Thermal-Aware Design for Approximate DNN Accelerators

Georgios Zervakis, Iraklis Anagnostopoulos, *Member, IEEE,*
Sami Salamin, *Student, IEEE,* Ourania Spantidi, Isai Roman-Ballesteros,
Jörg Henkel, *Fellow, IEEE,* and Hussam Amrouch, *Member, IEEE*

Abstract—Recent breakthroughs in Neural Networks (NNs) have made DNN accelerators ubiquitous and led to an ever-increasing quest on adopting them from Cloud to edge computing. However, state-of-the-art DNN accelerators pack immense computational power in a relatively confined area, inducing significant on-chip power densities that lead to intolerable thermal bottlenecks. Existing state of the art focuses on using approximate multipliers only to trade-off efficiency with inference accuracy. In this work, we present a thermal-aware approximate DNN accelerator design in which we additionally trade-off approximation with temperature effects towards designing DNN accelerators that satisfy tight temperature constraints. Using commercial multi-physics tool flows for heat simulations, we demonstrate how our thermal-aware approximate design reduces the temperature from 139℃, in an accurate circuit, down to 79℃. This enables DNN accelerators to fulfill tight thermal constraints, while still maximizing the performance and reducing the energy by around 75% with a negligible accuracy loss of merely 0.44% on average for a wide range of NN models. Furthermore, using physics-based transistor aging models, we demonstrate how reductions in voltage and temperature obtained by our approximate design considerably improve the circuit's reliability. Our approximate design exhibits around 40% less aging-induced degradation compared to the baseline design.

Index Terms—Approximate Computing, Deep Neural Networks, Neural Processing Unit, Reliability, Systolic MAC Array, Temperature, Thermal Design, VLSI

1 Introduction

Recent advancements in Neural Networks (NNs) have brought significant breakthroughs in a wide variety of artificial intelligence-based applications. Particularly, Deep NNs (DNNs) have achieved remarkable results in numerous fields, such as speech and image recognition accuracy [1]. An important characteristic of DNNs is their inherent computational intensity, which has been significantly increased as modern DNNs have become deeper, wider, and more complex, in order to further increase accuracy. To that end, there is an increasing need to boost the inference of DNNs. DNN accelerators prevail as the solution to this growing demand for performance due to their specialized hardware that speeds up DNN inference. Thus, DNN accelerators have been rapidly emerging from data centers in the Cloud all the way down to embedded devices in edge computing.

The core arithmetic operation of DNNs during inference is the multiply-accumulate (MAC) operation. DNNs perform millions of MAC operations, particularly on their convolution and fully connected layers [1], [2]. For this very reason, a DNN accelerator is primarily composed of a vast number of MAC units packed in a relatively small area, forming a *systolic MAC array*. A MAC array is able to perform simultaneously tera of operations per second (TOPS) [1]. As an example, the Google Tensor Processing Unit (TPU) consists of a 64K systolic MAC array, allowing a raw throughput of 92 TOPS [1], while Samsung integrates within their devices a Neural Processing Unit (NPU) which comprises 6K MAC units, offering up to 14.7 TOPS [3].

This enormous amount of performed TOPS in DNN accelerators makes them contingent on high on-chip power densities. This results in excessive on-chip temperatures and localized hot spots that eventually form thermal bottlenecks [4]. Thus, DNN accelerators led to the arising of new thermal challenges. High on-chip temperature severely accelerates transistor and circuit aging [5], seriously impacting the reliability and the lifetime of chips. Currently, two approaches are used to address this problem. First, traditional frequency/voltage scaling is applied to control and reduce the power consumption of the DNN accelerator. However, this approach throttles the chip's performance and ends up deteriorating the inference latency and throughput of the accelerator. The second approach is to use advanced cooling solutions, which results in a considerable increase in the total infrastructure power consumption and costs. For instance, liquid-based instead of convection air-based cooling was recently introduced by Google, for the first time, for the cooling of TPU v3 [6]. Unlike other compute-intensive workloads in high performance systems, DNN

- *Georgios Zervakis, Sami Salamin, Isai Roman-Ballesteros, and Jörg Henkel are with Chair for Embedded System (CES), Department of Computer Science at Karlsruhe Institute of Technology (KIT), Karlsruhe 76131, Germany.*
- *Iraklis Anagnostopoulos and Ourania Spantidi are with the School of Electrical, Computer and Biomedical Engineering, at the Southern Illinois University Carbondale 62901, USA.*
- *Hussam Amrouch is with the Chair for Semiconductor Test and Reliability (STAR) in the Computer Science, Electrical Engineering Faculty at the University of Stuttgart, Stuttgart 70569, Germany.*

This work is partially supported by the German Research Foundation through the project "ACCROSS: Approximate Computing aCROss the System Stack." Corresponding Author: Georgios Zervakis (georgios.zervakis@kit.edu) Manuscript received Mars 17, 2021, revised October 7, 2021.

accelerators impose a more profound thermal challenge because the large power consumed by the systolic MAC array is confined in a very small area footprint.

Advanced cooling solutions are not available on mobile devices and thus, novel alternatives are required. In this work, we go beyond these traditional approaches and we utilize the principles of approximate computing as a novel way to manage the temperature of DNN accelerators. Approximate computing has emerged as a design technique that exploits error resilience, in order to trade-off computational accuracy with improvements in delay, area, and/or power consumption [7], [8], [9], [10], [11], [12], [13], [14]. In the past years, the design of approximate multipliers and adders attracted a significant research interest, since they constitute the core building blocks of most error-tolerant application domains, such as image/video processing and machine learning [7], [8], [9], [10], [11], [12]. In particular, approximate multipliers have been widely explored to address the increased computational and energy demands of DNN accelerators [11], [12], [13], [14], [15], [16]. In such works, accurate multipliers are replaced by approximate ones, delivering significant gains in terms of performance and energy consumption. However, all these works do not consider thermal constraints, which is one of the most limiting factors of a DNN accelerator's performance [2]. It is noteworthy that approximate multipliers not only feature reduced power consumption but also reduced area [8], [9], [10], [11], [12]. Hence, despite the fact that power consumption decreases, this is not always the case for the *power density*. As we will demonstrate next, in most cases the power density increases due to the high area reduction. As a result, applying approximate computing in DNN accelerators in a naive manner, i.e., by just replacing accurate multipliers with approximate ones [11], [12], [13], will not always deliver the expected performance and/or energy gains, due to thermal limitations.

In this work, we implement a holistic evaluation framework, from circuit to system, and we elucidate the real impact of approximate computing on DNN accelerators when considering temperature constraints. Moreover, we utilize the proposed framework to implement a thermal-aware approximate DNN accelerator design. Overall, our design methodology leverages the area gains that result from approximation, to scale the MAC array size and further increase throughput. In addition, part of the throughput gain is then traded with voltage scaling in order to significantly decrease power consumption, power density, and on-chip temperature. We demonstrate that approximate computing can be employed to efficiently address the elevated on-chip temperature of systolic MAC arrays and satisfy tight temperature constraints. Experimental results show that our proposed approximate MAC array design consumes 59% less energy compared to the accurate MAC array. In addition, we are able to achieve from 6% up to 23% lower inference latency, while satisfying tight accuracy loss thresholds. Note that our framework and our conducted analysis can be seamlessly extended to any application domain. In this work, we evaluate NN accelerators due to their excessive temperatures that limit their performance [2].

Finally, degradation effects induced by different transistor aging phenomena impose profound challenges on circuit designers to sustain reliability for the entire projected lifetime. In this work, we additionally investigate how introducing approximation in the design allows MAC circuits to operate at lower voltage and temperature, leading to the significant mitigation of the aging-induced degradation represented by the transistor threshold voltage increase.

Our novel contributions are as follows:

(1) This is the first work that evaluates the impact of approximate multipliers on mitigating the excessive on-chip temperatures induced by DNN accelerators during inference. Moreover, we demonstrate that simply replacing accurate multipliers with approximate ones does not necessarily resolve the thermal hotspot.

(2) We implement a thermal-aware design of approximate systolic MAC arrays that leverages approximation to coordinately trade-off area gain, throughput, and voltage. For a marginal accuracy loss (0.44%), our design methodology delivers high temperature reduction (42%), while decreasing inference latency (6%), and delivering very high energy savings (70%) without any area overhead.

(3) We also demonstrate how reductions in temperature and voltage obtained by our thermal-aware approximate design significantly mitigate transistor aging and hence, improve the reliability and lifetime of DNN accelerators.

2 Related works

There has been great interest around approximate computing, with research focusing both on hardware and software levels. Usually, floating-point precision is used during the training phase of NNs. However during the inference phase of NNs, the conducted operations can be quantized from floating-point numbers to narrow integers (such as 8-bit) [1], [17]. By performing this conversion to low-precision numerical representations, fixed point multipliers are used, since they can deliver high inference speedup and energy gains. Significant research interest is shown in the design of approximate fixed point multipliers [8], [9], [10], [11], [12], [13]. For instance, [11], [12] designed approximate multipliers for NN inference. The authors in [11] proposed a compact and energy-efficient multiplier-less artificial neuron. In order to recover accuracy loss caused by approximate multiplier utilization however, this method is based on NN retraining. The authors in [12] introduce approximate multipliers to different convolution layers, however they also apply retraining after approximation. In [13], the authors demonstrate that employing approximate multiplication can improve both the classification accuracy and the energy consumption by using approximate multipliers and NN retraining.

In [18], authors proposed a novel accuracy-reconfigurable stochastic computing in order to manage power and trade-off accuracy versus energy. In [16], the authors present a heterogeneous architecture built with 8-bit approximate multipliers from [8]. [16] reconfigures the applied approximate during inference by selecting a different approximate multiplier per layer and power-gating the unused ones. The work in [16] avoids retraining by applying a fast weight tuning and this layer-wise approximation. In this method, however, there is potentially throughput loss induced by under-utilization of the hardware, and also a heterogeneous design is also

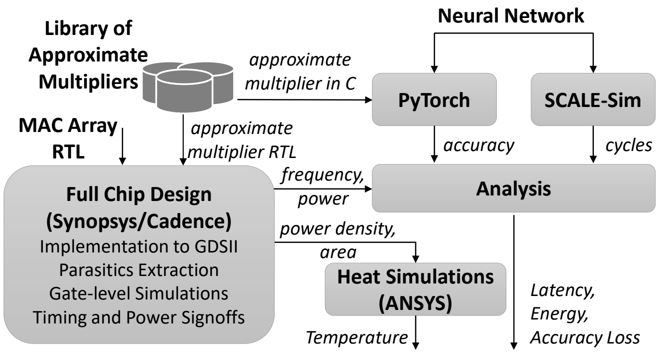

Fig. 1. Our implemented cross-layer evaluation framework to evaluate the temperature-latency-energy-accuracy trade-off of approximate MAC arrays. Throughout the entire evaluation stack, we employ industry level tools and libraries to obtain most precise analysis. Section 3 describes in detail all the sub-modules of our framework illustrated in this figure.

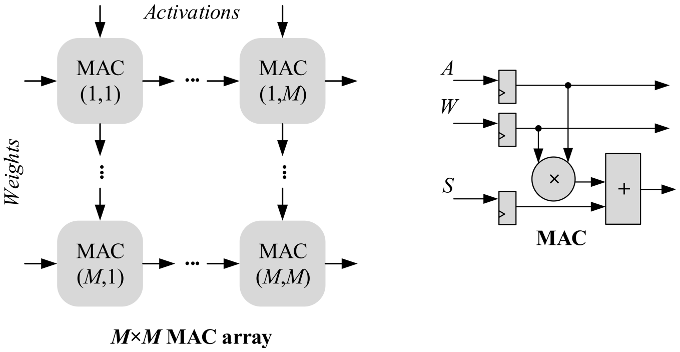

Fig. 2. Overview of a $M \times M$ systolic MAC array similar to the one used in the Google TPU [1] microarchitecture. The examined MAC arrays (i.e., w.r.t approximation) are described in Verilog RTL and then a full-chip design is performed using industry-strength tools.

required. Similarly, the work presented in [7] applies layer-wise approximation too and does so by employing Simulated Annealing to produce approximate circuits with dynamically reconfigurable accuracy at run-time. This strict layer-wise approach is, however, minimizing potential benefits. The proposed method in [15] utilizes approximate MAC units and modifies the multiplier to implement an error compensation technique. Nevertheless, [15] is only evaluated on the LeNet architecture which has a very small (single-digit) number of layers. Such a shallow architecture cannot compare to the number of operations performed by a more recent DNN. In [19], canonic sign digit encoding with truncation is used for the weights representation in order to generate energy-efficient approximate multipliers for NN inference accelerators. However, [19] requires a different circuit implementation for each NN, limiting its applicability. [20] investigated voltage over-scaled (VOS) DNN accelerators. Although VOS is a powerful technique to reduce power density, timing violations due to VOS may lead to circuit instability akin to flip-flop violations. Moreover, VOS errors are non-controlled and usually high in magnitude [10] while logic and algorithmic approximation (as in our work) lead to controllable errors. Finally, while memory features very high energy consumption [21], on-chip memory exhibits significantly less power density than the systolic MAC array and does not contribute to the overall on-chip temperature [2].

The approach proposed in [22] presents an approximate reconfigurable circuit framework that addresses thermal guardbands. However, [22] considers only a very simple IDCT architecture. In [2] a dynamic thermal management technique for DNN accelerators is proposed. On top of frequency scaling and advanced on-chip cooling, [2] applies run-time dynamic precision scaling to further reduce the temperature at the cost of some accuracy loss. A large temperature reduction in [2] is attributed to the use of emerging on-chip superlattice cooling, whereas in our work we focus on traditional approaches.

Distinguish from Existing State of the Art: We are the first to design a temperature-aware approximate systolic MAC array, in which we trade the area gain due to approximation for larger MAC array size and eventually for lower voltage. Hence, we satisfy both tight thermal and latency constraints, while delivering high energy savings.

3 Our Cross-Layer Evaluation Framework

In this section, we describe our implemented framework to evaluate the performance, power, temperature, and inference accuracy of approximate systolic MAC arrays. An abstract overview of our framework is illustrated in Fig. 1. Our evaluation starts from the circuit level where a full-chip design is implemented, and expands up to the system level where we evaluate the accuracy and latency of the NN inference. In addition, we employ multi-physics simulations to obtain precise thermal analysis. All the steps described hereafter are implemented on top of state-of-the-art/industrial tools to ensure accurate analysis and compatibility with existing design flows.

3.1 MAC Array Modeling

In this work, we consider a microarchitecture as implemented in the Google TPU [1], where the core component is a systolic $M \times M$ MAC array. For example, the Google cloud TPUv1 comprises a 256×256 systolic MAC array, while the Google Edge TPU consists of a 64×64 systolic MAC array. The TPU uses 8 bits for the activations and weights. Each MAC unit consists of an 8×8 fixed point multiplier, followed by a K-bit fixed-point adder. The value of K is given by $K = \lceil \log_2(M \times (2^{16} - 1)) \rceil$ to avoid accumulation overflow [14]. The systolic MAC array is illustrated in Fig. 2. We develop the MAC array RTL in Verilog, based on the available optimized arithmetic components that the industrial Synopsys DesignWare library provides. To generate an approximate MAC array, the accurate multiplier of each MAC unit is replaced by an approximate one from our approximate multiplier library (details in Section 3.4).

In this work, we rely on the 14nm FinFET technology node, calibrated with measurements from Intel FinFET production quality [23], to perform the full-chip design. For the chip's design and implementation, we follow the standard EDA tool flows. We start with logic synthesis using the Synopsys Design Compiler to synthesize the different MAC arrays (e.g., accurate and approximate), targeting to maximize performance. This is achieved by applying tight constraints (e.g., zero slack), and by using the `compile_ultra` command that enables the highest efforts in optimization. The physical implementation of the full chip is then done by implementing the chip floorplan and power delivery network. Next, we perform a highly optimized place and route including clock tree synthesis, targeting the maximum performance. The design of the physical chip is done using Cadence tools. Finally, we perform accurate power and delay analysis using signoff tools on the final post-layout circuit netlist including parasitics. The latter ensures very accurate results despite the significantly increased simulation time. The nominal voltage of 0.7V is used. Moreover, for accurate power estimations, we apply a post-layout analysis by employing Mentor QuestaSim to extract the switching activity of the synthesized gate-level netlist, with representative input traces extracted from the inference phase of the examined NNs. The extracted switching activity is then used to estimate accurately the total power consumption of the MAC array using the signoff tool.

3.2 Temperature Chip Modeling

To provide an accurate temperature analysis, we rely on advanced multi-physics simulations for temperature modeling. All the analysis is performed using finite element methods in which the setup is fully implemented in ANSYS, a commercial multi-physics simulation tool [24]. It offers thermal tool flows to accurately model the very complex interactions between the PCB, silicon die, underneath heat flux, heat spreader, heat sink and air convection, etc. Note that all the aforementioned parts have been included in the 3-D modeling of the performed multi-physics simulations to ensure a more realistic temperature modeling. In all thermal simulations, we consider the maximum forced-convection of air, i.e., a Heat Transfer Coefficient (HTC) of $100\text{W}/\text{m}^2\text{K}$ in order to consider the highest capability of convectional air-based cooling. All details related to our thermal modeling and thermal analysis (e.g., dimensions, simulated heat transfer coefficient, material priorities, etc.) are provided in our previous work [2].

To explore the impact of other chip components on the MAC array temperature, we have simulated a chip's die that has a floorplan similar to Google TPU [1]. Here, we assume that the systolic MAC array occupies 24% of the chip area and the on-chip SRAM memory occupies 29%. The rest of the chip forms 47% and it is employed for accumulating the MAC results and controlling the chip (e.g., host interface, misc I/O, PCI interface, etc.). A power density of systolic MAC array of $219.64\text{W}/\text{cm}^2$ (as obtained from the baseline MAC array analysis). We assume the low power-density of the on-chip SRAM memory component to be $8.4\text{W}/\text{cm}^2$. We have estimated this power density from CACTI. Since the background power density (i.e., the remaining of the chip's die) is unknown, we perform the analysis for two different cases. (i) power density of $10\text{W}/\text{cm}^2$, which is similar to the power density of the on-chip memory component. (ii) $20\text{W}/\text{cm}^2$, which is twice the assumed power density of the on-chip memory component. The obtained results from the multi-physics simulations for the MAC array were 138.9°C and 141.5°C, for the aforementioned cases (i) and (ii), respectively. This aligned with our obtained temperature results when studying the MAC systolic array alone (i.e., 139°C for the baseline case "W0A0" shown later in Fig. 5).

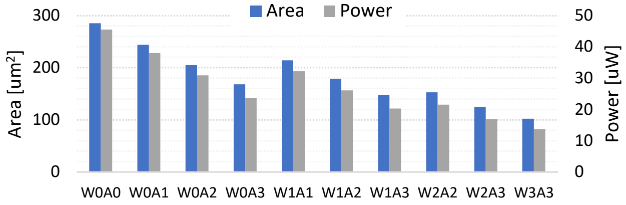

Fig. 3. Post-synthesis area and power evaluation of the multipliers WxAy, $\forall x, y \in [0, 3]$.

3.3 Inference Latency Modeling

In order to accurately model the inference latency, we use the cycle-accurate CNN simulator SCALE-Sim from ARM [25]. In our work, we use the latency of the MAC array, i.e., the product of MAC array operating cycles by MAC array clock period, as a *proxy* of the inference latency. SCALE-Sim takes as input the topology file of the evaluated NN and the dimensions of the systolic MAC array. In the case of DNNs, it could be cumbersome to create such topology files. To this end, we extended SCALE-Sim by creating a wrapper that parses the PyTorch description of the NN, and extracts the configuration of all the 2D convolution (2DConv) and fully connected (FC) layers to generate the required topology file.

3.4 Accuracy Modeling and Approximate Multipliers

We created a library comprising 50 state-of-the-art approximate multipliers from [8], [9], [10], [11], [13], [26] and we described them in C. Next, using the approximate extension of TensorFlow [16], we replaced the multiplication operations with the approximate C multiplication function of each respective approximate multiplier and captured the inference accuracy. Note that 8-bit inference is considered as our baseline [1], [3] None of these approximate multipliers resulted in less than 10% accuracy loss for all the examined NNs (listed in Section 5). For example, for the MobileNet network, only 9 approximate multipliers satisfied this threshold. To increase the achieved accuracy, re-training is mainly used [11] to help NNs adapt to the approximate hardware. Therefore, to avoid time consuming NN retraining, as our case study in this analysis, we consider low-precision multipliers as our approximate multipliers in the MAC array. Hereafter, we denote WxAy the approximate multiplier that reduces by x bits the precision of the multiplier's weight input, and by y bits the precision of the activation input. Next, we apply post-training quantization [27], [28] to map the weights and activations to the multiplier's range and thus, compensate for the induced accuracy loss. To capture the accuracy of the low precision multipliers, we used PyTorch since it provides inherent support for low bit-width quantization. However, that similar results to the ones presented hereafter are expected for [7], [8], [10], [11], [13], if approximation-aware re-training was performed.

For completeness Fig. 3 presents the power and area values of several WxAy multipliers (where W0A0 is the accurate one). The results in Fig. 3 refer to post-synthesis analysis since the multipliers WxAy, $\forall x, y$, are pure combinational circuits and also are considered very small for

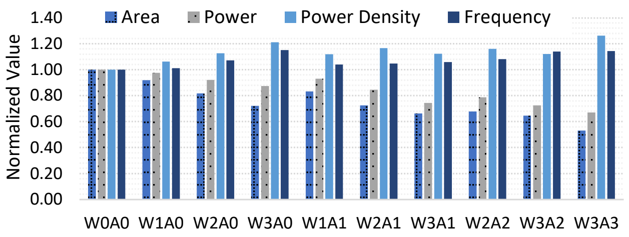

Fig. 4. Hardware evaluation of the approximate 64×64 MAC arrays. The presented values are normalized over the respective ones of the accurate 64×64 MAC array.

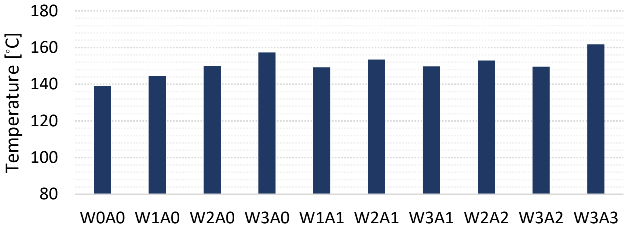

Fig. 5. Temperature evaluation of the 64×64 approximate MAC arrays.

post-layout analysis. Though, as aforementioned, in the following sections, for the hardware evaluation of MAC arrays, we perform power and area analysis on the final post-layout circuit netlist including parasitics, clock tree synthesis, etc. As shown, in Fig. 3, at the multiplier level, the area reduction achieved by the consider low-precision multiplier is 40% ranging from 14% to 64%. Similarly, the power reduction is 45% ranging from 16% to 70%.

4 Circuit-Level Hardware Analysis

4.1 MAC Array Hardware Analysis

As our baseline design, we consider a 64×64 systolic MAC array that uses 8-bit accurate multipliers, i.e., W0A0. Using our evaluation framework (described in Section 3), we did a full chip design of our baseline MAC array. At maximum clock frequency, the obtained power density is 219.64W/cm^2. The latter results in an unsustainable temperature of 139°C as reported by ANSYS. We repeat this procedure for the approximate MAC arrays. For these MAC arrays, we use the previously described approximate multipliers WxAy, where $1 \leq x \leq 3$, and $0 \leq y \leq x$. Although recent research also examines lower precision [29], [30], without loss of generality, $x > 3$ is not evaluated in our work. Fig. 4 depicts the frequency, power, area, and power density of the approximate MAC arrays. The presented values are normalized over the respective ones of the baseline MAC array. The approximate MAC arrays feature significant gains in terms of area, power, and frequency. The average frequency gain is 8% (ranging from 1% to 15%), while the average area reduction is 27% (ranging from 8% to 47%). Similarly, the average power reduction is 17% (ranging from 2% up to 33%). As shown in Fig. 4, both the area and power consumption decrease and thus, the power gains that originate from the usage of approximate multipliers, do not yield a power density decrease. It is noteworthy, that in most cases the decrease in the area is higher than the decrease in power, leading to an increase in power density. Fig. 5 shows the temperature for both the accurate and approximate MAC arrays. As shown in Fig. 5, the temperature remains very high and above the unsustainable value of 131°C. Moreover, in most cases, the temperature of the approximate MAC array is higher than the one of the accurate MAC array (i.e., W0A0). The main finding of this analysis is that, although approximate components can significantly reduce the power consumption, this power gain does not necessarily translate to a temperature reduction. As a result, a naive approximate computing application that just replaces accurate components with approximate ones, might in fact worsen the circuit's thermal profile.

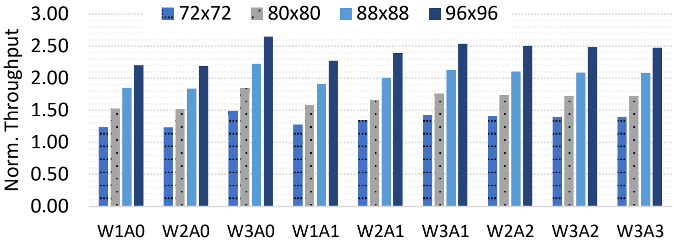

Fig. 6. Throughput evaluation of the approximate MAC arrays for array sizes between 72×72 and 96×96. The values are normalized over the respective ones of the accurate 64×64 MAC array.

4.2 Trading Area Gain for Throughput

As demonstrated in Fig. 4, the approximate MAC arrays show increased area reduction when compared to the baseline accurate MAC array. Leveraging this area gain, we increase the size of the approximate MAC arrays in order to increase their throughput. For the approximate MAC arrays, we evaluate the following array sizes: 72×72, 80×80, 88×88, and 96×96. Note that, the 64×64 MAC array requires a 22-bit adder for the accumulation of the partial sums. On the other hand, a 23-bit adder is required for MAC array sizes between 72×72 and 96×96. As a result, the area of each MAC unit will increase, and its frequency will slightly decrease. Fig. 6 presents the raw throughput of the approximate MAC arrays normalized over the raw throughput of the baseline (i.e., 64×64 W0A0). The raw throughput of a MAC array is calculated by:

$$Throughput = 2 \times M^2 \times frequency. \tag{1}$$

As shown in Fig. 6, the approximate MAC arrays achieve significantly higher throughput than the baseline. The throughput gain is on average 1.95x, raging from 1.24x up to 2.80x. In Fig. 7, we present the corresponding temperature for the approximate MAC array of Fig. 6. For MAC array sizes between 72×72 and 96×96, the architecture of each MAC unit is the same (i.e., a 23-bit adder is required). Thus, considering high MAC array utilization as in CNNs [1], [2], when increasing the size from 72×72 to 96×96, the area and the power will scale in the same manner. Thus, the power density remains almost unaffected, and so does the temperature. Note that SCALE-Sim [25], showed ~100% average utilization for all but two of the the examined NNs (listed in Section 5). Therefore, Fig. 7 reports the temperature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2022.3141054, IEEE Transactions on Computers

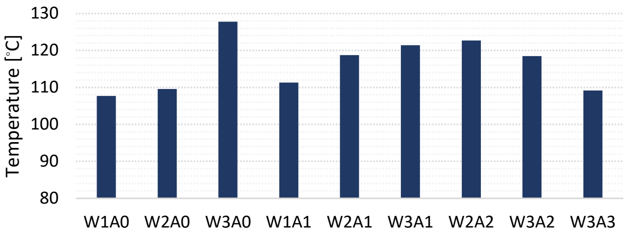

Fig. 7. Temperature evaluation of the approximate MAC arrays for array sizes between 72×72 and 96×96.

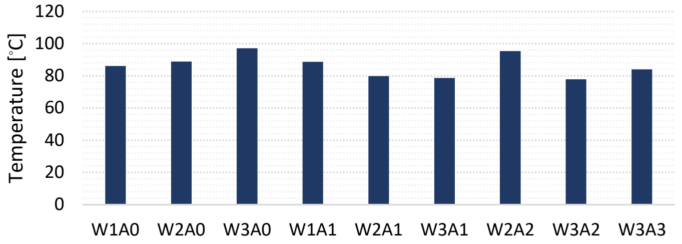

Fig. 8. Temperature evaluation of the approximate MAC arrays at 0.6V for array sizes between 72×72 and 96×96.

value with respect only to the approximate multiplier used. Compared to Fig. 5, the temperature of the approximate MAC arrays decreases. This is explained by the fact that the MAC units required in the 72×72 to 96×96 MAC arrays feature a higher area than power increase compared with the MAC units required in the 64×64 array. As a result, both the power density and temperature decrease. However, the temperature still remains high and above the critical temperature of 105°C as typically defined by Intel [31].

4.3 Trading Throughput Gain for Power

As demonstrated in Fig. 6, the approximate MAC arrays achieve significantly higher throughput compared to the baseline MAC array. Leveraging this high throughput gain, we decrease the voltage value of the approximate MAC arrays from 0.7V to 0.6V to reduce their power consumption[1]. Voltage scaling is very attractive and impactful since both power and power density depend quadratically on the voltage value [32]. Nevertheless, decreasing the voltage value will also decrease the operating frequency and therefore, the throughput. Fig. 8 depicts the temperature of the approximate MAC arrays when decreasing the voltage value to 0.6V. As in Section 4.2, the temperature is almost unaffected by the size (for sizes between 72×72 and 96×96). Compared to the baseline, the approximate MAC arrays feature high temperature reduction that ranges from 42°C up to 61°C. As shown in Fig. 8, the temperature of the approximate MAC arrays is well constrained below the 105°C threshold. Specifically, the temperature of the approximate MAC arrays is at most 97°C.

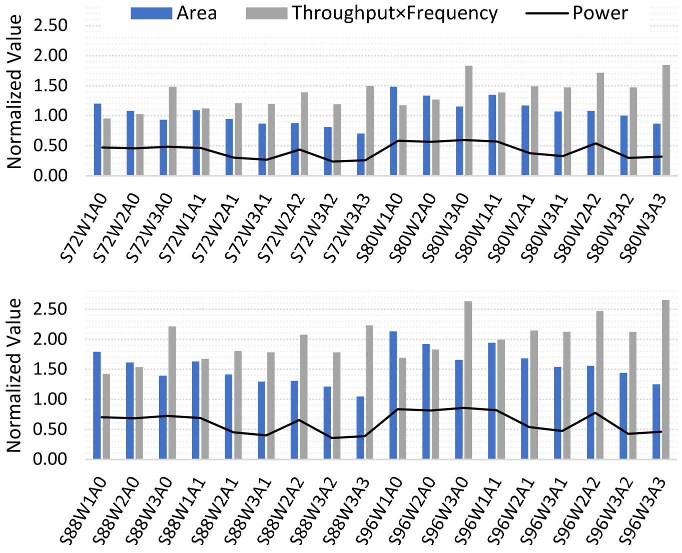

Fig. 9. Hardware evaluation of the approximate MAC arrays at 0.6V for array sizes between 72×72 and 96×96. The values are normalized over the respective ones of the accurate 64×64 MAC array at 0.7V.

TABLE 1
Hardware Characteristics of the selected approximate MAC arrays that feature area less or equal to the baseline

Design	Power Reduction (%)	Normalized Throughput	Temperature (°C)
S72W3A0	52	1.4	97
S72W2A1	70	1.2	80
S72W3A1	73	1.2	79
S72W2A2	56	1.3	95
S80W3A2	70	1.5	78
S80W3A3	68	1.7	84

The number of cycles that a MAC array requires to run the inference phase of an NN is not always proportional to the MAC array size but also depends on the NN architecture. Hence, the inference latency depends on the throughput, as well as the frequency of the MAC array. Due to the voltage decrease, some of the approximate MAC arrays feature lower frequency than the accurate one. Therefore, we use the metric $Throughput \times Frequency$ to evaluate the inference speed of the approximate MAC arrays, when their voltage value is reduced to 0.6V. In Fig. 9, we depict the area, the $Throughput \times Frequency$ metric, and the power consumption of the approximate MAC arrays at 0.6V. All the values in Fig. 9 are normalized values with respect to the corresponding ones of the baseline MAC array at 0.7V. Hereafter, when referring to a MAC array with configuration SMWxAy, we refer to an $M \times M$ MAC array using the WxAy multiplier. As a result, the configuration of our baseline MAC array is S64W0A0. As shown in Fig. 9, the approximate MAC arrays can achieve significantly higher $Throughput \times Frequency$ for a small area overhead. For example, S88W3A3 achieves 2.23x higher $Throughput \times Frequency$ than the baseline for only 4.8% area overhead. For our system-level evaluation next, from Fig. 9 we select only the configurations that exhibit normalized $Throughput \times Frequency$ higher or equal to 1. In addition, in order to conduct a fair evaluation, we only keep the approximate MAC arrays that feature normalized area less than or equal to 1, i.e., *no area overhead*. If more than one configuration is extracted for an approximate multiplier

[1] We also evaluated lower voltage values but the MAC arrays became very slow, resulting in high throughput loss.

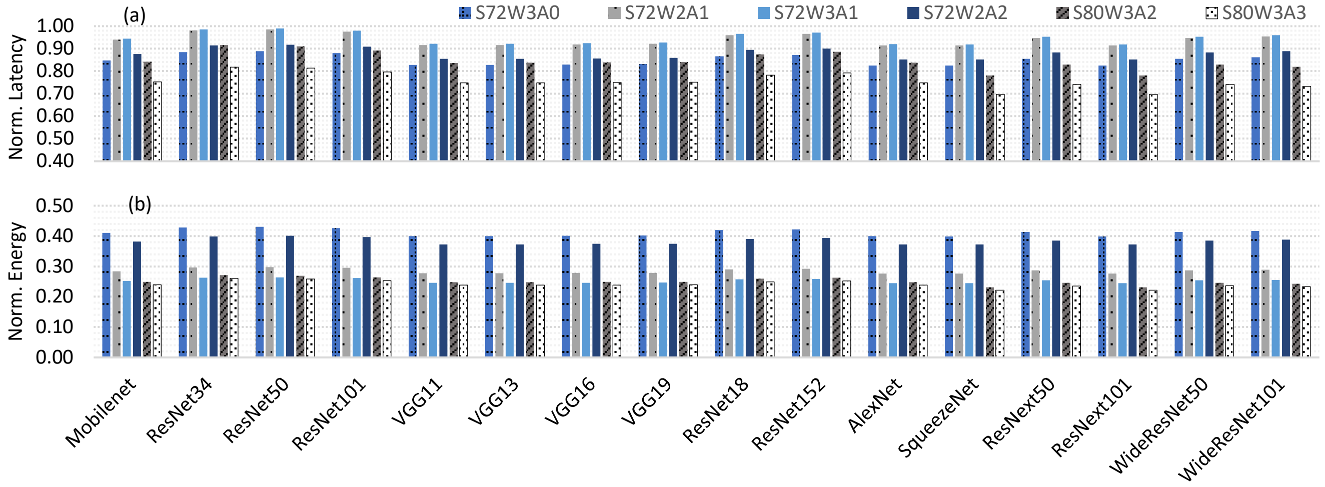

Fig. 10. The a) latency and b) energy evaluation of our approximate MAC arrays for varying NNs. The values are normalized over the respective ones of the accurate 64×64 MAC array at 0.7V.

WxAy, we only keep the MAC array with the maximum size. Table 1 summarizes the power reduction, normalized throughput, and temperature of the selected approximate MAC arrays. Finally, note that applying voltage scaling on the approximate MAC arrays without increasing the MAC array size, results in high throughput loss compared to the baseline. Similarly, increasing the size of the approximate MAC arrays without applying voltage scaling, results in higher power consumption than the baseline. Moreover, in the latter case, the temperature is also higher than the critical 105°C threshold. Hence, both size and voltage scaling are mandatory in the design of our final solution.

5 System-Level Software Analysis

In this section, we consider several CNNs (Table 2) with varying characteristics and evaluate the impact of the approximate systolic MAC arrays at the system level, i.e., inference latency, energy, and accuracy. For all the examined NNs, Fig. 10 presents the latency and energy gains delivered by the approximate systolic MAC arrays presented in Table 1. The energy and latency are normalized over the respective values of the baseline (i.e., S64W0A0 at 0.7V). As shown, the average latency gain is 13% (ranging from 1% up to 30%). Similarly, the energy gain is on average 70% (ranging from 57% up to 78%). Note that, the energy gain is subject to both the latency gain and the lower power consumption of the approximate MAC arrays. Table 2 reports the accuracy loss for each of the examined NNs. For each NN, we also report the number of convolutional and fully connected layers. The accuracy loss is calculated with respect to the accuracy achieved by the baseline (i.e., when using 8bit inference: W0A0). As aforementioned the methods of [27], [28] (that do not require retraining) are used for CNN quantization. For each approximate MAC array, Fig.11 summarizes the average latency and energy gains as well as the corresponding average accuracy loss. As shown, S72W2A1 achieves 71.5% and 5.8% average energy and latency gains respectively, for an average accuracy loss of 0.44%. The respective values for S80W3A2 are 75% and 15% average energy and latency gains, for a 1.95% average accuracy loss. The temperature of S72W2A1 and S80W3A2 is 80°C and 78°C respectively, while the temperature of the baseline is 139°C. Therefore, our designed approximate MAC arrays not only deliver high performance at a low energy cost, but also satisfy strict temperature and accuracy loss constraints. In our analysis, we consider maximum air cooling capability for both the baseline and the approximate MAC arrays. Hence, the baseline, to achieve operation below 105°C, must decrease significantly its operating frequency (i.e., high throughput/latency loss), or it must employ a more sophisticated cooling mechanism (e.g., liquid cooling) that will increase the cooling cost (e.g., energy). As a result, the approximate MAC arrays will achieve even higher latency

TABLE 2
Accuracy evaluation on Cifar100 and ImageNet. Accuracy loss (in %) is reported w.r.t. the accuracy achieved when using W0A0.

CNN (#layers)	W3A0	W2A1	W3A1	W2A2	W3A2	W3A3
			Cifar100			
Mobilenet (35)	1.1	0.6	1.1	0.6	1.4	2.4
ResNet34 (34)	0.9	0.4	1.0	0.3	0.8	1.1
ResNet50 (50)	3.0	0.1	2.9	0.1	2.9	3.0
ResNet101 (101)	2.5	0.4	2.5	0.4	2.6	2.9
VGG11 (11)	1.2	0.5	1.3	1.5	2.1	6.4
VGG13 (13)	0.8	0.2	0.9	0.8	1.6	5.2
VGG16 (16)	1.1	0.6	1.3	1.2	1.8	7.3
VGG19 (19)	2.1	0.7	2.4	1.6	3.2	9.3
			ImageNet			
ResNet18 (18)	0.8	0.3	0.8	0.5	1.2	1.3
ResNet152 (152)	1.1	0.3	1.1	0.3	1.2	1.4
AlexNet (8)	0.8	0.4	1.0	0.5	1.0	1.5
SqueezeNet (26)	3.5	0.5	3.6	1.5	3.7	5.2
ResNext50 (50)	2.9	1.0	3.2	1.1	3.0	3.6
ResNext101 (101)	2.1	0.8	2.0	0.9	2.0	2.4
WRN50 (50)	0.8	0.1	0.7	0.2	1.0	1.3
WRN101 (101)	1.3	0.2	1.3	0.4	1.6	2.0
Average (all)	1.62	0.44	1.69	0.75	1.95	3.52

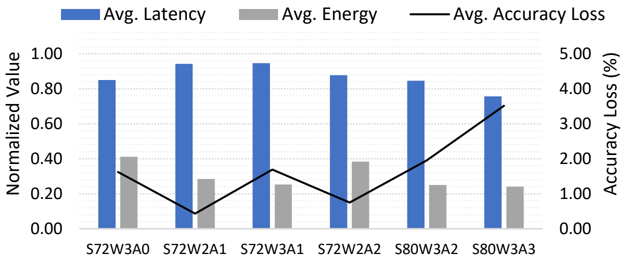

Fig. 11. Latency/Energy vs. accuracy trade-off of our approximate MAC arrays. The latency and energy values are normalized over the respective ones of the accurate 64×64 MAC array at 0.7V.

and/or energy gains. For example, the maximum operating frequency at which the baseline satisfies the 105°C, results in a 21% latency increase.

6 Comparison with State of the Art

In [2], PFS-TE is proposed to mitigate the excess temperatures of systolic MAC arrays. PFS-TE employs three techniques to reduce temperature: frequency scaling, advanced on-chip cooling using thermoelectric coolers [4], and dynamic precision scaling. Advanced on-chip cooling integrates a superlattice thermoelectric (TE) cooling within the chip packaging, and provides very efficient cooling for localized hot-spots with the capability to cool heat fluxes up to 1300W/cm^2 [33], [34]. When a current flows through a superlattice TE, a thermal gradient is created between the upper and lower surfaces due to the Peltier effect, which enables strong heat-pumping [2]. As the current that flows through a superlattice TE increases, the cooling capability of the TE increases and the temperature decreases significantly. Nevertheless, the chip's total power consumption also increases considerably [2]. This power increase is the associated cooling cost of the advanced on-chip cooling.

For our evaluation, we target high performance (i.e., maximum frequency). In addition, we set the accuracy loss constraint to 1% since we are aiming for high accuracy. We consider two temperature thresholds, 105°C and 90°C. The former is the critical temperature as typically defined by Intel [31], while the latter is a typical temperature constraint for mobile devices [35]. Table 3 reports the latency, energy consumption, and accuracy achieved by [2] and our S72W2A1 for several CNNs trained on Cifar100 or ImageNet. Note that, S72W2A1 is selected since in Table 2 it achieves up to 1% accuracy loss for all the considered NNs. The temperature of S72W2A1 is 80°C and thus, satisfies both temperature constraints. PFS-TE [2] achieves the same latency for both temperature constraints, i.e., it enables operation at maximum frequency. However, the energy consumption of PFS-TE increases from 105°C to 90°C. This is explained by the fact that [2] needed to increase its cooling capability by increasing the input current of the employed thermoelectric cooler in order to sustain the maximum frequency (i.e., lowest latency) at a lower temperature threshold. Hence, the cooling cost (i.e., power consumption) and thus the energy consumption increased. As shown in Table 3, S72W2A1 significantly outperforms [2]. S72W2A1 achieves on average 7% lower latency for both temperature constraints. Moreover, S72W2A1 achieves 70% and 78% lower energy consumption than [2] for the 105°C and 90°C constraints, respectively. Note that PFS-TE is implemented exactly as described in [2], including the dynamic precision scaling. Actually, the dynamic precision scaling is the source of the accuracy loss of PFS-TE reported in Table 2. In addition, to apply [2], we performed an exhaustive exploration over the entire design space defined by cooling cost, frequency scaling, and precision scaling to identify the configuration that i) satisfies the temperature constraint, ii) features accuracy loss less than 1%, and iii) achieves the highest performance.

TABLE 3
Comparison against state-of-the-art for 105℃ and 90℃ temperature constraints

Neural Network	Method	Energy [mJ]	Latency [ms]	Accuracy Loss [%]
Mobilenet	S72W2A1	0.78	1.11	0.60
	[2] @ 105°C	2.46	1.18	0.60
	[2] @ 90°C	3.35	1.18	0.60
ResNet34	S72W2A1	0.82	1.17	0.40
	[2] @ 105°C	2.48	1.19	0.30
	[2] @ 90°C	3.38	1.19	0.30
ResNet101	S72W2A1	0.47	0.67	0.10
	[2] @ 105°C	1.43	0.69	0.40
	[2] @ 90°C	1.94	0.69	0.40
VGG11	S72W2A1	2.28	3.25	0.50
	[2] @ 105°C	8.32	3.55	0.40
	[2] @ 90°C	12.27	3.55	0.40
VGG19	S72W2A1	2.49	3.55	0.70
	[2] @ 105°C	9.03	3.85	0.40
	[2] @ 90°C	13.32	3.85	0.40
ResNet18	S72W2A1	0.35	0.49	0.27
	[2] @ 105°C	1.07	0.51	0.31
	[2] @ 90°C	1.45	0.51	0.31
ResNet152	S72W2A1	2.03	2.89	0.30
	[2] @ 105°C	6.23	2.99	0.51
	[2] @ 90°C	8.47	2.99	0.51
AlexNet	S72W2A1	1.09	1.55	0.37
	[2] @ 105°C	3.53	1.70	1.00
	[2] @ 90°C	4.80	1.70	1.00
ResNext101	S72W2A1	12.51	17.84	0.75
	[2] @ 105°C	40.62	19.51	0.60
	[2] @ 90°C	55.22	19.51	0.60
WRN50	S72W2A1	2.09	2.99	0.10
	[2] @ 105°C	7.39	3.15	0.51
	[2] @ 90°C	10.90	3.15	0.51
Average	S72W2A1	2.49	3.55	0.41
	[2] @ 105°C	8.26	3.83	0.50
	[2] @ 90°C	11.51	3.83	0.50

7 Impact of Approximation on Reliability

Transistor aging is one of the major reliability challenges and as described in Section 1, higher on-chip temperature largely accelerates transistor and circuit aging. Aging phenomena like BTI and HC increase the threshold voltage of the transistor (V_{TH}), which leads to considerable performance degradation of the entire circuit, and aging-induced timing errors could be catastrophic for the accuracy of DNN accelerators [36], [37]. Transistor aging strongly depends on the

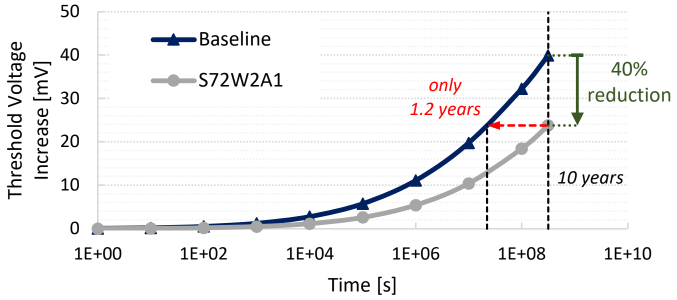

Fig. 12. The increase in the transistor threshold voltage (V_{TH} due to BTI and HCI aging mechanisms from the beginning until the end of circuit's lifetime (10 years). Compared to the baseline design, our approximate design largely mitigate the impact of aging due to obtained reduction in voltage (0.6V instead of 0.7V) and temperature ($80°$C instead of $139°$C).

operating voltage and temperature. Therefore, our proposed approximation is expected to improve the reliability of MAC array circuits (i.e., mitigating the increase in V_{TH} during the circuit's lifetime), because they allow the circuit to operate at a lower voltage as well as to exhibit a lower temperature. We present in Fig. 12 the increase in the transistor V_{TH} from the beginning until the end of the projected lifetime (10 years). The reliability analysis is performed using physics-based aging models [38], [39], [40]).

In this analysis (Fig. 12), we select, as before, our S72W3A1 design to be compared against the baseline design, since it provides a very large reduction in temperature (features $80°$C) for negligible accuracy loss. As shown in Fig. 12, our design strongly mitigates the aging-induced ΔV_{TH}, resulting in a 40% reduction. In practice, for the same lifetime of 10 years of operation, the ΔV_{TH} induced by our design is around 23.8mV, compared to 39.94mV induced by the baseline design. Furthermore, for the same degradation induced by our design after 10 years of operation ($\Delta V_{TH} = 23.8$mV), the baseline design reaches the same degradation after merely 1.2 years of operation. This demonstrates the large improvement in the circuit's reliability obtained by our proposed approximation.

8 Conclusion

With the recent and rapid advancements in the area of machine learning, Deep Neural Networks (DNNs) have become the driving force both for Cloud and edge computing domains. To realize that, dedicated DNN accelerators are employed in order to satisfy the increased computational demands of DNNs and accelerate the inference execution. The heart of DNN accelerators is an array of Multiply-accumulate (MAC) units. However, as MAC units are packed together within a relatively confined area, the chip is subject to increased power density and temperature. This is the first work that utilizes the concept of approximate computing to reduce the temperature on DNN accelerators and investigates the impact of approximate multipliers when considering thermal constraints. We demonstrate that just replacing accurate multipliers with approximate ones, as usually implemented by other state-of-the-art approaches, does not decrease the temperature to acceptable levels, despite the obtained power reduction. Our analysis shows that our thermal-aware approximate design, which coordinately trades-off accuracy, area gain, throughput, and voltage, efficiently addresses tight temperature constraints for marginal inference accuracy loss. At the same time, the performance of the DNN accelerator is improved, and its energy consumption is highly reduced. Particularly, for only a small accuracy drop of 0.44% and without any area overhead, our design methodology reduces energy consumption by 70%, decreases temperature down to $80°$C, and improves the inference time by 6%. Finally, we demonstrate how reductions in voltage and temperature obtained by our approach result in a considerable improvement in the circuit's reliability due to the large mitigation of degradation effects induced by transistor aging mechanisms (about 40%).

Acknowledgement

We would like to thank Souvik Mahapatra and his team from IIT Bombay, India, for the valuable support in transistor aging modeling.

References

[1] N. P. Jouppi *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.

[2] H. Amrouch, G. Zervakis, S. Salamin, H. Kattan, I. Anagnostopoulos, and J. Henkel, "Npu thermal management," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3842–3855, 2020.

[3] J. S. Park *et al.*, "9.5 a 6k-mac feature-map-sparsity-aware neural processing unit in 5nm flagship mobile soc," in *2021 IEEE Int. Solid- State Circuits Conference (ISSCC)*, vol. 64, 2021, pp. 152–154.

[4] F. Kaplan, M. Said, S. Reda, and A. K. Coskun, "Locool: Fighting hot spots locally for improving system energy efficiency," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 4, pp. 895–908, 2020.

[5] H. Amrouch, S. B. Ehsani, A. Gerstlauer, and J. Henkel, "On the efficiency of voltage overscaling under temperature and aging effects," *IEEE Trans. Comput*, vol. 68, no. 11, pp. 1647–1662, 2019.

[6] Google. [Online]. Available: https://www.datacenterknowledge.com/google-alphabet/google-brings-liquid-cooling-data-centers-cool-latest-ai-chips

[7] G. Zervakis, H. Amrouch, and J. Henkel, "Design automation of approximate circuits with runtime reconfigurable accuracy," *IEEE Access*, vol. 8, pp. 53 522–53 538, 2020.

[8] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "Evoapproxsb: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods," in *Design, Automation & Test in Europe Conference & Exhibition*, 2017, pp. 258–261.

[9] S. Hashemi, R. I. Bahar, and S. Reda, "Drum: A dynamic range unbiased multiplier for approximate applications," in *International Conference on Computer-Aided Design*, 2015, pp. 418–425.

[10] G. Zervakis, K. Koliogeorgi, D. Anagnostos, N. Zompakis, and K. Siozios, "Vader: Voltage-driven netlist pruning for cross-layer approximate arithmetic circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 6, pp. 1460–1464, 2019.

[11] S. S. Sarwar, S. Venkataramani, A. Ankit, A. Raghunathan, and K. Roy, "Energy-efficient neural computing with approximate multipliers," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 2, pp. 1–23, 2018.

[12] V. Mrazek, S. S. Sarwar, L. Sekanina, Z. Vasicek, and K. Roy, "Design of power-efficient approximate multipliers for approximate artificial neural networks," in *Proceedings of the 35th International Conference on Computer-Aided Design*, 2016, pp. 1–7.

[13] M. S. Ansari, V. Mrazek, B. F. Cockburn, L. Sekanina, Z. Vasicek, and J. Han, "Improving the accuracy and hardware efficiency of neural networks using approximate multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 2, pp. 317–328, 2020.

[14] Z. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, "Weight-oriented approximation for energy-efficient neural network inference accelerators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, pp. 1–14, 2020.

[15] M. A. Hanif, F. Khalid, and M. Shafique, “Cann: Curable approximations for high-performance deep neural network accelerators,” in *Design Automation Conference (DAC)*, 2019, pp. 1–6.

[16] V. Mrazek, Z. Vasicek, L. Sekanina, M. A. Hanif, and M. Shafique, “Alwann: Automatic layer-wise approximation of deep neural network accelerators without retraining,” in *International Conference on Computer-Aided Design*, Nov 2019, pp. 1–8.

[17] J. Yang *et al.*, “Quantization networks,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

[18] S. Yu, H. Zhou, S. Peng, H. Amrouch, J. Henkel, and S. X.-D. Tan, “Run-time accuracy reconfigurable stochastic computing for dynamic reliability and power management: Work-in-progress,” in *Int. Conf. on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*, 2020, pp. 1–3.

[19] M. Riaz *et al.*, “Caxcnn: Towards the use of canonic sign digit based approximation for hardware-friendly convolutional neural networks,” *IEEE Access*, vol. 8, pp. 127 014–127 021, 2020.

[20] 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 *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018, pp. 1–6.

[21] M. Horowitz, “1.1 computing’s energy problem (and what we can do about it),” in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14.

[22] B. Boroujerdian, H. Amrouch, J. Henkel, and A. Gerstlauer, “Trading off temperature guardbands via adaptive approximations,” in *Int. Conf. Computer Design*, 2018, pp. 202–209.

[23] S. Natarajan *et al.*, “A 14nm logic technology featuring 2nd-generation finfet, air-gapped interconnects, self-aligned double patterning and a 0.0588 µm2 sram cell size,” in *2014 IEEE International Electron Devices Meeting*, 2014, pp. 3.7.1–3.7.3.

[24] Ansys Mechanical Enterprise. [Online]. Available: https://www.ansys.com/products/structures/ansys-mechanical-enterprise

[25] A. Samajdar, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna, “Scale-sim: Systolic cnn accelerator simulator,” *arXiv preprint arXiv:1811.02883*, 2018.

[26] I. Hammad, L. Li, K. El-Sankary, and W. M. Snelgrove, “CNN Inference Using a Preprocessing Precision Controller and Approximate Multipliers With Various Precisions,” *IEEE Access*, vol. 9, pp. 7220–7232, 2021.

[27] B. Jacob *et al.*, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2704–2713.

[28] R. Banner, Y. Nahshan, and D. Soudry, “Post training 4-bit quantization of convolutional networks for rapid-deployment,” in *Neural Information Processing Systems (NeurIPS)*, 2019.

[29] J. Fang, A. Shafiee, H. Abdel-Aziz, D. Thorsley, G. Georgiadis, and J. H. Hassoun, “Post-training piecewise linear quantization for deep neural networks,” in *European Conference on Computer Vision (ECCV)*, 2020, pp. 69–86.

[30] Y. Nahshan *et al.*, “Loss aware post-training quantization,” *arXiv preprint arXiv:1911.07190*, 2019.

[31] Intel. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/joule-products/intel-joule-thermal-management.pdf

[32] S. Lee, L. K. John, and A. Gerstlauer, “High-level synthesis of approximate hardware under joint precision and voltage scaling,” in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2017, pp. 187–192.

[33] G. Bulman *et al.*, “Superlattice-based thin-film thermoelectric modules with high cooling fluxes,” *Nature Communications*, vol. 7, p. 10302, Jan. 2016.

[34] I. Chowdhury *et al.*, “On-chip cooling by superlattice-based thin-film thermoelectrics,” *Nature Nanotechnology*, vol. 4, pp. 235–8, May 2009.

[35] H. Kattan, S. W. Chung, J. Henkel, and H. Amrouch, “On-demand Mobile CPU Cooling with Thin-Film Thermoelectric Array,” *IEEE Micro*, pp. 1–1, 2021.

[36] S. Salamin, G. Zervakis, O. Spantidi, I. Anagnostopoulos, J. Henkel, and H. Amrouch, “Reliability-Aware Quantization for Anti-Aging NPUs,” in *Design, Automation and Test in Europe Conference (DATE)*, Feb 2021.

[37] Z.-G. Tasoulas and I. Anagnostopoulos, “Performance and aging aware resource allocation for concurrent GPU applications under process variation,” *IEEE Trans. Nanotechnol.*, vol. 18, pp. 717–727, 2019.

[38] S. Mahapatra and N. Parihar, “Modeling of nbti using bat framework: Dc-ac stress-recovery kinetics, material, and process dependence,” *IEEE Trans. Device Mater. Rel.*, vol. 20, no. 1, pp. 4–23, 2020.

[39] V. M. van Santen *et al.*, “BTI and HCD degradation in a complete 32× 64 bit SRAM array–including sense amplifiers and write drivers–under processor activity,” in *International Reliability Physics Symposium (IRPS)*, 2020, pp. 1–7.

[40] S. Mishra *et al.*, “A simulation study of nbti impact on 14-nm node finfet technology for logic applications: Device degradation to circuit-level interaction,” *IEEE Trans. Electron Devices*, vol. 66, no. 1, pp. 271–278, 2018.

Georgios Zervakis is a Research Group Leader at the Chair for Embedded Systems (CES) at the Karlsruhe Institute of Technology (KIT), Germany. He received the Diploma and the Ph.D. degree from the Department of Electrical and Computer Engineering (ECE), National Technical University of Athens (NTUA), Greece, in 2012 and 2018, respectively. Before joining KIT, Georgios worked as a primary researcher in several EU-funded projects as member the Institute of Communication and Computer Systems (ICCS), Athens, Greece. His research interests include approximate computing, low power design, design automation, and integration of hardware acceleration in cloud.

Iraklis Anagnostopoulos is an Assistant Professor at the Electrical and Computer Engineering Department at Southern Illinois University Carbondale. He is the director of the Embedded Systems Software Lab, which works on run-time resource management of modern and heterogeneous embedded many-core architectures, and he is also affiliated with the Center for Embedded Systems. He received his Ph.D. in the Microprocessors and Digital Systems Laboratory of National Technical University of Athens. His research interests lie in the area of constrained application mapping for many-core systems, design and exploration of heterogeneous platforms, resource contention minimization and power-aware design of embedded systems.

Sami Salamin received his B.Sc. degree in computer systems engineering and M.Sc. degree (first rank) from Palestine Polytechnic University, Hebron, Palestine in 2005 and 2012, respectively. Since 2016, he is pursuing his Ph.D. degree with the Chair of Embedded Systems (CES), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. He holds HiPEAC Paper Award. ORCID 0000-0002-1044-7231

Ourania Spantidi received her B.Sc. in Informatics and Telecommunications from the National and Kapodistrian University of Athens, and her M.Sc. degree in Computer Science from the Southern Illinois University Carbondale. She is currently pursuing her Ph.D. degree with the Electrical, Computer and Biomedical Engineering department, Southern Illinois University Carbondale, as a member of the Embedded Systems Software lab. ORCID 0000-0002-3631-1607

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2022.3141054, IEEE Transactions on Computers

Isai Roman-Ballesteros received his B.Sc. in Computer Systems Engineering and M.Sc. in Intelligent Systems from Monterrey Institute of Technology, Monterrey, Mexico in 2003 and 2006. From 2007 until 2019 he worked as an IT Consultant. Since 2019 he is being pursuing his M.Sc. in Computer Science from Karlsruhe Insitute of Technology (KIT), Karlsruhe, Germany.

Jörg Henkel (M'95-SM'01-F'15) is with Karlsruhe Institute of Technology and was before a research staff member at NEC Laboratories, Princeton, NJ. He has received six best paper awards from, among others, ICCAD, ESWeek and DATE. For two terms he served as the Editor-in-Chief for the ACM Transactions on Embedded Computing Systems. He is currently the Editor-in-Chief of the IEEE Design&Test Magazine and is/has been Associate Editor for major ACM and IEEE Journals. He has led several conferences as a General Chair incl. ICCAD, ESWeek and serves as Steering Committee chair/member for leading conferences and journals for embedded and cyber-physical systems. Prof. Henkel coordinates the DFG program SPP 1500 "Dependable Embedded Systems" and is a site coordinator of the DFG TR89 collaborative research center "Invasive Computing". He is the chairman of the IEEE Computer Society, Germany Chapter, and a Fellow of the IEEE.

Hussam Amrouch (S'11-M'15) is a Junior Professor heading the Chair of Semiconductor Test and Reliability (STAR) within the Computer Science, Electrical Engineering Faculty at the University of Stuttgart. He received his Ph.D. degree with distinction (Summa cum laude) from the Karlsruhe Institute of Technology in 2015. He holds seven HiPEAC Paper Awards and three best paper nominations at top EDA conferences: DAC'16, DAC'17 and DATE'17 for his work on reliability. He has 135+ publications in multidisciplinary research areas, starting from semiconductor physics to circuit design all the way up to CAD and computer architecture. Dr. Amrouch has delivered 9 tutorial talks in major EDA conferences like DAC and DATE and 23 invited talks (including 2 Keynotes) in several international conferences, universities, and companies.