Comprehensive Evaluation of Supply Voltage Underscaling in FPGA on-chip Memories

Behzad Salami*†, Osman S. Unsal*, and Adrian Cristal Kestelman*†‡
*Barcelona Supercomputing Center (BSC), Barcelona, Spain.
†Universitat Politcnica de Catalunya (UPC), Barcelona, Spain.
‡IIIA - Artificial Intelligence Research Institute CSIC - Spanish National Research Council, Spain.
Emails: {behzad.salami, osman.unsal, and adrian.cristal}@bsc.es

***Abstract*—In this work, we evaluate aggressive undervolting, i.e., voltage scaling below the nominal level to reduce the energy consumption of Field Programmable Gate Arrays (FPGAs). Usually, voltage guardbands are added by chip vendors to ensure the worst-case process and environmental scenarios. Through experimenting on several FPGA architectures, we measure this voltage guardband to be on average 39% of the nominal level, which in turn, delivers more than an order of magnitude power savings. However, further undervolting below the voltage guardband may cause reliability issues as the result of the circuit delay increase, i.e., start to appear faults. We extensively characterize the behavior of these faults in terms of the rate, location, type, as well as sensitivity to environmental temperature, with a concentration of on-chip memories, or Block RAMs (BRAMs). Finally, we evaluate a typical FPGA-based Neural Network (NN) accelerator under low-voltage BRAM operations. In consequence, the substantial NN energy savings come with the cost of NN accuracy loss. To attain power savings without NN accuracy loss, we propose a novel technique that relies on the deterministic behavior of undervolting faults and can limit the accuracy loss to 0.1% without any timing-slack overhead.**

I. Introduction

The power and energy dissipation of digital circuits is directly related to their supply voltage. Usually, conservative voltage guardbands are added by chip vendors to ensure the worst-case process and environmental scenarios. However, in real-world applications, these voltage margins are unnecessarily conservative and eliminating them can directly deliver significant power and energy efficiency. In recent years, it has been shown that aggressive undervolting, i.e., supply voltage underscaling below the standard nominal level can substantially improve the energy efficiency of real hardware including CPUs [1], [2], [3], [4], [5], [6], GPUs [7], ASICs [8], DRAMs [9], and SRAMs [10]. This paper extends the aggressive undervolting approach for commercial FPGAs. FPGAs are increasingly employed within the modern data centers, as expected to be in 30% of supercomputers by 2020 [11], thanks to their massively parallel architecture and streaming execution model.

As a downside, aggressive undervolting below the voltage guardband can lead to timing related faults as the result of circuit delay increase, which can cause applications to crash or terminate with wrong results. Understanding the behavior of these faults and efficiently mitigate them can deliver further power and energy savings in low-voltage designs without performance degradation. It is worth noting that unlike Dynamic Voltage and Frequency Scaling (DVFS) technique [12], [13], the frequency is not scaled down in the undervolting approach, therefore energy savings can be more significant.

More concentration of this paper is on-chip memories, or Block RAMs (BRAMs) of commercial FPGAs, since BRAMs play a key role in the acceleration of state-of-the-art applications such as Neural Networks (NNs) [14]. Also, unlike many components of commercial FPGAs, the supply voltage of BRAMs can be independently regulated, which allows detailed power and reliability trade-off analysis.

To a thorough study, we perform our experiments on several representative FPGA platforms from Xilinx, a main vendor. We experimentally observe that data can be safely retrieved without any observable fault when the supply voltage is underscaled below the nominal and until a certain level, i.e., V_{min}. V_{min} of FPGA BRAMs is measured to be on average 39% of the nominal level. By eliminating this large voltage guardband, BRAMs power consumption is reduced more than an order of magnitude without compromising to reliability or performance. However, further voltage underscaling below V_{min} causes faults in some locations of some BRAMs, with an exponentially increasing fault rate that varies between studied platforms.

As a case study application, we push a typical FPGA-based NN accelerator to operate under low-voltage BRAMs. In consequence, the power is significantly reduced; however, undervolting below V_{min} the NN accuracy is reduced as the result of faults. We mitigate these faults by a proposed mitigation technique, in which constraints the placement stage to leverage BRAMs that are relatively highly-reliable to protect the most sensitive NN parameters against faults.

The aim of this paper is to experimentally understand the power and reliability trade-off of commercial FPGAs under aggressive low-voltage operations and its impacts on a FPGA-based NN accelerator. The main contributions of this work are summarized as follows:

- This paper is the first effort to empirically study aggressive voltage underscaling of FPGAs below the standard nominal level. Through experimenting on four FPGA platforms, we confirm a large guardband, which is measured to be on average 39% of the nominal level for

on-chip memories, slightly different among studied platforms.

- We perform the first detailed experimental bit-level characterization study of the behavior of faults when the supply voltage of FPGA on-chip memories is underscaled below the safe voltage, i.e., V_{min}. More specifically, we observe that *i)* a vast majority of these faults manifest themselves as "1" to "0" bit flips, *ii)* the location of these faults do not change over time, *iii)* faults are fully non-uniformly distributed over various BRAMs, and *iv)* higher temperature leads to reduced fault rates, which implies lower V_{min} at higher temperatures.
- We perform the first study of the efficiency of a NN accelerator under low-voltage FPGA operations. To attain the subsequent power saving without NN accuracy loss below V_{min}, we present a low-overhead application-dependent BRAMs placement technique that relies on the deterministic behavior of undervolting faults.

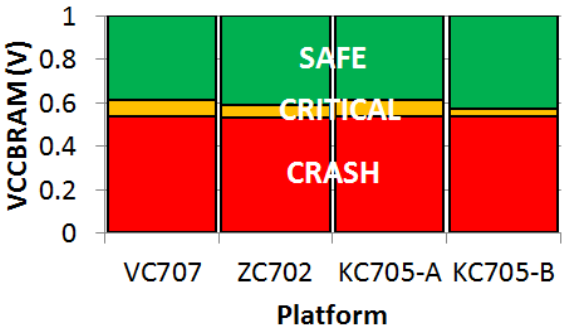
This paper is organized as follows. In Section II, we elaborate the behavior of FPGA-based BRAMs under low-voltage operations, e.g., voltage guardband and detailed fault characterization. In Section III, we present and discuss results of the FPGA-based NN accelerator at low-voltage regimes. The previous work is reviewed in Section IV and finally, the paper is concluded and summarized, in Section V.

II. Low-Voltage Operations in FPGA-based BRAMs

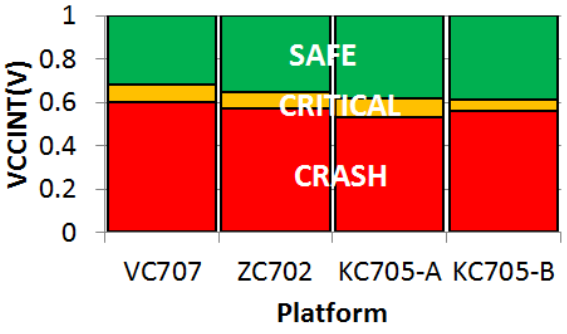
In this section, we present and discuss the behavior of FPGA BRAMs under aggressively low-voltage operations.

A. Experimental Methodology

Our experiments include several representative platforms from Xilinx, a main vendor, i.e., VC707 (performance-optimized architecture) [15], ZC702 (hardware-software architecture) [16], and two identical samples of KC705 (power-optimized architecture) [17] platforms. These four platforms allow us to study different architectures and also the impact of die-to-die process variation for KC705. All platforms are fabricated with 28nm technology. These platforms are composed of many components. Some of these components such as BRAMs, internal components (Look-Up Table (LUTs) and Digital Signal Processors (DSPs)), and Auxiliary I/O can be independently regulated. Among these components, we perform the first study on two voltage rails, i.e., V_{CCBRAM} and V_{CCINT} that supply BRAMs and tightly-coupled internal components, respectively, since these are on-chip resources. We aim to discover the minimum safe voltage, i.e., V_{min} of those FPGA components. As can be seen in Fig. 1, for both V_{CCINT} and V_{CCBRAM}, there are substantial voltage guardbands for all platforms below the nominal level (**SAFE** region), which is measured on average 34% and 39% for V_{CCINT} and V_{CCBRAM}, respectively. This voltage guardband is experimentally measured 12%, 20%, and 16% for CPUs [18], GPUs [7], and DRAMs [9], respectively. Further undervolting, cause observable faults (**CRITICAL** region), until a voltage level at which the platforms stop operating



(a) V_{CCBRAM}



(b) V_{CCINT}

Fig. 1: Undervolting FPGA components, i.e., BRAMs (V_{CCBRAM}) and Internal (V_{CCINT}) voltages.
* **SAFE**: no observable fault occur. **CRITICAL**: faults manifest. **CRASH**: FPGA stops operating.

(**CRASH** region). Below the V_{crash} region, we observed that the DONE pin is unset, which at nominal levels indicates incorrect bitstream. Note that for all platforms, the nominal voltage level of both voltage rails is $V_{nom} = 1V$; however, V_{min} and V_{crash} slightly vary for different platforms, which can be the consequence of the architectural difference of platforms as well as the impact of process variation.

For more detailed study, we concentrate on V_{CCBRAM}, since its independent voltage rail allows to evaluate BRAMs individually in fine-grain level at the critical voltage region, unlike the V_{CCINT} that feeds several components such as LUTs and DSPs. Further power and reliability of the BRAMs at the critical region is discussed later in this section. BRAMs are small memory blocks that are distributed over the chip, and each basic BRAM block is a matrix of bitcells composed of rows and columns. In studied platforms the size of each basic-setup BRAM is 16 Kbits with 1024 rows and 16 columns. BRAMs can be either individually accessed or cascaded to build larger memories (with some overheads). This methodology provides flexibility for the FPGA designers to have single-cycle access to on-chip memories as per bandwidth or size needs. Detailed specifications of our tested platforms are summarized in TABLE I.

Through Power Management Bus (PMBUS) standard [19],

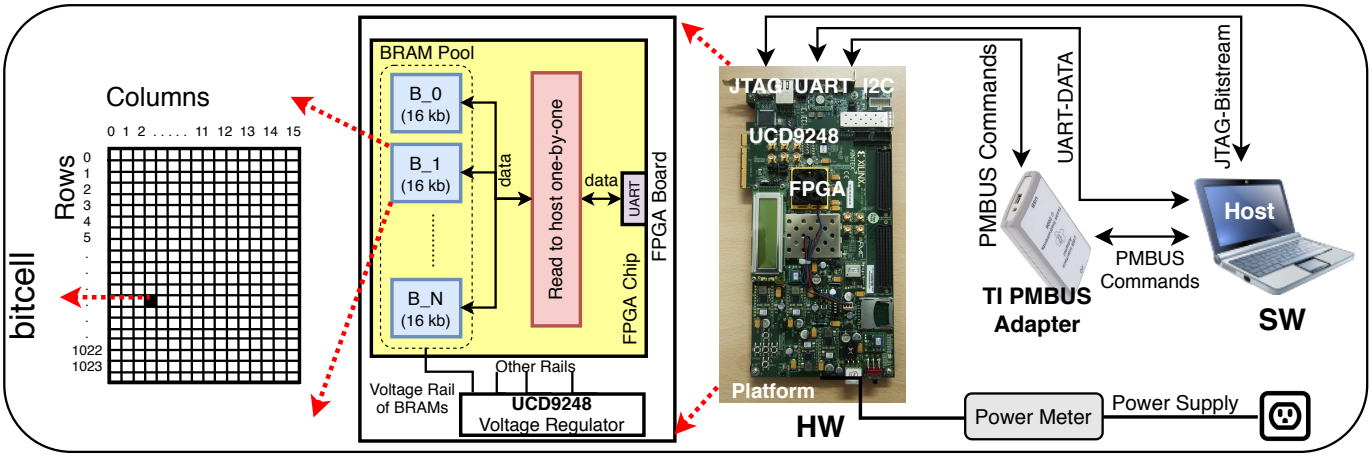


Fig. 2: Experimental setup to perform the study on FPGA BRAMs aggressive voltage underscaling.

TABLE I: Specifications of tested FPGA platforms from Xilinx, a main vendor.

Hardware Platform (Board)	VC707 [15]	ZC702 [16]	KC705-A* [17]	KC705-B* [17]
Device Family	Virtex-7	Zynq7000	Kintex-7	Kintex-7
Chip Model	XC7VX485T-ffg1761-2	XC7Z020-CLG484-1	XC7K325T-ffg900-2	XC7K325T-ffg900-2
Speed Grade	-2	-1	-2	-2
Serial Number (S/N)	1308-6520	630851561533-44019	604018691749-76023	604016111717-65664
Number of BRAMs	2060	280	890	890
Basic Size of Each BRAM	1024*16-bits**	1024*16-bits	1024*16-bits	1024*16-bits
Manufacturing Process Technology	28nm	28nm	28nm	28nm
Nominal V_{CCBRAM} (V_{nom})	1V	1V	1V	1V

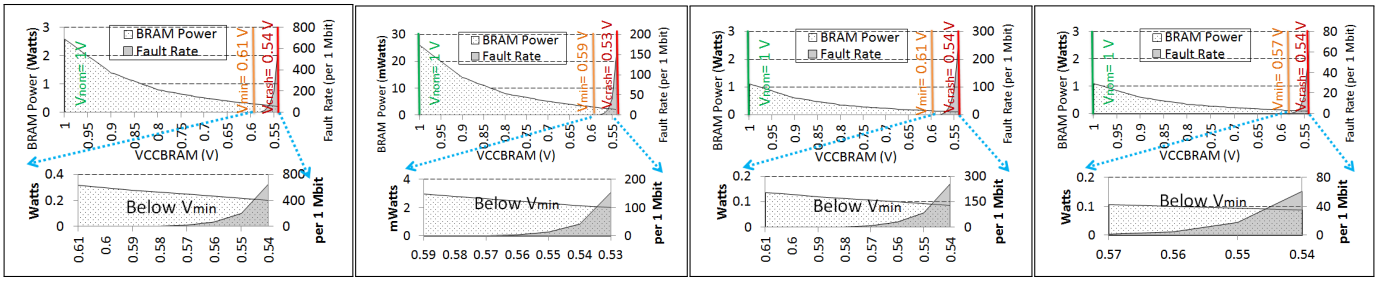
* Two identical samples of KC705 to study the impact of die-to-die process variation.
** Each row of BRAMs has two additional bits as parity, which are not considered in our experiments.

Listing 1: Pseudo-code to study FPGA BRAMs undervolting at the **CRITICAL** region, on the setup of Fig. 2.

```
1: VCCBRAM =Vmin;
2: while(VCCBRAM >= Vcrash) begin
3:    while(numRun <= 100) begin
4:       delay(1sec);
5:       Transfer content of BRAMs to the host;
6:       Analyse faulty data (rate and location);
7:       numRun++;
8:    end
9:    VCCBRAM− = 10(mV);
10: end
```

it is possible to access the on-board voltage regulator, with the part number of TI-UCD9248 in the studied platforms, and in turn, to regulate and monitor different voltage rails such as V_{CCBRAM} and V_{CCIO}. For this aim, we use Texas Instruments (TI) PMBUS USB Adapter and the provided C-based Application Programming Interface (API), which facilitates accessing the on-board voltage controller through the host [20]. The experimental setup of BRAMs undervolting evaluation is shown in Fig. 2. It is composed of two distinct hardware and software components. The task of the hardware FPGA platform is to access BRAMs and transmit their content to the host, using a serial interface. In ZC702, this serial interface is controlled by the on-board Arm processor; however, in other platforms, we built our own hardware serial interface. Note that we verify and validate that this interface is entirely reliable at any V_{CCBRAM} level and is not affected by the BRAMs undervolting.

On the other side, the host issues the required PMBUS commands to set a certain voltage to V_{CCBRAM}. Also, the host initializes BRAMs and analyzes potentially faulty data retrieved from BRAMs. On this setup, the reduced V_{CCBRAM} below the minimum safe voltage, i.e., V_{min} can cause the timing violations and in turn, corrupting some of bitcells of some of BRAMs. We follow the method shown in Listing 1 to comprehensively analyze the behavior of these faults. As explained in Listing 1, we retrieve contents of BRAMs one-by-one and within each BRAM row-by-row, and transfer them to the host. In the host, we analyze the rate and location of faults. This process is repeated 100 times for each voltage level to obtain statistically significant results. The reported results in this paper are the median of these 100 tests. After a soft reset, we gradually decrease V_{CCBRAM} by 10mV and repeat the process until the lowest voltage that our design can operate, i.e., V_{crash}. For each voltage level, the fault rate and power consumption of BRAMs are recorded. Finally, to measure the power consumption with acceptable accuracy, we use a power meter, while to extract the power contribution of BRAMs in the nominal voltage level, we use Xilinx Power Estimation (XPE) tool. Thus, we report total power consumption



(a) VC707 (b) ZC702 (c) KC705-A (d) KC705-B

Fig. 3: Major observations under low-voltage operations in FPGA BRAMs for studied commercial platforms. y-axis are with different scales. Power results are reported as mWatts in ZC702 and in Watts for others.

including dynamic and static, which are both directly reduced by undervolting. Note that BRAMs considered in this paper internally operate at a fixed frequency of $\sim$ 500mhz [21] and the whole design operates on the maximum frequency without timing violation at the nominal voltage level that is calculated by Vivado, i.e., Xilinx compiling tool.

B. Power & Reliability Trade-off of FPGA BRAMs Through Undervolting

As can be seen in Fig. 3, our experiments on the underscaling of V_{CCBRAM} below V_{nom} demonstrate two thresholds. *First*, a voltage guardband or V_{min} that separates the fault-free and faulty regions. *Second*, V_{crash} that is the lowest level of the voltage at which our design practically operates. For all tested platforms, factory set V_{nom} is 1V. However, through our experiments, we observe slightly different $V_{min}V$ and V_{crash} among the platforms. Note that repeating these tests in more noisy and harsh environments can cause observable faults above observed V_{min}, as well.

When underscaling V_{CCBRAM} below V_{min} the fault rate exponentially increases, while the power consumption significantly reduces, but with different scales for different platforms. When $V_{CCBRAM} = V_{min}$, more than an order of magnitude of power gain is achieved over $V_{nom} = 1V$, without compromising on performance or reliability. As can be seen, both power consumption and reduction in KC705s are less than VC707, which is the consequence of having relatively less BRAMs as well as the inherent power optimizations adopted for KC705 by the vendor. Also, BRAMs power consumption in ZC702 is relatively less than other platforms, since it is composed of much less number of BRAMs.

Undervolting below V_{min}, the fault rate exponentially increases, up to 652, 153, 254, and 60 per 1 Mbits ($\sim$ 0.06%, 0.01%, 0.03%, and 0.005%)[1] at V_{crash}, for VC707, ZC702, KC705-A, and KC705-B, respectively (with initial pattern= 16'hFFFF). As can be seen, VC707 exhibits the worst fault rate, up to 652 fault per 1 Mbit, which can be the consequence of the adopted architectural and technological performance optimization techniques, by the vendor. Also, a significant difference is observed between the two KC705 platforms, i.e., KC705-A and KC705-B. As can be seen, KC705-A shows a 4.1X higher fault rate than KC705-B, which can be due to the die-to-die process variation. Finally, ZC702 shows a fault rate of up to 153 faults per 1 Mbit. The conclusion of this experimental comparison is that the reliability degradation through aggressive voltage underscaling not only depends on the architecture of the underlying FPGA, but also, it can significantly vary for different FPGAs of a same platform as the result of the process variation.

[1]Since the overall fault rates are very small, instead of percentage (%), we present them in terms of number of faults per 1Mbit, for clearer charts.

C. Fault Characterization Through FPGA BRAMs Undervolting Below V_{min}

In this section, we comprehensively characterize the behavior of undervolting faults, where V_{CCBRAM} is underscaled from V_{min} to V_{crash}. Understanding this behavior is useful to deploy efficient mitigation techniques and in turn, take further advantage of power savings through undervolting. These experimental findings are used in the case study application in Section III.

1) The Impact of Data Pattern: We repeat experiments with different data patterns as the initial content of BRAMs and observe that the vast majority of undervolting faults are "1" to "0" bit-flips. This type of bit-flips is measured to be on average 99.9% for studied platforms. In other words, the fault rate is proportional to the number of "1" bits; for example, with pattern= 16'hFFFF the fault rate is almost double than pattern= 16'hAAAA, and with pattern= 16'h0000 only few faults manifest, as shown in Fig. 4 for VC707. In the same line, we did not observe any meaningful correlation when permutations with the same number of "0"s and "1"s are used in the input data pattern. For instance, as can be seen, the fault rate of patterns= 16'hAAAA, 16'h5555, and a random pattern composed of 50% of "0"s and 50% of "1"s are almost the same. Considering the behavior as mentioned above, we present the rest of results in this section, for the data pattern= 16'hFFFF, which corresponds to the highest fault rate among the tested input patterns.

2) Stability of Fault Over Time: As earlier mentioned, we repeat each test, i.e., consecutively reading the content of BRAMs under low-voltage operation, 100 times to get

TABLE II: Fault stability over time. This table includes fault rates of 100 consecutive runs at V_{crash} with pattern=16'hFFFF for VC707.

Parameter	VC707	ZC702	KC705-A	KC705-B
AVERAGE fault rate*	652	153	254	60
MINIMUM fault rate*	630	140	237	51
MAXIMUM fault rate*	669	162	264	69
STD. DEV of fault rates	7.3	5.9	4.8	1.8

* per 1 Mbit.

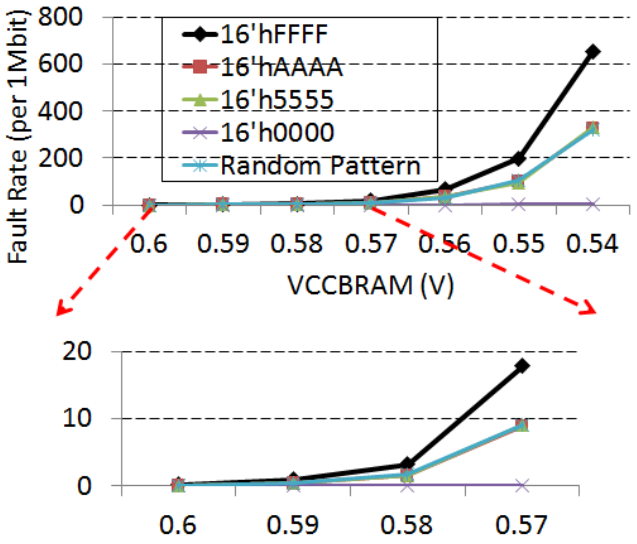


Fig. 4: The impact of data pattern in the fault rate on VC707. A similar behavior is observed for other platforms.

statistically significant results. We observe that the fault rate and location for different runs do not significantly change. For instance, average, minimum, maximum, and standard deviation of fault rate of these 100 runs at V_{min} are summarized in TABLE II. As seen, the difference between various runs is negligible. Consequently, undervolting faults show a stable and deterministic behavior over time.

3) Fault Variability among BRAMs: By statistically analyzing the experimental results, we observe that faults are not uniformly distributed over different BRAMs. Also, common for all platforms, we observe that a significant percentage of BRAMs, e.g., 38.9% in VC707 never experience faults even at the lowest voltage level at $V_{crash} = 0.54V$. For instance, on VC707 when $V_{CCBRAM} = V_{crash} = 0.54V$, the maximum, minimum, and average fault rate within BRAMs are 2.84%, 0%, and 0.04%, respectively. For further analysis, we clustered this statistical information in low-, mid-, and high-vulnerable classes of BRAMs, using the k-means clustering algorithm. For all platforms, a vast majority of BRAMs are clustered as low-vulnerable. For instance, we show detailed results of VC707 in Fig. 5. As can be seen, 88.6% of BRAMs are recognized as low-vulnerable with an average fault rate of 0.02%, $\sim$ 3.4 faults within an individual BRAM with the size of 16-kbits.

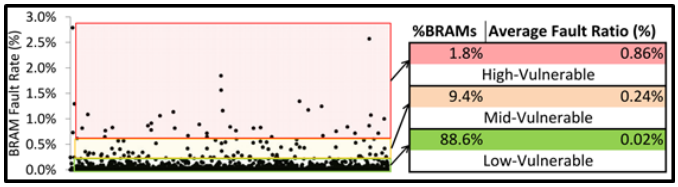


Fig. 5: Clustering BRAMs in low-, mid-, and high-vulnerable classes using K-mean algorithm. This figure shows the clustering at $V_{crash} = 0.54V$ for VC707.

This significant fault variability among BRAMs can be due to either the chip-dependent process variation or uncertainties of the design tools for place and route. We verify this argument by performing the following test; for our test design, we extracted the fault rate and location in different BRAMs with several place-and-route compilations. Repeating the voltage underscaling operation on these various bitstreams, we observe almost an identical fault rate and locations in the corresponding physical locations of BRAMs. Hence, we conclude this fault variability among BRAMs is the result of the within-die process variation. According to this test and also the deterministic behavior of faults, as earlier explained, we construct a chip-dependent Fault Variation Map (FVM). FVM is extracted by mapping the observed fault rates to the physical location of BRAMs on the tested chips. Through Vivado, we extract the required information to build FVM, including the floorplan of the chip and the placement information of BRAMs. For instance, FVM of VC707 is shown in Fig. 6, when V_{CCBRAM} is underscaled from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$ by gradually voltage undercaling by 10 mV steps.

4) Impact of Die-to-Die Process Variation: We perform a further analysis of understanding the effect of voltage scaling on two identical samples of the same platform, i.e. KC705-A and KC705-B, which can show the impact of the die-to-die process variation. As earlier noted, KC705-A shows a relatively higher fault rate. Furthermore, with extracting their FVMs, we observe a significant difference in the fault map among BRAMs, Fig 7 that shows FVM of these platforms at V_{crash}. For instance, BRAM#(116,1) has high-vulnerability in KC705-A; however, it has low-vulnerability in KC705-B. The consequence is the significant impact of the die-to-die process variation in the reliability behavior of FPGA BRAMs under aggressively reduced voltage levels.

D. Impact of the Environmental Temperature

We perform an experiment to study the effect of the environmental temperature on the behavior of faults when V_{CCBRAM} is lowered below V_{min}. Toward this goal, we place the hardware board inside a heat chamber where we regulate the temperature. We monitor the on-board temperature using PMBus commands. Through experiments, BRAMs fault rates are extracted and shown in Fig. 8 under the on-board temperatures of 50°C (default temperature), 60°C, 70°C, and 80°C. As can be seen, with heating up, the fault rate constantly reduces;

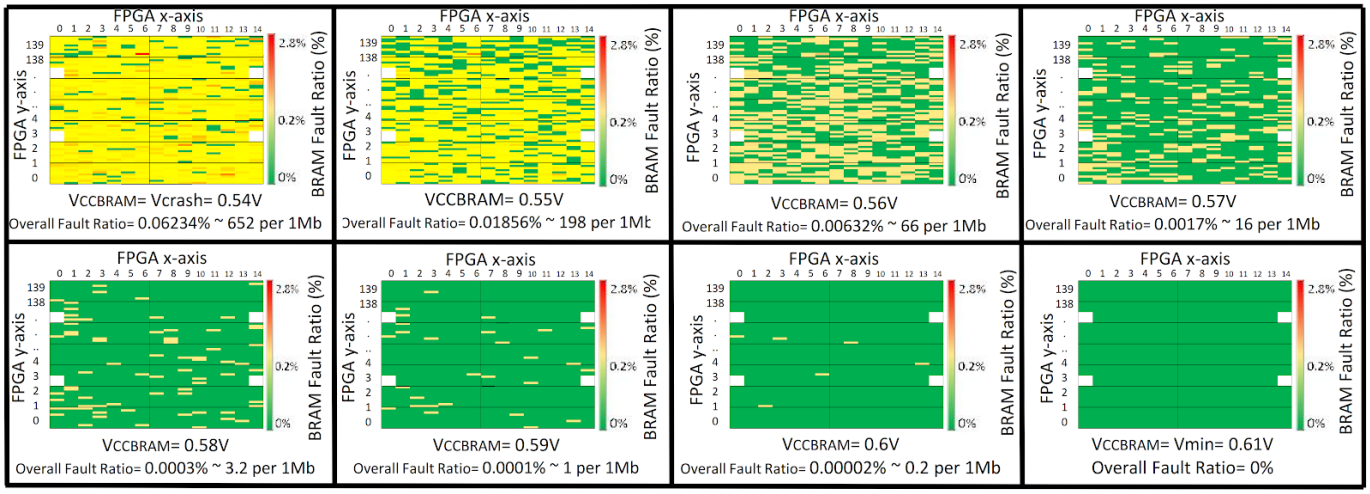
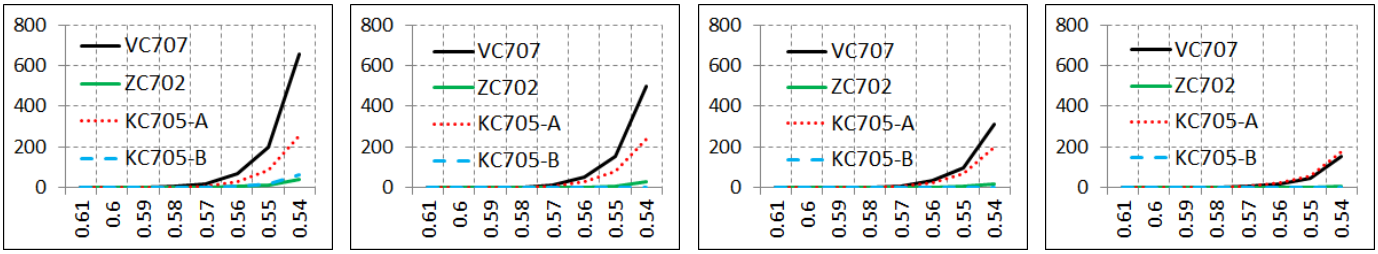
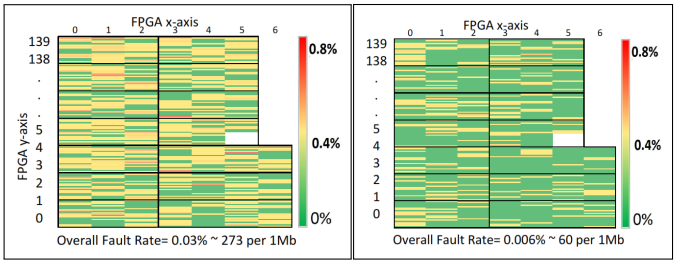


Fig. 6: BRAMs Fault Variation Map (FVM), scaling V_{CCBRAM} from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$.
* Each small rectangular box represents a BRAM mapped to the corresponding X and Y physical location on FPGA, shown for Virtex-7 FPGA in VC707 platform containing 2060 BRAMs.
** White boxes represent the empty physical locations of BRAMs.
*** For a clearer representation, other FPGA components such as LUTs and DSPs are not shown



(a) On-board Temperature= 50°C (b) On-board Temperature= 60°C (c) On-board Temperature= 70°C (d) On-board Temperature= 80°C

Fig. 8: The correlation among on-board temperature, supply voltage, architectural technology, and fault rate for FPGA BRAMs. x-axis represents V_{CCBRAM} from V_{min} to V_{crash} and y-axis shows the fault rate per 1Mbit.



(a) KC705-A (b) KC705-B

Fig. 7: FVM for two identical samples of KC705 at V_{crash}. Different fault rates and fault locations (FVM) are experimentally observed.

for instance, by more than 3X in VC707, with the temperature is increased from 50°C to 80°C. This phenomenon is the consequence of the Inverse Thermal Independence (ITD) property [22]. ITD is a thermal property of digital devices with nano-scale technology nodes; and states that under ultra low-voltage operations, the circuit delay reduces at higher temperatures. The reason is that as the technology node scales down, the supply voltage approaches the threshold voltage. Hence, at low-voltage regimes, increasing the temperature reduces the threshold voltage and allows the device to switch faster. In turn, with the circuit delay decreasing, the number of critical paths, and subsequently, the fault rate reduces. This property is experimentally verified in our case, for commercial FPGAs. Also, as can be seen, the fault rate in VC707 is reduced more aggressively than KC705-A. A relatively 156% more fault rate in 50°C is reduced to 11.6% less fault rate in 80°C, for VC707 vs. KC705-A. The architectural and technological difference between these platforms can be the reason since their design goal is different, i.e., performance (VC707) vs. power (KC705-A). Also, by heating up, the fault rate is significantly lower for VC705-B than KC705-A, as the consequence of the process variation.

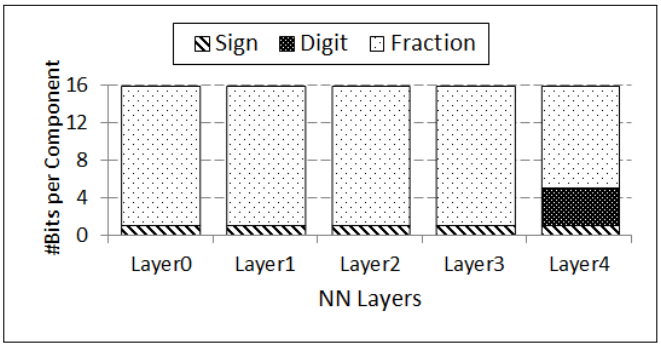


Fig. 9: Minimum precision to represent weights of NN by following a minimum per-layer fixed-point data representation model.

III. FPGA-based Neural Network (NN) Acceleration Based-on Low-voltage BRAMs

In this section, we present and discuss results of our study on the impact of BRAMs undervolting in a typical FPGA-based NN accelerator. More specifically, our study includes the power consumption and NN accuracy trade-off, characterizing the NN accuracy under low-voltage BRAMs operations, and an effective application-aware fault mitigation technique for operating below V_{min}. We perform our experiments on a fully-connected NN in the classification (classification/prediction/recognition) phase. The major results presented in this section are on the MNIST dataset [23]; however, to show the generality of findings, we briefly discuss two other NN benchmarks, i.e., Forest [24] and Reuters [25], as well. MNIST is an image recognition benchmark suite, working on black and white digitized handwritten digits, each image 784*8-bit pixels, the output infers the number from 0 to 9. This model is composed of input, hidden, and output layers, where all neurons of adjacent layers are fully connected to each other, as a simplified model is shown in Fig. 12a. The intensity of each connection is determined by weights, whose values are tuned in the training phase as an offline stage. The first layer is called input layer and contains one neuron for each component in the input vector (e.g., each pixel). The last layer is called output layer and contains one neuron for each component in the output vector (e.g., each output class). Between the input and output layers, there are hidden layers. Each neuron of the NN uses an activation function (typically logarithmic sigmoid) to quantize its output. Finally, in the output layer, a softmax function generates the final output of the NN. The softmax function determines the probability distribution of the input vector over all different possible outcomes.

A. Introducing the Experimental Setup of the NN Evaluating

Detailed specifications of our experimental setup are summarized in Table III. In our system architecture, weights of the NN are located inside BRAMs and 10000 input images of MNIST are being streamed through an off-chip DDR-3. The required calculation of the image classification, i.e., matrix multiplication and sigmoid function activation, are performed in parallel by leveraging DSPs and LUTs of the FPGA in a stream-fashion model. This is a typical setup for most of the FPGA-based NN accelerator, as surveyed in [14]. Note that to save the space we present results on VC707 since we reach to similar conclusions by repeating experiments on other FPGA platforms.

TABLE III: Detailed specifications of the baseline NN.

Neural Network (NN)	
Type	Fully-Connected Classifier
Topology (number of layers)	6L (1L input, 4L hidden, 1L output)
Per Layer Size (number of neurons)	(784, 1024, 512, 256, 128, 10)= 2714
Total Number of Weights	∼1.5 million
Activation Function	Logarithmic Sigmoid (logsig)
Major Benchmark	
Name-Type	MNIST [23]- Handwritten Digits
Number of Images	Training: 60000, Inference: 10000
Number of Pixels per Image	28*28= 784
Number of Output Classes	10
Additional Benchmarks	
1. Forest	[24]
2. Reuters	[25]
Data Representation Model	
Type	16-bits Fixed-Point
Precision	Min sign and digit per layer (Fig. 9)
An Example Synthesize of RTL NN on FPGA	
FPGA Platform-Chip	VC707-Virtex7
Operating Frequency	100Mhz
BRAM Usage (Total: 2060)	70.8%
DSP Usage (Total: 2800)	8.6%
FF Usage (Total: 303,600)	3.8%
LUT Usage (Total: 607,200)	4.9%

Our tested NN has a 6-layers topology, i.e., ($\{L_i, i \in [0,5]\}$), composed of one input, four hidden, and one output layer(s). These layers have 784, 1024, 512, 256, 128, and 10 neurons, respectively. Thus, there are five set of weights, i.e., ($\{Layer_j, j \in [0,4]\}$), where $Layer_j$ refers to the weight set between L_j and L_{j+1}. This setting leads to ∼1.5 million weights, which fills more than 70.8% BRAMs in the FPGA of VC707. Note that this size of weights exceeds the available BRAMs of other studied platforms; thus, for them we dynamically reload weights to BRAMs from DDR-3. The maximum operating frequency of our design is 100Mhz and the area utilization is summarized as 58.3%, 13.1%, and 43.1% for DSP, FF, and LUT, respectively.

The training phase of the NN is performed off-line using 60000 training images of MNIST, by a MATLAB implementation. Then, we export weights and biases of the trained NN, initialize BRAMs with these parameters, and repeat the classification of 10000 images at various levels of V_{CCBRAM}. Also, for representing data, we use the fixed-point low-precision model. Note that lowering the precision of data is a common technique for applications in the approximate computing domain, such as NN [26] and multimedia [27], to achieve power and performance efficiency with negligible accuracy loss. Following these studies, we use per-layer minimum precision fixed-point model. The bit-width of weights

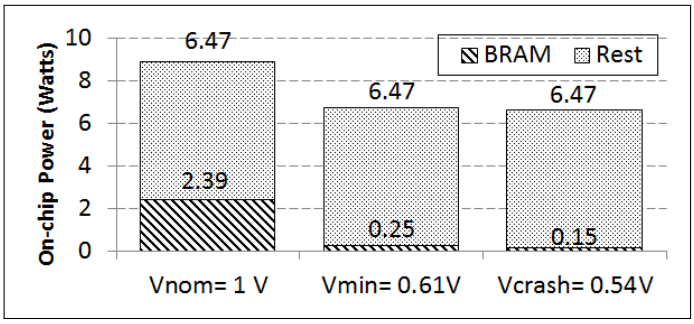


Fig. 10: On-chip total power consumption breakdown of our FPGA-based NN at V_{nom}, V_{min}, and V_{crash} (VC707). Rest includes on-chip power consumption of DSPs, LUTs, routing resource, etc.

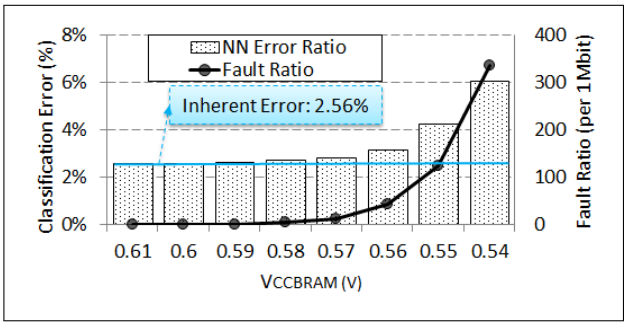


Fig. 11: Impact of BRAMs voltage scaling in the NN classification error, lowering V_{CCBRAM} from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$ (VC707).

is fixed to 16-bits, composed of the sign, digit, and fraction components. Toward this goal, with a pre-processing analysis, we extract the minimum bit-widths of the sign and digit components per layer, and the rest of 16-bits are exploited as fraction components. As can be seen in Fig. 9, except the last layer, i.e., $Layer_4$, weights of other layers are in (-1, 1) margin, which means that there is no need for the digit component. However, for the $Layer_4$, a 4-bit digit component is used, i.e., the minimum width to represent data without any NN accuracy loss.

On this setup, the total on-chip power consumption breakdown at various $V_{CCBRAM}s$, i.e., $V_{nom} = 1V$, $V_{min} = 0.61V$, and $V_{crash} = 0.54V$, is shown in Fig. 10. As can be seen, more than an order of magnitude BRAM power dissipation reduction is achieved by underscaling V_{CCBRAM} from $V_{nom} = 1V$ to the guardband gap on $V_{min} = 0.61V$, which in turn, delivers 24.1% total on-chip power reduction. Further voltage underscaling to $V_{crash} = 0.54V$, reduces a further 40% of BRAM power over $V_{min} = 0.61V$; however, as a result of the timing faults, the NN classification error is in turn impacted. This impact and the proposed fault mitigation technique are discussed later in this section.

B. The Impact of BRAMs Undervolting Below V_{min}

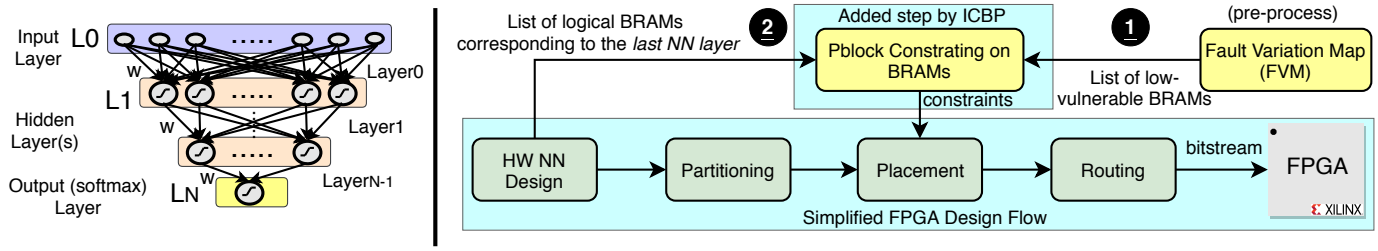
When V_{CCBRAM} is underscaled in the critical region between $V_{min} = 0.61V$ until $V_{crash} = 0.54V$, faults occurring in some of bitcells degrades the NN accuracy. Hence, the classification error is increased from 2.56% (inherent classification error without any fault) to 6.15% when $V_{CCBRAM} = V_{crash} = 0.54V$, see Fig. 11. The NN classification error (left y-axis) increases exponentially, correlated directly with the fault rate increase in BRAMs (right y-axis), as expected. Also, we observe that the fault rate in BRAMs filled with the NN weights is significantly less than the default pattern= 16h'FFFF as earlier shown in Fig. 3a. The reason is that 76.3% of the studied NN weight bits having the logic value "0". These bits have a negligible probability to be flipped, considering that most of the undervolting faults are "1" to "0" bit-flips; thus, it can be concluded that MNIST is inherently fault-tolerant against undervolting faults. Through an statistical experimentation, we confirm this data sparsity for other NN benchmarks such as Forest and Reuters. Also, other state-of-the-art works have already confirmed the sparsity of many NN benchmarks [28], [29], and a wider range of other applications, as well [30]. It means these applications would be inherently fault-tolerant for the type of failures experienced through FPGA BRAM undervolting.

C. Fault Mitigation Technique

To prevent the NN accuracy loss under low-voltage FPGA BRAM operations, we propose an efficient technique that we term Intelligently-Constrained BRAM Placement (ICBP). The overall methodology of ICBP, as an additional patch for BRAMs placement stage, is shown in Fig. 12b. It relies on our two key observations:

- ❶ As detailed in Section II, we observed that faults occur in reduced voltage BRAMs have deterministic and chip-dependent behavior with a fully non-uniform distribution between different BRAMs that is exposed as FVM. As earlier mentioned, FVM extraction is a pre-process stage.
- ❷ We observed that various layers of the given NN have different inherent vulnerability to faults. We conducted a pre-processing analysis and observed that inner layers (layers closer to the output) are relatively more vulnerable, as similarly observed in [31], [32], [33], since faults in these layers have relatively less probability to be masked through the quantification in the activation functions. The sensitivity of NN layers, i.e., $\{Layer_j, j \in [0, 4]\}$ is evaluated by injecting simulated randomly-generated faults in corresponding weights of individual layers at the Register-Transfer Level (RTL). In other words, we inject a number of random faults in weights of individual NN layers and let the NN to accomplish the classification. By monitoring the classification error of the faulty NN, we can evaluate the vulnerability of each NN level.

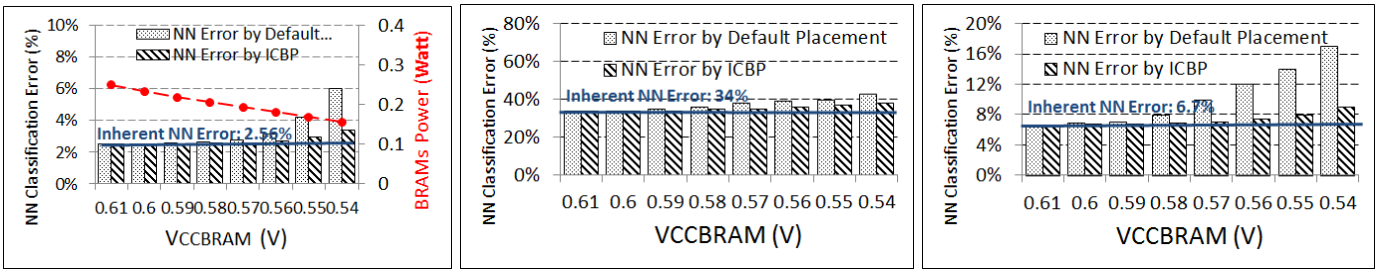
  For further analysis, we present detailed statistical information of the different layers of the given NN in Fig. 13, i.e., the size (in terms of utilized number of BRAMs to



(a) A typical fully-connected NN model

(b) The methodology of ICBP on the FPBA-based NN accelerator

Fig. 12: Methodology of our mitigation technique in FPGA-based NN, Intelligently-Constrained BRAM Placement (ICBP).



(a) MNIST [23]

(b) Forest [24]

(c) Reuters [25]

Fig. 14: Efficiency of ICBP on FPGA-based NN accelerator for MNIST, Forest, and Reuters benchmarks on VC707. (* Different scales in y-axis. *)

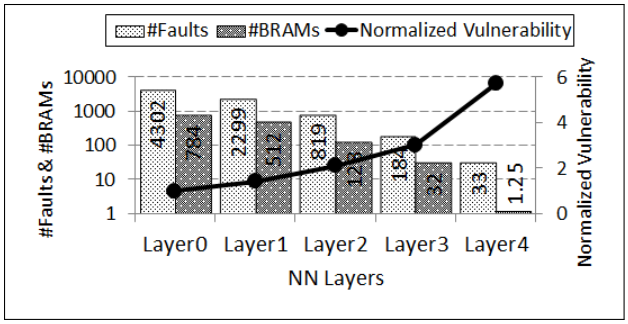


Fig. 13: Statistical analysis of layers of the given NN in terms of the size (#BRAMs), #faults (at $V_{crash} = 0.54V$ for VC707), and normalized vulnerability.

locate weights of the corresponding NN layer), number of undervolting faults that we observed in our experiments, and the normalized vulnerability of the individual layers as we performed as a pre-process study. As can be seen, for instance the output/last layer, i.e., $Layer_4$ is 6X more vulnerable than the input/first layer, i.e., $Layer_0$, which means that the same rate of faults injected in $Layer_4$ causes 6X NN classification error than injecting the same number of faults in $Layer_0$. Also, outer layers (closer to the input layer) are relatively larger, which can potentially experience more faults.

Due to these observations, ICPB introduces a simple yet effective BRAMs placement algorithm that maps the weights of the last NN layer to low-vulnerable BRAMs, targeting to mitigate faults and achieve power-savings with minimized NN classification accuracy loss. In other words, ICBP proposes additional constraints for the placement algorithm during the FPGA compiling process. To apply intelligent constraint on the BRAMs placement, we exploit the Physical Blocks (Pblocks) facility [34] of Vivado. Pblocks provides a facility to constrain logical blocks, e.g., BRAMs, to a physical region in the FPGA. Hence, we force the tool at the placement stage to locate logical BRAMs including the weights of the last NN layer, i.e., $Layer_4$, into those BRAMs that are tagged as low-vulnerable. As earlier detailed, the last NN layer is the smallest; however, the most-sensitive against faults; thus, it has the most priority to be protected. The time slack overhead of this technique is negligible since a very small number of BRAMs in the last NN layer, i.e., two BRAMs, are constrained to exploit the low-vulnerable BRAMs. Consequently, as can be seen in Fig. 14 for VC707, on average 38.1% power savings is achieved at $V_{crash} = 0.54V$ over $V_{min} = 0.61V$, by 0.6% NN accuracy loss from the inherent fault-free classification error of 2.56% for MNIST; however, the same amount of power in the default placement is dissipated by more than 3.59% NN accuracy loss. Also, we repeat the similar methodology for Forest and Reuters benchmarks and as can be seen in Fig. 14b and Fig. 14c, undervolting faults are significantly covered, which in turn, leads to prevention of the NN accuracy loss for them, as well. Among studied benchmarks, Reuters is less sparse; thus, undervolting faults more significantly impacts the NN accuracy loss; however, mostly covered by ICBP.

IV. Related Work

In comparison to traditional general-purpose CPU-based systems, hardware accelerators provide better power, performance, and energy efficiency in various domains such as database processing [35], [36], [37], speech recognition [38], [39], [40], and neural network applications [14], [41]. However, with the rise of the size of data, energy consumption is still a key concern. As an effective architecture-level energy saving mechanism, aggressive undervolting has recently brought attention, which is summarized in this section.

A. Aggressive Undervolting Technique on Real Hardware

Aggressive undervolting as an efficient technique to optimize the energy efficiency, has been recently studied for several commercial/customized hardware devices, which are summarized below. However, to the best of our knowledge, such a comprehensive study on commercial FPGAs has not been undertaken yet.

1) Voltage Guardband: Most commercial devices are designed with a voltage guardband below the standard nominal supply voltage to ensure the correct functionality in the worst case environmental conditions and process variations. This voltage guardband is fully vendor- and system-dependent; for instance, it was measured to be 20%, 16% , and 12% of the nominal voltage level in modern GPUs [7], DRAMs [9], and CPUs [18], respectively. We confirm a large voltage guardband for Xilinx FPGAs, which is experimentally measured to be on average of 39%. This gap provides an opportunity to decrease the supply voltage until V_{min} without any reliability degradation, in our case delivering more than an order of magnitude power savings.

2) Simultaneous Voltage and Frequency Underscaling: Further voltage underscaling below the voltage guardband gap, i.e., V_{min}, impacts the timing and increases the delay, which can in turn, cause fault occurring. In this regard, the simultaneous frequency underscaling, i.e., DVFS is a common approach to prevent these faults. The DVFS mechanism guarantees that the design works as close to, but always above, the critical operating point, the point where further underscaling frequency or voltage will result in observable faults [42]. A recent DVFS mechanism implemented on FPGAs, [43], showed 70% energy savings. However, the impediment of DVFS is the performance degradation as a result of the frequency lowering, which in turn, can potentially limits the energy efficiency and thus, applicability of this approach for power-hungry scenarios such as mobile environments. DVFS is not targeted in our paper.

3) Aggressive Undervolting into Critical Regions: Tackling with the increased delay in low-voltage regions below V_{min}, a more aggressive approach is to allow designs to experience timing faults and in turn, effectively tolerating faults. Characterizing and mitigating these faults can allow better power and reliability trade-offs, without significant performance degradation as is for DVFS approach. Due to the advantageous as mentioned for this approach, many recent studies have been conducted to evaluate the efficiency of aggressive undervolting on different hardware devices as summarized as follows. Our paper studies extends this studies for the first time to commercial FPGAs.

- *Modern Processors:* There are multiple studies on the voltage lowering below V_{min} in modern processors. For instance, [2] revisited the microarchitecture of the processor design to be adaptable in the critical voltage regions to minimize the voltage at which a soft architecture encounters the maximum allowable fault rate, and [3] presented a methodology for reliability-aware design space exploration. [4] extends aggressive undervolting to multi-core CPUs and [18] leveraged built-in Error-Correcting Code (ECC) technique to detect and mitigate faults in Intel Itanium II.
- *GPUs*: As an example of commercial GPUs, [44] studied this approach in GPU register files and proposed an architectural solution that leverages long register dead time to enable reliable operations from unreliable register file at low voltages.
- *ASICs*: As an example of ASICs, [45] evaluated the Floating Point Units (FPUs) under timing violations and accordingly, presented a bit-level fault model.
- *Memory Systems:* Along with processing units, the storage modules are also studied for very low voltage operation. For instance, [9] comprehensively studied the modern DRAM chips from various vendors. They analyzed its impacts on the DRAM's access latency and reliability, by characterizing the behavior of faults and presenting effective mitigation techniques. In the same line, [10] and its later version [46] evaluated the effect of supply voltage scaling in SRAMs that they specifically fabricated. It is reported that the supply voltage reduction of 310mV could save 2.9X of power consumption.

In parallel, there are several industrial/reseach projects running in this area [47], [48], [49].

4) Mitigation of Undervolting Faults: To detect and/or mitigate faults several general techniques are proposed in different domains such as Triple Modular Redundancy (TMR) [50], Razor [51], [52], ECC using Hamming code [53], Hardware Transnational Memory (HTM) [54], frequency underscaling [12], among others. These techniques can be potentially customized to detect and/or mitigate timing faults in low-voltage regions, as well; however, with timing, area, or power costs. In this paper, instead of these costly operations we performed our study to comprehensively understand the behavior of faults under low-voltage operations and accordingly, develop application-dependent efficient mitigation technique, which has negligible timing/power/area overhead.

B. Recent Related Studies on NNs

NNs are inherently power-hungry applications, due to the computations, storages, and data movements requirement for the large matrices. Addressing this concern, several application-level power-optimization techniques are proposed such as low-precision data representation model [26], node pruning [55], data compressing [56], among others. These

techniques are customized for different underlying platforms such as CPUs, GPUs, FPGAs, and ASICs, as in detail surveyed in [57]. Alternatively, as an architecture-level power-savings technique, voltage underscaling of the underlying hardware is a promising approach. Since it has been shown that NNs are inherently resilient and can tolerate with quite high fault rates [58], [59], [32], aggressive undervolting can lead to significant energy savings. Below, we summarize recent works on the voltage scaling and the subsequent resilience studies, i.e., fault characterization and/or mitigation for NNs. The vast majority of works are simulations-based; however, there are a few efforts on real hardware, as well.

1) Simulation-Based Resilience Study of NNs Under Voltage Scaling: A vast majority of existing efforts on the NNs fault tolerance study is based on either fault injection in the software level or theoretical analysis, as surveyed in [58]. More specifically, aggressively voltage underscaling has been recently studied mostly on ASIC-based NN accelerators. For instance, Minerva proposed an automated co-design approach across the algorithm, architecture, and circuit levels to optimize ASIC accelerators of fully-connected NN using SPICE simulations for low-voltage SRAMs [28]. As another recent effort, ThUnderVolt is proposed as a framework to enable the voltage scaling study on ASIC-based Deep NN (DNN) accelerators; however, they modeled timing faults via post-synthesis gate-level simulations in ModelSim [60]. Ares [61] is a framework for quantifying the resilience of deep neural networks. Also, [31] studied an RTL model of the NN from resilience perspective by injecting faults in the registers of the design. Also, recently [33] studied the fault propagation in an ASIC model of NN focused on the vulnerability of different NN layers. [62] analyzes and mitigates the impact of permanent faults on a systolic array based neural network accelerator by an Spice model on Google TPU [63]. In the same line, [64] presents an in-memory NN classifier in standard SRAM array and performs the subsequent fault study under low voltage operations; however, by a Monte Carlo simulations. It is appear that this approach lacks the exact information of the fault model and thus, their validation on the silicon remains a key question.

2) Real Hardware-Based Resilience Study of NNs Under Voltage Scaling: There are limited publicly-available works on the low-voltage NNs on real-hardware; however, there are some efforts for ASICs and SRAMs, as summarized below:

– *ASICs:* There are several energy-efficient fabricated ASICs for NNs, e.g., Google TPU [63], Eyeriss [65], YodaNN [66], and [8]. However, only [8] has briefly studied the behavior of NN below nominal level scaling beyond V_{min}, where timing faults manifest. They fabricated a 28nm SOC with a programmable accelerator design for fully-connected NN, where a Razor circuit is used to detect timing faults in the datapath under aggressively reduced supply voltages. However, this paper is targeted for ASICs and also, does not propose a detailed fault characterization study on NNs.

– *SRAMs:* [10], [46] proposed a partially silicon-validated NN study on aggressively reduced voltage on SRAMs. In other words, they fabricated an 8KB SRAM with 28nm technology and evaluated the resilience of NN, while input images are located on the reduced-voltage SRAM. However, it is suffering from several limitations, e.g., *i*) without detailed bit-level characterization, *ii*) this study is on only input data (not weights), and *iii*) a software-level NN is used, which does not allow to apply any mitigation technique on the datapath of NN on the silicon. Also, tests are performed on a specialized SRAM cells, not on standard SRAM library cell, which makes it difficult to expand results of this paper for real accelerators.

3) The Key Novelty of our Paper Related to NN Study: FPGAs are attractive devices to accelerate NN since they represent an intermediate point between the power and performance efficiency of ASICs and the programmability of CPUs and GPUs [67], [68], [69], [70]. One of the key components of FPGAs that directly impacts the performance of FPGA-based NNs is built-in BRAMs, due to the high-demand of NN computations for the parallel data access, as described in detail for recent FPGA-based accelerators in this survey paper [14]. Motivated by these studies, we evaluated the BRAMs voltage scaling impacts on the power and accuracy trade-offs of FPGA-based NN accelerator. We push a typical FPGA-based NN to work in low-voltage regimes to take advantage of the power savings, targeted for FPGAs. To the best of our knowledge, this is the first effort to perform such aggressive voltage scaling study of commercial FPGAs, while running an NN application. In addition, we proposed a novel and efficient fault mitigation technique that relies on the behavior of undervolting faults.

V. Conclusion and Future Work

This paper experimentally evaluated the supply voltage underscaling below the nominal level in commercial FPGAs. We discovered that there is a significant voltage guardband gap, where data can be safely retrieved from BRAMs. However, by further undervolting observable faults occur, as a result of the timing delay increase. We extensively characterized the behavior of these faults, more specifically for on-chip memories of FPGAs. Finally, we evaluated the impact of the undervolting in the accuracy and power of an FPGA-based NN accelerator in the inference phase. To attain the power efficiency without NN accuracy loss, we proposed an efficient application-aware BRAM placement algorithm that relies on the behavior of undervolting faults. As an ongoing work, we are working on a more comprehensive voltage scaling in other components of commercial FPGAs and on different FPGA technologies of vendors.

Acknowledgment

We thank anonymous reviewers of this paper from Micro51 for their feedbacks and comments. Also, we thank Pradip Bose, Alper Buyuktosunoglu, and Augusto Vega from IBM Watson for their contribution to this work. The research leading to these results has received funding from the European Union's Horizon 2020 Programme under the LEGaTO Project (www.legato-project.eu), grant agreement n° 780681.

REFERENCES

[1] R. Bertran, P. Bose, D. M. Brooks, J. Burns, A. Buyuktosunoglu, N. Chandramoorthy, E. Cheng, M. Cochet, S. Eldridge, D. Friedman, H. M. Jacobson, R. V. Joshi, S. Mitra, R. K. Montoye, A. Paidimarri, P. Parida, K. Skadron, M. Stan, K. Swaminathan, A. Vega, S. Venkataramani, C. Vezyrtzis, G. Wei, J. D. Wellman, and M. M. Ziegler, "Very Low Voltage (VLV) Design", in *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, pp. 601-604, 2017.

[2] A. B. Kahng, S. Kang, R. Kumar, and J. Sartori, "Designing a processor from the ground up to allow voltage/reliability tradeoffs", in *Proceedings of the 16th IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1-11, 2010.

[3] K. Swaminathan, N. Chandramoorthy, C. Y. Cher, R. Bertran, A. Buyuktosunoglu, and P. Bose, "Bravo: Balanced reliability-aware voltage optimization", in *Proceedings of the 23th IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 97-108, 2017.

[4] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, P. Lawthers, and S. Das, "Harnessing voltage margins for energy efficiency in multicore CPUs", in *Proceedings of the 50th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 503-516, 2017.

[5] G. Yalcin, S. K. Rethinagiri, O. Palomar, O. S. Unsal, A. Cristal, and D. Milojevic, "Exploring Energy Reduction in Future Technology Nodes via Voltage Scaling with Application to 10nm", in *Proceedings of the 24th IEEE Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pp. 184-191, 201Kahng20106.

[6] G. Yalcin, E. Islek, O. Tozlu, P. Reviriego, A. Cristal, O. S. Unsal, and O. Ergin, "Exploiting a fast and simple ECC for scaling supply voltage in level-1 caches", in *Proceedings of the 20th IEEE International Symposium on On-Line Testing Symposium (IOLTS)*, pp. 1-6, 2014.

[7] J. Leng, A. Buyuktosunoglu, R. Bertran, P. Bose, and V. J. Reddi, "Safe limits on voltage reduction efficiency in GPUs: a direct measurement approach", in *Proceedings of the 48th ACM International Symposium on Microarchitecture (MICRO)*, pp. 294-307, 2015.

[8] 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 *Proceedings of the International Conference of Solid-State Circuits Conference (ISSCC)*, pp. 242-243, 2017.

[9] K. K. Chang, A. G. Yaaliki, S. Ghose, A. Agrawal, N. Chatterjee, A. Kashyap, D. Lee, M. O'Connor, H. Hassan, and O. Mutlu, "Understanding reduced-voltage operation in modern DRAM devices: Experimental characterization, analysis, and mechanisms", in *Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 10, 2017.

[10] L. Yang and B. Murmann, "SRAM voltage scaling for energy-efficient convolutional neural networks", in *Proceedings of the 18th International Symposium on Quality Electronic Design (ISQED)*, pp. 7-12, 2017.

[11] M. Feldman, "Good Times for FPGA Enthusiasts.", in *Top500*, 2016. https://www.top500.org/news/good-times-for-fpga-enthusiasts/

[12] Nunez-Yanez, et al. "Energy optimization in commercial FPGAs with voltage, frequency and logic scaling", in *IEEE TC*, 2016.

[13] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott, "Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling", in *Proceedings of the 8th IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 29-40, 2002.

[14] K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, "A Survey of FPGA Based Neural Network Accelerator", *arXiv:1712.08934*, 2017.

[15] Xilinx, https://www.xilinx.com/products/boards-and-kits/ek-v7-vc707-g.html

[16] Xilinx, https://www.xilinx.com/products/boards-and-kits/ek-z7-zc702-g.html

[17] Xilinx, https://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html

[18] A. Bacha and R. Teodorescu, "Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors", in *Proceedings of the 40th IEEE International Symposium on Computer Architecture (ISCA)*, pp. 297-307, 2013.

[19] "Power Management Bus (PMBUS)." http://pmbus.org

[20] Texas Instruments (TI), "Fusion Digital Power Designer". http://www.ti.com/tool/FUSION_DIGITAL_POWER_DESIGNER

[21] Xilinx, "7 Series FPGA Memory Resources." https://www.xilinx.com/support/documentation/user_guides/ug473_7Series_Memory_Resources.pdf

[22] K. Neshatpour, W. Burleson, A. Khajeh, & H. Homayoun. "Enhancing Power, Performance, and Energy Efficiency in Chip Multiprocessors Exploiting Inverse Thermal Dependence", in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*,pp. 778-791, 2018.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", in *IEEE Proceeding* , vol. 86, no. 11, p. 2278-2324, 1998.

[24] J. A. Blackard. "Comparison of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types", PhD thesis, *Colorado State University*, 1998.

[25] A. Cardoso-Cachopo. "Improving Methods for Single-label Text Categorization", PhD thesis, *Universidade Tecnica de Lisboa*, 2007.

[26] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision", in *Proceedings of the International Conference on Machine Learning (ICML)*. pp. 1737-1746, 2015.

[27] C. Alvarez, J. Corbal, and M. Valero, "Fuzzy memoization for floating-point multimedia applications", in *IEEE Transactions on Computers*, vol. 54, no. 7, pp. 922-927, 2005.

[28] B. Reagen, P. N. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernndez-Lobato, G. Y. Wei, and D. M. Brooks, "Minerva: Enabling low-power, highly-accurate deep neural network accelerators", in *ACM SIGARCH Computer Architecture News*, vo. 44, no. 3, pp. 267-278, 2016.

[29] B. Moons and M. Verhelst, "An energy-efficient precision-scalable ConvNet processor in 40-nm CMOS", in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 903-914, 2017.

[30] University of Florida, "Sparse Matrix Collection." https://sparse.tamu.edu/

[31] B. Salami, Osman S. Unsal, and Adrian Cristal Kestelman, "On the Resilience of RTL NN Accelerators: Fault Characterization and Mitigation", in *in Proceedings of High Performance Machine Learning Workshop (HPML) in conjunction with 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 2018.

[32] O. Temam, "A defect-tolerant accelerator for emerging high-performance applications", in *Proceedings of the 39th IEEE International Symposium on Computer Architecture (ISCA)*, pp. 356-367, 2012.

[33] 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 (SC)*. pp. 8, 2017.

[34] Xilinx, https://www.xilinx.com/support/documentation/sw_manuals/xilinx12_4/Floorplanning_Methodology_Guide.pdf

[35] B. Salami, G. A. Malazgirt, O. Arcas-Abella, A. Yurdakul, and N. Sonmez, "AxleDB: A novel programmable query processing platform on FPGA", in *Microprocessors and Microsystems: Embedded Hardware Design (MICPRO)*, vol. 51, pp. 142-164, 2017.

[36] O. Arcas-Abella, A. Armejach, T. Hayes, G. A. Malazgirt, O. Palomar, B. Salami, and N. Sonmez, "Hardware acceleration for query processing: leveraging FPGAs, CPUs, and memory", in *Computing in Science & Engineering*, 18(1), pp.80-87, 2016.

[37] B. Salami, O. Arcas-Abella, and N. Sonmez, "HATCH: hash table caching in hardware for efficient relational join on FPGA", in *23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 163-163, 2015.

[38] H. Tabani, "Low-power architectures for automatic speech recognition." PhD Thesis, in *Universitat Politcnica de Catalunya (UPC)*, 2018.

[39] H. Tabani, J.M. Arnau, J. Tubella, and A. Gonzlez, "Performance analysis and optimization of automatic speech recognition", in *IEEE Transactions on Multi-Scale Computing Systems*, 2017.

[40] H. Tabani, J.M. Arnau, J. Tubella, and A. Gonzlez, "An ultra low-power hardware accelerator for acoustic scoring in speech recognition", in *26th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pp. 41-52, 2017.

[41] J. Hauswald, Y. Kang, M. A. Laurenzano, Q. Chen, C. Li, T. Mudge, R. G. Dreslinski, J. Mars, and L. Tang, "DjiNN and Tonic: DNN as a service and its implications for future warehouse scale computers", in *42nd Annual International Symposium on Computer Architecture (ISCA)*, pp. 27-40, 2015.

[42] J. Patel, "CMOS process variations: A critical operation point hypothesis", *Online Presentation*, 2008.

[43] J. Nunez-Yanez, "Adaptive voltage scaling in a heterogeneous FPGA device with memory and logic in-situ detectors", in *Microprocessors and Microsystems: Embedded Hardware Design (MICPRO)*, vol. 51, pp. 227-238, 2017.

[44] J. Tan, S. L. Song, K. Yan, X. Fu, A. Mrquez, and D. J. Kerbyson, "Combating the reliability challenge of GPU register file at low supply voltage", in *Proceedings of the International Conference on Parallel Architecture and Compilation Techniques (PACT)*, pp. 3-15, 2016.

[45] G. Tziantzioulis, A. M. Gok, S. M. Faisal, N. Hardavellas, S. O. Memik, S. Parthasarathy, "b-HiVE: A bit-level history-based error model with value correlation for voltage-scaled integer and floating point units", in *Proceedings of the 52th ACM Design Automation Conference (DAC)*. pp. 105, 2015.

[46] L. Yang and B. Murmann, "Approximate SRAM for Energy-Efficient, Privacy-Preserving Convolutional Neural Networks", in *Proceedings of the IEEE International Symposium on VLSI (ISVLSI)*, pp. 689-694, 2017.

[47] A. Cristal, O. S. Unsal, X. Martorell, P. Carpenter, R. D. L. Cruz, L. Bautista, D. Jimenez, C. Alvarez, B. Salami, S. Madonar, M. Perics, P. Trancoso, M. v. d. Berge, G. Billung-Meyer, S. Krupop, W. Christmann, F. Klawonn, A. Mihklafi, T. Becker, G. Gaydadjiev, H. Salomonsson, D. Dubhashi, O. Port, Y. Etsion, V. Nowack, C. Fetzer, J. Hagemeyer, T. Jungeblut, N. Kucza, M. Kaiser, M. Porrmann, M. Pasin, V. Schiavoni, I. Rocha, C. Gttel, P. Felber. "LEGaTO: towards energy-efficient, secure, fault-tolerant toolset for heterogeneous computing". in *Proceedings of the 15th ACM International Conference on Computing Frontiers(CF)*, pp. 276-278, 2018.

[48] A. Cristal, O. S. Unsal, X. Martorell, P. Carpenter, R. D. L. Cruz, L. Bautista, D. Jimenez, C. Alvarez, B. Salami, S. Madonar, M. Perics, P. Trancoso, M. v. d. Berge, G. Billung-Meyer, S. Krupop, W. Christmann, F. Klawonn, A. Mihklafi, T. Becker, G. Gaydadjiev, H. Salomonsson, D. Dubhashi, O. Port, Y. Etsion, V. Nowack, C. Fetzer, J. Hagemeyer, T. Jungeblut, N. Kucza, M. Kaiser, M. Porrmann, M. Pasin, V. Schiavoni, I. Rocha, C. Gttel, P. Felber. "LEGaTO: First Steps Towards Energy-Efficient Toolset for Heterogeneous Computing". in *Proceedings of the International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulations (SAMOS)*, 2018.

[49] uniserver: A Universal Micro-server Ecosystem by Exceeding the Energy and Performance Scaling Boundaries http://www.uniserver2020.eu/

[50] M. J. Wirthlin, "Improving the reliability of FPGA circuits using triple-modular redundancy (TMR) & efficient voter placement". in *Proceedings of the 12th ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA)*, pp. 252:252, 2004.

[51] D. Ernst, N. S. Kim, S. Das, S. Pant, R. R. Rao, T. Pham, C. H. Ziesler, D. Blaauw, T. M. Austin, K. Flautner, and T. N. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation", in *Proceedings of the 36th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. pp. 7, 2003.

[52] E. A. Stott, J. M. Levine, P. Y. K. Cheung, and N. Kapre, "Timing fault detection in FPGA-based circuits", in "Proceedings of the 22th IEEE International Symposium Field-Programmable Custom Computing Machines (FCCM)", pp. 96-99, 2014.

[53] G. Miller, C. Carmichael, and G. Swift, "Single-event upset mitigation for xilinx FPGA block memories", *XILINX Application Note*, 2007.

[54] G. Yalcin, O. S. Unsal, and A. Cristal, "FaulTM: error detection and recovery using hardware transactional memory", in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 220-225, 2013.

[55] J. Yu, A. Lukefahr, D. J. Palframan, G. S. Dasika, R. Das, and S. A. Mahlke, "Scalpel: Customizing DNN pruning to the underlying hardware parallelism", in *Proceedings of the 44th ACM International Symposium on Computer Architecture (MICRO)*, pp. 548-560, 2017.

[56] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. Horowitz, and B. Dally, "EIE: efficient inference engine on compressed Deep Neural Network", in *Proceedings of the 43th IEEE/ACM International Symposium on Computer Architecture (ISCA)*, pp. 243-254, 2016.

[57] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient processing of Deep Neural Networks: A tutorial and survey", *arXiv:1703.09039*, vol. 105, no. 12, pp. 2295-2329, 2017.

[58] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review", in *IEEE Access*, vol. 5, pp. 17322-17341, 2017.

[59] D. S. Phatak and I. Koren, "Complete and partial fault tolerance of feedforward neural nets", in *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 446-456, 1995.

[60] J. Zhang, K. Rangineni, Z. Ghodsi, and S. Garg, "ThUnderVolt: Enabling Aggressive Voltage Underscaling and Timing Error Resilience for Energy Efficient Deep Neural Network Accelerators", in *Proceedings of 55th Design Automation Conference (DAC)*, 2018.

[61] R. Brandon, U. Gupta, L. Pentecost, P. Whatmough, S. K. Lee, N. Mulholland, D. Brooks, and G. Wei. "Ares: a framework for quantifying the resilience of deep neural networks." in *Proceedings of the 55th Annual Design Automation Conference, p. 17. ACM*, 2018.

[62] J.J. Zhang, T. Gu, K. Basu, and S. Garg. "Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator". in *36th VLSI Test Symposium (VTS)*, pp. 1-6, 2018.

[63] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon, "In-datacenter performance analysis of a tensor processing unit", in *Proceedings of the 44th IEEE/ACM International Symposium on Computer Architecture (ISCA)*. pp. 1-12, 2017.

[64] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array", in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 915-924, 2017.

[65] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for Deep Convolutional Neural Networks", in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, 2017.

[66] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "Yodann: An architecture for ultralow power binary-weight CNN acceleration", in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 48-60, 2018.

[67] E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. O. G. Hock, Y. T. Liew, K. Srivatsan, D. J. M. Moss, S. Subhaschandra, and G. Boudoukh, "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?", in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, pp. 5-14, 2017.

[68] S. I. Venieris and C. S. Bouganis, "Latency-driven design for FPGA-based Convolutional Neural Networks", in *Proceedings of the IEEE International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-8, 2017.

[69] J. Cheng, P. Wang, G. Li, Q. Hu, and H. Lu, "Recent advances in efficient computation of Deep Convolutional Neural Networks", in *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 64-77, 2018.

[70] S. K. Rethinagiri, O. Palomar, J. A. Moreno, O. S. Unsal, and A. Cristal, "Trigeneous platforms for energy efficient computing of HPC applications", in *Proceedings of the 22th IEEE International Conference on High Performance Computing (HiPC)*, pp. 264-274, 2015.