Bit-Upset Vulnerability Factor for eDRAM Last Level Cache Immunity Analysis

Navid Khoshavi, Xunchao Chen, Jun Wang and Ronald F. DeMara
College of Engineering and Computer Science
University of Central Florida, Orlando, FL
{nkhoshavi, xchen, jwang, demara} @eecs.ucf.edu

***Abstract*—Whereas contemporary Last Level Cache (LLC) designs occupy a significant fraction of total die area in chip-multiprocessors (CMPs), approaches to deal with the vulnerability increase of LLC against Single Event Upset (SEU) and Multi-Bit Upsets (MBUs) are sought. In this paper, we focus on reliability assessment of eDRAM LLC to propose a more accurate and application-relevant vulnerability estimation approach compared to conventional LLC SEU analysis methods. In particular, the eDRAM Bit Upset Vulnerability Factor (BUVF) is proposed and an algorithm is developed to assess its behavior for soft errors using experimental benchmark suites. BUVF explicitly targets the vulnerable portion of the eDRAM refresh cycle where the critical charge varies depending on write voltage, storage and bit-line capacitance. Results for the PARSEC benchmark suite indicated that vulnerable sequences account for about 27.2% of data array lifetime in the cache, among which the Read-Read (RR) access sequence contributes about 23.4%. Furthermore, regardless of the size of the vulnerable data set located in an RR sequence over a short interval, the corresponding region of cache is seen to contribute negligible vulnerability to BUVF, which results from spending a small fraction of program execution time undergoing RR sequences. We recast the problem of reliable eDRAM LLC design as a straightforward search for reduced BUVF.**

I. Introduction

Soft errors can be induced by impacts of energetic particles that penetrate the silicon substrate and generate electron-hole pairs along their tracks. Whenever a the critical charge (Q_C) collects in the cell junction, then it can flip the cell state and generate a soft error in the memory cell [1]. Even though the Soft Error Rate (SER) per bit is small, the large density of Last Level Cache (LLC) increases the SER per chip. In particular, given roughly 50% of chip is occupied by cache memory structure, high-dense large-scale LLC, becomes highly susceptible to soft errors [2]. Moreover, the high potential of residing a data block in LLC for millions of cycles between two consecutive accesses has significantly increased the vulnerability of LLC cache blocks to soft errors [3].

Recently, large embedded Dynamic Random Access Memory (eDRAM) cache has been introduced as the LLC or L3 cache between L2 SRAM cache and main memory to further alleviate the core-memory speed gap [3] [4]. The employment of on-chip L3 eDRAM offers higher cache capacity compared to SRAM [3] [5] [6] and provides faster on-chip communication through on-chip high bandwidth [7] [8]. The eDRAM requires periodic refresh operation less than $retention\ time$ to prevent data loss due to the leakage of capacitors over time. The eDRAM cell's retention time is defined as the interval between refresh operations at which an eDRAM cell can retain a reliable bit state [9] [10]. As the feature size continues to decrease, the eDRAM cell size reduction imposes challenges to eDRAM design by decreasing the retention time of eDRAM cell, but also leads to reduced reliability in terms of increasing the susceptibility of eDRAM cell to the soft errors [11] [12] [13] [14].

The charge in each a bit cell is refreshed to preserve data integrity. This means the data stored in eDRAM is required to be read out and written back into cell by each refresh operation. Previously, analysis was performed to consider the write behavior of the cache lines as a mechanism to modulate the refresh rate relative to the soft error rate [15]. This new lifetime sequence needs to be investigated to accurately reveal how different lifetime sequences of cache data contribute to vulnerability. Additionally, in the case of eDRAM, the refresh mechanism can also restrict the feasibility of accessing the bit cell. Given these challenges, we propose a more accurate and application-relevant lifetime model while taking the refresh operation scheme for LLC into consideration. One successful approach that has been utilized with other memory technologies such as SRAM, is classifying the cache line accesses throughout the residency lifetime [16].

Herein, a lifetime model categorizes the lifetime sequences to six classes for each data item located in eDRAM LLC. Each class is designated into one of two groups, $vulnerable$ and $nonvulnerable$ sequences [16]. A vulnerable sequence is characterized by the fact that any error that is encountered during this sequence has the potential to be fetched by the CPU or saved back to the memory by a write operation. However, if the exposure occurs during nonvulnerable sequence, failures visible to the output would not occur.

The proposed model relies on LLC ensemble behavior analysis using trace files obtained from PARSEC benchmark suites running on an extended version of MARSSx86 [10]. The experimental results show that the a only 24% of LLC access are vulnerable over the data array lifetime in the cache.

The remainder of the paper is organized as follows: Section 2 describes the eDRAM cell organization and reliability background. Section 3 introduces our lifetime model used to compute the vulnerability of data. We analyze and evaluate the data array vulnerability to soft errors in Section 4. In addition, the details of experimental results are presented in this section. Section 5 concludes this work.

II. Background

A. eDRAM Cell Organization and Operation

The eDRAM cache structure is similar to the DRAM system organization demonstrated in [17]. The eDRAM cells are organized in a two-dimensional arrays as illustrated in Fig. 1 (a). Each eDRAM cell consists of a dedicated transistor which connects the $bitline$ wire to its associated capacitor. The access transistor in a row are connected and controlled by a wire called $wordline$.

The following steps are required to access the data stored in each row:

- First, all bitline wires need to be $precharged$ to $V_{DD}/2$ as illustrated in Fig. 1 (b).
- As shown in Fig. 1 (c), the row is activated with wordline overdriven to V_{DD} which follows by connecting all capacitors in the row to their corresponding bitlines.
- In the $Charge\ Sharing$ process, the charge either flows from capacitor to bitline or vice versa according to the amount of charge stored in the capacitor.
- The voltage change in the bitline is detected and enlarged using the respective connected sense amplifier. As the final step shown in Fig. 1 (e), the sense amplifier drives the bitline entirely either to V_{DD} or 0 V.

B. eDRAM Retention Time for Different Technology Node

The charge of capacitor is lost overtime through access transistor which makes the refresh operation inevitable for eDRAM cells. The threshold voltage (V_{th}) of the access transistor plays an important role to determine the retention time of an eDRAM cell. The high V_{th} leads to lower leakage which means the eDRAM cell retains state for longer time while low V_{th} provides less retention time due to higher leakage power. The retention time of an eDRAM cell is defined as the leakage time at which the capacitor loses α portion of the stored charge [18] and can be defined as follows:

$$T_{ret} = \frac{\alpha}{\beta} \tag{1}$$

where α is the allowable amount of storage charge which a capacitor can lose and β is the drain current through the access transistor when it is off. Kong et al. [18] showed that the eDRAM cell can stay operational while α is equal to 6/10th of the stored charge. The T_{ret} can also be expressed as in Eq. 2 which comes from [18] and [13].

$$T_{ret} = \alpha \times \frac{L}{W} \times 10^{V_{th}/S_{th}} \times 10^{9}/300sec \tag{2}$$

where W and L are the width and length of the access transistor channel, and S_{th} is subthreshold slope. A detailed explanation of Eq. 2 can be found in [13].

As the technology scales down, the V_{th} of the access transistor reduces accordingly. Hence, if the V_{th} is substituted with a smaller V_{th} in Eq. 2, the eDRAM retention time is reduced.

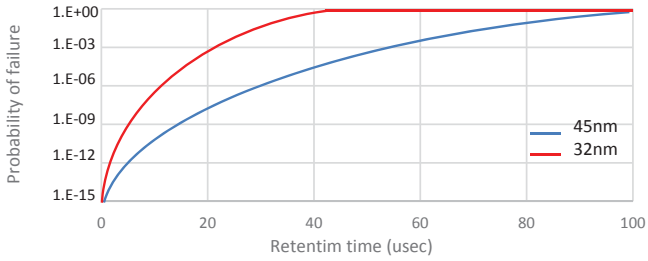


Fig. 2. eDRAM retention time distribution for 45nm and 32nm technology nodes.

This phenomenon has been shown in Fig. 2 in which the probability of a retention failure in eDRAM cell fabricated by $32nm$ technology node is higher than $45nm$ eDRAM cell (derived from Figure 3 in [18]).

C. Single Event Effect

According to the [19], the injection of a high-energy particle into the sensitive region of an electronic device may induce the Single Event Effect (SEE) which probably leads to the bit-flip of the memory cell. This phenomenon begins with the collection of the generated electron-hole pairs in the pn junction through the so-called funneling mechanism as shown in Fig. 3. While the most of released charges are absorbed in the struck junction, the remained charges are diffused into the substrate. In particular, if these impinging charges are collected by a sensitive node such as the reverse biased drain pn junction of the access transistor in an eDRAM cell while it is on the off state, it probably changes the amount of the potential at the drain node and results in the flip of the initial state [11] [20]. The Radiation-induced Soft Error Rate (SER) can be expressed as:

$$SER \approx Area \times exp^{Q_C/Q_{Coll}} \tag{3}$$

where $Area$ is diffusion area of collected charges which is linearly proportional to the cell size of eDRAM, Q_C is the minimum collected charges incurring soft error and Q_{Coll} is the collected charges compiled by drift (Q_{drift}) in depletion layer and diffusion (Q_{diff}) from the silicon substrate. As the size of cell junctions shrinks in eDRAM, the amount of charge collected decreases due to lowered-depth of sensitive depletion region. However, the amount of Q_C decreases even more rapidly due to the power supply reduction that is used for increasing the performance of succeeding generation of eDRAM. Thus, the SER exponentially increases for a new generation device activated by lowered power supply. Empirical data is presented in [14].

D. LLC Reference Characteristics

The eDRAM LLC is shared by all on-die cores to offer worthwhile benefits for such workloads exhibiting significant data-sharing. As illustrated in Fig. 4, the time varying behavior of each benchmark shows multiple sequences of cache reliability issue where some sequences are vulnerable and some sequences that are not. For example, the $streamcluster$ benchmark is a read intensive workload in which a high proportion of shared cachelines are accessed by consecutive read memory operations. Such a cache line generation information

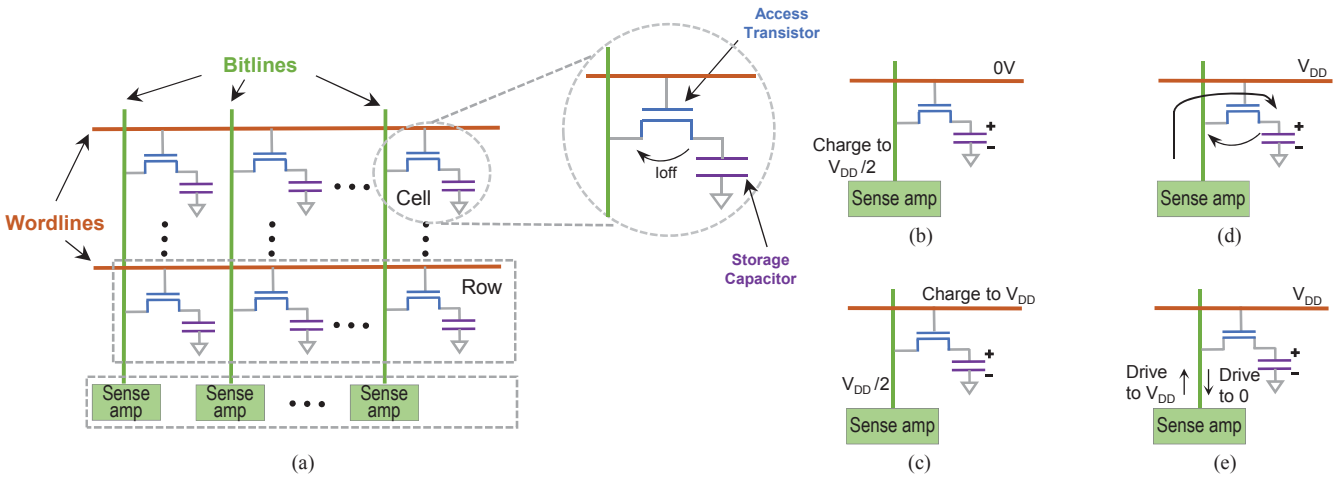


Fig. 1. Typical eDRAM system organization, (a) eDRAM bank, (b) Precharged, (c) Raise of Wordline, (d) Charge sharing, (e) Sensing.

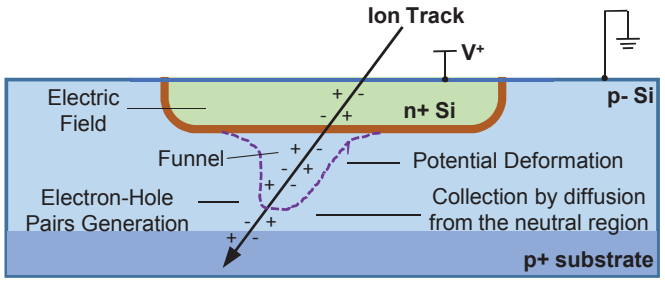


Fig. 3. The impact of ionizing particle in a reverse-biased p-n junction (based on [19]).

can be exploited to show the correlation among the LLC access pattern and the cacheline vulnerability factor.

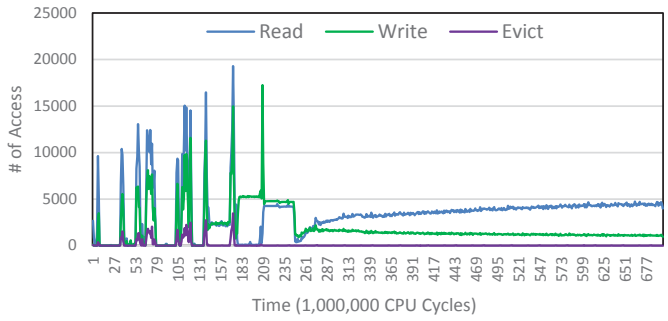
III. The Proposed eDRAM Vulnerability Estimation Algorithm

A. A general Lifetime Model for Cache line

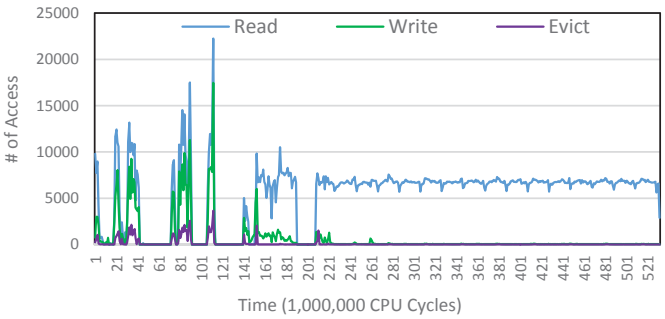
A cache line is brought into the LLC on a read or write miss. Then, it will be accessed either by reads or writes, and finally, it will be replaced by a new cache line. According to the existing LLC reliability analysis methodologies [16] [21], the lifetime of a cacheline can be categorized into the following sequences:

- Read-Read (RR): the lifetime between two consecutive reads
- Write-Read (WR): the lifetime of a write operation up until its first read
- Write-Evict (WE): the lifetime between the last write and the cache line replacement by a new cacheline
- Read-Evict (RE): the lifetime between the last read before cache line replacement
- Read-Write (RW): the lifetime between the last read before the write operation
- Write-Write (WW): the lifetime between two consecutive writes

A *lifetime sequence* is defined as *vulnerable sequence* if an error may propagate out of cache, either to the CPU or to the lower level of memory hierarchy. Apparently, the first four intervals, RR, WR, WE, and RE can be considered as vulnerable sequences. The reason for this is that if any error occurs in aforementioned sequences, it has this potential to be either read by CPU or committed to the memory. On the other hand, the RW and WW sequences are *nonvulnerable sequence* in which if an error occurs, it is simply masked off through overwritten operation, presenting no program failure.



(a)



(b)

Fig. 4. LLC memory access for (a) canneal (b) streamcluster.

B. The Proposed Lifetime Model for eDRAM Data Array

The critical charge of eDRAM is dependent on storage and bit-line capacitance, write voltage, and the minimum voltage difference required by the sense amplier [1]. These factors make an eDRAM cell vulnerable to soft error for a particular portion of lifetime sequence unlike SRAM cell which is vulnerable for the entire sequence period. There are three kinds of SER modes for an eDRAM cell wherein, if any exposure occurs it may result in potential errors:

- Memory mode: This SER mode is the consequence of the injection of a high-energy particle into an eDRAM cell when logic 1 is stored in the storage capacitance.
- Bit mode: The soft errors are produced in this mode if logic 1 is stored in the cell to be read and SEU strikes a bit-line junction during the bit/bit-bar floating time.
- Bit-bar mode: If logic 0 is stored in the cell to be read and a SEU is injected into bit-bar junction during bit/bit-bar floating time, then it can produce bit-bar mode soft errors.

The error produced is vulnerable exclusively if it occurs in abovementioned general lifetime sequences because it is either consumed by CPU or committed to the memory. Thus, to estimate the vulnerable sequence of eDRAM, it is necessary to integrate the existing lifetime model with the SER modes for eDRAM. In order to estimate the BUVF, all three kinds of SER modes for eDRAM cell must be considered. The memory and bit modes soft errors are taken place when the storage node has a logic 1 value while bit-bar mode occurs when the eDRAM cells contains logic 0 value.

To estimate BUVF for an eDRAM cell containing logic 1 value, the entire lifetime sequence can be considered as vulnerable sequence if the second memory operation in two consecutive accesses to that data item is either read or evict. As shown in Fig. 5 (a), the *sequence A* is a WW sequence which is not vulnerable to soft errors. The *sequence B* represents the vulnerable WR sequence in which the last write operation is following by three refresh operations and then by a read operation.

On the other hand, the BUVF for eDRAM cell containing logic 0 value is estimated according to the bit/bit-bar floating time allocated in vulnerable sequence. As shown in Fig. 5 (b), *sequence C*, *E* and *G* are vulnerable sequences where the eDRAM cell is in its refresh mode when logic 0 is stored in it. Even though this vulnerable sequence may seem negligible, it shares a significant portion of BUVF when we take the large size of eDRAM and number of refresh operations originating within each vulnerable sequence into consideration.

C. Bit-Upset Vulnerability Factor Analysis of eDRAM Data Array

The impact of soft errors in the SRAM-based LLC has been investigated by numerous scholars in the past [22], [21], [16]. However, by replacing the low-capacity SRAM-based LLC with the new LLC design which benefits from high density eDRAM technology, new vulnerable sequences are introduced to the system. This observation inspired us to introduce new bit-upset vulnerability factor called BUVF which is inspired

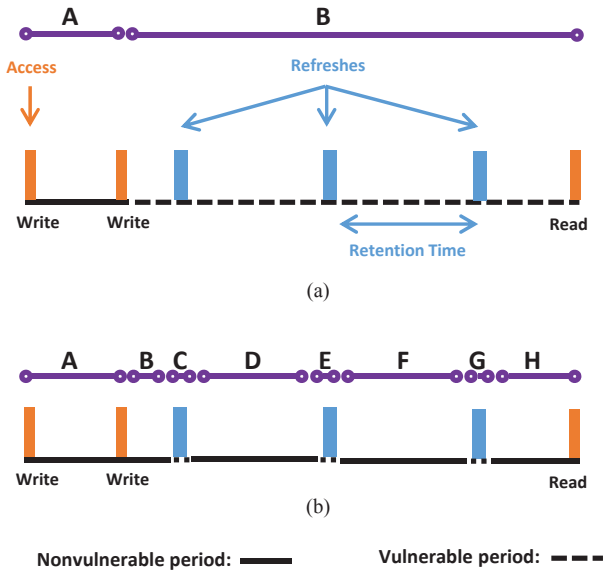


Fig. 5. The lifetime of eDRAM cell. (a) the logic 1 is stored in the cell, (b) the logic 0 is stored in the cell.

by TVF for SRAM [16]. The BUVF is the proportion of data items in the vulnerable sequences w.r.t the total data items that are located in the cache. The BUVF can be calculated as follows:

$$BUVF = \frac{(vdi(0) + vdi(1))}{\sum(exec\_time \times data\_items)} \tag{4}$$

where $vdi(0)$ and $vdi(1)$ are the vulnerable period for eDRAM cell containing logic value 0 and 1, respectively. Thus, BUVF can be calculated as the summation of $BUVF_0$ and $BUVF_1$ for eDRAM cell storing logic value 0 and 1, respectively. The $BUVF_0$ can be calculated as:

$$BUVF_0 = \frac{\sum_{i=1}^{n}(data\_item(0)_i \times ref\_dur \times \sum_{j=1}^{l} vul\_sequence_j)}{\sum(exec\_time \times data\_items)} \tag{5}$$

where $data\_item(0)_i$ is the i^{th} eDRAM cell storing logic 0 value, ref_dur is the period time for each refresh operation and l represents the number of refreshes occurring in the duration time of j^{th} vulnerable sequence for $data\_item(0)_i$. Meanwhile, the $BUVF_1$ is:

$$BUVF_1 = \frac{\sum_{i=1}^{n}(data\_item(1)_i \times \sum_{j=1}^{k} vul\_sequence_j)}{\sum(exec\_time \times data\_items)} \tag{6}$$

where $data\_item(1)_i$ is the i^{th} eDRAM cell containing logic 1 value, $vul\_sequence_j$ is the duration time of j^{th} vulnerable sequence for $data\_item(1)_i$ and k represents the boundary which is the end of memory access to $data\_item(1)_i$.

A high value of BUVF for a data item located in eDRAM LLC indicates reduced resiliency to soft errors. Therefore, we recast the problem of reliable eDRAM LLC design as a straightforward search for reduced BUVF.

IV. Experimental Setup

We analyze LLC ensemble behavior using traces from an extended version of MARSSx86 [10]. The traces capture

access type to LLC which are typically read, write, and evict. A Chip MultiProcessor (CMP) is selected for modeling that has eight x86 cores which are single-threaded. The detail of our model can be found in Table I. Twelve applications from the PARSEC benchmark suite are selected and 500 million instructions are executed starting at the Region Of Interest (ROI) after warming up the cache for 5 million instructions. Furthermore, the simsmall input sets are used for all PARSEC applications.

We evaluate the proposed scheme for all vulnerable sequences. However, RR and WR are the dominant contributors to the BUVF of LLC data array because the large size of LLC provides long-term residency for data items in LLC which results in the remained vulnerable sequences to be less active during program execution. Accordingly, the results of data vulnerability analysis in two major contributors to the BUVF are described in this paper.

Fig. 6 (a) shows the proportion of data pieces residing in vulnerable sequences. We categorized each vulnerable sequence into three lengths based on the time interval between two consecutive accesses to data: *1) Short, 2) Medium, 3) Long*, denoted as SRR/SWR, MRR/MWR, or LRR/LWR, respectively. The short duration sequence indicates data instances with an interval less than 1 million cycles while this interval increases from 1 million to 50 million cycles for medium sub-vulnerable sequence. The data elements with intervals exceeding 50 million cycles are considered as long duration sequence. For example, the rightmost orange column for each benchmark represents the long-term WR sequence which averages 0.23% across the suite. The second green column from the left shows the RR elements with medium duration interval contributing the largest share to the BUVF.

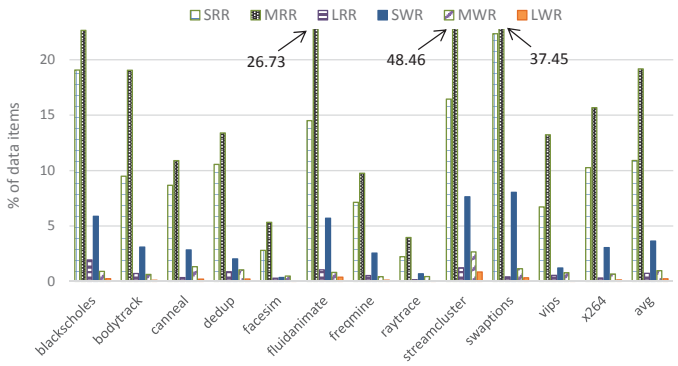
Fig. 6 (b) illustrates the sequence distribution of data instances in eDRAM LLC. The vulnerable sequences account for about 27.24% of data array lifetime in the cache, among which RR contributes about 23.45%. The data instances in MRR and LRR sequences are significant contributors to the he RR time overall. Although the second largest set of vulnerable data are elements located in SRR sequence, this portion of cache space contributes the second least vulnerability to BUVF which is the result of spending a small fraction of program execution time in SRR sequence. This observation confirms that the long-term residency of a data block in LLC between two consecutive accesses significantly increases its vulnerability to soft errors. The profile results convince us that the read intensive benchmarks with medium and long read-read instances, i.e. $streamcluster$, contribute the most BUVF in the data cache.

The results obtained by BUVF characterization and analysis are shown in Fig. 7. The lower BUVF value corresponds to a lower SER. The BUVF values for $raytrace$ and $facesim$ benchmarks are 0.020 and 0.022, respectively, indicating the high resiliency of them to soft errors. The main reason behind the reduced BUVF value for these benchmarks is that the data remained for a shorter period within the vulnerable sequences. On the other hand, the BUVF value of $streamcluster$ is 0.76 which is the highest BUVF among benchmarks under study.

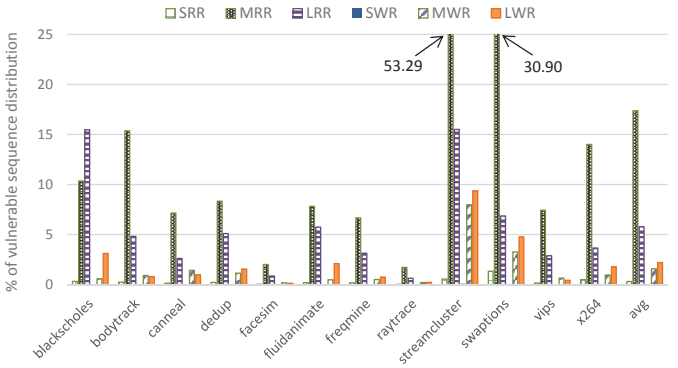
Even though the data located in LRR sequence only account for 0.73 percent of total data items as shown in Fig. 6 (a), this sequence is the second largest contributor to BUVF. Furthermore, the two potential duration sequences MRR and LRR together account for 0.16 of vulnerability factor across all workloads, pointing out that the RR sequence contributes the most to BUVF. This agrees with typical distribution of data access being read-predominant versus write-perdominant.

TABLE I. SIMULATOR CONFIGURATION

Processor	3GHz processor Fetch/Exec/Commit width 4
Private L1-I/D	SRAM, 32 KB, 8-way set assoc., MESI cache
Private L2 Conf.	SRAM, 512 KB, 8-way set assoc., MESI cache
Shared L3	eDRAM, 96 MB, 16-way set assoc., 16 bank, WB cache
Main memory	8 GB, 1 channel, 4 ranks/channel, 8 bank/rank



(a)



(b)

Fig. 6. Distribution of lifetime: (a) the proportion of data items in vulnerable sequence, (b) the sequence distribution of data items.

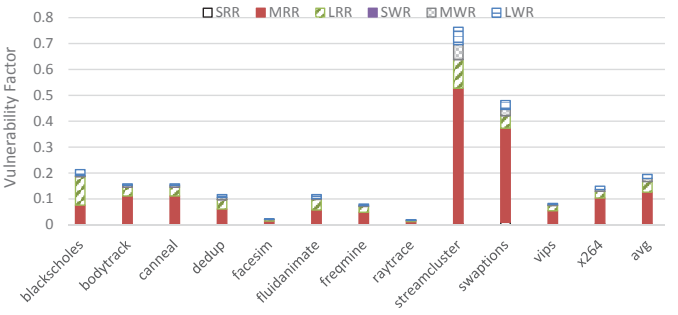


Fig. 7. Measured BUVF behavior for the workloads of the PARSEC suite.

V. Related Work

Asadi et al. [22] proposed algorithms of vulnerability computation for both L1 and L2 caches. The authors indicated the vulnerability breakdown of data, tag-addresses, and status bits. Tag error is analyzed in detail and classified into three categories. Whereas Asadi′s research is forced on L1 and L2 private cache, Maghsoudloo et al. [21] analyzed the influence of coherence protocol on the susceptibility of shared LLC. Two prediction schemes are proposed to provide correction ability for dirty data.

The authors of [16] conducted a study to provide insight into the cache design for manufacturing highly efficient reliable on-chip cache by assessing the reliability behavior of cache memories. Meanwhile, an analytical framework is a proposed by Suh et al. [2] to measure the failure rate in L2 SRAM cache under any soft error protection scheme. However, these two works are not specifically designed for eDRAM.

In terms of eDRAM and DRAM reliability analysis work, Fang et al. [14] developed an efficient method to predict scaling trends for both neutron- and alpha- soft error rate. However, our scheme is suitable for a wide range of soft errors. Shin [1] proposed a unified model for alpha-particle-induced charge collection.

VI. Conclusion

The recent on-chip eDRAM cache LLC can leverage cache residency lifetime to assess the vulnerability to soft error. The proposed BUVF model investigates three SER modes for eDRAM cells in which the minimum collected charge has the potential to induce soft error. The Memory, bit and bit-bar SER modes are identified and integrated into the proposed BUVF to accurately capture the various kinds of possible soft errors in eDRAM cell. We evaluated our model using LLC ensemble behavior traces obtained from MARSSx86 running PARSEC 2.1 applications. The major contributors to the LLC vulnerability are identified according to the results obtained from our BUVF analysis. Our results indicate that the two potential duration sequences MRR and LRR together contribute the most to BUVF, and thus are the primary sources of the instability within LLC on a multi-core processor die.

Acknowledgment

This work is supported in part by the US National Science Foundation Grant CCF-1527249, CCF-1337244 and National Science Foundation Early Career Award 0953946.

References

[1] H. Shin, "Modeling of alpha-particle-induced soft error rate in dram," vol. 46, no. 9, 1999, pp. 1850–1857.

[2] J. Suh, M. Manoochehri, M. Annavaram, and M. Dubois, "Soft error benchmarking of l2 caches with parma," vol. 39, no. 1, 2011, pp. 85–96.

[3] G. H. Loh and M. D. Hill, "Efficiently enabling conventional block sizes for very large die-stacked dram caches," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44. New York, NY, USA: ACM, 2011, pp. 454–464.

[4] G. Kurian, O. Khan, and S. Devadas, "The locality-aware adaptive cache coherence protocol," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ser. ISCA '13. New York, NY, USA: ACM, 2013, pp. 523–534.

[5] G. Loh and M. Hill, "Supporting very large dram caches with compound-access scheduling and missmap," *Micro, IEEE*, vol. 32, no. 3, pp. 70–78, May 2012.

[6] M. Watanabe, N. Niioka, T. Kobayashi, R. Karel, M.-A. Fukase, M. Imai, and A. Kurokawa, "An effective model for evaluating vertical propagation delay in tsv-based 3-d ics," in *Quality Electronic Design (ISQED), 2015 16th International Symposium on*, March 2015, pp. 519–523.

[7] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. Loh, D. McCauley, P. Morrow, D. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb, "Die stacking (3d) microarchitecture," in *Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on*, Dec 2006, pp. 469–479.

[8] Y. Deng and W. P. Maly, "Interconnect characteristics of 2.5-d system integration scheme," in *Proceedings of the 2001 International Symposium on Physical Design*, ser. ISPD '01. New York, NY, USA: ACM, 2001, pp. 171–175. [Online]. Available: http://doi.acm.org/10.1145/369691.369763

[9] S. Roy, S. Chatterjee, C. Giri, and H. Rahaman, "Faulty tsvs identification and recovery in 3d stacked ics during pre-bond testing," in *3D Systems Integration Conference (3DIC), 2013 IEEE International*, Oct 2013, pp. 1–6.

[10] M.-T. Chang, P. Rosenfeld, S.-L. Lu, and B. Jacob, "Technology comparison for large last-level caches (l3cs): Low-leakage sram, low write-energy stt-ram, and refresh-optimized edram," in *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, Feb 2013, pp. 143–154.

[11] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," in *Device and Materials Reliability, IEEE Transactions on*, vol. 5, no. 3, Sept 2005, pp. 305–316.

[12] S. Ganapathy, R. Canal, D. Alexandrescu, E. Costenaro, A. Gonzalez, and A. Rubio, "A novel variation-tolerant 4t-dram cell with enhanced soft-error tolerance," in *Computer Design (ICCD), 2012 IEEE 30th International Conference on*, Sept 2012, pp. 472–477.

[13] A. Agrawal, A. Ansari, and J. Torrellas, "Mosaic: Exploiting the spatial locality of process variation to reduce refresh energy in on-chip edram modules," in *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, Feb 2014, pp. 84–95.

[14] Y.-P. Fang, B. Vaidyanathan, and A. Oates, "Soft error rate cross-technology prediction on embedded dram," in *Reliability Physics Symposium, 2009 IEEE International*, April 2009, pp. 925–928.

[15] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: exploiting generational behavior to reduce cache leakage power," *ACM SIGARCH Computer Architecture News*, vol. 29, no. 2, pp. 240–251, 2001.

[16] S. Wang, J. Hu, and S. Ziavras, "On the characterization and optimization of on-chip cache reliability against soft errors," vol. 58, no. 9, Sept 2009, pp. 1171–1184.

[17] J. Liu, B. Jaiyen, Y. Kim, C. Wilkerson, and O. Mutlu, "An experimental study of data retention behavior in modern dram devices: Implications for retention time profiling mechanisms," in *ACM SIGARCH Computer Architecture News*, vol. 41, no. 3. ACM, 2013, pp. 60–71.

[18] W. Kong, P. Parries, G. Wang, and S. Iyer, "Analysis of retention time distribution of embedded dram - a new method to characterize across-chip threshold voltage variation," in *Test Conference, 2008. ITC 2008. IEEE International*, Oct 2008, pp. 1–7.

[19] L. Ratti, "Ionizing radiation effects in electronic devices and circuits," *INFN Laboratori Nazionali di Legnaro, National Course in Detectors and Electronics for High Energy Physics, Astrophysics, Space Applications and Medical Physics*, 2013.

[20] J. Yang and et al., "Radiation-induced soft error analysis of stt-ram: A device to circuit approach," in *Computer-aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 2015.

[21] M. Maghsoudloo and H. Zarandi, "Dirty data vulnerability mitigation by means of sharing management in cache coherence protocols," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2012 IEEE International Symposium on*, Oct 2012, pp. 205–210.

[22] H. Asadi, V. Sridharan, M. Tahoori, and D. Kaeli, "Vulnerability analysis of l2 cache elements to single event upsets," in *Design, Automation and Test in Europe, 2006. DATE '06. Proceedings*, vol. 1, March 2006, pp. 1–6.