A Method and Case Study on Identifying Physically Adjacent Multiple-Cell Upsets Using 28-nm, Interleaved and SECDED-Protected Arrays

Michael Wirthlin, *Senior Member, IEEE*, David Lee, *Member, IEEE*, Gary Swift, *Member, IEEE*, and Heather Quinn, *Senior Member, IEEE*

***Abstract*—Extracting information about MCUs from SEU data sets can be a challenge without physical layout information. Many modern static-random access memory (SRAM) components interleave memory cells to improve the robustness of error-correcting codes (ECC) that detect and correct errors in the memory array. Bit interleaving has also become popular with other components with large SRAM arrays, including field-programmable gate arrays (FPGAs). In this paper, we present a technique for extracting MCUs statistically from radiation test data. Further, we use this technique to extract MCU information from a 28-nm FPGA that uses interleaving to protect the configuration memory.**

***Index Terms*—Field programmable gate arrays (FPGAs), multiple-bit upset, reconfiguration, soft errors, single event effect (SEE), testing techniques.**

I. Introduction

As semiconductors continue to scale, single-event effects (SEEs) have an increasing impact on semiconductor circuits [1]. Furthermore, the shrinking feature sizes of transistors has lead to an increase in single-event upsets (SEUs) that affect multiple physically adjacent memory cells [2], which are often called *multiple-cell upsets* (MCUs). MCUs reduce the efficacy of error correcting codes (ECC), such as single-error correct/double-error detect codes (SECDED) that are not designed to correct multiple simultaneous errors. In recent years, interleaving memory cells has reduced the impact of MCUs in static-random access memory (SRAM) arrays by translating an MCU into individual SRAM errors that can be corrected using SECDED [3]. Unfortunately, these interleaving schemes make it more difficult to identify MCUs in radiation testing unless the physical layout is known. In this paper we will discuss a technique for extracting information about MCUs from bit interleaved memory cells when the physical layout is unknown.

Memory cells can be viewed either as being *physically* or *logically* organized. The physical representation of the array is necessary for determining which SEUs are MCUs, as the physical adjacency of multiple SEUs identifies the MCU. The logical organization, such as a word in SRAM arrays or a cache line in a processor, determine how the memory is accessed by the user. Both representations are important when analyzing and understanding the MCU behavior in a component. When ECC is used, it is necessary to determine how MCUs overlay onto the logical structure to determine whether MCUs overcome the encoding scheme. While the physical location and the logical address are related, logical adjacency does not imply physical adjacency, especially if the cells are interleaved [4].

This difference between the physical and logical representations of the memory organization leads to confusion in the terminology used in the literature. In this paper we are using these definitions for the physical representations of errors:

- **Multiple-Cell Upset (MCU)** A single particle causes more than one SEU regardless of the logical relationship. MCUs refer to *all* of the cells that are upset by the particle regardless of logical organization.
- **Single-Cell Upset (SCU)** A single particle that causes only one memory cell to upset.

On the logical realm we use this term:

- **Multiple-Bit Upset (MBU)** An MCU where multiple SEUs occur in a single logical "word." An MBU refers only to the portion of the MCU that occurs within a single logical word and ignores the any part of the MCU that affects other words.

Manuscript received July 21, 2014; revised October 06, 2014; accepted October 30, 2014. Date of publication November 20, 2014; date of current version December 11, 2014. This work was supported by the I/UCRC Program of the National Science Foundation under Grant 0801876 and by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under Contract DE-AC04-94AL85000 . This work has been authored by an employee ofNo. Los Alamos National Security, LLC, operator of the Los Alamos National Laboratory under Contract DE-AC52-06NA25396 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting this work for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce this work, or allow others to do so for United States Government purposes. Los Alamos National Laboratory strongly supports academic freedom and a researchers right to publish; however, the Laboratory as an institution does not endorse the viewpoint of a publication or guarantee its technical correctness.

M. Wirthlin is with the NSF Center for High-Performance Reconfigurable Computing (CHREC), Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602 USA (e-mail: wirthlin@ee.byu.edu).

D. Lee is with Sandia National Laboratories, Albuquerque, NM 87185-0986 USA.

G. Swift is with Swift Engineering and Radiation Services, San Jose, CA 95154 USA.

H. Quinn is with Los Alamos National Laboratory, Los Alamos, NM, 87545 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2014.2366913

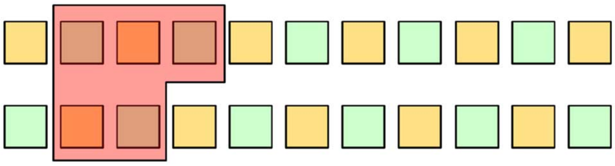


Fig. 1. 5-Bit Multi-Cell Upset within Interleaved Frames.

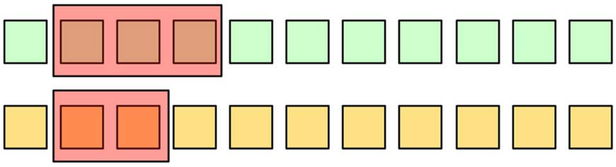


Fig. 2. Two MBUs Visible to the User.

As an artifact of testing, it is possible to construct MCUs through the accumulation of SEUs [5]. For this situation we use this term:

- **Coincident SEU (CSEU)** Two or more SEUs, whether MCU or SCU, that are physically adjacent in such a manner that an MCU is constructed (i.e., "fake MCU"). Because this effect is an artifact of accelerated testing, such events are statistically rare in deployed systems.

We are particularly interested in how MCUs affect SRAM-based FPGAs. These components share many of the same problems as SRAM memory, as the architecture leverages SRAM memory to store the internal state that is used for specifying the behavior of programmable logic, internal block memory, user flip-flops, and control circuitry (internal state machines, and system registers). The configuration memory on a modern FPGA might include millions of SRAM cells that are often sensitive to SEUs and MCUs [5]–[7]. Xilinx has started using bit interleaving and a 32-bit SECDED ECC word in the 7-series FPGAs to help correct and detect SEUs as they occur. These 32-bit ECC words are used to detect and correct SCUs in the logical "word" in an FPGA, which is called a *frame*. Fig. 1 demonstrates how a 5-bit MCU would affect the contents of two interleaved frames. This same MCU corresponds to two MBUs as seen in Fig. 2. The two MBUs are easily detected by the user through configuration readback. The actual MCU, however, is not detectable by the user.

SRAM and FPGA components share another similarity–the physical layout and logical organization is proprietary. With physical layout information, there a number of different MCU extraction techniques and analyses that can be completed, such as the effect of well depth, well contacts, cluster size or input data dependence [8]–[10]. Other researchers have looked at methods for extracting MCUs from SEU data sets when the physical layout was not known. [11] uses an analytically model based on the geometric distribution that modeled "grouped arrivals" as a proxy for MCU effects and was validated using the data from [3]. Several researchers has used logical adjacency to extract the MCUs from the SEU data [12]–[14]. In [15] MCUs were extracted from the SEU data set for the Spartan-3 by running at very low fluences, so that the statistical probability of having more than one upset in the configuration memory at one time was very low. Previously, the authors have relied upon reverse engineering or proprietary information to translate logical addresses to physical addresses in Xilinx FPGAs [5], [7]. Our technique is a statistical method that can be used with a minimum of information about physical layout of the component and any SRAM array. This technique uses the logical addresses of SEUs to determine the probability of physical adjacency and statistically define the physical adjacency model, which can be used to extract MCUs from the original data set.

This paper presents a technique for extracting MCU information for FPGAs from radiation test data using a statistical method (Section II). A discussion of uncertainty and validation are in Section III. The MCU extraction technique is then applied to the radiation test data of the Xilinx Kintex-7 (Section IV). We compare these results to results collected on previous FPGAs. While broadly applied to the Kintex-7, we believe this technique will be generally useful to independent researchers studying SRAM and FPGA components that do not have the benefit of the physical layout information.

II. MCU and MBU Extraction Technique

The MCU extraction technique uses radiation test data and the dimensions of an array (SRAM or FPGA) to determine statistically which SEUs in the test data are MCUs. This process involves these steps:

1. Collect SEU data with radiation testing and organize upset data into logical addresses,
2. Determine the logical distances between SEUs and create a histogram of common upset pairs,
3. Create physical adjacency model from statistical data, and
4. Extract MCUs and MBUs from the SEUs using physical adjacency model.

These steps will discussed in detail in this section.

A. Collect SEU Data

The first step in this process is to collect SEU data from static radiation testing. In many cases the SEU data collected for the SEU cross section can be used for MCU analysis. However, it is important that the number of SEUs in each read of the component are kept low so that coincident SEUs (CSEUs) do not contaminate the data, which is discussed in detail in Section III. Collecting SEU data for MCU analysis requires either many short beam runs for static test methodologies or using a semi-dynamic/dynamic test methodologies that allows for frequent full component read outs.

To perform physical adjacency analysis, the individual upset data must be represented in some two dimensional form. For example, in an FPGA the x-dimension is defined by the number of configuration frames and the y-dimension is defined by the number of bits in the frame. An SRAM could use the number of words for the x-dimension and the number of bits in the word as the y-dimension. This coordinate system does not necessarily represent any physical organization and is used primarily for bounding the locations of the SEUs. It is not necessary for the 2D array to be square, as this array does not need to have a one-to-one correlation with the physical layout of the component.

For this work, an upset (u_i), is represented as a (x_i, y_i) tuple, where x corresponds to the frame number location of the upset and y corresponds to the bit number of the upset within the

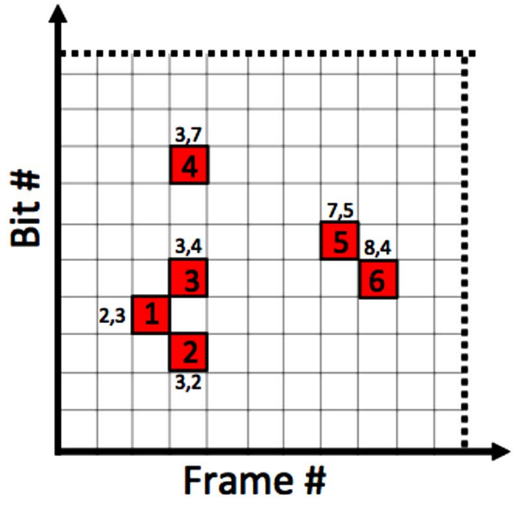


Fig. 3. FPGA Upset Coordinate System and Upset Labeling.

frame. A similar process is possible using the number of words and the word size for traditional SRAM arrays. Fig. 3 demonstrates six different upsets labeled using this coordinate system. Upset u_1, denoted by the box with the number '1', indicates an upset in frame #2 ($x_1 = 2$) and bit #3 within this frame ($y_1 = 3$).

B. Identifying Upset Pair Offsets

Once the SEUs from a test have been converted into (x, y) coordinates in the 2D array, it is possible to determine the *upset pair offsets* and identify common upset patterns. Patterns are identified by comparing *each* SEU location to all other SEU locations in the list. During this process for a list of N SEUs, $N(N-1)$ distinct *upset pairs* are evaluated. The example in Fig. 3 contains 15 unique upset pairs that must be evaluated. An upset pair, $UP_{i,j}$, is represented as an ordered set of two upsets, (u_i, u_j).

To identify common coordinate offsets between upset pairs, the upset pair offset, $UPO_{i,j}$, is computed for each upset pair:

$$UPO_{i,j} = (\Delta x_{i,j}, \Delta y_{i,j}) = (x_j - x_i, y_j - y_i), \qquad (1)$$

for all upset pairs where $i \neq j$. For example, the upset pair offset for u_1 and u_2 in Fig. 3 is $UPO_{1,2} = (2-1, 2-3) = (1, -1)$. To ensure that there is only a single upset pair offset for each pair of upsets, the upsets are ordered using the following convention: if $x_i > x_j$, then $u_i > u_j$. If $x_i = x_j$ and $y_i > y_j$, then $u_i > u_j$. Upset pairs, $UPO_{i,j}$, are created such that $u_i < u_j$.

Computing all of the upset pair offsets for large 2D arrays with a large number of SEUs is not practical. The maximum number of upset pair offsets for a given 2D array is defined as

$$(|x| - 1) \times (2(|y| - 1) + 1), \qquad (2)$$

where $|x|$ and $|y|$ are the dimensions of the 2D array. For the example of Fig. 3 where $|x| = 10$ and $|y| = 10$, the number of unique upset pair offsets is 171. For the Kintex-7 325T, where $|x| = 22,546$ and $|y| = 3,232$, the maximum number of unique upset pair offsets is 147,708,335, making it impractical to calculate every upset pair offset.

To simplify the computation of upset pair offsets, only a small subset of upset pair offsets is considered. Based on previous FPGA test results, we assume that physical adjacency is most likely with configuration bits with relatively close frame numbers and bit numbers. The upset offset values that will be searched are limited by a frame distance and bit off distance of 32 as follows: $0 \leq \Delta x \leq 31$ and $-31 \leq \Delta y \leq 31$. This restriction limits the upset pair offset search space to 1,891 upset pairs.

Once all of the pair upsets are determined from the SEU test data, a histogram of all the offset patterns is created. In the example of Fig. 3, there is one upset pair offset with more than one count: $(\Delta x = 1, \Delta y = -1) = 2$. This upset pair offset $(\Delta x = 1, \Delta y = -1)$ is seen by the following upset pairs: (u_1, u_2) and (u_5, u_6). These upset pair offsets suggest that they represent physical adjacency.

C. Create Physical Adjacency Model

After collecting the individual bit upset data and tabulating the histogram of upset offset counts, the upset pair offsets will be analyzed and used to build a physical adjacency model. Specifically, a physical adjacency model is created by selecting specific upset offsets and tagging such offsets as "physically adjacent". If a particular upset pair offset represents physical adjacency and MCUs occur with this upset offset, this particular offset will appear in the upset offset histogram more frequently that offsets that do not have physical adjacency. For example, if the upset offset $(\Delta x, \Delta y) = (0, 1)$ corresponds to physical adjacency and MCUs occur between SEUs with this offset then this offset will appear in the upset log with a far higher frequency than upset offsets that do not correspond to physical adjacency.

If no MCUs occur in a radiation test and the upsets occur uniformly over the array, then each upset offset should appear within the radiation test data at constant rate. If a specific upset offset corresponds to physical adjacency then the upset offset rate will be much higher than this constant upset rate. Those upset offsets that demonstrate this higher rate are chosen as "physically adjacent" offsets. Any configuration upset pair that matches this particular upset offset will be identified as an MCU.

D. MCU and MBU Extraction

After the physical adjacency model has been identified, discrete MCUs can be created by comparing all pairs of upsets within individual runs of the radiation test data. Any upset pairs observed in the radiation test data that matches one of the chosen "adjacent" pairs are assumed to be caused by the same particle and are combined to form a single 2-bit MCU. For example, if an upset pair offset of $(0, 1)$ has been tagged as "physically adjacent", any upset pair that matches this upset offset will be combined into a 2-bit MCU.

The process of creating MCUs from individual SCUs continues iteratively to build larger and larger MCUs to create maximally sized clusters. Initially, the offset of all upset pairs are compared to create 2-bit MCUs from the full set of discrete configuration upsets. Next, all 2-bit MCUs are compared against other SCUs to see if they are physically adjacent. If these MCUs are bridged by an upset pair, they are combined to form a larger MCU. This process continues until no MCUs contain upsets that are physically adjacent to any individual upset in the test data. This algorithm similar to the algorithm described in [5] to group adjacent upsets into larger MCUs.

To illustrate this process, the upset map of Fig. 3 will be used with the chosen adjacency coordinates to demonstrate how MCUs are clustered. First u_1 is compared to u_2 and a upset pair offset match is found (1,-1). These two upsets are grouped into a single MCU. Next, u_1 is compared to u_3 and again an upset pair offset match is found (1,1). This upset is added to the MCU. When u_1 is then compared to u_4, u_5, u_6, no adjacencies are found. This process continues by comparing each upset in sorted order to all remaining upsets to iteratively form larger MCUs. Three distinct upset events will be identified with the physical adjacency model described above: (u_1, u_2, u_3), (u_4), and (u_5, u_6).

The extracted MCU data can also be used to identify independent MBUs. An MBU is extracted from the MCU data when more than one upset from the same MCU is in the same frame. For example, one MBU will be extracted from the MCU events of the example in Fig. 3. Three upsets are seen in frame 3 of this upset map (u_2, u_3, and u_4). Two of the upsets belong to the same MCU u_3 and u_2. Because upsets u_2 and u_3 belong to the MCU, they are identified as an MBU in frame 3. Because upset u_4 is not associated with an MCU, it is classified as an independent SCU.

III. Validation and Quantifying Uncertainty

Accurately extracting MCUs from existing SEU data can be a challenge. There can be a number of issues in which uncertainty can be inserted into the process. For our technique there are two sources of uncertainty:

- CSEUs in the data set and
- Lack of physical layout information.

The first source of uncertainty needs to be addressed through experimental test methodologies. The second source of uncertainty is quantified by comparing results to a known data set. We will discuss both of these issues in this section.

A. CSEUs

CSEUs are caused by allowing too many SEUs to accumulate in the SRAM array or FPGA before reading out the results. As the number of SEUs increases, the probability of constructing an MBU from two existing SEUs increases. Estimating the probability of a CSEU is an important part of the test design process, as experimenters might need to limit either the exposure time or the flux to keep the SEU rate below a certain level for MCU extraction. In [16], the authors discuss methods for estimating CSEUs by analyzing the shape of MBUs, analytically and through Monte Carlo simulations, which lead to the use of Monte Carlo simulations in [5], [7] to estimate the probability of CSEUs. In [17], the author addresses the problem statistically using an extension of the well-known birthday problem. In this paper, Monte Carlo experiments and the equations from [17] provide bounds on the bias from CSEUs.

The Monte Carlo experiment was designed to be parameterized with variables for the shape of the 2D array, the number of SEUs per trial and the number of trials. For our experiment, we chose the shape of the Kintex-7 2D array (22,546 frames, 3,232 bits). A variety of values were chosen for the number of SEUs per trial to determine how the CSEU probability changed as more of the 2D array was upset. The Monte Carlo experiments were all run for 1,600,000 trials for statistical significance. From [17] we used equation 17:

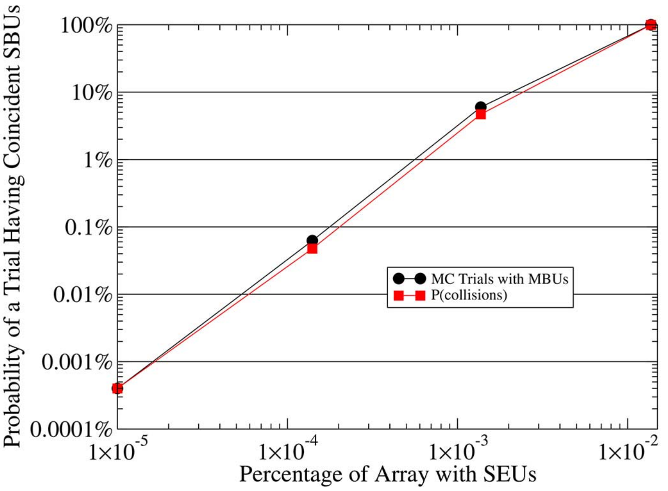


Fig. 4. Probability of a CSEU as a function of the percentage of the Array that has SCUs.

$$P_k(n,p)(Collision) \approx 1 - e^{\frac{-p(p-1)(2k-1)}{2n}}, \tag{3}$$

where n is the number of bits in the 2D array, k is the collision range (set to 31) and p is the number of upsets. Both of these methods were used to determine the probability that a given trial has CSEUs. The Monte Carlo experiments are also used to determine characteristics about the CSEUs, such as the expected number of CSEUs in a trial.

The probabilities calculated by the Monte Carlo experiments and [17] are shown in Fig. 4. Not only do these techniques show a good correlation to each other, the output shows the effect of accumulating SEUs. This figure shows that when SEUs comprise 0.00137%–0.01372% of the array the probability of having a trial with a CSEU goes from 5% to 99%. Fig. 5 shows the histogram of CSEUs per trial. These figures show that when there are very few SEUs in the array, the probability of a CSEU is low and the probability that a trial has more than one CSEU in a trial is very low. As SEUs accumulate in the array, the probability of CSEUs increases rapidly and non-linearly. For this experiment, this problem is negligible due to low limits on the number of upsets per trial.

B. Validation with Physical Layout

While this paper discusses the results of Kintex-7 radiation testing, the authors have previously studied other FPGAs. We were able to use one of our historical data sets of the Virtex-5 to validate the efficacy of the technique. The MCU results for the Virtex-5 were published in [7]. The MCUs in this report were extracted using a tool that was designed using proprietary information from Xilinx. We used the proton data set from [7] to validate the technique in this paper. We present the comparison between these two tools in Table I. These results show that our technique are accurate to within 2%–28% of results using

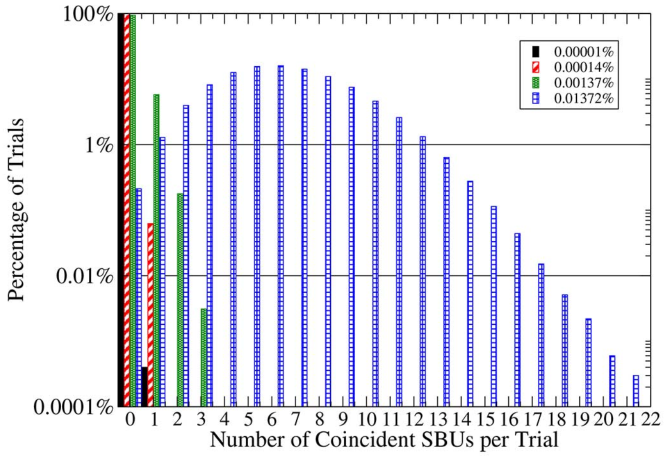


Fig. 5. Histogram of CSEUs per Trial.

TABLE I
COMPARISON OF VALIDATION

Energy (MeV)	MCU Extraction	Propietary Tool	Difference
200	10.43%	10.17%	1.02
200	9.64%	10.30%	0.93
65	7.43%	5.77%	1.28

the physical layout. Furthermore, the technique properly recognized the component did not use bit interleaving.

IV. XILINX KINTEX-7 28 NM FPGA CASE STUDY

Now that the technique and its limitations have been described, we will apply it to a radiation data set collected on the Kintex-7 325T. In this section we will describe how these tests were setup, the results from MCU extraction and historical trends with other Xilinx FPGAs.

A. Radiation Test Setup

Radiation testing was performed using the Xilinx KC705 evaluation board containing the 28-nm Xilinx Kintex-7 325T FPGA (see Fig. 6). This FPGA has 22,546 frames and each frame has 3,232 bits. Testing was performed using the 16 MeV heavy ion cocktail at Lawrence Berkeley National Laboratory in September 2014. The ions, fluences and total number of upsets are shown in Table II. The component was tested at normal incidence, nominal voltages and nominal temperatures. The configuration memory operates on the "VCCINT" power supply with a nominal voltage of 1.0 V. A complete set of cross-section results for the configuration memory, internal Block memory (BRAM), and the user Flip-Flops can be found in [6]. In this paper, we focus solely on the configuration memory for MCU analysis.

To limit the presence of CSEUs, the test for each ion was broken up into many small "runs." For these tests the FPGA was configured before the test started and continuous readbacks were done during the tests. These readback operations create readback files which include the contents of the configuration memory, including SEUs. The FPGA was not configured while the beam was on to reduce the chance of overwriting SEUs. The number of runs and the average number of upsets per run are listed in Table II. The flux was chosen to provide under 500 upsets per run. Any runs that contained more than 1000 upsets were removed from the analysis.

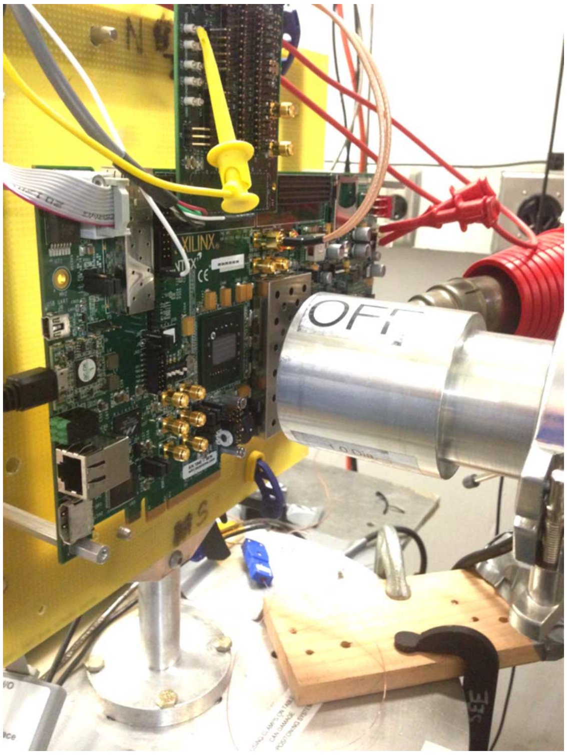


Fig. 6. Radiation Test Setup Using the Xilinx KC705 Evaluation Board.

TABLE II
HEAVY ION BEAM PARAMETERS. THE INCIDENT LET IS THE LET AT THE SURFACE OF THE ACTIVE VOLUME

Ion	Incident LET $(\frac{MeV-cm^2}{mg})$	Fluence (Particles)	Total Upsets	Runs	Average Upsets per Run
N	1.16	1.28×10^7	36,047	81	445
O	1.54	4.06×10^6	42,017	98	428
Ne	2.39	5.93×10^5	12,918	71	182
Si	4.35	5.89×10^5	22,379	102	219
Ar	7.27	3.52×10^5	16,013	38	421
Cu	16.5	2.14×10^5	18,878	43	439
Kr	25.0	2.50×10^5	23,605	240	93
Xe	49.3	2.08×10^5	66,293	132	502

The FPGA was configured with a known *bitstream* implementing a simple user circuit. After the test was over, the readback files were compared to the original bitstream to identify the upset locations. These sets of upset locations are then used to create the physical adjacency model.

B. Create Physical Adjacency Model

The individual upsets collected during radiation testing were used to create a physical adjacency model of the Kintex-7 configuration memory. Using the procedure described in Section II, all upset pairs for each individual run were analyzed

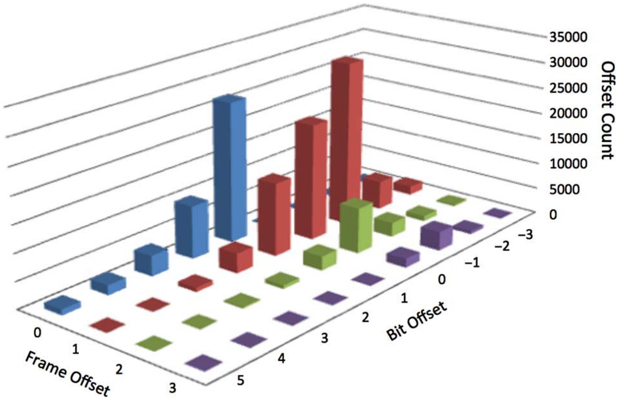


Fig. 7. Occurrence of Upset Offset Pairs.

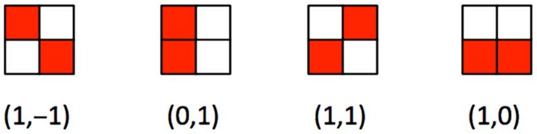


Fig. 8. Highest Probability Upset Pairs (frame offset).

and counted to identify high probability upset offset pairs. To limit the search space, pairs are upsets were only considered with a frame distance of 32 or less (i.e. $\Delta x \leq 32$) and a bit number offset of 32 or less (i.e., $\Delta y \leq 32$). The count of all upset pairs in the complete data set is summarized in Fig. 7.

A wide variety of upset offset pairs are seen in the data set. However, several upset offset pairs were seen with far more frequency than others. The most common upset pair observed was (1,-1), or a pair of configuration bits in adjacent frames (i.e., $\Delta x = 1$) and where the bit offset of the second upset is one less than the bit offset of the first upset (i.e., $\Delta y = -1$). This clearly suggests that some form of frame interleaving is being performed–configuration bits in sequential frames are physically adjacent to maximize the benefits of the error correction coding. Other upset offset pairs that occurred with a high frequency include (0,1), (1,1), and (1,0). All four upset patterns occur with a much higher frequency than other upset patterns and are selected as “physically adjacent”. These four patterns are summarized graphically in Fig. 8 and are used to extract MCUs from the upset data.

C. Extracting Kintex-7 MCUs

The physical adjacency model described above was used to extract MCUs from the upset data collected in the radiation tests. The approach described in Section II was applied to the upset list from each readback file. The result of this process was to group individual upsets into MCUs using the adjacency model and to identify the remaining upsets as SCUs. The majority of events were SCUs but a number of MCUs were found. Table III summarizes the top nine MCU shapes that were extracted during this process (including the single-bit SCU “shape” of one). This table lists the percentage of shapes extracted for each ion used during radiation testing. The total number of shapes extracted is summarized in the last row.

The sizes of extracted MCU events is plotted in Fig. 9 as a function of LET and MCU size. At low LETs, only a small percentage of events are MCUs. As the LET increases, the percentage of SCUs decrease and the MCUs significantly increases.

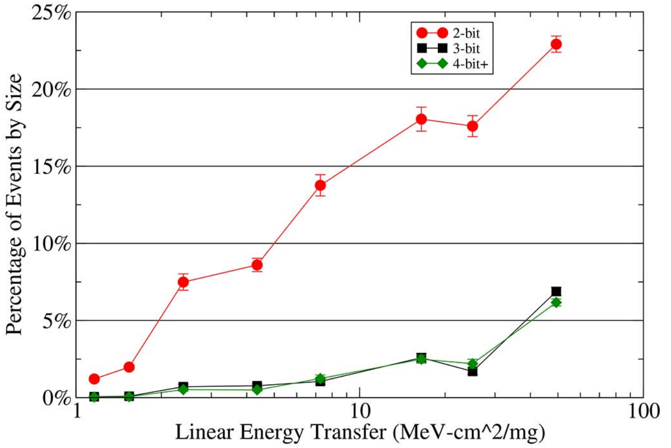
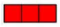
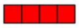
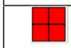
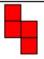


Fig. 9. MCU sizes as a percentage of the total observed MCUs as a function of LET.

TABLE III
PERCENTAGE OF MCU SHAPES EXTRACTED FOR EACH ION

	Ion							
Shape	N	O	Ne	Si	Ar	Cu	Kr	Xe
	98.7	97.9	93.1	90.1	84.0	76.9	78.5	64.0
	0.2	0.3	0.9	1.4	2.5	6.4	5.0	15.1
	0.7	1.1	4.5	5.2	7.6	6.5	7.6	4.4
	0.1	0.2	0.9	1.0	2.6	4.1	3.9	2.7
	0	0.1	0.6	0.6	0.7	2.0	1.2	4.1
	0.2	0.4	1.2	1.1	1.1	1.0	1.0	0.8
	0	0	0.1	0.1	0.4	0.5	0.7	1.8
	0	0	0	0	0.2	0.5	0.4	0.6
	0	0	0	0.1	0.1	0.2	0.2	0.6
Total	35517	41032	11657	19952	13275	14122	18144	41230

With only 100-500 upsets per run the contamination from CSEUs is quite low. The expected number of CSEUs erroneously counted as MCUs for this data is between 0.0006-0.062. In comparison to the experimental error, which is calculated using 95% Poisson confidence intervals, the contribution to the error from CSEUs is negligible.

D. MBU Data Analysis

As described earlier, MBU events for FPGAs correspond to an SEU event causing more than one cell to upset within a configuration frame. MBUs refer only to the upsets of an MCU event that occur in a single frame. Under normal circumstances, MBU events are relatively easy to identify by the FPGA user. These events are identified by performing a configuration readback on a frame and comparing all bits of the frame against the corresponding golden configuration frame. If the number of bit

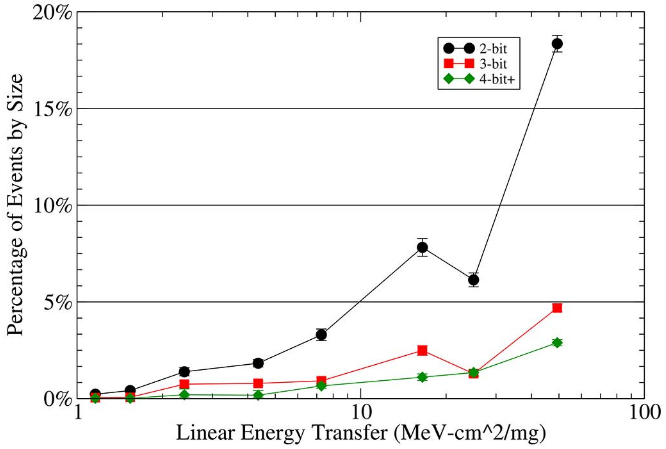


Fig. 10. MBU sizes as a percentage of the total observed MBUs as a function of LET.

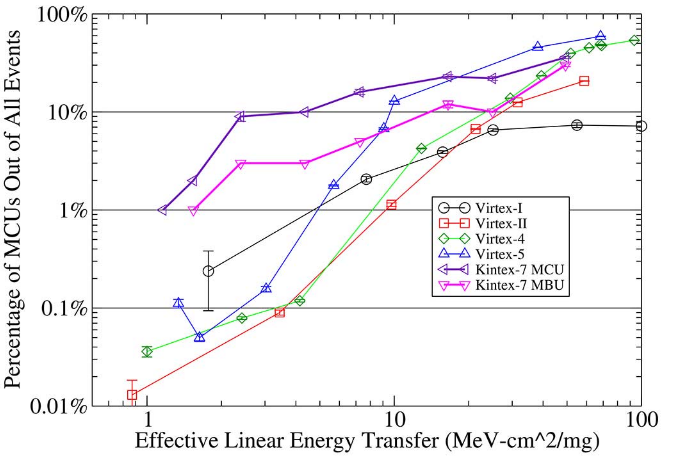


Fig. 11. MCU Events as a Percentage of SEUs for Five Xilinx FPGA Families.

differences in the frame is greater than one, the event is classified as an MBU.

The technique described in Section II for extracting MBUs from MCUs was used on the Kintex-7 MCU data set. The sizes of extracted MBU events is plotted in Fig. 10 as a function of LET and size. As with MCUs, at low LETs most of the events are SCUs (99.23% at 1.5 $\frac{MeV-cm^2}{mg}$). As the LET increases, the percentage of MCUs and MBUs increase but the percentage of MCUs is always higher than MBUs. At the highest tested LET (60 $\frac{MeV-cm^2}{mg}$), 38.1% of the events are classified as an MCU while only 29.5% of the events are classified as MBUs. This result suggests that interleaving is used to improve the SECDED memory protection of individual frames. Because MBUs cause the internal scan circuitry to stall until there is an external repair of the bitstream, interleaving decreases the need for external intervention particularly in terrestrial environments.

It is important to note that there are fewer MBU events than MCU events–the MCU events represent events that often span multiple logical frames. When viewed as MBUs, these events are split up into multiple distinct events each in a logical frame. These results suggest that interleaving cells between frames is effective and that the internal SECDED coding scheme can repair more MCU events by breaking such events into multiple MBU events.

E. MCUs and Technology Scaling

It is interesting to compare the MCU and MBU behavior of the 28-nm Kintex-7 against older Xilinx FPGA families. Fig. 11 compares the percentage of events that cause MCUs (more than one bit upset) for several different families. The MCU data for the Virtex-5 (65-nm, UMC), Virtex-4 (90-nm, UMC), Virtex-II (150-nm, UMC), and Virtex (180-nm, UMC) technology nodes was obtained from [18]. For all families, the percentage of MCUs increases with higher LET values [5], [7].

The dotted line represents the 28-nm Kintex-7 MCUs as a percentage of observed events. As seen in this graph, the 28-nm FPGA is more sensitive to MCUs at low LET. As the energy increases, the MCU rate tracks the older Virtex-II series FPGA suggesting that the manufacturer invested additional effort to protect the configuration cells from multiple-cell upsets in this smaller process geometry. The dashed line represents 28-nm Kintex-7 MBUs and highlights the advantage of configuration memory interleaving. By interleaving the configuration memory, the effective MBU rate is lower than all but the Virtex-I FPGA family.

V. Conclusion

A method was introduced for identifying the physical adjacency of dense memory arrays from radiation test data that allows experimenters to understand both the MBU and MCU effects of SRAM components. In this paper, we illustrated the steps necessary for extracting MCUs and MBUs: translating logical addresses to a coordinate system, determining the upset pair offsets, modeling the physical adjacency through the offsets, and extracting MCUs/MBUs. This technique was applied to a Xilinx 7-Series, 28-nm FPGA. These results show that bit interleaving has reduced the impact of MCUs on the 7-Series FPGA by distributing the upsets across frame boundaries and relying on error correction codes for memory protection.

References

[1] F. Wrobel, J.-M. Palau, M.-C. Calvet, O. Bersillon, and H. Duarte, “Simulation of nucleon-induced nuclear reactions in a simplified SRAM structure: Scaling effects on SEU and MBU cross sections,” *IEEE Trans. Nucl. Sci.*, vol. 48, no. 6, pp. 1946–1952, Dec. 2001.

[2] F. Ruckerbauer and G. Georgakos, “Soft error rates in 65 nm SRAMs–analysis of new phenomena,” in *Proc. 13th IEEE Int. On-Line Testing Symp. (IOLTS ’07)*, Jul. 2007, pp. 203–204.

[3] S. Baeg, S. Wen, and R. Wong, “SRAM interleaving distance selection with a soft error failure model,” *IEEE Trans. Nucl. Sci.*, vol. 56, no. 4, pp. 2111–2118, Aug. 2009.

[4] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, “Investigation of multi-bit upsets in a 150 nm technology SRAM device,” *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2433–2437, Dec. 2005.

[5] H. Quinn, P. Graham, J. Krone, M. Caffrey, and S. Rezgui, “Radiation-induced multi-bit upsets in SRAM-based FPGAs,” *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2455–2461, Dec. 2005.

[6] D. Lee, M. Wirthlin, G. Swift, and A. Le, “Single-event characterization of the 28 nm Xilinx Kintex-7 field-programmable gate array under heavy-ion irradiation,” in *IEEE Radiation Effects Data Workshop (REDW)*, Dec. 2014, to be published.

[7] H. Quinn, K. Morgan, P. Graham, J. Krone, and M. Caffrey, “Static proton and heavy ion testing of the Xilinx Virtex-5 device,” in *IEEE Radiation Effects Data Workshop*, Jul. 2007, pp. 177–184.

[8] N. Mahatme, B. Bhuva, Y.-P. Fang, and A. Oates, “Analysis of multiple cell upsets due to neutrons in SRAMs for a deep-n-well process,” in *Proc. IEEE Int. Reliab. Phys. Symp. (IRPS)*, Apr. 2011, pp. SE.7.1–SE.7.6.

[9] H. Fuketa, R. Harada, M. Hashimoto, and T. Onoye, “Measurement and analysis of alpha-particle-induced soft errors and multiple-cell upsets in 10T subthreshold SRAM,” *IEEE Trans. Device Mater. Reliab.*, vol. 14, no. 1, Mar. 2014, dOI:10.1109/TDMR.2013.2252430.

[10] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, “Investigation of multi-bit upsets in a 150 nm technology SRAM device,” *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2433–2437, Dec. 2005.

[11] S. Baeg, P. Reviriego, J. Maestro, S. Wen, and R. Wong, “Analysis of a multiple cell upset failure model for memories,” in *Proc. IEEE Workshop Silicon Errors in Logic-Syst. Effects*, Oct. 2009,2014 [Online]. Available: http://rsc.hanyang.ac.kr/papers/int_conf/11.pdf

[12] C. Underwood, R. Ecoffet, S. Duzeffier, and D. Faguere, “Observations of single-event upset and multiple-bit upset in non-hardened high-density SRAMs in the TOPEX/Poseidon orbit,” in *Proc. IEEE Radiation Effects Data Workshop*, Jul. 1993, pp. 85–92.

[13] K. Grurmann, D. Walter, M. Herrmann, F. Gliem, H. Kettunen, and V. Ferlet-Cavrois, “SEU and MBU angular dependence of samsung and micron 8-Gbit SLC NAND-Flash memories under heavy-ion irradiation,” in *Proc. IEEE Radiation Effects Data Workshop (REDW)*, Jul. 2011, dOI:10.1109/REDW.2010.6062521.

[14] K. Grurmann, M. Herrmann, F. Gliem, H. Schmidt, G. Leibeling, H. Kettunen, and V. Ferlet-Cavrois, “Heavy ion sensitivity of 16/32-Gbit NAND-Flash and 4-Gbit DDR3 SDRAM,” in *Proc. IEEE Radiation Effects Data Workshop (REDW)*, Jul. 2012, dOI:10.1109/REDW.2012.6353718.

[15] A. Manuzzato, S. Gerardin, A. Paccagnella, L. Sterpone, and M. Violante, “On the static cross section of SRAM-based FPGAs,” in *Proc. IEEE Radiation Effects Data Workshop*, Jul. 2008, pp. 94–97, dOI:10.1109/REDW.2008.24.

[16] H. Quinn, P. Graham, M. Wirthlin, B. Pratt, K. Morgan, M. Caffrey, and J. Krone, “A test methodology for determining space readiness of Xilinx SRAM-based FPGA devices and designs,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 10, pp. 3380–3395, Oct. 2009.

[17] H. Tausch, “Simplified birthday statistics and Hamming EDAC,” *IEEE Trans. Nucl. Sci.*, vol. 56, no. 2, pp. 474–478, Apr. 2009.

[18] H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey, and K. Lundgreen, “Domain crossing errors: Limitations on single device triple-modular redundancy circuits in Xilinx FPGAs,” *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, pp. 2037–2043, Dec. 2007.