Characterizing Soft-Error Resiliency in Arm's Ethos-U55 Embedded Machine Learning Accelerator

Abhishek Tyagi
Department of Computer Science
University of Rochester
Rochester, NY, USA
atyagi2@ur.rochester.edu

Reiley Jeyapaul
Reliability, Availability, and Serviceability (RAS)
AMD
Austin, TX, USA
reiley.jeyapaul@ieee.org

Chuteng Zhou
Central Technology
ARM Inc
Austin, TX, USA
chu.zhou@arm.com

Paul Whatmough
AI Research
Qualcomm
Boston, MA, USA
pwhatmou@qti.qualcomm.com

Yuhao Zhu
Department of Computer Science
University of Rochester
Rochester, NY, USA
yzhu@rochester.edu

***Abstract*—As Neural Processing Units (NPU) or accelerators are increasingly deployed in a variety of applications including safety critical applications such as autonomous vehicle, and medical imaging, it is critical to understand the fault-tolerance nature of the NPUs. We present a reliability study of Arm's Ethos-U55, an important industrial-scale NPU being utilised in embedded and IoT applications. We perform large scale RTL-level fault injections to characterize Ethos-U55 against the Automotive Safety Integrity Level D (ASIL-D) resiliency standard commonly used for safety-critical applications such as autonomous vehicles. We show that, under soft errors, all four configurations of the NPU fall short of the required level of resiliency for a variety of neural networks running on the NPU.**

We show that it is possible to meet the ASIL-D level resiliency without resorting to conventional strategies like Dual Core Lock Step (DCLS) that has an area overhead of 100%. We achieve so through *selective protection*, where hardware structures are selectively protected (e.g., duplicated, hardened) based on their sensitivity to soft errors and their silicon areas. To identify the optimal configuration that minimizes the area overhead while meeting the ASIL-D standard, the main challenge is the large search space associated with the time-consuming RTL simulation. To address this challenge, we present a statistical analysis tool that is validated against Arm silicon and that allows us to quickly navigate hundreds of billions of fault sites without exhaustive RTL fault injections. We show that by carefully duplicating a small fraction of the functional blocks and hardening the Flops in other blocks meets the ASIL-D safety standard while introducing an area overhead of only 38%.

I. Introduction

Machine learning accelerators, especially those that target Deep Neural Networks (DNNs), are increasingly used in safety-critical applications, such as autonomous vehicles [27], [28], [87] and medical devices [29]. Ensuring reliable and resilient operations have become essential [90]. Among all sources of vulnerabilities, we focus on soft errors [65], [74], [78], which are transient faults induced by radiation or other external factors (e.g., voltage droops) that can compromise the integrity of data and computations within an NPU. This paper focuses on Arm's Ethos-U55 [8] microNPU, a commercial DNN accelerator used primarily for embedded applications. We provide a thorough characterization of U55's resiliency against soft errors and evaluate how U55's resiliency is impacted by a number of commonly used soft-error mitigation techniques.

Using the RTL of U55, we perform a large-scale fault injection campaign (Sec III). We show that U55, across a range of hardware configurations and DNNs, shows a Silent Data Corruption (SDC) rate lower than 0.1×10^{-15} per inference. While exceedingly low and indeed lower than (i.e., satisfies) the Automotive Safety Integrity Level (ASIL) B and C standards, the SDC rate still violates the ASIL-D standard, the most strict form of ASIL. The SDC rate, perhaps unsurprisingly, increases with the scale of the NPU (e.g., MAC array/on-chip SRAM sizes).

We then dive deeper into individual functional blocks in the U55 NPU. We show that different functional blocks in the NPU (e.g., MAC array vs. DMA vs. control block) have inherently different sensitivity toward soft errors: generally the units responsible for managing dataflow and for decoding weights from memory, when experiencing a soft error, could lead to a higher rate of overall system SDC than other hardware structures. Critically, this sensitivity pattern holds under different process nodes but changes significantly depending on whether faults in logic elements are considered.

We then characterize how U55's soft-error resiliency can be improved by common, existing soft-error protection/mitigation techniques (Sec IV). This is an important study because all protection techniques, such as modular redundancy [31], [81] or flop hardening [10], [42], [44], [55], introduce area overhead

[1]. However, we find that different function blocks in U55 have different area-vs-resiliency trade-offs. For instance, the control unit tends to be small but is sensitive to soft errors. Therefore, there exists an optimal protection strategy given an area budget, which is an important figure of merit in embedded applications as U55 is commonly used.

To characterize the area-vs-resiliency trade-off of U55, we present an internal statistical analysis tool that is validated against Arm silicon and that allows us to quickly navigate hundreds of billions of fault sites without exhaustive RTL fault injections. We show that in order to meet the most stringent ASIL-D standard, some form of modular redundancy must be introduced. However, one does not have to duplicate all the function blocks. In particular, Ethos-U55 meets ASIL-D standard when only Traversal Unit (TSU) and Weight Decoder (WD) blocks are duplicated and DMA and MAC Unit blocks have their FFs hardened.

In summary, this paper makes the following contributions:

- To the best of our knowledge, this is the first large-scale resiliency characterization of a commercial NPU based on RTL fault injections. See Sec VI for comparison with prior works in commercial accelerator reliability analysis.
- We report the soft error resiliency of all the key functional blocks of Ethos-U55 NPU; these blocks are representative as they are found in common ML inference processors in the industry. Such reliability analysis helps us understand, at a per functional block level, the overhead-vs-resiliency trade-off of various protection mechanisms and how they affect the overall reliability of the IP.
- We also show that when searching for soft-error detection strategies to meet the highest safety standards under silicon area constraints, it is in the designer's interest to look at a mixture of detection schemes rather than choosing one scheme for the entire IP.
- We describe a fast and faithful resiliency characterization methodology used inside Arm. The methodology combines functional block level RTL fault injection (using Synopsys Z01X [80]) and (RTL-validated) statistical fault analysis (Thales [83]).

II. Background

We first describe the scope of our work (Sec II-A). We then describe the basics of soft-errors (Sec II-B). We describe in detail the architecture and use cases of Arm's Ethos-U55 (Sec II-C). We end the section by discussing the existing methods of soft-error resilience (Sec II-D).

A. Scope and Assumptions

We are interested in characterizing the soft-error reliability of Ethos-U55 [8]. While transient soft-errors can occur anywhere on the chip [21], [22], [44], [51], [61], they are most damaging to logic structures and Flip-Flops (FF); other storage structures are usually protected by error correction codes [13], [77]. In this work, we use FIT rate and SDC rate as metrics to quantify the reliability of the NPU. In the context of DNN accelerators, an SDC is an inference mis-prediction [17], whereas FIT rate not only considers SDCs but crashes as well.

[1]They will introduce power overhead too, but we are not allowed to share detailed power results.

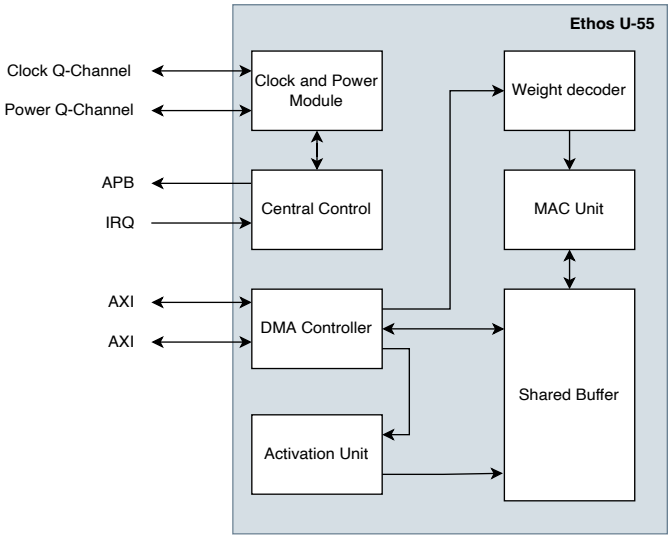


Fig. 1. Ethos-U55 functional blocks diagram [7]

B. Soft-Errors

The Single Event Effects (SEEs) encompass both Single Event Transients (SETs) and Single Event Upsets (SEUs). A SET occurs as a voltage glitch at the output of a combinational gate when an incident particle deposits adequate charge in the gate's sensitive region. Subsequently, the SETs can propagate to sequential cells and induce a change in the stored logic value, leading to a soft error or SEU. Alternatively, soft errors may result from energetic particles directly impacting sequential logic components like flip-flops and latches.

C. Ethos-U55 Overview

Ethos-U55 is Arm's first microNPU designed for the embedded market and meets the requirements for performance with low area and power giving 90% energy reduction with up to 480x performance increase as compared to Cortex-M series alone. U55 is designed to operate while coupled to Cortex-M [2], [3] series processors, which act as controllers.

U55 is being used widely in the market by companies such as Alif Semiconductors for their Ensemble Series of IP. Their E1, E3, E5, and E7 series [75] uses U55 for applications such as wearables, security camera systems, medical devices, and retail applications. NXP has also integrated U55 and U65 [1] with their i.MX [5] series of processors to be used in systems such as driver monitoring systems in the automotive sector. With U55 being utilized in safety-critical applications, it becomes important to characterize the inherent reliability of the design for meeting stringent safety requirements.

Fig 1 showcases the various functional blocks comprising the IP, whose functionality is described below:

- **Direct Memory Access (DMA) Controller**: manages the movement of data from external memory to on-chip memory.

- **Central Controller (CC)**: is responsible for managing the distribution of tasks to all the units in the NPU. We divide the CC in two parts:
  - **Traversal Unit (TSU)**: manages the dataflow to and from the MAC unit to maintain correct execution
  - **Register File (REG)**: stores configuration values in CC.
- **Weight Decoder (WD)**: reads the weights from either on-chip RAM or from an internal buffer and dispatches weights to the MAC array.
- **MAC Array**: carries out the Multiply and Accumulate operations on the input and weights.
- **Activation Unit (AO)**: receives the output feature map from the MAC array and can apply either activation functions or add bias to the read values.
- **Shared Buffer**: is used to store the intermediate output feature maps, input activations, and/or weights. We assume that memory structures (such as SRAM, DRAM) are protected using either ECC and/or parity [59].

D. Existing Soft-Error Resilient Approaches

Dual Modular Redundancy (DMR) duplicates a functional block executing a program and comparing the two sets of outputs. If the checking logic detects any mismatch between the outputs, OS can send a request a re-execution of the program or employ recovery mechanisms. DMR is an effective detection strategy against bit-flips incurred by direct charge injection in a FF and also for latching a SET due to an error injection in combinational logic.

Flop Hardening modifies a FF such that a bit-flip is significantly less likely to take place. While a hardened FF will make it difficult for a particle strike to flip the bit stored in the FF, it would not prevent the FF from latching onto a SET that reaches the input of the FF [72]. The Dual Interlocked Storage Cell (DICE) is a widely utilized custom rad-hard flip-flop design [10]. An alternative approach, Quatro, based on Cascode Voltage Switch Logic (CVSL), has been proposed to achieve better performance at high LET values [42], [44], [55]. Some custom flip-flop designs have also addressed Single Event Transients (SETs). For example, an improved DICE implementation with integrated tunable delay elements for SET filtering was suggested [48].

III. Ethos-U55 Soft Error Characterization

We first describe our methodology for obtaining the characterization data (Sec. III-A). We then describe how the SoC FIT rate is translated to NPUs SDC per inference (Sec. III-B). We then put forward the resiliency data for Ethos-U55 for various configurations and applications (Sec. III-C). We then describe in detail various factors constituting the resiliency behavior of Ethos-U55 (Sec. III-D). And finally, we end the section by discussing the absence of a correlation between area of a functional block and its inherent resiliency (Sec. III-E).

A. Fault Injection Setup

We use RTL fault injection to obtain precise soft-error resiliency data. We use Synopsys Z01X™ [80], which is an industrial-scale RTL fault injection tool, to pick hardware fault sites, represented as $< \textbf{Cycle}, \textbf{FF}, \textbf{BP} >$, to flip.

TABLE I
Workloads used for soft-error resiliency characterization. ASR standards for Automatic Speech Recognition.

Category	Total Parameters	Network	Dataset
Classification	1.1×10^6	CifarNet [37]	CIFAR-10 [50]
Classification	11.2×10^6	ResNet-18 [34]	ImageNet [23]
ASR	23×10^6	Wav2Letter [20]	LibriSpeech [64]

For a single run, the tool picks a single fault-site to flip and performs the RTL simulation that runs the application to the end. For each application, we inject over 2 million faults into the Arms Ethos U55 RTL and run the RTL simulations. This ensures that the resiliency data has less than 1% of error margin with a 99% confidence interval per application.

Tbl. I lists the applications we use to evaluate the reliability of Arms Ethos U55. We choose applications that are utilized in safety-critical scenarios and vary in their sizes, as the size of a network has previously been shown to affect the resiliency of neural networks [41]. CifarNet [37] is a widely used neural network in embedded autonomous platforms [49]. ResNet-18 [34] is utilized as the backbone of the majority of object detection networks deployed in autonomous vehicles (traditional object detection networks are not supported on U55 [9]. We also use Wav2Letter [20], an automatic speech recognition (ASR) network by Meta. ASR has been utilized in safety-critical systems such as aviation to improve flight efficiency and air traffic control to improve communications [6], [47].

B. Translating SoC FIT Rate to NPU SDC per Inference

Synopsys Z01X™compares the output of a fault-injected run with a faultless run and flags an error on output mismatch. For our applications, we consider an SDC to take place when there is a top-1 label mismatch for the image classification tasks and a decrease in word error rate (WER) for the ASR task. We note that using the per-inference misprediction approach, while applies to image classification and ASR tasks this paper focuses on, may not apply to all ML tasks; the notion of SDC, indeed, must be defined on a per-task basis, because different tasks have different task-level masking. For LLMs [24], [82] and generative AI tasks [63], [68], it is still an open question as to how SDCs should be defined.

For applications such as self-driving cars, the overall FIT rate of the chipset should be less than 10 (failures) in 1 billion hours of operation to meet ASIL-D standards for critical components such as airbags and antilock braking [79]. Other ASIL standards are more relaxed. Specifically, for ASIL-B and ASIL-C standards (enforced on brake lights and active suspension [79]) the FIT rate should be less than 100 (failures) per 1 billion hours of operation.

Since NPU is just a fraction of an SoC, the NPU's FIT rate requirement should just be a fraction of that of the SoC. The fraction equals the area of the NPU with respect to the entire SoC, as described by Li et al. [54] and Fidelity [35]. For an SoC such as Tesla FSD Chip [87], an Ethos-U55 will occupy a fraction of the area depending on the MAC configuration

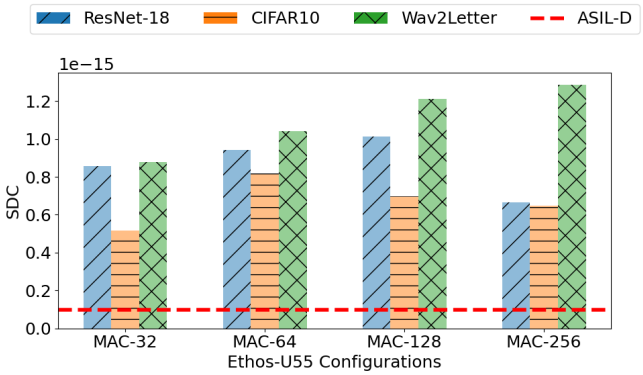


Fig. 2. SDC Rate of Ethos-U55 while running ResNet-18, CifarNet, and Wav2Letter at TSMC 16nm technology node.

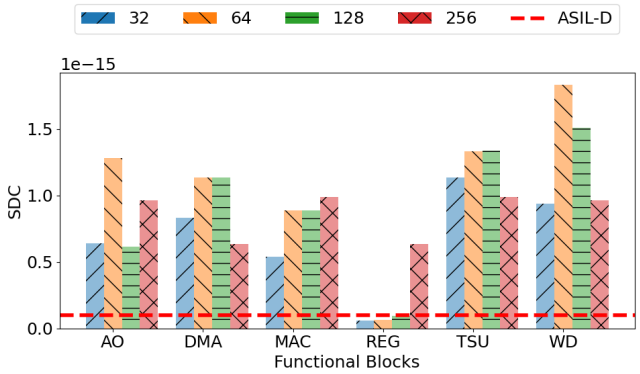


Fig. 3. Functional block SDC contribution for different configurations of Arm Ethos-U55.

(0.12 for MAC-32, 0.14 for MAC-64, 0.17 for MAC-128 and 0.27 for MAC-256). Based on the fraction of area, the FIT requirements for each MAC configuration become 0.12, 0.14, 0.17, and 0.27 failures per 10^9 hours for MAC-32, MAC-64, MAC-128, and MAC-256 respectively.

We use Synopsys Z01X™ to perform RTL-level fault injections, which provides the relative FIT rate *assuming* a fault has occurred. Based on the raw FIT rate data of flip flops [11] and the inference time of a DNN, we can then calculate the absolute FIT rate of the NPU. Using a conservative inference time of 0.3 ms (which accounts for the slowest running application in our experiments), we estimate that the required FIT rate has to be less than 0.1×10^{-15}, 0.12×10^{-15}, 0.15×10^{-15} and 0.23×10^{-15} to be comparable with ASIL-D standards for MAC-32, MAC-64, MAC-128, and MAC-256 configurations respectively.

As mentioned previously, while calculating FIT rate, SDC and crashes both are considered. We show in Sec. III-C, that even while considering just the SDCs for Ethos-U55, its resiliency falls short of ASIL-D standards. Moreover, crashes are easier to detect than SDCs and do not require the same amount of overhead as SDC detection and protection. Due to these reasons, we do not consider crashes in this work and hence can use the FIT rate calculated above as the required SDC rate per inference to meet the ASIL-D standards.

C. How Resilient is Ethos-U55 to Soft Errors?

Fig 2 shows that the SDC rate of the NPU varies significantly with the application it is running as well as the underlying hardware configuration. CifarNet [37] consistently performs best on the resiliency aspects on all the four Ethos-U55 configurations whereas, Wave2Letter [20] is the worst performing application on all the hardware configurations. For the given set of applications, MAC-32 configuration is most resilient to soft-errors.

More interestingly, as per our experiments, Ethos-U55 does not to meet the ASIL-D standards as the reported SDC rate (or the FIT) is $\geq 0.1 \times 10^{-}15$ for all configurations. Therefore, it becomes important to understand what factors affect the resiliency of Ethos-U55. We dissect the NPU and look at the contribution of each individual functional block in the NPU to the overall resiliency of Ethos-U55, for all the different MAC configurations, running the given applications, on a chip fabricated in possible different technology nodes.

D. Factors Shaping Functional Block Resilience

1) Sensitivity to MAC Sizes: The resiliency of NPU functional blocks is sensitive to MAC array size which dictates how many multiply-and-accumulate computations can take place at any point in time. It also dictates the manner in which any large computation is broken down into smaller tasks, which affects the reuse of weights and/or activations. Intuitively, this changes the number and/or position of the faulty neuron in the neural network, eventually resulting in a variation in the MAC SDC rate. Fig 3 shows the variation in SDC contribution of each functional block running CifarNet on four possible configurations of Arm Ethos-U55, proving our intuition correct.

In addition, as shown in Fig 3, the sensitivity of the SDC rate of the MAC unit to changes in MAC fabric size is logical, but it is surprising that other functional blocks also exhibit sensitivity to this variation. We see such a behavior because a change in MAC fabric size changes the task execution chunk size. To adapt to the new chunk size, all other functional blocks in the IP have to modify their execution flow which changes the block ultimately affecting the SDC rate of the block.

2) Sensitivity to Applications: Applications running footprint on the NPU impact the resiliency of a functional block within the IP. Each of the applications have a unique utilization footprint on each of the functional blocks which leads to varying performance on the underlying hardware.

To verify our intuition, we carry out RTL fault injection on Ethos-U55 with three different sets of applications. Fig. 4 shows that blocks like DMA might be less resilient than AO while running ResNet-18, but that might not be the case when CifarNet is running on U55. Therefore, if a protection scheme is devised with just few applications in mind, it might fall short of the required resiliency levels for other applications.

3) Sensitivity to Technology Node: A chosen technology node can dictate the soft-error reliability of a single FF [60] and hence that of functional blocks comprising the FFs. [73] et.al show how FFs Soft Error Rates (SER) have reduced

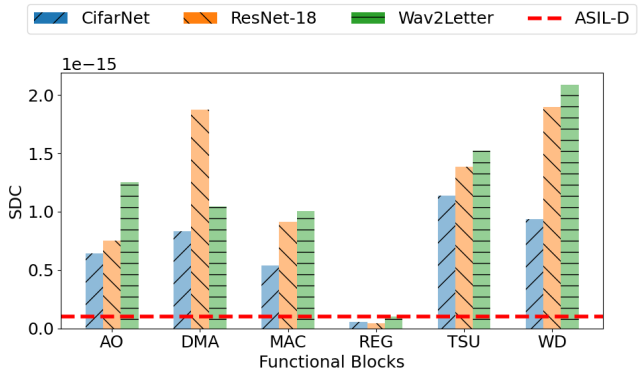


Fig. 4. Block-wise SDC contribution for different applications running on Arm Ethos-U55 [8] with MAC-32 configuration.

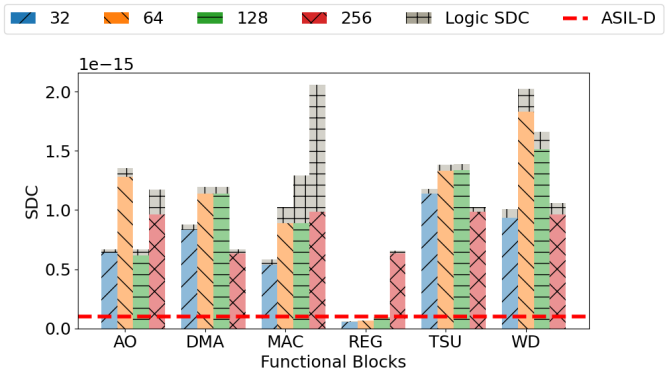


Fig. 6. Block-wise SDC contribution of Arm Ethos-U55 [8], while considering and not considering logic faults for all the four MAC configurations in TSMC 16 nm technology node running ResNet-18.

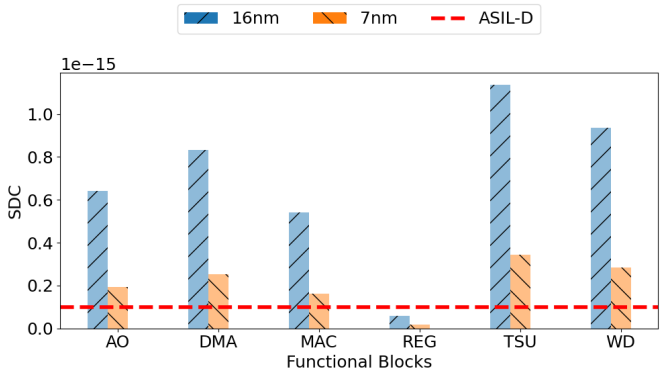


Fig. 5. Variation in the reliability of Arms Ethos U55 [8], for TSMC 16 nm and 7 nm technology nodes for MAC-32 configuration running ResNet-18.

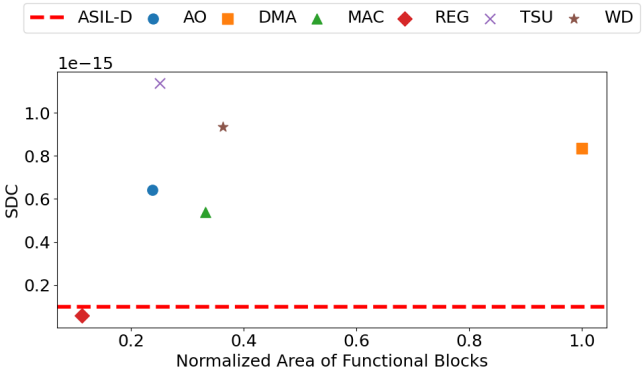


Fig. 7. Block-wise SDC contribution of Arm Ethos-U55 for MAC-32 configuration while running ResNet-18.

drastically with advanced technology nodes, which makes them less susceptible to soft errors. However, it has been observed that the soft error rates of FFs resulting from faults in combinational logic elements have increased as technology nodes advance. Consequently, if an NPU design necessitates fabrication with a newer technology node, a soft-error resilient scheme tailored to the behavior of functional blocks in an older technology node may not be optimal.

To further illustrate these findings, Fig. 5 depicts the variation in the soft-error reliability of Ethos U55 [8] functional blocks in 16nm and 7nm Bulk FinFET technologies. We calculate the SDCs for each functional block as described in Sec. IV-D and use the Soft Error Rate (SER) FIT values for the two technologies as mentioned in prior work [11].

The SDC rates of the functional blocks in 7nm is on average 3.3× less than 16nm. This resiliency behavior over technology nodes is down to factors such as the sensitive area of a storage cell, Critical Charge (Q_{crit}) and Collected Charge (Q_{coll}). For the 7nm FinFET node, the amount of charge collected, i.e. $Q_{coll_{7nm}}$ is less than $Q_{coll_{16nm}}$ which results in higher SER FIT rate for the FF in 16nm technology node [11].

4) Combinational Logic Faults.: Neglecting faults in combinational logic elements leads to an overestimation of reliability. Prior works mostly ignore combinational logic faults because soft errors in FF and memory are present for a longer time whereas a Single Event Transient (SET) generated at the output of a combinational element affects the system only if it gets latched by a FF. Previously, multiple levels of masking [76] has rendered such a case unlikely. However, with technology and voltage downscaling and increasing clock frequency, the total contribution of SETs to Soft Error Rates (SERs) has increased beyond negligible [57].

We show in Fig. 6 the difference in reported SDC contribution of each functional block in Ethos-U55, for the cases when logic faults are and are not considered (See Sec. IV-D for logic fault SDC contribution methodology). Clearly, the reliability of all functional blocks is lower when combinational faults are considered, showing that studying logic faults is warranted. The sensitivity of functional block SDC rate to logic faults consideration adds another variable to the search of an optimal soft-error resilient scheme.

E. Area vs SDC Tradeoff Analysis for Ethos-U55

Our experiments have shown that the resiliency of the NPU is a function of numerous factors interacting in a non-trivial manner. With Ethos-U55 being utilised in safety-critical applications, it becomes crucial to understand how a resilient

version of Ethos-U55 can be designed with existing soft-error mitigation and detection strategies to meet safety standards.

Fig. 7 portrays the spectrum of SDC rates and corresponding area footprint of NPUs functional blocks for MAC-32 configuration running a CifarNet [37]. Functional blocks such as Clock and Power Module (CPM) produces no SDCs (hence are absent from the plot) as the block generates a clock for the IP, not affecting any computation (they lead to crashes). Traversal Unit (TSU) is the most vulnerable block due to its function: TSU is part of CC, which manages the order of execution and traverses the inputs correctly for the output-stationary data-flow.

Critically, the number of FF (or area) of a functional block is in no way an indication of their inherent soft-error resiliency behavior. Intuitively, resiliency of a block is owing to multiple factors such as the design, dataflow, utilization of the block, workload, etc. Prior work [66] has showed this trend for traditional CPUs using the variation in Architectural Vulnerability Factor (AVF) [53] of functional components such as L1 Cache, Physical Register File, and Reorder Buffer.

For instance, AO and TSU, despite having almost the same area, differ drastically in their inherent soft-error resiliency.This is because AO is responsible for applying non-linear activations to the output computed by MAC unit. Therefore, the effect of a bit-flip in AO is likely to be masked by either the non-linear activation function or the approximate nature of neural networks [12], [36]. Similarly, DMA has 2× the area of WD but the two share a similar soft-error resilience. This is because of DMA's sporadic use in U55, as the off-chip data movement is very limited due to high reuse.

The absence of a positive correlation between the block area and its resiliency provides an opportunity to understand which blocks to make more resilient to meet system resiliency requirements under area constraints.

IV. Understanding Ethos-U55's Resiliency Under Existing Soft-Error Mitigation Techniques

We first introduce the SDC rate per inference of the NPU (SDC_{NPU}) formulation (Sec IV-A) using an example hardware with just three fault sites. We then show how SDC_{NPU} can be formulated as a function of SDC contribution of various functional blocks (Sec IV-B). We then introduce how SDC_{NPU} can be estimated accurately and feasibly (Sec IV-C). And lastly, we explain how SDC_{NPU} can be calculated when combinational faults are considered (Sec IV-D).

A. SDC_{NPU} Formulation

With the varying level of functional block level resiliency, the search space for finding an optimal soft-error mitigation scheme is a vast one, which requires solving the following constrained optimization problem,

$$\begin{aligned} &\text{minimize} \quad SDC_{NPU} \\ &\text{subject to} \quad area \leq a_{budget} \end{aligned} \tag{1}$$

where SDC_{NPU} is the SDC rate per inference of the NPU, and can be calculated by particle beam experiments [56], RTL fault injection [83] or modelling the hardware error behavior [58].

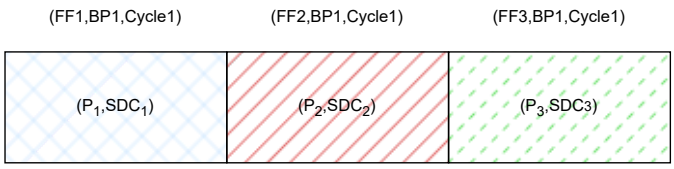


Fig. 8. An illustration of hardware fault-site (i.e., a bit position of a chosen FF at a particular cycle). Each fault site is characterized by its probability to experience a bit flip (P_i) and the SDC rate of the fault site (SDC_i).

Our idea is illustrated in Fig 8, where each box represents a hardware FF or fault-sites (we do not require FF to be in close proximity. In a real system, multiple bit-flips can occur in FF which may or may not belong to one functional block). Each fault site is represented by two parameters:

- P_i: is the probability that a bit-flip occurs at fault-site i at any instant of time and $P_i^{'} = (1 - P_i)$.
- SDC_i: is the SDC rate of the system, when a bit-flip occurs at fault-site i.

The probability P_i is dictated by the raw FIT rate FIT_i of the FF i. Raw FIT rate gives the total number of failures, i.e., bit-flips, expected in the FF in 1 Billion hours of operation. Hence, the probability that a failure can occur at any instant of time (at a cycle), can be written as:

$$P_i = \frac{FIT_i}{2^{20} \times 8 \times 10^9 \times 3600 \times freq} \tag{2}$$

where FIT rate of a FF is given as *FIT/MB* and $freq$ is the frequency of operation of the NPU.

The central question is, what is the SDC rate of the NPU if we know the probabilities and SDC rate of *each* fault site? The crux is to consider all possible events when a particle strike happens at the hardware and model how SDC of each fault-site contributes to SDC_{NPU}. Consider Fig 8 as a representative hardware with three fault sites. If a particle strike happens on this hardware, the resulting behavior of the hardware can be categorized as either of the following three categories:

- **Single Bit-Flip**: In this event, a bit-flip occurs only at one fault site i.e., either at fault-site FS_1, FS_2 or FS_3.
- **Multiple Bit-Flips**: In this case, more than one underlying flip-flops can sustain bit-flips. Which means affected fault-sites could be FS_1FS_2, FS_1FS_3, FS_2FS_3 or $FS_1FS_2FS_3$.
- **No Bit-Flip**: And finally, there could be a case where no bit-flip occurs in the hardware.

We assume that the occurrence or non-occurrence of any event does not affect the probability of other events happening. This is a fair assumption to make as we can see from Equ 2, the probability of a fault occurring in a FF is dependent on the raw FIT rate, which is an intrinsic property of the FF. In that case, with basic probability theory, we can calculate the probability of all possible 8 events in the case of given three fault sites.

- *(Case 1)* Fault occurs at fault-site 1: $P_1P_2^{'}P_3^{'}$
- *(Case 2)* Fault occurs at fault-site 2: $P_1^{'}P_2P_3^{'}$

- *(Case 3)* Fault occurs at fault-site 3: $P_1^{'}P_2^{'}P_3$
- *(Case 4)* Fault occurs at fault-site 1 and 2: $P_1P_2P_3^{'}$
- *(Case 5)* Fault occurs at fault-site 1 and 3: $P_1P_2^{'}P_3$
- *(Case 6)* Fault occurs at fault-site 2 and 3: $P_1^{'}P_2P_3$
- *(Case 7)* Fault occurs at fault-site 1,2 and 3: $P_1P_2P_3$
- *(Case 8)* No fault occurs:$P_1^{'}P_2^{'}P_3^{'}$

Having worked out the probabilities of all possible events, SDC_{NPU} can be written as a sum of the SDC contribution from each possible event. In other words,

$$\begin{aligned} SDC_{NPU} =& P_1P_2^{'}P_3^{'}SDC_1 + P_1^{'}P_2P_3^{'}SDC_2 + P_1^{'}P_2^{'}P_3SDC_3 + \\ & P_1P_2P_3^{'}SDC_{12} + P_1P_2^{'}P_3SDC_{13} + P_1^{'}P_2P_3SDC_{23} + \\ & P_1P_2P_3SDC_{123} + P_1^{'}P_2^{'}P_3^{'}SDC_{none} \end{aligned} \tag{3}$$

where SDC_i is the SDC of the system when only the i^{th} event occurs. If we look at Equ 3, we can simplify it further. Firstly, $SDC_{none} = 0$ as there are no SDCs to be observed when no fault occurs. Secondly, since the order of $P_i \approx 10^{-18}$, we can approximate $P_i^{'} \approx 1$. Lastly, the probability of multiple-bit flips taking place is a product of two or more probabilities, which ranges from 10^{-36} to 10^{-54}. With such low probabilities, it can be safely assumed that the likelihood of such an event occurring is negligible, hence simplifying Equ 3 as:

$$SDC_{NPU} \approx P_1SDC_1 + P_2SDC_2 + P_3SDC_3 \tag{4}$$

B. SDC_{NPU} formulated as functional block SDC

If we look at Equ 3, we can simplify it further. For instance, firstly, $SDC_{none} = 0$ as there are no SDCs to be observed when no fault occurs. Secondly, since the order of $P_i \approx 10^{-18}$, we can approximate $P_i^{'} \approx 1$. And lastly, for *(Case 4)* to *(Case 7)*, the probabilities of the event taking place is a product of two or more probabilities, the order of which ranges from 10^{-36} to 10^{-54}. With such low probabilities, it can be safely assumed that the likelihood of such an event occurring is negligible, hence simplifying Equ 3 as:

$$SDC_{NPU} \approx P_1SDC_1 + P_2SDC_2 + P_3SDC_3 \tag{5}$$

Extending our three fault site example to a hardware with N fault sites, we can generalize Equ 5 as:

$$SDC_{NPU} \approx \sum_{i=1}^{N} P_iSDC_i \tag{6}$$

Intuitively, Equ 6 is just the summation of the product probability of a fault occurring at fault-site i and the SDC rate of the fault-site.

As we are specifically interested in quantifying the reliability of an NPU such as in Fig 1, we can re-write

$$N = N_{CC} + N_{DMA} + N_{AU} + N_{WU} + N_{MAC} + N_{OU} \tag{7}$$

where N_K is the total number of fault-sites in functional block K. We can make use of Equ 7 to re-write SDC_{NPU} as

$$\begin{aligned} SDC_{NPU} \approx & \sum_{i=1}^{N_{CC}} P_{CC}SDC_i + \sum_{j=1}^{N_{DMA}} P_{DMA}SDC_j + \\ & \sum_{k=1}^{N_{AU}} P_{AU}SDC_k + \sum_{l=1}^{N_{WU}} P_{WU}SDC_l + \\ & \sum_{m=1}^{N_{MAC}} P_{MAC}SDC_m + \sum_{n=1}^{N_{OU}} P_{OU}SDC_n \end{aligned} \tag{8}$$

Equ 8 can be interpreted as writing SDC_{NPU} as the contribution of SDC from each functional block. A key observation to make is that for any of the functional blocks, we assume that the probability of a fault occurring in a fault-site within the block is uniform, hence the omission of $P_i, P_j ....P_n$ from Equ 8. This is a fair assumption to make, as for our purposes we assume that a chosen protection/detection scheme is applied to the entirety of a functional block.

C. Estimating SDC_{NPU}

Equ 6 calculates SDC_{NPU} precisely. However, it is impractical to calculate that equation because of the large number of fault sites N. Calculating SDC_i requires running RTL simulations for each fault site over the entire test set (MobileNet [38] has more than 1 billion fault sites).

We use *THALES* [83], a reliability estimation tool validated against RTL fault injection to estimate the SDC_{NPU} as per our formulation in Sec IV-A for all the possible configurations of Ethos-U55. We observe that SDC calculation can be formulated as integrating a discrete function over a finite domain:

$$SDC_{NPU} = \sum_{j=1}^{N} P_jSDC_j, \quad j \in \mathbb{Z} : j \in [1, N] \tag{9}$$

In Equ 9, integrand $f(\cdot)$ does not have an analytical form that can be calculated in practice. In such a case, we propose to solve the integration numerically using Monte Carlo integration [67]. Formally, SDC_{NPU} can be estimated by drawing K independent samples using a Probability Density Function(PDF) and calculate:

$$\overline{SDC_{NPU}} = \frac{1}{K}\sum_{j=1}^{K}\frac{f(X_j)}{PDF(X_j)}, \quad \sum_{j=1}^{N} PDF(X_j) = 1 \tag{10}$$

where $\overline{SDC_{NPU}}$ is the Monte Carlo Estimator of SDC_{NPU}. For our purposes, we chose $PDF = \frac{1}{N}$, where we sample the fault-space uniformly. With the PDF selected for our Monte Carlo Estimator of SDC_{NPU}, we can estimate Equ 6 as

$$\overline{SDC_{NPU}} = \frac{N}{K}\sum_{i=1}^{K} P_iSDCi \tag{11}$$

where K is the total number of independent samples drawn from all the fault sites. As we are interested in the resiliency

characteristics of functional blocks, we can re-write the Equ 11 using Equ 7 as follows:

$$\begin{aligned}\overline{SDC_{NPU}} =& P_{CC} \times \frac{N_{CC}}{K_{CC}} \sum_{i=1}^{K_{CC}} SDC_i + P_{DMA} \times \frac{N_{DMA}}{K_{DMA}} \sum_{j=1}^{K_{DMA}} SDC_j + \\ & P_{AU} \times \frac{N_{AU}}{K_{AU}} \sum_{k=1}^{K_{AU}} SDC_k + P_{WU} \times \frac{N_{WU}}{K_{WU}} \sum_{l=1}^{K_{WU}} SDC_l + \\ & P_{MAC} \times \frac{N_{MAC}}{K_{MAC}} \sum_{m=1}^{K_{MAC}} SDC_m + P_{OU} \times \frac{N_{OU}}{K_{OU}} \sum_{n=1}^{K_{OU}} SDC_n \end{aligned} \tag{12}$$

with $K_{functional-block}$ being the total number of independent samples drawn from the *functional block*.

Equ 12 clearly articulate the respective contributions of each functional block to the overall SDC_{NPU}. Consequently, it serves as a valuable tool for evaluating the potential impact of employing specific soft-error mitigation strategies within individual functional blocks or in combination.

D. Estimating SDC_{NPU} With Logic Faults

The SDC_{NPU} modeling so far ignores logic faults. When logic faults are taken into consideration, the probability that a bit-flip occurs in a FF is higher than that when logic errors are ignored. We model this behavior as an increase in the FIT rate of a FF. Specifically, we can modify the Equ 2 to:

$$P_i = \frac{FIT_i^{'}}{2^{20} \times 8 \times 10^9 \times 3600 \times freq} \tag{13}$$

$$FIT_i^{'} = FIT_i + \alpha \tag{14}$$

where α is the factor by which FIT rate of a FF increases, and according to Seifert et al. [71], is calculated as

$$\alpha = SER_{Comb} \times \frac{10^9}{T} \tag{15}$$

$$SER_{Comb} = flux \times CrossSectionArea \tag{16}$$

where T is the number of hours of operation, SER_{Comb} $(error/hr)$ is the SER from the logic, $flux$ describes the amount of particles that are bombarded per unit area of silicon per unit time $(particles/cm^2/hr)$, and *Cross Section Area* (cm^2) is the logic gate area that is sensitive to charged particles.

An upper bound of SER_{Comb} can be calcualted by using the methodology described by Gill et al. [30], with the values described in Tbl. II. The formulation describes SER_{Comb} as a percentage of nominal latch SER and is calculated as:

$$\begin{aligned}&\frac{\text{SER}_{\text{comb}}}{\text{SER}_{\text{Latch}}}\% \approx \text{LD}_{\text{comb}} * freq * \\ &* \left(FOM * \begin{cases} \frac{(\text{Fanin}^{<\text{d}>+1}-1)}{(\text{Fanin-1})}; & \text{Fanin} > 1 \\ <\text{d}>; & \text{Fanin} = 1 \end{cases} \right)\end{aligned} \tag{17}$$

where Figure of Merit (FOM) is a technology and frequency-dependent parameter. Since we are calculating an upper bound on SER_{Comb}, $LD_{Comb} = 1$ (all logic faults reach a FF to get captured), frequency = 1GHz, and $SER_{Latch} = \frac{1}{2} SER_{FF}$ [30], where SER_{FF} is calculated for each block. (See Sec. III).

Fanin is the average number of inputs to logic gates in the circuit. The higher the fan-in the larger the number of susceptible logic gates at a fixed depth <d> feeding into one equivalent latch. Since we only consider alpha-particles in our study, we use d = 3.5 as mentioned by Gill et al. [30]. We calculate the average Fanin for our four MAC configurations by using the netlist obtained after synthesis and using the all-fanin command available in Synopsys Design Compiler.

TABLE II
CONDITIONS FOR ESTIMATING LOGIC FAULTS AT DIFFERENT TECHNOLOGY NODES. [FLUX($particles/cm^2/hr$) = 0.001].

Technology Node	Voltage (V)	FOM (%) [88]	Cross Section Area (cm^2) [33]	Critical Charge (fC)	FF FIT Rate (FIT/MB)
16 nm	0.75	0.5	3×10^-11	0.9477	50
7 nm	0.7	0.05	0.306×10^-11	0.8059	10

TABLE III
RESILIENCE TECHNIQUES AND ASSOCIATED PARAMETERS FOR ESTIMATING SDC_{NPU}. $FIT_{Hardened}$ IS THE FIT RATE OF THE HARDENED FF AND δ IS AREA OVERHEAD OF THE CHECKING LOGIC.

Type	Technique	Area Cost	$\frac{FIT_{Hardened}}{FIT_{Unhardened}}$	$\frac{\alpha^{'}}{\alpha}$
Redundancy	DMR	100 + δ%	-	0
FF Hardening	Quatro [44]	157%	0.98	1
FF Hardening	TSPC-DICE [43]	46.05%	0.75	0

V. ETHOS-U55 RESILIENCY IMPROVEMENTS UNDER DMR AND FLOP HARDENING

With the analytical model developed in Sec. IV-B, we can now study the impact of conventional protection schemes on U55's resiliency. We start by discussing our experimental setup (Sec V-A) for estimating the resiliency of possible Ethos-U55 configurations. We then show how DMR, flop hardening, and a mix of the two techniques impact the overall resiliency of Ethos-U55 (Sec V-B).

A. Experimental Setup

1) Resilient Configurations: The formulation in Equ 12 is general; the component-wise SDC (SDC_i) and faulty probability (P_1), however, change with the protection scheme listed in Tbl. III, which we describe next.

When **DMR** is used, it is assumed that none of the errors will go undetected from the block. This results in

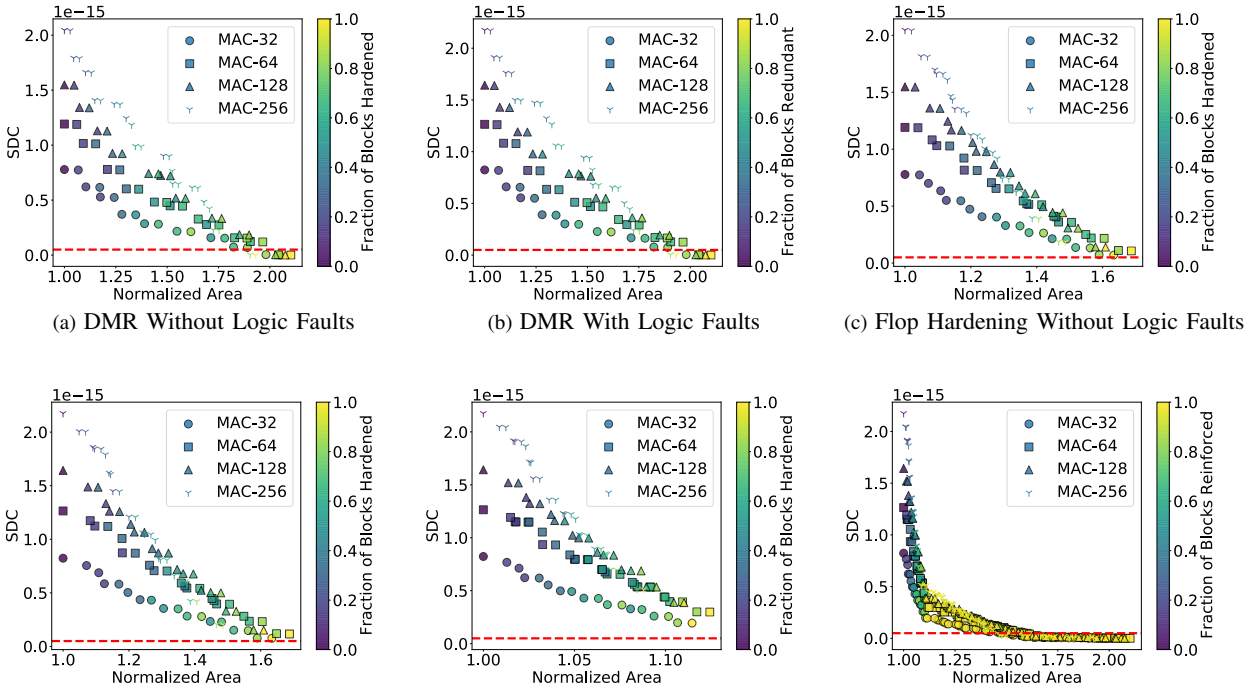
$$SDC_{Block} = 0, \tag{18}$$

and can be used in Equ 12 to estimate SDC_{NPU}.

Similarly, when **FF hardening** is used for flops in a functional block, it reduces the probability of a bit-flip occurring in any FF present in that block as hardening results in a reduction in the raw FIT rate of the FF. This implies that

$$P_{Block}^{'} = \frac{FIT_{Hardened}}{FIT_{Unhardened}} \times P_{Block} \tag{19}$$

where the ratio of the raw FIT rate of a hardened flop to that of an unhardened flop is listed in Tbl. III. The calculated $P_{Block}^{'}$



(a) DMR Without Logic Faults (b) DMR With Logic Faults (c) Flop Hardening Without Logic Faults

(d) Flop Hardening With Logic Faults (e) DMR, Flop Hardening + SET (f) Flop Hardening + SET W/ Logic Faults

Fig. 9. SDC rate per inference vs. area running Wav2Letter at TSMC 16nm and using a) DMR b) DMR with logic faults considered c) Flop hardening d) Flop hardening with logic faults considered e) Flop hardening supporting logic fault elimination and f) using a mixture of DMR, flop hardening, and flop hardening supporting logic fault elimination for the functional blocks in Ethos-U55.

becomes the probability of the block under FF hardening and can replaces P_{Block} in Equ 12 to estimate SDC_{NPU}.

Lastly, when FF hardening with SET elimination is used, it reduces the probability of a bit-flip occurring in the flop, and we can use Equ 14 to estimate the new fault probability as

$$P'_{Block} = \frac{FIT_{Hardened} + \alpha'}{FIT_{Unhardened} + \alpha} \times P_{Block} \tag{20}$$

where α' is the new increase in the raw FIT rate of the FF. The calculated P'_{Block} becomes the probability of the block under FF hardening and can be used in Equ 12 to estimate SDC_{NPU}.

2) Area Evaluation: We use the Synopsys Design Compiler with the TSMC 16nm and 7nm library to obtain the area numbers for the Arm Ethos-U55 base configurations of MAC-32, MAC-64, MAC-128, and MAC-256. We estimate the area overhead of each protection scheme to each configuration in Tbl. III. Specifically, we synthesized the checking logic, which is estimated to have an overhead (δ in Tbl. III) of 7.3%, 8.4%, 10.7% and 13.5% at 16 nm and 5.1%, 6.6%, 7.4% and 10.1% at 7 nm for MAC-32, MAC-64, MAC-128 and MAC-256 respectively.

B. Results

*1) **Ethos-U55 with functional block DMR**:* With functional block level DMR, Ethos-U55 is able to reduce its SDC rate down to ASIL-D levels with around 2× area overhead for all the MAC configurations. Fig. 9a shows SDC rate per inference vs. area of various configurations of Ethos-U55. Each block in U55 can either be left unprotected or be protected by either DMR, flop hardening, or flop hardening supporting logic fault elimination. The heatmap shows out of 6 functional blocks in Ethos-U55, what fraction of blocks are protected. We show only the Pareto optimal configurations. The horizontal dashed line shows the required SDC rate per inference to meet ASIL-D standards as calculated in Sec III-B.

If we look at the bottom right of Fig. 9a, we find that configurations yielding the lowest SDC rates have almost all the functional blocks redundant because with DMR we either make an entire block redundant, or we do not, latter resulting in unprotected blocks contributing to the overall SDC_{NPU}.

DMR can also detect soft-errors that might occur due to faults in combinational logic. And that is why, for the optimal configurations in Fig. 9b no extra silicon is spent as compared to the configurations in Fig. 9a to achieve the lowest possible SDC rate when logic faults are considered.

*2) **Ethos-U55 with flop hardening**:* Ethos-U55 is not able to achieve the required SDC rate to meet ASIL-D standard when block-level flop hardening is employed as shown in Fig 9c. We see that with just around 60% area overhead, flop hardening SDC_{NPU} gets extremely close to the desired SDC rate when all the blocks are hardened. We can understand this behavior by looking at the Equ 12, where we see that SDC_{NPU} is dependent on the individual SDCs of the functional block, along with the probability of a soft-error occurring in that

TABLE IV
U55 CONFIGURATIONS MEETING THE ASIL-D SDC RATE PER INFERENCE REQUIREMENT AT TSMC 16NM AND 7NM TECHNOLOGY NODES. HERE, 0 = NO PROTECTION, 1 = FF HARDENING, 2 = DMR, AND 3 = FF HARDENING SUPPORTING LOGIC FAULT ELIMINATION. BLOCK ORDER: [AO, DMA, MAC, REG, TSU, WD].

MAC Config	16 nm	7 nm
MAC-32	[1, 3, 3, 3, 2, 2]	[3, 3, 3, 3, 2, 2]
MAC-64	[1, 3, 3, 3, 2, 2]	[1, 3, 3, 3, 2, 2]
MAC-128	[2, 3, 3, 3, 2, 2]	[2, 1, 3, 3, 2, 2]
MAC-256	[2, 2, 2, 3, 2, 2]	[2, 2, 2, 3, 2, 2]

block. When flops are hardened in a block, it reduces the probability of a soft-error occurring in the block (in this case by 98%), but is not sufficient to achieve the desired NPU level SDC rate.

Fig 9d shows that just flop hardening is also not enough to achieve ASIL-D level SDC rate when logic faults are taken into consideration. We use Quatro [44] FFs for our analysis which do not offer protection against SETs, i.e. if a SET carrying enough charge, travels to the input of a hardened flop while meeting the setup and hold timing requirements, the FF will capture it as a normal input resulting in a bit flip due to a combinational logic error.

*3) **Ethos-U55 with flop hardening and SET protection**:* When TSPC-DICE [43] FFs are used for mitigating soft-errors in Ethos-U55, we observe that U55 does not meet the ASIL-D level SDC rate for any of the MAC configurations as shown in Fig 9e. Moreover, when compared with the resiliency of Etho-U55 with Qautro [44] FFs we see that Quatro FFs overall achieve a SDC rate much closer to the desired levels as compared to the TSPS-DICE ones. This is because even though TSPC-DICE FFs can mitigate SETs (and hence logic errors), it does not reduce the probability of a fault occurring at a fault site to the same extent as a Quatro FF which results in a poor resiliency performance.

*4) **Ethos-U55 with a mix of DMR, flop hardening, and SET protection**:* In this evaluation, each block can either be left unprotected or use one of either DMR, Quatro FFs, or TSPSC-DICE FFs. Fig. 9f shows that while using a combination of DMR, Quatro FFs, and TSPC-DICE FFs, we can achieve our required resiliency with as low as 53% increase in the silicon. This is a significant improvement upon the configurations that used DMR, Quatro FFs, and TSPC-DICE FFs in isolation. Also, as evident from Fig. 9f, there are multiple different configurations available that have the required level of resiliency, giving designers the option to choose from to optimise for power and performance as well.

We see from Fig. 9f that there is a sharp decrease in the SDC rate as the functional blocks are protected. With just 15% area overhead, we are able to achieve an SDC rate of around $0.3 \times 10^{-}15$, but to reach the desired ASIL-D standard SDC rates, another 30% silicon area is required. We discuss the reasons for this behavior along with our findings from the optimal configurations of Fig. 9f in Sec V-B5.

*5) **What did we learn about Ethos-U55?**:* Tbl. IV lists the Ethos-U55 configurations that achieve ASIL-D level resiliency under area constraints for all four MAC configurations. We see that for all 8 configurations TSU and WD functional blocks have DMR as the preferred technique for mitigating the effects of soft-error. This is expected because TSU and WD blocks have the largest block level SDC rate per inference as shown in Fig. 7 and DMR ensures that these blocks have zero contribution to the SDC_{NPU} in the optimal configurations. We also observe that for MAC-32 and MAC-64 configurations, required resiliency can be achieved without duplicating half of the blocks and hence saving up on the silicon area.

If we look at the optimal configurations for both 16 nm and 7 nm, we see that the blocks that occupy the highest area (DMA in this case) are avoided for both DMR and flop hardening as both techniques have a huge area overhead, except for the case of MAC-256 configuration. In the case of MAC-256, most blocks are made redundant as the SDC contribution of each individual functional block is highest for MAC-256 among all the MAC configurations.

We see that the optimal configurations have similar structures for both 16nm and 7nm technology nodes for all the MAC configurations. However, the overall area overhead of the optimal configuration in 7nm is 21.7% less than that of the same configuration in 16nm owing to the reduction in silicon area due to technology scaling.

VI. RELATED WORK

The main novelty of our work is three-fold. First, we carry out a large-scale, RTL-based, reliability analysis of a commercial NPU that is currently used by a number of customers in the market. Other than works on GPUs [25], [40], [41], [85], most of the reliability analysis is carried out by making use of non-commercial ML inference accelerators. G. Li et al. [54] use accelerators such as Diannao [14] DaDiannao [16], and Eyeriss [15]. Reagan et al. [69] carries out reliability analysis on their in-house accelerator [86], and so do Choi et al. [18] and Zhang et al. [89].

Commercial accelerators such as NVDLA [4] have been characterized for their soft-error reliability with Fidelity [35], and TPUs [46]have been analyzed by Rech et al. [70]. Rech et al. carry out beam experiments to study the reliability, which involves a sophisticated and not readily accessible facility and Fidelity makes use of a framework validated on 15X fewer fault injections (larger the number of fault injections, the higher the accuracy) as compared to our work.

Secondly, we characterize the functional blocks of the NPU for their soft-error resiliency behavior as the functional blocks are common across various designs of ML inference accelerators. Whereas, prior works have primarily focused on two factors related to accelerators: memory [18], [19], [40], [52], and processing element [18], [40], [45].

Lastly, we analyze the effects of heterogeneous soft-error protection schemes, where one selectively applies different protection strategies to different functional blocks. Selective resiliency methods are not new and have been studied at the architecture level [26], [62], application level [32], [59], [84] and hardware level [32], [39]. To that end, our paper presents detailed studies in soft-error resiliency of individual functional

blocks, which are missing in the prior art and can be useful for future studies.

VII. Conclusion

We perform a thorough characterization of the Arm Ethos-U55 NPU, which targets embedded space, against soft errors. We show that while U55 is designed to meet ASIL B/C standards, it does not meet the ASIL D standard. In order to meet the ASIL D standard, we should that a calculated trade-off between area and resiliency must be made. We show that selectively duplicating certain function blocks while hardening FFs in others allows us to meet the ASIL D standard while minimizing the area overhead.

VIII. Acknowledge

We thank anynomous reviewers from ISPASS 2024 for their valuable comments. We thank the Arm Academic Access program for providing us accesses to the Ethos-U55 IP and the associated tools. The research is partially supported by NSF Award #2044963 and a gift grant from Arm.

References

[1] "Arm Ethos-U65, howpublished = https://www.arm.com/products/silicon-ip-cpu/ethos/ethos-u65."

[2] "Cortex-M0, howpublished = https://developer.arm.com/documentation/ddi0432/latest/"

[3] "Cortex-M1, howpublished = https://developer.arm.com/documentation/ddi0413/latest/"

[4] "Nvdla open source project, howpublished = http://nvdla.org/primer.html, note = Accessed: 2018."

[5] "NXP's i.MX 93 Applications Processor Family Powers a New Era of Secure Edge Intelligence, howpublished = https://www.globenewswire.com/news-release/2021/11/09/2329931/0/en/nxp-s-i-mx-93-applications-processor-family-powers-a-new-era-of-secure-edge-intelligence.html"

[6] N. Ahrenhold, H. Helmke, T. Mühlhausen, O. Ohneiser, M. Kleinert, H. Ehr, L. Klamert, and J. Zuluaga-Gómez, "Validating automatic speech recognition and understanding for pre-filling radar labels—increasing safety while reducing air traffic controllers' workload," *Aerospace*, vol. 10, no. 6, p. 538, 2023.

[7] ARM. Arm ethos-u55 npu technical reference manual. [Online]. Available: https://developer.arm.com/documentation/102420/0200/Functional-description/Functional-blocks-

[8] ARM. Arm micronpu ethos-u55. [Online]. Available: https://www.arm.com/products/silicon-ip-cpu/ethos/ethos-u55

[9] ARM-Software. Arm model zoo. [Online]. Available: https://github.com/ARM-software/ML-zoo#object-detection

[10] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron cmos technology," *IEEE Transactions on nuclear science*, vol. 43, no. 6, pp. 2874–2878, 1996.

[11] J. Cao, L. Xu, B. L. Bhuva, S.-J. Wen, R. Wong, B. Narasimham, and L. W. Massengill, "Alpha particle soft-error rates for d-ff designs in 16-nm and 7-nm bulk finfet technologies," in *2019 IEEE International Reliability Physics Symposium (IRPS)*. IEEE, 2019, pp. 1–5.

[12] A. Chan, N. Narayanan, A. Gujarati, K. Pattabiraman, and S. Gopalakrishnan, "Understanding the resilience of neural network ensembles against faulty training data," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, 2021, pp. 1100–1111.

[13] C.-L. Chen and M. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM Journal of Research and development*, vol. 28, no. 2, pp. 124–134, 1984.

[14] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, "Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," *ACM SIGARCH Computer Architecture News*, vol. 42, no. 1, pp. 269–284, 2014.

[15] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[16] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun *et al.*, "Dadiannao: A machine-learning supercomputer," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 2014, pp. 609–622.

[17] Z. Chen, N. Narayanan, B. Fang, G. Li, K. Pattabiraman, and N. DeBardeleben, "Tensorfi: A flexible fault injection framework for tensorflow applications," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2020, pp. 426–435.

[18] W. Choi, D. Shin, J. Park, and S. Ghosh, "Sensitivity based error resilient techniques for energy efficient deep neural network accelerators," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–6.

[19] 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.

[20] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[21] D. A. G. G. de Oliveira, L. L. Pilla, T. Santini, and P. Rech, "Evaluation and mitigation of radiation-induced soft errors in graphics processing units," *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 791–804, 2015.

[22] V. Degalahal, N. Vijaykrishnan, and M. J. Irwin, "Analyzing soft errors in leakage optimized sram design," in *16th International Conference on VLSI Design, 2003. Proceedings.* IEEE, 2003, pp. 227–233.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] F. F. dos Santos, P. F. Pimenta, C. Lunardi, L. Draghetti, L. Carro, D. Kaeli, and P. Rech, "Analyzing and increasing the reliability of convolutional neural networks on gpus," *IEEE Transactions on Reliability*, vol. 68, no. 2, pp. 663–677, 2018.

[26] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Shoestring: probabilistic soft error reliability on the cheap," *ACM SIGARCH Computer Architecture News*, vol. 38, no. 1, pp. 385–396, 2010.

[27] Y. Gan, Y. Qiu, J. Leng, M. Guo, and Y. Zhu, "Ptolemy: Architecture support for robust deep learning," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2020, pp. 241–255.

[28] Y. Gan, P. Whatmough, J. Leng, B. Yu, S. Liu, and Y. Zhu, "Braum: Analyzing and protecting autonomous machine software stack," in *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2022, pp. 85–96.

[29] F. Gertz and G. Fleutsch, "Applications of deep learning in medical device manufacturing," 2020.

[30] B. Gill, N. Seifert, and V. Zia, "Comparison of alpha-particle and neutron-induced combinational and sequential logic error rates at the 32nm technology node," in *2009 IEEE international reliability physics symposium*. IEEE, 2009, pp. 199–205.

[31] R. Giterman, L. Atias, and A. Teman, "Area and energy-efficient complementary dual-modular redundancy dynamic memory for space applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 2, pp. 502–509, 2016.

[32] M. A. Hanif and M. Shafique, "Dependable deep learning: Towards cost-efficient resilience of deep neural network accelerators against soft errors and permanent faults," in *2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 2020, pp. 1–4.

[33] R. Harrington, J. Kauppila, J. Maharrey, T. Haeffner, A. Sternberg, E. Zhang, D. Ball, P. Nsengiyumva, B. Bhuva, and L. Massengill, "Empirical modeling of finfet seu cross sections across supply voltage," *IEEE Transactions on Nuclear Science*, vol. 66, no. 7, pp. 1427–1432, 2019.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[35] Y. He, P. Balaprakash, and Y. Li, "Fidelity: Efficient resilience analysis framework for deep learning accelerators," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2020, pp. 270–281.

[36] L.-H. Hoang, M. A. Hanif, and M. Shafique, "Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1241–1246.

[37] J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4073–4082.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[39] K. Huang, P. H. Siegel, and A. Jiang, "Functional error correction for robust neural networks," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 267–276, 2020.

[40] Y. Ibrahim, H. Wang, and K. Adam, "Analyzing the reliability of convolutional neural networks on gpus: Googlenet as a case study," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*. IEEE, 2020, pp. 1–6.

[41] Y. Ibrahim, H. Wang, M. Bai, Z. Liu, J. Wang, Z. Yang, and Z. Chen, "Soft error resilience of deep residual networks for object recognition," *IEEE Access*, vol. 8, pp. 19 490–19 503, 2020.

[42] S. Jagannathan, T. Loveless, B. Bhuva, S.-J. Wen, R. Wong, M. Sachdev, D. Rennie, and L. Massengill, "Single-event tolerant flip-flop design in 40-nm bulk cmos technology," *IEEE Transactions on Nuclear Science*, vol. 58, no. 6, pp. 3033–3037, 2011.

[43] S. M. Jahinuzzaman and R. Islam, "Tspc-dice: A single phase clock high performance seu hardened flip-flop," in *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*. IEEE, 2010, pp. 73–76.

[44] S. M. Jahinuzzaman, D. J. Rennie, and M. Sachdev, "A soft error tolerant 10t sram bit-cell with differential read capability," *IEEE Transactions on Nuclear Science*, vol. 56, no. 6, pp. 3768–3773, 2009.

[45] X. Jiao, M. Luo, J.-H. Lin, and R. K. Gupta, "An assessment of vulnerability of hardware neural networks to dynamic voltage and temperature variations," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2017, pp. 945–950.

[46] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, 2017, pp. 1–12.

[47] M. Kleinert, H. Helmke, S. Shetty, O. Ohneiser, H. Ehr, A. Prasad, P. Motlicek, and J. Harfmann, "Automated interpretation of air traffic control communication: The journey from spoken words to a deeper understanding of the meaning," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–9.

[48] J. E. Knudsen and L. T. Clark, "An area and power efficient radiation hardened by design flip-flop," *IEEE Transactions on Nuclear Science*, vol. 53, no. 6, pp. 3392–3399, 2006.

[49] J. Kocić, N. Jovičić, and V. Drndarević, "An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms," *Sensors*, vol. 19, no. 9, p. 2064, 2019.

[50] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[51] L. Lantz, "Soft errors induced by alpha particles," *IEEE Transactions on Reliability*, vol. 45, no. 2, pp. 174–179, 1996.

[52] M. Lee, K. Hwang, and W. Sung, "Fault tolerance analysis of digital feed-forward deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5031–5035.

[53] R. Leveugle, A. Calvez, P. Maistri, and P. Vanhauwaert, "Statistical fault injection: Quantified error and confidence," in *2009 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 2009, pp. 502–506.

[54] G. Li, S. K. S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, and S. W. Keckler, "Understanding error propagation in deep learning neural network (dnn) accelerators and applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017, pp. 1–12.

[55] Y.-Q. Li, H.-B. Wang, R. Liu, L. Chen, I. Nofal, S.-T. Shi, A.-L. He, G. Guo, S. Baeg, S.-J. Wen *et al.*, "A quatro-based 65-nm flip-flop circuit for soft-error resilience," *IEEE Transactions on Nuclear Science*, vol. 64, no. 6, pp. 1554–1561, 2017.

[56] Z. Li, C. Elash, C. Jin, L. Chen, S.-J. Wen, R. Fung, J. Xing, S. Shi, Z. W. Yang, and B. L. Bhuva, "Seu performance of schmitt-trigger-based flip-flops at the 22-nm fd soi technology node," *Microelectronics Reliability*, vol. 146, p. 115033, 2023.

[57] N. Mahatme, N. Gaspard, T. Assis, S. Jagannathan, I. Chatterjee, T. Loveless, B. Bhuva, L. W. Massengill, S. Wen, and R. Wong, "Impact of technology scaling on the combinational logic soft error rate," in *2014 IEEE international reliability physics symposium*. IEEE, 2014, pp. 5F–2.

[58] A. Mahmoud, N. Aggarwal, A. Nobbe, J. R. S. Vicarte, S. V. Adve, C. W. Fletcher, I. Frosio, and S. K. S. Hari, "Pytorchfi: A runtime perturbation tool for dnns," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2020, pp. 25–31.

[59] A. Mahmoud, S. K. S. Hari, C. W. Fletcher, S. V. Adve, C. Sakr, N. Shanbhag, P. Molchanov, M. B. Sullivan, T. Tsai, and S. W. Keckler, "Optimizing selective protection for cnn resilience," in *32nd IEEE International Symposium on Software Reliability Engineering, ISSRE 2021*. IEEE Computer Society, 2021, pp. 127–138.

[60] B. Narasimham, S. Gupta, D. Reed, J. Wang, N. Hendrickson, and H. Taufique, "Scaling trends and bias dependence of the soft error rate of 16 nm and 7 nm finfet srams," in *2018 IEEE international reliability physics symposium (IRPS)*. IEEE, 2018, pp. 4C–1.

[61] R. Naseer, Y. Boulghassoul, J. Draper, S. DasGupta, and A. Witulski, "Critical charge characterization for soft error rate modeling in 90nm sram," in *2007 IEEE International Symposium on Circuits and Systems*. IEEE, 2007, pp. 1879–1882.

[62] M. Nikseresht, J. Vankeirsbilck, D. Pissoort, and J. Boydens, "A selective soft error protection method for cots processor-based systems," in *2021 XXX International Scientific Conference Electronics (ET)*. IEEE, 2021, pp. 1–5.

[63] J. Oppenlaender, "The creativity of text-to-image generation," in *Proceedings of the 25th International Academic Mindtrek Conference*, 2022, pp. 192–202.

[64] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[65] G. Papadimitriou and D. Gizopoulos, "Demystifying the system vulnerability stack: Transient fault effects across the layers," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2021, pp. 902–915.

[66] ——, "Avgi: Microarchitecture-driven, fast and accurate vulnerability assessment," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2023, pp. 935–948.

[67] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[68] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.

[69] B. Reagen, U. Gupta, L. Pentecost, P. Whatmough, S. K. Lee, N. Mulholland, D. Brooks, and G.-Y. Wei, "Ares: A framework for quantifying the resilience of deep neural networks," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*. IEEE, 2018, pp. 1–6.

[70] R. L. Rech and P. Rech, "Reliability of google's tensor processing units for embedded applications," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 376–381.

[71] N. Seifert, V. Ambrose, B. Gill, Q. Shi, R. Allmon, C. Recchia, S. Mukherjee, N. Nassif, J. Krause, J. Pickholtz *et al.*, "On the radiation-induced soft error performance of hardened sequential elements in advanced bulk cmos technologies," in *2010 IEEE International Reliability Physics Symposium*. IEEE, 2010, pp. 188–197.

[72] N. Seifert, B. Gill, S. Jahinuzzaman, J. Basile, V. Ambrose, Q. Shi, R. Allmon, and A. Bramnik, "Soft error susceptibilities of 22 nm tri-gate devices," *IEEE Transactions on Nuclear Science*, vol. 59, no. 6, pp. 2666–2673, 2012.

[73] N. Seifert, S. Jahinuzzaman, J. Velamala, R. Ascazubi, N. Patel, B. Gill, J. Basile, and J. Hicks, "Soft error rate improvements in 14-nm technology featuring second-generation 3d tri-gate transistors," *IEEE Transactions on Nuclear Science*, vol. 62, no. 6, pp. 2570–2577, 2015.

[74] N. Seifert, P. Slankard, M. Kirsch, B. Narasimham, V. Zia, C. Brookreson, A. Vo, S. Mitra, B. Gill, and J. Maiz, "Radiation-induced soft error rates of advanced cmos bulk devices," in *2006 IEEE International Reliability Physics Symposium Proceedings*. IEEE, 2006, pp. 217–225.

[75] A. Semiconductor, "Introducing the ensemble and crescendo families of fusion processors and microcontrollers."

[76] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proceedings International Conference on Dependable Systems and Networks*. IEEE, 2002, pp. 389–398.

[77] C. W. Slayman, "Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 397–404, 2005.

[78] V. Sridharan and D. R. Kaeli, "Eliminating microarchitectural dependency from architectural vulnerability," in *2009 IEEE 15th International Symposium on High Performance Computer Architecture*. IEEE, 2009, pp. 117–128.

[79] Synopsys. What is asil? [Online]. Available: https://www.synopsys.com/automotive/what-is-asil.html#a

[80] Synopsys. Z01x functional safety assurance. [Online]. Available: https://www.synopsys.com/verification/simulation/z01x-functional-safety.html

[81] J. Teifel, "Self-voting dual-modular-redundancy circuits for single-event-transient mitigation," *IEEE Transactions on Nuclear Science*, vol. 55, no. 6, pp. 3435–3439, 2008.

[82] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[83] A. Tyagi, Y. Gan, S. Liu, B. Yu, P. Whatmough, and Y. Zhu, "Thales: Formulating and estimating architectural vulnerability factors for dnn accelerators," *arXiv preprint arXiv:2212.02649*, 2022.

[84] Z. Wan, Y. Gan, B. Yu, S. Liu, A. Raychowdhury, and Y. Zhu, "Vpp: The vulnerability-proportional protection paradigm towards reliable autonomous machines," in *Proceedings of the 5th International Workshop on Domain Specific System Architecture (DOSSA)*, 2023, pp. 1–6.

[85] J. Wei, Y. Ibrahim, S. Qian, H. Wang, G. Liu, Q. Yu, R. Qian, and J. Shi, "Analyzing the impact of soft errors in vgg networks implemented on gpus," *Microelectronics Reliability*, vol. 110, p. 113648, 2020.

[86] P. N. Whatmough, S. K. Lee, H. Lee, S. Rama, D. Brooks, and G.-Y. Wei, "14.3 a 28nm soc with a 1.2 ghz 568nj/prediction sparse deep-neural-network engine with> 0.1 timing error rate tolerance for iot applications," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 242–243.

[87] WikiChip. Fsd chip - tesla. [Online]. Available: https://en.wikichip.org/wiki/tesla_(car_company)/fsd_chip

[88] Y. Xiong, N. J. Pieper, A. T. Feeley, B. Narasimham, D. R. Ball, and B. L. Bhuva, "Single-event upset cross-section trends for d-ffs at the 5-nm and 7-nm bulk finfet technology nodes," *IEEE Transactions on Nuclear Science*, 2022.

[89] J. Zhang, K. Rangineni, Z. Ghodsi, and S. Garg, "Thundervolt: enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6.

[90] Y. Zhu, V. J. Reddi, R. Adolf, S. Rama, B. Reagen, G.-Y. Wei, and D. Brooks, "Cognitive computing safety: the new horizon for reliability/the design and evolution of deep learning workloads," *IEEE Micro*, vol. 37, no. 01, pp. 15–21, 2017.