A self-scrubbing scheme for embedded systems in radiation environments

Yufan Lu
CSEE
University of Essex
Colchester, UK
yl19888@essex.ac.uk

Xiaojun Zhai
CSEE
University of Essex
Colchester, UK
xzhai@essex.ac.uk

Sangeet Saha
CSEE
University of Essex
Colchester, UK
sangeet.saha@essex.ac.uk

Shoaib Ehsan
CSEE
University of Essex
Colchester, UK
sehsan@essex.ac.uk

Klaus McDonald-Maier
CSEE
University of Essex
Colchester, UK
kdm@essex.ac.uk

***Abstract*—SRAMs are very sensitive to radiation effects. When embedded systems working in the extreme radiation environments, bit flips may occur frequently and decrease the reliability of the systems significantly.**

In this paper, a self-scrubbing scheme is proposed for embedded systems in extreme radiation environments. In the proposed scheme, both scrubbing and error correcting codes are used to mitigate a large number of the errors in the RAMs. Along with this, a separate scrubber is designed to scrub the RAM independently, when the CPUs are busy. In addition, the scrubber is a portable module and the hardware cost does not grow with the size of the available RAM. The results of the experiments show that, in the neutron radiation environments where the error rate in the unhardened RAMs is approximately $1.2bit/(\mathbf{KB}{\cdot}h)$, while the error rate of the self-scrubbing RAMs is less than $8.7 \times 10^{-5} bit/(\mathbf{KB}{\cdot}h)$, which is one fifth of the error rate of the conventional ECC RAMs.

***Index Terms*—Neutron Radiation, SRAM, SEU Mitigation, ECC.**

I. Introduction

Radiations exists throughout the entire solar system, and includes solar flares, solar wind, galactic cosmic radiation and radiation emitting from nuclear reactions. It is well known that radiation rays with high energy (e.g. X-ray, gamma, neutron, electrons, protons and heavy ions rays) can cause heavy damage on semi-conductor devices [1]–[3]. When an electronic system operates in high radiation environments, the radiation may change the status of the circuits by producing the incorrect outputs, decrease performance and can even damage the electronic devices permanently [4].

Cumulative effects and single event effects (SEE) are two main types of radiation effects on the integrated circuits. Cumulative effects (e.g. total ionizing dose (TID) [5], [6] and displacement damage (DD) [7]–[9]) are unrecoverable long-term effects. They can change the parameters of semiconductor materials and make the circuits malfunction completely. SEE is an effect caused by single charged particle. SEE includes a number of types: 1) single event transient (SET), 2) multiple-Bit upset (MBU), 3) single event functional interrupt (SEFI), 4) single event latch-up (SEL), 5) single event burnout (SEB), 6) single event gate rupture (SEGR) and 7) the single event upset (SEU). Among these, SET and SEU are the most common effects causing soft errors (recoverable or transient errors), while SEL, SEGR and SEB cause permanent errors in electronic systems [10]–[13].

Therefore, it is necessary to harden the embedded systems which are supposed to work in radiation environments. Compared with the normal embedded systems, the system dedicated to the nuclear environments need to be able to operate in the environments with much higher radiation levels. For example, the radiation level was more than 10 Sv/h during the three mile island nuclear accident [14]. In such high radiation environments, system failures occur within a short time and operation time also decreases drastically. As reported in the Chernobyl nuclear accident in 1986, the radiation dedicated robot, STR-1, only operated for total 200 hours to clean the building surface even it was protected by shells [15]. In the areas with higher radiation levels, operation time could be shorter which ranges from few minutes to few hours.

Considering that those devices are not likely reusable after exposure under radiation, a strategy is proposed to use cheaper devices instead of expensive radiation hardened devices in the radiation tasks to lower the total costs. However, those cheap devices, especially SRAM parts in the embedded systems, are vulnerable to the radiation effects. Due to the feedback mechanism for data retention in the SRAMs, the bit flips occur frequently in the high radiation environments [16]. The SRAM parts have to be hardened for radiation tasks. In order to mitigate errors in the memory systems, a series of error mitigation strategies have been taken into consideration including the triple modular redundancy (TMR) technology, Error correcting codes (ECC) and Scrubbing technology [17]–[21]. TMR is widely used to protect the key-systems. Although, it triples the hardware resources to achieve high reliability [22], [23], it aso increases the costs of the radiation tasks significantly. In addition, it can not correct the cumulative errors, typical for the extreme radiation environments. ECC is widely known as an anti-interference encoding strategy to enhance the reliability of memory devices and communication systems [24]–[29]. However, this strategy is vulnerable to the multi-bit error [30]–[34]. If the number of the errors in the

Manuscript received February 20, 2020

Y. Lu, X. Zhai, S. Saha, S. Eshan and K. McDonald-Maier are with the School of Computer Science and Electronic Engineering, University of Essex, UK.

memory units exceeds the fault tolerant limit, the errors can not be corrected. Scrubbing technologies [35] are used to mitigate the cumulative errors. It uses free clocks periods to read back the data from the memory to operate the correction. However, if the CPUs are busy, the scrubbing rates may be lower than error rates in the extreme radiation environments.

In this paper, a self-scrubbing scheme with error correcting code is proposed to harden the embedded systems for extreme radiation environments. The scrubbing operation is controlled by a separate module, which is running at the twice the frequency of the system clock. In this way, there will be double bandwidth to operate the instructions from the CPUs and the scrubbing operations from the separate controller simultaneously. The original instructions will not be affected. Therefore, it can be easily transplanted in the existing embedded systems. Real world radiation experiments were conducted with a neutron source at the ChipIr facility at ISIS, Didcot, UK [36]. The flux of the neutron beam is $5 \times 10^6 cm^2 s^{-1}$. In the experiment, the observed errors rates in unhardened RAM is $1.2bit/(\text{KB}{\cdot}h)$. The errors rates in conventional ECC ram is approximately $4.3 \times 10^{-4} bit/(\text{KB}{\cdot}h)$, while the self-scrubbing RAM is less than $8.7 \times 10^{-5} bit/(\text{KB}{\cdot}h)$.

The main contributions of this article are stated as follows:

1) The proposed design is highly flexible and reliable. Compared to the conventional scrubbing approach, the proposed design requires no free periods to conduct the scrubbing operations. The memory units can be scrubbed at the high frequency to ensure the errors can be corrected in time. In addition, instructions from the CPUs and the scrubbing operations can be carried out simultaneously. The original program will not be affected. Therefore, it can be applied in various embedded systems without software modifications.
2) The proposed design is an area-efficient design, which can be used to harden low-cost computer systems. The separate scrubber consumes significantly less hardware resources in comparison to TMR designs. The implementation on an entry-level FPGA (i.g. XC7A15T-1CPG236C) shows that it only costs 150 LUTs to build the controller for a 32KB RAM. The scalability tests also show that the costs of the design does not scale up linearly with the RAM size. When the RAM size scales from 8KB to 256 KB, the costs of the LUTs scale from 128 to 212.
3) The proposed SEU mitigation design can achieve high SEU correction rate for the radiation hardened system. The results of the real world radiation experiments show that it can correct most errors in the RAM under neutron radiation, where the error rates in unhardened RAMs is approximately $1.2bit/(\text{KB}{\cdot}h)$. In the 6-hours radiation exposure experiments, the error rate of the self-scrubbing RAM is $8.7 \times 10^{-5} bit/(\text{KB}{\cdot}h)$, which is significantly less than the error rate of the conventional ECC RAM.

The paper is organised as follows. Section II presents the architecture of the Self-Scrubbing scheme. Section III shows the hardware implementation of the systems and the scalability of the design. Section IV shows the results of neutron radiation experiment designs and section V concludes the paper.

II. Architecture of the self-scrubbing scheme

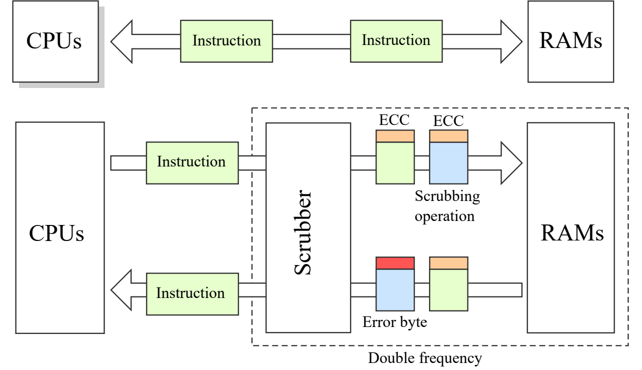

Fig. 1. Sequence of input operations stream

The architecture of self-scrubbing scheme is shown in Fig. 1. Compared to the conventional CPU-RAM structure, a separate scrubber is introduced between CPUs and RAMs. The original data stream and instructions between CPUs and RAMs will be transmitted by the scrubber. In this scheme, the scrubber and the RAM part operates at double the frequency of the systems, so there will be twice the bandwidth for the scrubber to conduct the scrubbing operations (e.g. fault detection and error correction). When the scrubber conducts the instructions from CPUs and the scrubbing operation simultaneously, half of the system clock periods will be used for the scrubbing operations and the original timing sequence of the instructions for the CPUs will not be changed. Therefore, it can be easily implemented in the existing radiation dedicated designs.

A. The design of the scrubber

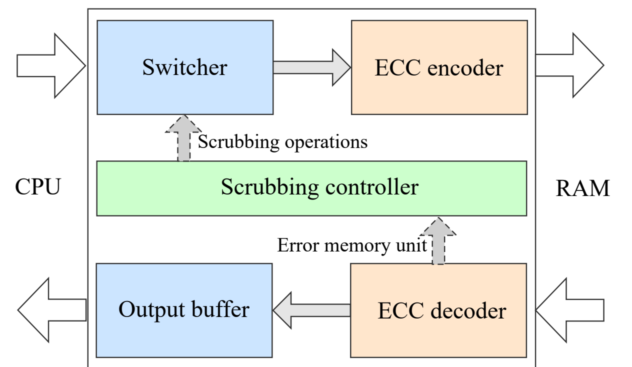

Fig. 2. Sequence of input operations stream

The design of the scrubber is shown in the Fig. 2. There are 5 modules in the scrubber which are 1) switcher, 2) scrubbing controller, 3) output buffer, 4) ECC encoder and 5) ECC decoder. The scrubbing controller generates the scrubbing operations, which has two working modes, 1) detection and 2)

correction modes. If there is no detected error, the controller will work in the detection mode and generate the reading instructions to read the data from the RAM. If the error bits in the return data are detected, the controller will work in the correction mode and generate writing instructions to overwrite the error bits. As mentioned, the scrubber is operating at the double frequency, so the scrubbing controller and the CPUs are working in the different clock domains. The switcher is the module to mix the operations from different clock domains. In this design, (12,8) hamming code is used as correcting code. It is a simple correcting code, that can be easily implemented. The ECC encoder and decoder are combinational circuits. There is no additional delay for CPUs' instructions. After decoding, the data required by CPUs will be sent to the output buffer to rebuild the timing sequence. The rest of the data will be sent to controller for detection.

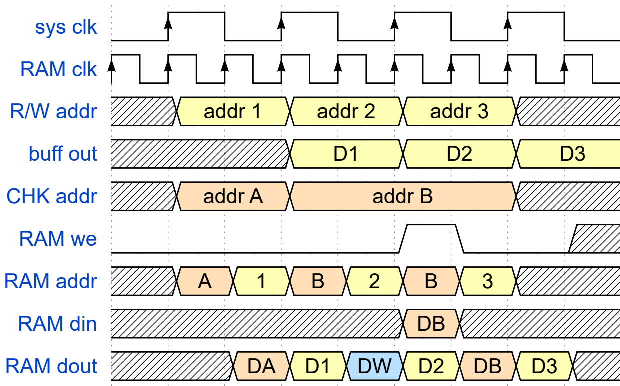

Fig. 3. Timing Diagram of the RAM.

B. Timing sequence of the scrubbing

The timing diagram of the RAM is shown in the Fig. 3. RAM clk and sys clk represent the RAM clock and the system clock respectively. RAM clock operates at twice of the frequency of the system clock. The CHK addr is the checking address which indicates that the address is being checked. The R/W addr is the address operated by the external modules. In this figure, the CHK addresses and R/W addresses are represented by alphabet and numbers respectively to indicate that the external module and the internal module are accessing the different addresses. The 'we', 'addr', 'din' and 'dout' of the RAM in this figure represent the write enable, address, data in and data out of the RAM respectively. The buff out is the outputs of the scrubber which is connected to the external modules.

In this figure, the CPU is accessing address 1, address 2 and address 3. The scrubber is accessing the address A and address B. The R/W operations and the scrubbing operations are arranged one by one. The 'D1', 'D2', 'D3', 'DA' and 'DB' represent the data from the address 1, 2, 3, A and B. The 'DW' indicates the detected error data. When data from address B is incorrect, the scrubber will trigger a scrubbing mode with a writing operation to overwrite the memory unit. If there is no detected error, it will take two RAM clock cycles to check the unit. If there are detected errors, it will require four RAM clock cycles to correct the errors.

C. Address conflicts

It is possible that the CPUs and the scrubber access the same memory units at the same moment, which could cause address conflicts. In this case, the scrubber could overwrite the units with the outdated data. In order to solve that issue, the scrubber will monitor the current address. If there is a address conflict, the scrubber will skip the current checking address. In addition, considering that most R/W operations are conducted in the order of addresses, the sequence of the checking addresses are designed in the reverse address order.

III. Hardware implementation of the design

A. Architecture of the evaluation system

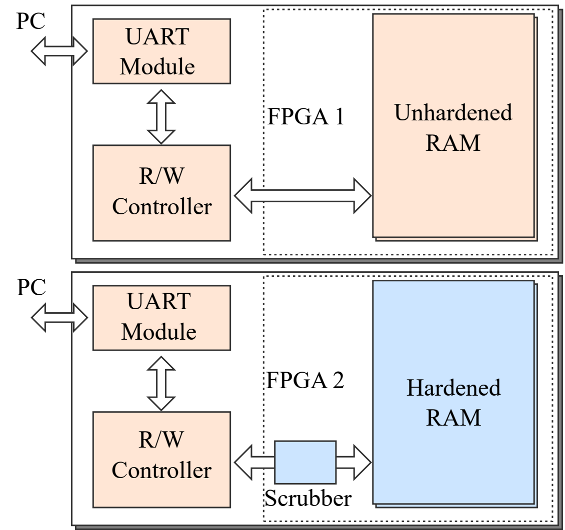

Fig. 4. The architecture of the prototype system

The architecture of the system is shown in the Fig. 4. The hardware design includes an UART module, R/W controller and RAM. The UART module is responsible for the UART communication. When the boards start to work, the host will sent reading and writing instructions to operate the RAM via UART. The R/W controller is the module which is used to conduct the reading and writing instructions. The RAM parts in the system is replaceable. It can be unhardened RAMs or hardened RAMs according to the respective requirements. In this figure, the design on the FPGA 1 is used to represent the system without hardened RAM. The design on the FPGA 2 is used to represent the system hardened by the self-scrubbing scheme.

When the FPGA boards are exposed in the radiation environments, the host will send the instructions to read back the data from the RAM. By comparing the current data to the previous data, the error bits can be detected. By using this system, we can analyze the error rate of the unhardened and hardened RAM systems in the radiation environments.

B. Costs of the hardware resources

The comparison has been carried out between the unhardened RAMs and the hardened RAMs with the same available RAM size and the same bandwidth for CPUs and external modules. As the self-scrubbing RAM uses more bandwidth and more memory for error correction, the actual frequency and the size of the self-scrubbing RAM is higher.

Two types of the hardware resources, memory size and the LUTs, are used to build the self-scrubbing design. The additional costs of the RAMs are used to store correcting codes. The LUTs are used to build the scrubber.

In this scheme, (12,8) hamming codes are used as the error correcting code. Therefore, for each memory units, there will be 12 bits in total. Among them, 8 bits are for data and the other 4 bits are used for correcting codes. Therefore, it consumes 50% more memory. The costs of the RAM grow linearly with the size of the available RAM size.

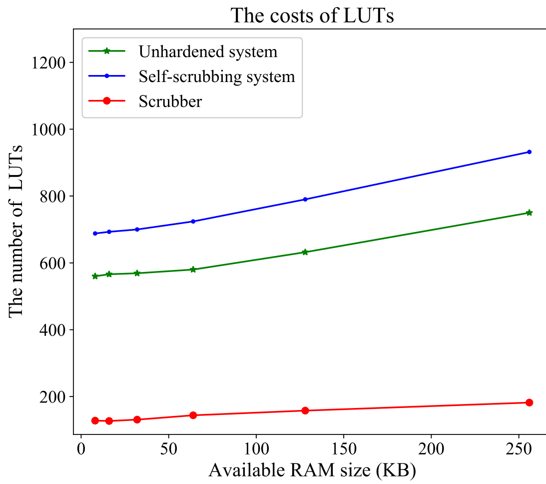

Fig. 5. The the hardware costs with the scale of the design

The comparison of the LUT costs is shown in the Fig. 5. The costs of the self-scrubbing system and the unhardened systems grow simultaneously with the size of the available RAM. The costs of the scrubber, which is the additional part of the scrubber, does not grow linearly with size of the available RAM. When the available RAM size is 8KB, the unhardened and hardened designs consume 560 and 688 LUTs respectively. When the available RAM size is 256KB, the unhardened and hardened designs consume 720 and 932 LUTs respectively. The additional number of the LUTs only increases slightly for higher address width in scrubber controller.

IV. Neutron Radiation Experiment

In this paper, we conducted the real world radiation experiments to evaluate the performance of the design. The experiments were conducted with a neutron source at the ChipIr facility at ISIS, STFC Rutherford Appleton Laboratory, Didcot, UK [36]. ChipIr provides a neutron spectrum which is suitable to emulate effects of terrestrial neutrons in electronic devices and systems. The ChipIr neutron flux (with $E_n > 10$ MeV) has been measured to be approximately $5 \times 10^6 cm^2 s^{-1}$. The neutron flux at ChipIr is about 8 to 9 orders of magnitude higher than the terrestrial flux at sea level.

In the experiments, the operating FPGA boards operate in the radiation room for hours. The host PC in the controller room is connected to the FPGAs via UART. It will keep reading data from the RAMs in the working FPGA every 5 seconds. If the return data from FPGAs is different from the previous data, the error bits can be detected.

The self-scrubbing design is implemented on the Digilent Cmod A7-15T. It is a low price entry-level FPGA development board. There is a Artix-7 XC7A15T-1CPG236C FPGA with 112.5 KB block RAM on the boards. We designed three experiments in this paper: 1) Comparison between unhardened RAMs with different memory size, 2) Comparison between unhardened RAMs and the self-scrubbing RAMs 3) Comparison between the conventional ECC RAMs and the self-scrubbing RAMs.

A. Comparison between unhardened RAMs with different memory size

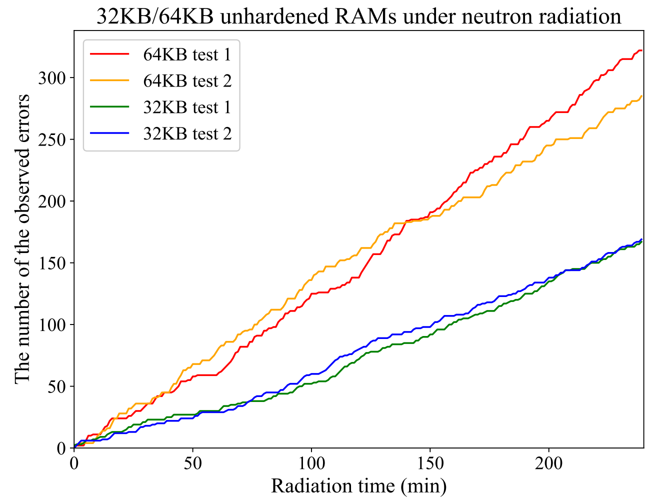

Fig. 6. The number of the errors in the unhardened RAMs

The number of the errors in the unhardened RAMs with different memory size is shown in the Fig. 6. In the experiments, we conducted the experiments with the 32KB RAMs and 64 KB RAMs for 4 hours. In the experiments, both RAMs were conducted twice. We use average numbers of the errors to compare the error rates.

As we can see in the figure, the number of the observed errors in both 32KB and 64KB RAMs increase linearly with the radiation time. After 4 hours radiation experiments, the average number of the observed errors in the 32KB RAMs is 159. The average number of observed errors in the 64KB RAMs is 303, which is approximately double of the number

of the errors in the 32KB. The errors rates of the RAM in the given radiation environments can be represented as follow:

$$\boldsymbol{R} = \frac{N_{error}}{T_{radiation} \cdot S_{RAM}}, \quad (1)$$

where the R represents the error rates of the RAM in the given environment, N_{errors} represents the number of the error bits in total, $T_{radiation}$ represents the radiation time and the S_{ram} represents the size of the RAM. Therefore, It can calculated that the observed error rates of the RAMs under the neutron radiation is approximately $1.2bit/(\text{KB}\cdot h)$. The results also show that the error rates remain robust irrespective of the RAM size.

B. Comparison between unhardened RAMs and the self-scrubbing RAMs

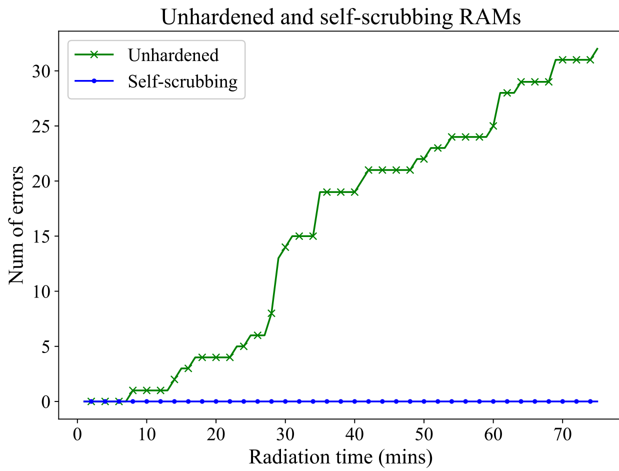

Fig. 7. Setup of Neutron radiation experiment

The comparison between unhardened RAM and self-scrubbing RAM is shown in the Fig. 7. The available unhardened RAM and the self-scrubbing RAM are both 32KB. The number of bit flips in self-scrubbing ECC RAM remains at zero during the entire experiment while the number of bit flips in unhardened RAM rises to 32 in the initial 1 hours. As both RAMs are working in the same radiation environment, it proves that the design of self-refresh ECC RAM is effective for SEU mitigation.

C. Comparison between the conventional ECC RAMs and the self-scrubbing RAMs

The comparison between built-in ECC RAM and self-scrubbing RAM is shown in the Fig. 8. In this experiments, the built-in ECC RAM is the normal RAM haredened by Xilinx official ECC modules, which is used as the reference RAM. Both built-in ECC RAM and the self-scrubbing RAM are operating under neutron radiation for more than 6 hours. After the 360 minutes radiation experiment, the total number of observed errors in the conventional ECC RAM is five, while the number of the errors in the self-scrubbing RAM is only

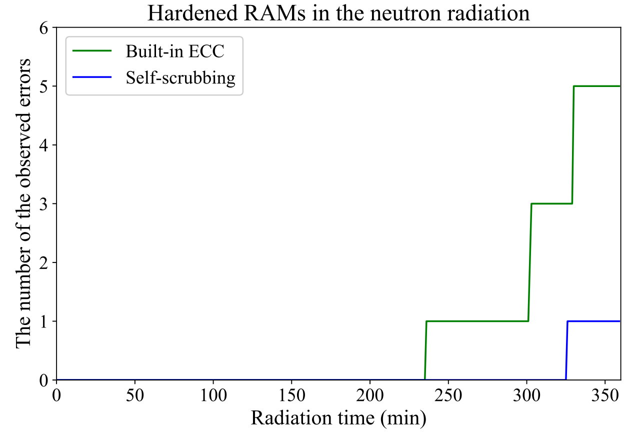

Fig. 8. Setup of Neutron radiation experiment

one. It can also be seen there are increasing number of errors after 200 minutes radiation. It is because that the conventional ECC RAMs use the ECC decoder to correct the output data without refreshing the stored data inside the RAMs. When there are more error bits accumulated in the RAM, it is more likely that the number of error bits in one unit will exceed the fault tolerance. The hamming code used in this paper is one bit error correction code, so the fault tolerance is one bit.

In the experiment, the first observed error in the conventional ECC RAM is detected after 234 minutes radiation, while the first observed error in the self-scrubbing RAM is detected after 320 minutes. It shows that the self-scrubbing RAMs have the better performance of error mitigation than the conventional ECC RAMs in the neutron radiation environments. In the experiments, the error rates in the conventional ECC RAM is approximately $4.3 \times 10^{-4} bit/(\text{KB}\cdot h)$, while the self-scrubbing RAMs is less than $8.7 \times 10^{-5} bit/(\text{KB}\cdot h)$, which is one fifth of the error rate of the conventional ECC RAMs.

V. Conclusion

This paper proposes a scheme that combines ECC and scrubbing methods to mitigate SEUs for the devices are supposed to work in extreme radiation environments. Considering the high error rates in the extreme radiation environments, both ECC and scrubbing are used in the scheme to achieve high reliability.

In the scheme, a scrubber is designed to scrub the RAM data separately and simultaneously. Hence, the scrubbing operations can be carried out even when the CPUs are busy. The hardware costs of the scrubber only grow slightly with the available RAM size. Therefore, it is suitable for the cheap devices with limited resources.

The experiments are conducted in the neutron radiation environments. It is shown that the error rates remain robust irrespective of the RAM size. The comparison of the radiation experiments also show that the self-scrubbing scheme is an

effective strategy to hardened the embedded system and the error rate of the self-scrubbing RAM is one fifth of the conventional ECC RAM.

ACKNOWLEDGMENT

This work is supported by the UK Engineering and Physical Sciences Research Council through grants EP/R02572X/1 and EP/P017487/1. Authors would like to express their gratitude to ChipIR radiation instrument scientist Dr Christopher Frost and Dr Carlo Cazzaniga for their support.

REFERENCES

[1] G. Messenger and M. Ash, "The effects of radiation on electronic systems," 1986.

[2] J. W. Howard and D. M. Hardage, "Spacecraft Environments Interactions: Space Radiation and Its Effects on Electronic Systems," Tech. Rep.

[3] F. N. Flakus, "Radiation detection Detecting and measuring ionizing radiation-a short history," Tech. Rep.

[4] R. Trivedi, U. Mehta, and U. S. Mehta, "Article ID: IJECET_07_01_008 Cite this Article: Rakesh Trivedi and Usha S Mehta. Space Application," *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, no. 1, pp. 75–86.

[5] L. Gonella, F. Faccio, M. Silvestri, S. Gerardin, D. Pantano, V. Re, M. Manghisoni, L. Ratti, and A. Ranieri, "Total Ionizing Dose effects in 130-nm commercial CMOS technologies for HEP experiments," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 582, no. 3, pp. 750–754, dec 2007.

[6] H. J. Barnaby, "Total-ionizing-dose effects in modern CMOS technologies," in *IEEE Transactions on Nuclear Science*, vol. 53, no. 6, dec 2006, pp. 3103–3121.

[7] C. J. Dale, P. W. Marshall, E. A. Burke, G. P. Summers, and E. A. Wolicki, "High energy electron induced displacement damage in silicon," *IEEE Transactions on Nuclear Science*, vol. 35, no. 6, pp. 1208–1214, 1988.

[8] J. R. Srour, C. J. Marshall, and P. W. Marshall, "Review of displacement damage effects in silicon devices," *IEEE Transactions on Nuclear Science*, vol. 50 III, no. 3, pp. 653–670, jun 2003.

[9] Y. Liu, W. Chen, C. He, C. Su, C. Wang, X. Jin, J. Li, and Y. Xue, "Analysis of displacement damage effects on bipolar transistors irradiated by spallation neutrons," *Chinese Physics B*, vol. 28, no. 6, p. 067302, 2019.

[10] E. Stassinopoulos and J. Raymond, "The space radiation environment for electronics," *Proceedings of the IEEE*, no. 11, pp. 1423–1442.

[11] G. C. Messenger and M. S. Ash, *The effects of radiation on electronic systems*. Van Nostrand Reinhold Co.

[12] S. Saha, S. Ehsan, A. Stoica, R. Stolkin, and K. McDonald-Maier, "Real-Time Application Processing for FPGA-Based Resilient Embedded Systems in Harsh Environments," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2018*. Institute of Electrical and Electronics Engineers Inc., nov 2018, pp. 299–304.

[13] S. Li, Z. Huang, L. Han, and C. Jiang, in *Concurrency Computation*.

[14] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments," *Progress in Nuclear Energy*, vol. 111, pp. 109–124, 2019.

[15] E. Potemkin, P. Astafurov, A. Osipov, M. Malenkov, V. Mishkinyuk, and P. Sologub, "Remote-controlled robots for repair and recovery in the zones of high radiation levels," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*. IEEE, 1992, pp. 80–82.

[16] N. S. Kim, S. C. Draper, S. T. Zhou, S. Katariya, H. R. Ghasemi, and T. Park, "Analyzing the impact of joint optimization of cell size, redundancy, and ECC on low-voltage SRAM array total area," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, no. 12, pp. 2333–2337, dec.

[17] Po-Yuan Chen, Chin-Lung Su, Chao-Hsun Chen, and Cheng-Wen Wu, "Generalization of an Enhanced ECC Methodology for Low Power PSRAM," *IEEE Transactions on Computers*, vol. 62, no. 7, pp. 1318–1331, jul 2013.

[18] E. Petritoli and F. Leccese, "Reliability andSEE mitigation in memories for space applications," in *2016 IEEE Metrology for Aerospace (MetroAeroSpace)*. IEEE, jun, pp. 136–140.

[19] S. Bianchi, R. Paggi, G. L. Mariotti, and F. Leccese, "Why and when must the preventive maintenance be performed?" in *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*. IEEE, may, pp. 221–226.

[20] S. Golshan, H. Kooti, and E. Bozorgzadeh, "Seu-aware high-level data path synthesis and layout generation on sram-based fpgas," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 6, pp. 829–840, 2011.

[21] M. R. Rohanipoor, B. Ghavami, and M. Raji, "Improving combinational circuit reliability against multiple event transients via a partition and restructuring approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[22] K. S. Morgan, D. L. McMurtrey, B. H. Pratt, and M. J. Wirthlin, "A Comparison of TMR With Alternative Fault-Tolerant Design Techniques for FPGAs," *IEEE Transactions on Nuclear Science*, no. 6, pp. 2065–2072, dec.

[23] J. Yao, W. Shaojun, M. Ning, and P. Yu, "A SEU test and simulation method for Zynq BRAM and flip-flops," in *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*. IEEE, oct, pp. 1–5.

[24] M. Franklin and K. Saluja, "Pattern sensitive fault testing of RAMs with built-in ECC," in *[1991] Digest of Papers. Fault-Tolerant Computing: The Twenty-First International Symposium*. IEEE Comput. Soc. Press, pp. 385–392.

[25] J. Yamada, T. Mano, J. Inoue, S. Nakajima, and T. Matsuda, "A submicron 1 Mbit dynamic RAM with a 4-bit-at-a-time built-in ECC circuit," *IEEE Journal of Solid-State Circuits*, no. 5, pp. 627–633, oct.

[26] T. Yamada, H. Kotani, J. Matsushima, and M. Inoue, "A 4-Mbit DRAM with 16-bit concurrent ECC," *IEEE Journal of Solid-State Circuits*, no. 1, pp. 20–26, feb.

[27] J. Yamada, "Selector-line merged built-in ECC technique for DRAMs," *IEEE Journal of Solid-State Circuits*, no. 5, pp. 868–873, oct.

[28] J. Hong, J. Kim, S. Han, and E.-Y. Chung, "A locality-aware compression scheme for highly reliable embedded systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 3, pp. 453–465, 2018.

[29] P. Papavramidou and M. Nicolaidis, "Iterative diagnosis approach for ecc-based memory repair," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[30] I. Herrera-Alzu and M. Lopez-Vallejo, "Design Techniques for Xilinx Virtex FPGA Configuration Memory Scrubbers," *IEEE Transactions on Nuclear Science*, no. 1, pp. 376–385, feb.

[31] T. Jiang, P. Huang, and K. Zhou, "Scrub Unleveling: Achieving High Data Reliability at Low Scrubbing Cost," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, mar, pp. 1403–1408.

[32] N. Kim and K. Choi, "A design guideline for volatile STT-RAM with ECC and scrubbing," in *ISOCC 2015 - International SoC Design Conference: SoC for Internet of Everything (IoE)*. IEEE, nov, pp. 29–30.

[33] G. Mayuga, Y. Sato, and M. Inoue, "Highly reliable memory architecture using adaptive combination of proactive aging-aware in-field self-repair and ecc," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[34] W. Jang, "Error-correcting code aware memory subsystem," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 11, pp. 1706–1717, 2014.

[35] M. Kumar, D. Digdarsini, N. Misra, and T. V. Ram, "SEU mitigation of Rad-Tolerant Xilinx FPGA using external scrubbing for geostationary mission," in *2017 4th International Conference on Signal Processing and Integrated Networks, SPIN 2017*. IEEE, feb, pp. 414–418.

[36] C. Cazzaniga and C. D. Frost, "Progress of the Scientific Commissioning of a fast neutron beamline for Chip Irradiation," in *Journal of Physics: Conference Series*, 2018.