2021 24th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS) | 978-1-6654-3595-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/DDECS52668.2021.9417059

A Model-Based Framework to Assess the Reliability of Safety-Critical Applications

Lucas Matana Luza*, Annachiara Ruospo†, Alberto Bosio‡, Ernesto Sanchez† and Luigi Dilillo*§
*University of Montpellier, LIRMM, Montpellier, France. Email: {lucas.matana-luza, dilillo}@lirmm.fr
†Politecnico di Torino, Torino, Italy. Email: {annachiara.ruospo, ernesto.sanchez}@polito.it
‡Univ Lyon, ECL, INSA Lyon, CNRS, UCBL, CPE Lyon, INL, UMR5270, France. Email: alberto.bosio@ec-lyon.fr
§Centre National de la Recherche Scientifique (CNRS), Paris, France

***Abstract*—Solutions based on artificial intelligence and brain-inspired computations like Artificial Neural Networks (ANNs) are suited to deal with the growing computational complexity required by state-of-the-art electronic devices. Many applications that are being deployed using these computational models are considered safety-critical (e.g., self-driving cars), producing a pressing need to evaluate their reliability. Besides, state-of-the-art ANNs require significant memory resources to store their parameters (e.g., weights, activation values), which goes outside the possibility of many resource-constrained embedded systems. In this light, Approximate Computing (AxC) has become a significant field of research to improve memory footprint, speed, and energy consumption in embedded and high-performance systems. The use of AxC can significantly reduce the cost of ANN implementations, but it may also reduce the inherent resiliency of this kind of application. On this scope, reliability assessments are carried out by performing fault injection test campaigns. The intent of the paper is to propose a framework that, relying on the results of radiation tests in Commercial-Off-The-Shelf (COTS) devices, is able to assess the reliability of a given application. To this end, a set of different radiation-induced errors in COTS memories is presented. Upon these, specific fault models are extracted to drive emulation-based fault injections.**

***Index Terms*—safety-critical applications, reliability, approximate computing, neural network, fault injection**

I. Introduction

Today, a wide variety of application domains requires smarter electronic devices and high-performance computing systems. Solutions based on artificial intelligence and brain-inspired computations like Artificial Neural Networks (ANNs) are suited to cope with this growing computational complexity. For their outstanding computational capabilities, ANNs are being increasingly deployed in many application domains. Many of these are considered safety-critical due to the gravity that a failure could cause. One peculiar characteristic of these brain-inspired models is their intrinsic robustness. Despite the claimed built-in robustness of ANNs [1], there is a pressing need for evaluating their reliability, especially if they are deployed on resource-constrained hardware devices for safety-critical applications. More precisely, it is essential to evaluate whether a hardware or software fault may cause a system failure. In parallel with reliability assessment, a lot of effort has been made to reduce the memory and energy footprint of ANNs, for instance, by leveraging on reduced bit-width data type in either training or inference phase. It is crucial to cope with the emerging trend towards deploying ANNs on resource-constrained hardware implementations that are fast and ultra-low-power, optimised for solving specific tasks. This contrasts with the trend that sees the growing complexity of the state-of-the-art ANNs that can require some kB or MB of memory for storing their parameters. In this light, Approximate Computing (AxC) has become a significant field of research to improve memory footprint, speed, and energy consumption in embedded and high-performance systems [2]. By relaxing the need for fully precise or completely deterministic operations, AxC substantially improves energy efficiency and reduces the memory requirement. The use of AxC can significantly reduce the cost of ANN implementations, but it may also reduce the inherent resiliency of this kind of application, which can be more prone to errors due to external perturbation (e.g., radiation harsh environment like space). The radiation-induced effects may be amplified by the AxC techniques, which can result in unacceptable outputs. On this scope, reliability assessments are carried out by performing fault injection campaigns. Depending on how faults are injected and at which abstraction level, several methodologies can be drawn. The most precise but also costly in terms of hardware resources and exposure time is radiation tests.

The main intent of the paper is to bring a framework to enable the reuse of results from radiation test campaigns in Commercial-Off-The-Shelf (COTS) devices. The reuse of the results is reached with a detailed evaluation of the radiation-induced faults. It is possible to define realistic fault models used to assess the reliability of a given application through a model-based fault injection when emulating its functionality. The ultimate goal is to use the proposed framework for quick but still accurate reliability estimation of approximate ANNs to be deployed in resource-constrained hardware devices for safety-critical applications.

The rest of the paper is organised as follows. Section II presents different AxC techniques and related works. Section III presents the proposed framework by targeting the

This study has been achieved thanks to the financial support of the VAN ALLEN Foundation (Contract No. UM 181387) and the Region Occitanie (Contract No. UM 181386), and by the projects "IDEX Lyon OdeLe" and "ReACT".

978-1-6654-3595-6/21/$31.00 ©2021 IEEE

transition from a radiation-based fault-injection to a model-based one using realistic fault models extracted from the radiation test campaigns. Finally, Section IV concludes the paper.

II. Approximate Computing Techniques

AxC techniques provide a set of design choices for a performance-accuracy trade-off evaluation, bringing the possibility to enhance, for example, speed, energy efficiency, and memory usage, at the cost of reducing the accuracy constraints at different levels of abstractions [3]. Some techniques are explored in the following subsections.

A. Voltage and Frequency Scaling

Voltage scaling is a technique that aims at reducing a supply voltage level at the cost of impacting the computation timing of processing blocks or even affecting the accuracy of the final result of an application [4]. In [5], by exploring the inherent noise resilience of binary neural networks and over scaling the supply voltage of the memories used in the implemented system-on-chip (SoC) at the cost of 1% of classification accuracy degradation. Furthermore, the voltage scaling may reduce the voltage threshold that is used to define the logical '1' or '0', which can result in data cells becoming stuck at a logical value or even increasing the probability of soft error occurrence, leading to a degradation in the hardware reliability and even affecting the data precision [6].

Besides the clock frequency scaling that can be applied alongside the supply voltage scaling (as shown with the DVFS technique), another approach is reducing the execution rate of systematic operations. In [7], this technique is explored by means of reducing the refresh frequency of an embedded DRAM (eDRAM) implemented in a SoC. In [8], authors propose the use of sub-optimal refresh rates on DRAM by applying a quality-aware approximate system, where, based on characterisations testes, the critical data is allocated in physical pages where the number of errors generated by the sub-optimal refresh rates is low.

B. Data Precision Reduction

The memory footprint can be directly reduced by changing the data representation of the parameters (e.g., weights, activations) of an ANN implementation. Reducing memory footprint can lead to a reduction in the energy consumption of the implementation since there is a decrease in the amount of data transferred from/to the memory [9]. Methods of reducing the floating-point precision, or even the bit-width used for data representation, can significantly reduce the energy consumption with a cost of degradation in the outcomes of an application [10]. In [11], the authors present a framework to integrate the use of fixed-point data representation and a reduction in the floating-point data representation to improve the energy efficiency in mobile GPUs, reaching a reduction of about 30% with acceptable degradation in the rendered images. Reducing the bit-width of the adopted data representation can also improve the robustness of the neural network, as demonstrated in [12]. The implementation of these techniques are not restricted to software-based implementations but also can be implemented at the hardware level and directly in FPGA (Field Programmable Gate Array) applications. This type of improvement on hardware projects does not directly impact the software level positively. From one side, with FPGA implementations, area and energy costs can be reduced by implementing mathematical functions (e.g., logarithms). However, it can increase the application execution time at the software level, in which the data handling and operations are implemented [13].

III. Proposed Framework

We propose a framework to assess the reliability of AxC on ANN-based applications, enabling the trade-off estimation between the impact of radiation-induced faults and resource usage in terms of hardware and software on these applications. The proposed framework enables the use of fault models extracted from the characterisation of COTS memories in conjunction with the target environment conditions (e.g., radiation type, flux, and dose) to analyse the different scenarios regarding the reliability of the system. The idea is based on three steps: radiation-based fault injection, fault model evaluation, model-based fault injection.

A. Radiation-Based Fault Injection

Radiation-based fault injections aim at characterising a device or application and validate predictive models of events produced by the impact of ionising particles (e.g., neutrons, alphas, heavy ions, protons) on a given electronic device. The reliability assessment is carried out on the actual platform during the radiation test campaign.

One common outcome is the evaluation of Single-Bit Upsets (SBUs) due to the radiation effects. Also, Single-Event Latch-Up (SEL) may occur in the device. This kind of event can lead to permanent damage, affecting not only the memory array but also its controller part. The SEL can be observable by a sudden increase in the nominal supply current [14].

Moreover, in [15], the authors spotted that the retention time of a COTS DRAM memories has a significant decrease when exposed to ion irradiation. This characteristic plays an important role when scaling the refresh rate once the retention capability of the cells will decrease, and the downscaled refresh rate will be not reliable when the system is exposed to a harsh environment containing this type of particle.

Furthermore, supplementary to the device characterisation, the reliability assessment directly in the application can be explored by radiation test at both hardware and software level. Several works have been published on this scope. In [16], the authors studied the radiation-induced soft-errors based on AxC techniques applied to the data representation of weights in a Convolutional Neural Network (CNN). Withal, by using an accelerated neutron beam to inject transient errors and fault injection experiments for permanent errors, the reliability of a 54-layer CNN is assessed by exposing the entire GPU to the radiation source [17].

B. Fault Model Evaluation

Radiation-based fault injection exposed the system to similar conditions of an in-field application, e.g., space applications. From the radiation tests, it is possible to evaluate the electronic device by identifying the types of the generated faults and their frequency and appearance conditions. Based on the research group activities, several types of memories were already exposed to different radiation sources. In [18], the authors review the techniques and results of several radiation test campaigns on two commercial SRAMs (90 nm and 65 nm) technology nodes. Moreover, the effects on a SLC (Single Level Cell) NAND Flash under heavy-ions and proton irradiation were evaluated in [19]. Also, in [20], several tests on a MRAM show that the memory is prone to suffering from SELs. Studies on these memories show that SEEs can occur in different ways, such as SBUs, Multiple Cell Upsets (MCUs), Single-Event Functional Interruptions (SEFIs), or SELs, and it generates different kinds of fault behaviours on the devices.

From recognizing the different types of errors and identifying the occurrences of each one, the estimated event cross section provides the probability of its occurrences in a given scenario. The cross section is defined as

$$\sigma_{\text{bit}} = \frac{N}{F \times M} \tag{1}$$

where N is the number of events, F is the beam fluence in n/cm^2, and M is the number of bits [21].

C. Model-Based Fault Injection

Based on the radiation-induced soft errors, the model-based fault injection is proposed as a case study, where it is shown the main ideas to enable the injection of realistic fault models during the simulation or emulation of a target application. In order to define the scope, the next steps are based on neutron irradiation on a COTS self-refresh DRAM memory (results can be explored in more details in [22]), and the reliability assessment is based on the injection of realistic faults extract from this test campaign in applications that may use this memory to run. Then, starting from the radiation test campaigns, based on the results presented in [22], it was defined the cross sections for three different types of errors that were identified: SBUs, stuck-at bits, and block errors. From the estimated cross sections, we can define the E-SER (Execution Soft Error Rate), which provides the expected amount of events within a run of the defined application. The equation is:

$$E\text{-}SER = \sigma_{\text{bit}} \times M \times \bar{\phi} \times t \tag{2}$$

where σ_{bit} is the calculated cross section (cm^2/bit), M is the memory size in bits used by the application (stored in the target), $\bar{\phi}$ is the average neutron flux (n/cm^2/s) of the target environment, and t is the application execution time in seconds.

A case study is proposed by using the CNN LeNet-5 [23] as the application target. The trained network was exported as C code using three different data representations for the weights: the accurate network having 32-bit floating-point real numbers (*Float 32*), and two approximated versions: a 16-bit integer quantised CNN (*Int 16*), and an 8-bit integer quantised CNN (*Int 8*). The three replicas were then ported to an embedded system using an SoC, which provides an ARM Cortex™A9 processor attached to a 28 nm Artix®7 FPGA. This platform was then implemented to have the HyperRAM memory allocating the weights of the CNN. From this implementation, we extract two parameters used by the E-SER equation. The allocated resources in the HyperRAM memory gives us the M. Furthermore, by running the application for the determined quantity of inferences (1,000 images), we extracted t. The estimated E-SER for the three different versions were calculated with the cross sections defined in [22], with an average flux of 5×10^6 n/cm^2/s. The memory usage was approximately 4 Mb for *Float 32*, 2 Mb for *Int 16* and 1 Mb for *Int 8* with respect to the network weights. The execution time was approximately 1800 s for *Float 32*, 960 s for *Int 16*, and 600 s for *Int 8*. The estimated E-SER are presented in Tab. I.

TABLE I
ESTIMATED EVENT RATE FOR THE THREE DIFFERENT CNN VERSIONS.

CNN	**E-SER [events/run]**		
version	**SBUs**	**Stuck-at bits**	**Block errors**
Float 32	1.04	0.53	0.40
Int 16	0.30	0.15	0.21
Int 8	0.10	0.05	0.13

As explored in the previous sections, the radiation-induced errors in memories can be seen in different ways, from just SBUs up to errors spanning a significant range of addresses. A model-based fault injection must take into account its differences. From the three types of errors that were identified in [22], the SBUs are the simplest ones to be injected on a target application since they can be injected on random addresses of the target memory accordingly the calculated E-SER. Stuck-at faults also occurred in random addresses, but two characteristics should be taken into account for this type of errors: it presented an intermittent behaviour that should be explored by defining a behavioural equation based on the number of occurrences for each bit address, and the injected faults may be permanent and not recoverable. Finally, block errors have shown normally a faulty address pattern when analysed by logical bitmaps. Then, the block error model should take into account its shape and size, as well as its behaviour, once write operations is generally capable of restoring the cells access.

From these assumptions, an emulator can be designed based on two processes. The application process is responsible for running the target neural network, and the injector process is responsible for introducing errors in memory locations. Both processes are executed in parallel and share the same memory resources, which should be implemented with concurrency control mechanisms. The E-SER and the fault models should be provided, and the injector process will use these inputs to determine where and when a fault should be injected into the running application. Fig. 1 depicts a top-level diagram of the emulator.

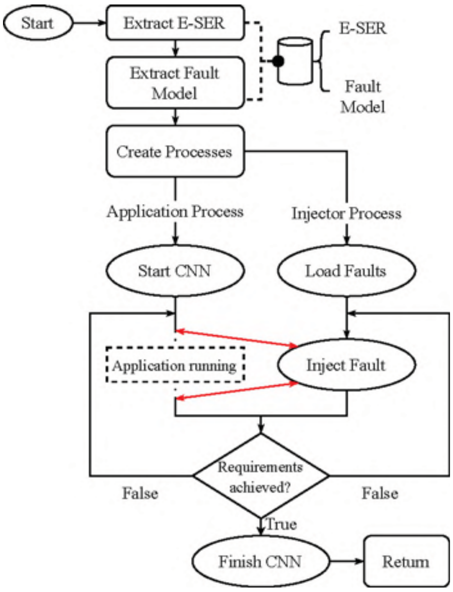

Fig. 1. Emulator top level diagram.

IV. Conclusion

In this work, we presented a framework to assess the reliability of approximate computing applications. The framework leverages the results from radiation tests in COTS devices to assess a given application's reliability. We presented radiation-induced errors targeting COTS memories, which can be just SBUs and stuck-at bits, including events that generate errors spanning a significant range of addresses. This paper also highlighted that the errors could appear in different manners and with different behaviours. Finally, it is presented an emulator architecture aimed at injecting the extracted fault models into a specific application. We explored the use of previous results to enable the reliability assessment by proposing a case study. In the future, the group's target aims to use the framework to deploy an emulator capable of handling and injecting the defined fault models. They will be used to estimate the reliability of approximate ANNs.

References

[1] C. Torres-Huitzil and B. Girau, "Fault and error tolerance in neural networks: A review," *IEEE Access*, vol. 5, pp. 17 322–17 341, 2017, doi: 10.1109/ACCESS.2017.2742698.

[2] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 62:1–62:33, Mar. 2016. [Online]. Available: http://doi.acm.org/10.1145/2893356

[3] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*, May 2013, pp. 1–6, doi: 10.1109/ETS.2013.6569370.

[4] V. K. Chippa, D. Mohapatra, K. Roy, S. T. Chakradhar, and A. Raghunathan, "Scalable effort hardware design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 2004–2016, 2014, doi: 10.1109/TVLSI.2013.2276759.

[5] A. D. Mauro, F. Conti, P. D. Schiavone, D. Rossi, and L. Benini, "Always-on 674μW@4GOP/s error resilient binary neural networks with aggressive SRAM voltage scaling on a 22-nm IoT end-node," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 11, pp. 3905–3918, 2020, doi: 10.1109/TCSI.2020.3012576.

[6] V. Chandra and R. Aitken, "Impact of technology and voltage scaling on the soft error susceptibility in nanoscale CMOS," in *2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, 2008, pp. 114–122, doi: 10.1109/DFT.2008.50.

[7] S. Ganapathy, A. Teman, R. Giterman, A. Burg, and G. Karakonstantis, "Approximate computing with unreliable dynamic memories," in *2015 IEEE 13th International New Circuits and Systems Conference (NEWCAS)*, 2015, pp. 1–4, doi: 10.1109/NEWCAS.2015.7182027.

[8] A. Raha, H. Jayakumar, S. Sutar, and V. Raghunathan, "Quality-aware data allocation in approximate DRAM*," in *2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, 2015, pp. 89–98, doi: 10.1109/CASES.2015.7324549.

[9] G. Rodrigues, F. Lima Kastensmidt, and A. Bosio, "Survey on approximate computing and its intrinsic fault tolerance," *Electronics*, vol. 9, no. 4, p. 557, 2020, doi: 10.3390/electronics9040557.

[10] C. Rubio-González *et al.*, "Precimonious: Tuning assistant for floating-point precision," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, Nov. 2013, pp. 1–12, doi: 10.1145/2503210.2503296.

[11] C. C. Hsiao, S. L. Chu, and C. Y. Chen, "Energy-aware hybrid precision selection framework for mobile GPUs," *Computers & Graphics*, vol. 37, no. 5, pp. 431–444, 2013, doi: 10.1016/j.cag.2013.03.003.

[12] A. Ruospo, A. Bosio, A. Ianne, and E. Sanchez, "Evaluating convolutional neural networks reliability depending on their data representation," in *2020 23rd Euromicro Conference on Digital System Design (DSD)*, 2020, pp. 672–679, doi: 10.1109/DSD51259.2020.00109.

[13] J. G. Pandey, A. Karmakar, C. Shekhar, and S. Gurunarayanan, "An FPGA-based fixed-point architecture for binary logarithmic computation," in *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Dec. 2013, pp. 383–388, doi: 10.1109/ICIIP.2013.6707620.

[14] R. Gaillard, "Single event effects: Mechanisms and classification," in *Soft Errors in Modern Electronic Systems*, M. Nicolaidis, Ed. Springer US, 2011, pp. 27–54, doi: 10.1007/978-1-4419-6993-4.

[15] D. Söderström *et al.*, "Stuck and weakened SDRAM cells due to heavy-ion irradiation," in *2019 30th anniversary European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, Sept. 2019, pp. 1–4.

[16] L. Matana Luza *et al.*, "Investigating the impact of radiation-induced soft errors on the reliability of approximate computing systems," in *2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, Oct. 2020, pp. 1–6, doi: 10.1109/DFT50435.2020.9250865.

[17] A. Lotfi *et al.*, "Resiliency of automotive object detection networks on GPU architectures," in *2019 IEEE International Test Conference (ITC)*, Nov. 2019, pp. 1–9, doi: 10.1109/ITC44170.2019.9000150.

[18] L. Dilillo, G. Tsiligiannis, V. Gupta, A. Bosser, F. Saigne, and F. Wrobel, "Soft errors in commercial off-the-shelf static random access memories," *Semiconductor Science and Technology*, vol. 32, no. 1, p. 013006, 2016, doi: 10.1088/1361-6641/32/1/013006.

[19] L. M. Luza, A. Bosser, V. Gupta, A. Javanainen, A. Mohammadzadeh, and L. Dilillo, "Effects of heavy ion and proton irradiation on a SLC NAND Flash memory," in *2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2019, pp. 1–6, doi: 10.1109/DFT.2019.8875475.

[20] A. L. Bosser, "Single-event effects from space and atmospheric radiation in memory components," Ph.D. dissertation, Université de Montpellier and Jyväskylän yliopisto, Montpellier, Dec. 2017, available: http://www.theses.fr/2017MONTS085.

[21] E. Petersen, *Single Event Effects in Aerospace*. John Wiley & Sons, 2011, doi: 10.1002/9781118084328.

[22] L. Matana Luza *et al.*, "Effects of thermal neutron irradiation on a self-refresh DRAM," in *IEEE 15th International Conference on Design & Technology of Integrated Systems in Nanoscale Era*, Apr. 2020, pp. 1–6, doi: 10.1109/DTIS48698.2020.9080918.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998, doi: 10.1109/5.726791.