Reliability Analysis on Case-Study Traffic Sign Convolutional Neural Network on APSoC

Israel C. Lopes, Fabio Benevenuti, Fernanda Lima Kastensmidt, Altamiro A. Susin and Paolo Rech
Instituto de Informática – PGMICRO
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil
israel.lopes@inf.ufrgs.br, fbenevenuti@inf.ufrgs.br, fglima@inf.ufrgs.br,
altamiro.susin@ufrgs.br and prech@inf.ufrgs.br

***Abstract*—Deep learning has been widely used to solve computer vision applications such as autonomous cars and Advanced Driver Assistance Systems (ADAS). One of these computer vision applications is traffic sign recognition that plays an essential role in autonomous cars and ADAS. All-Programmable System-on-Chip (APSoC) is an excellent target platform for implementing ADAS applications due to its flexibility. However, as its Programmable Logic (PL) is implemented with SRAM cells and traffic sign recognition is a safety critical application, the reliability under radiation must be analyzed and hardening or mitigation techniques may be required. Thus, random and piled-up fault injections in the APSoC configuration bits were carried out as well as neutron-radiation experiments in order to evaluate the reliability of a Convolutional Neural Network (CNN). Comparison with related work shows that timing-multiplexing neural network architectures present an insignificantly higher Architecture Vulnerability Factor (AVF) than parallel pipelined architectures.**

***Index Terms*—Deep Learning, Traffic-sign recognition, APSoC, Reliability, SEU**

I. Introduction

In the past, radiation reliability on integrated circuits was only a concern in space and particle-accelerator applications. However, due to the need for more density and low-power circuits, they became more sensible to radiation effects. Thus, nowadays it also a concern for avionics, and ground applications (e.g. cars, trains and communication). Autonomous cars and Advanced Driver Assistance System (ADAS) use a considerably amount of electronic components to safely deliver the passengers to their destination and avoid accidents, thus the radiation reliability of those components must be characterized [1] [2]. It also a requirement of the International Organization for Standardization (ISO) 26262 standard [3].

All-Programmable System-on-Chip (APSoC) is commonly used for implementing autonomous car and ADAS applications due its flexibly and energy efficiency. However, the hardware flexibility of the the programmable logic (PL) part out of the APSoC also increases its vulnerability to radiation effects, being a primary concern for SRAM-based FPGAs [4]. In this way, a random-accumulated fault injection in the configuration bits of the PL was carried out in order to evaluate the impact of Single Event Upsets (SEU) in the reliability of the system. Additionally, neutron-induced radiation experiments were conducted to evaluated the radiation single effects on all the board, being a more realistic and embracing experiment.

As Design Under Test (DUT), a Convolutional Neural Network (CNN), which has successfully been employed to recognize traffic-signs for autonomous car and ADAS, was implemented in the PL [5] [6]. It was trained with the Convolutional Architecture for Fast Feature Embedding (Caffe) framework to classify 6 speed-limit traffic-sign classes (20, 30, 50, 69, 70, 80 km/h) from the German Traffic Sign Recognition Benchmark (GTSRB) dataset [7] [8].

The fault injection results showed that the reliability of a Neural Network implemented in the PL is considerably influenced by its hardware architecture and the radiation experiment results showed that the failure rate is suitable in three scenarios, but for the worst case scenario the failure rate must be improved. The main contribution of this work is to analyze the reliability of a CNN implemented on an APSoC instead of Graphics Processing Units (GPUs) [9].

II. Design Under Test

A. Dataset

The case study CNN was trained to classify 6 classes out of the 43 classes of the GTSRB by using the Caffe framework [7] [8] reaching an average precision of 90%. As a minor preprocessing the images were converted to grayscale, resized and their contrast was enhanced.

B. Convolutional Neural Network

Convolutional Neural Networks (CNN) extract features from signals (images, audio, time-series) and classifies them into classes. The feature extraction is performed by computing the cross-correlation, as described in Equation 1, and the results are divided in feature maps [10].

$$g(x,y) = \left[\sum_{n} \sum_{s} \sum_{t} w(s,t,n) f(x+s, y+t, n) \right] + b \quad (1)$$

The authors are partially supported by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) of Brazilian government.

978-1-5386-1472-3/18/$31.00 © 2018 IEEE

In Equation 1 the first sum iterates the previous feature map, the second and the third iterates the x and y coordinates of the kernel, w is the weight, f the kernel, and b is the bias.

In the proposed model the first convolutional layer has 6 feature maps and the second has 12, both with a kernel size of 5 and stride 1.

The convolutional layers are followed by pooling layers that subsample the feature maps reducing their sensitivity to variations. This operation is made by computing the maximum value or the average of a kernel with a specific stride [10]. In the pooling layers, kernel sizes and strides of 2 were used.

The last pooling layer is succeeded by Fully-connected (FC) layers that are neurons that receive features and classified them into classes. FC neurons multiply an input vector by a weight vector and sum all the products with a bias, also known as inner product operation, as described in Equation 2 [10]

$$y = \sum_{i=0}^{n} w(i)x(i) + b \tag{2}$$

where n the number of inputs, w is the weight, x is the input, and b the bias.

The proposed CNN model has 50 neurons in the first FC layer, 20 in the second and 6 in the third.

The output of the FC layers, except the last, is connected to a Rectified Linear Unit (ReLU) activation function that employs the function $r(y) = max(0, y)$, that is widely used in deep neural networks as non-linear function due to its implementation simplicity [10].

The last FC layer is followed by the softmax function that computes the probability vector of the classes, as defined in Equation 3 [11]

$$O_j = \frac{\exp I_j}{\sum_k \exp I_k} \tag{3}$$

where I is the FC outputs, j is the index of the current FC neuron and k is the index of the others FC neurons.

C. Timing-multiplexing architecture

The hardware description language of the case study CNN was automatically generated by a tool developed by the authors [12]. The tool generates a timing-multiplexing architecture where the inputs and weights of the Multiply and Accumulate (MACC) units are multiplexed in time. The CNN top module is illustrated in Figure 1. Each layer has a control component that controls the timing multiplexing of the current layer and the kernel of the next layer, but only the "CONV1_CTRL" component controls the true dual Block Random Access Memory (BRAM).

The internal part of the Convolutional layer 1 is illustrated in Figure 2. All the layers have similar architectures, the operational part of the convolutional and fully-connected layers are composed by MACC units that implement the neurons and Look-Up Table (LUT) Read-Only Memories (ROMs) that store the weights and bias, whereas pooling layers has only Max comparators. The ReLU function is computed in the control component of the Fully-connected layers. As the softmax

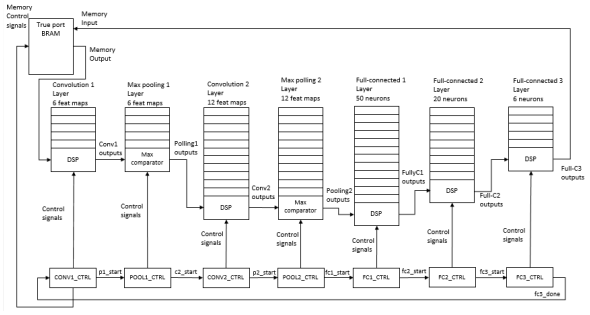

Fig. 1. CNN implementation top module

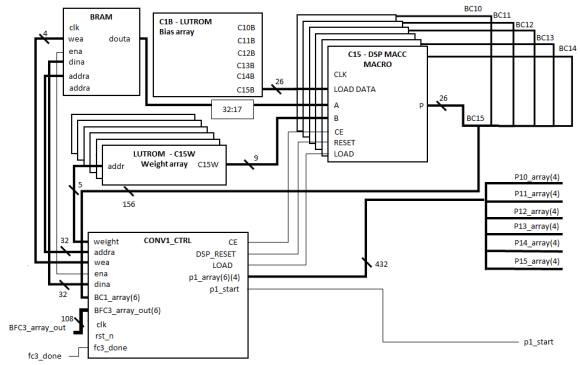

Fig. 2. Convolutional 1 - control and datapath

function requires a great amount of hardware resources due to the sums of exponential, it was implemented in software.

III. Test Methodology

A. Failure model

Some bits out of a hardware design are responsible for its Architecturally Correct Execution (ACE) and the remaining bits are called un-ACE bits [13]. Undetected errors due to bit-flips in ACE bits are referred to as Silent Data Corruptions (SDCs) [13].

A SDC in the proposed CNN design, i.e a difference in the CNN output, is referred to as "error" when it does not change the correct traffic-sign classification, otherwise it is referred to as "failure". A "failure" is classified as critical SDC when it is due to a difference in the CNN output and it is classified as a time-out when the CNN does not generate the output in its specified execution time, and consequently the traffic-sign is not classified. An illustration showing how a SEU (represented as flash marks) can cause different types of effects is given in Figure 3.

B. FPGA reliability evaluation

Field Programmable Gate Arrays (FPGAs) are programmed by configuration bits, but only a percentage out of these bits are used by the user design. These bits are called essential bits. The essential bits of a design can be extracted by the Synthesis tool of Xilinx FPGAs. Critical bits are essential bits that once flipped cause a system failure. Thus, the susceptibility of a

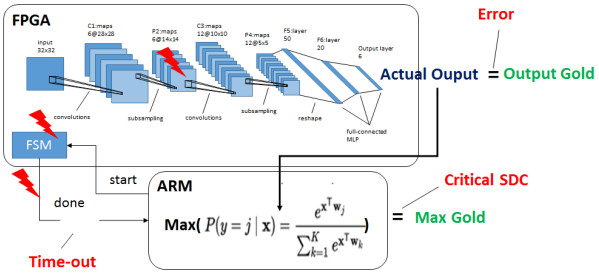

Fig. 3. Failure model APSoC CNN

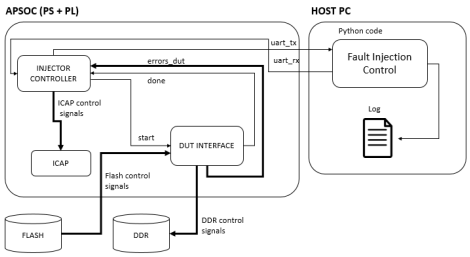

Fig. 4. Fault injection setup

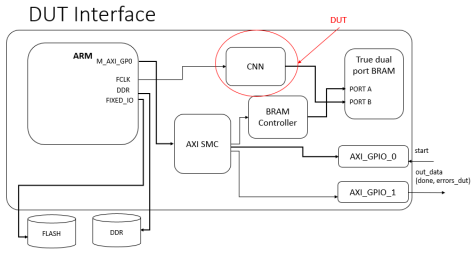

Fig. 5. DUT interface component expansion

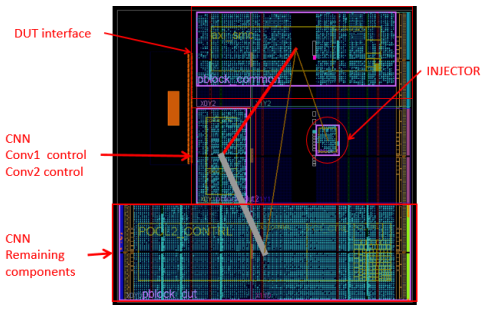

Fig. 6. Implemented design floorplanning

FPGA design can be measured by the percentage of the critical bits among the essential bits [14].

The reliability of a FPGA design can be evaluated by computing the Architectural Vulnerability Factor (AVF) that is the probability that a visible system error will occur given a bit flip in a storage cell [13]. Thus, the reliability can be expressed as proportional to the complement of the AVF (100% - AVF).

C. Fault injection experimental setup

The experimental setup is composed by a Xilinx Zynq-7000 APSoC and a host PC, as illustrated in Figure 4. The Python application running in the PC randomizes the configuration frame and the bit position and then send them to the injector control component implemented in the Programmable Logic (PL) part out of the APSOC by using an Universal Asynchronous Receiver/Transmitter (UART) port emulated with an Universal Serial Bus (USB) connection.

The injector controller sends the frame to the Internal Configuration Access Port (ICAP) in order to inject the fault, then it starts the DUT interface using a General Purpose Input/Output (GPIO) port. This port is connected to the ARM processor that reads an image from the Double Data Rate (DDR) memory and writes it as well a "start" signal in a BRAM memory by using the Advanced eXtensible Interconnect Static Memory Controller (AXI-SMC) bus. The image is read by the CNN, and the CNN results are written in the same BRAM. The ARM receives the result, classifies the fault, and send a done signal as well the fault classification to the injector control component. Finally the injector control component sends the fault classification to the Python application that writes it in a log file, and then the loop resumes.

In the beginning of the code, the ARM application must disable the Processor Configuration Access Port (PCAP) in order to enable the ICAP usage and disable all the cache levels to avoid data corruption. The ARM application classifies the faults by comparing the CNN output with the gold CNN outputs and the maximum value of the soft-max function with the gold maximum values. The time-outs are signalized when the CNN does not write a done signal in the BRAM after its specified execution time.

The Xilinx Zynq-7000 floorplanning (Figure 6) is divided into 3 rows, one row (row 0) lies on the top region and the other two rows (row 0 and row 1) are situated on the bottom region. Due to the high area utilization of the CNN design, it had to be divided into two physical blocks (pblocks), one on the left of the bottom row 0 and another in the entire bottom row 1. The injector controller is isolated from the DUT regions and strategically placed near the ICAP, the remaining components of the design that do not belong to the DUT are isolated from the DUT regions in the top row 0 with a reliable level of distance from the DUT region on the bottom row 0.

D. Radiation experimental setup

The Zynq-7000 Zedboard development board implementing the DUT was irradiated at the ICE House facility at the Los Alamos Neutron Science Center (LANSCE) [15], Los Alamos, NM. LANSCE is a spallation neutron source, which produces neutrons through a pulsed proton beam hitting a tungsten target. Neutrons are then directed towards the devices under test.

LANSCE produces neutrons with a spectrum of energies with a shape similar to the one of terrestrial neutrons at ground level. The neutron flux available at LANSCE, however, is several orders of magnitude higher than natural one, allowing the gather of a statistically significant amount of data in a short time.

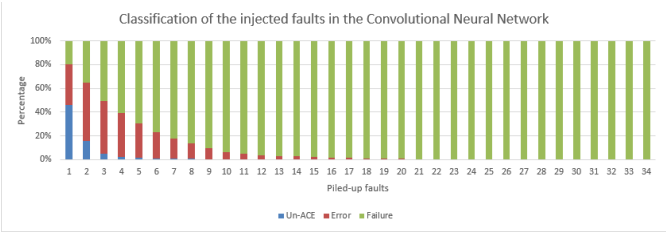

Fig. 7. Classification of the injected faults in the Convolutional Neural Network

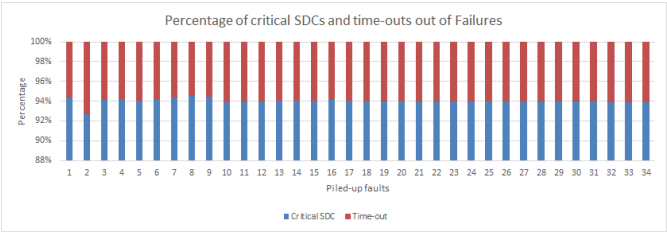

Fig. 8. Percentage of critical SDCs and time-outs out of Failures

For instance, while the reference neutron spectrum at New York [16] is 3.6×10^{-3} N/cm^2/s, the average flux during our experiments was measured to be about 1.93×10^{6} N/cm^2/s.

This flux measurement is provided by a fission detector recorded during the experiment in intervals of 10 s. During these experiments it was obtained an average count of 100 events/s with standard deviation of 10. The fission counting is converted to flux by a factor recorded during the experiment and having an average of 22117 and standard deviation of 94. The product gives the estimated flux of 1.93×10^{6} N/cm^2/s.

The fission detector giving these measurements is located at a distance r=19.7 m from the source and the board running the DUT was positioned at a distance d=1.4 m from the detector so that the measured flux is reduced by a factor of

$$\frac{r^2}{(r+d)^2} = 0.87$$

resulting in a flux at the board estimated as ϕ=1.68×10^{6} N/cm^2/s.

IV. Results

A. Fault injection results

The percentage of un-ACE, errors and failures versus the piled-up faults is plotted in Figure 7. 1000 piled-up fault injection campaigns were performed, where the maximum number of piled-up faults was set to 300. Nevertheless, it will be shown in the results that the piled-up faults do not reach this value because when a failure occurs the campaign is ended.

As it can be seen in Figure 7 when one fault is injected (SEU), the un-ACE effects comprise 47.3% out of the fault injections campaigns and this percentage is exponentially decreased with a high slope as the faults are piled-up. Un-ACE effects occur up to 31 faults are piled-up. Indeed, un-ACE under many piled-up faults is pretty unlikely (0.1%), provided that from 6 piled-up faults onwards the un-ACE probability of happening is less than 1%.

Regarding the errors under one injected fault (SEU), their percentage comprises 33.2% out of the fault injection campaigns and it dominates the portion of ACE effects (errors + failures). However, as the faults are piled-up, the portion of errors is decreased, whereas the portion of failures is increased. This difference is exponentially increased so that after 20 piled-up faults, approximately all the ACE effects are failures. The errors occur up to 34 faults are piled-up but it has an extremely little probability of happening (0.1%).

These results imply that the proposed CNN has a high number of critical bits among the essential bits, consequently its susceptibility is rapidly increased with the piled-up faults in the essential bits. Once the essential bits comprise about 57.33% out of the physical block configuration bits and 52.7% (100 - 47.3) of the configuration bits injected are ACE bits, approximately 91.92% (52.7/57.33) out of the essential bits are able to cause a difference in the system output. Fortunately, instead of the portion of failures, the portion of errors will not change the correct traffic-sign classification, because they do not change the maximum value of the softmax vector.

Concerning failures, the percentage of critical SDCs and time-outs out of the failures versus the piled-up faults are shown in Figure 8. It is possible to note that regardless of the number of piled-up faults, the critical SDCs and time-outs percentage practically remain the same, the critical SDC percentage varies around 92% to 94% out of the failures. It happens due to an architecture feature of the design. The time-outs are caused by upsets in LUTs that implement the logic of the state machines, or routing signals used by these state-machines, consequently the CNN does not complete the computation and never sends the “done” signal. Given the fact that these resources comprise a little portion of the total design resources, it is pretty unlikely that the faults are injected in the configuration bits associated to these resources. Rather, CNNs use a lot of LUTs to store the Fully-connected weights and many routing signals to connect layer outputs to layer inputs. An upset in these resources does not affect the state of the system, but only corrupts the system output. Hence, the critical SDCs are more likely to occur.

It is also important to know how many images from the dataset were incorrectly classified in order to analyze the impact of the failures. The dataset contains one image per class, so from one to six images can be wrongly classified. When a critical SDC occurs, there can be different numbers of wrong classifications whereas when a time-out occurs, all the images from the dataset (6 images) are wrongly classified. The percentage of failures by which from one to six images are wrongly classified versus the piled-up faults is plotted in Figure 9.

Regardless the number of piled-up faults, the percentage of each number of wrong classifications almost remains the same. Only one image was wrongly classified in about 40% out of the failures, 2 images were wrongly classified in about 20% and 3, 4, 5 and 6 images were wrongly classified in

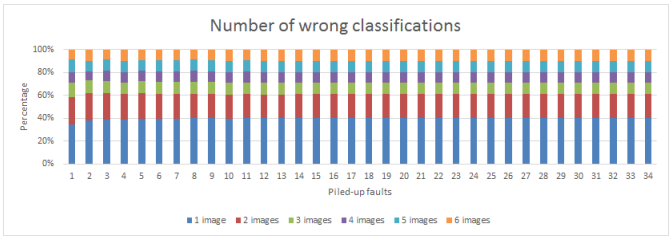

Fig. 9. Number of wrong classifications

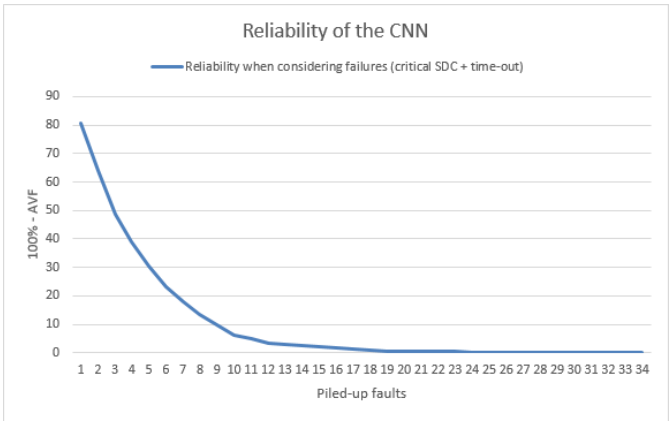

Fig. 10. Failure reliability

approximately 10%. As upsets in the PL configuration bits are persistent effects, multiple wrong classifications are expected instead of a single wrong classification. However, in 40% of the cases, only 1 image is wrongly classified and the other five are correctly classified. This may occur, because one image from the dataset may be the most difficult image of being classified, another image must be the second most difficult and the remaining images are equally difficult.

The reliability curve when considering failures, is plotted in Figure 10.

As it can be seen in Figure 10, the reliability of the CNN considering failures (critical SDC + time-outs) is about 80.5% and exponentially decreases with the number of piled-up faults. The maximum number of piled-up faults before a failure occurs is 34 but this case has a minimum chance of happening. As the essential bits comprise 57.33% out of the entire physical block configuration bits and the AVF due to SEU is about 19.5%, the critical bits of the proposed design are approximately 34.03% (19.5/57.33) out of the essential bits.

Related work [17] [18] shows that Neural Networks can have different AVFs depending on the topology of the Neural Network and its hardware architecture. In [17] the Neural Network AVF is about 0.62% while the AVF of the proposed CNN is about 19.5%. Nevertheless, in [17] a full-parallel approach was employed, thus an upset affects only a dedicated node and can be masked by the combination of the other nodes. As the area resources are reused in the proposed CNN, a persistent effect will affect many cycles being more likely to be propagated to the output. The AVF of the Hopfield Neural Network (HNN) implemented by [18] is about 15.64%, that is near the AVF of the proposed CNN, 19.5%. This Neural Network topology also reuses hardware resources, inasmuch as it has feedback and after a specific number of cycles the output converges to a value. Therefore, it is possible to note that Neural Networks that reuse hardware resources are more susceptible than parallel pipe-lined Neural Networks.

Furthermore, there is the possibility of a fault being injected in a configuration bit that transforms LUT used as memory into a shift-register. In this case, a SEU in this specific configuration bit will cause multiple differences in other configuration bits [19]. As mentioned in section II-C, LUTs as memory were extensively used to store the weights, therefore there is a great probability that these LUTs will be affected by the random fault injection.

B. Radiation results

The board running the DUT was irradiated during 9.5 h. After removing spurious periods due to occasional beam interruption it resulted in N_{SEU}=134 failure events accounting to 23,190 s of run time.

The time to failure events recorded was converted to neutron particle fluence Φ as

$$\Phi = \phi \times t$$

giving the reliability curve presented on Fig. 11.

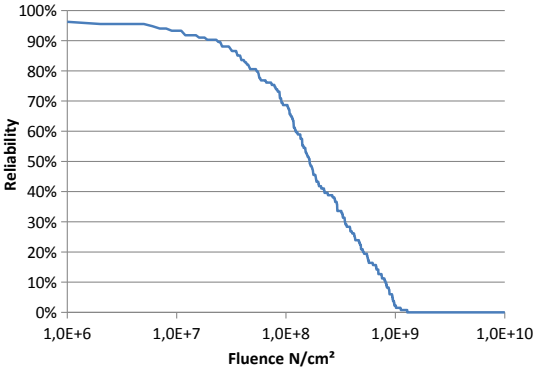

Fig. 11. Reliability curve from irradiation experiment

Also from fluence we can compute dynamic cross section as

$$\sigma_{SEU} = \frac{N_{SEU}}{\Phi} = \frac{N_{SEU}}{\phi \times t} = \frac{134}{1.68 \times 10^6 \times 23,190} = 3.45 \times 10^{-9} cm^2.$$

Using the reference flux in New York city ϕ_{NYC}=3.6×10^{-3} N/cm^2/s = 13 N/cm^2/h [16] we can estimate the soft error rate (SER) at ground level as

$$SER = \sigma_{SEU} \times \phi_{NYC} \times 1 \times 10^9 / FIT = 3.45 \times 10^{-9} \times 13 \times 10^9 = 44.8\ FIT.$$

The required FIT of an electronic component of an autonomous cars or ADAS is established by the ISO 26262 standard and it is defined by the Automotive Safety Integrity Level (ASIL). It is expressed in terms of Probabilistic Metric for Hardware Failure (PMHF). The ASIL classification defines the type of the risk of a failure and it depends on three metrics: Severity, Controlability and Exposure. Severity means the impact of a failure like no-injuries, severe, life-threatening, and fatal injuries. Exposure represents the probability of occurrence such as incredible, very-low, low, medium, and high probability. Lastly, controlability describes how difficult it is to fix a failure, so it can be controllable in general, simply, normal, difficult to control or uncontrolable. The ASIL is classified into A, B, C, D, where D is the highest level and A the lowest level, as shown in Table I [3].

TABLE I
PMHF REQUIREMENT IN ISO 26262

ASIL	PMHF requirement	note
D	<1E-8/hr $(10FIT)^3$	required
B	<1E-7/hr $(100FIT)^3$	required
C	<1E-7/hr $(100FIT)^3$	recommended
A	-	not required

In the context of speed-limit traffic-sign recognition, the ASIL requirement can vary according to the scenario. The severity of a wrong classification depends on the difference of speed. For instance, if the speed limit of the traffic-sign was 20km/h and the system recognize an 80 km/h traffic-sign, an accident can happen, characterizing a fatal injury severity, in addition it will be more difficult to control a car at high speed. On the other hand, low speed differences (20 km/h to 30 km/h) can hardly cause an accident. It also depends on the environment, that is, if pedestrians or other cars are near [20]. Therefore, if the ASIL B, C were considered, the resultant FIT will be within the required range, but if the worst case is considered (ASIL D), fault mitigation techniques must be employed.

V. CONCLUSION

The fault injection results show that the reuse of hardware resources for MACC operations in deep neural networks or recurrent neural networks, considerably increases its AVF. However, pipelined full-parallel neural networks achieve significantly lower AVF with the cost of high resource utilization. Thus, design exploration techniques and fault injection campaigns must be performed in order to achieve neural networks with balanced area-reliability trade-offs, as well other hardware issues. The radiation experiment results show that the FIT reached is within the ASIL B and C ranges, but not within the ASIL D requiring that fault mitigation techniques are applied. However, the best way to define the exact ASIL is by using a complex simulation that models the environment in which the car is inserted.

REFERENCES

[1] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 241–246.

[2] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

[3] ISO-26262, "ISO 26262-1:2011 preview road vehicles – functional safety – part 1: Vocabulary," Available at: https://www.iso.org/standard/43464.html, 2011, accessed Aug. 23, 2017.

[4] F. L. Kastensmidt, L. Carro, and R. A. da Luz Reis, *Fault-tolerance techniques for SRAM-based FPGAs*. Springer, 2006, vol. 32.

[5] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[6] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2022–2031, 2016.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[8] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *IEEE International Joint Conference on Neural Networks*, 2011, pp. 1453–1460.

[9] F. F. dos Santos, L. Draghetti, L. Weigel, L. Carro, P. Navaux, and P. Rech, "Evaluation and mitigation of soft-errors in neural network-based object detection in three gpu architectures," in *Dependable Systems and Networks Workshop (DSN-W), 2017 47th Annual IEEE/IFIP International Conference on*. IEEE, 2017, pp. 169–176.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[11] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Springer, 1990, pp. 227–236.

[12] I. Lopes, "Convolutional neural network reliability on an apsoc platform a traffic-sign recognition case study," 2017. [Online]. Available: http://hdl.handle.net/10183/171094

[13] S. S. Mukherjee, C. Weaver, J. Emer, S. K. Reinhardt, and T. Austin, "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor," in *Microarchite osium on*. IEEE, 2003, pp. 29–40.

[14] R. Le, "Xilinx soft error mitigation using prioritized essential bits," 2012. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp538-soft-error-mitigation-essential-bits.pdf

[15] S. F. Nowicki, S. A. Wender, and M. Mocko, "The Los Alamos Neutron Science Center spallation neutron sources," *Physics Procedia*, vol. 90, no. Supplement C, pp. 374 – 380, 2017, conference on the Application of Accelerators in Research and Industry, CAARI 2016, 30 October 4 November 2016, Ft. Worth, TX, USA. [Online]. Available: https://doi.org/10.1016/j.phpro.2017.09.035

[16] JEDEC, "Measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductor devices," Arlington, VA, October 2006, JEDEC Standard, JESD89A. [Online]. Available: https://www.jedec.org

[17] I. C. Lopes, F. L. Kastensmidt, and A. A. Susin, "SEU susceptibility analysis of a feedforward neural network implemented in a SRAM-based FPGA," in *Test Symposium (LATS), 2017 18th IEEE Latin American*. IEEE, 2017, pp. 1–6.

[18] J. A. Clemente, W. Mansour, R. Ayoubi, F. Serrano, H. Mecha, H. Ziade, W. El Falou, and R. Velazco, "Hardware implementation of a fault-tolerant Hopfield neural network on FPGAs," *Neurocomputing*, vol. 171, pp. 1606–1609, 2016.

[19] G. L. Nazar and L. Carro, "Fast single-FPGA fault injection platform," in *Defect and fault tolerance in VLSI and nanotechnology systems (DFT), 2012 IEEE international symposium on*. IEEE, 2012, pp. 152–157.

[20] R. Johansson and J. Nilsson, "The need for an environment perception block to address all asil levels simultaneously," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 1–4.