This is the author's final, peer-reviewed manuscript as accepted for publication (AAM). The version presented here may differ from the published version, or version of record, available through the publisher's website. This version does not track changes, errata, or withdrawals on the publisher's site.

Evaluating and mitigating neutrons effects on COTS EdgeAI accelerators

Sebastian Blower, Paolo Rech, Carlo Cazzaniga, Maria Kastriotou, Christopher D. Frost

Published version information

Citation: S Blower et al. 'Evaluating and mitigating neutrons effects on COTS EdgeAI accelerators'. IEEE Trans Nucl Sci vol. 68, no. 8 (2021): 1719-1726.

DOI: 10.1109/TNS.2021.3086686

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This version is made available in accordance with publisher policies. Please cite only the published version using the reference above. This is the citation assigned by the publisher at the time of issuing the AAM. Please check the publisher's website for any updates.

This item was retrieved from **ePubs**, the Open Access archive of the Science and Technology Facilities Council, UK. Please contact epublications@stfc.ac.uk or go to http://epubs.stfc.ac.uk/ for further information and policies.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

Evaluating and Mitigating Neutrons Effects on COTS EdgeAI Accelerators

Sebastian Blower[1], Paolo Rech[2], Carlo Cazzaniga[1], Maria Kastriotou[1], Christopher D. Frost[1]
1)ISIS Facility, STFC, Rutherford Appleton Laboratory, Didcot, OX11 0QX, UK
2)Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil and Politecnico di Torino, Italy

***Abstract*—EdgeAI is an emerging AI accelerator technology which is capable of delivering improved AI performance at both a lower cost and a lower power level. With the aim of implementation in large quantities and in safety-critical environments, it is imperative to understand how Single Event Effects (SEE) affect the reliability of this new family of devices, and to propose efficient hardening solutions. Through neutron beam experiments and fault-injection analysis of a COTS EdgeAI device, we are able to identify the device's SEE failure-modes, separate the error rate contributions of the device's different resources, and characterise the device's SEE reliability. During this analysis we discovered that the vast majority of single bit flips have no appreciable effect on the output. After this analysis, we propose a hardening solution which implements TMR in the device without changing its physical architecture. We experimentally validate this solution and show that we are able to correct 96% of the misclassifications (critical errors) with nearly-zero overhead.**

***Index Terms*—Artificial neural networks, embedded systems, fault tolerance, reliability, safety-critical**

I. Introduction

ARTIFICIAL Neural Networks (ANNs) are today adopted in various domains that range from high performance computing to data analysis and autonomous vehicles for automotive and aerospace applications. While general purpose devices such as GPUs or FPGAs are widely used to prototype ANNs, this new programming paradigm has pushed the demand for dedicated, efficient, and low-cost Commercial-Off-The-Shelf (COTS) devices for the execution and acceleration of ANNs. These emerging devices, named *EdgeAI*, bring the once computationally expensive process of decision making via a neural network down to a lower cost and a lower power level (the NeuroShield discussed here consumes 0.38W under load, the Quadro GV100 GPU consumes 250W [1]).

The huge computing power of high-end GPUs is required to solve highly complex problems, such as detecting objects in a scene. If adopted in self-driving cars, these problems need to be solved in real-time (i.e., at least 40 frames per second), forcing the use of highly expensive, power hungry parallel devices. Besides costs, the huge number of available resources in GPUs makes them highly susceptible to faults [5]. The lower computing capabilities of EdgeAI devices limit the number of object classes that can be identified and detection (intended as the location in space of the object) is normally not supported. While unsuitable to real-time object detection, Edge AI devices have a range of applications: classification, anomaly detection, and noise removal, and are currently extremely popular in Internet of Things (IoT) applications [2][3]. The expected high distribution of IoT devices increases the probability of seeing a radiation-induced corruption. It is then fundamental to evaluate EdgeAI device's Failure In Time, and eventually propose effective hardening solutions to improve their reliability if the error rate is found to be too high.

EdgeAI devices usually embed a general-purpose CPU (typically ARM based), that acts as an interface, and a programmable accelerator that implements the neural network which is defined and trained by the user. EdgeAI devices are a very attractive alternative for the execution of ANNs when compared to the expensive and/or power-hungry GPUs and FPGAs; especially for applications with very limited power requirements. Additionally, FPGAs and GPUs have previously been shown to have serious reliability weaknesses, mostly due to their inefficient mapping of ANNs [4][5]. As dedicated accelerators are likely to be considered for the execution of ANNs in safety-critical applications [6], it is mandatory to deeply investigate their reliability, identify eventual weaknesses and failure modes, and propose efficient hardening solutions.

This work aims to characterize Edge AI device reliability using the COTS NeuroShield as a case study. The NeuroShield is a 576 neurons accelerator, that can be configured (through learning) to identify several patterns [8][9]. The flexibility of the architecture allows the user to perform single or multi layers perceptions connecting the neutrons in the device or even connecting various devices in a chain. For our testing we only implement a single layer perceptron. While the results we present are focused on a specific accelerator, the methodology we adopted remains valid for any EdgeAI device. Through fault-injection, we investigated how errors propagate through the device. Then, through neutron beam experiments we measured the FIT rates of an ANN running on the NeuroShield and separated the error rate contributions of its different resources. Finally, based on our analyses, we propose a hardening method which increases the hamming distance of the dataset categories using Triple Modular Redundancy (TMR). We experimentally validate our solution and show that we correct 96% of the misclassifications (critical errors) with nearly-zero overhead.

After a brief background on Neural Network reliability and EdgeAI architectures (Section II), we discuss the experimental setup (Section III and IV) and how the cross-section of the NeuroShield was determined (Section V). We then present error

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

rate estimation results (Section VI), propose an efficient hardening solution for EdgeAI devices (Section VII), and evaluate it (Section VIII). Section IX then draws conclusions.

II. Background

A. Neutron Effects in Neural Networks

On an ANN accelerator affected by radiation-induced faults, you would expect to see the following effects:

- *No effect observed* on the output (the fault is masked).
- *Silent Data Corruption* (SDC) that modifies the network output. SDC does not necessarily affect the accuracy as some corruptions can lead to correct classifications when the corrupted output has a value which is sufficiently close to the correct one.
- *Program / device crash* requiring reboot or power cycle.

The probability of events occurring as well as the overall impact of the event on the ANN is dependent on the implementation. As an example, GPU accelerated Convolutional Neural Networks (CNN) can see their reliability significantly impacted by neutron effects; even when Error-Correcting Code (ECC) is used. This is due to the GPU's microarchitecture having a tendency to propagate single faults, affecting several output elements [5]. On SRAM FPGAs, errors in the configuration memory have been shown to alter the implemented circuit, leading to misdetection [4]. On an EdgeAI device such as IBM's TrueNorth, it has been shown that although overall accuracy remains roughly constant, categories can show an increase or decrease in accuracy [7]. Our work will look for further effects in Edge AI devices and propose and validate a hardening solution.

B. NM500 Architecture and Neutron Effects

The Device Under Test (DUT) in this work is an Arduino shield (developed by General Vision, produced by nepes) known as the NeuroShield [8][9]. This device implements a singular NM500 chip, an Edge AI Accelerator. In our tests, the chip contained a single layer neural network of up to 576 neuron cells. Each cell consists of 6 registers, to control behaviour, several logic gates, and a block of 256-byte, 8-bit memory to store a model [10]. The network's behaviour is controlled by 15 Network Control Registers (NWCRs). These registers control the process of learning new patterns, such as when and how to learn. The NWCRs also allow the classification method to be switched between *Radial Basis Function* (which allows the network to return an uncertain classification) and *K-Nearest Neighbours* (which will always return a classification) [11].

The NM500 classifies input vectors in the following way. An input vector is fed into each neuron simultaneously. This input passes through various logic blocks within each neuron, Fig. 1. This logic compares the input against a unique internal model to produce a value known as *distance*, the summation of the bitwise differences between an input vector and the neurons internal model. This can be thought of as the *Confidence of the Classifying Neuron* and will be referred to in this way for the remainder of the paper. Once the neurons have processed the input, the network logic identifies a singular neuron with the highest level of confidence in its classification and propagates that classification to the output. The selected neuron is known as the *Firing Neuron.* This process is shown in Fig. 2.

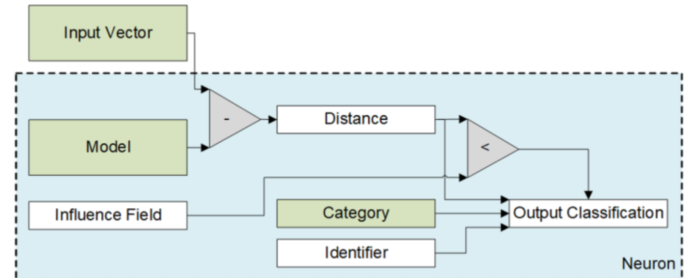

Fig. 1 A representation of the logic within each of the 576 neurons within the NM500 chip. A neuron produces a classification based on the distance (difference between the input and the internal model), normalized with the influence field.

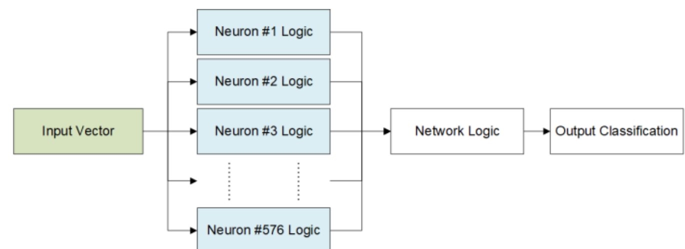

Fig. 2. A representation of the NM500's architecture showing the data flow.

In total, there are approximately 1,212,000 bits in the NM500 chip that can be corrupted, calculated from the NM500 User Manual [10].

One of the contributions of our reliability evaluation is to understand, with controlled fault-injection, the possible sources for errors (Section IV-D) and to propose effective hardening solutions. Due to the architecture of the device, in fact, some of these bits are more influential in their effects on the ANN than others. The architecture itself of EdgeAI devices, then, is likely to be a promising target for efficient (and effective) hardening. For instance, a bit flip in a neuron model would have an almost unnoticeable effect, as it would only cause a small change to the confidence of the classifying neuron [12]. A bit flip in the NWCRs FORGET register, on the contrary, could have much more disastrous consequences. In fact, if the network was told to forget its knowledge, every neuron's category would be set to 0, and no inputs would be classified until the network is re-trained. Another area with a potentially significant effect on the output is the category register of each neuron. All 576 neurons contain one of these 16-bit registers which stores a number which eventually makes up the classification output of the network. A SEU here would result in the neuron reporting an incorrect category for all classifications it is involved in. It is within this register that the hardening method proposed in Section VII focused its attention.

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

III. Datasets Used in Experiments

Datasets are large collections of pre-classified input patterns (commonly images) which are used for training and testing a neural network. A dataset commonly used for experimenting with neural networks is MNIST, a collection of 70,000 images of handwritten digits from 0 to 9 [13]. Datasets are also usually split into a training set and a testing set, MNIST is split into 60,000 training images, and 10,000 testing images. The work presented here uses the full MNIST training set for all training applications and a randomly obtained subset of the MNIST testing set for testing applications.

We have selected a reduced dataset to decrease the time required to process all the images. To process the full data set of 10,000 images it would take over 30 seconds, while 2 seconds are sufficient to process the reduced dataset of 600 images. The data set reduction is necessary to:

1. Decrease the chance of multiple neutrons causing corruption during the processing of the dataset.
2. Increase the number of tests performed.

The subset of images was selected randomly while guaranteeing an equal number of images for each category. The selected subset of images has practically the same accuracy of the full MNIST dataset (79% vs 80%).

A. *Modifying the Datasets*

Due to the NM500's architecture, the maximum input length is 256 bytes. However, the MNIST dataset contains images of 784 bytes in length. By down sampling the MNIST images with a mean-box-filtering technique [14], the images were reduced to 256 bytes, making them compatible with the device. Since this approach is used to reduce computational complexity even in powerful ANN devices [15], we claim that this is still an effective test case scenario. The accuracy of the network with the down sampled dataset was approximately 80%. A comparison between one of the original and downscaled images is shown in Fig. 3.

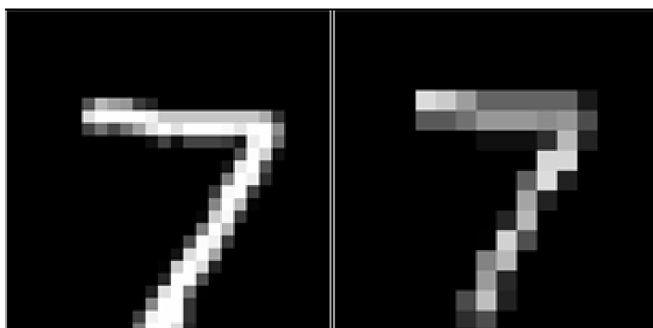

Fig. 3. Comparison of original 28x28 pixel MNIST image (left) to its down-scaled 16x16 pixel counterpart generated with *Mean Box Filtering*. Down-scaled image is also an example of a NeuroShield input pattern.

IV. Experimental Setup Used at ChipIr

The setup described in this section was utilised for all the neutron radiation experiments performed in this work. The only variation in setup was the distance between the ChipIr beam-stop and the DUT due to space limitations in the beamline.

The experiments were controlled and monitored by a LabVIEW program running on a computer located in ChipIr's screened room. This computer communicated with an Arduino Due, located inside ChipIr's block house via an RS232 UART serial connection. The Arduino was responsible for data output and operation of the NeuroShield. To prevent radiation from interfering with the Arduino's functions, the NeuroShield was connected via a custom-made shield extension board. Finally, a Newport motorised linear stage controlled the radiation exposure by moving the NeuroShield, the DUT, into and out of the beam as desired. An image of the setup used at ChipIr is shown in Fig. 4.

The expected neutron flux at the ChipIr beam-stop is $(5.6 \pm 0.6)\times10^{6}$ n cm^{-2} s^{-1} (with neutron energies greater than 10 MeV [18]). The neutron flux of the beamline was characterised with calibrated activation foils [16] and silicon detectors [17], which are also used as an active monitor during the irradiation. Due to the nature of the neutron source and high energies of neutrons in use, the beam is divergent even when passed through the collimator. The best approximation of the neutron transport is a point-like source at 13.6 meters from the standard user position (ChipIr beam-stop) [18]. As a result, all measurements of neutron flux taken from the activation foils and silicon detectors positioned at the front of ChipIr must be corrected according to the inverse square law.

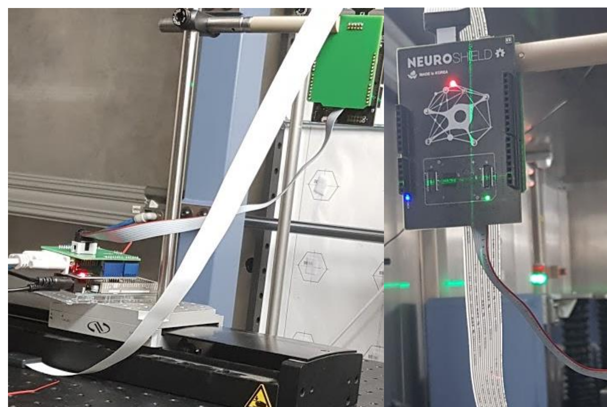

Fig. 4. The experimental setup at ChipIr, ready to be irradiated.

V. Determination of the Cross-Section

A. *Experimental Methodology*

The experimental setup presented in Section IV was utilised here. The DUT was positioned 0.8m away from the ChipIr beam-stop with a collimated beam size of 70x70mm.

The network was initialised with a previously generated knowledge file containing a snapshot of the contents of every register on the network (i.e., the network's weights). This file was generated by the Arduino after successfully training the network with the dataset described in Section III. It took approximately 2 seconds for the knowledge file to be read from an external SD card and written to the network. To ensure initial accuracy of the DUT's register contents, the process of

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

initialising the knowledge was performed outside of the neutron beam before being validated through a singular out-of-beam test. Following validation of the knowledge, the device was traversed into the path of the neutron beam for 30 seconds.

A test duration of 30 seconds was chosen because it was found through early experimentation that any longer than a minute would cause errors to begin to saturate and increased the likelihood of observing a knowledge failure error (mass bit corruption due to error induced in network control logic). By using 30 seconds, it enable us to maintain a usable error throughput whilst minimising the likelihood of a neutron striking the same bit more than once. After the time elapsed, the device was traversed out of the path of the beam. The now corrupted contents of the network were then output via a serial link to the external monitoring computer running the LabVIEW program, where it was logged ready for analysis.

For calculating the total number of bit flips induced in the device, the output after exposure was compared to a control output with no exposure. The difference between these two outputs became the number of errors, Ne. The expected neutron flux at the ChipIr beam-stop is $(5.6 \pm 0.6)\times10^{6}$ n cm^{-2} s^{-1} (with neutron energies greater than 10 MeV [18]). To account for the divergence of the flux, the flux was adjusted according to the inverse square law to give an expected neutron flux at the DUT of $(5.0 \pm 0.54)\times10^{6}$ n cm^{-2} s^{-1}. The particle fluence was then calculated by multiplying the expected neutron flux at the DUT by the time of exposure, 30 seconds. Finally, the cross-section of the device for a singular test was calculated using (1). Where σ is the per-bit cross-section, N_e is the number of errors, and ϕ is the neutron fluence.

$$\sigma = \frac{N_e}{\phi} \tag{1}$$

B. C. Cross-Section of the NeuroShield

Over the course of the experiment, the DUT was exposed to approximately 3 hours of neutron radiation, considering a terrestrial sea-level neutron flux of 3.6×10^{-3} n cm^{-2} s^{-1} [19], this equated to approximately 533,000 device-years of atmospheric exposure. During this time, a total of 2646 bit-errors were observed. From this, the per-bit cross-section of the NeuroShield was found to be $(4.9 \pm 0.2)\times10^{-8}$ cm^{2}, giving a FIT rate of 635 single bit corruptions in 10^{9} device-hours of operation. This was found to be comparable to that of SRAMs of a similar technology node: $(4.42 \pm 1.49)\times10^{-8}$ cm^{2} [20]

VI. Determination of the Error Rates

A. Error Types

Each classification (output from the NeuroShield) is composed of three pieces of information: Category, Confidence of the Classifying Neuron, and Neuron Identification Number. When a SEU is induced, a change can be present in one or more of these three pieces of information.

Each distinct classification can thus contain one or more of the following four error types:

- *Category Error (CE)* - The category information of the classification is different than expected: that is, a misclassification has occurred. Think of the network classifying a picture of the number 2 as a number 6.
- *Distance Error (DE)* - The distance, confidence of the classifying neuron, information of the classification is different than expected: think of the network's confidence in its classification being changed.
- *Firing Neuron Error (NE)* - The classification's firing neuron is different than expected. The new neuron can either classify the input correctly or incorrectly, thus NE can be accompanied by CE and almost always by DE.
- *Knowledge Failure (KF)* - The network's knowledge has become damaged such that the input could not be classified. Note the difference between CE; KF can be thought of as an inability or refusal to classify an input.

The proportions of these errors in a test should be proportional to the number of bits responsible for causing them (each error type's critical bits). For example, bit flips in the registers which store neuron models should cause DE to be present in the output. Since neuron model bits make up 95.5% of the total bits, DE should account for 95.5% of all errors observed in testing. This section will aim to test this relationship, thus identifying which device bits are most critical for each failure mode and inform where hardening efforts should be focused for most benefit.

B. Experimental Methodology

In order to test for bit flips as-fast-as-possible, a new test set was produced from the contents of each neuron cell's model (an example of a neuron model is shown in Fig. 5; notice the similarity to the MNIST images of Fig. 3). This new test set consisted of 576 images, one for each neuron cell's model. A single test refers to the classification of all 576 of these images (576 input patterns). An error is said to occur when any part of a test is different from expected; each error can be categorised based on the four categories presented in Section VI-A. When this test was processed by the network, the NM500 will output the contents of its neurons in the most time and size efficient manner. It is the equivalent of reading each neuron register in the NM500 sequentially via the SPI interface but much quicker. The time savings occur due to the read operation making use of the parallel arrangement of the neurons. This test set has the added benefit of ensuring every single neuron in the network is examined, an important consideration for evaluating the radiation hardness of the device.

Testing consisted of two separate phases: first was extensive fault-injection analysis, a testing method in which device bits are flipped artificially to simulate the impact of SEUs on a system [21]. Second was an experiment involving device operation under exposure to neutron radiation, this provided a realistic fault model and thus allowed a realistic determination of the DUT's FIT rate for each error type.

The fault-injection was controlled by the Arduino which selected a random register and bit location and flipped its contents. As this method requires the registers to be writable via the NeuroShield's SPI interface, two neuron cell registers were inaccessible and thus un-injectable. Furthermore, since a lot of

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

the NWCRs were un-injectable and predicted to be responsible for solely KF, none of them were included in the injection process. This had the added benefit of increasing throughput by removing the need for power cycling of the device in the event of KF.

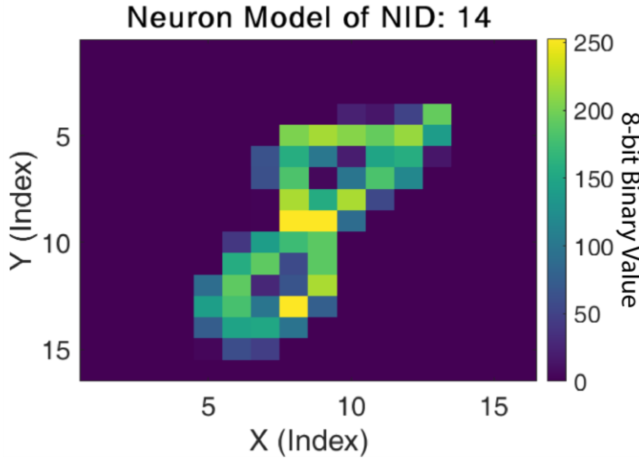

Fig. 5. Pixel values of a neuron cell's model. The representation is based on the MNIST images in Fig. 3 and shows the similarity between neuron models and the input patterns used to generate them. The image from Fig. 3 would make up an input vector and the image form Fig. 5 would make up a neurons unique model. The more different these 2 images are, the larger the "distance" between them, and the smaller the confidence of the classifying neuron

The DUT was positioned to be 2.5m away from the ChipIr beam-stop with a collimated beam size of 70x70mm. The network was always trained with the knowledge file produced from the down-sampled dataset presented in Section III. The device was power cycled before each test.

The network was first initialised with a previously generated knowledge file containing a snapshot of the contents of every register on the network. Then, the device was traversed into the beam and classifications were performed until an error was detected. Once an error was detected, the device was traversed out of the beam and all results were output to the external monitoring computer running a LabVIEW program.

C. Experimental Predictions

Each error type's probability of occurrence was initially garnered by first assuming them to be directly proportional to the size of their responsible registers. In order to determine which registers are responsible for each error type, a combination of theoretical device operation analysis and highly controlled fault-injection was used. The results of these analyses are presented in the third column of Table I.

Following this, the *Bit Proportion* of each error type is calculated by dividing the total bit size of contributing registers by the total device bits, shown in the fourth column of Table I.

Also shown in the third column of Table I are *No Error* types, this occurs when the registers produce no output effect when corrupted due to either not being involved in the classification processes or through self-correction.

Conveniently, the two un-injectable neuron cell registers, are thought to have such a small effect on the output in response to single bit flips, that they can be assumed to present a *No Error* type. Furthermore, following the injection of faults into the *Influence Field* registers of the neurons, it was found that a single bit flip was very unlikely to cause an output error (none were observed). Thus, the un-injectable neuron cell registers have been combined with the *Influence Field* registers to make up *Other Neuron Reg.* in the first column of Table I.

TABLE I
REGISTERS, THEIR SIZE AND POSSIBLE ERROR TYPES

Register	Size (bits)	Error Type	Bit Proportion
NWCRs	278	KF (70) No Error (208)	0.00567% 0.0168%
Neuron Memory	576 * 2048	DE	95.5%
Neuron Context Reg.	576 * 8	NE	0.373%
Neuron Category Reg.	576 * 16	CE	0.746%
Other Neuron Reg.	576 * 72	No Error	3.36%
Total Device Bits:	1,235,222		

The two neuron cell registers which could not be injected were the *Distance* Register and the *Identifier* Register [12]:

- *Distance* Register (16-bit) - Stores the confidence of the classifying neuron. Bit flips are self-corrected due to recalculations of this value.
- *Identifier* Register (24-bit) - Stores the location of the neuron in the network. The value is output as the *Neuron Identification Number*, bit-flips do not affect the classification and only cause difficulties with identifying problematic neurons.

The *Bit Proportions* of Table I can be thought of as each error type's probability of occurrence. By combining the DUTs per-bit cross-section from Section V with each error type's probability of occurrence, the *Expected FIT* for each type could be calculated and is shown in the third column of Table II.

Bit Proportions of the *Distance Register* and the *Identifier Register* (576 * (16+24) bits) have been combined with those of *No Error* NWCRs (208 bits), to produce the *Un-injectable No Error* probability. Finally, the remaining *Bit Proportions* of *Other Neuron Reg.* (576 * (72 - 40) bits) make up the *Injectable No Error* probability. These values are shown in Table II.

TABLE II
PREDICTED ERROR PROBABILITIES AND RATES

Error Type	Error Probability	Expected FIT
DE	95.5%	610
CE	0.746%	4.8
NE	0.373%	2.3
KF	0.00567%	0.036
Un-injectable No Error	1.89%	-
Injectable No Error	1.40%	-

D. Error Probabilities from Fault-Injection

The fault-injection performed a total of 35,094 bit-flips, 2.9% of the device's total bits, taking approximately 50 hours. The results are shown in Table III. Throughout the entirety of the testing, no KF was observed. Since NWCRs were not injected, this adds credence to the idea that KF originates in the

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

NWCRs. It is important to note that KF could still occur as a result of bit flips in either the *Identifier* or *Distance* registers but since they cannot be written to via the SPI interface they could not be investigated here.

TABLE III
FAULT-INJECTION PREDICTIONS VS RESULTS

Error Type	Predicted Probability	Injected Error Probability
DE	95.5 %	97.4 %
CE	0.746 %	0.707 %
NE	0.373 %	0.430 %
Injectable No Error	1.40 %	1.48 %

Since the results are in reasonable agreement with the predictions, it can be assumed that the critical bits for DE, CE, and NE have been identified. This provides a basis for Section VII where the hardening method will aim to reduce the criticality of the bits responsible for CE.

E. Error Rates from Neutron Radiation

Radiation testing involved approximately 14 hours of operation under neutron exposure, this equates to approximately 2,500,000 device-years of operation in the earth's atmosphere. A total of 2,818 errors were observed during the testing. In Fig. 6, the predicted FIT Rates from the third column of Table II are shown alongside the FIT Rates measured from the experiment. No KF errors were observed during testing, however this is likely due to needing 98 hours of radiation to expect to see even a single KF event. Furthermore, no Arduino or DUT crashes were observed.

From the data shown in Fig. 6, DE is confirmed to be the most likely error type to occur with CE being more likely than NE. The larger discrepancy for NE is unexpected and further testing will be needed to identify the cause.

The error rates determined in this section provide a benchmark upon which the hardening method presented in Section VII-B can be evaluated.

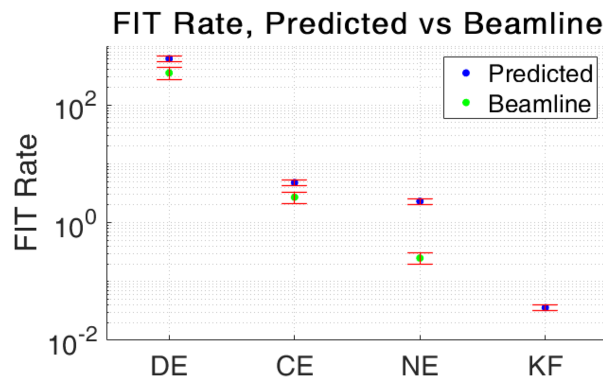

Fig. 6. A graph with a logarithmic y-axis comparing the predicted FIT Rates with the experimental FIT Rates obtained from the beamline.

VII. Hardening Method Development

A. Critical Errors and Critical Bits

For the purposes of this hardening method, a critical error will be defined as one where a bit flip in the device results in an input pattern being classified incorrectly. When related back to the error types and their rates presented in Section VI, it can be seen that even though DE occurs most frequently, only CE and KF are guaranteed to produce a critical error. Since KF occurs so rarely (as evidenced by the radiation experiment of Section VI-E), only the bits responsible for CE will be targeted for hardening. From the fault-injection of Section VI-D, we know that CE is caused by bit flips in the neuron cell category registers and so this is where the hardening method will focus its attention.

B. The Hardening Method

In Section VI-A, error types were defined, and their corresponding critical bits identified. In Section VII-A a critical error was defined and CE's critical bits (the neuron cell category registers) were chosen as the area to harden. In order to harden these bits, it is necessary to first understand their function as part of the wider system.

The category register is a 16-bit register located within each of the 576 neuron cells [12]. Thus, there are a total of 9,216 critical bits for the CE error type that will need to be considered. Of these 16 bits per register, only 15 represent the dataset's categories; the 16th is a reserved *Neuron Degenerated Flag*, which (if set) shows that care should be taken when relying on the neuron's response [10]. It is important to understand that a bit flip in any of these 16 bits would cause a CE and thus a critical error. Since only 4 bits are being used to represent the MNIST dataset's 10 categories, 11 bits remain unused in each category register.

Triple Modular Redundancy (TMR) has been demonstrated as an effective hardening solution in FPGAs [22][23]. Typically, this method involves the repetition of device components such as memory registers. Since we do not have the ability to change the physical architecture of the NeuroShield, we must look at how we can implement repetition with the hardware already available on the device. As established previously, there are 11 unused bits in each category register when representing the MNIST categories, these bits can be used to add *Informational Redundancy* at no additional cost. More specifically, the hardening method will implement TMR by repeating the 4 bits required for the MNIST categories 3 times within each of the category registers. This method can be thought of as a majority voting system where each copy of the category data is used to represent a vote for the correct output. The most common of these three votes becomes the final output of the device, the classification, Fig. 7.

It is worth mentioning that arithmetic coding has been used in the past to improve the reliability of data storage and transmission [23][24]. *Cyclic Redundancy Checks* are a popular method, but their complexity made it difficult to consider them for implementation on this device.

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

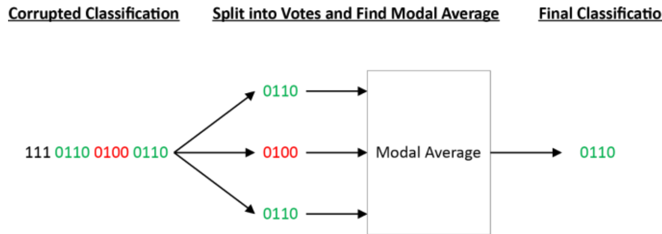

Fig. 7. The TMR decoding process. The black 111 shows redundant bits which can be ignored. The green 0110 shows the uncorrupted votes containing the correct category. The red 0100 shows a vote with a corrupted 3rd bit. Once the modal average is taken, 0110 becomes the output.

The method was implemented by replacing the category of all entries in the dataset with the TMR equivalent category. If the category was originally 6 (000 0000 0000 0110 in 15-bit binary), it would be replaced with 30310 (111 0110 0110 0110 in 15-bit binary). Once the MNIST dataset was modified in this manner, the network was re-trained, and a new knowledge file containing the contents of all registers was produced.

The efficacy of the hardening method we have designed depends on the number of classes the CNN needs to classify. The higher the number of classes, the lower the redundant bits available. Given the TMR requirements, with the 16-bit register, we will still guarantee error correction for a dataset of up to 2^5 = 32 classes and detection (with DMR) for up to 2^7 = 128 classes. The NeuroShield, as other EdgeAI devices, are not meant to be employed in highly complex scenarios, such as autonomous cars, where hundreds of different classes of objects need to be identified. These devices are designed for Internet of Things (IoT) applications, where 128 classes are more than sufficient. We then claim that the proposed solution is valid for real-world use cases of EdgeAI devices.

This is evidenced to be more than sufficient for current real-world use-cases of this device such as pass/fail inspections which require only 2 categories [3], and an industrial olive pitting machine where only 4 categories are required [25]

It is important to note that a separate device is needed to implement the output decoding logic. This could be performed by the attached microcontroller or even by a dedicated circuit. A dedicated circuit composed of low-power Schottky devices was simulated in LTspice and was shown to add no more than approximately 200ns of propagation delay.

VIII. Hardening Method Evaluation

A. Experimental Methodology

The hardening method was evaluated using the same neutron radiation test presented in Section VI-B with the experimental setup presented in Section IV. The only difference being the modifications to the categories of the MNIST dataset used to train the network. The DUT was positioned to be 2.5m away from the ChipIr beam-stop with a collimated beam size of 70x70mm. The decoding of the classification output was performed post irradiation at the data analysis stage in MATLAB.

B. Error Rates from Neutron Radiation

The device was exposed to the same amount of neutron radiation as in the radiation test of Section VI-E. A total of 22 critical errors were observed in the output of the DUT, this was reduced by over 95% to 1 critical error once passed through the decoding logic. The critical error cross section was reduced from $(2.2 \pm 0.6)\times10^{-10}$ cm^2 to $(9.7 \pm 2.4)\times10^{-12}$ cm^2 (shown in Fig. 8), a very good result, which highlights the effectiveness of even a simple hardening solution in reducing the impact of Single Event Effects (SEE) in COTS EdgeAI devices.

The hardening solution we propose is designed based on the experimental observations and is dedicated to the NeuroShield, which is why it results to be so efficient. Generic solutions, such as DMR, can always be applied and are highly generic. However, they come with a not negligible overhead. While the proposed solution is dedicated to the NeuroShield device, there is no reason why an efficient, dedicated, and experimentally-tuned hardening solution should not be found for other accelerators. As we have shown in our paper, by analysing with both fault injection and beam experiments the behaviour of an EdgeAI device affected by transient faults, it is possible to design highly efficient and effective hardening solutions.

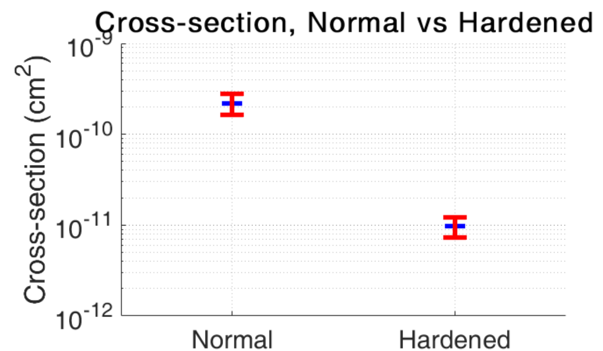

Fig. 8. A graph with a logarithmic y-axis comparing the critical error cross-section of the DUT, before and after hardening.

IX. Conclusion

This work has examined the effects of SEE on a COTS EdgeAI accelerator, the NeuroShield. The cross-section measured in Section V is comparable to that of similar SRAM technology nodes. The device's critical error FIT Rate, 4.8, already seems promising for use in applications requiring good reliability. The TMR hardening method presented in Section VII-B and evaluated in Section VIII has been shown to perform very well, reducing the number of critical errors by 96%. While this solution is validated only for the NeuroShield device, we have shown that it is possible, by analyzing fault injection and beam experiment results, to design dedicated (and thus highly efficient) hardening solutions for EdgeAI devices.

References

[1] NVIDIA Corporation, *Data Sheet: Quadro GV100*. (2018). [Online]. Available: https://images.novatech.co.uk/website2017/DeepLearning/assets/datasheets/PNY%20Quadro-GV100.pdf

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Nuclear Science. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNS.2021.3086686

[2] General Vision Inc. “CogniSight engines.” general-vision.com. [Online]. Available: https://www.general-vision.com/cognisight/ (accessed 27-Jan-2020).

[3] General Vision, “Neuromorphic applications.” [Online]. Available: https://www.general-vision.com/solutions/ (accessed Jan. 14, 2021).

[4] M. Amrbar, F. Irom, S. M. Guertin and G. Allen, "Heavy Ion Single Event Effects Measurements of Xilinx Zynq-7000 FPGA," 2015 IEEE Radiation Effects Data Workshop (REDW), Boston, MA, 2015, pp. 1-4, doi: 10.1109/REDW.2015.7336714.

[5] F. Fernandes dos Santos et al., “Analyzing and Increasing the Reliability of Convolutional Neural Networks on GPUs,” *IEEE Trans. Rel.*, vol. 68, no. 2, pp. 663-677, Jun. 2019, doi: 10.1109/TR.2018.2878387.

[6] MIT Technology Review. “On-Device AI.” technologyreview.com. [Online]. Available: https://www.technologyreview.com/hub/ubiquitous-on-device-ai/ (accessed 7-Oct-20).

[7] R. M. Brewer et al., “The Impact of Proton-Induced Single Events on Image Classification in a Neuromorphic Computing Architecture,” *IEEE Trans. Nucl. Sci.*, vol. 67, no. 1, pp. 108–115, 2019, doi: 10.1109/TNS.2019.2957477.

[8] General Vision. “NM500, neuromorphic chip with 576 neurons.” general-vision.com. https://www.general-vision.com/hardware/nm500/ (accessed Jun. 29, 2020).

[9] General Vision. “NeuroShield.” General-Vision.com. [Online]. Available: https://www.general-vision.com/hardware/neuroshield/ (accessed Jun. 29, 2020).

[10] nepes, *NM500 User’s Manual v 1.6.3*. (2019). [Online]. Available: https://general-vision.com/documentation/TM_NM500_Hardware_Manual.pdf (accessed Apr. 4, 2021).

[11] General Vision. “Powerful RBF classifier.” general-vision.com. [Online]. Available: https://www.general-vision.com/neuromem/rbf_knn/ (accessed Jun. 29, 2020).

[12] General Vision. *NeuroMem Technology Reference Guide v5.4*. (2019). [Online]. Available: https://www.general-vision.com/documentation/TM_NeuroMem_Technology_Reference_Guide.pdf (accessed Apr. 4, 2021).

[13] L. Deng, “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web],” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, Nov. 2012, doi: 10.1109/MSP.2012.2211477.

[14] R. Fisher et al. “Mean Filter.” HIPR2. http://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm (accessed Jun. 29, 2020).

[15] C. Peng et al., “To What Extent Does Downsampling, Compression and Data Scarcity Impact Renal Image Analysis?”. [Online]. Available: https://arxiv.org/abs/1909.09945 (accessed Apr. 4, 2021).

[16] D. Chiesa et al., "Measurement of the neutron flux at spallation sources using multi-foil activation." Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 902, pp. 14-24, 2018, doi: 10.1016/j.nima.2018.06.016.

[17] Cazzaniga, Carlo, et al. "Study of the Deposited Energy Spectra in Silicon by High-Energy Neutron and Mixed Fields." IEEE Transactions on Nuclear Science, vol. 67, pp. 175-180, 2019, doi: 10.1109/TNS.2019.2944657.

[18] C. Cassaniga et al., “First tests of a new facility for device-level, board-level and system-level neutron irradiation of microelectronics,” *IEEE Trans. Emerg. Topics. Comput. Early Access*, vol. 9, pp. 104-108, Nov. 2018, doi: 10.1109/TETC.2018.2879027.

[19] Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices, JESD89A, Oct. 2006. [Online]. Available: https://www.jedec.org/sites/default/files/docs/JESD89A.pdf (accessed Apr. 4, 2021).

[20] A. Prado et al., “Effects of cosmic radiation on devices and embedded systems in aircrafts,” INAC 2013: international nuclear atlantic conference; Recife, PE (Brazil); 24-29 Nov 2013

[21] M. Rouse. “DEFINITION, fault injection testing.” SearchSoftwareQuality. [Online]. Available: https://searchsoftwarequality.techtarget.com/definition/fault-injection-testing (accessed Jun. 29, 2020).

[22] Actel. *Design Techniques for Radiation-Hardened FPGAs*. (1997). [Online]. Available: https://www.microsemi.com/document-portal/doc_download/129913-ac128-design-techniques-for-radhard-fpgas-app-note

[23] P. Balasubramanian et al., “Majority and Minority Voted Redundancy Scheme for Safety-Critical Applications with Error/No-Error Signaling Logic,” *Electronics*, vol. 7, no. 11, Art. no. 272, Oct. 2018, doi: 10.3390/electronics7110272.

[24] J. Chou and K. Ramchandran, "Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission," in IEEE Journal on Selected Areas in Communications, vol. 18, no. 6, pp. 861-867, June 2000, doi: 10.1109/49.848240.

[25] A. Lucas Pascual et al., “Analysis of the Functionality of the Feed Chain in Olive Pitting, Slicing and Stuffing Machines by IoT, Computer Vision and Neural Network Diagnosis,” Sensors, vol. 20, no. 5, Art. no. 1541, Mar. 2020, doi: 10.3390/s20051541.

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.