Single-event Effects Test Report of the MSP430FR5969-SP

ABSTRACT

This report summarizes the effect of heavy-ion irradiation on the single-event effect performance of the MSP430FR5969-SP FRAM microcontroller. Heavy-ions with LET_{EFF} ranging from 3.3 to 61.5 MeV-cm^2/mg were used to irradiate devices with fluences up to 5×10^6 ions/cm^2 per run.

Contents

List of Figures

List of Tables

Trademarks

1 Overview

This report details the Single Event Effects (SEE) testing of the MSP430FR5969-SP. The MSP430FR5969-SP is a space grade radiation hardened mixed-signal FRAM based microcontroller (MCU). Accurate and useful SEE testing of MCUs is challenging due to the complexity of the devices. This complexity provides for extremely large variety of internal possible upsets that, in turn, can result in many potential detectable signatures. Some of these events are relatively benign, and can be caught and corrected by the MCU itself, or by external components monitoring the system. Other events, such as those that can cause code or data corruption are potentially much more significant and must be detected and handled robustly. Most SEE characterization of MCUs is done using a system approach versus a characterization of each individual functional block. It is very difficult to disassociate the performance of an individual block in the MCU from the processor itself. Virtually every block relies on the processor to some degree for configuration of peripherals, or moving and accessing data. Thus, a holistic system approach is desirable, as it eliminates many of the issues associated with characterizing individual blocks within the MCU. Single event errors using the system approach can be summarized by how many times the system did not perform its desired function correctly. This minimizes the many different potential signatures. The approach used for this characterization is primarily systems based. This approach also implements error detection and correction along with internal tracking of event sources within the non-volatile FRAM memory. This method allows the generation of a functional FRAM cross section along with other SEE types of cross sections by post processing of the saved memory dumps. Section 6 discusses the SEE types.

This report includes some key aspects to consider that will help ensure the ability to achieve optimum results. Section 3 shows the key considerations. It is also worthy to note that the FRAM itself is extremely robust to corruption from irradiation. Any corruption to data stored in the FRAM array is a result of an error induced in the data path or the FRAM controller logic and sense amps. The test code was designed to take advantage of this fact as it reserves key data and code for error checking and repair. The error checking and repair code are only executed and accessed during boot time. Implementing in this fashion minimizes the probability of this critical code being corrupted due to the small amount of time that it is exposed to ions. In order to help create more realistic cross sections, attempts were made to minimize events from being able to take place during the boot process. The approach used for data taken at the TAMU facility used a mechanical shutter. The shutter closes when DUT is reset, or after the DUT detected an event prior to rebooting. This helps minimize effects of additional events occurring during the recovery of an event as this is a function of the accelerated testing environment. Unfortunately, the shutter hardware did not function properly in a vacuum environment due to overheating and longer cabling. Thus, it was not used for data taken at Berkeley. Section 7 elaborates on the details of the shutter implementation.

Table 1. Overview Information (1)

TI PART NUMBER	MSP430FR5969-SP
Device Function	Radiation Hardened Mixed-Signal Microcontroller
Exposure Facilities	Lawrence Berkeley National Laboratory
	Texas A & M Cyclotron Facility

(1) TI may provide technical, applications or design advice, quality characterization, and reliability data or service providing these items shall not expand or otherwise affect TI's warranties as set forth in the Texas Instruments Incorporated Standard Terms and Conditions of Sale for Semiconductor Products and no obligation or liability shall arise from Semiconductor Products and no obligation or liability shall arise from TI's provision of such items.

Table 1. Overview Information [(1)] (continued)

TI PART NUMBER	MSP430FR5969-SP
Targeted Heavy Ion Fluence per Run	$1x10^6$ to $5x10^6$ ions/cm^2
Irradiation Temperature	25°C

Go to the MSP430FR5969-SP product page for more detailed technical specifications, user guides, and application notes.

2 Microcontroller Characterization System Implementation

The firmware executing on the DUT during the characterization implements a loop, reading a portion of the FRAM to validate that the correct checkerboard pattern exist in the array. This checkerboard pattern is written only on the very first execution of the test prior to any irradiation. The FRAM is accessed by the internal controller 64 bits at a time. Each test loop reads 64 bits (8 bytes) at a time to match the physical access to the FRAM. The value read is compared to the expected value. If the value does not match expected, then a single FRAM error is counted and stored in a persistent variable in the FRAM. The correct value is also written back into the FRAM. At this point, the DUT de-asserts a signal (Booted) on P1.2 causing an external monitoring controller to close the shutter and block the beam prior to asserting reset. This methodology of resetting DUT after a single detected error allows for error checking and correction to occur immediately. This helps prevent the propagation of a systematic error. Additionally, further corruption of the checkerboard due to a corrupted comparison value is also prevented.

The implementation of the FRAM checkerboard read consists of reading a checkerboard array occupying address range 0x4800 to 0xE700. This represents 40 704 bytes read and compared for each loop. The loop is constructed to allow for a variable duty cycle by implementing a timer interrupt to trigger the start of the read loop. The data presented here is based on a 100% duty cycle. The interrupt for the timer is set to expire faster than the loop can complete. This results in the test immediately starting the next read loop after completing a loop. Figure 1 shows the software flow diagram.

During boot, the DUT performs various tasks including error checking, error correction, error logging, and normal device initialization requirements. At first power up or power on reset (POR), the device runs a pre-init routine to check the status of the stored SYSRSTIV register. The SYSRSTIV register is stored in FRAM when any of the SYSRSTIV sources are triggered before the device automatically executes a reset. Thus, when one of these critical interrupts is triggered resulting in a system reset, the cause can be determined on next boot and logged. See Section 3.1 for more information on SYSRSTIV and other interrupt sources. Execution of the boot code caused by SYSRSTIV reset source cannot be protected from the beam with the mechanical shutter during the reboot due to amount of time required to externally detect that a reset has occurred. After execution of the pre-init function a special section of code called BOOTINIT is copied from the upper block of FRAM (FRAM2) storage location into RAM. This code contains the error checking routine to calculate CRC on all FRAM code space.

The DUT evaluates if this is the very first boot. If it is the very first boot, then some extra functions are executed, including unlocking the Memory Protection Unit (MPU) and copying the code space from 0xE800-0x10400 to 0x11000. Figure 2 shows the FRAM Memory Map. Based on placement of code in the memory map, this effectively creates a backup copy of main code, interrupt vectors, and the BOOTINIT code. During this very first boot, a CRC code is calculated using the built in CRC engine (CRC-CCITT Standard) and stored in a persistent variable. After completing the first boot setup, all subsequent boots continue with checking the status of some key persistent variables to evaluate if they have been corrupted. The primary variables of concern are the static variables named data0 and data1 used to compare the checkerboard in the FRAM. These variables are checked against a copy that is only accessed during boot. If the two do not match, the backup copy is written back to the primary copy along with logging the data0/1 corruption. This methodology has proven effective to eliminate systemic errors introduced from the data comparison value being corrupted versus the value being read from the array being corrupted. Along with data0/1 corruption check, the entire code space has the CRC value calculated and compared to the value stored at first boot. If the value does not match, then the backup copy of entire code is copied from FRAM2 0x11000 back into 0xE800. This error is logged and DUT rebooted.

After completion of the error checking, the device is initialized to operate with an 8 Mhz external crystal oscillator. The initialization routine also configures necessary GPIO pins, a timer for read duty cycle, and the watchdog timer. The first test run at Berkeley implemented the clock initialization and other configuration prior to the error checking routine. A single run was identified in which an event created a corruption to the clock initialization routine. This corruption caused the clock initialization to fail and, hence, the DUT to go into a never ending reboot loop. The corruption could not be repaired since it occurred prior to the error correction routines. By moving the clock initialization after the error checking, it allowed the error checking to operate with default DCO clocking and, thus, allowing the clock and other initialization code to be repairable.

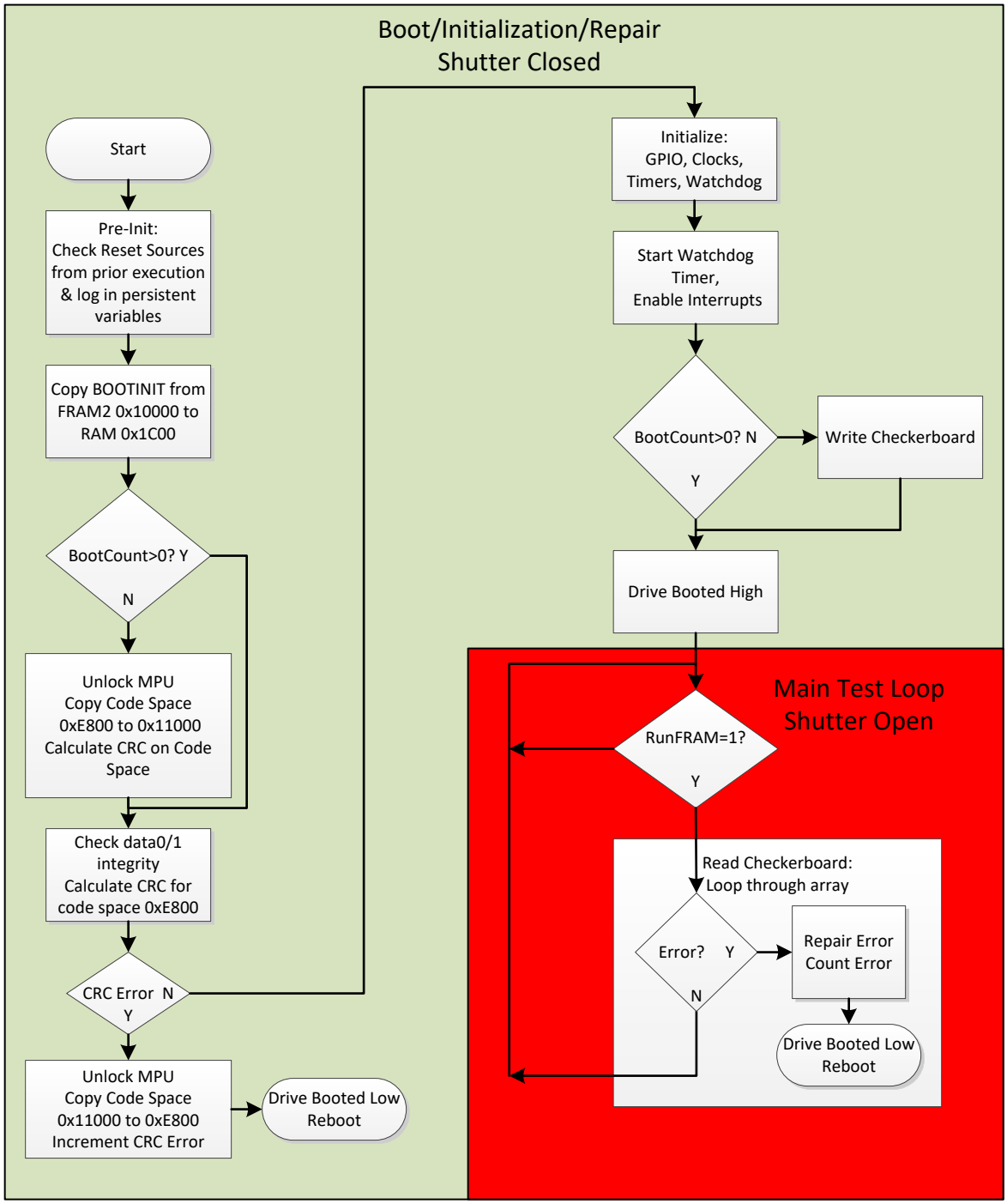


Figure 1. Software Flow Diagram

3 Key Considerations

There are multiple key considerations when implementing a radiation test suite on the MSP430FR5969-SP. These same considerations are equally important when implementing the associated software and system that is deployed in radiation environment.

3.1 Error Handling

It is imperative that all critical interrupts are trapped and handled. This characterization routine handles interrupts as follows. All interrupts that occur that are not part of systems required operation are handled by counting the occurrence of the interrupt in the interrupt service routine, then triggering a reboot. This allows determination of the quantity and source of erroneous interrupts and, also, invoke the error detection and correction algorithms that are part of the boot process. Interrupts triggered by SYSRSTIV are a special case. SYSRSTIV is the reset interrupt vector and it represents a critical event that does not allow interrupt service routine execution when they occur (such as a brown out event). The MCU stores the value of the SYSRSTIV in FRAM as part of the reset process. This allows the firmware to evaluate the status of SYSRSTIV value stored at next boot to determine the source of the last reset.

This characterization test suite also implements interrupt service routine (ISR) handling for all SYSSNIV and SYSUNIV sources along with post reset logging of all SYSRSTIV sources. Future enhancements may be employed to include generic reset triggering for all unused peripheral interrupt sources. These are not expected to have significant change to results, as two independent errors are be required to cause an issue. It would require the interrupt to be enabled, and then an actual erroneous trigger of the unused peripheral interrupt.

3.2 MPU Configuration

Use of the Memory Protection Unit (MPU) is also critical as it adds another layer of protection. The MPU can protect memory partitions by assigning privileges to each of three segments. The privileges can be combinations of (R)ead, (W)rite, and e(X)ecute. For example, the memory segment occupied by code is assigned RX privileges. By assigning these privileges and enabling the MPU, the MPU can prevent an erroneous attempt to write to code space. An attempt to write to a segment of memory that is not enabled for write, can cause either an NMI reset or a power up clear (PUC) depending on settings. This is true as well for an attempt to execute with data read from memory that does not have execute privilege. Use of the MPU can be somewhat challenging, as there are only three segments allowed, and they are defined by two addresses. An additional factor that plays into the best selection of the MPU segment boundaries is the fact that the ISR vectors are located at the end of FRAM1 at the boundary of FRAM2 just below 0x10000. Another constraint is related to the added overhead and complexity of executing code in FRAM2 above the 0x10000 boundary. With these constraints in mind, the memory map and MPU segments below were chosen. FRAM2 is used only as backup copy of code, and does not have write or execute privilege. Figure 2 shows the FRAM memory map.

3.3 Logical Concerns

Take care with the use of variables, constants, and pointers that are accessed frequently, as they are more prone to corruption. Periodic validation of these variables should occur, especially after an error has been detected. Validation of the variable is dependent on the system usage. A back up copy (or copies) of variables can also be stored and accessed infrequently. These backups copies can be used to validate and replace a variable that is constantly in use for system operation. Variables that have known bounded ranges from system operation can also be validated to be within proper range. Three or more copies of these key variables can be updated and used for voting to assure that errors are detected and corrected. Small loops can also be a concern, as there is a high likelihood that all of the code is stored in the more sensitive cache. Since the data may be static in the cache for longer periods of time, it is potentially more susceptible to corruption. It is worthy to note, that the cache is also preventing a write back to FRAM due to a potentially corrupted value. This can also lead to a logical error, without any persistent code corruption.

3.4 Memory Map

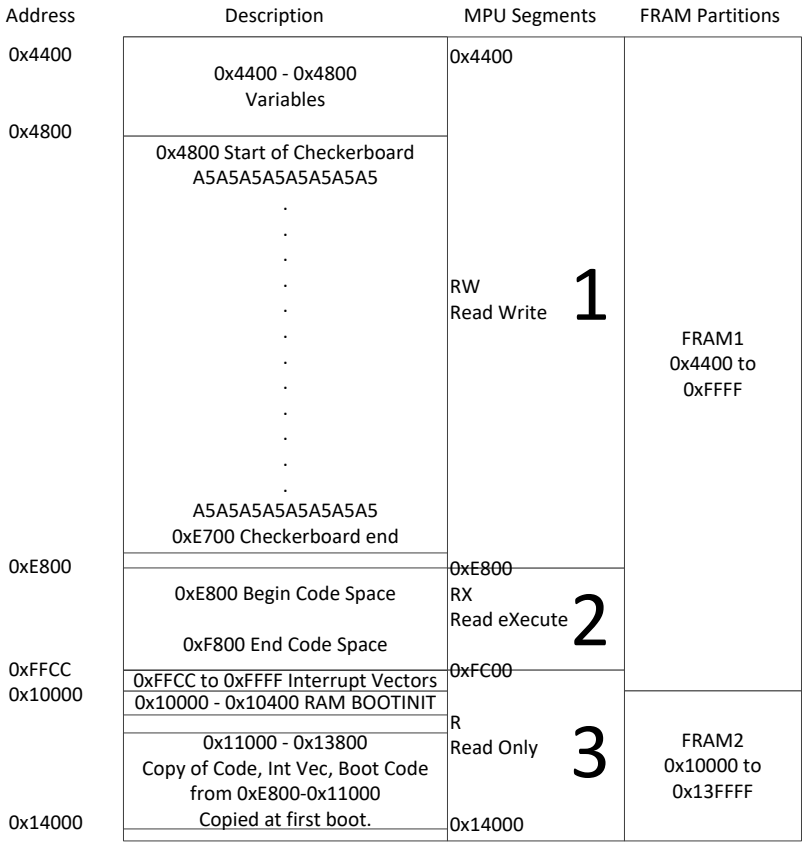


Figure 2. FRAM Memory Map

The FRAM memory map shows the three defined MPU segments along with the location of the various aspects of the system firmware. The lower range of FRAM1 contains the error logging variables from 0x4400 to just below 0x4800. Starting at 0x4800, the area allocated for the checkerboard exists up until 0xE700. A gap from 0xE700 to start of code at 0xE800 was intentionally left to help protect against a pointer index getting corrupted and exceeding the area allocated for checkerboard and accidentally writing into code space. The memory range starting at 0xE800 up to 0xF800 contains the main code, and is contained in MPU segment two with RX privilege. At 0xFC00 MPU segment three is defined to be read only. Note, that this includes the interrupt vector offsets located from 0xFFCC to 0xFFFF. This is not an issue, as the MPU knows this region, and gives it special privilege without requiring explicit execute privilege.

The BOOTINIT code is stored starting in FRAM2 at 0x10000 with read only privilege. This code does not execute from FRAM2, but is copied into RAM during boot. Note, this is the only usage of RAM. All other execution and stack remains in FRAM. The region beginning at 0x11000 to the end of FRAM2 is used for a copy of the entire code region. This includes the main code, the interrupt vectors, and a copy of BOOTINIT. During the first the boot this region is written with a copy of the code. The MPU is unlocked during this transaction to allow this region to be written to. After reboot, the MPU is re-enabled protecting this region from an error induced write.

4 Irradiation Facilities and Setup

The heavy-ion species used for the SEE studies on this product were provided and delivered by the TAMU Cyclotron Radiation Effects Facility [1], and the Lawrence Berkeley 88-inch cyclotron, using a superconducting cyclotron and advanced electron cyclotron resonance (ECR) ion source, respectively. At the fluxes used, ion beams had good flux stability and high irradiation uniformity over a 1-in diameter circular cross sectional area for the in-air station (TAMU) and vacuum chamber (Berkeley). Uniformity is achieved by means of magnetic defocusing. The flux of the beam is regulated over a broad range spanning several orders of magnitude. For the bulk of these studies, ion fluxes between 6.7×10^2 and 2.2×10^4 ions/s-cm^2 were used to provide heavy-ion fluences between 1×10^6 and 5×10^6 ions/cm^2.

For these experiments, neon (^{20}Ne) copper (^{63}Cu), silver (^{109}Ag), and xenon (^{136}Xe) ions, at angles of 0° and 28°, were used for LET_{EFF} of 3.3 to ~61.5 MeV-cm^2/mg. The ions used had a total kinetic energy of 400 MeV to 2.114 GeV, in the vacuum (15-MeV/amu line at TAMU and 10-MeV/amu at Berkeley).

The MSP430FR5969-SP test setup used for the experiments at the TAMU facility is shown in Figure 5. Although not visible in this photo, the beam port has a 1-mil Aramica (DuPont® Kevlar®) 1-in diameter window to allow in-air testing while maintaining the vacuum within the accelerator with only minor ion energy loss. The air space between the device and the ion beam port window was maintained at 40 mm for all runs in TAMU. Lawrence Berkeley 88-inch cyclotron did not provide an in-air solution; rather, the device is tested in a vacuum chamber.

5 Depth, Range, and LET_{EFF} Calculation

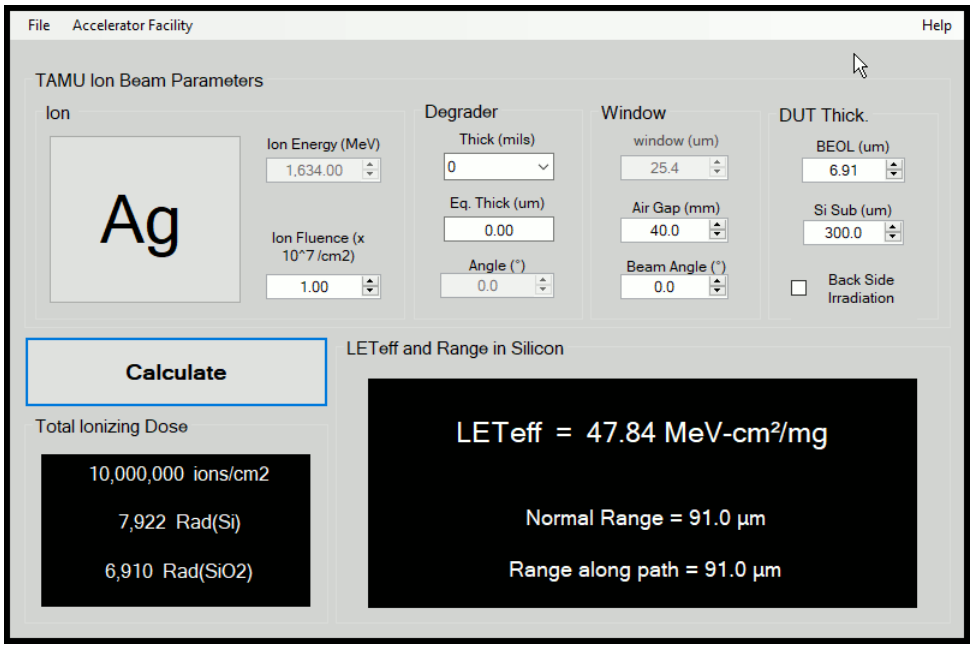


Figure 3. GUI of RADsim Application Used to Determine Key Ion Parameters

The MSP430FR5969-SP is fabricated in the HPE035 technology with a back-end-of-line (BEOL) stack consisting of six levels of metal with a nominal stack height of 5.515 µm. The total stack height from the surface of the passivation to the silicon surface is 6.91 µm based on nominal layer thickness. Accounting for energy loss through the 1-mil thick Aramica beam port window, the 40-mm air gap (valid for TAMU), and the BEOL stack over the MSP430FR5969-SP, the effective LET (LET_{EFF}) at the surface of the silicon substrate, the depth, and the ion range were determined with the custom RADsim - IONS application (developed at Texas Instruments and based on the latest SRIM-2013 [2] models). Table 2 shows the results of the simulator. The stack was modeled as a homogeneous layer of silicon dioxide (valid since SiO_2 and aluminum density is similar).

Table 2. Ion LET_{EFF}, Depth, and Range in Silicon

ION TYPE	ANGLE OF INCIDENCE (°)	DEPTH IN SILICON (µm)	RANGE IN SILICON (µm)	LET_{EFF} (MeV-cm²/mg)	FACILITY
Ne	0	176.8	176.8	3.3	Berkeley
Cu	0	116.4	116.4	20.3	TAMU
Cu	0	101.9	101.9	21.5	Berkeley
Cu	28	102.0	115.5	23.0	TAMU
Ag	0	91.0	91.0	47.8	TAMU
Ag	0	83.0	83.0	49.6	Berkeley
Ag	28	79.6	90.1	54.4	TAMU
Xe	0	81.7	81.7	61.5	Berkeley

6 SEE Definitions/Methodology

Many single event upsets are recorded and categorized by the DUT. These include every interrupt source, along with other error sources that are functions of the error checking and correction algorithms. There are 48 different error types that are captured as part of the test system and DUT firmware logging. The vast majority of the DUT logged interrupt based events do not have any events recorded throughout entire testing. To simplify creation and communication of the cross sections, the event types have been reduced down to five types.

6.1 Testing Methodology

During beam testing, the DUT outputs a heartbeat signal during the execution of each read loop. This signature is monitored with an oscilloscope to determine if the test system is still functioning. There are five SEE types defined based on results of the testing. All of the types are defined as Single Event Functional Interrupts (SEFI) with the exception of the Error Detected type. The SEFI types are defined this way based on the fact that a reset is required to recover. The Error Detected type is defined as Single Event Upsets (SEU). However, the distinctions are not clear due to the nature that all recoverable errors are recovered through a reset and reboot with error correction. The types defined are the following:

- Program SEFI
- Power SEFI
- Error Detected
- SelfReset SEFI
- Pre-Probe

6.2 Program SEFI

A Program SEFI requires reprogramming the firmware in order to return the DUT to normal system operation. There were a total of three occurrences of this event type in all of the testing. Two of these three occurred during Berkeley testing, and the third at TAMU. Based on reverse engineering the post memory dump, the first occurrence indicated that there was a corruption to the code responsible for the clock initialization. This caused the DUT to go into reboot loop each time it encountered the corrupted code. This error was made to be correctable with a change in the firmware order. The bulk of the firmware initialization code was moved to post error detection and correction. This is detailed in Section 2. The test runs executed at TAMU facility included this enhancement. The second occurrence is somewhat similar, as it appears that the CRC check value became corrupted. This caused the firmware to constantly go into a reboot loop to attempt to repair the code. This error can be prevented with storing the CRC in multiple locations and using a vote in case of CRC mismatch. The final occurrence left the array erased when attempting to read the memory dump. This signature is known to be caused by corruption of the ISR vectors. When a Boot Strap Loader (BSL) access is attempted, the BSL programmer must respond with the correct BSL key. This key is a function of the values stored for the Interrupt Service Routine (ISR) vectors. If the key does not match, the BSL ROM code erases the entire FRAM array as a means to prevent reverse engineering and protect IP. It is speculated that two events occurred to cause this and prevent the MCU reboot from correcting the error in the ISR space. This is believed to be a function of the accelerated testing environment and not a signature that would occur with much lower ion fluxes. More information on BSL programming and ISR keys is located in the *MSP430FR58xx, MSP430FR59xx, and MSP430FR6xx Family User's Guide*.

Additional precautions can be implemented to mitigate any potential occurrence of a Program SEFI. A backup unpowered MSP430FR5969-SP can be designed to wake up in the event that the primary MSP430FR5969-SP stopped responding. This backup MCU can reprogram the firmware via BSL programming and resume normal system operation.

6.3 Power SEFI

Each loop through the checkerboard read results in the pulse of a GPIO that represents the heartbeat of the DUT. If the heartbeat signature stops, or changes frequency, the operator applies a manual reset to the system. If this does not restore functionality, a power cycle is applied. If this restores functionality, it is recorded as a power SEFI and execution of the test continues. There were a total of 53 occurrences of a power SEFI.

6.4 Error Detected SEU

The Error Detected SEU type captures all errors that the MCU was able to detect, record, and issue its own controlled reset. This SEU type contains errors generated from all interrupts from SYSNMI and SYSUNIV sources along with all the errors detected during the checkerboard comparison. Most of these errors are the result of checkerboard comparison error detected during the read of the checkerboard loop. When the checkerboard is read, and the 8 bytes of compared data do not match the expected value, the error count is incremented, along with correcting the value in the checkerboard. At this point, the DUT is reset and test resumes after boot. Keep in mind that the boot process checks the integrity of the compare

values for the checkerboard. If these values (data0 and data1) are in error, they are corrected and logged. Following this process, the boot process finishes with a CRC check of the entire code space. If the CRC value does not match, the entire code space is replaced and logged. The full Error Detected SEU rate is determined from the sum of FRAM checkerboard errors, data0/1 errors, and the number of times that the CRC code detected and repaired code space along with the SYSNMI and SYSUNIV sources. It is likely that a data0/1 error is responsible for more than one error. The data0/1 value likely caused the original FRAM read error, and would also cause the incorrect data compare word to be written into the FRAM. So, one data0/1 error is counted as three errors (two FRAM read errors, and one data0/1 error). This count is not adjusted, so, in this respect, the event rate is pessimistic. Additionally, the CRC code replacement has no way to indicate the number of bits in error.

Using this algorithm, a CRC code space error was detected and corrected a total of 14 times. The SYSNMI and SYSUNIV error counts are comprised of the following interrupt sources:

- SYSSNIV_NONE
- SYSSNIV_RES02
- SYSSNIV_UBDIFG
- SYSSNIV_RES06
- SYSSNIV_MPUSEGPIFG
- SYSSNIV_MPUSEGIIFG
- SYSSNIV_MPUSEG1IFG
- SYSSNIV_MPUSEG2IFG
- SYSSNIV_MPUSEG3IFG
- SYSSNIV_VMAIFG
- SYSSNIV_JMBINIFG
- SYSSNIV_JMBOUTIFG
- SYSSNIV_CBDIFG
- SYSUNIV_NONE
- SYSUNIV_NMIIFG
- SYSUNIV_OFIFG

Of these, only 8 have recorded errors:

- SYSSNIV_VMAIFG 138
- SYSUNIV_OFIFG 61
- SYSSNIV_NONE 18
- SYSSNIV_JMBOUTIFG 9
- SYSSNIV_CBDIFG 9
- SYSSNIV_MPUSEGPIFG 8
- SYSSNIV_JMBINIFG 3
- SYSUNIV_NMIIFG 3

The most frequent interrupts came from VMAIFG (Vacant memory access) and OFIFG (Oscillator fault). There were a total of 692 errors detected for this category across all runs.

6.5 Self Reset SEFI

Self Reset SEFI type represents all errors that caused immediate reset of the MCU without ability to execute code to detect or count the error when it occurred. These all are represented by triggering SYSRSTIV sources. When an SYSRSTIV source is triggered, the MCU will save the value of the SYSRSTIV register in FRAM as part of the reset process. Unfortunately, this error type cannot be detected quickly enough to trigger the external shutter to close prior to the reset sequence completing. The following is a list of reset types, and their count, for all included test runs:

- SYSRSTIV_RSTNMI 1440
- SYSRSTIV_BOR 805

- SYSRSTIV_SECYV 49
- SYSRSTIV_SVSHIFG 7
- SYSRSTIV_NONE 3
- SYSRSTIV_MPUSEG1IFG 2
- SYSRSTIV_DOBOR 1
- SYSRSTIV_RES0C 1
- SYSRSTIV_WDTTO 1
- SYSRSTIV_PERF 1
- SYSRSTIV_CSPW 1
- SYSRSTIV_ACCTEIFG 1
- SYSRSTIV_LPM5WU 0
- SYSRSTIV_RES10 0
- SYSRSTIV_RES12 0
- SYSRSTIV_DOPOR 0
- SYSRSTIV_WDTKEY 0
- SYSRSTIV_FRCTLPW 0
- SYSRSTIV_UBDIFG 0
- SYSRSTIV_PMMPW 0
- SYSRSTIV_MPUPW 0
- SYSRSTIV_MPUSEGPIFG 0
- SYSRSTIV_MPUSEGIIFG 0
- SYSRSTIV_MPUSEG2IFG 0
- SYSRSTIV_MPUSEG3IFG 0

RSTNMI has an exaggerated count due to the fact that every attempt to read the memory of the DUT will cause one reset to occur. Every attempt to reset the device in case of heartbeat stop also causes an additional RSTNMI to be counted. The individual meanings of each interrupt source can be examined by referencing the *MSP430FR58xx, MSP430FR59xx, and MSP430FR6xx Family User's Guide*.

6.6 Pre-Probe

The last error type is the single most problematic type. This error type represents the DUT becoming unresponsive and not recovering after a reset, or power cycle, and is unable to be reprogrammed. During these combined test runs, this error type has only occurred a single time. However, prior data from third parties indicate that this error type is possible. Prior to these test runs, data from a single unit from third party that exhibited this behavior was examined. The theory of the failure is that low level boot data is getting corrupted and causing the MCU to not initialize for final operation. During the low level boot, the MCU evaluates stored values that represent elements of the device test and configuration flow. If these values do not have proper values, or if they do not have correct checksums, then the low level ROM boot code assumes that the device has not been tested yet. If the MCU detects that values are not in the final state, it assumes that the device is ready to be probed, so it is mass erased and put in a state that is ready to be probed. This state does not allow the MCU to function as a final device. The third party evaluation unit, and the single unit from the TAMU testing, were evaluated and conclusively determined that they were both in the pre-probe state.

With the understanding of the device boot flow, along with the robustness of the FRAM memory, it is theorized that this catastrophic error has an extremely low cross section as it requires the low level boot data to be corrupted. This corruption can only be possible during an extremely small window of time that the boot data is being accessed immediately after powering up or after executing a BOR. The reading of the low level boot data occurs once per BOR or power cycle. Thus, the occurrence of this error is deemed to be due to accelerated testing and frequent exposure of the DUT to low level resets in the beam.

7 Shutter Based Boot Protection Scheme

Early data from third parties that experienced a pre-probe device led to attempts to limit the exposure of the DUT to the ion flux after a reset, and during the boot process. Limiting the ion exposure of the low level boot code and the code performing error checking and correction is intended to create more realistic cross sections. Exposing the device to higher flux ions without means to protect the boot process introduces secondary events into code that is already responding to an event. Unfortunately, the self reset category of events cannot be detected quickly enough to be able to close the shutter prior to the low level boot data access. Unit 2 tested at TAMU that became pre-probe in run 28 was confirmed that the last error recorded was a self reset. All other event types are controlled, so the resets and booting are protected, as the DUT signals the external shutter controller to close the shutter prior to externally asserting reset. Figure 4 shows the shutter controller state diagram. In the cases where the DUT was reset from a SYSRSTIV source, the device already has read the low level boot data and is executing user code by the time the shutter was completely closed. The mechanical shutter requires approximately 25 ms to close once the signal to close the shutter is driven. The DUT completes the reading of the low level boot data in less than 1 ms in most cases.

The state diagram for the shutter controller indicates input signals on the top of the bubble and outputs on the bottom. The signal "Beam" represents status of the beam from the test facility. The signal "BootedIn" is the output from the DUT indicating that the boot has completed and DUT is ready for test. It also indicates when a detectable error has occurred and signals the shutter controller to assert DUT reset after closing the shutter. This can be seen in state transition two to three. In the event of a self reset, the same process occurs, however, the "BootedIn" signal is tri-stated from the DUT during the reset process versus driven low. The output signal "ShutEn" is used to drive the shutter to open or close. "ResetOut" is used as an open drain output to actively drive the DUT reset line low while waiting for shutter to close. Finally, the output signal "Hold" is used to tell the DUT to not execute the read checkerboard routine while waiting for the test to begin. A test mode is implemented to allow simulation of the beam being on, so the heartbeat signal can be confirmed prior to evacuating the cave.

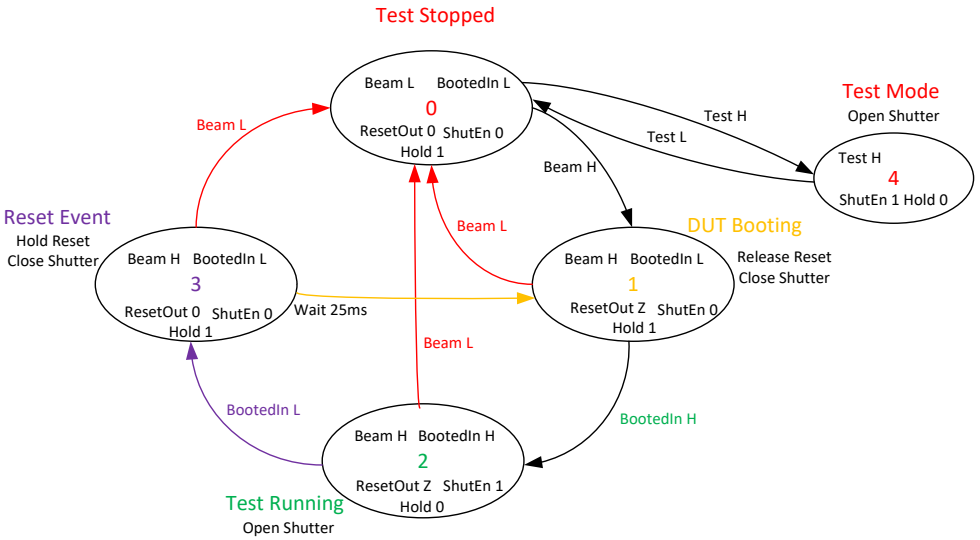


Figure 4. Shutter Controller State Diagram

8 Test Setup

The MSP430FR5969-SP DUT test setup is shown outside the beam in Figure 5 using an MSP-TS430RGZ48C modified socketed EVM along with an Elprotronic Xstream ISO programmer and external MSP430FR5969 based controller and monitor. The BSL tool is instrumental during the testing to perform manual resets to the DUT if the heartbeat was missing. The BSL tool also is used to program the DUT, and read the memory for post run analysis.

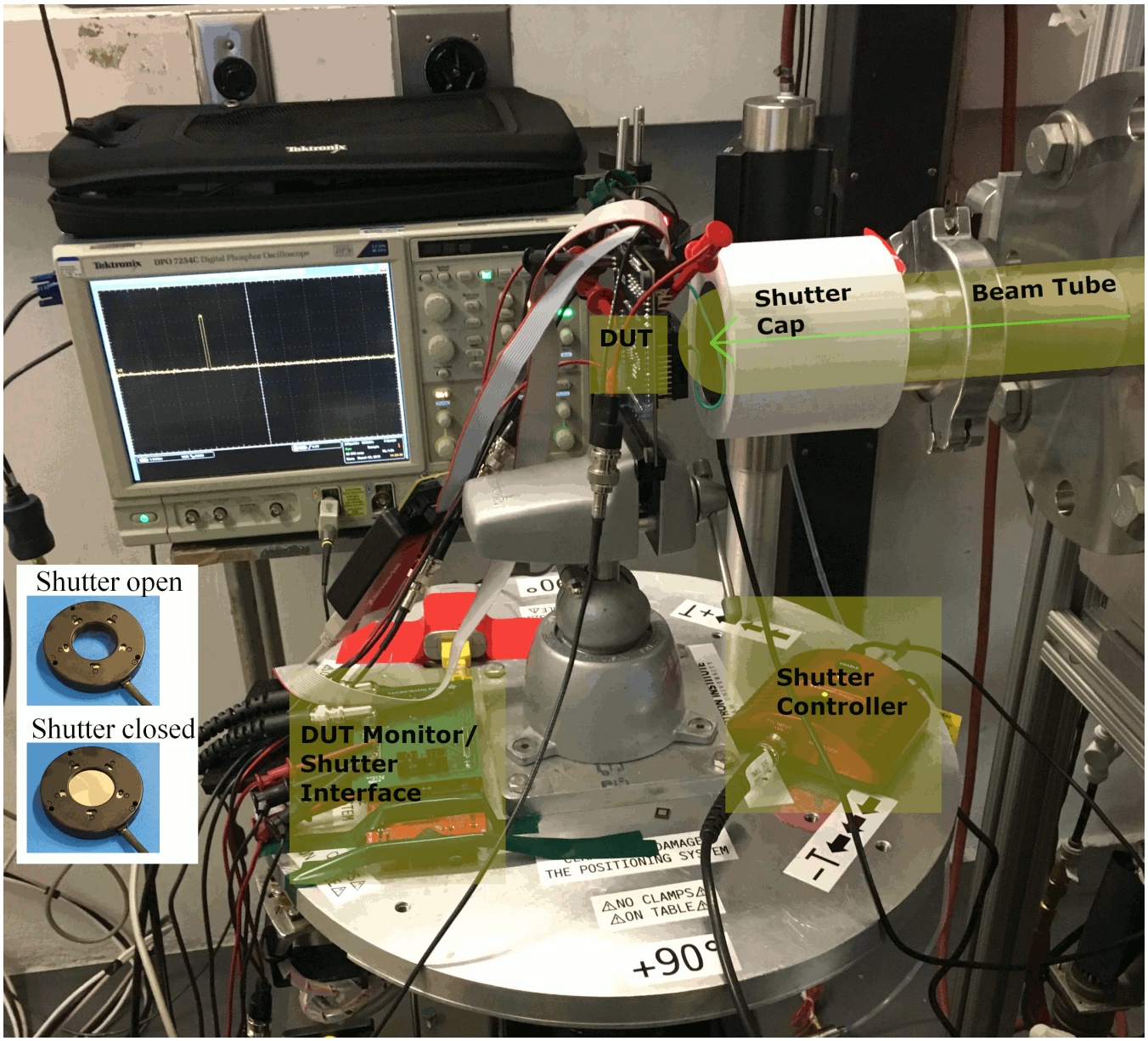


Figure 5. Photograph of the MSP430FR5969-SP Mounted in Front of the Heavy-Ion Beam Exit Port at the Texas A&M Cyclotron

9 Results

Table 3 represents a summary of all valid test runs used to generate this report. Table 4 contains the breakdown of the number of events by SEE type.

The cross section plots shown in through are based on combined errors from all runs with the same LET_{EFF}. Error bars represent the 95% confidence level. For error counts less than 50, a Poisson distribution is used to calculate the 95% confidence level. For errors counts higher than 50, the error bars are estimated as normal distribution with two standard deviations.

Table 3. Summary of MSP430FR5969-SP Radiation Runs

RUN #	DEV #	FACILITY	ION TYPE	ANGLE OF INCIDENCE (°)	LET_{EFF} (MeV-cm^2/mg)	FLUX (ions/s-cm^2)	FLUENCE (ions/cm^2)
6	1	Berkeley	Ag	0	49.6	2.00×10^3	1.00×10^6
7	1	Berkeley	Ag	0	49.6	2.00×10^3	4.78×10^5
8	1	Berkeley	Ag	0	49.6	2.00×10^3	1.00×10^6
9	1	Berkeley	Cu	0	21.5	2.20×10^4	1.00×10^6
10	1	Berkeley	Cu	0	21.5	2.80×10^3	1.00×10^6
12	2	Berkeley	Cu	0	21.5	2.35×10^3	1.00×10^6
13	2	Berkeley	Cu	0	21.5	2.35×10^3	1.00×10^6
14	2	Berkeley	Ne	0	3.3	2.35×10^3	1.00×10^6
15	2	Berkeley	Ne	0	3.3	2.35×10^3	1.00×10^6
16	2	Berkeley	Ne	0	3.3	2.35×10^3	1.00×10^6
17	2	Berkeley	Xe	0	61.5	2.50×10^3	7.15×10^5
18	2	Berkeley	Xe	0	61.5	2.50×10^3	1.00×10^6
19	2	Berkeley	Xe	0	61.5	2.50×10^3	1.00×10^6
5	3	TAMU	Ag	0	47.8	7.34×10^2	1.09×10^6
6	3	TAMU	Ag	0	47.8	6.76×10^2	1.00×10^6
7	3	TAMU	Ag	28	54.4	3.83×10^3	5.00×10^6
27	4	TAMU	Cu	0	20.3	1.57×10^3	5.00×10^6
28	4	TAMU	Cu	28	23.0	1.44×10^3	4.08×10^6
32	4	TAMU	Cu	28	23.0	1.76×10^3	5.00×10^6

Table 4. Event Summary of MSP430FR5969-SP

RUN #	DEV #	FACILITY	LET_{EFF} (MeV-cm^2/mg)	FLUENCE (ions/cm^2)	EVENTS				
					Program SEFI	Power SEFI	Error Detected	SelfReset SEFI	Pre-Probe
6	1	Berkeley	49.6	1.00×10^6	0	0	16	49	0
7	1	Berkeley	49.6	4.78×10^5	1 (1)	0	8	19	0
8	1	Berkeley	49.6	1.00×10^6	0	1	19	52	0
9	1	Berkeley	21.5	1.00×10^6	0	1	3	19	0
10	1	Berkeley	21.5	1.00×10^6	0	0	8	26	0
12	2	Berkeley	21.5	1.00×10^6	0	1	10	29	0
13	2	Berkeley	21.5	1.00×10^6	0	0	7	24	0
14	2	Berkeley	3.3	1.00×10^6	0	1	4	4	0
15	2	Berkeley	3.3	1.00×10^6	0	2	1	5	0
16	2	Berkeley	3.3	1.00×10^6	0	0	1	2	0
17	2	Berkeley	61.5	7.15×10^5	1 (1)	0	48	136	0
18	2	Berkeley	61.5	1.00×10^6	0	3	26	63	0
19	2	Berkeley	61.5	1.00×10^6	0	3	33	71	0
5	3	TAMU	47.8	1.09×10^6	0	1	34	65	0
6	3	TAMU	47.8	1.00×10^6	0	2	53	63	0
7	3	TAMU	54.4	5.00×10^6	1	11	130	305	0
27	4	TAMU	20.3	5.00×10^6	0	11	149	132	0
28	4	TAMU	23.0	4.08×10^6	0	9	82	159	1
32	4	TAMU	23.0	5.00×10^6	0	7	60	157	0

(1) This Program SEFI should be preventable with improved firmware.

Table 5. Combined Run Cross Section Summary

RUN(S) #	DEV(S) #	FACILITY	LET_{EFF} (MeV-cm²/mg)	FLUENCE (ions/cm²)	CROSS SECTION cm² / DEVICE				
					Program SEFI	Power SEFI	Error Detected	SelfReset SEFI	Pre-Probe
14,15,16	2	Berkeley	3.3	3.00×10^{6}	1.23×10^{-6} (1)	1.00×10^{-6}	2.00×10^{-6}	3.67×10^{-6}	1.23×10^{-6} (1)
27	4	TAMU	20.3	5×10^{6}	7.38×10^{-7} (1)	2.20×10^{-6}	2.98×10^{-5}	2.64×10^{-5}	7.38×10^{-7} (1)
9,10,12,13	1,2	Berkeley	21.5	4.00×10^{6}	9.22×10^{-7} (1)	5.00×10^{-7}	7.00×10^{-6}	2.45×10^{-5}	9.22×10^{-7} (1)
28,32	4	TAMU	23.0	9.08×10^{6}	4.06×10^{-7} (1)	1.76×10^{-6}	1.56×10^{-5}	3.48×10^{-5}	1.10×10^{-7}
5,6	3	TAMU	47.8	2.09×10^{6}	1.77×10^{-6} (1)	1.44×10^{-6}	4.16×10^{-5}	6.12×10^{-5}	1.77×10^{-6} (1)
6,7,8	1	Berkeley	49.6	2.48×10^{6}	4.04×10^{-7}	4.04×10^{-7}	1.74×10^{-5}	4.84×10^{-5}	1.49×10^{-6} (1)
7	3	TAMU	54.4	5.00×10^{6}	2.00×10^{-7}	2.20×10^{-6}	2.60×10^{-5}	6.10×10^{-5}	7.38×10^{-7} (1)
17,18,19	2	Berkeley	61.5	3.00×10^{6}	3.33×10^{-7}	2.00×10^{-6}	3.57×10^{-5}	9.00×10^{-5}	1.23×10^{-6} (1)

(1) This cross section based on upper bound Chi-Squared distribution described in Appendix B.

9.1 Power SEFI

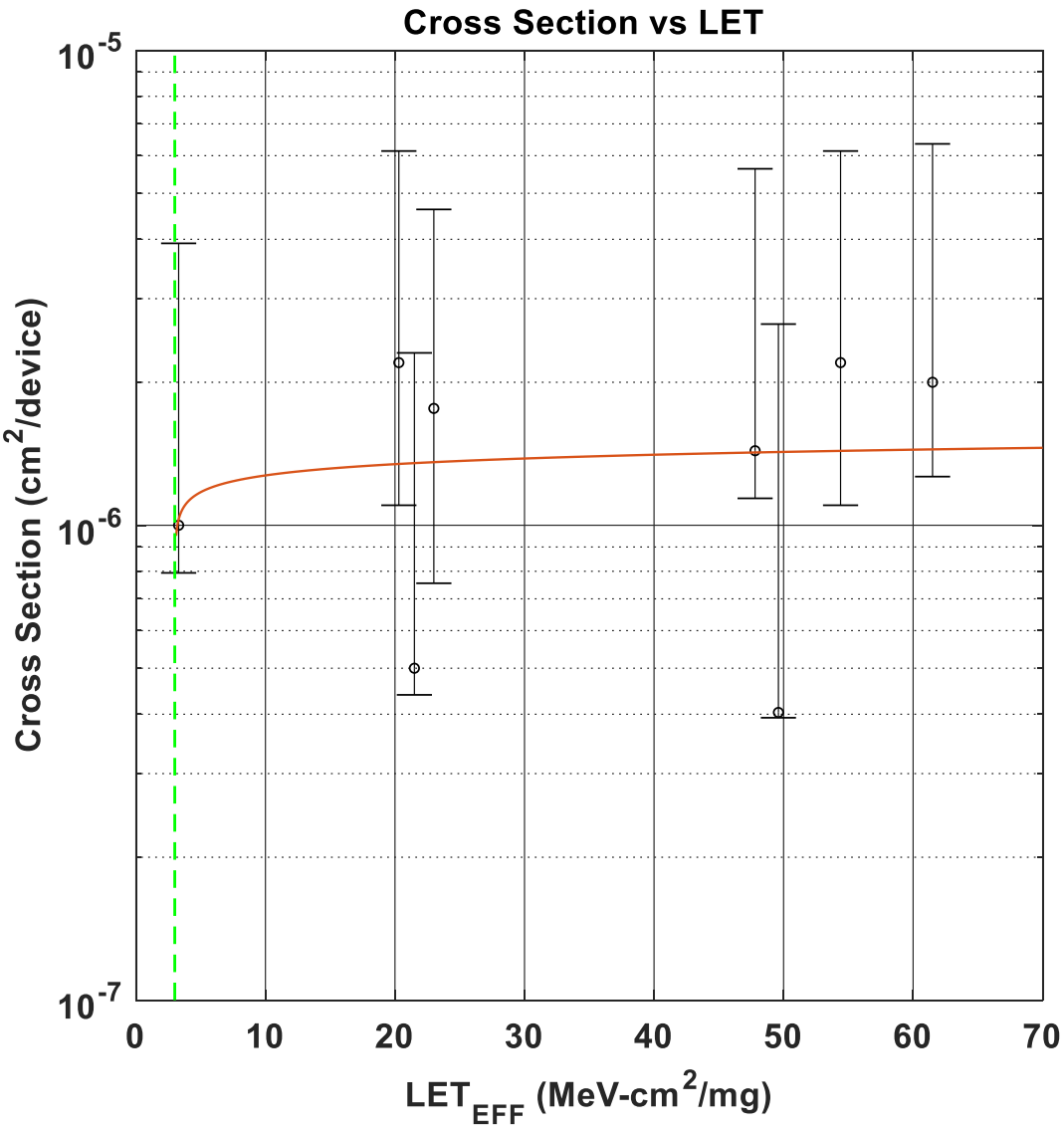


Figure 6. Combined Power SEFI Cross Section

Table 6. Power SEFI Weibull Parameters

PARAMETER	EVENTS FIT
σ_{SAT}	2.2×10^{-6}
Onset	3
W	30
s	0.1

9.2 Error Detected SEU

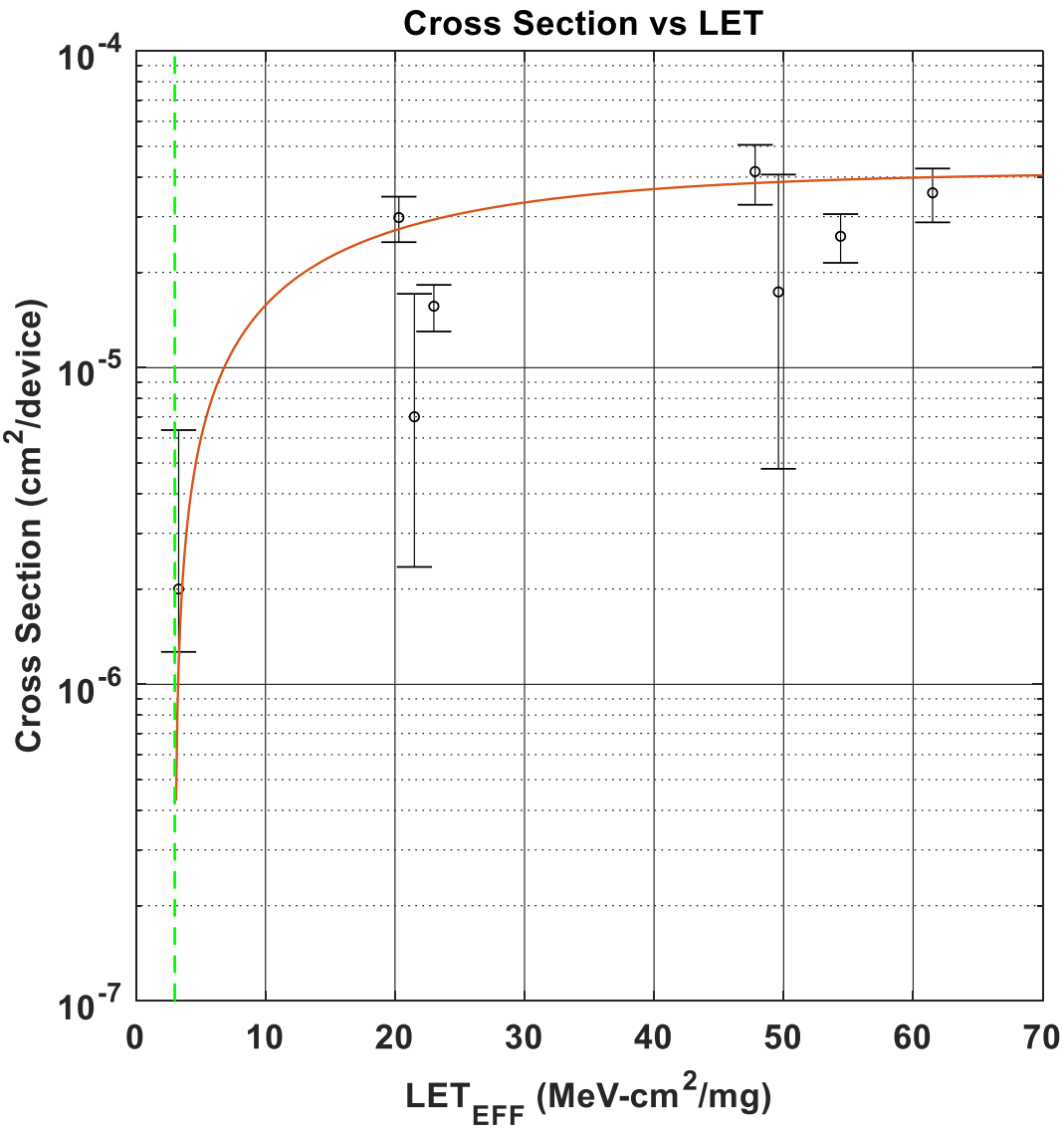


Figure 7. Combined Error Detected SEU Cross Section

Table 7. Error Detected Weibull Parameters

PARAMETER	EVENTS FIT
σ_{SAT}	4.16×10^{-5}
Onset	3
W	16

Table 7. Error Detected Weibull Parameters (continued)

PARAMETER	EVENTS FIT
s	0.9

9.3 Self Reset SEFI

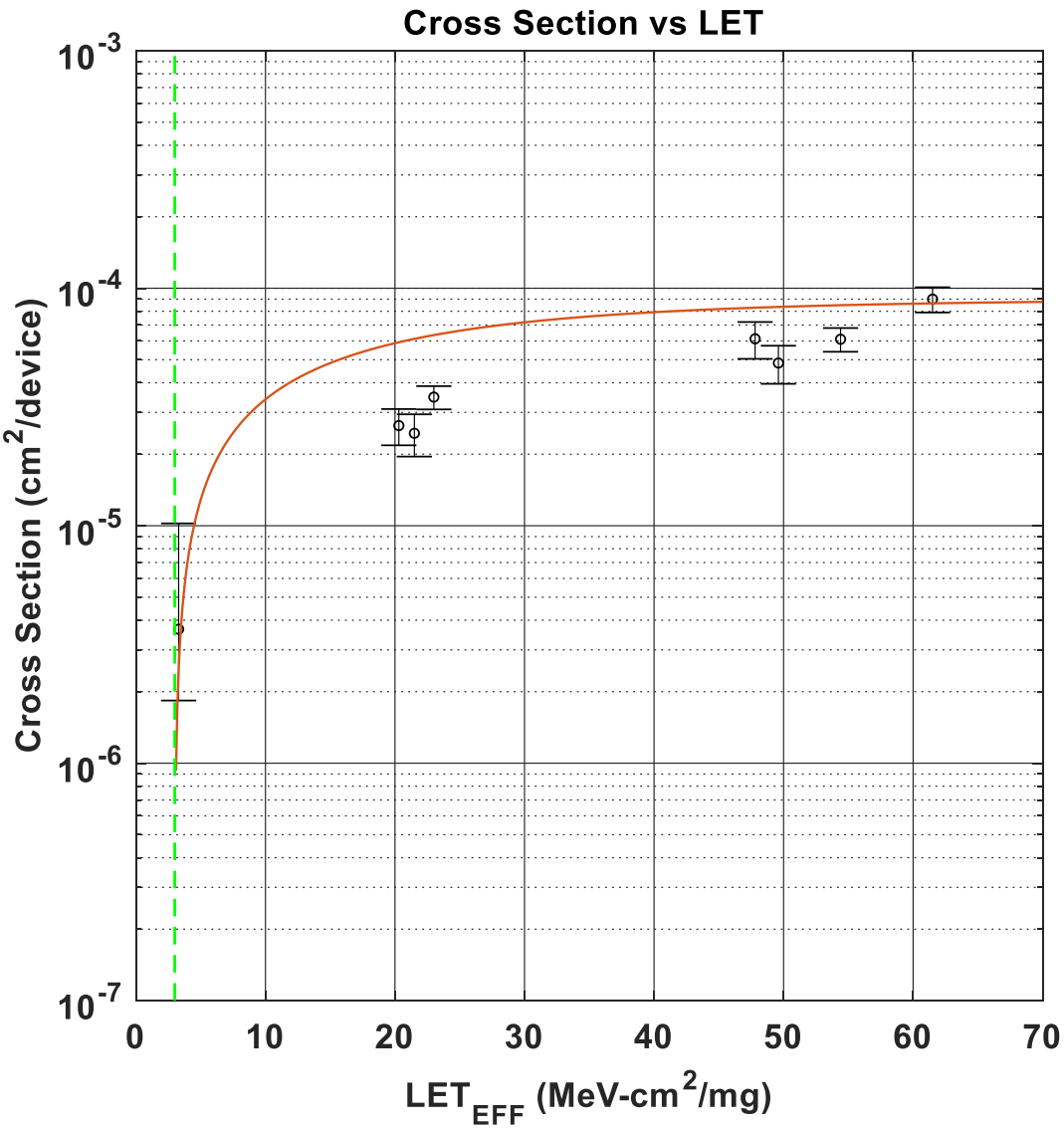


Figure 8. Combined Self Reset SEFI Cross Section

Table 8. Self Reset Weibull Parameters

PARAMETER	EVENTS FIT
σ_{SAT}	9.0×10^{-5}
Onset	3
W	16
s	0.9

9.4 Orbital Event Rate Calculations

Event rates were calculated for LEO (ISS) and GEO environments by combining CREME96 orbital integral flux estimations and simplified SEE cross-sections according to methods described in Appendix B. A minimum shielding configuration of 100 mils (2.54 mm) of aluminum and “worst-week” solar activity is assumed, which is similar to a 99% upper bound for the environment. Calculations are using the 95% upper-bounds for all event rates.

Table 9. Program SEFI Event Rate Calculations for Worst-Week LEO and GEO Orbits

ORBIT TYPE	ONSET LET_{EFF} (MeV-cm^2/mg)	CREME96 Integral FLUX (/day-cm^2)	σ_{SAT} (cm^2)	EVENT RATE (/DAY)	EVENT RATE (FIT)	MTBE (YEARS)
LEO (ISS)	49.6	4.09×10^{-4}	4.04×10^{-7} (1)	1.65×10^{-10}	6.88×10^{-3}	1.66×10^{7}
GEO		1.33×10^{-3}		5.38×10^{-10}	2.24×10^{-2}	5.09×10^{6}

(1) Based on Chi-Squared distribution with onset LET of next lower value of reported event.

Table 10. Power SEFI Event Rate Calculations for Worst-Week LEO and GEO Orbits

ORBIT TYPE	ONSET LET_{EFF} (MeV-cm^2/mg)	CREME96 Integral FLUX (/day-cm^2)	σ_{SAT} (cm^2)	EVENT RATE (/DAY)	EVENT RATE (FIT)	MTBE (YEARS)
LEO (ISS)	3	3.02×10^{2}	2.2×10^{-6}	6.65×10^{-4}	2.77×10^{4}	4.12
GEO		2.87×10^{3}		6.32×10^{-3}	2.64×10^{5}	4.33×10^{-1}

Table 11. Error Detected Event Rate Calculations for Worst-Week LEO and GEO Orbits

ORBIT TYPE	ONSET LET_{EFF} (MeV-cm^2/mg)	CREME96 Integral FLUX (/day-cm^2)	σ_{SAT} (cm^2)	EVENT RATE (/DAY)	EVENT RATE (FIT)	MTBE (YEARS)
LEO (ISS)	3	3.02×10^{2}	4.16×10^{-5}	1.26×10^{-2}	5.23×10^{5}	2.18×10^{-1}
GEO		2.87×10^{3}		1.19×10^{-1}	4.97×10^{6}	2.29×10^{-2}

Table 12. Self Reset SEFI Event Rate Calculations for Worst-Week LEO and GEO Orbits

ORBIT TYPE	ONSET LET_{EFF} (MeV-cm^2/mg)	CREME96 Integral FLUX (/day-cm^2)	σ_{SAT} (cm^2)	EVENT RATE (/DAY)	EVENT RATE (FIT)	MTBE (YEARS)
LEO (ISS)	3	3.02×10^{2}	9.0×10^{-5}	2.72×10^{-2}	1.13×10^{6}	1.01×10^{-1}
GEO		2.87×10^{3}		2.58×10^{-1}	1.08×10^{7}	1.06×10^{-2}

Table 13. Pre-Probe Event Rate Calculations for Worst-Week LEO and GEO Orbits

ORBIT TYPE	ONSET LET_{EFF} (MeV-cm^2/mg)	CREME96 Integral FLUX (/day-cm^2)	σ_{SAT} (cm^2)	EVENT RATE (/DAY)	EVENT RATE (FIT)	MTBE (YEARS)
LEO (ISS)	21.5	5.15	1.10×10^{-7} (1)	4.22×10^{-7}	1.76×10^{1}	6.49×10^{3}
GEO		4.00×10^{1}		3.27×10^{-6}	1.36×10^{2}	8.39×10^{2}

(1) Based on Chi-Squared distribution with onset LET of next lower value of reported event.

Total Ionizing Dose From SEE Experiments

The production MSP430FR5969-SP MCU is rated to a total ionizing dose (TID) of 50 krad(Si). In the course of the SEE testing, the heavy-ion exposures delivered ≈ 8 krad(Si) per 10^7 ions/cm^2 run. The cumulative TID exposure for each device respectively, over all runs, is determined to be well below 50 krad(Si). All devices used in the studies described in this report stay within specification and are fully-functional after the heavy-ion SEE testing is completed with the exception of the single Pre-Probe device.

Confidence Interval Calculations

For conventional products where hundreds of failures are seen during a single exposure, you can determine the average failure rate of parts being tested in a heavy-ion beam as a function of fluence with a high degree of certainty and reasonably tight standard deviation, and thus have a good deal of confidence that the calculated cross section is accurate.

With radiation-hardened parts however, determining the cross section becomes more difficult since often few, or even, no failures are observed during an entire exposure. Determining the cross section using an average failure rate with standard deviation is no longer a viable option, and the common practice of assuming a single error occurred at the conclusion of a null-result can end up in a greatly underestimated cross section.

In cases where observed failures are rare or non-existent, the use of confidence intervals and the chi-squared distribution is indicated. The chi-squared distribution is particularly well-suited for the determination of a reliability level when the failures occur at a constant rate. In the case of SEE testing where the ion events are random in time and position within the irradiation area, one expects a failure rate that is independent of time (presuming that parametric shifts induced by the total ionizing dose do not affect the failure rate), and thus the use of chi-squared statistical techniques is valid (since events are rare an exponential or Poisson distribution is usually used).

In a typical SEE experiment, the device-under-test (DUT) is exposed to a known, fixed fluence (ions/cm^2) while the DUT is monitored for failures. This is analogous to fixed-time reliability testing and, more specifically, time-terminated testing, where the reliability test is terminated after a fixed amount of time whether or not a failure has occurred (in the case of SEE tests fluence is substituted for time and hence it is a fixed fluence test) [3]. Calculating a confidence interval specifically provides a range of values which is likely to contain the parameter of interest (the actual number of failures/fluence). Confidence intervals are constructed at a specific confidence level. For example, a 95% confidence level implies that if a given number of units were sampled numerous times and a confidence interval estimated for each test, the resulting set of confidence intervals would bracket the true population parameter in about 95% of the cases.

In order to estimate the cross section from a null-result (no fails observed for a given fluence) with a confidence interval, start with the standard reliability determination of lower-bound (minimum) mean-time-to-failure for fixed-time testing (an exponential distribution is assumed):

$$MTTF = \frac{2nT}{\chi^2_{2(d+1);100\left(1-\frac{\alpha}{2}\right)}} \quad (1)$$

where

- *MTTF* is the minimum (lower-bound) mean-time-to-failure
- *n* is the number of units tested presuming each unit is tested under identical conditions
- *T* is the test time
- x^2 is the chi-square distribution evaluated at $100(1 - \alpha / 2)$ confidence level
- d is the degrees-of-freedom (the number of failures observed)

With slight modification, invert the inequality and substitute *F* (fluence) in the place of *T*:

$$MFTF = \frac{2nF}{\chi^2_{2(d+1);100\left(1-\frac{\alpha}{2}\right)}}$$

where

- *MFTF* is mean-fluence-to-failure
- *F* is the test fluence
- x^2 is the chi-square distribution evaluated at $100(1 - \alpha / 2)$ confidence
- *d* is the degrees-of-freedom (the number of failures observed) (2)

The inverse relation between MTTF and failure rate is mirrored with the MFTF. Thus the upper-bound cross section is obtained by inverting the MFTF:

$$\sigma = \frac{\chi^2_{2(d+1);100\left(1-\frac{\alpha}{2}\right)}}{2nF} \quad (3)$$

Assume that all tests are terminated at a total fluence of 10^6 ions/cm^2. Also assume that you have a number of devices with very different performances that are tested under identical conditions. Assume a 95% confidence level (σ = 0.05). Note that as *d* increases from 0 events to 100 events, the actual confidence interval becomes smaller, indicating that the range of values of the true value of the population parameter (in this case the cross section) is approaching the mean value + 1 standard deviation. This makes sense when you consider that as more events are observed the statistics are improved such that uncertainty in the actual device performance is reduced.

Table 14. Experimental Example Calculation of Mean-Fluence-to-Failure (MFTF) and σ Using a 95% Confidence Interval (1)

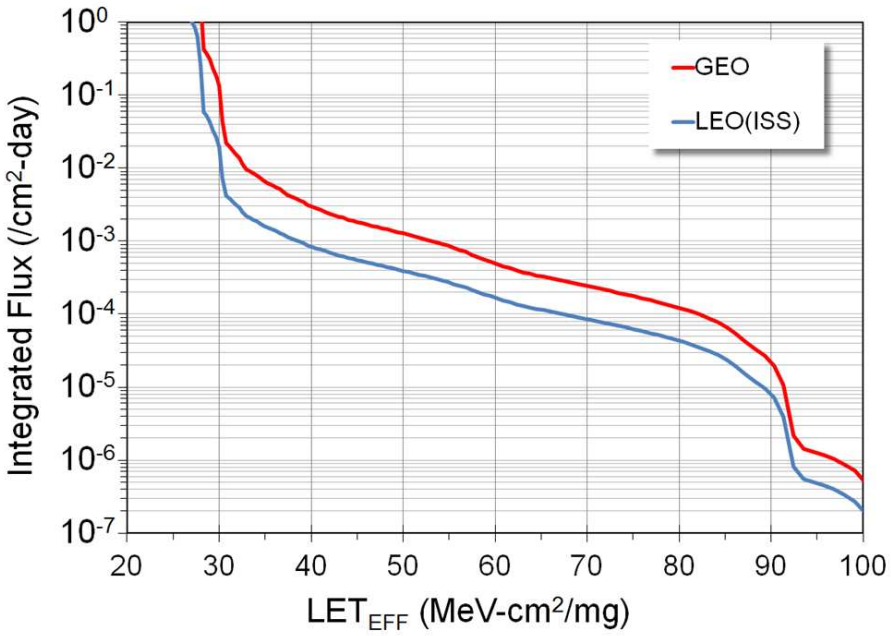
DEGREES-OF-FREEDOM (d)	2(d + 1)	χ^2 @ 95%	CALCULATED CROSS SECTION (cm^2)		
			UPPER-BOUND @ 95% CONFIDENCE	MEAN	AVERAGE + STANDARD DEVIATION
0	2	7.38	3.69×10^{-6}	0.00	0.00
1	4	11.14	5.57×10^{-6}	1.00×10^{-6}	2.00×10^{-6}
2	6	14.45	7.22×10^{-6}	2.00×10^{-6}	3.41×10^{-6}
3	8	17.53	8.77×10^{-6}	3.00×10^{-6}	4.73×10^{-6}
4	10	20.48	1.02×10^{-5}	4.00×10^{-6}	6.00×10^{-6}
5	12	23.34	1.17×10^{-5}	5.00×10^{-6}	7.24×10^{-6}
10	22	36.78	1.84×10^{-5}	1.00×10^{-5}	1.32×10^{-5}
50	102	131.84	6.59×10^{-5}	5.00×10^{-5}	5.71×10^{-5}
100	202	243.25	1.22×10^{-4}	1.00×10^{-4}	1.10×10^{-4}

(1) Using a 95% confidence for several different observed results (d = 0, 1, 2...100 observed events during fixed-fluence tests) assuming 10^6 ion/cm^2 for each test.

Orbital Environment Estimations

In order to calculate on-orbit SEE event rates, you need both the device SEE cross section and the flux of particles encountered in a particular orbit. Device SEE cross sections are usually determined experimentally while flux of particles in orbit is calculated using various codes. For the purpose of generating some event rates, a Low-Earth Orbit (LEO) and a Geostationary-Earth Orbit (GEO) are calculated using CREME96. CREME96 code, short for Cosmic Ray Effects on Micro-Electronics, is a suite of programs [4][5] that enable estimation of the radiation environment in near-Earth orbits. CREME96 is one several tools available in the aerospace industry to provide accurate space environment calculations. Over the years since its introduction, the CREME models have been compared with on-orbit data and demonstrated their accuracy. In particular, CREME96 incorporates realistic “worst-case” solar particle event models, where fluxes can increase by several orders-of-magnitude over short periods of time.

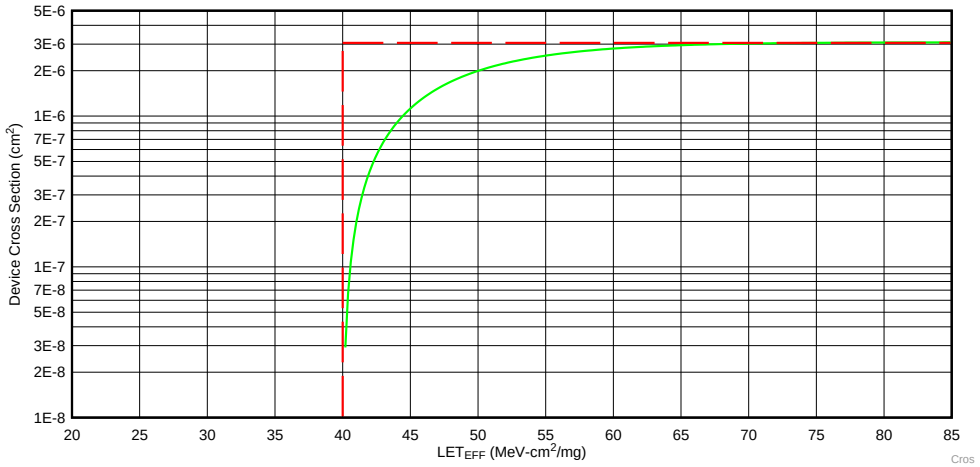
For the purposes of generating conservative event rates, the worst-week model (based on the biggest solar event lasting a week in the last 45 years) was selected, which has been equated to a 99%-confidence level worst-case event [6][7]. The integrated flux includes protons to heavy ions from solar and galactic sources. A minimal shielding configuration is assumed at 100 mils (2.54 mm) of aluminum. Two orbital environments were estimated, that of the International Space Station (ISS), which is LEO, and the GEO environment. Figure 9 shows the integrated flux (from high LET to low) for these two environments.



(1) LEO(ISS) (blue) and a GEO (red) environment as calculated by CREME96, assuming worst-week and 100 mils (2.54 mm) of aluminum shielding.

Figure 9. Integral Particle Flux vs LET_{EFF}

Using this data, you can extract integral particle fluxes for any arbitrary LET of interest. To simplify the calculation of event rates, assume that all cross section curves are square – meaning that below the onset LET the cross section is identically zero while above the onset LET the cross section is uniformly equal to the saturation cross section. Figure 10 illustrates the approximation, with the green curve being the actual Weibull fit to the data with the "square" approximation shown as the red-dashed line. This allows you to calculate event rates with a single multiplication, the event rate simply becoming the product of the integral flux at the onset LET, and the saturation cross section. Obviously, this leads to an overestimation of the event rate since the area under the square approximation is larger than the actual cross section curve – but for the purposes of calculating upper-bound event rate estimates, this modification avoids the need to do the integral over the flux and cross section curves.



(1) Weibull Fit (green) is "simplified" with the use of a square approximation (red dashed line).

Figure 10. Device Cross Section vs LET_{EFF}

To demonstrate how the event rates in this report are calculated, assume that you want to calculate an event rate for a GEO orbit for the device whose cross section is shown in Figure 10. Using the red curve in Figure 9 and the onset LET value obtained from Figure 10 (~40 MeV-cm²/mg), you find the GEO integral flux to be ~2.97 × 10^{-3} ions/cm²-day. The event rate is the product of the integral flux and the saturation cross section in Figure 10 (~3.09 × 10^{-6} cm²):

$$GEO\ Event\ Rate = \left(2.97 \times 10^{-3}\ \frac{ions}{cm^2 \times day}\right) \times (3.09 \times 10^{-6}\ cm^2) = 9.17 \times 10^{-9}\ \frac{events}{day} \quad (4)$$

$$GEO\ Event\ Rate = 3.82 \times 10^{-10}\ \frac{events}{hr} = 0.382\ FIT \quad (5)$$

$$MTBF = 298901\ Years\ ! \quad (6)$$

References

(1) TAMU Radiation Effects Facility website. http://cyclotron.tamu.edu/ref/

(2) “The Stopping and Range of Ions in Matter” (SRIM) software simulation tools website. http://www.srim.org/index.htm#SRIMMENU

(3) D. Kececioglu, “Reliability and Life Testing Handbook”, Vol. 1, PTR Prentice Hall, New Jersey,1993, pp. 186–193.

(4) CREME-MC Technology Page

(5) A. J. Tylka, et al., "CREME96: A Revision of the Cosmic Ray Effects on Micro-Electronics Code", *IEEE Trans. Nucl. Sci.*, Vol. 44(6), 1997, pp. 2150–2160.

(6) A. J. Tylka, W. F. Dietrich, and P. R. Boberg, “Probability distributions of high-energy solar-heavy-ion fluxes from IMP-8: 1973–1996”, *IEEE Trans. Nucl. Sci.*, Vol. 44(6), Dec. 1997, pp. 2140–2149.

(7) A. J. Tylka, J. H. Adams, P. R. Boberg, et al.,“CREME96: A Revision of the Cosmic Ray Effects on Micro-Electronics Code”, *IEEE Trans. Nucl. Sci.*, Vol. 44(6), Dec. 1997, pp. 2150–2160.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265