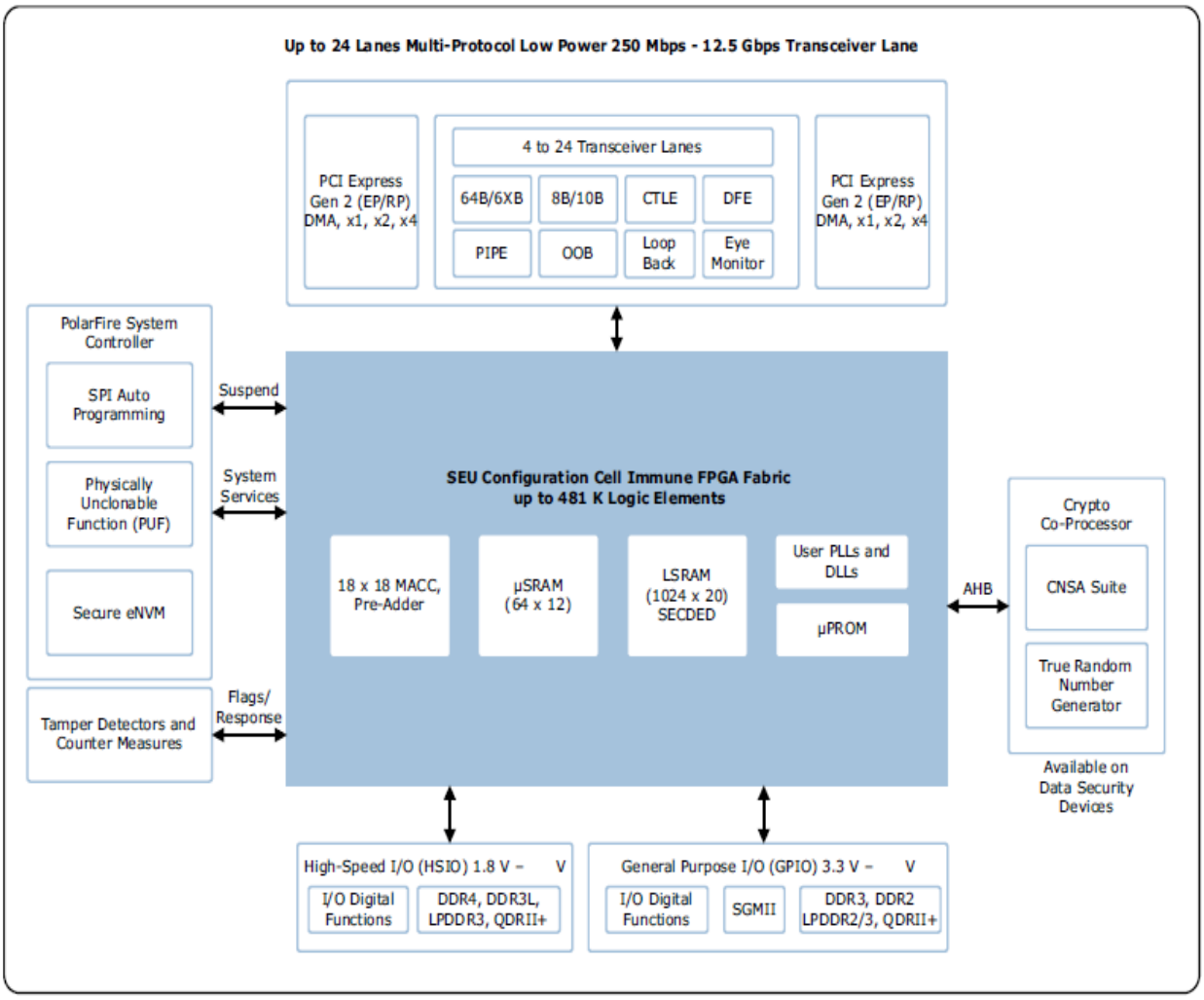


Figure 1: PolarFire® block diagram (programmable user fabric)

3. BACKGROUND OF FLASH-BASED FPGA TEST METHODOLOGY

Analyzing FLASH-based FPGA susceptibilities to heavy-ions requires the user to investigate SEU behavior and error contribution from the following internal device components: Configuration, data path, global structures and hidden logic (single-event functional interrupt (SEFI) susceptibility). This section describes the test methodology that will be used to evaluate the PolarFire® DUT.

3.1 General Test Approach

An FPGA is a complex device containing an assortment of components with varying susceptibilities to SEUs. FPGA heavy-ion SEE experiments produce data that characterize SEU susceptibilities by using metrics relating SEU responses to particle linear energy transfer (LET) values. There are experiments that produce countable events per test and those that run until failure (i.e., one event per test). The details of each are explained in Table 2.

Table 2: SEE Experiment Catagorization

Experiment Reference Type	General Explanation of Experiments	Measurement
A	• Embedded component level testing targeted to evaluate device specific SEE responses (general susceptibilities… not design specific susceptibilities). • Test cases are designed to monitor a variety of potential SEE error signatures. • Objective is to study SEE contributions of FPGA embedded components. Information is used to guide SEE data extrapolation for reliability/survivability prediction and analysis of tactical systems. However, it is important to note that data are not easily extrapolated for characterizing tactical reliability/survivability.	Countable errors (cross-sections(σ))
B	• Test cases target design specific SEF responses. • Additional observation points added to device internals to gain insight into internal failures that do not propagate to I/O • Objective is to evaluate, based on particle exposure, when the mapped DUT-design fails to operate (failure defined per type of design being investigated. Data are better extrapolated to tactical designs.	Mean-fluence-to-failure (MFTF = $1/\sigma$)

3.2 Single Event (Test Data) Metrics

Metric nomenclature and definitions used in this document can be found in Table 3.

Table 3: Description of Cross-Sections and Their Metrics

Cross-Section	Description	Metric
σ_{sel}	FPGA latch-up	A current jump will occur per latch-up. Tests continue and the number of SEL events can be counted over the total number of particles during test as in (1)
σ_{sef}	Design failure	MFTF as in (2)
$\sigma_{configuration}$	Configuration single-bit flips	The number of configuration bit upsets per number of particles of exposure as in (1)
σ_{bram}	BRAM single-bit flips	The number of BRAM bit upsets per number of particles of exposure as in (1)
$\sigma_{functionalLogic}$	Data path elements	Depending on the test-design, can be measured in MFTF (2) or (1)
σ_{sefi}	Global routes/buffers, hidden logic (analog, configuration scrubber technology, flip-flops, combinatorial logic, inaccessible memory, etc…), and large blocks of affected configuration or BRAM	MFTF as in (2)

Conventionally, SEU responses are evaluated per selected particle linear energy transfer (LET) values; and the metric for SEU characterization is an SEU cross-section ($\sigma_{seu}(LET)$). $\sigma_{seu}(LET)$ is defined as the number of upsets per particle exposure at a given LET as shown in Eqn. 1.

$$\sigma(LET)_{SEU} = \frac{\#errors}{\#Particles/cm^2} \qquad (1)$$

There are some tests (such as configuration tests) where the number of errors accumulated within an experiment will be greater than 1. Alternatively, there will be some experiments that will stop at failure (i.e., #errors=1). This specific type of testing uses Eqn. 1 but also uses a metric called mean-fluence-to-failure (MFTF). It is the inverse of the cross-section when the number of errors is one as shown in Eqn. 2.

$$MFTF = \frac{1}{\sigma(LET)_{seu}} = \#Particles/cm^2 \qquad (2)$$

To optimize the integrity of MFTF data, these experiments require special hardware to synchronize the beam fluence on/off signal to the test equipment. This phase of testing used an Arduino board that posed as a level shifter from the facility's beam indicator signal to the FPGA on the tester board (Tester-ML605 board).

$\sigma_{configuration}$, $\sigma_{functionalLogic}$, and σ_{sefi} evaluations fall under Type-A experiments (see Table 2). Subsequently, they are investigated independently by strategic tests meant to target each component's cross-section. The purpose of Type-A experiments is to study dominant mechanisms of failure and to analyze potential component error signatures in response to beam exposure.

σ_{sef} characterizes a user's tactical design. It is a function of design topology and how it utilizes the DUT's internal components as shown in Eqn. 3.

$$\sigma_{sef} = f\left(\sigma_{configuration}, \sigma_{bram}, \sigma_{functionalLogic}, \sigma_{HiddenLogic}\right) \text{ per design} \quad (3)$$

The FPGA components and their $\sigma_{seu}(LET)$ can be categorized as in Table 3 (each σ_{sef} is assumed to be per LET in this document and hence does not contain the LET parameter).

3.3 Configuration Test and Analysis

Configuration SEUs are checked by performing a verify function via Microsemi Libero software. The software does not return an error count. It returns a pass-fail indication. Being limited to a pass-fail indicator versus obtaining a specific memory cell error count per test is deemed an acceptable evaluation (of configuration) because flash-based configuration is considered to have negligible susceptibility to SEUs. This has been shown as a plausible approach (for NEPP and Microsemi) with previous Microsemi RTG4 and Microsemi ProASIC3 SEU testing.

It is important to note that configuration upsets may or may not cause a design (system level) error. If a configuration bit, that has changed state because of an SEU, is being used by the design and is in an enabled path of logic, then there is a chance that the changed state will cause system malfunction. Dynamic testing is required to investigate configuration upsets and their impact on system level functionality; i.e., this is not considered a configuration test – it is considered a dynamic functional test. In general, Configuration SEUs that directly affect active circuitry have error signatures that resemble stuck faults. This is because the flash configuration cannot be scrubbed; nor will a power cycle correct non-volatile configuration errors. The device will have to be reconfigured in order to correct configuration SEUs.

3.3.1 Dynamic Test and Configuration Analysis

Dynamic testing concentrates on evaluating the SEU susceptibility of the design under analysis (DUA) during accelerated radiation testing. Such testing requires the DUT to be operating and its outputs to be monitored during irradiation. Deviations from expected values are reported and noted as error responses.

This is a challenging study because, as previously mentioned, the complexity of the FPGA devices has various components with different error signatures and susceptibilities. The component contribution to overall design σ_{sef}s was given in equation (3). If tests are not constructed to isolate these components during the test and analysis phase, radiation data can be convoluted and difficult to process.

4. TEST SAMPLES AND EXPERIMENT PREPARATION

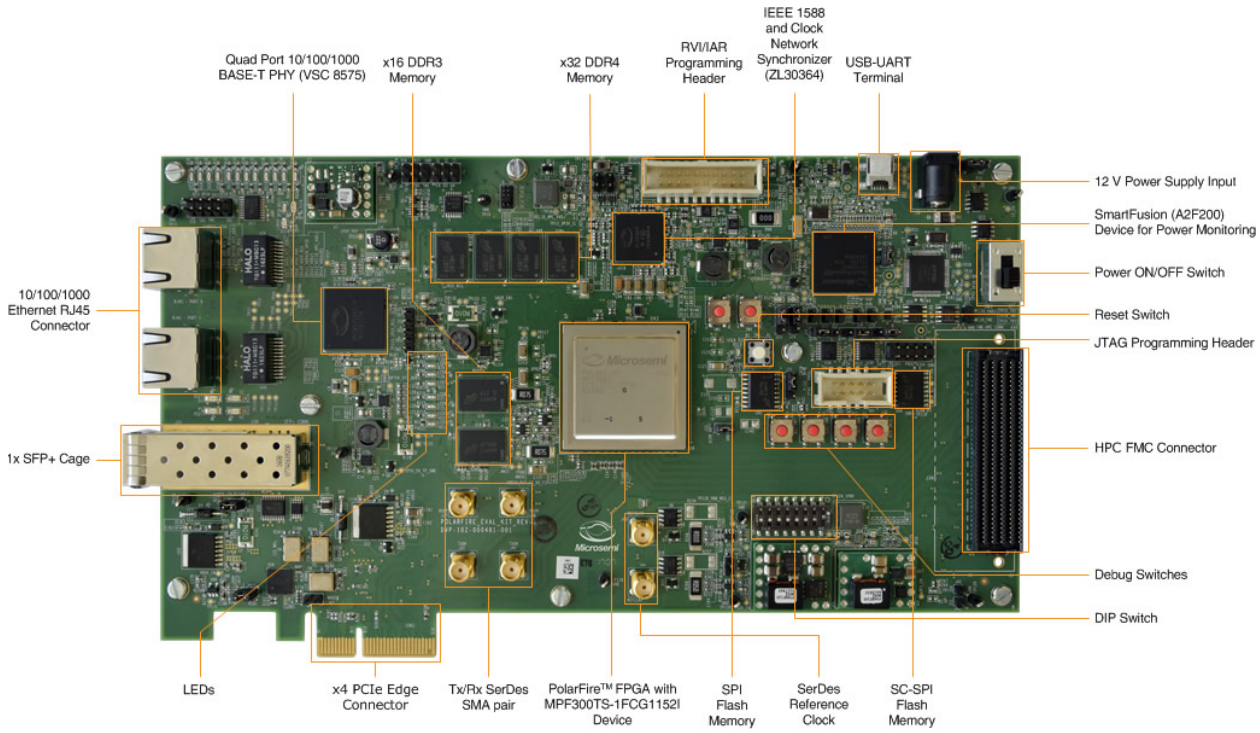


Figure 2: PolarFire® Daughterboard

NEPP FPGA test systems contain a Motherboard (DUT-control, DUT-monitor, and DUT-data-collector), and a Daughterboard (DUT-housing). For this study, the Daughterboards were the MPF300-EVAL-KIT PolarFire® Evaluation boards; and were provided by Microsemi. Figure 2 is a picture of the PolarFire® evaluation board.

List of components contained on the PolarFire® Daughterboard:

- 300K LE PolarFire FPGA in an FCG1152 package (MPF300TS-1FCG1152I)
- HPC FMC connector
- 1x SFP+ cage
- IEEE1588 PLL
- SMA connectors for testing of full-duplex 12.7 Gbps SERDES channel
- 4 GB DDR4 x32 and 2 GB DDR3 x16
- PCI Express (x4) edge connector
- 2 x RJ45 for 10/100/1000 Ethernet using SGMII on GPIO
- Dual 10/100/1000BASE-T PHY (VSC8575) for synchronous ethernet (SyncE) and 1588 application
- SATA interface
- Power management unit for 1 or 1.05 V PolarFire FPGA core voltage
- USB to UART interface
- Embedded programming and debugging using SPI and JTAG
- On-board power monitoring
- 2 x 1 Gb SPI Flash memory

For this first-look study, only one board was prepared for SEE testing. DUT specific information can be found in Table 4.

Table 4: PolarFire® Part Identification Information

Part Number	MPF300TS-1FCG1152EES
Manufacturer	Microsemi
Lot Date Code	n/a
Additional Case Markings	n/a
Quantity Tested	1
Part Function	Reprogrammable FPGA
Part Technology	28 nm Flash Configuration FPGA
Package	FCG1152

The first-look DUT was thinned using mechanical etching via an Ultra Tec ASAP-1 device preparation system (illustrated in Figure 3). The part was successfully thinned to 100um–120um.

Figure 3: Ultra Tec ASAP-1 device preparation system

5. TEST SYSTEM

5.1 Overview

NEPP FPGA test systems contain two primary printed circuit boards (PCBs):

- **Daughterboard:** contains the DUT and other essential components (power control, memory, oscillators, connectors, etc.)
- **Motherboard:** Complex board that contains a central controller (reprogrammable FPGA) and supporting components.

NEPP created a new test system Motherboard using the Xilinx KCU105 Evaluation board. The central component of the Motherboard is the Kintex-UltraScale (XCKU040-2FFVA1156E) FPGA. The Motherboard also includes two FPGA Mezzanine Card (**FMC**) high-speed connectors. For this test system only one Motherboard FMC connector was used. It was the primary interface between the Motherboard and the target DUT-Daughterboard.

Because the Motherboard FPGA is reprogrammable, it is possible to customize control/monitors (test system designs) and download them to the Motherboard FPGA per experiment type. This enables specialized control and monitoring of hundreds of DUT I/O at speeds of MHz-GHz. Subsequently, the NEPP test harness is significantly more powerful than a processor or microcontroller.

5.2 Test System Setup

The DUT block diagrams of the test system are illustrated in Figure 4 and Figure 5. Pictures of test system hardware with FMC connection is shown in Figure 6 and Figure 7.

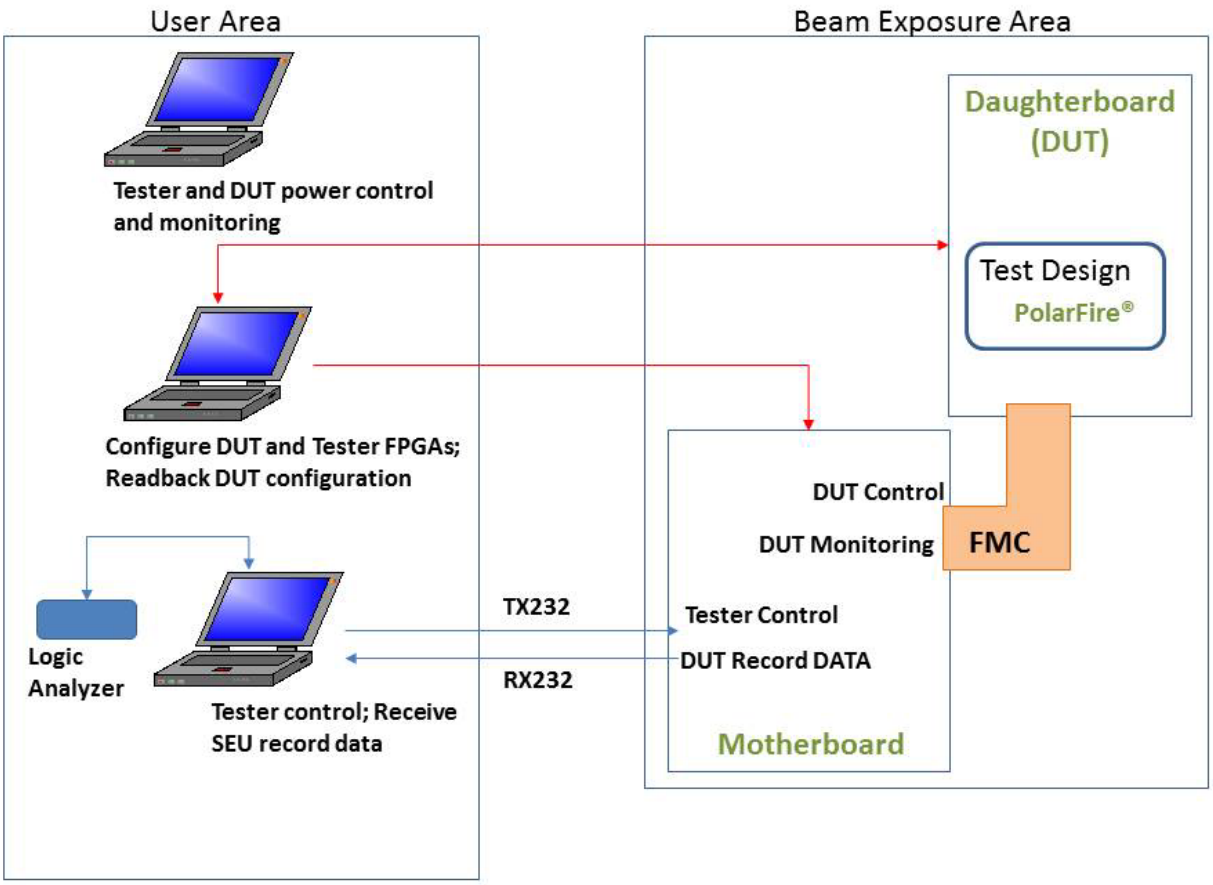


Figure 4: Test Harness includes Motherboard, PolarFire® Daughterboard, and a variety of PCs for real-time monitoring of DUT responses during applied stress conditions.

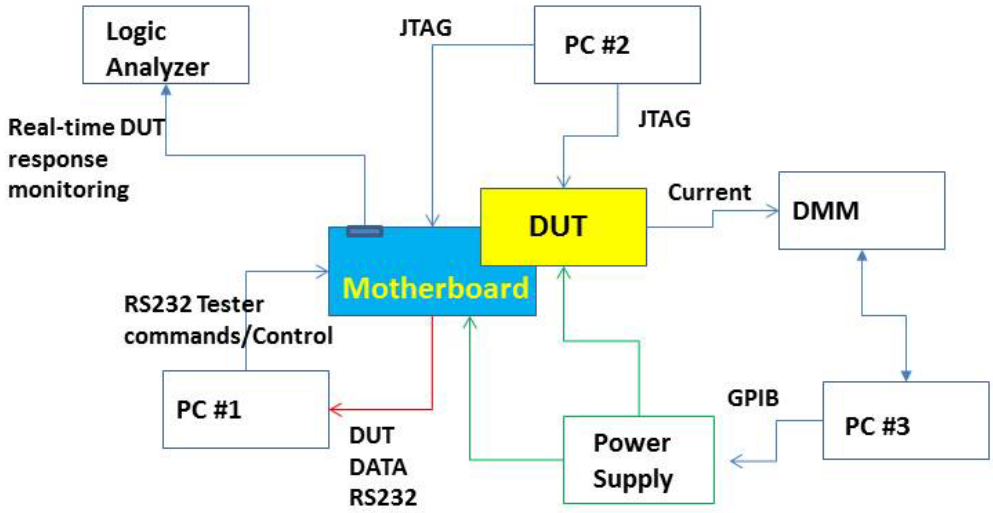


Figure 5: Test harness command control and monitor system. A digital multimeter (DMM) is used to measure current (per voltage plane) and DUT board temperature.

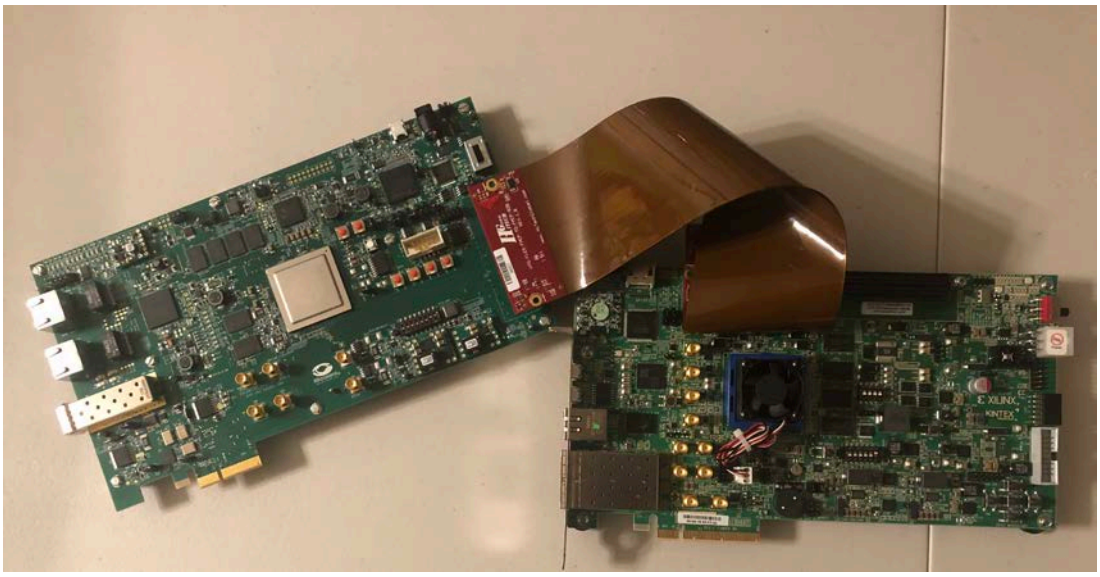


Figure 6: Motherboard (KCU105) and Daughterboard (PolarFire®) connection. FMC Connector is bent to show Motherboard FPGA

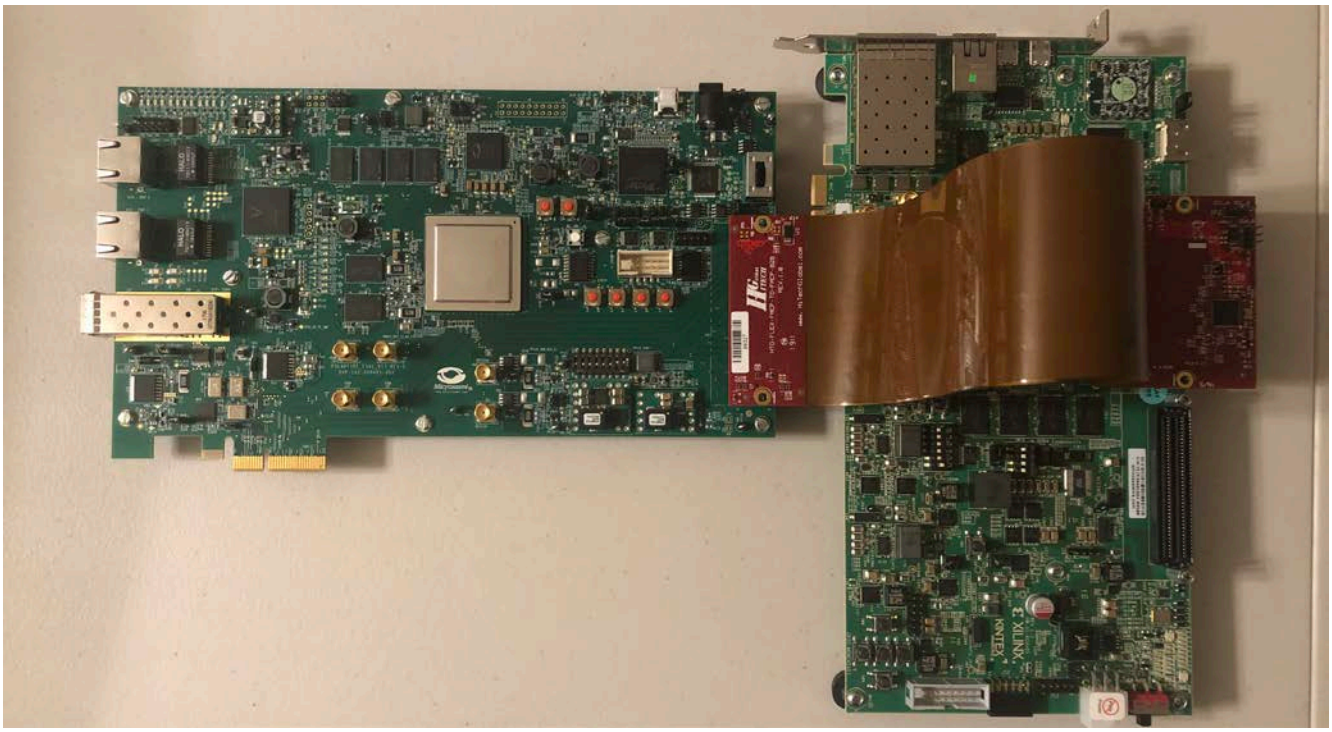


Figure 7: Motherboard (KCU105) and Daughterboard (PolarFire®) connection. FMC Connector is placed in actual position for SEE testing

The following provides descriptions of test system equipment as depicted in Figure 5:

- Motherboard (FPGA Based Controller/Processor): The Motherboard contains mapped designs that are responsible for controlling and monitoring DUT activity, receiving commands from a host computer, processing data, and packetizing/reporting DUT behavior to a host computer and logic analyzer.
- Daughterboard (containing the DUT and its associated board-level circuitry): Test designs (firmware) are mapped into the daughter board DUT for SEE evaluation.
- Host PC1:
    - Configures Daughterboard FPGA via USB-JTAG (Microsemi Libero software)
    - Readback Daughterboard FPGA configuration via USB-JTAG (Libero software)
- Host PC2:
    - Configures Motherboard FPGA via USB-JTAG (IMPACT software).
    - Manages and displays logic analyzer Data.
    - Manages Power Supplies
    - Sends commands to Motherboard via USB-UART.
    - Reads DUT data packets via USB-UART.
    - Reads and displays DUT PMBus (current, voltage, and temperature).
- Host PC3:
    - Monitors DMM: Voltage/current monitors are placed on each voltage plane using a DMM. Voltage/current data is observed visually during beam exposure (using a NEPP custom made LabVIEW data graphing tool), however, data capture/storage is also automated for post-processing. The granularity of voltage/current monitoring and recording is in the order of *ms.*
    - Controls power supply of the Daughterboard and Motherboard
- Power supplies: The DUT and mother board receive separate 12V power rails from separate power supplies. Each power supply is controlled through its GPIO port via a host computer to GPIB.
- Logic Analyzer Pod: The logic analyzer pod is capable of 500 MHz sampling rates and is used to perform real-time visualization of the DUT responses. Motherboard monitors DUT responses and directly sends data to the logic analyzer.
- Customized Beam Indicator level shifter: This circuit is used to synchronize the output of the beam on/off signal to the Motherboard tester. The circuit connects the output of the cyclotron beam reference signal to the Motherboard. This signal is essential for MFTF particle (fluence) counts.

5.3 Motherboard Overview

As previously mentioned, the test system consists of a motherboard (FPGA Based Controller/Processor) and a daughter board (containing the DUT and its associated necessary circuitry). The DUT controller/processor is instantiated as a subcomponent within the Motherboard FPGA (Kintex-UltraScale).

The objective of this DUT controller/processor is to supply inputs to the DUT device and perform data processing on the outputs of the DUT. A customized control and data handler (C&DH) was designed to perform streamline tasked specifically for SEE testing. The Motherboard's C&DH communicates with a user-controlled PC (PC #1). The user PC interface is LabVIEW. LabVIEW code was designed to send user specified commands to the Motherboard and receive information from the Motherboard.

5.3.1 RS232 communication from the Motherboard Command and Data Handler to the Host PC

All RS232 communication from the Motherboard C&DH to the host PC is prefaced with a header. Information from the C&DH to the Host PC is categorized as one of the following types listed in Table 5: an alive-timer, a command echo, or an error-record.

Table 5: List of the Motherboard C&DH to Host PC RS232 Header bytes. Only the Motherboard C&DH uses header information. The host PC sends pure commands to the Motherboard-tester without headers.

Header	Description
00 FA F3 20	Alive Header: No data bytes follow (i.e. only the header is sent from the Motherboard-tester to Host PC #1)
00 FA F3 22	Command Echo: 4 data bytes follow that represent the command that was previously sent from the Host PC #1 to the Motherboard-tester.
00 FA F3 21	Data Error Record: 23 bytes follow.

5.4 RS232 communication From the Host PC to the Motherboard-tester (C&DH)

Communication from the host PC to the Motherboard C&DH does not contain a header. Information sent from the host PC to the C&DH are commands and are all 4 bytes in length. The interface is controlled by a user GUI designed with NEPP custom LabVIEW software.

5.4.1 User GUI

Commands are sent by typing specific values into LabVIEW fields or controlling LabVIEW on/off buttons listed on the screen.

5.4.2 User Interface and Command Control

The user controls the tests via a LabVIEW interface running on a PC. The PC communicates with the Motherboard with a RS232 serial link. The format of communication is a command/Data 4 byte word.

Table 6 : Summary of Commands Used in DUT Tests

Command #	Command	D0	D1	D2	Description
01	Reset Motherboard	N	N	N	Resets Tester and DUT (stops DUT clocks and makes DUT reset active)
03	Reset Counter	N	N	N	RESETS Counters without resetting tester circuitry (only used in counter tests)
02	Start Testing	N	N	N	Starts the DUT clocks
A0	Clock divider	Y	Y	N	D0 (LSB) and D1 (MSB) are the Clock frequency divider of 100 MHz. The synthesized Motherboard-tester clock output is sent to the DUT. This signal is the DUT's system clock. Default=0 (no division)
91	Pattern Selection	Y	N	N	Only used in WSR tests: D0 = 0=> all 0's D0 =1 => all 1's D0 >1 => checkerboard
06	WSR Halt command	N	N	N	Stops the DUT clocks. Does not perform a reset.

6. DUT TEST STRUCTURES AND DYNAMIC ACCELERATED TESTING

In this first-look study, we used simple test structures. Test Structure considerations were taken from the NASA Electronics Parts and Packaging (NEPP) FPGA SEU Test Guidelines: https://nepp.nasa.gov/files/23779/fpga_radiation_test_guidelines_2012.pdf .

As referenced in the FPGA SEU Test Guidelines manual, simple test structures should have the ability to flush SEU's (post-detection)… i.e., test circuits should have the ability to keep working after an SEU occurs to increase statistics of each beam run. This helps to increase statistics during testing (tests do not have to stop upon each SEU… i.e., SEUs are countable per test). In addition, the tester (for these simple test structures) should be robust enough to resynchronize to new DUT expected values caused by SEUs; i.e., expected values should be dynamic based off of impact of SEUs. These statements are not necessarily implementable in a complex system. However, in shift registers and counters; i.e., flushable test systems, it is feasible.

6.1 Overview of DUT Test Structures

Table 7 is an overview of the DUT test structures that were analyzed in heavy ion testing. Some of the test structure designs were created with variations for trend analysis.

Table 7: Overview of Test Structures

Test Structure	Frequency Range	SEU Flow-Through (Flushable)
Shift registers (WSR)	25 MHz	Yes
Counters (Count Array): Parallel independent counters	25 MHz	Yes
Embedded LSRAM	25 MHz	Yes

6.2 Windowed Shift Registers (WSRs) DUA

6.2.1 General Operation

A windowed shift register (WSR) is based on the traditional shift register used for SEU testing. The difference between a WSR and a traditional shift register is that the last 4 stages of the shift register are captured into a 4-bit window. This helps with signal integrity and reliability of data capture between the DUT and the tester. In this first-look study, no inverters were placed within the WSR stages. Figure 8 illustrates a WSR. Four (4) WSR chains were instantiated within the DUT test design.

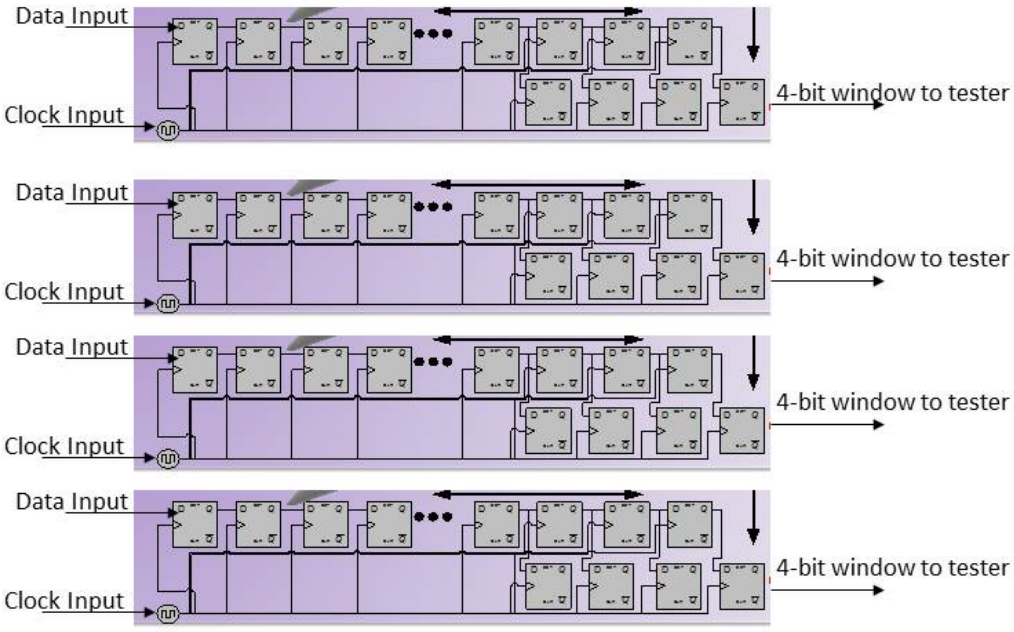


Figure 8: Four Instantiated WSR Chains. No inverters are used in this first-look study

Figure 9 is a timing diagram that illustrates WSR operation under normal conditions.

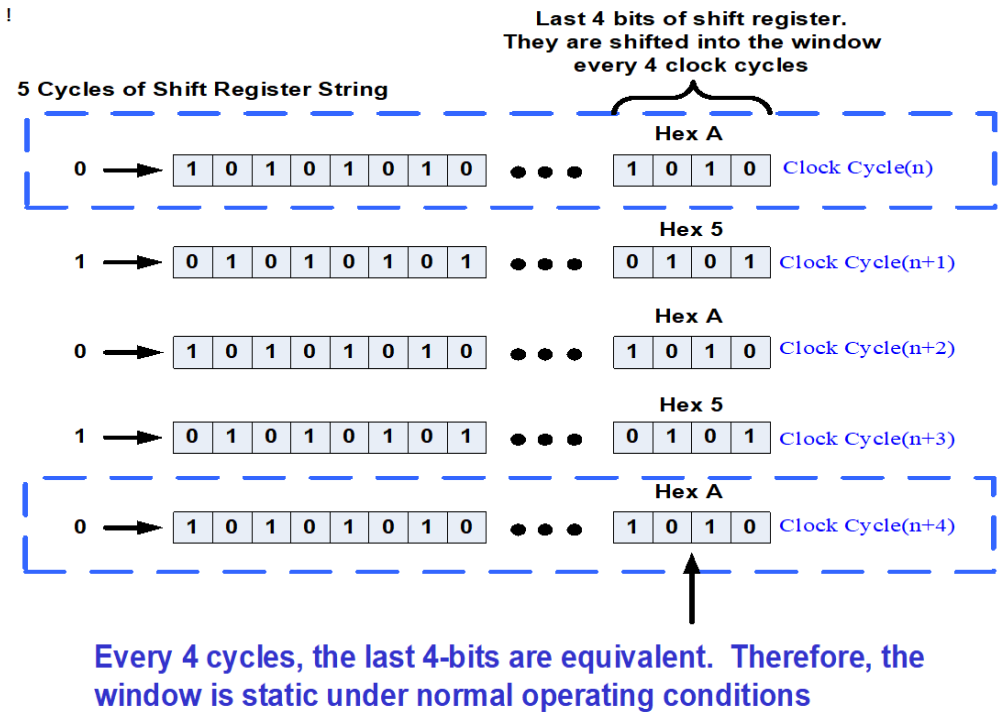


Figure 9: WSR Functional Behavior

6.2.2 WSRs and Motherboard-tester Requirements

The following are requirements for Motherboard control of the DUT WSR chains:

- Motherboard must supply DUT clocks. Clock frequency is command driven by the user and physically controlled by the Motherboard. However, for the first-look study, chains were driven at 25 MHz.
- Active high Resets (shared across all WSR chains) – RST.
- 2-bit MUX control (see Table 8) to select WSR data input pattern (however, the first-look study only used checkerboard):

Table 8: MUX Control Patterns

MUX: Bit1 Bit0	Data Pattern
00	All 0's
01	All 1's
10	Checkerboard
11	Checkerboard

6.2.3 WSR SEU Data Monitoring and Capture

- Heartbeat monitoring:
  - DUT heartbeat output is ¼ the speed of the DUT clock input.
  - DUT heartbeat is oversampled by > 4x of the heartbeat speed by the Motherboard and is passed through a metastability (synchronizer) circuit. Output of the heart detect metastability circuit is edge detected (rising). The edge is under watchdog to make sure that the system hasn't died.
- Expected data monitoring:
  - Because the window should be static, any change in the window is detected and reported to the host PC. Motherboard oversamples the DUT output window in order to potentially detect change.
- Each WSR window is monitored for data change. Data change signifies SEU.
- SEUs are differentiated: single bit, versus stuck, versus burst upsets (global).

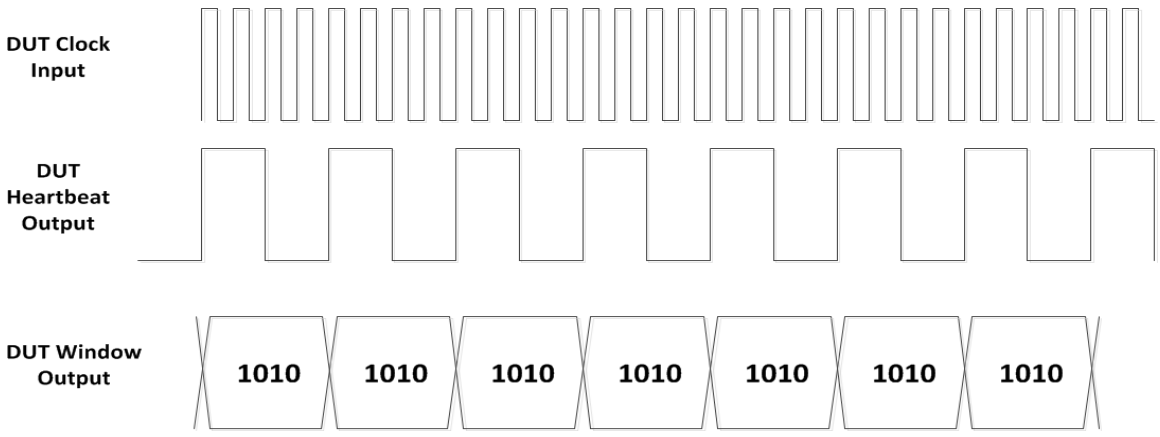
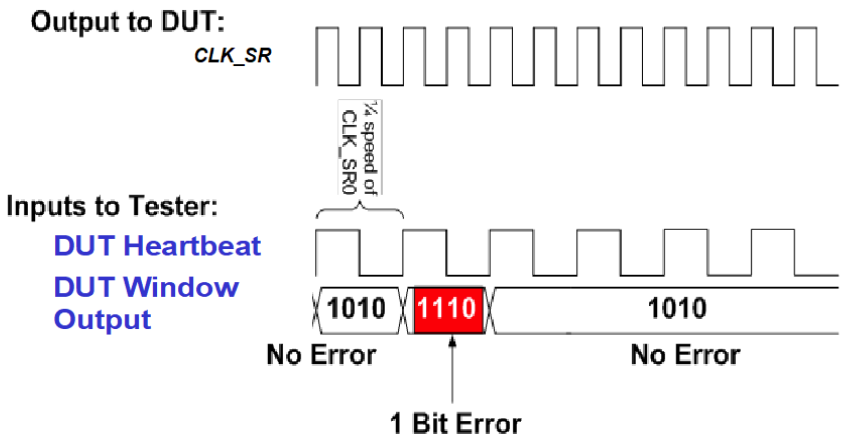


Figure 10: Timing diagram of WSR output under normal operation



WSR window stays constant unless there is a SEU. WSR provides optimal signal integrity for SEE testing

An error record is created and is sent to the Host PC

Figure 11: Timing diagram of WSR Operation with SEU

6.2.4 WSR Error Reporting Record

Table 9: Error record for First Round of WSR Testing

Record Bit Position	Data Field
15:0	Bad data of current clock cycle
31:16	Data last clock cycle
57:56	Pattern Selection: 00 All 0's 01 All 1's 10 Checkerboard 11 Checkerboard
64	Chain0 Heartbeat (indicates if a chain's heart beat is alive): 0 indicates alive, 1 indicates dead)
65	Chain1 Heartbeat (indicates if a chain's heart beat is alive): 0 indicates alive, 1 indicates dead)
66	Chain2 Heartbeat (indicates if a chain's heart beat is alive): 0 indicates alive, 1 indicates dead)
67	Chain3 Heartbeat (indicates if a chain's heart beat is alive): 0 indicates alive, 1 indicates dead)
95:80	Frequency selection for WSR
180:146	Timer – clock cycles (granularity of 1 timer clock cycle is equal to DUT operational speed)
183:181	Record indicator: 000: debug 010: SEU error record 101: In timeout (one of the heart beats has died) 111: Out of timeout (one of the heart beats was dead and now has come alive) 110: Beam on indicator 100: Beam off indictor

6.2.5 Post Processing of WSR Data and SEU Cross-Section Calculations

Post processing is performed on all error records (see previous section).

As previously mentioned, error signatures are calculated per chain and are differentiated between:

- 1-bit errors within a window. Characterized using Eqn. 4
- Burst Errors: Clock/reset error signatures with WSRs generally last nanoseconds to microseconds. Characterized using Eqn. 5
- Timeouts: WSR Heartbeat disappears. Characterized using Eqn. 6
- MFTF: Monitoring first failure out of any of the four (4) WSRs. Characterized using Eqn. 7

SEU cross sections are calculated for each error signature. They are calculated by counting the number of SEUs per type. Consecutive window upsets are considered part of a burst and are not part of a single bit upset.

$$\sigma(LET)_{WSR\_bit} = \frac{\#WSR\_SEUs}{\left(\#\frac{ParticlesToFailure}{cm^2}\right)*\#WSR\_bits} \quad (4)$$

$$\sigma(LET)_{WSR\_bursts} = \frac{\#WSR\_bursts}{\left(\#\frac{ParticlesToFailure}{cm^2}\right)} \quad (5)$$

$$\sigma(LET)_{WSR\_timeouts} = \frac{\#WSR\_timeouts}{\left(\#\frac{ParticlesToFailure}{cm^2}\right)} \quad (6)$$

$$\sigma(LET)_{WSR\_Device} = \frac{1}{\#ParticlesToFailure/cm^2} = \frac{1}{\left(TimeToFailure*\frac{flux}{cm^2}\right)} \quad (7)$$

6.3 Counter Array DUA

For the SEU testing of the PolarFire® Counters the standard "counter-array" architecture as described in [2] was implemented.

6.3.1 Counter Array Implementation

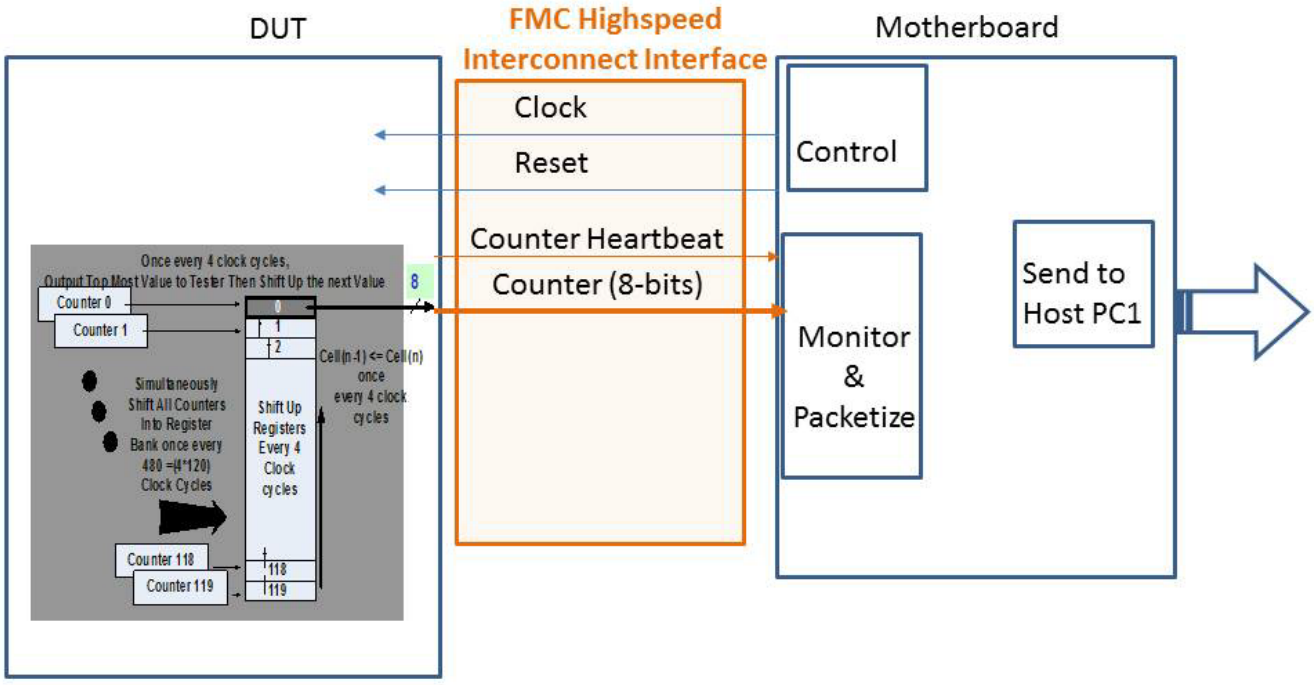


Figure 12: Schematic of the 8-bit Counters and their Output Selection Logic

The counter array developed by NEPP is illustrated in Figure 12. The array contains 200 counters that were 8-bits wide. Because it is impossible to simultaneously output 200 by 8-bits, requiring 1600 outputs, an output scheme had to be employed that would not compromise the number or speed of the counters yet ensure that each counter is an observable node. Conventional thinking would suggest employing a multiplexer that sequences through the array and selects one of the 1600 counters to be output at a time. Unfortunately, the function of the multiplexer, selecting 200 items, requires many levels of combinatorial logic that can be problematic during radiation testing and will slow down the operation of the circuit. Such a large block of logic can potentially mask the primary objective – i.e.,

which is characterizing counter SEU susceptibility. Therefore, a novel output methodology had to be established.

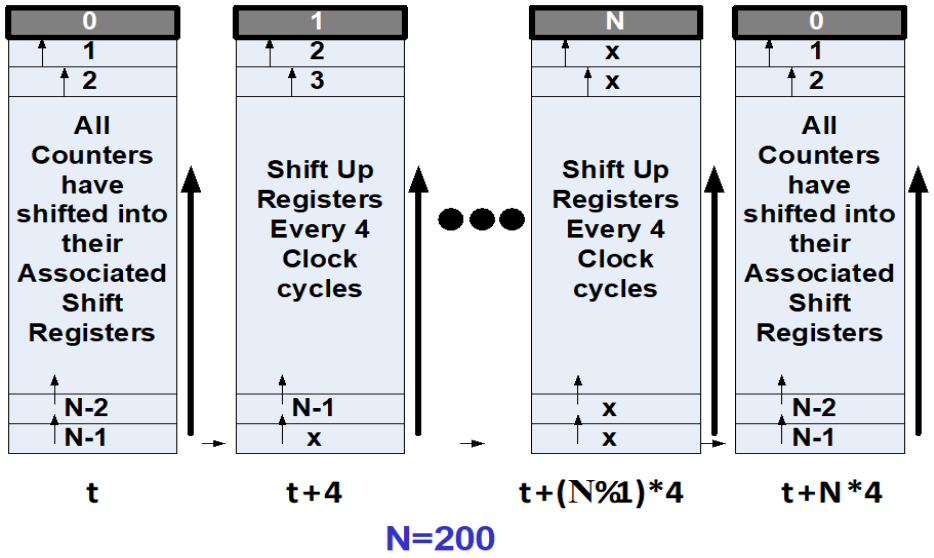


Figure 13: Counter Shift Register Cycles inside a snapshot bank; Values in Snapshot shift registers represent counter labels at a given moment in time. Regarding this figure, if there is an x with the Snapshot register, then it is considered a "don't-care" state. N=200 for the PolarFire® Radiation Test.

As an alternative, a "snapshot" solution was implemented. With this methodology, each counter is captured simultaneously at a given time into a bank of snapshot registers. The number of registers is equivalent to the number of counters (i.e. each counter has its own snapshot register). This is illustrated in Figure 13. The top of the register bank (register 0) is the only register that is accessible by the tester and is 8-bits wide. Subsequently, the DUT to tester interface is simplified.

Figure 12 and Figure 13 illustrate the utilization of the snapshot shift register for each clock cycle. The nomenclature pertaining to the counter circuits throughout this document is as follows:

- n: counter label number
- k: snapshot cycle. First cycle out of reset all of the counter values are snapshot (shifted over) to the shift up register bank. K is 0 for this cycle. The next snapshot of counter values is 800 clock cycles later and k will increment to 1.
- $X_{n,k}$ is the counter-n value that was snapshot into the shift up register for snapshot cycle k.

Coming out of reset, each counter has an initial value (X_{n0}) equal to the counter label number (n), e.g. Counter 0 has a reset value of 0 ($X_{0,0}=0$) and counter 199 has a value of 199 ($X_{199,0}=199$). As the circuit comes out of reset, the counter values are loaded into its corresponding snapshot register. The counters continue to increment simultaneously as the shift registers shift counter values up every 4 clocks cycles – illustrated in Figure 13. The purpose of the snapshot register is to output all counters to the tester. The snapshot registers are loaded and then each value is shifted up so each counter-value can reach register 0 (the output window to the tester). After all loaded counters residing in the snapshot bank have reached register 0 ($\tau+4N$ = once every 4*200 = 800 clock cycles), all of the counters are reloaded into the snapshot shift register bank.

As a summary, the key of the snapshot output scheme, is that the shift register array has now replaced a huge multiplexer. The benefits are as follows:

1. Counter upsets can easily be identifiable.
2. Counters are incrementing and changing state every cycle. Hence maximum performance can be tested.
3. If a counter becomes upset, it will stay upset and it will eventually be captured during the snap shot period Counters are continuous and are not interrupted due to an elaborate output scheme.
4. Routing complexity is exclusive to just the counter array.
5. The shift register architecture allows for high speed counter testing. A large multiplexer creates long paths of combinatorial logic and significantly slows down system speed.

The state space of the DUT should be deterministic and traversable. Pertaining to equation (6), for an 8-bit counter running at 25MHz and a shift up period of once every 4 clock cycles, it will take a little less than 1s for every state to be reached for all counters. Radiation tests generally last for several minutes. Hence, counter states are considered fully traversed within each test run.

$$\frac{2^8}{f_s} = \frac{256}{25\ MHz} = 10.24\ us \qquad (6)$$

6.3.2 Counter I/O Interface and Expected Outputs

Table 10 lists the I/O for the counter DUA. The DUT receives a clock (CLK) and a reset (RESET) from the tester. The DUT responds with two primary outputs:

- Counter: 8-bit counter output of DUT. This is the output from the top snapshot register that is forwarded to the tester.
- Counter_Shift_CLK: This output is ¼ the speed of the CLK (input from the tester). It is ¼ the speed indicating the shift speed (as previously mentioned, the snapshot register shifts once every 4-CLK cycles).

Table 10: DUT1 WSR Outputs.

I/O Name	Bits	Direction wrt to DUT	Description
Counter_Shift_CLK	1	OUT	Counter shift Clock
COUNTER	8	OUT	8-bit Counter output
CLK	1	IN	Clock to counter circuitry
RESET	1	IN	Reset to counter circuitry

The expected output (COUNTER) is purely an increment by 1 starting at value 0. The first COUNTER_Shift_CLK output will pertain to counter 0, followed by counter 1 after 4-CLK cycles, counter 2… up to counter 199. After the value of counter 199 has reached the top of the snapshot register and is output to the tester, a new snapshot is performed (all counters are shifted into their associated snapshot register). After the new snapshot load, the top of the snapshot register bank now has the value for counter 0. Hence, COUNTER_OUT (top of snapshot register) will provide the tester loaded value of counter 0.

With respect to the separate counters, Counter(n) is output every 800-cycles (800-cycles = 1 snapshot cycle). Therefore, the counter values that represent each counter will increment by 800 for each snapshot cycle.

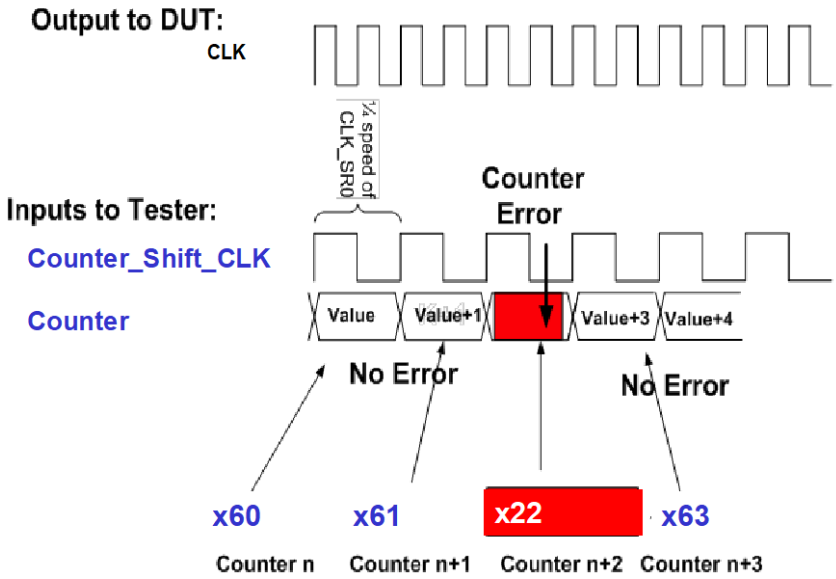


Figure 14: Typical SEE Counter Outputs. Each output represents a value from a different counter in the array. Counter selection is sequential, hence, the counter number and the counter values all increment by 1 each Counter_Shift_Clk cycle.

6.3.3 Counter Array and Motherboard Specifics

DUA contains the following:

- 200 8-bit counters
- 200 8-bit snapshot registers

All counters and snapshot registers are connected to the same clock tree and RESET. The clock tree is fed by the CLK input from the Motherboard-tester.

The Motherboard-tester will send the following signals to the DUT:

- 1 clock
- 1 reset

6.3.4 Processing the DUT Outputs during Testing

The outputs of the DUT are fed to the tester for data processing. The objective of the data processing is to capture data from the DUT, compare to an expected value, and report to the host PC if there is an error. The DUT system clock and reset signals are generated in the Motherboard-tester. The following describes how the counter expected values are calculated. As a note, each counter will have a uniquely stored expected value.

6.3.4.1 Counter Array data processing

All counters within the array independently and continuously increment every clock cycle. Their states are captured (snapshot) once every 800 (4*200) clock cycles into a snapshot array. The top most register of the snapshot array is the only interface to the tester.

In order to avoid metastable events due to an error in the output, data is registered twice before evaluation. Both the data and the counter number are expected to increment every 4 cycles and will wrap around at its boundaries as listed in Table 11.

Table 11 Counter Value and Counter Number Wrap around Boundaries

	Bit Width	Wrap Around Value
8 bit Counter	8	2^8-1 (after 2^8-1 next value is 0)

At the beginning of each snapshot cycle (at capture time), each counter should have incremented by either 800MOD8 from the last snapshot capture. If not, then an error record is created by the Motherboard tester and is later sent to the Host PC #1. The tester keeps a copy of each counter value during a snapshot cycle. This copy is kept around as a comparison point for the next snapshot cycle. During the next snapshot cycle, the old copy becomes: "Last counter value" and the current counter value (just captured) becomes the New counter value. The tester is expecting the increment as follows:

Expected Checks: New Counter Value = (Last counter value + 800) MOD 8 … for an 8-bit counter

If this does not occur, an error is registered. The following is a list of potential incorrect counter values and their corresponding sources of error:

- A single-bit (or multiple-bit) upset in a counter,
- An upset while the value is sitting in the snapshot register,
- Configuration SEU or multiple bit upset (MBU),
- Global upset, or
- Catastrophic event

Due to the complexity of this device, error signatures can be convoluted. Consequently, a significant amount of post processing on radiation data is required to differentiate the SEUs.

6.3.4.2 Counter SHIFT_CLK (Heartbeat) Processing

Regarding the SHIFT_CLK associated with the Counter Arrays, it is used to alert the tester that the counters are alive and new counter data is ready to be taken. The SHIFT_CLKs are always ¼ of the speed of the DUT system clock. SHIFT_CLK can be interpreted as an operation heartbeat.

Due to the interface delays and device latencies and in order to consequently decouple the DUT to tester timing restrictions, the DUT SHIFT_CLK is considered asynchronous to the tester and is sampled using the Motherboard DUT monitor's system clock (50 MHZ). Thus, the tester's sampling clock will always be 4 times as fast as SHIFT_CLK. The SHFT_CLK is fed into a metastability filter and an edge detect. This process takes 1 to 2 clock cycles of the sampling clock (detection will be delayed by 1 to 2 sampling clock cycles of the actual edge).

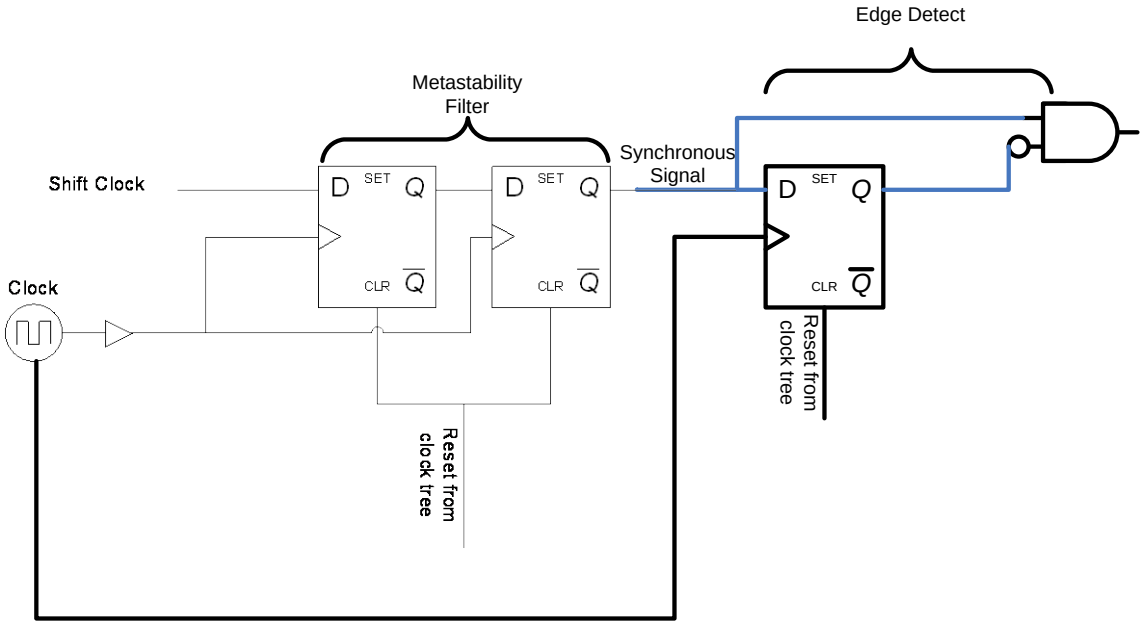


Figure 15: Shift_ClK Capture consists of a Metastability Filter and an Edge Detect

The SHIFT_CLK edge is expected to come at a frequency that is ¼ of the Motherboard DUT monitoring clock. If the edge is stopped, glitched, or missing, the event is reported to the host PC by the Motherboard-tester.

6.3.5 Counter Array Error Record

A significant amount of post processing was performed on this data. Subsequently, the error record should contain enough information to comprehend and differentiate between events.

Table 12: Counter Array Error Record Fields: Yellow indicates Fields generated from Tester capture of DUT Inputs. Other fields indicate values calculated internal to the tester

Field	Bits	Description
Data Cycle N	16	Current DUT output. Error: If it is not an increment of 1 from the previous counter value and not an increment of 2 from the data value received cycle (n-2) No error (recover from error): If it is not an increment of 1 from the previous value but is an increment of the data value received cycle (n-2) Otherwise: DUT is in a burst of error
Data Cycle N-1	16	Capture cycle n-1 DUT output (capture cycle n-1 is actually 4 Motherboard-tester clock cycles from Data cycle N)
Data Cycle N-2	16	Capture cycle n-2 DUT output (capture cycle n-2 is actually 8 Motherboard-tester clock cycles from Data cycle N)
Data Cycle N-3	16	Capture cycle n-3 DUT output (capture cycle n-3 is actually 12 Motherboard-tester clock cycles from Data cycle N)
Counter Number	16	Motherboard-tester local copy of expected counter number. (0 through 299)
Error Count		
Time Stamp	32	Time that error was observed. Time stamp is a cycle counter. It is later converted to seconds for post processing.
Status flags	3	Record indicator: 000: debug 010: SEU error record 101: In timeout (one of the heart beats has died) 111: Out of timeout (one of the heart beats was dead and now has come alive) 110: Beam on indicator 100: Beam off indictor

6.3.6 Counter Array Post Processing

Each error record was post-processed. Error records (SEUs) were differentiated via the following:

- Bit or multiple-bit upset: This example is illustrated in Figure 3; and is as follows: One counter-bit flips its state, and the counter continuously increments as normal from that flipped state. This error signature is attributed to a data-path SEU.
- Snapshot register:
  - A snapshot register bit incurs an SEU. In this case a counter will show up as incorrect for two consecutive snapshot registers.
  - Snap shot register can either stop shifting- in this case the output values will remain constant or become noise
  - Snap shot register can skip a cycle, in this case the counter values will be off the number of skipped shift cycles from their expected values
- Burst of broken counters: Counters stop counting and their values either stay constant or become complete noise. This error signature is attributed to either a misconfiguration or a global upset.
  - Because the counters are flushable, a global structure SEU error can be recoverable by using a reset.
  - Because configuration upsets can only be fixed by performing a full configuration, the associated error signature will appear as a stuck fault.

$$\sigma_{COUNTERSEU\_bit} = \frac{\#CounterArray_{errors}}{fluence * \#Counter\_ARRAY\_DFFs} \quad (8)$$

$$\sigma_{COUNTERSEU\_device} = \frac{1}{\#ParticlesToFailure/cm^2} = \frac{1}{\left(TimeToFailure * \frac{flux}{cm^2}\right)} \quad (9)$$

6.4 Internal SRAM Memory DUA

There are two types of SRAM memory that are accessible to a PolarFire® user. There are a total of 0.74 Mb Micro-SRAM (uSRAM) memory bits and a total of 18.59 Mb of Large-SRAM (LSRAM) memory bits. Each type of SRAM has the option of embedded SECDED (error correction). For this first-look study, only LSRAM with no SECDED was tested. A high-level schematic representation of the PolarFire® internal SRAM with EDAC circuitry is illustrated in Figure 16.

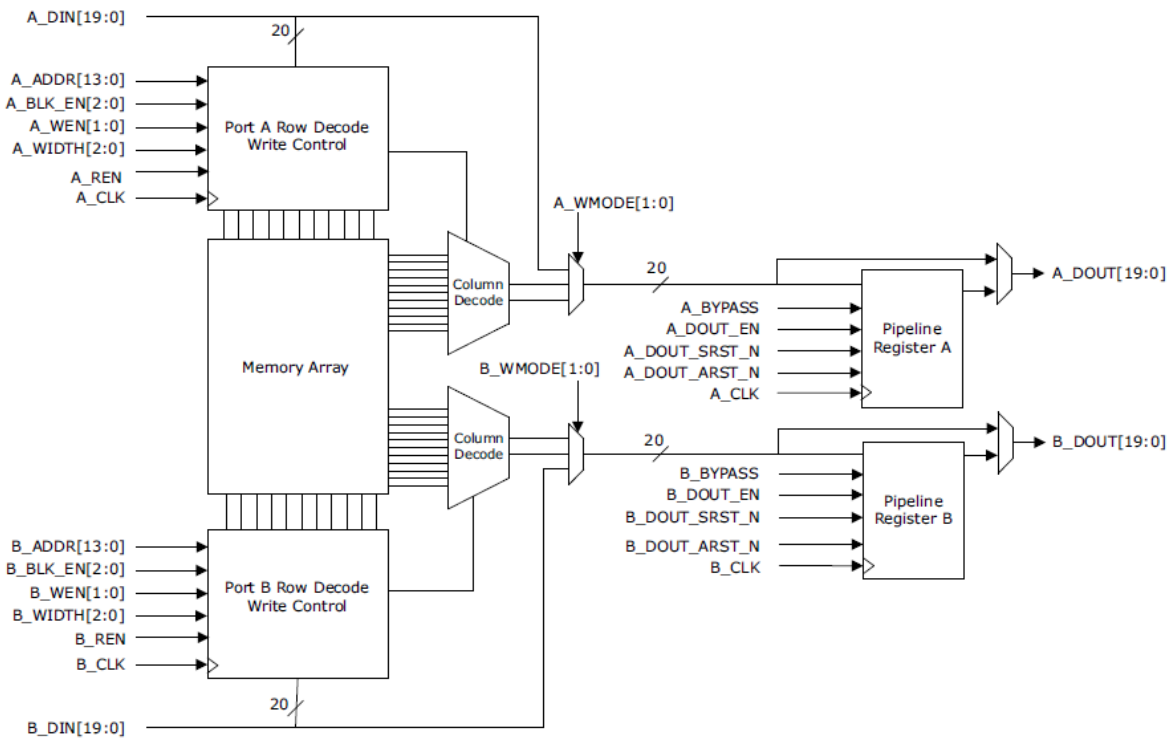


Figure 16: High-level schematic of PolarFire® internal SRAM including EDAC circuitry

6.4.1 Implementation of First-look PolarFire® internal SRAM DUAs

The DUA LSRAM instantiation for this investigation had the following parameters and is depicted in Figure 17:

- LSRAM: 8-bit wide data bus with 524,288 address locations (19-bit address bus). Due to time limitations, no error detection and correction (EDAC) was implemented with the LSRAM. However, if time and money permits, additional testing that includes LSRAM EDAC is expected to be performed.

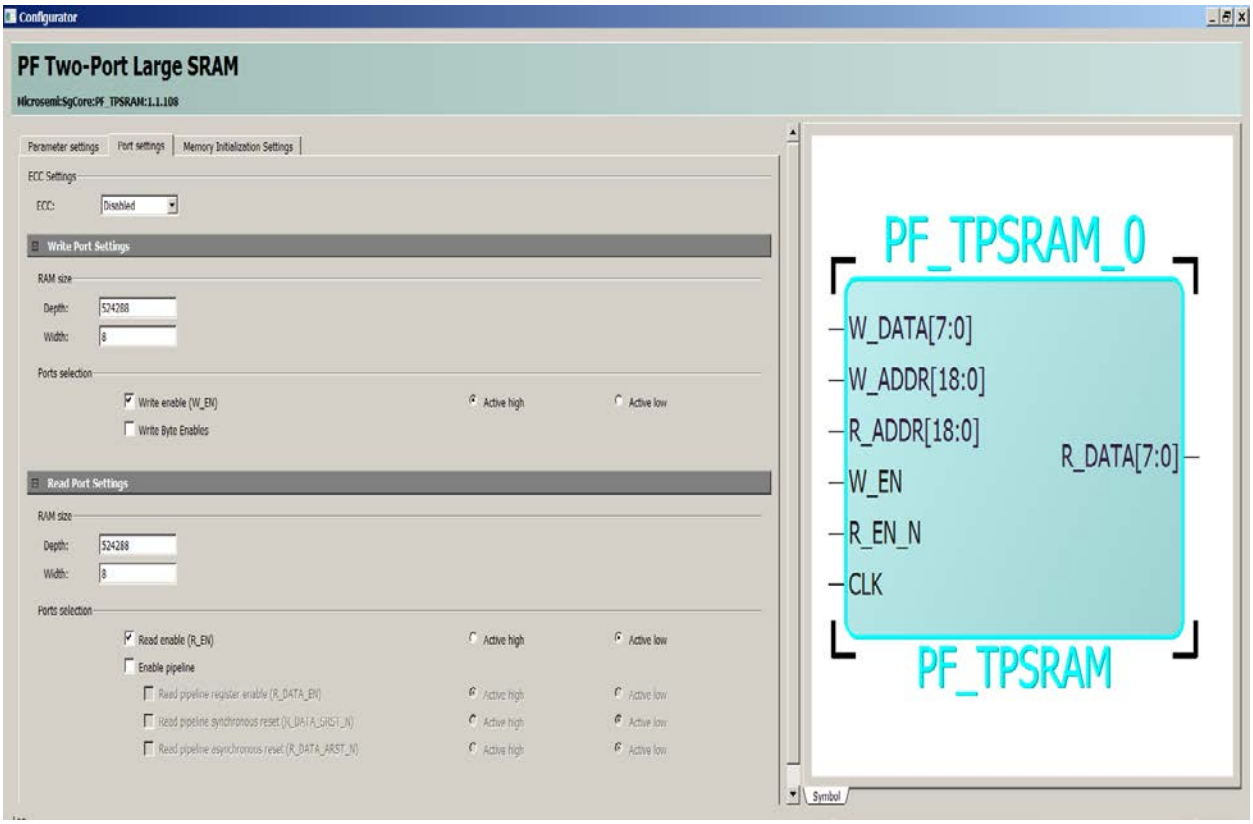


Figure 17: Libero user-GUI for the LSRAM DUA instantiation

6.4.2 SRAM DUA Operation

The following is the algorithm for PolarFire® SRAM DUA testing:

- Write the entire address space. Each address location is written with an 8-bit counter. The value of the counter is equivalent to the lower 8-bits of its address.
- Upon command, start the read-write process. Sequentially each address is read. The output is provided from the DUT to the tester. Following, the same address is re-written with its appropriate counter value.

6.4.3 SRAM DUT to Motherboard-tester Requirements

The Motherboard-tester is required to provide a clock, read address, and write address, write enable, and read enable to the DUT. The DUA performs the expected read or write function depending on the write enable and read enable state.

6.4.4 LSRAM Data Processing

Upon an active read enable (activated by the Motherboard-tester), the DUA will provide the address location contents on the read port. After a delay, the Motherboard DUT monitor reads the DUT port. Once the data is read, the Motherboard-tester compares the read data to the current lower 8-bits of the address. If they are not equal, the Motherboard-tester sends a message to the host PC#1.

6.4.5 LSRAM Error Record

A significant amount of post processing was performed on this data. Subsequently, the error record should contain enough information to comprehend and differentiate between events.

Table 13: LSRAM Error Record Fields: Yellow indicates Fields generated from Tester capture of DUT Inputs. Other fields indicate values calculated internal to the tester

Field	Bits	Bit Position	Description
Read Data	8	7:0	DUA output of data after a read enable
Expected DATA	8	55:40	Current Write Address
Address B	19	34:16	Current Read Address
Error Counter	32	145:114	Count the number of errors observed by the Motherboard DUT monitor
Time Stamp	35	180:146	Time that error was observed. Time stamp is a cycle counter. It is later converted to seconds for post processing.
Status flags	3	183:181	Record indicator: 000: debug 010: SEU error record 101: In timeout (one of the heart beats has died) 111: Out of timeout (one of the heart beats was dead and now has come alive) 110: Beam on indicator 100: Beam off indictor

6.4.6 LSRAM Data Post Processing

Each error record was post-processed. Error records (SEUs) were differentiated via the following: Single bit or multiple-bit upset. In addition, the efficacy of the EDAC circuitry was heavily analyzed.

$$\sigma_{LSRAM\_Singlebit} = \frac{\#LSRAMSingleBit_{errors}}{fluence * 524288 * 8} \quad (10)$$

$$\sigma_{LSRAM\_Multiplebit} = \frac{\#LSRAMMultipleBit_{errors}}{fluence * 524288 * 8} \quad (11)$$

7. DUT ACCELERATED HEAVY ION TEST PROCEDURES

7.1 Summary of DUT-Tester operation for an SEE Experiment.

Overview of performing an accelerated SEU test:

- Bias the device, turn on clocks, and toggle reset.
- For dynamic tests: DUA logic operates simultaneously with tester monitoring and capturing DUA outputs. Tester compares DUA captured data with expected pattern (verify no errors).
- Irradiate DUT.

- During irradiation: Tester reads DUT and compares to expected value:
    - If error during read, then the Motherboard DUT Monitor records that an error has occurred and sends the data value with timestamp to host PC #1
- Stop Beam
- Reset Tester and DUT to prepare for next test

7.2 Running a Full Test

7.2.1 Files required for running a test

In order to run a test, the tester-bit file is downloaded from the host PC, via Xilinx-impact (JTAG), to the KCU105 (Motherboard) FPGA's configuration memory; the DUT-configuration file is downloaded from the host PC, via Microsemi-Libero Flash programmer, to the DUT configuration memory. After each test is finished, the DUT configuration is read-back, via Microsemi-Libero flash programmer. The read-back verify function will indicated a pass-fail.

7.2.2 Procedures for running a test

The following sections and tables list the procedures required for running a test.

7.2.2.1 WSR

Table 14: Running a WSR Test (1st round and 2nd round with PLLs)

Action	**Explanation**
Beam is off	
Program tester (download bit file to tester)	Via KCU105 JTAG.
Program DUT (download bit file to DUT)	Via DUT JTAG.
Hardware reset	
Command 01 x x x	Via LabVIEW: PC#1 sends to Motherboard command Soft Reset.
Command A0 nn nn x	Via LabVIEW: PC#1 sends to Motherboard Frequency Control.
Command 02 x x x	Via LabVIEW: PC#1 sends to Motherboard Start the tester/DUT system (clocks are started; and reset is released)
Turn Beam on	FPGA tester will collect DUT outputs and determine if an SEU has affected operation. FPGA tester will send the information to the host computer system. Upsets can be observed via LabVIEW window and Logic Analyzer.
Turn Beam off	Reset system after turning beam off and log information

7.2.2.2 Counter Arrays

Table 15: Running a Counter Array Test

Action	**Explanation**
Turn beam off	
Program tester (download bit file to tester)	Via KCU105 JTAG.
Program DUT (download bit file to DUT)	Via PolarFire® JTAG.
Hardware reset	
Command 01 x x x	Via LabVIEW: PC#1 sends to Motherboard command Soft Reset.
Command 99 x x x	Via LabVIEW: PC#1 sends to Motherboard command Soft Reset 2.
Command A0 nn nn x	Via LabVIEW: PC#1 sends to Motherboard Speed Control.
Command 02 x x x	Via LabVIEW: PC#1 sends to Motherboard Start the tester/DUT system (clocks start and reset is released)
Turn Beam on	FPGA tester will collect DUT outputs and determine if an SEU has affected operation. FPGA tester will send the information to the host computer system. Upsets can be observed via LabVIEW window and Logic Analyzer.
Turn Beam off	Reset system after turning beam off and log information

7.2.2.3 SRAM DUAs

SRAM tests do not vary by frequency or pattern.

Table 16: Running a Dynamic SRAM Test

Action	**Explanation**
Turn beam off	
Program tester (download bit file to tester)	Via KCU105 JTAG.
Program DUT (download bit file to DUT)	Via DUT JTAG.
Hardware reset	
Command 01 x x x	Via LabVIEW: PC#1 sends to Motherboard command Soft Reset.
Command 99 x x x	Via LabVIEW: PC#1 sends to Motherboard command Soft Reset 2.
Command 02 x x x	Via LabVIEW: PC#1 sends to Motherboard Start the tester/DUT system (clocks start; and reset is released)
Turn Beam on	FPGA tester will collect DUT outputs and determine if an SEU has affected operation. FPGA tester will send the information to the host computer system. Upsets can be observed via LabVIEW window and Logic Analyzer.
Turn Beam off	Reset system after turning beam off and log information

Table 17: Running a Halt SRAM Test

Action	**Explanation**
Turn beam off	
Program tester (download bit file to tester)	Via KCU105 JTAG.
Program DUT (download bit file to DUT)	Via DUT JTAG.
Hardware reset	
Command 01 x x x	Via LabVIEW: PC#1 sends to Motherboard command Soft Reset.
Command 99 x x x	Via LabVIEW: PC#1 sends to Motherboard command Soft Reset.
Command 02 x x x	Via LabVIEW: PC#1 sends to Motherboard command Start the tester/DUT system (clocks start and reset is released)
Command 04 x x x	Halt operation
Turn Beam on	DUT is in a halt state so Motherboard is not monitoring DUT output. However, Motherboard is monitoring power/current of the DUT
Turn Beam off	Reset system after turning beam off and log information
Command 06 x x x	Start read/write operation after halt
Turn Beam off	Reset system after turning beam off and log information

8. HEAVY ION TEST FACILITY AND TEST CONDITIONS

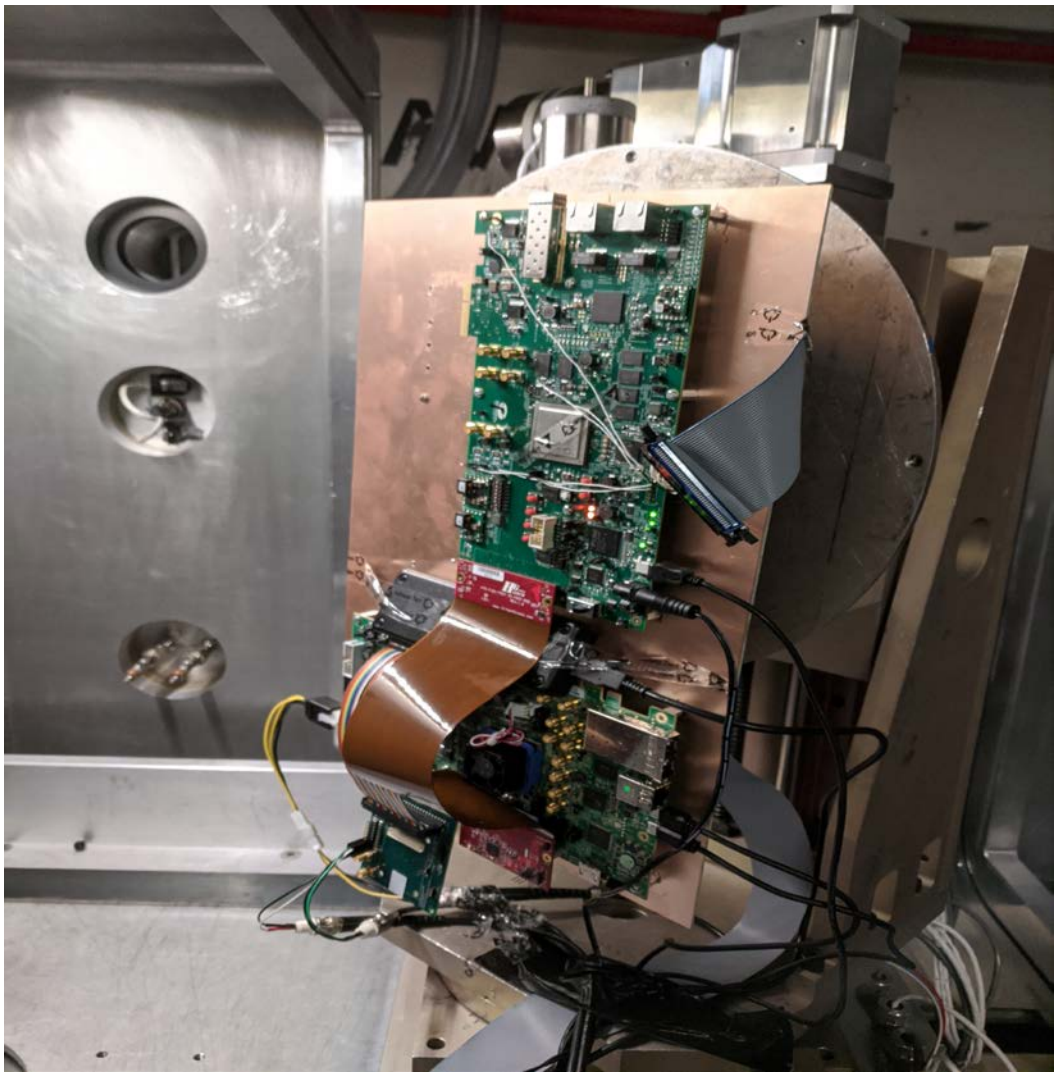


Figure 18: Motherboard (KCU105) and Daughterboard (PolarFire®) connection. Setup is in the Lawrence Berkeley National Laboratory (LBNL) SEE vacuum chamber.

Heavy ion testing was performed at Lawrence Berkeley National Laboratories 88inch Cyclotron (LBNL). The vacuum chamber setup is shown in Figure 18. Table 18 and Table 19 list experiment logistics while at the test facility.

Table 18: Overview of Testing Logistics

Facility	Lawrence Berkeley National Laboratories 88inch Cyclotron, 16 MeV/amu tune
Flux	$1.0x10^3$ to $1.0x10^5$ particles/cm^2/s
Fluence	All tests were run to 1 x 10^7 particles/cm^2 or until destructive or functional events occurred.
Test Temperature	Room Temperature
Power Supply Voltages	V_{cc} = 1.2V; V_{IO} = 2.5V

Table 19: LBNL LET Table for 16 MeV Normal Incidence

Ion	Energy (MeV/Nucleon)	Effective LET(MeV·cm^2/mg) at 0°
N	16	1.16
O	16	1.54
Ne	16	2.39
Si	16	4.35
Ar	16	7.27
V	16	10.9
Cu	16	16.5
Kr	16	25.0
Xe	16	49.3

9. POLARFIRE® HEAVY ION TEST RESULTS

9.1 Overview

Table 18 lists the targeted ions for the test campaign. However, due to repercussions from the wildfires, beam time was limited. Consequently, only N, O, and Ne were able to be used for the first-look experiments.

The following sections provide SEE data for WSRs, Counter Arrays, and LSRAM DUAs instantiated in a PolarFire® FPGA device.

9.2 Anomalies (Single Event Functional Interrupts)

One significant anomaly was observed during heavy-ion testing. The error signature is a current drop from normal operational current to approximately 800 mA; and is signified in this document as a single event functional interrupt (SEFI). The measured SEFI cross-sections are shown in Figure 19 and Figure 20.

The following is a list of observation notes pertaining to the current drop SEFI.

- The core-current dropped below 100 mA when normal operational current was marked at approximately 2.75 A as illustrated in Figure 21.
- The current drop was always recoverable.
- The current drop lasted for approximately 1.7 *ms* except for one event which lasted for approximately 177*s*. Note that the event shown in Figure 21 is not the 1.7 *ms* event; alternatively, it is a depiction of the 177 *s* event.
- The current drop is significant enough to stop operation.
- A reset is required after the current drop.
- No configuration is lost after a current drop (read-back passes with no SEUs).
- The current drop occurred for every test at every LET (that was used during the first-look study); and has an onset LET lower than 1 MeV·cm^2/mg.

- Lower LET values are required to achieve a more accurate reliability/survivability calculation per environment; and will be tested in subsequent test campaigns.
- Microsemi is aware of the anomaly (SEFI). Microsemi is suggesting that this event is an artifact of a user-defined mode setting. The SEFI will be further investigated during NEPP's next PolarFire test campaign.

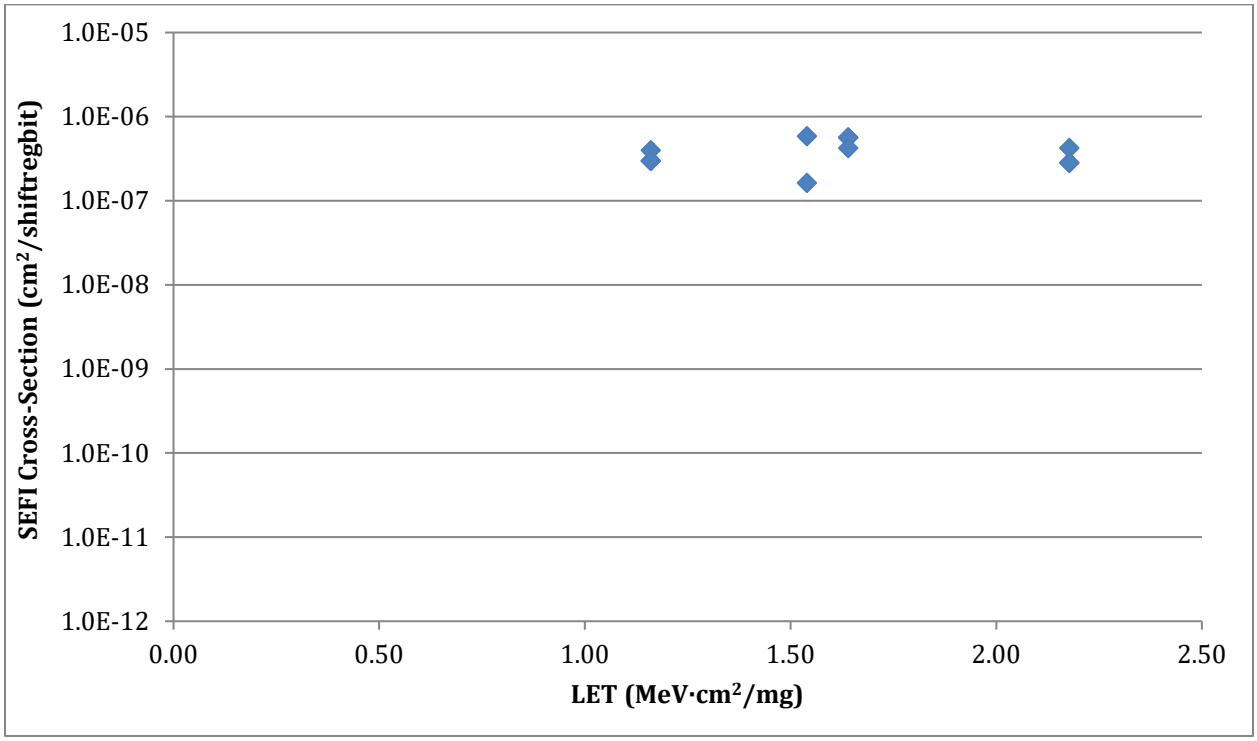


Figure 19: SEFI Cross Sections versus LET for Shift Register Experiments

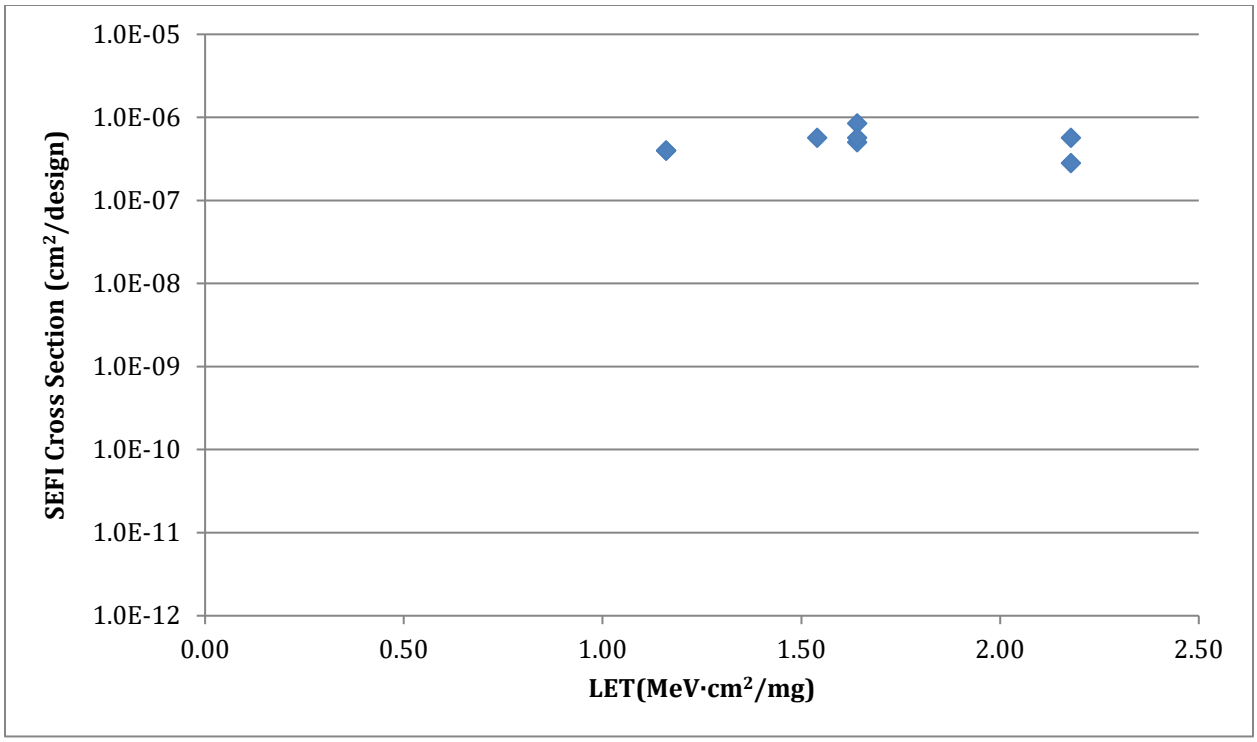


Figure 20: SEFI Cross Sections versus LET for Counter Experiments

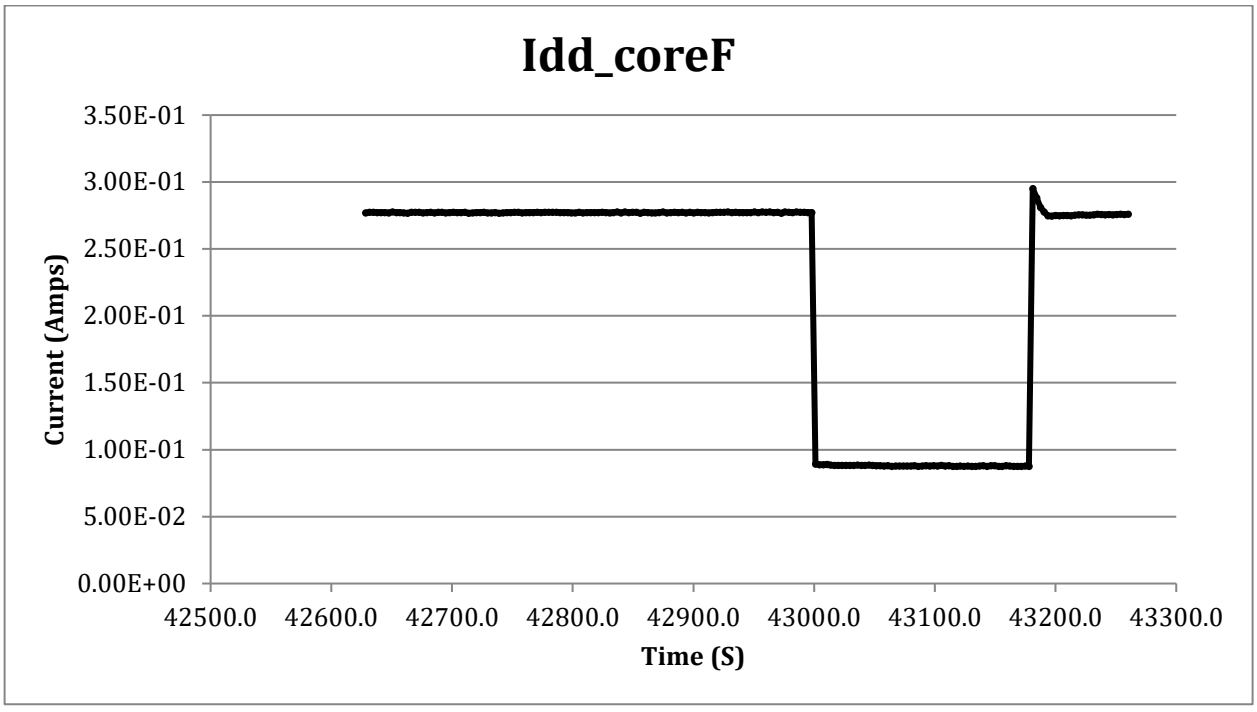


Figure 21: Current drop within core voltage domain. Drop lasted for 177 *s* (expected time length of current drop is 1.7 *ms*)

9.3 WSR Heavy Ion Test Results

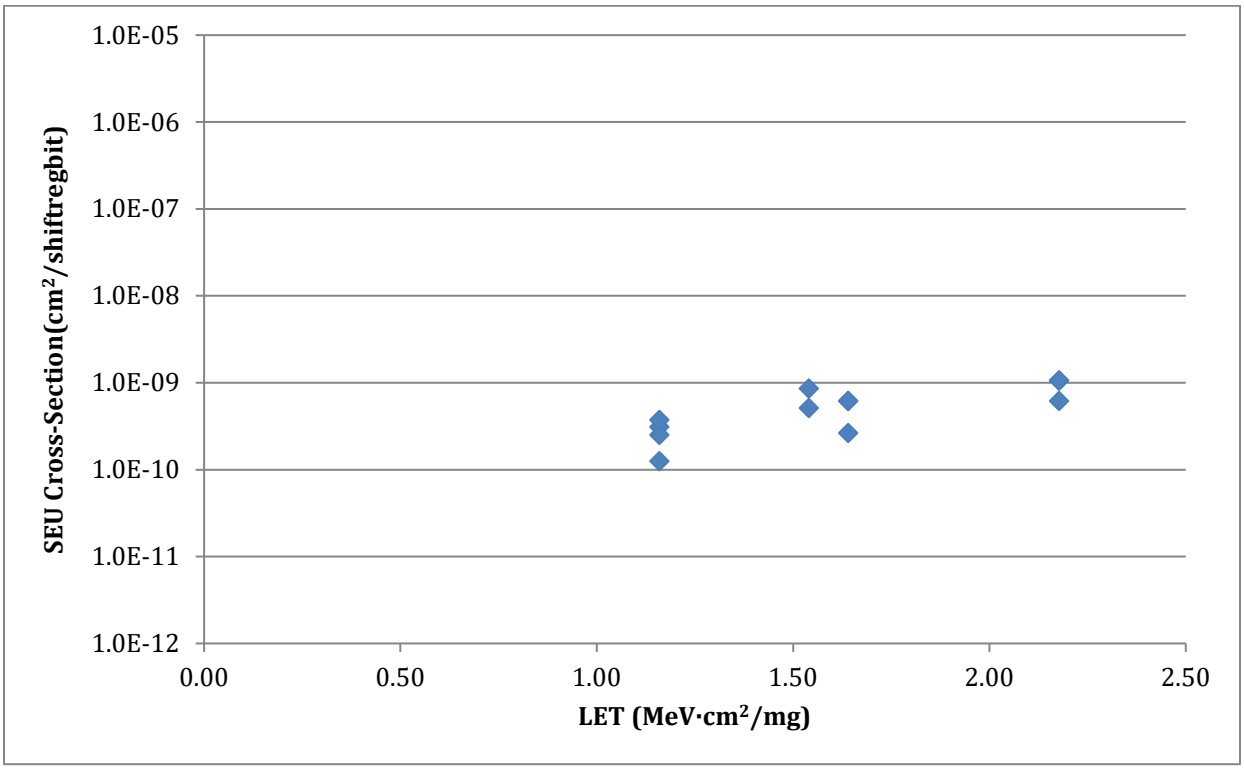


Figure 22: Shift Register SEU Cross Sections Normalized per Shift Register Bit versus LET

Figure 22 and Figure 23 show the SEU cross-sections per DFF bit and per burst accordingly. Regarding Figure 22, DFF upsets were single bit SEUs that were flushed out by the following shift register cycle DFF write; i.e., no SEUs lasted for more than one clock cycle and no data-paths were broken unless a SEFI (current-drop timeout) occurred.

Regarding Figure 23, bursts were observed (separate event than current-drop SEFI). Bursts were able to be differentiated and only lasted for *ns-µs*. Bursts occurred from clock or reset SETs.

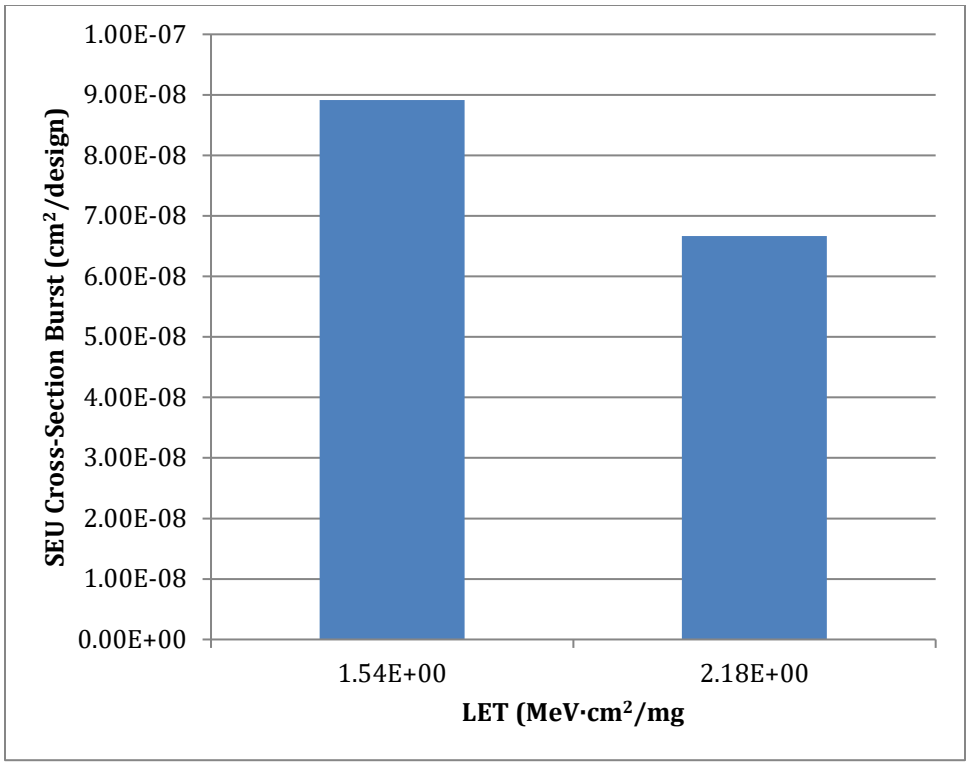


Figure 23: Burst SEU Cross-Sections per Shift Register versus LET

9.4 Counter Heavy Ion Test Results

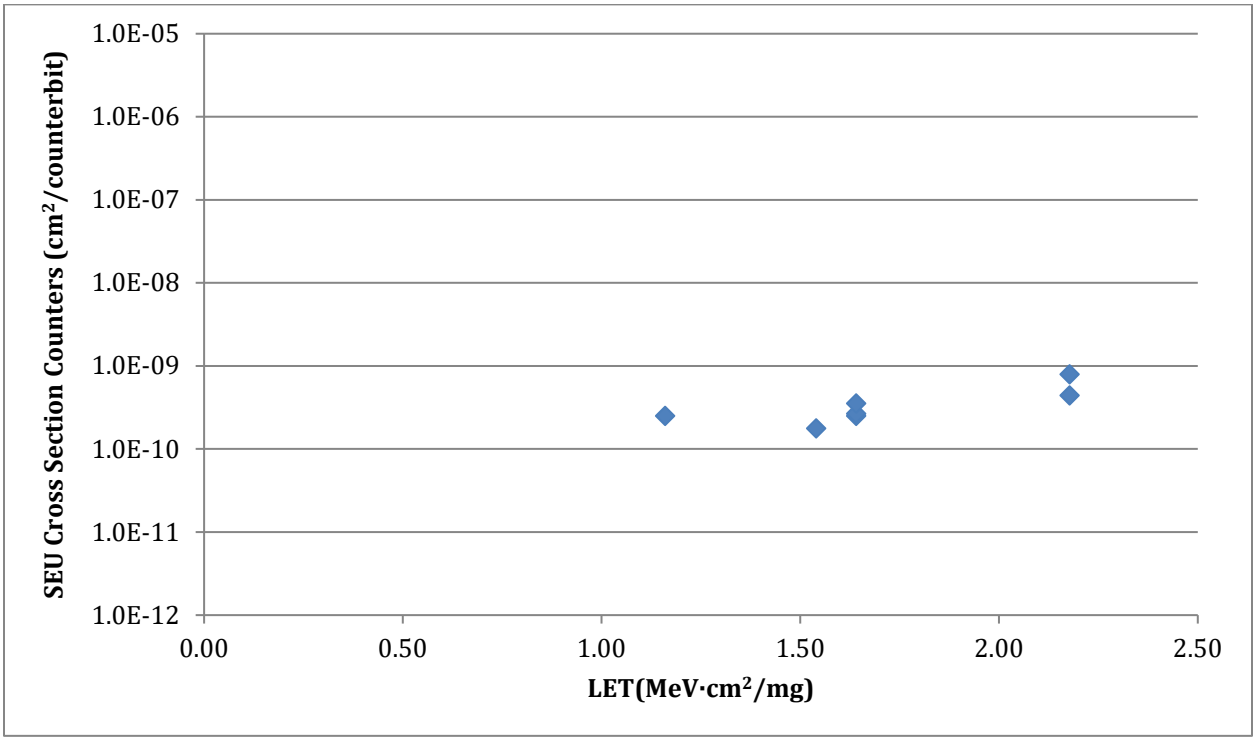


Figure 24: Counter SEU Cross Sections Normalized per Counter Bit versus LET

Counter array SEU cross sections are illustrated in Figure 24. It is interesting to note that the counter array SEU cross sections per DFF are statistically equivalent to the WSR SEU cross-sections (both operating at the same frequency). This is noted because the WSR does not have any combinatorial logic; and subsequently suggests (for low LET values) that the DFF nodes will be the dominant mechanisms of failures. Additional testing is required.

9.5 LSRAM Heavy Ion Test Results

Figure 25 shows the LSRAM SEU cross-sections per bit. All LSRAM SEUs were single bit with exception to Single Event Functional Interrupts (SEFIs). This suggests that they will be correctable when implementing error correction (single error correct double error detect (SECDED)). SEFIs were due to the current drops; and have been verified by duration of SEFI responses during beam exposure. In all SEFI cases, BRAM cells could be restored by an overwrite. However, SECDED would not be able to be applied (SEFIs are not single bit errors).

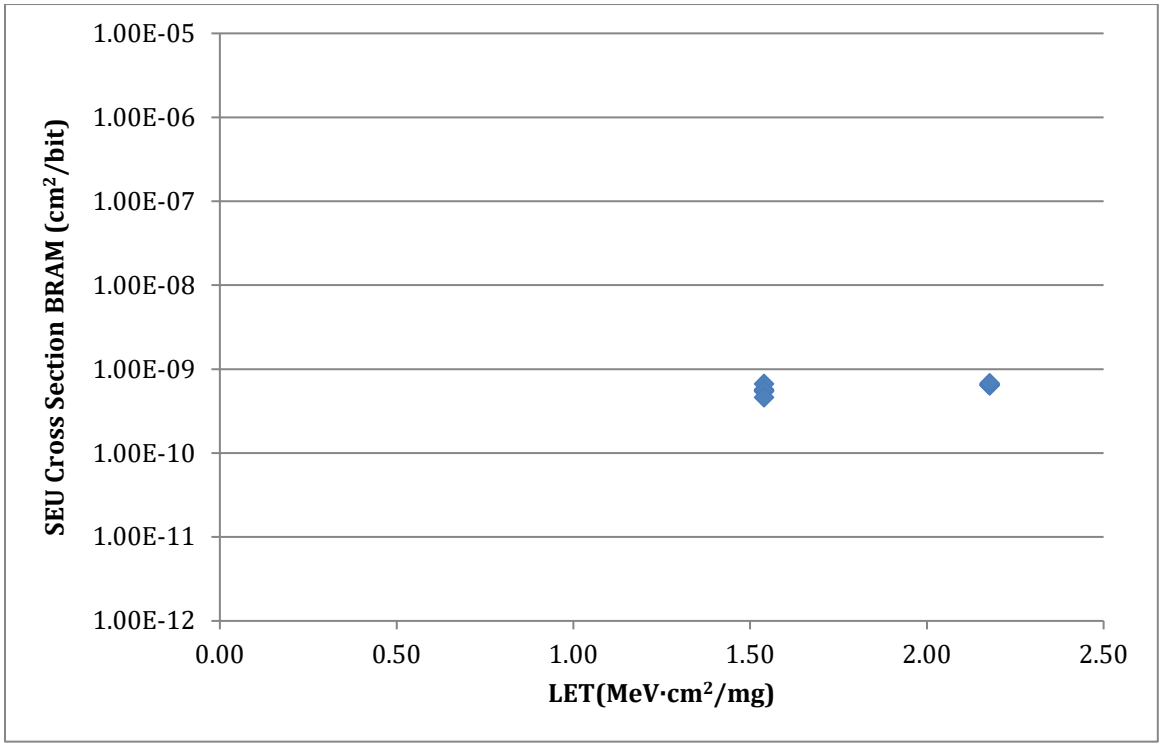


Figure 25: LSRAM SEU Cross Sections Normalized per BRAM Bit versus LET

10. APPENDIX 1:

[1] Microsemi Userguide: "UG0680 PolarFire FPGA Fabric User Guide Revision 5.0"

[2] NASA Electronics Parts and Packaging (NEPP) FPGA SEU Test Guidelines: https://nepp.nasa.gov/files/23779/fpga_radiation_test_guidelines_2012.pdf

[3] M. Berg "An Analysis of Single Event Upset Dependencies on High Frequency and Architectural Implementations within Actel RTAX-S Family Field Programmable Gate Arrays," IEEE Trans. Nucl. Sci., vol. 53, n° 6, Dec. 2006.

[4] J. J. Wang, Nadia Rezzak, and Durwyn Dsilva, "RTG4 Radiation Test Results and Test Plans," presented at Microsemi Space Forum 2015.

11. ACRONYMS

Table 20 Acronyms

Acronym	Definition
1MB	1 Megabit
3D	Three Dimensional
3DIC	Three Dimensional Integrated Circuits
ACE	Absolute Contacting Encoder
AHB	Advanced high performance bus
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AMD	Advanced Micro Devices Incorporated
AMS	Agile Mixed Signal
AXI	Advanced extensible interface
BGA	Ball Grid Array
BRAM	Block Random Access Memory
CAN	Controller Area Network
CCC	clock conditioning circuit
CCI	Correct Coding Initiative
CGA	Column Grid Array
CMOS	Complementary Metal Oxide Semiconductor
CNSA	Crypto engines (NSA)
COTS	Commercial Off The Shelf
CRC	Cyclic Redundancy Check
CSE	Crypto Security Engineer
CTLE	Ethernet signal conditioning (linear equalization)
CU	Control Unit
DC	Direct current
DCU	Distributed Control Unit
DDR	Double Data Rate (DDR3 = Generation 3; DDR4 = Generation 4)
DFE	Ethernet signal conditioning
DFF	Flip-flop
DLL	Delay locked loop
DMM	Digital Multimeter
DMA	Direct Memory Access
DSP	Digital Signal Processing
DSPI	Dynamic Signal Processing Instrument
Dual Ch.	Dual Channel
DUT	Device under test
ECC	Error-Correcting Code
EDAC	Error detection and correction
EEE	Electrical, Electronic, and Electromechanical
EMAC	Equipment Monitor And Control
EMIB	Multi-die Interconnect Bridge
EPCS	Extended physical coding layer
eTimers	Event Timers

FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
Gb	Gigabit
Gbps	Gigabit per second
GIC	Global Industry Classification
GPIO	General purpose input/output
GPIB	General purpose interface bus
GPU	Graphics Processing Unit
GR	Global Route
GSFC	Goddard Space Flight Center
GTH/GTY/GTX	Transceiver Type
HALT	Highly Accelerated Life Test
HAST	Highly Accelerated Stress Test
HBM	High Bandwidth Memory
HDIO	High Density Digital Input/Output
HDR	High-Dynamic-Range
HiREV	High Reliability Virtual Electronics Center
HKMG	high-k metal gate
HMC	Hybrid Memory Cube
HPC	High Performance Connector
HPIO	High Performance Input/Output
HSTL	High speed transceiver logic
I/F	interface
I/O	input/output
I2C	Inter-Integrated Circuit
i2MOS	Microsemi second generation of Rad-Hard MOSFET
IC	Integrated Circuit
I-Cache	independent cache
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group (FPGAs use JTAG to provide access to their programming debug/emulation functions)
KB	Kilobyte
LET	Linear energy transfer
LP	Low Power
LUT	Look-up table
LVCMOS	Low-voltage Complementary Metal Oxide Semiconductor
LVDS	Low-Voltage Differential Signaling
LVTTL	Low –voltage transistor-transistor logic
LSRAM	Large embedded static random access memory
M/L BIST	Memory/Logic Built-In Self-Test
Mil-STD	Military standard
MFTF	Mean fluence to failure
µPROM	Micro programmable read-only memory
µSRAM	Micro SRAM
Mil/Aero	Military/Aerospace
MIPI	Mobile Industry Processor Interface
MMC	MultiMediaCard
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MP	Microprocessor
MP	Multiport
MPFE	Multiport Front-End

MPSoC	Multiprocessor System on a chip
MPU	Microprocessor Unit
Msg	message
MTTF	Mean time to failure
NAND	Negated AND or NOT AND
NASA	National Aeronautics and Space Administration
NEPP	NASA Electronic Parts and Packaging
NOR	Not OR logic gate
NV(M)	Non-volatile (memory)
OCM	On-chip RAM
OSC-TMR-PLL	Embedded triple modular redundant phase locked loop
OSC	Oscillator
PC	Personal Computer
PCB	Printed Circuit Board
PCIe	Peripheral Component Interconnect Express
PCIe Gen2	Peripheral Component Interconnect Express Generation 2
Pconfiguration	SEU cross-section of configuration
Pfunctional_logic	SEU cross-section of functional logic
PHY	Physical layer
PLL	Phase Locked Loop
PLOL	Phase Locked Loop loss of lock
PMA	Physical Medium Attachment
POR	Power on reset
PPM	Parts per million
Proc.	Processing
PS-GTR	High Speed Bus Interface
PSEFI	SEU cross-section from single event functional interrupts
Psystem	System SEU cross-section
PUF	Physically unclonable function
QDR	quad data rate
QFN	Quad Flat Pack No Lead
QML	Qualified manufactures list
QSPI	Serial Quad Input/Output
RC	Resistor capacitor
R&M	Reliability and Maintainability
RGB	Red, Green, and Blue
RH	Radiation Hardened
RT	Radiation Tolerant
RTD	Representative tactical design
SATA	Serial Advanced Technology Attachment
SCU	Secondary Control Unit
SD	Secure Digital
SD/eMMC	Secure Digital embedded MultiMediaCard
SD-HC	Secure Digital High Capacity
SDM	Spatial-Division-Multiplexing
SEE	Single Event Effect
SEF	Single event failure
SEFI	Single Event Functional Interrupt
SEL	Single event latchup
SERDES	Serializer/deserializer
SET	Single event transient
SEU	Single event upset

Si	Silicon
SOC	Systems on a Chip
SPI	Serial Peripheral Interface
sRIO	Serio Rapid I/O
SSTL	Sub series terminated logic
TBD	To Be Determined
Temp	Temperature
THD+N	Total Harmonic Distortion Plus Noise
TMR	Triple Modular Redundancy
T-Sensor	Temperature-Sensor
TSMC	Taiwan Semiconductor Manufacturing Company
UART	Universal Asynchronous Receiver/Transmitter
UltraRAM	Ultra Random Access Memory
USB	Universal Serial Bus
VNAND	Vertical NAND
WDT	Watchdog Timer
WSR	Windowed shift register
XAUI	Extended 10 Gigabit Media Independent Interface
XGXS	10 Gigabit Ethernet Extended Sublayer
XGMII	10 Gigabit Media Independent Interface)