UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Aggressive undervolting of FPGAs: power & reliability trade-offs

Behzad Salami

Aggressive Undervolting of FPGAs: Power & Reliability Trade-offs

Behzad Salami

Department of Computer Architecture

Universitat Politècnica de Catalunya (UPC)

A thesis submitted for the degree of

Doctor of Philosophy in Computer Architecture

November, 2018

Advisor: Dr. Adrián Cristal Kestelman

Co-Advisor: Dr. Osman S. Unsal

For my Family

Acknowledgements

First of all, I would like to express my deep and sincere gratitude to my Ph.D. advisors Dr. Adrian Cristal Kestelman and Dr. Osman S. Unsal for giving me this great opportunity to attend their research group at BSC, and also, for their advice, guidance, confidence, and patient demeanor later during my Ph.D. studies. Without their deep technical knowledge to guide me, my Ph.D. studies would never successfully happen. With a warm and friendly behavior, they showed me in practice how to enjoy of working and how research can provide satisfaction if I am building something useful. What I learned from them will stay with me in my professional and personal life forever.

I would also like to express my acknowledge to Professor Mateo Valero. Mateo's leadership has fostered an outstanding academic atmosphere at BSC that I was fortunate to be a part of. I had also a great chance to travel with Mateo and his lovely wife Angela to my home country, Iran. For me, this trip was full of good times, incredible experiences, and unique lessons. These experiences and lessons will play considerable role to shape of my future professional and personal life.

During my Ph.D. studies, I had chances to collaborate with many deeply knowledgeable researchers. Some of these experiences were really unique for me. I would like to use this opportunity to send my thanks message to Dr. Pradip Bose, Dr. Alper Buyuktosunoglu, and Dr. Augusto Vega

from IBM Watson, a major part of this thesis was done in collaboration with them, and also, to Dr. Dirk Koch from University of Manchester who was my internship advisor.

I would also like to extend my thanks to the my Ph.D. defense and pre-defense committee, including Prof. Hamid Sarbazi-Azad, Prof. Onur Mutlu, Dr. Daniel Jimenez Gonzalez, Dr. Miquel Moreto, and Dr. Carles Hernandez Luz for their time to review and precise comments for improving the thesis.

Also, I acknowledge to outstanding and friendly staff at BSC and UPC for their support over the years.

I was fortunate to have had brilliant colleagues which turned into great friends including Oscar Palomar, Daniel Nemirovski, Azam Seyyedi, Burcu Ozcelik Mutlu, Gulay Yalcin, Nikola Marković, Nehir Sonmez, Oriol Arcas-Abella, Gorker Alp Malazgirt, Ivan Ratković, Alberto Gonzalez, Julian Pavon, Leonardo Bautista-Gomez, Oyku Melikoglu, Albert Kahira, Cristobal Ramirez, Kai Keller, Santhosh Kumar Rethinagiri, Milan Stanić, Tugberk Arkose, Damian Roca, Josue Esparza, Javier Arias, Adria Armejach, and Gina Alioto.

During my stay at Barcelona, I had the chance to meet many people in my after-work life, and some of them turned into great friends. We shared our good or bad times together. I would like to thanks my friends including Jamileh Jafari, Hamid Tabani, and Pouya Esmaili.

Finally, I would like to send my deep thanks to my lovely family, my brothers Sadegh, Turaj, and Ayhan, my sister, Zarrin, and specially my parents, Hossein and Roghiyeh, for their boundless dedication and encouragement.

This thesis received funding from the European Union's Horizon 2020 Programme under the LEGaTO project (www.legato-project.eu), grant agreement n° 780681, and the European Union Seventh Framework Program (FP7) under the AXLE project (www.axleproject.eu), grant agreement n° 318633. Also, I got grant from HiPEAC for my three-months internship at University of Manchester. Finally, this thesis is in part supported by Ministry of Economy and Competitiveness of Spain under contract n° TIN2015-65316-p.

Abstract

In this work, we evaluate aggressive undervolting, *i.e.*, voltage underscaling 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 confirm a large voltage guardband on FPGAs. In turn, significant power consumption is saved, by eliminating this voltage guardband; however, further undervolting may cause reliability issues as the result of the circuit delay increase, and faults might start to appear. We perform a detailed fault characterization in terms of the rate, location, type, as well as experimentally analyzing the sensitivity to environmental temperature, with a primarily focused on FPGA on-chip memories, or Block RAMs (BRAMs). Understanding this behavior can allow to deploy efficient mitigation techniques, and in turn, FPGA-based designs can be improved for better energy, reliability, and performance trade-offs.

Finally, as a case study, we evaluate a typical FPGA-based Neural Network (NN) accelerator when the FPGA voltage is underscaled. In consequence, the substantial NN energy savings come with the cost of NN accuracy loss. To attain power savings without NN accuracy loss below the voltage guardband gap, we propose a novel technique and also evaluated the built-in ECC mechanism of BRAMs. Hence, we develop an

application-dependent BRAMs placement technique that relies on the deterministic behavior of undervolting faults and mitigates these faults by mapping the most reliability sensitive NN parameters to BRAM blocks that are relatively more resistant to undervolting faults. Finally, as a more general technique, we apply the built-in ECC of BRAMs and observe a significant fault coverage capability thanks to the behavior of undervolting faults, with a negligible power consumption overhead.

Keywords: FPGA, Voltage Scaling, Power Consumption, Reliability

Abbreviations

ASIC.....Application-Specific Integrated Circuit
BRAM.....Block Random Access Memory
COP.....Critical Operating Point
CPU.....Central Processing Unit
DRAM.....Dynamic Random Access Memory
DSP.....Digital Signal Processor
DVFS.....Dynamic Voltage Frequency Scaling
ECC.....Error Correction Code
FIP.....Fault Inclusion Property
FPGA.....Field Programmable Gate Array
FPU.....Floating Point Unit
FVM.....Fault Variation Map
GPU.....Graphic Processor Unit
HDL.....Hardware Description Language
HLS.....High-Level Synthesis
HPC.....High Performance Computing
ICBP.....Intelligently-Constrained BRAMs Placement

ITD.....Inverse Temperature Dependency
LUT.....Look-Up Table
ML.....Machine Learning
NN.....Neural Network
RAM.....Random Access Memory
RTL.....Register-Transfer Level
SECDED.....Single-Error Correction and Double-Error Detection
SOC.....System On Chip
SRAM.....Static Random Access Memory
TMR.....Triple Modular Redundancy

Contents

1 Introduction

In this chapter, we provide background information about the scope of the thesis and later on, explain the key challenges, motivations, our solutions, and finally, introduce the thesis outline.

1.1 Background

The concentration of this thesis is on aggressive undervolting for commercial Field Programmable Gate Arrays (FPGAs). Hence, in this section, we briefly introduce these concepts.

1.1.1 FPGA Architecture

In modern computing systems, FPGAs play a crucial role to accelerate state-of-the-art applications, thanks to their inherent capability to execute computations in streaming fashion on a massively parallel substrate. FPGAs are increasingly employed within the modern data centers and are expected to be in 30% of data centers by 2020 [7]. They are used to accelerate many state-of-the-art applications such as database query processing [19], [139], [141], [140], Neural Networks (NN) [59], [74], and genome sequence analytic [14], among others. FPGAs combine the

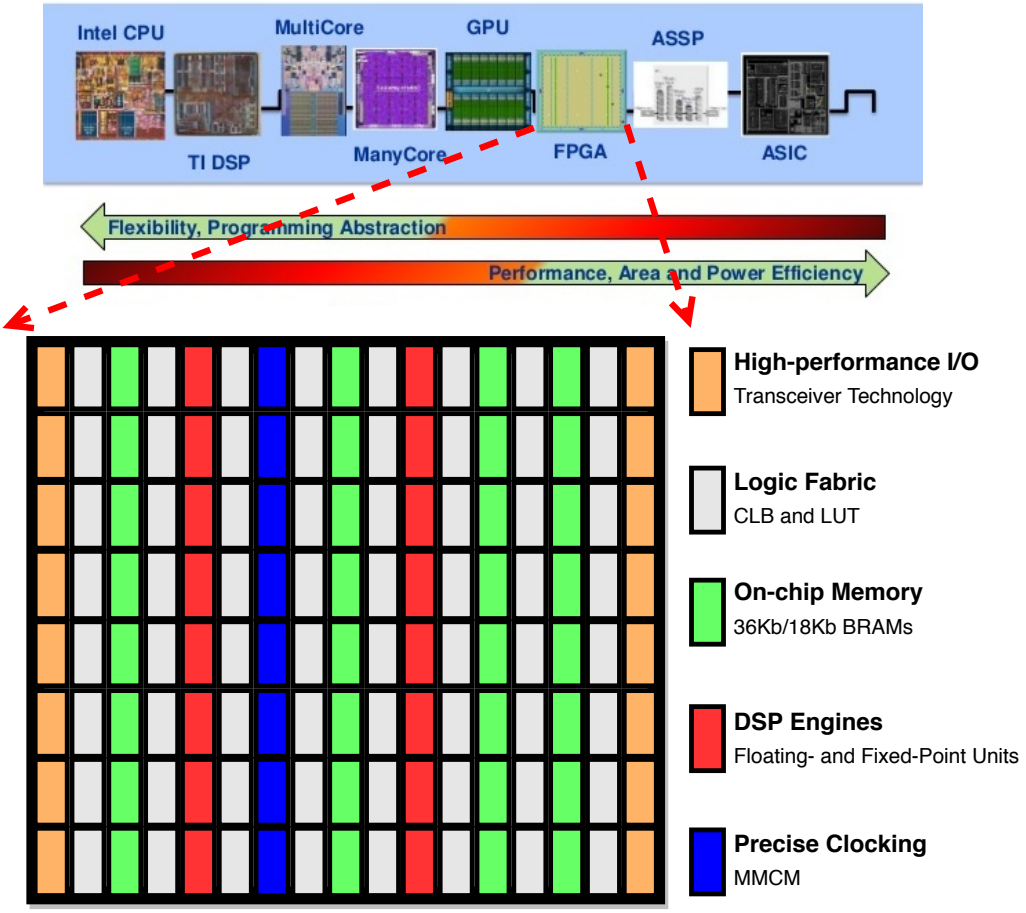


Figure 1.1: FPGAs among other digital devices (source: Intel/Altera [2]).

flexibility of CPUs with the efficiency of Application-Specific Integrated Circuits (ASICs), see Figure 1.1. Hence, the concentration of this thesis is to study FPGAs with the aim of making them more power-efficient, which can suit them for power-constrained environments. Modern FPGAs are composed of a wide range of reconfigurable components, *e.g.*, Block RAMs (BRAMs), Digital Signal Processors (DSPs), Configurable Logic Blocks (CLBs), among others. These components in a tightly-coupled structure can be efficiently exploited to accelerate computation-, memory, or I/O-intensive applications to achieve the goal of high-throughput computation. Usually, these components are floorplanned in a column-oriented way, as

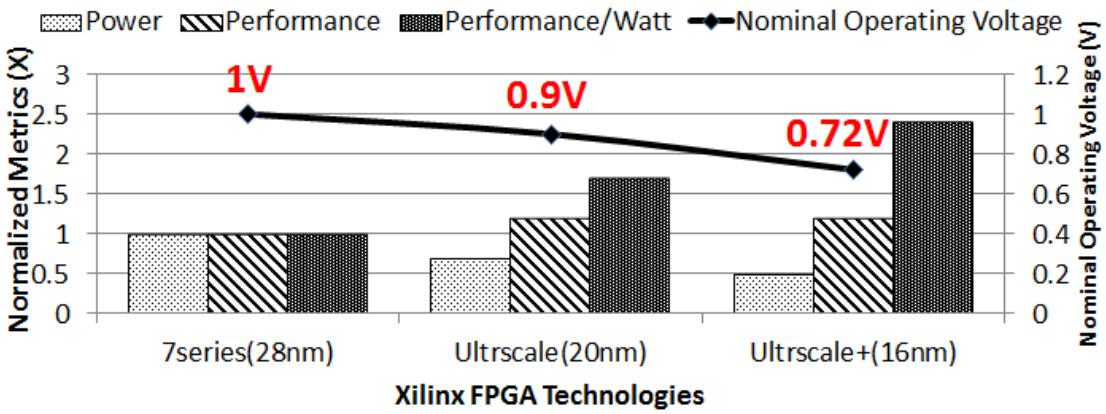


Figure 1.2: Voltage scaling in in Xilinx device family generations [10].

illustrated in Figure 1.1. For computations, corresponding components need to be appropriately configured at run- or compile-time. Most of the state-of-the-art commercial FPGAs including the studied commercial FPGAs in this thesis are Static RAM (SRAM)-based, which means that the configuration bitstream is stored in on-chip SRAMs and FPGA components get configured by reading the corresponding parts of this bitstream.

1.1.2 Aggressive Undervolting

The power consumption of digital circuits, *e.g.*, FPGAs is directly related to their supply voltage level. Hence, any voltage underscaling can directly deliver power/energy efficiency gains. For instance, as shown in Figure 1.2, for Xilinx FPGA generations, the nominal operating voltage has been lowered from 1V in Virtex-7 series (28nm) to 0.72V in Ultrascale+ series (16nm); resulting in 1.2X and 2.4X in power and performance/watt efficiency, respectively [10]. As a more aggressive effort, for each technology node, the supply voltage underscaling below the standard nominal level can deliver further power savings. We target this approach in the thesis.

For different types of chips such as CPUs [22], [72], [153], [127], [172], [171], Graphics Processing Units (GPUs) [91], ASICs [165], Dynamic RAMs (DRAMs) [36], and SRAMs [174] it has been experimentally shown that the nominal operating voltages set by vendors are extremely conservative for real-world applications. This phenomenon is due to the voltage guardband added by vendors to ensure the correct operation under worst-case environmental and process conditions. Thus, as earlier mentioned, the promising approach to achieve energy efficiency is the aggressive undervolting, *i.e.*, voltage underscaling below the standard nominal level. However, as the works above as well as our experimental studies confirm, the potential of the aggressive undervolting is fully vendor-, chip-, architecture- dependent. The concentration of this thesis is to study aggressive undervolting for commercial FPGAs experimentally.

However, the downside of the aggressive undervolting is that voltage underscaling below the voltage guardband can cause reliability issues and faults might start to appear. Unlike the DVFS technique, the frequency is not scaled down in the aggressive undervolting approach. Therefore energy savings can be more significant. However, aggressive undervolting leads to timing related faults, which can cause applications to crash or terminate with wrong results. This thesis aims to extend the aggressive supply voltage underscaling approach, *i.e.*, power and reliability trade-off, detailed fault characterization, and effective mitigation for FPGAs.

1.2 Key Challenges and Motivations

In comparison to ASICs, the power consumption of FPGAs is a first-order concern, especially in nano-scale manufacturing technologies. It has been shown that the power and energy efficiency of FPGAs is estimated to be ∼10X-∼20X worse than in the corresponding ASIC designs [135], [183], [83], [122], [123] as it is also

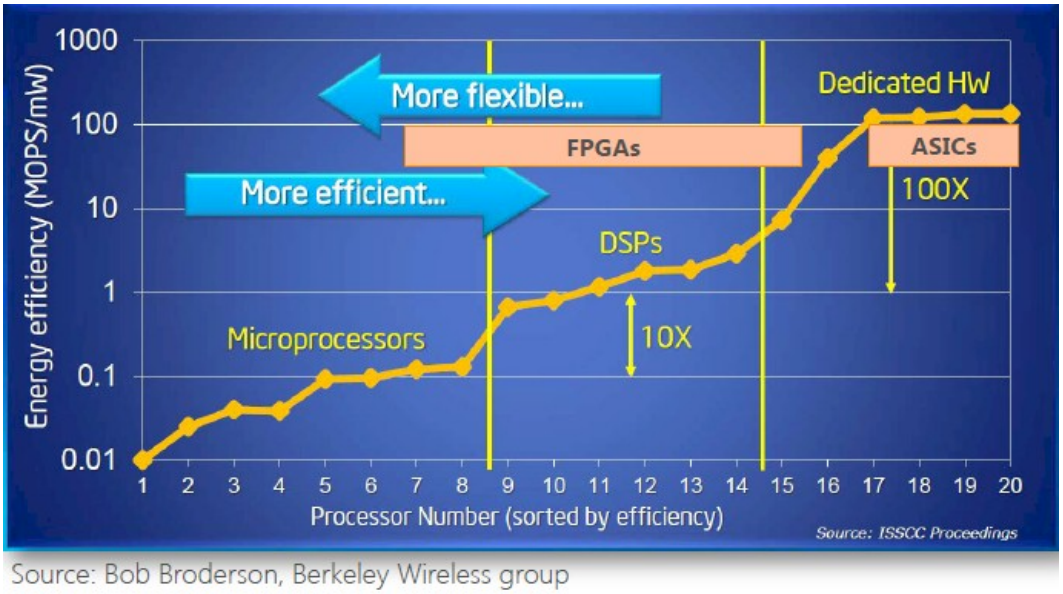


Figure 1.3: Detailed energy efficiency of FPGAs [1].

confirmed from industry perspective, see Figure 1.3. This gap is due to the inherent structure of FPGA resources, which provide the configurability as an advantage against ASICs; however, configurability incurs additional power consumption overhead. The relatively lower power and energy efficiency of FPGAs can make them less attractive for power-limited environments such as high performance embedded computing scenarios and mobile environments, among others. Thus, the key challenge that this thesis aims to tackle is the power/energy dissipation of FPGAs.

There are many techniques deployed to minimize the power consumption of FPGAs, such as architectural improvements [10] [3], power-aware tools [43] [85] [49], bitstream compression [125], clock or voltage gating [126] [182], [95], [162] among others. This thesis concentrates on an orthogonal approach, aggressive undervolting. As summarized in equations 1.1, 1.2, and 1.3, the total power consumption including the dynamic, *i.e.*, the signal transition power, and static power, *i.e.*, leakage power, are directly functions of the supply voltage [5]; thus, as expected, any

undervolting can directly lead to the power consumption reduction.

$$P_{total} = P_{dynamic} + P_{static} \tag{1.1}$$

$$P_{dynamic} = \alpha.C.f.V^2 \tag{1.2}$$

$$P_{static} = \sum Leakage\_Current.V \tag{1.3}$$

, where α, C, f, V, and $Leakage\_Current$ are the technology-dependent constant coefficient, capacitance, working frequency, supply voltage, and total leakage current of the digital circuit, respectively.

With the aim of evaluating the aggressive undervolting technique to achieve energy-efficient FPGA-based accelerators, the key research questions that are answered by this thesis are listed below:

1. Is there any potential in FPGAs to take advantage of aggressive undervolting?
2. Do technology architecture, process variation, etc., play any role in the impacts of aggressive undervolting in FPGAs?
3. Is the effect of aggressive undervolting on the reliability deterministic or stochastic?
4. Which kind of real-world application can take advantage of the energy efficiency through aggressive undervolting?

1.3 Scope of the Thesis

To have a thorough study, our experiments include several representative platforms from Xilinx, a main vendor, *i.e.*, VC707 (performance-optimized architecture) [11],

ZC702 (FPGA integrated with ARM-core) [13], and two identical samples of KC705 (power-optimized architecture) [12] platforms. These four platforms allow us to study different architectures and also the impact of die-to-die process variation for KC705. Experimentally confirming the voltage guardband for multiple components of FPGAs, we observe that data can be safely retrieved without any observable fault when the supply voltage is underscaled below the nominal level, *i.e.*, V_{nom}, and until a certain minimum safe voltage level, *i.e.*, V_{min}. Further voltage underscaling causes faults.

For a more detailed study, the concentration of this thesis is BRAMs, since BRAMs play a key role in the acceleration of state-of-the-art applications such as NNs and bioinformatics [14], and also, they considerably contribute in the total power consumption of such FPGA designs of up to 30% [59]. Also, unlike many FPGA components, the supply voltage of BRAMs can be independently regulated, which allows detailed power and reliability trade-off analysis. Hence, the reliability aspects of BRAMs under aggressively low-voltage operations are extensively studied. This study includes the characterization of faults in terms of the rate, location, type as well as the impact of the environmental temperature.

As a case study application, we concentrate on the NN accelerator. NNs are state-of-the-art applications that are increasingly used in the context of many real-world environments such as autonomous cars [160], [143], [65], mobile scenarios [86], [90], [73], personalized medicine [39], [51], [67], game industry [147], among others. Also, due to the size of matrices that NNs needs to compute, the computation and power required is significant [146], [154]. To achieve energy-efficient NN, hardware accelerators such as GPU- [124], FPGA- [134], and ASIC-based [70], [135], [63] systems have recently received significant attention. Among them, FPGA-based accelerators have unique features such as the relatively short deployment time versus ASICs and more energy-efficient against GPUs. Hence, to achieve

an energy-efficient FPGA-based NN accelerator, we push a typical accelerator to operate under low-voltage FPGA BRAMs, and evaluate mitigation techniques to prevent NN accuracy loss as the result of undervolting faults.

1.4 Contributions

This thesis aims to evaluate the aggressive undervolting technique for commercial FPGAs empirically. Toward this goal, the thesis has three main contributions, which are summarized as follows:

1. **Voltage Guardband:** This thesis is the first effort to empirically study aggressive voltage underscaling of FPGAs below the standard nominal level. Through experimenting on four platforms, we confirm a conservative voltage guardband until the minimum safe voltage level, *i.e.*, V_{min} for different FPGA components. By eliminating this large voltage gap, a significant power saving gain is achieved without compromising to the performance or reliability, for instance, more than an order of magnitude power savings for on-chip BRAMs.

2. **Fault Characterization:** We perform the first detailed experimental bit-level characterization study of the behavior of faults when the supply voltage of FPGA on-chip BRAMs is underscaled below V_{min}. Understanding the behavior of these faults can provide an opportunity to deploy efficient mitigation techniques, and in turn, a better trade-off for low-voltage FPGA-based designs can be achieved. More specifically, we observe that:

    - The fault rate exponentially increases by further undervolting; however, with a considerable difference among platforms, which is the result of technological differences and also process variation.

- The location and rate of undervolting faults do not considerably change over time. In other words, undervolting faults exhibit a deterministic behavior.
- Within BRAMs, faults usually occur in certain few columns; however, these most-vulnerable columns are different among all BRAMs.
- Undervolting faults are fully non-uniformly distributed among BRAMs.
- Undervolting faults manifest themselves mostly as '1' to '0' bit-flips.
- On a given BRAM raw, undervolting faults lead mostly to single-bit type faults. Multi-bit type faults start to appear as the voltage is further reduced.
- At higher environmental temperatures, the fault rate reduces as the result of the Inverse Temperature Dependence (ITD) property of the nano-scale technology nodes [117].
- Undervolting faults follow the Fault Inclusion Property (FIP), *i.e.*, faults in a certain voltage level, stay (and potentially extend) in lower voltages, as well.

3. **FPGA-Based Accelerator:** We perform the first study of the efficiency of NN accelerators under the aggressively low-voltage operation of commercial FPGAs. We observe that the data sparsity of state-of-the-art NN benchmarks makes them inherently robust against undervolting faults; however, by aggressive undervolting, the NN accuracy is impacted. To attain the subsequent power saving without NN accuracy loss, we present two fault mitigation techniques, which rely on the behavior of undervolting faults.

1.5 Outline

The subsequent sections of this thesis are structured as follows. The FPGA undervolting experimental methodology, and also, the major behavior of the power and reliability trade-off is explained in Chapter 2. The fault characterization under aggressive low-voltage FPGA operations is detailed in Chapter 3. Chapter 4 explains the effect of FPGA undervolting in the typical NN and evaluate the proposed fault mitigation techniques. Chapter 5 reviews the recent related works and Chapter 6 summarizes our findings and lessons learned in this study. Finally, Chapter 7 includes the list of publications from the thesis.

2 Understanding FPGAs Undervolting

In this chapter, we introduce the experimental methodology, elaborate the FPGA platform undervolting, and also, discuss the behavior of FPGA BRAMs under aggressively reduced supply voltage.

2.1 Experimental Methodology

We perform our experiments on a set of representative commercial FPGA platforms from Xilinx, *i.e.*, one VC707, one ZC702, and two KC705s. Common for all platforms, BRAMs are distributed all over the chip with a unique size of 16 Kbits each. Each BRAM is a matrix of bitcells with 1024 rows, and 16 columns[1]. 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. More details of our tested platforms are shown in Table 2.1. All platforms are fabricated with 28nm technology, and the standard nominal voltage of BRAMs is the same, $V_{nom} = 1V$. However, VC707 is designed for performance while KC705 is optimized for the power consumption. Also, a different design approach is used for

[1]Each row has two additional bits as parity that is not considered in this section. We will elaborate on their role in Section 4.4.2.

Table 2.1: Specifications of tested FPGA platforms.

Hardware Platform (Board)	VC707	ZC702	KC705*
Device Family	Virtex-7	Zynq7000	Kintex-7
Chip Model	XC7VX485T	XC7Z020	XC7K325T
Speed Grade	-2	-1	-2
Number of BRAMs	2060	280	890
Basic Size of Each BRAM	1024*16-bit	1024*16-bits	1024*16-bits
Technology Node	28nm	28nm	28nm
Nominal V_{CCBRAM} (V_{nom})	1V	1V	1V
Design Consideration	Performance	FPGA-CPU Architecture	Power

* Two identical samples of KC705 (A & B) are tested.

ZC702, which is targeted for hardware-software (FPGA-CPU) co-designs. Furthermore, we choose two KC705 platforms that allow us to evaluate the undervolting effects on the same model, as well. Hence, for a thorough evaluation, we selected these representative platforms.

2.2 FPGA Undervolting: Idle Power Minimization

Through the Power Management Bus (PMBUS) standard [4], it is possible to independently and dynamically regulate and monitor the supply voltage of such FPGA components as BRAMs (V_{CCBRAM}), core logic (V_{CCINT}), *i.e.*, Look-Up Tables (LUTs) and Digital Signal Processors (DSPs), among others. An on-board voltage regulator is responsible for this aim. Although there is no standard for the list of these components with the capability of independently regulated, the difference among our studied platforms is not significant. For instance, the on-board voltage distribution is shown in Figure 2.1 for VC707. To modify supply voltages, we use Texas Instrument (TI) PMBUS USB Adapter, and the provided C-based Application Programming Interface (API), which facilitates accessing the on-board voltage

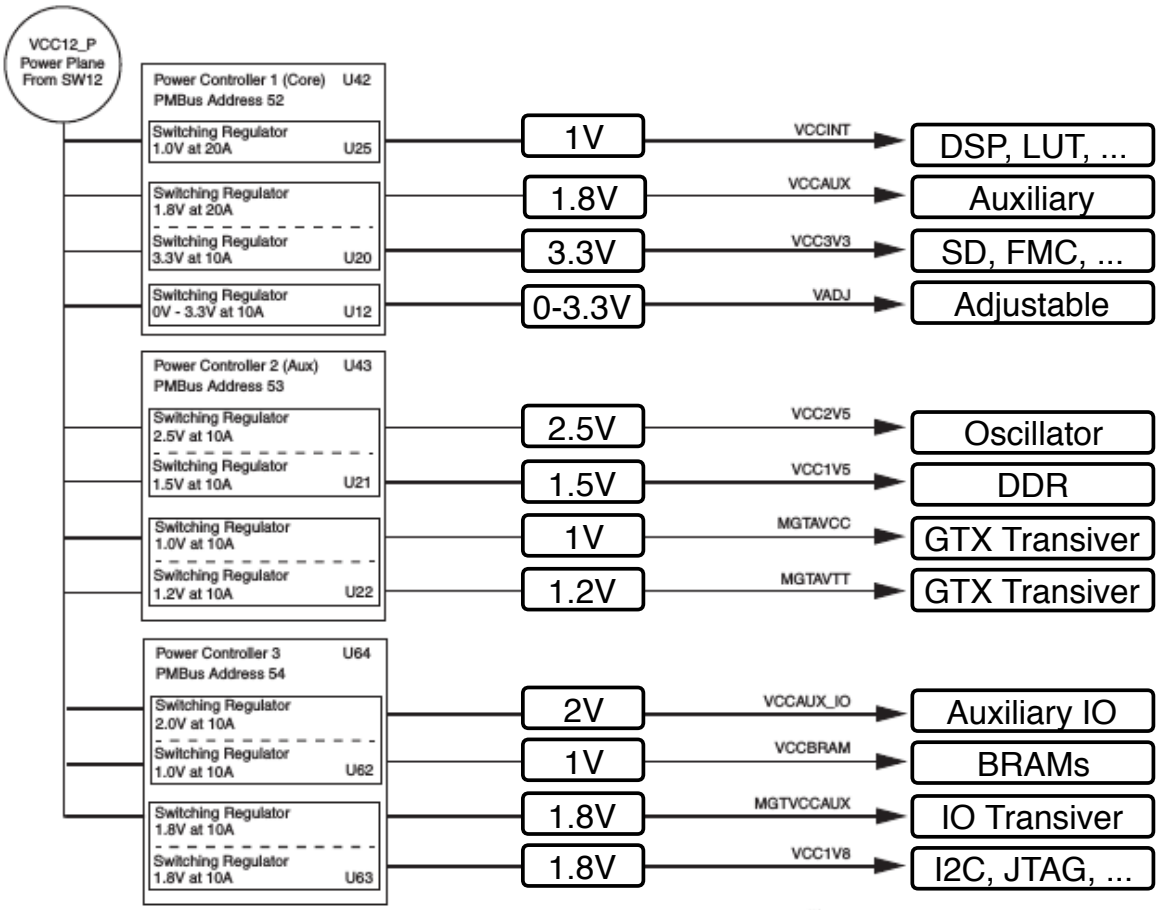


Figure 2.1: On-board voltage regulator for FPGAs, shown for VC707 [11].

controller through the host [6]. Note that in the studied platforms, the voltage regulator is hardwired to the host, and accessible through the PMBus standard.

As the first experiment, we aim to minimize the Idle power consumption of the FPGA platform, *i.e.*, the power that is dissipated when there is no application running on the board and all components can go to the idle mode, through undervolting the platform's components listed in Figure 2.1. Toward this goal, we set the supply voltage of the platform's components at a minimum voltage level that the platform does not crash, *i.e.*, $V_{idleMin}$. In turn, the on-board status LEDs are changed from Figure 2.3a at the nominal voltage level to 2.3b below the $V_{idleMin}$.

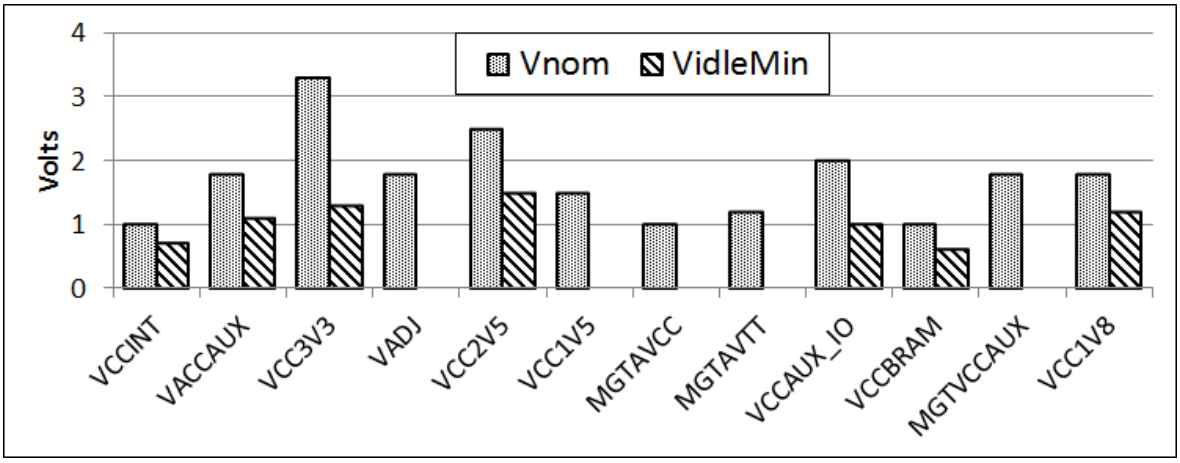


Figure 2.2: FPGA platform undervolting until the crash voltage level, shown for VC707 [11].

(a) At the standard nominal voltage level, *i.e.*, V_{nom}.

(b) Below the minimum Idle voltage level, *i.e.*, $V_{idleMin}$.

Figure 2.3: Status LEDs under different voltages, shown for VC707 [11].

It means that further undervolting causes the system crashing, which is exposed by an unset of the DONE pin. Note that at the $V_{idleMin}$, there is no guarantee that the system operates in a safe behavior; however, the FPGA bitstream is recognized as correct. Through this undervolting mechanism, the idle power consumption is significantly reduced, for instance, 1.9X for VC707 as shown in Figure 2.4.

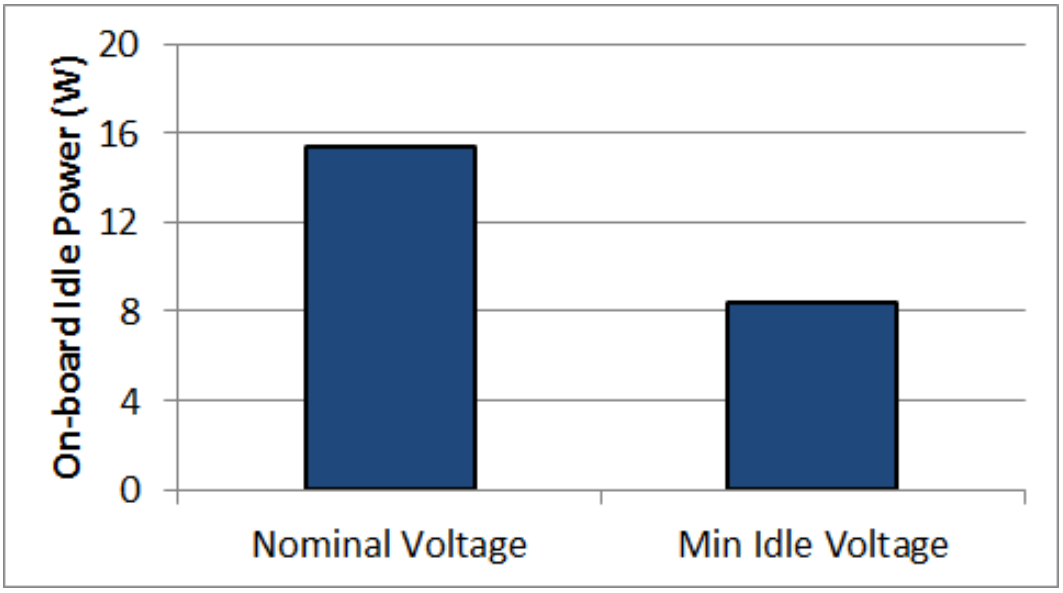


Figure 2.4: Minimized Idle power consumption through undervolting as detailed in Figure 2.2, shown for VC707 [11].

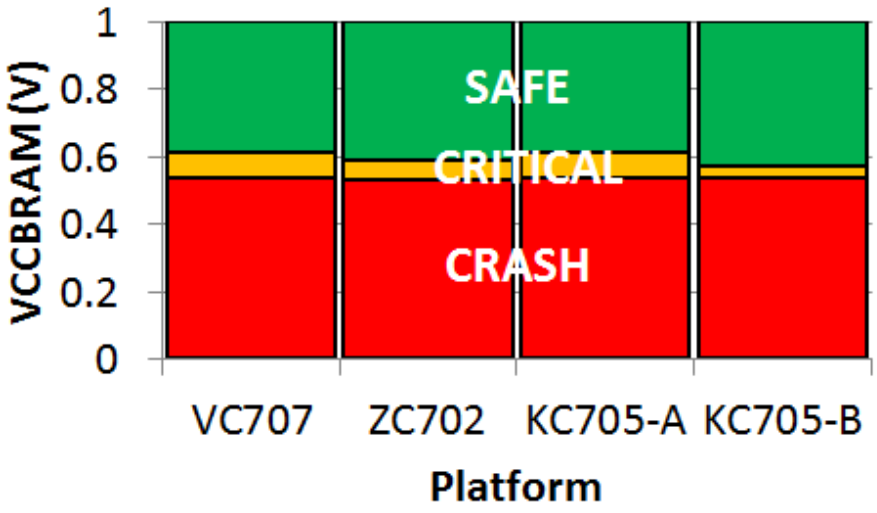
2.3 Safe, Critical, and Crash Voltage Regions

Unlike the idle power that is drawn when only the power supply is connected, application running can consume additional power consumption. In this section, we aim to discover the minimum safe voltage of such FPGA components as V_{CCINT} and V_{CCBRAM}. These voltage rails feed the most important on-chip FPGA components. As can be seen in Figure 2.5, for both V_{CCINT} and V_{CCBRAM}, there is a conservative voltage guardband for all platforms below the nominal level (SAFE), which creates an opportunity for energy savings. Further undervolting causes observable faults (CRITICAL), until a voltage level that platforms stop operating (CRASH). Note that for all platforms, the nominal voltage level of both voltage rails is 1V; however, other voltage levels slightly vary among platforms. For the 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

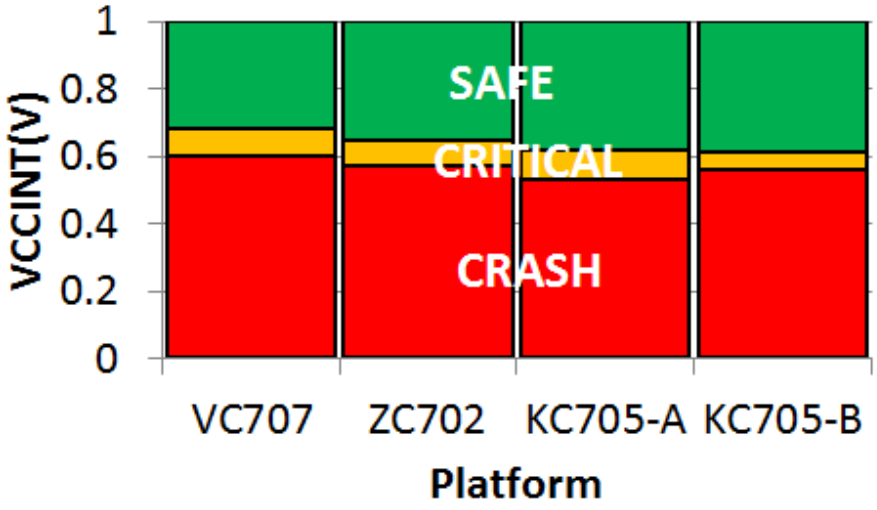
V_{CCINT} that feeds several components such as LUTs and DSPs. Further power and reliability trade-off of the BRAMs at the critical region is discussed later in this section.

The experimental setup of BRAMs evaluation is shown in Figure 2.6. 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 ARM processor; however, in other platforms, we built our 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, it initializes BRAMs and analyzes potentially faulty data retrieved from BRAMs. On this setup, the reduced V_{CCBRAM} can cause the timing violations and in turn, corrupting some of the bitcells of some of BRAMs. We follow the method shown in List. 2.1 to analyze the behavior of these faults comprehensively.

Then, we retrieve the 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 chapter 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 operate, 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 including dynamic and static, which are both directly reduced by undervolting. Note that BRAMs considered in this thesis internally operate



(a) VCCBRAM.



(b) VCCINT.

Figure 2.5: Undervolting FPGA components, *i.e.*, Internal (V_{CCINT}) and BRAM (V_{CCBRAM}) voltages. (SAFE: no observable fault occur. CRITICAL: faults manifest. CRASH: FPGA stops operating.)

at a fixed frequency of $\sim$ 500Mhz [10], and externally the design is operating on the maximum frequency without timing violation at the nominal voltage level,

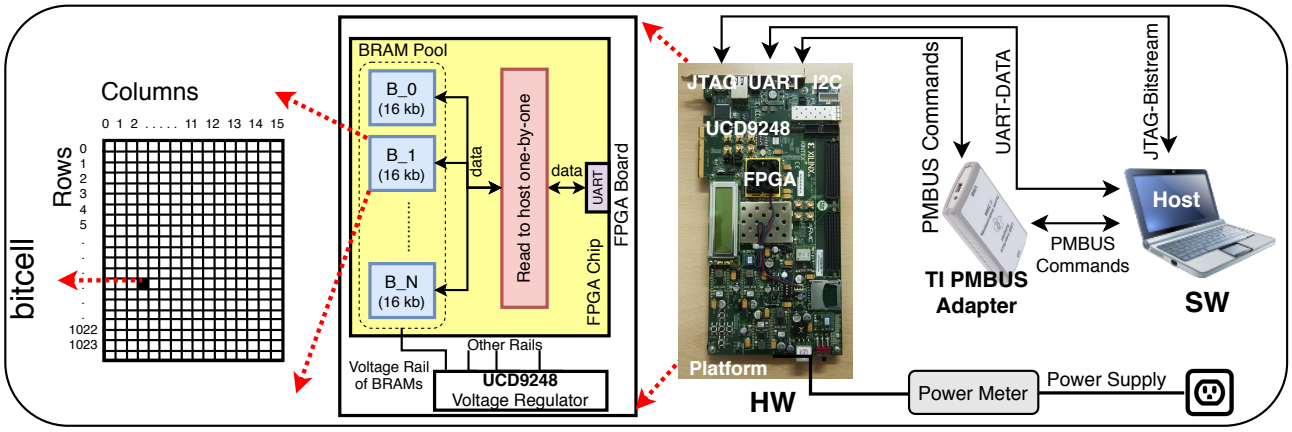


Figure 2.6: Experimental setup to perform fault characterization through FPGA BRAMs aggressive voltage underscaling.

List 2.1: Pseudo-code to restudy liability behavior of voltage scaling on FPGA BRAMs at the CRITICAL, on the experimental setup of Figure 2.6.

```
1: V_CCBRAM = V_min;
2: while(V_CCBRAM >= V_crash) 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:     V_CCBRAM− = 10(mV);
10: end
```

determined by the FPGA compiling tools.

2.4 Power and Reliability Trade-offs

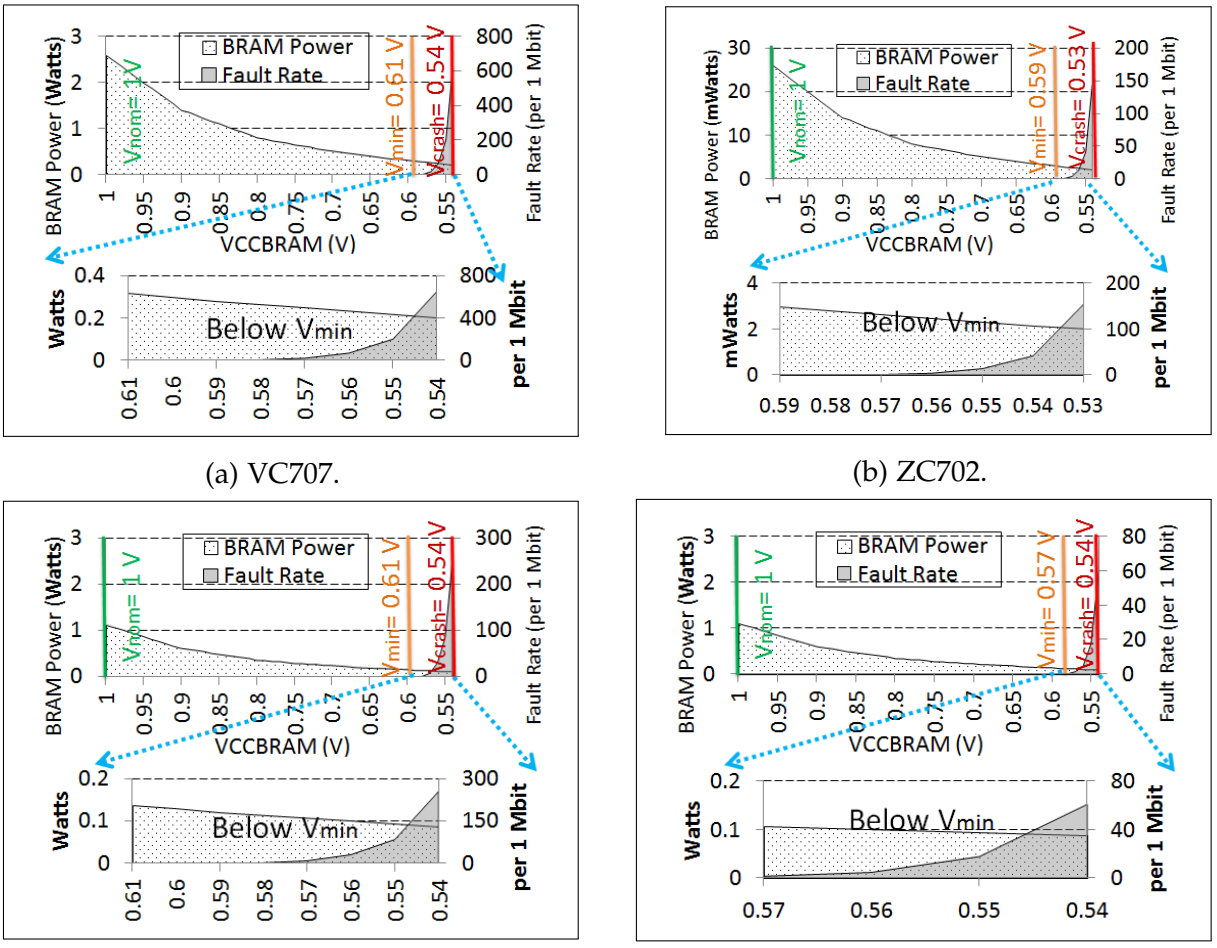
This section describes the overall behavior of FPGA BRAMs in terms of the power consumption and reliability trade-off when their supply voltage is aggressively re-

duced. We repeat experiments on all platforms, mentioned earlier. As can be seen in Figure 2.7, our experiments on lowering the supply voltage of BRAMs below nominal level, V_{nom}, demonstrate two thresholds. *First*, a voltage guardband, V_{min}, that separates the fault-free and faulty regions. *Second*, V_{crash} that is the lowest level of the voltage that our design practically operates. For all tested platforms, $V_{nom} = 1V$ due to the factory settings. However, through our experiments, we observe a slight difference for V_{min} and V_{crash}. Note that repeating these tests in more noisy and harsh environments, *i.e.*, worst case environmental conditions, can cause observable faults above observed V_{min}, as well. Below the V_{crash} region, we observed that the DONE pin is unset, which at nominal levels indicates incorrect bitstream. To have an initial exploration, we evaluated the environmental temperature; however, the large guardband is experimentally observed.

The common observation for studied platforms is that when $V_{CCBRAM} >= V_{min}$, no observable faults occur. However, underscaling V_{CCBRAM} below V_{min} the fault rate exponentially increases, while the power consumption quadratically reduces but with different scales for different platforms. When $V_{CCBRAM} = V_{min}$, significant BRAMs power savings gain is achieved over $V_{nom} = 1V$, more than an order of magnitude, without comprising any performance or incurring any reliability degradation. As can be seen, both power consumption and reduction are less in KC705 than VC707, which is the consequence of having relatively fewer BRAMs and also 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 a much smaller number of BRAMs.

Further 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},

[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.



(a) VC707.

(b) ZC702.

(c) KC705-A.

(d) KC705-B.

Figure 2.7: Major observations under low-voltage operations in FPGA BRAMs for studied commercial platforms.
* Different scales for different charts.
** power results are reported as mWatts in ZC702 and in Watts for others.
*** At ambient temperature.

for VC707, ZC702, KC705-A, and KC705-B, respectively (with pattern= 16'hFFFF). Note that through our experimental observations, the vast majority of these faults are '1' to '0' bit flips, on average 99.9% for all platforms. We verify this observation by repeating the same tests with other data patterns. The fault rate is proportional to the number of '1' bits; for example, with pattern= 16'hFFFF the fault rate is

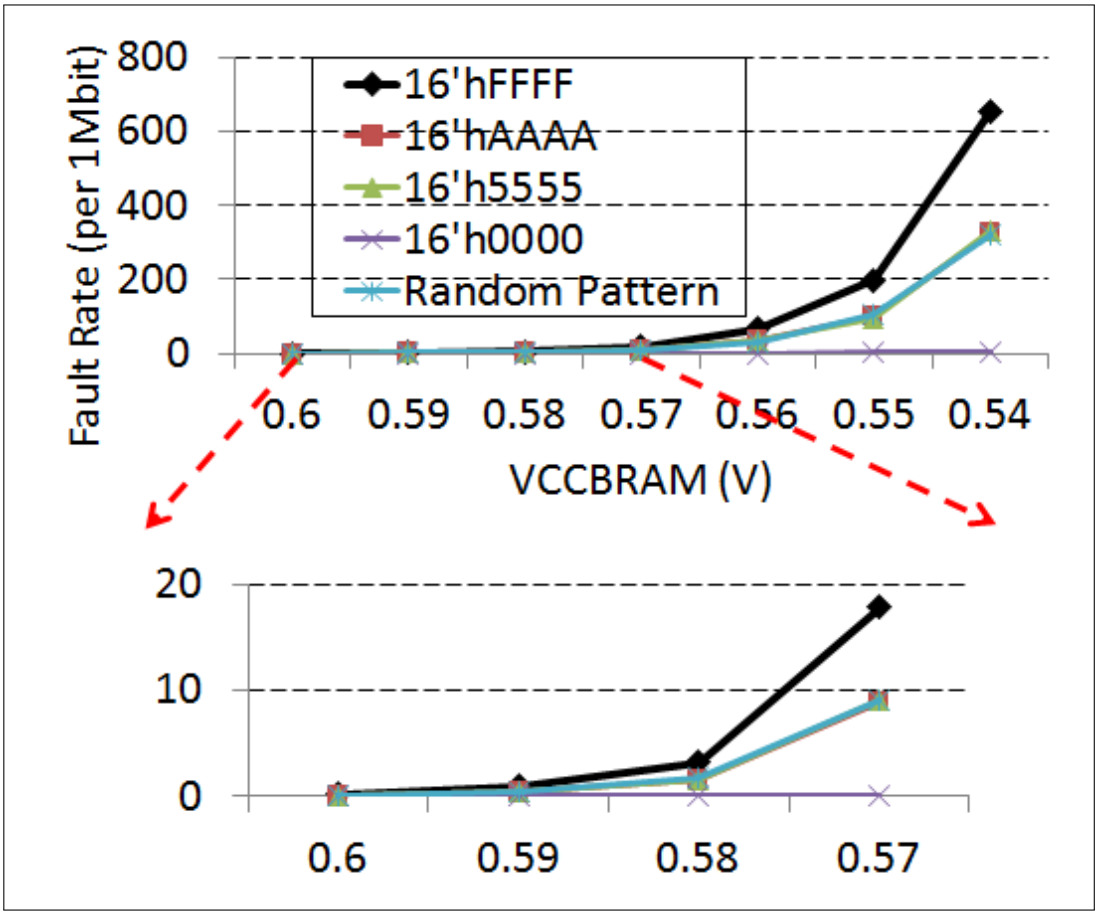


Figure 2.8: The impact of the data pattern in the fault rate on VC707 (similar behavior is observed for other platforms.)

almost double than pattern= 16'hAAAA, and with pattern=16'h0000 few faults are observed, as shown as an example on VC707 in Figure 2.8. In the same line, we did not observe any meaningful correlation in the various permutations of '0' and '1' in the data pattern, for instance, as can be seen, the fault rate of pattern= 16'hAAAA, 16'h5555, and a random pattern are almost the same.

3 Fault Characterization Through FPGA BRAMs Undervolting

In this section, we comprehensively characterize the behavior of faults, where V_{CCBRAM} is underscaled at the CRITICAL voltage region from V_{min} to V_{crash}. Considering the impact of the data pattern that is explained in Section 2, the detailed fault characterization in this section is for data pattern= 16'hFFFF, which corresponds the highest fault rate since as shown in Section 2, '1'-to-'0' bit flips are much more common than '0'-to-'1' flips.. Understanding the behavior of undervolting faults can allow the deployment of efficient fault mitigation techniques and in turn, better reliability, energy, and performance trade-off can be achieved for FPGA designs.

3.1 Fault Stability Over Time

As earlier mentioned, we repeat each test 100 times to get statistically significant results. We did not observe a significantly different results among different runs, as shown in Table 3.1. Thus, the fault rates and as experimentally observed, faults locations show a stable behavior over time without meaningfully changing over the time. This observation is considered in our application-aware fault mitigation

Table 3.1: Fault rate stability over time. (Fault rate analysis of 100 runs at V_{crash} with pattern=16'hFFFF.)

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.

technique that is discussed in Chapter 4.

3.2 Fault Variability Among BRAMs

By statistically analyzing the experimental results, we observe that faults are not uniformly distributed over different BRAMs. Common for all platforms, we observe that a big percentage of BRAMs, *e.g.*, 38.9% in VC707 at the lowest voltage level $V_{crash} = 0.54V$, never experience faults; however, faults manifest in a small percentage of them. 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 clusted as low-vulnerable. For instance, we show detailed results of VC707 in Figure 3.1. 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 1024*16-bits.

The fault rate variability among BRAMs is the result of the within-die process variation and as discussed earlier is permanent. Accordingly, 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,

Xilinx toolkit, 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 Figure 3.2, when V_{CCBRAM} is underscaled from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$. FVM has the granularity of BRAM. Note that in this figure other FPGA components are ignored for the sake of more clarity of FVM.

3.3 Fault Variability Within BRAMs

We performed a statistical analysis to analyze the fault distribution schema within those faulty BRAMs, from both column- and row-wise view.

3.3.1 Column-wise Fault Analysis

Our observations reveals that faults mostly occur in a few certain columns within BRAMs. In other words, for each faulty BRAM, a few (one or two) most vulnerable column of bitcells exist, among 16 available columns, as illustrated in Figure 3.3. Figure 3.4a shows the number of most vulnerable columns within faulty BRAMs. As can be seen, more than 80% of faults occur in a single column, 17% in two

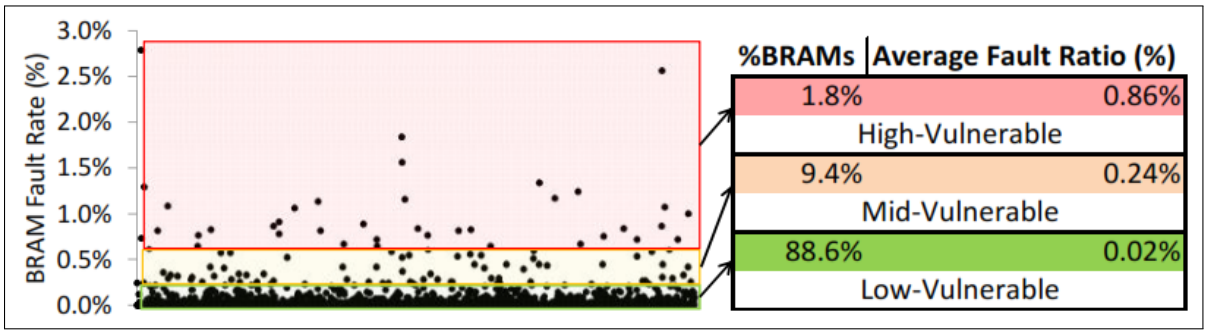


Figure 3.1: Clustering BRAMs to low-, mid-, and high-vulnerable classes using K-mean algorithm.
* This figure shows the clustering at $V_{crash} = 0.54V$ for only VC707 since very similar behavior is observed for other platforms.

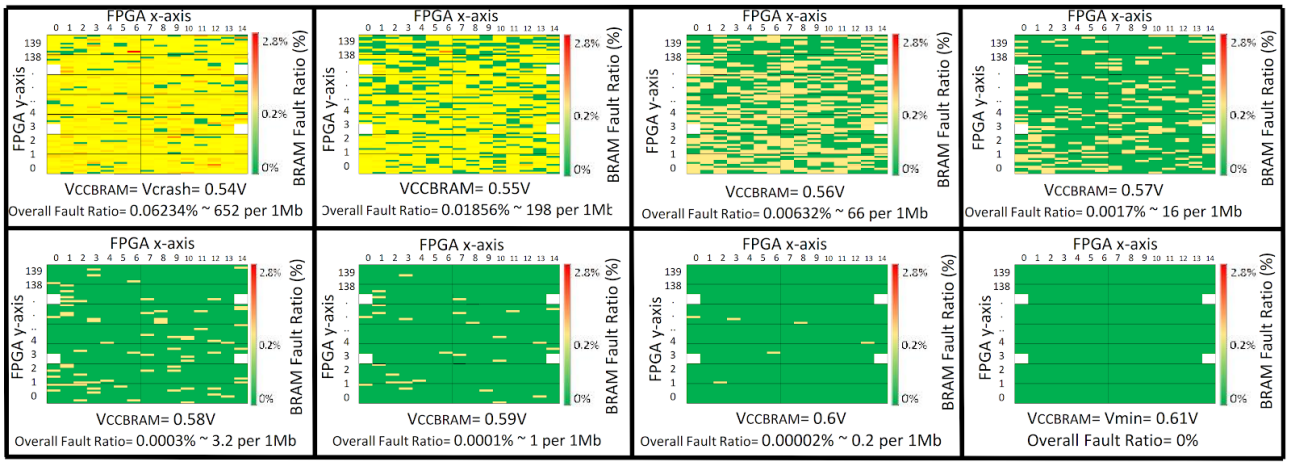


Figure 3.2: BRAMs Fault Variation Map (FVM), scaling V_{CCBRAM} from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$.

* 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.

columns, and so on. However, these most vulnerable columns do not occur in identical column indexes, for different BRAMs. As can be seen in Figure 3.4b, there is almost a uniform distribution of fault rate in different 16 available column indexes, when fault rate is averaged for all BRAMs.

3.3.2 Row-wise Fault Analysis

By a statistical analysis of locations of faults within faulty BRAMs, we observe that there is a spatial correlation between faulty rows. Our hypothesis is that the BRAM wordlines are weaker than bitlines to support low voltage operation. However, we were not able to verify this hypothesis since there is no publicly available document that details the circuit level design of Xilinx BRAMs. In other words, by increasing the distance between rows, the probability of the fault is considerably reduced. As

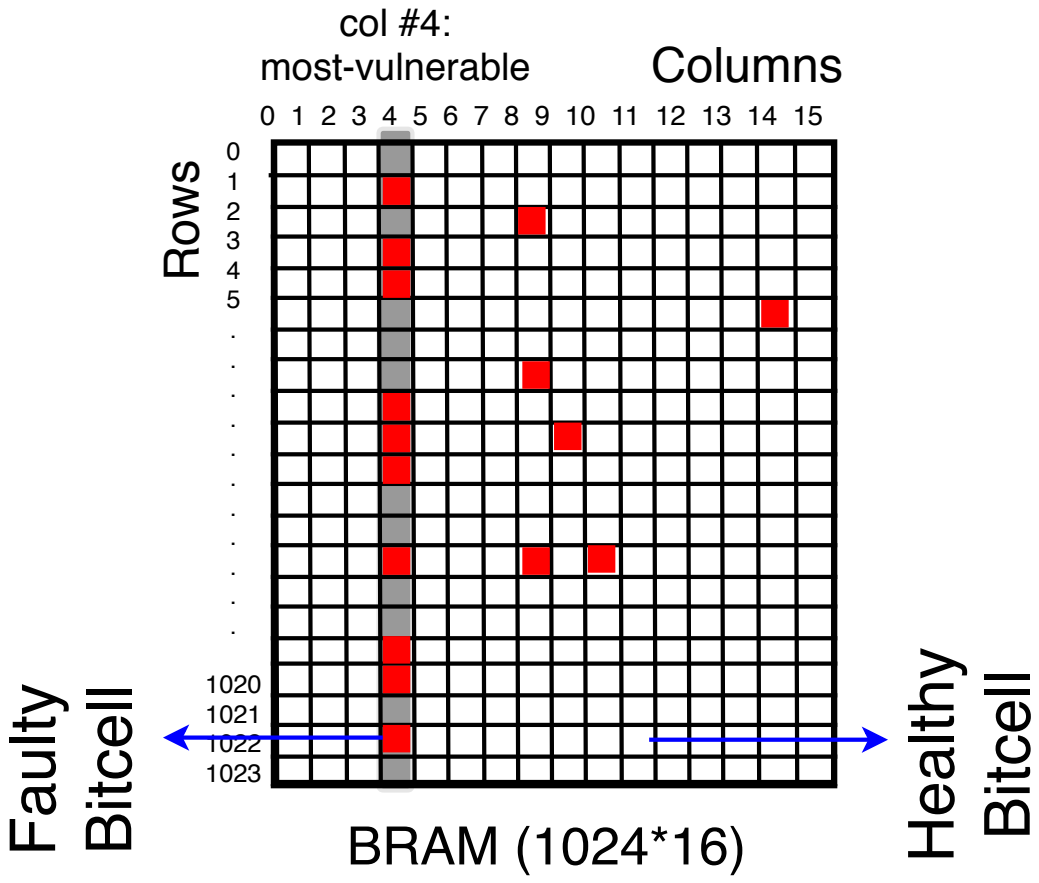
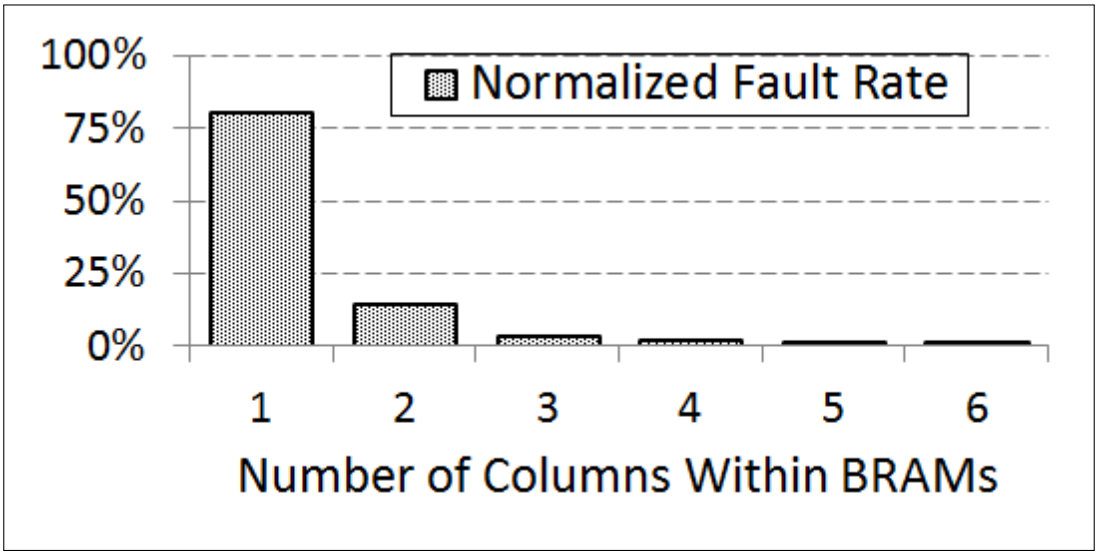


Figure 3.3: Illustration of column-wise fault distraction within BRAMs.

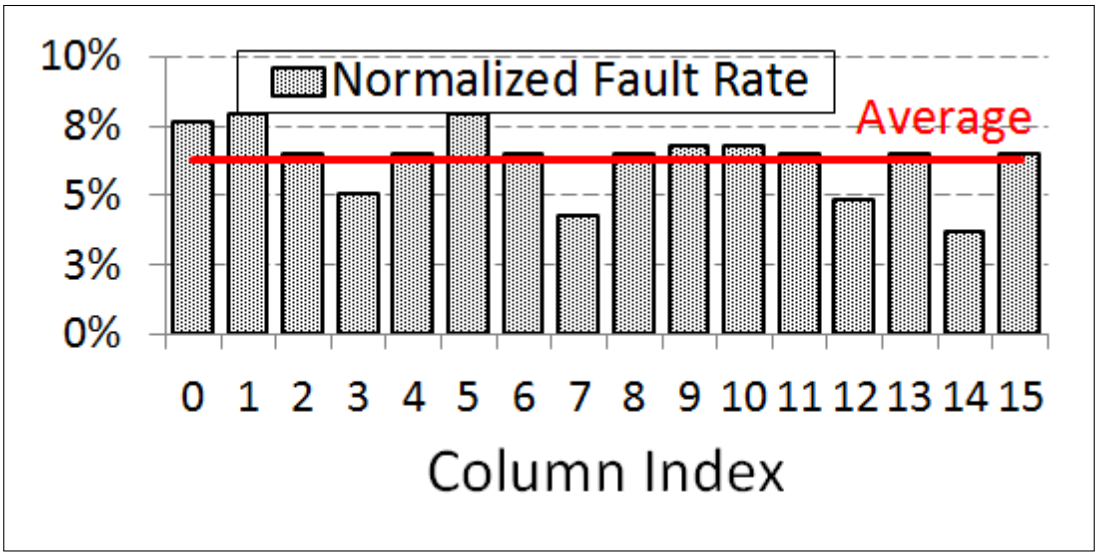
shown in Figure 3.5, within our BRAMs with 1024 rows, the minimum distance of more than 90% of faulty rows is on average of 20. A similar behavior is discussed for systematic process variation [145], [89]; thus, our conclusion is that the behavior observed about the undervolting faults is the direct consequence of the systematic process variation.

3.4 The Impact of the Die-to-Die Process Variation

We perform a further analysis of understanding the effects of voltage scaling on two samples of the same platform, *i.e.* KC705-A and KC705-B, which can show



(a) Column vulnerability within individual BRAMs.



(b) Column vulnerability among individual BRAMs.

Figure 3.4: Column-wise fault characterization within BRAMs.
* Shown for V_{CCBRAM} at $V_{crash} = 0.54V$ for VC707.

the impact of the die-to-die process variation. As earlier noted, KC705-A shows a significantly higher fault rate. Furthermore, with extracting their FVMs, we observe a significant difference in the fault map among BRAMs, see Figure 3.6, thanks to

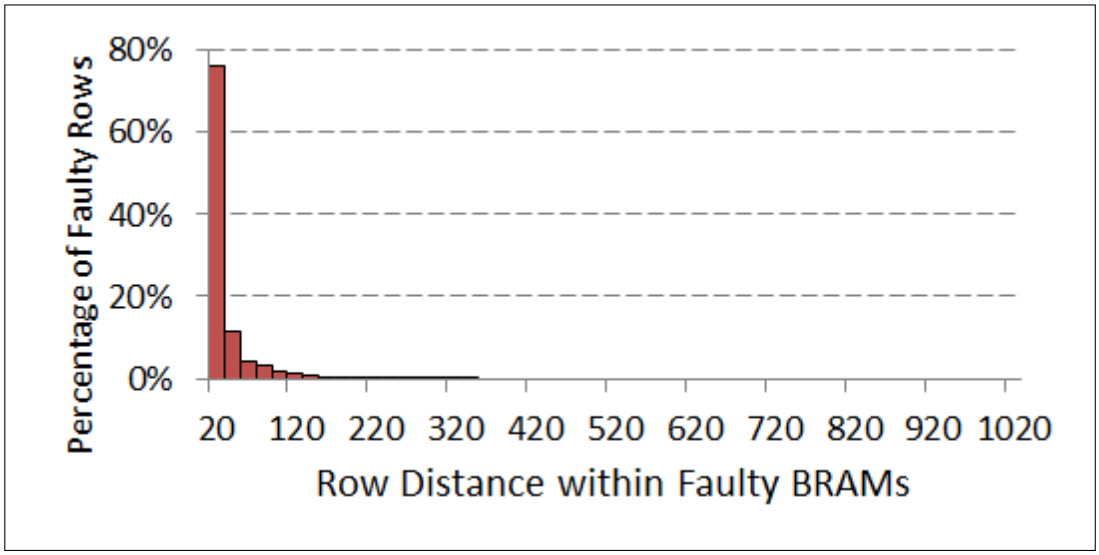
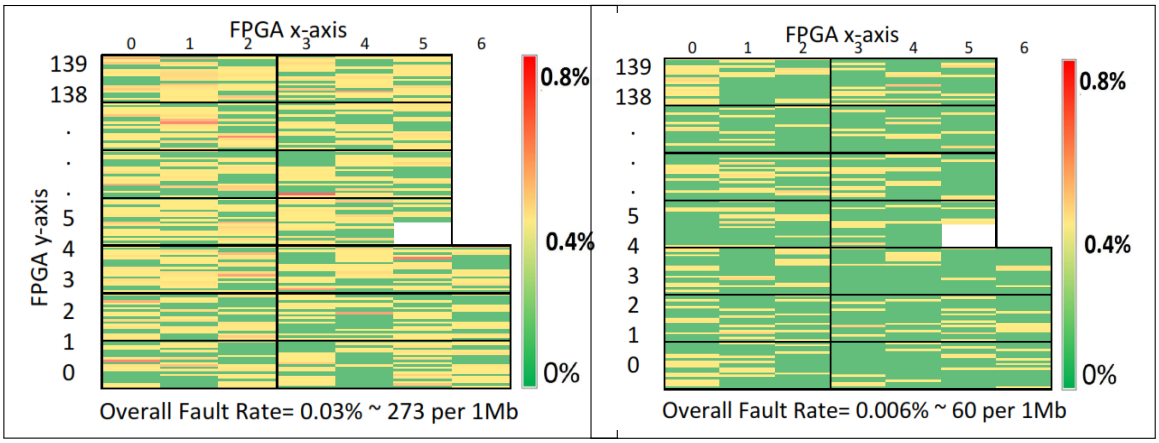


Figure 3.5: Row-wise fault characterization within BRAMs.
* Shown for V_{CCBRAM} at $V_{crash} = 0.54V$ for VC707.



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

Figure 3.6: FVM for two identical samples of KC705 at V_{crash}. Different fault rates and fault locations (FVM) are experimentally observed.

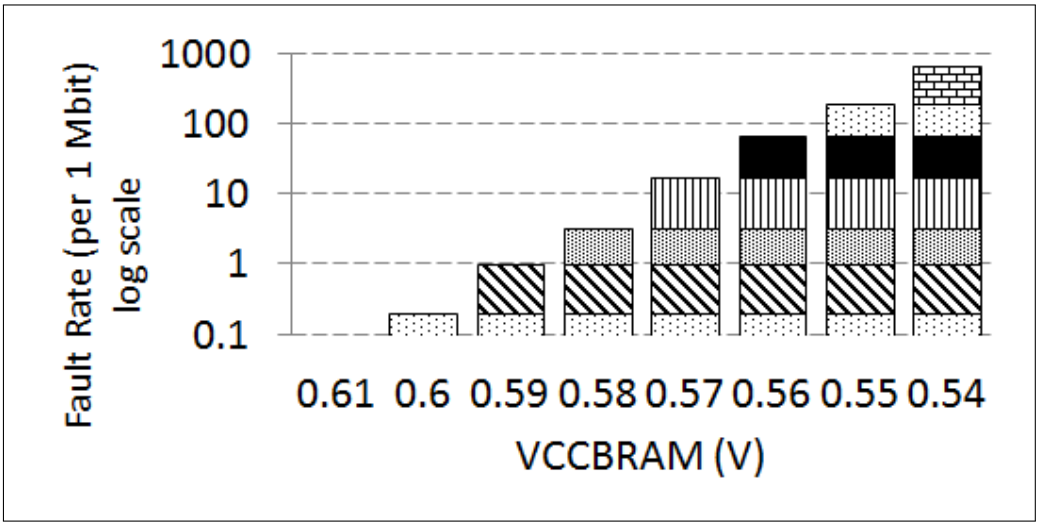
the die-to-die process variation. For instance, BRAM#(116,1) has high-vulnerability in KC705-A; however, it has low-vulnerability in KC705-B.

3.5 Fault Inclusion Property (FIP)

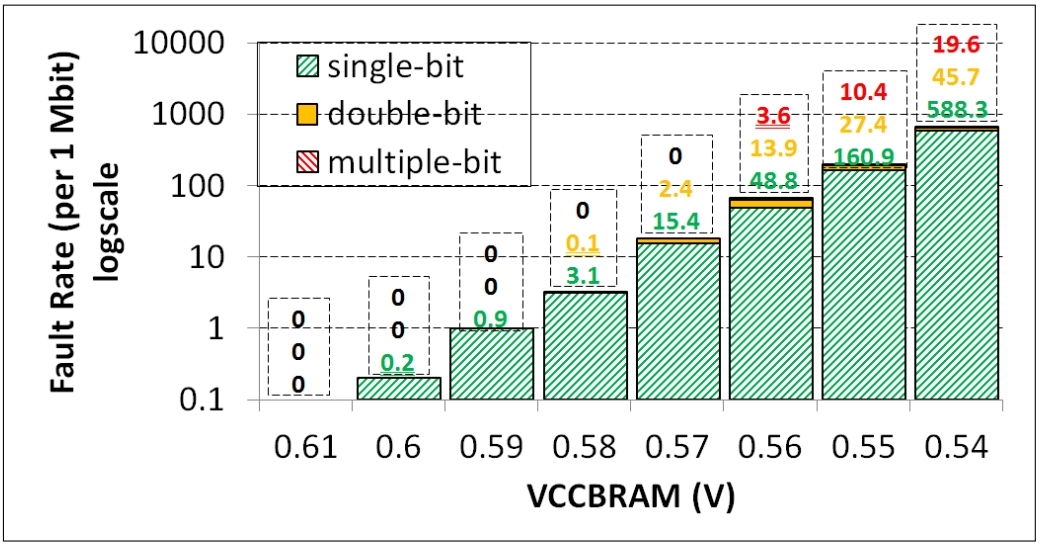
Fault Inclusion Property (FIP) is a property that we experimentally observed by monitoring the fault location and rate under various supply voltages below $V_{min} = 0.61V$. FIP is said to exist if all the faulty bits in a certain level of V_{CCBRAM} are still faulty in further reduced levels of voltage. FIP was previously observed for CPU cache structures [58], here we confirm that FIP holds for FPGA on-chip memories as well, as visualized in Figure 3.7a for VC707 (verified for other platforms, as well). While it may not be the best representation as the figure shows the stacked fault rates; however, through our experimental results we observed both location as well as rate of faults in a certain voltage level are exactly repeated in lower voltage levels. Also, FIP in the BRAM-level can be seen in the FVM of Figure 3.2.

3.6 Type of Faults: Single-, Double-, Or Multiple-Bit?

We categorize faults into or single-bit, double-bit, and multiple-bit faults. Figure 3.7b shows a histogram of these fault types, in different voltage levels at the critical voltage region, *i.e.*, from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$ on VC707. We observe that first, a vast majority of these faults are single- or multiple-bit faults; for instance, more than 90% and a further 7% at $V_{crash} = 0.54V$, respectively. Second, by further voltage underscaling, single-bit faults manifest before double-bit, and in turn, double-bit faults manifest before multiple-bit faults. The faults behavior mentioned above is the consequence of the FIP. In other words, within a memory row which experiences faults, by further undervolting, those initial faulty bits are still faulty and also potentially expanded to other bits. Consequently, single-bit faults can be potentially converted to double-bit and similarly, double-bit faults can be potentially converted to multiple-bit faults.



(a) Fault Inclusion Property (FIP).

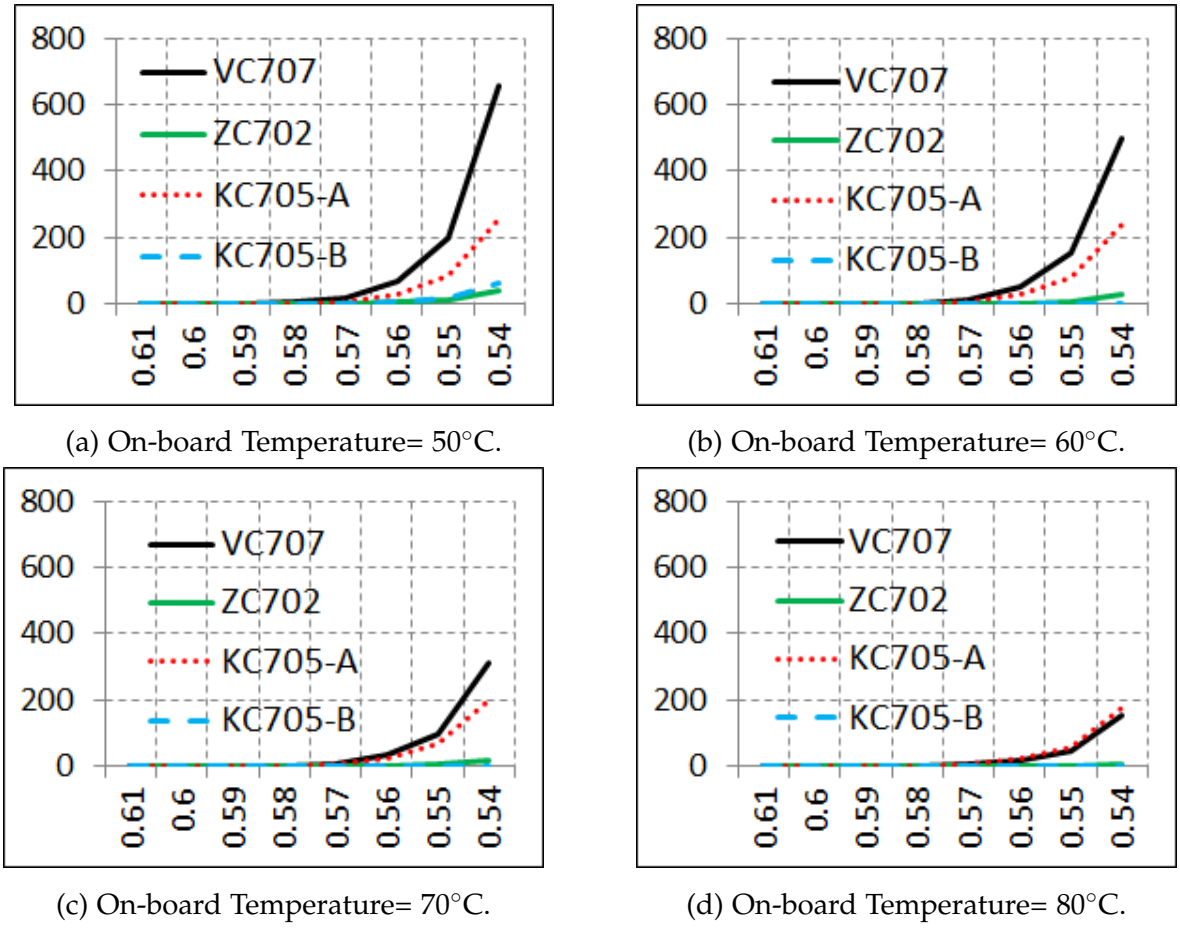


(b) Single-, Double-, or Multiple-bit faults.

Figure 3.7: Further analysis of faults location, undervolting BRAMs from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$, shown for VC707.

3.7 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,



(a) On-board Temperature= 50°C.

(b) On-board Temperature= 60°C.

(c) On-board Temperature= 70°C.

(d) On-board Temperature= 80°C.

Figure 3.8: The correlation among on-board temperature, supply voltage of BRAMs, technology, and fault rate.
* x-axis: V_{CCBRAM} from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$.
** y-axis: the fault rate per 1Mbit.

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 Figure 3.8 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; for instance, by more than 3X in VC707, when the temperature is increased from 50°C to 80°C. This observation is the consequence of the Inverse Thermal Independence (ITD)

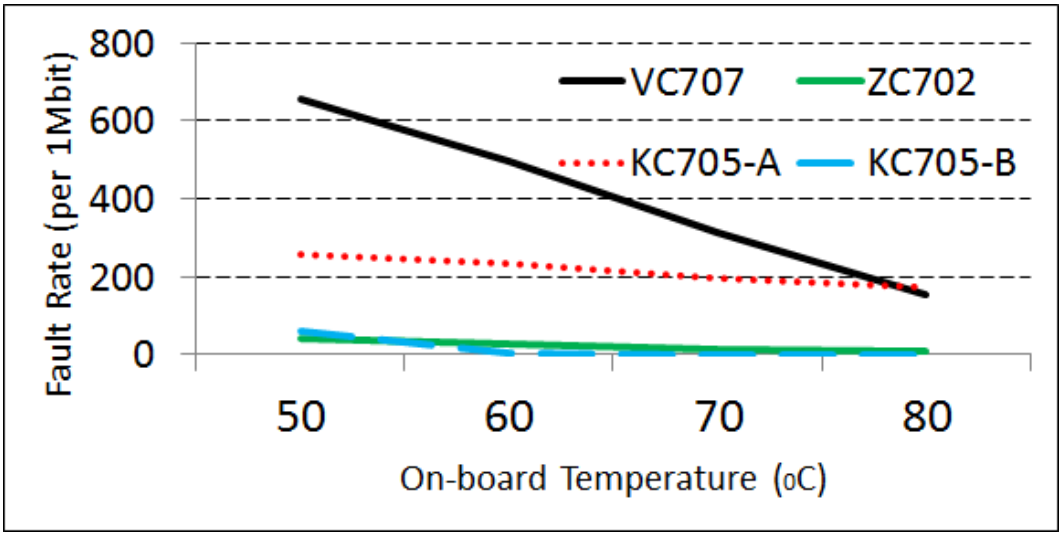


Figure 3.9: Different fault rate changes of the studied FPGA platforms over different temperatures at $V_{CCBRAM} = V_{crash}$.

property [117]. 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

Table 3.2: Summary of fault characterization in FPGA-Based BRAMs and comparing with the modern DRAMs, i.e,. DDR-3.

BRAM [our work]	**DRAM [36]**
Large voltage margin.	Large voltage margin.
Fault type: stuck-at-0	Fault type: No Info!
Pattern-dependent Faults	Pattern-free Faults
Exponential Fault Rate up to ∼0.1%	Exponential Fault Rate up to ∼20%
Significant Variation among BRAMs	Significant Variation among DRAM Banks
Follows Fault Inclusion Property (FIP)	No Info!
Inverse Relation of Temperature and Fault Rate	Direct Relation of Temperature and Fault Rate

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 in Figure 3.9, 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.

3.8 Summary

We presented comprehensive fault characterization results under BRAM undervolting below V_{min}. Our experimental observations such as stuck-at-0 behavior of faults and significant fault rate variability among BRAMs, can provide an opportunity to optimize power-reliability trade-offs in aggressively low-voltage regimes, for applications implemented onto FPGAs. We summarize our observations and findings in Table 3.2. Also, we compare our observations with a recent characterization work on DDR-3 [36], mostly in the behavioral-level. Although, there is a technological difference between them, *i.e.*, BRAMs are SRAM-based while DDR-3 are DRAM-based, the comparison highlights their significant similar fault behavior under low-voltage operations; although, there are some differences, as well. For instance, the effect of the environmental temperature and also the type of faults (mostly '1'-to-'0' bit flips for BRAMs versus more uniform for DDR-3) are the main differences, which can be due to the architectural difference among two memories.

4 Evaluating FPGA-based NN Accelerator on Low-Voltage FPGA BRAMs

In this chapter, we present and discuss the results of our study on the impact of the BRAM voltage scaling below nominal level, $V_{nom} = 1V$, in a typical FPGA-based NN accelerator. More specifically, our study includes the power consumption and NN accuracy trade-off, and investigation of two fault mitigation technique when NN is operating below V_{min}, *i.e.*, a proposed intelligent placement technique and also, built-in ECC. First, we briefly describe the NN resilience and the experimental methodology, and later on, discuss the efficiency of low-voltage FPGA-based NN.

4.1 Background on NN Resilience

Machine learning models and in particular NNs are increasingly being used in the context of nonlinear "cognitive" problems, such as natural language processing and computer vision. These models can learn from a dataset in the training phase and make predictions on a new, previously unseen data in the inference/prediction/classification phase with ever-increasing accuracy. However, the compute- and power-intensive nature of NNs prevents their effective deployment in resource-constrained environments, such as mobile scenarios [175]. Hardware acceleration,

e.g., FPGAs offers a roadmap for enabling NNs in these scenarios [152], [94], [135], [42], [16], [74]. However, similar to general purpose devices, hardware accelerators are also susceptible to faults (permanent/hard and transient/soft), as more specifically studied in this thesis, as the result of the aggressive voltage underscaling approach.

In recent years NN resilience is studied with different approaches, *e.g.*, software-level simulations or theoretical analyzes [156], [132], SPICE simulations [135], [93], [180], [181], and experimenting on the real hardware operating on low-voltage regimes, *e.g.*, SRAMs [165], [174], [173]. Among them, it is evident that software-level simulations and theoretical analyzes lack the information of the underlying hardware platform and are relatively less precise. In contrast, SPICE-based studies are more precise; however, these studies require significant circuit-level efforts.

Among the most relevant existing works on the NN resilience, Minerva [135] performs a characterization on the sparsity of data and analyzes the efficiency of leveraging fixed-point data representation model. In the same line, [157] studied the vulnerability of various layers of NN. Also, recently [93] studied the fault propagation in an ASIC model of NN focused on the vulnerability of different NN layers. This thesis approaches the resilience study on real faults that are generated through aggressive undervolting.

4.1.1 The Architecture of the NN Accelerator

Specifications of the experimented RTL NN with a baseline configuration is summarized in Table 4.1. Our study features a typical fully-connected NN that is also widely used in the structure of other NN models [135]. Our study targets the inference phase of NN since training is normally a one-time process; additionally, the inference is repeatedly performed to classify unknown data. As can be seen in Figure 4.1(a), this NN model is composed of input, hidden, and output layers,

where all adjacent layers are fully connected to each other. The first/last layer is the input/output layer and has one neuron for each component in the input/output vector. Between the input and output layers, there are single/multiple hidden layers. The interconnection between neurons of adjacent layers is determined based on a collection of weights and biases, whose values are tuned in the training phase. Each NN neuron uses an activation function to determine its output. Finally, in the output layer, a softmax function generates the final output of the NN. We perform our experiments on a 6-layer NN, *i.e.*, ($\{L_i, i \in [0,5]\}$), one input, four hidden, and one output layer(s). The four hidden layer sizes are fixed at 1024, 512, 256, 128 while input and output layer sizes are benchmark-dependent (784, 54 and 2437 for input while 10, 8 and 52 for output layers for the three NN applications studied in this thesis, *i.e.*, MNIST [87], Forest [24], and Reuters [25], respectively.). Thus, there are five matrix multipliers among adjacent layers, *i.e.*, ($\{Layer_j, j \in [0,4]\}$), where $Layer_j$ refers to the matrix multiplication of L_j and L_{j+1}. Among benchmarks, MNIST is a set of black and white digitized handwritten digits, each image composed of 784*8-bit pixels, the output infers the number from 0 to 9 (10 output classes), with 60000 training- and 10000 inference images. Forest includes cartographic observations for classifying the forest cover type. Reuters covers news articles for text categorization. MNIST is most widely-used by the ML community to evaluate the efficiency of novel NN methods. Hence, we use MNIST as the main benchmark to evaluate our resilience studies. To demonstrate the generality of experimental observations, we briefly present results for Forest and Reuters, as well.

For experiments, we first export weights and biases of the trained NN that is performed off-line using a MATLAB implementation, initialize BRAMs of FPGA, and then start streaming 10000 input images to perform the inference. Also, for representing data, we use the fixed-point low-precision model. Note that lowering

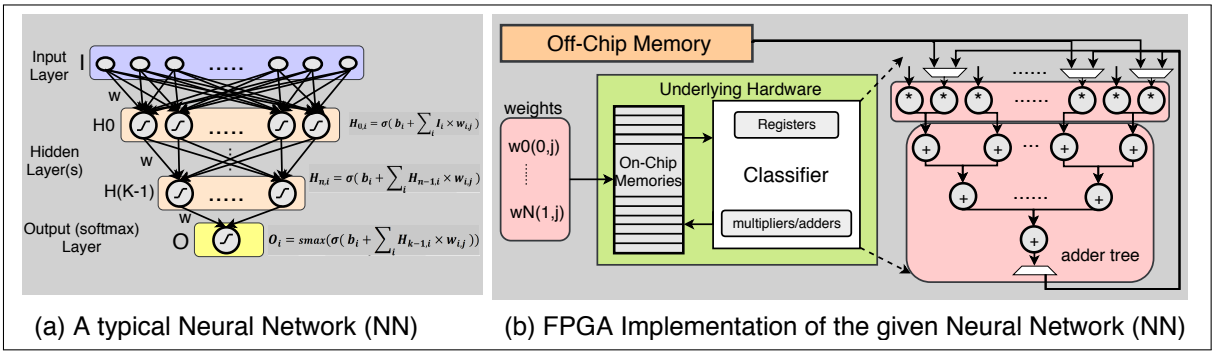


Figure 4.1: The overall methodology to resilience study of the RTL NN Accelerator.

the precision of data is a common technique for applications in the approximate computing domain, in particular for NNs performing inference [60], to achieve power and performance efficiency with negligible accuracy loss. Following this approach, we use a per-layer minimum precision fixed-point model. The bit-width of data (input, weights, and intermediate) 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 fraction component fills the rest of the 16 bits. As we experimentally observed, this quantification does not lead to any considerable accuracy loss in comparison to a full-precision data model. The minimum precision is used in this thesis is summarized in Figure 4.2.

4.2 Experimental Methodology of NN Evaluations

We perform our studies on the NN that is that is shown in Figure 4.1(b) and summarized in Table 4.1. In our system architecture, weights of the NN are located inside BRAMs and input images are being streamed through the off-chip DDR-3. The required calculation of the image classification, matrix multiplication plus sigmoid function activation, are performed in parallel by leveraging DSPs and LUTs

Table 4.1: Detailed specifications of the baseline RTL NN setup.

Neural Network (NN)	
Type	Fully-Connected
Phase	Inference
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
Original Activation Function	Logarithmic Sigmoid (logsig)
Original Benchmark	
Name	MNIST [87]
Type	Handwritten Digits (Images)
Number of Images	Training: 60000, Inference: 10000
Number of Pixels per Image	28*28= 784
Number of Output Classes	10
Additional Benchmarks	
1. Forest (cartographics of forest types)	[24]
2. Reuters (articles for text categorization)	[25]
Data Representation Model	
Type	16-bits Fixed-Point (Figure 4.2)
Sign-bit Precision	Minimum per layer (1 or 0 bit)
Digit-bit(s) Precision	Minimum per layer
Fraction-bit(s) Precision	16- (number of sign- and digit-bits)
An Example Synthesize Results	
FPGA Platform-Chip	VC707-Virtex7
Maximum 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%
Number of *PE*s	64

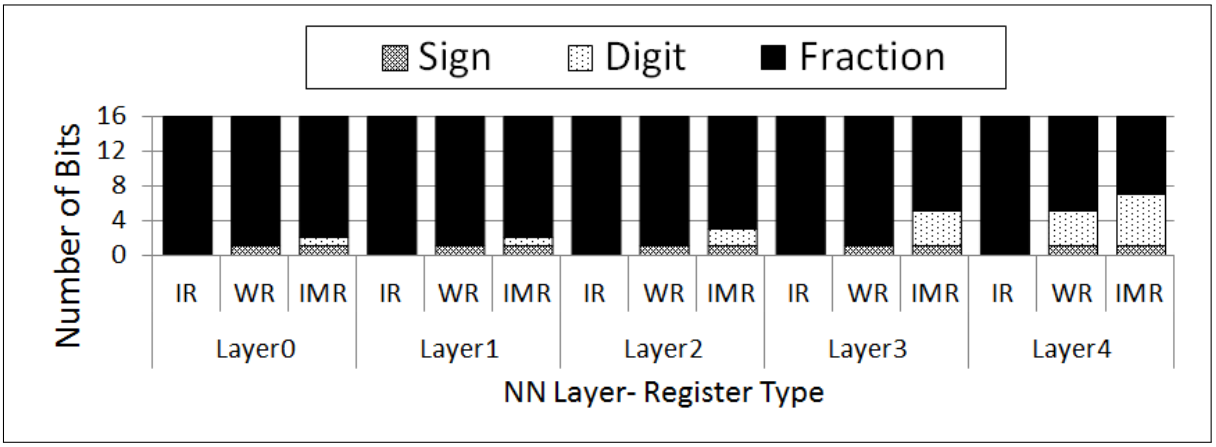


Figure 4.2: Minimum precision to represent data of RTL NN, *i.e.*, Inputs (IRs), Weighs (WRs), and Intermediate (IMRs).

of the FPGA and results are streamed to the host computer to perform the final step of the NN accuracy analysis. This setup is typical for most of the FPGA-based NN accelerator, as surveyed in [59]. In this section, we present results for the VC707 platform since a very similar efficiency is observed for other platforms.

On this setup, the on-chip power breakdown at various V_{CCBRAM}s, *i.e.*, $V_{nom} = 1V$, $V_{min} = 0.61V$, and $V_{crash} = 0.54V$, is shown in Figure 4.3. As can be seen, more than an order of magnitude BRAM power dissipation is reduced 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 lowering to $V_{crash} = 0.54V$, reduces 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.

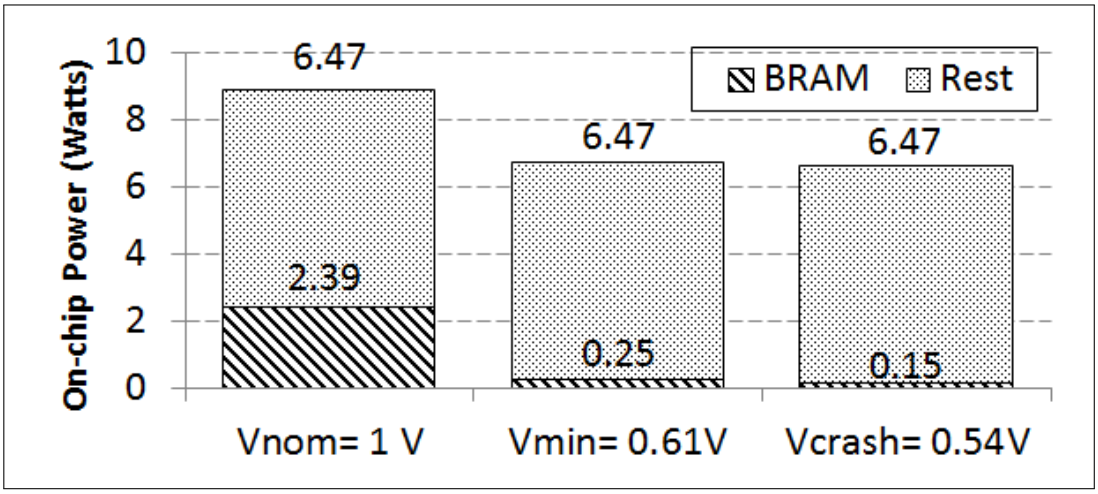


Figure 4.3: On-chip power breakdown of our FPGA-based NN at V_{nom}, V_{min}, and V_{crash} (VC707). Rest includes on-chip power consumption of DSPs, LUTs, routing resource, etc.

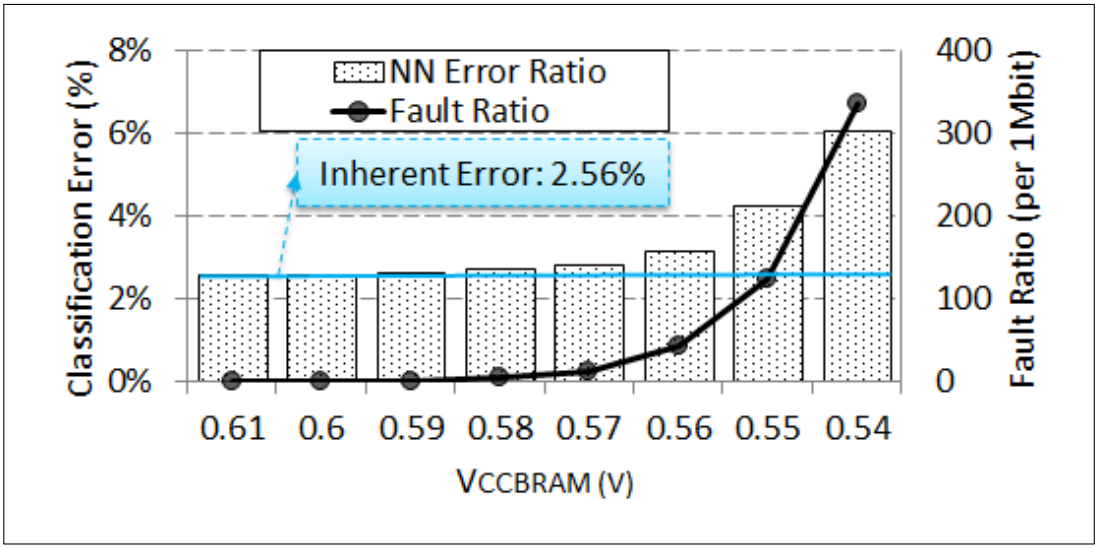


Figure 4.4: Impact of BRAM voltage scaling in the NN classification error, lowering V_{CCBRAM} from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$.

4.3 Impact of Voltage Scaling Below V_{min} on the NN Accuracy

When V_{CCBRAM} is underscaled in the critical region between $V_{min} = 0.61V$ and $V_{crash} = 0.54V$, faults occurring in some of BRAMs bitcells degrade the NN accuracy. In fact, 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 Figure 4.4. 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. The reason is that weights are sparse, our statistical analysis indicates that 76.3% of the bits having the logic value '0'. These bits have a negligible probability to be flipped, especially considering that most of the timing faults in the critical low voltage operation are stuck-at-0. This experimentally verified failure characteristic is the reason that MNIST application on our NN is inherently fault-tolerant against faults in extremely low-voltage operations on FPGA-based BRAMs. Through statistical experimentation, we confirm this data sparsity for other NN benchmarks such as Forest [24] and Reuters [25]. Also, other state-of-the-art has also confirmed the sparsity of many other NN benchmarks [135], [112], and a wider range of other applications, as well [8]. It means these applications would be inherently fault-tolerant for the type of failures experienced in FPGA BRAM undervolting.

4.4 Fault Mitigation Techniques

To prevent NN accuracy loss under low-voltage operations, we evaluate two techniques, *i.e.*, a novel BRAM placement technique and built-in ECC.

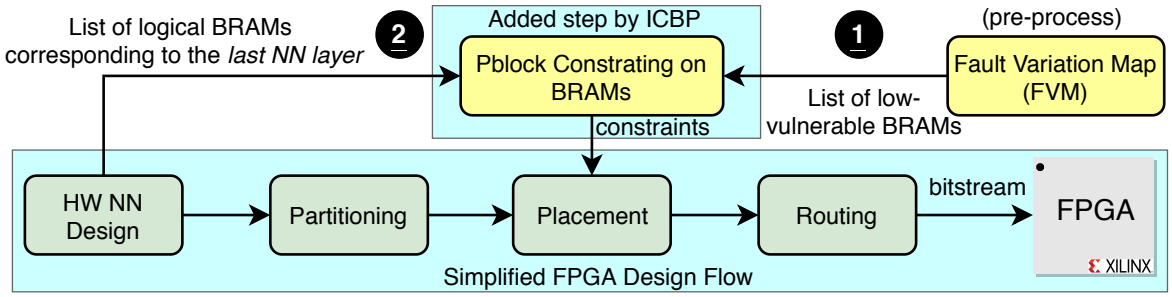


Figure 4.5: Methodology of Intelligently-Constrained BRAM Placement (ICBP).

4.4.1 Intelligently-Constrained BRAM Placement (ICBP)

The overall methodology of the proposed fault mitigation technique, Intelligently-Constrained BRAM Placement (ICBP), is shown in Figure 4.5. It can be used for low-voltage regions below $V_{min} = 0.61V$. The principal motivation is that low-voltage operations dramatically reduce power consumption (for our case, 40% in $V_{crash} = 0.54V$ over $V_{min} = 0.61V$); however, cause faults, which, in turn, leads to NN accuracy loss (for our case, 3.59% additional NN classification error in $V_{crash} = 0.54V$). The objective is to achieve this power-savings without significant impact on the NN classification error.

Elaborating ICBP

ICBP relies on two key observations:

- (1) As detailed in Section II, we observed that faults occur in reduced voltage BRAMs have deterministic and chip-dependent behavior with an entirely non-uniform distribution between different BRAMs that is exposed as FVM. As earlier mentioned, FVM extraction is a pre-processing stage.
- (2) We observed that various layers of the given NN have a 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 [142], [157], [93], 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 some random faults in weights of individual NN layers and let the NN accomplishes the classification. By monitoring the classification error of the faulty NN, we can evaluate the vulnerability of each NN level.

Due to these observations, ICPB introduces a simple yet effective BRAM placement algorithm that maps the weights of the inner NN layers to low-vulnerable BRAMs, targeting to mitigate faults and achieve power-savings with minimized NN classification accuracy loss. Note that FPGAs are uniquely suited to benefit from ICBP. In comparison, CPUÂt's have inherent disadvantages that make it challenging to apply aggressive undervolting ideas such as ICBP for their on-chip memories due to two reasons: *First*, it is extremely difficult, if not impossible to construct the undervolting fault map for CPU on-chip memories such as caches. *Second*, it is very cumbersome, if not impossible, to reconfigure and remap application data into cache regions that have a low vulnerability.

For further analysis, we present detailed statistical information of the different layers of the given NN, *i.e.*, the size (in terms of utilized number of BRAMs to locate weights of the corresponding NN layer), number of faults, and the normalized vulnerability of the individual layers, as shown in Figure 4.6. As can be seen, outer layers (closer to the input layer) are relatively larger, which experience more faults, as expected. Note that by statistical analysis, we observed that outer layers are relatively more sparse; thus, the per-layer fault rate is not precisely proportional to

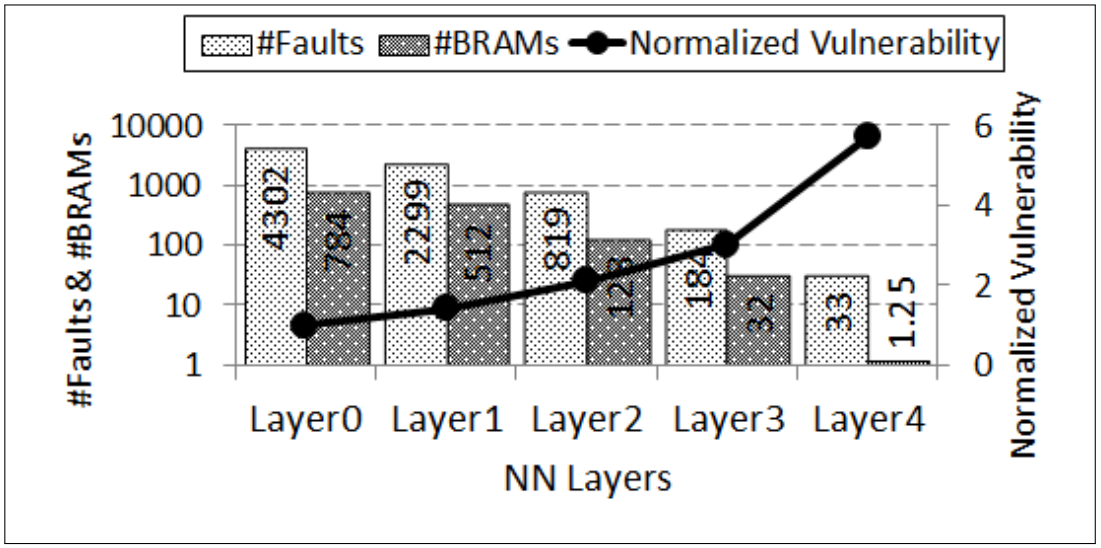


Figure 4.6: Statistical analysis of NN layers: size (#BRAMs), #Faults (at $V_{crash} = 0.54V$), and normalized vulnerability.

the per-layer size; however, both size (number of BRAMs) and the number of faults show an exponential behavior among various layers. Also, it is important to note that we conducted a software-based statistical fault injection campaign to extract the vulnerability of different NN layers. As can be seen, for instance, $Layer_5$ is around 6X more vulnerable than the first one, $Layer_0$, 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 $Layer_0$. The conclusion of analyzing different NN layers is that inner layers are significantly smaller and relatively less probable to experience faults; however, they are the most vulnerable layers. In other words, a fault in a inner NN layer has a more significant impact on the quality of the result; thus, they need better protection.

Implementation Methodology of ICBP

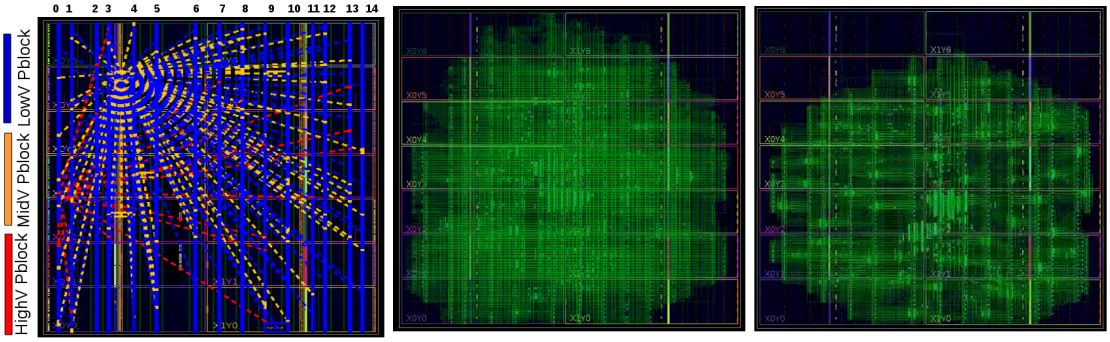
As earlier noted, in ICBP we aim to constrain the BRAM placement algorithm to map the logical BRAMs of inner NN layers to low-vulnerable physical BRAMs in

List 4.1: Illustrating Pblock Creation and BRAM Assignement for FPGAs.

```
1: create_pblock low-vulnerable_pblock
%Creation of a Pblock, low-vulnerable_pblock
2: resize_pblock [get_pblocks low-vulnerable_pblock] -add {RAMB18_X0Y44 RAMB18_X3Y91 RAMB18_X1Y26}
%Assigning 3 physical BRAMs to low-vulnerable_pblock
3: add_cells_to_pblock [get_pblocks low-vulnerable_pblock] [get_cells -quiet [list {l-BRAM[0]}{l-BRAM[1]}]]
%Assigning 2 logical BRAMs to low-vulnerable_pblock
```

the chip. Note that logical BRAMs are defined in the high-level Verilog design description and in contrast, physical BRAMs refers to the BRAM locations in the FPGA. Toward this goal, we exploit the Physical Blocks (Pblocks) facility of Vivado, a Xilinx implementation tool. Pblocks provides a fully flexible facility to constrain logical blocks, *e.g.*, BRAMs, to a physical region in the FPGA. As described in Section 3.2, we classify physical BRAMs into low-, mid-, and high-vulnerable classes. Having the list of physical locations of these BRAMs (XY), we first, create corresponding low-, mid-, and high-vulnerable Pblocks and then, appropriately assign the logical BRAMs into these Pblocks.

The example in List 4.1 illustrates the creation and BRAM assignment of Pblocks using TCL commands. This example creates a Pblock (low-vulnerable_pblock), assigns three physical BRAMs in locations (X0Y44, X3Y91, and X1Y26), and adds two logical BRAMs (l-BRAM[0] and l-BRAM[1]). This is a post-synthesize constraint that is added to the Xilinx Design Constraints (XDC) file. In consequence, the placement tool of Vivado will try to find the most efficient placement of these two logical BRAMs into the specified three physical BRAMs locations. By following this methodology, we create our three low-, mid-, and high-vulnerable Pblocks, each includes the corresponding physical BRAMs specified by our fault characterization.



(a) The schematic of Low-, Mid-, and High-vulnerable Pblocks.

(b) The Schematic of the Proposed ICBP in NN design.

(c) The schematic of the Default Placement in NN design.

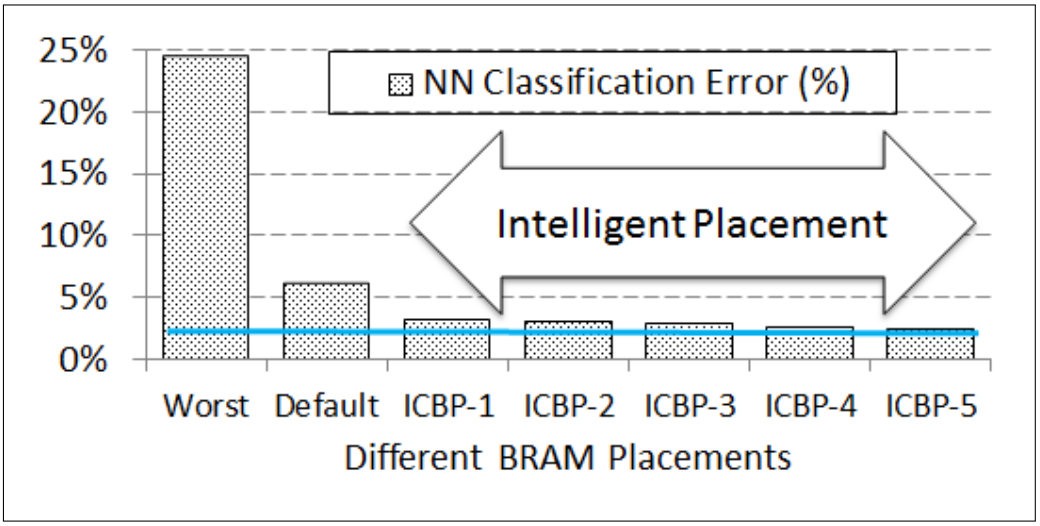
Figure 4.7: Pblocks and its Impact in the Schematic of the NN Design Placement in VC707 Platform.

Figure 4.7a shows a schematic of the created Pblocks in the VC707 platform. As expected, the low-vulnerable_pblock is dominating since it involves 88.6% of physical BRAMs. Also, the effect of ICBP in the final implementation of the design can be seen by comparing its schematic view in Figure 4.7b with the default placement (without any Pblock constraint) in Figure 4.7c. Note that Figure 4.7b refers to the design that all logical BRAMs of our design including all weights (not only the inner layers) are forced to be located in low-vulnerable BRAMs. As expected, the default placement results in more compacted design since the low-vulnerable BRAMs are distributed all over the chip, which limits ICBP to make a more compacted design. This specific design increases the timing slack by 50% over the default placement. However, as earlier noted, we aim to locate only inner layers of the NN into low-vulnerable BRAMs, which can eliminate this overhead, thanks to their significantly smaller sizes of inner layers. Later in this section, we discuss this trade-off and different aspects of ICBP.

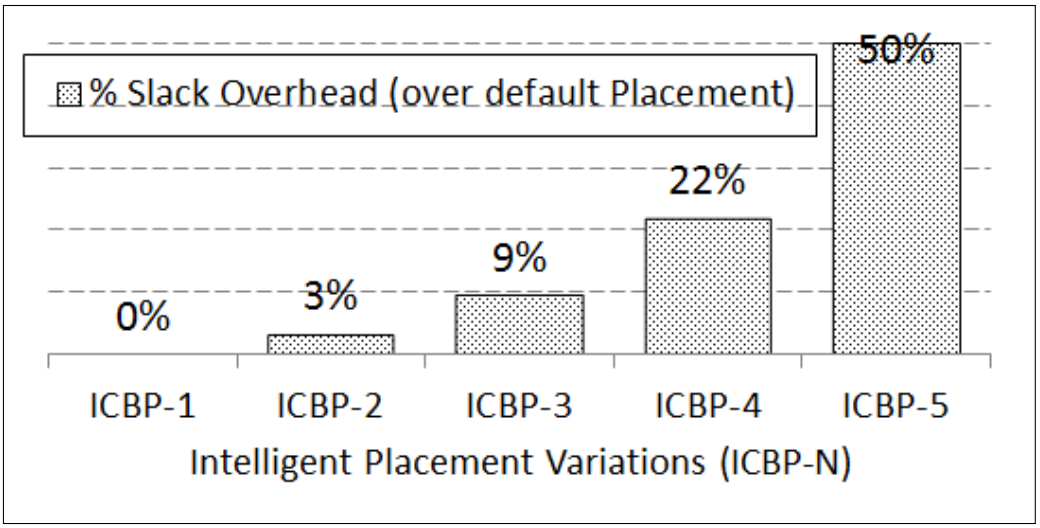
Experimental Results of ICBP

We evaluate several variations of the proposed mechanism, where "ICBP-N" refers to a version of ICBP that the last N layers of the NN are located in low-vulnerable BRAMs, $N \in \{1,2,3,4,5\}$. For instance, "ICBP-1" version means that weights of the "only last layer," *i.e.*, $Layer_4$, are forced to be located in low-vulnerable BRAMs, while a normal BRAM placement is applied to the other layers. In the same line, "ICBP-5" version means that all layers, *i.e.*, ($\{Layer_j, j \in [0,4]\}$) are located in low-vulnerable BRAMs, as its schematic placement is shown in Figure 4.7b. In Figure 4.8a, we compare the impact of different BRAM placement techniques in the NN classification error when $V_{CCBRAM} = V_{crash} = 0.54V$, *i.e.*, *i*) default BRAM placement, *ii*) different variations of the proposed BRAM placement technique, ICBP-N, $N \in \{1,2,3,4,5\}$, and *iii*) the worst-case placement where the inner NN layers are located in the high-vulnerable BRAMs, the rest in mid-, and low-vulnerable BRAMs in order. As can be seen, the classification error is decreased from 6.1% with the default placement to 3.01% in "ICBP-1" version, where only $Layer_4$ is forced to leverage low-vulnerable BRAMs. Note that $Layer_4$ is the smallest among the NN layers and most sensitive layer to faults. Thus, its protection significantly prevented the NN accuracy loss. By intelligently placing additional layers, the classification error is further decreased and reduces to 2.6%, which is very close to the inherent classification error of 2.56%.

However, since the low-vulnerable BRAMs are distributed all over the chip, ICBP may incur the timing overhead, as reported in Figure 4.8b in terms of the percentage of the timing slack increase over the default placement. As can be seen, the timing slack is significantly increased in the more aggressive versions and reaches to 50% in "ICBP-5", where all NN layers are located in low-vulnerable BRAMs. However, for "ICBP-1" this overhead is negligible since a very small number of BRAMs, two BRAMs, are forced to exploit the low-vulnerable BRAMs.



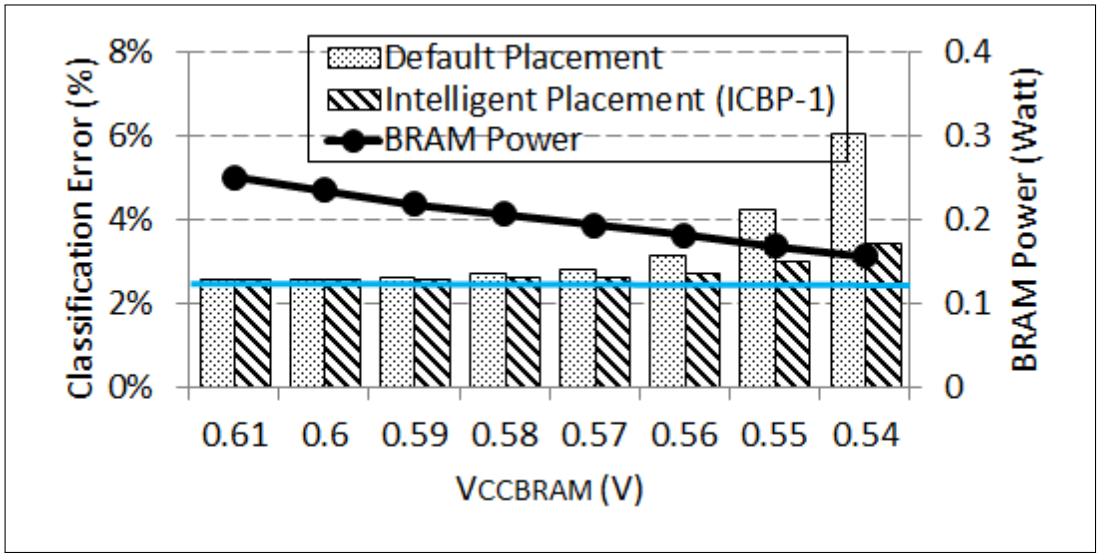
(a) Classification error of NN with different BRAM placements at $V_{crash} = 0.54V$.



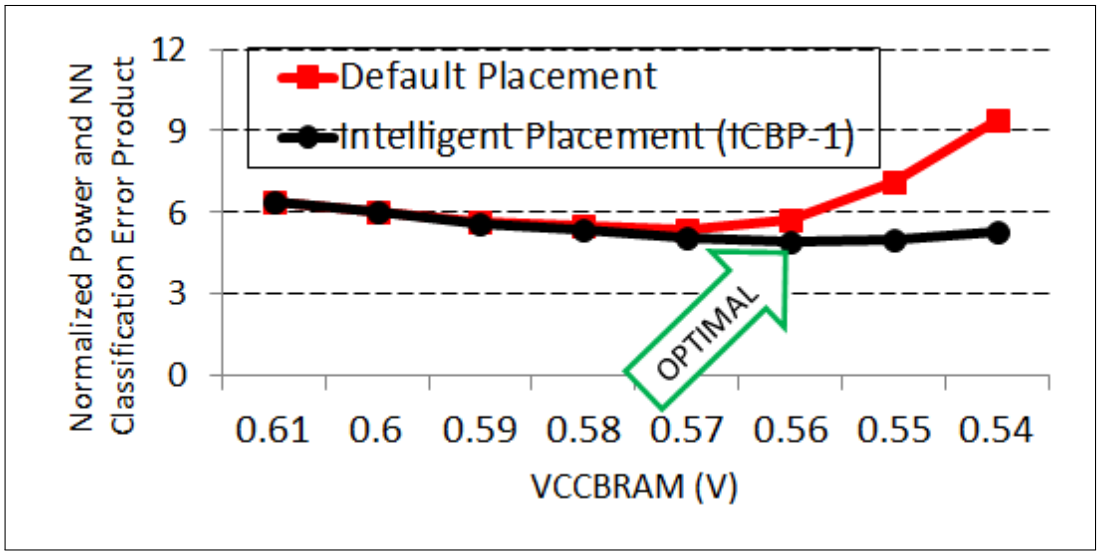
(b) Timing slack overhead of ICBP-N.

Figure 4.8: Evaluating our fault mitigation technique, *i.e.*, ICBP.

We repeat our most effective and timing cost-free version of the proposed mitigation technique, "ICBP-1", by lowering V_{CCBRAM} from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$, to explore the optimal voltage level in terms of power consumption and NN classification error trade-off, as shown in Figure 4.9. As can be seen in Figure 4.13a,



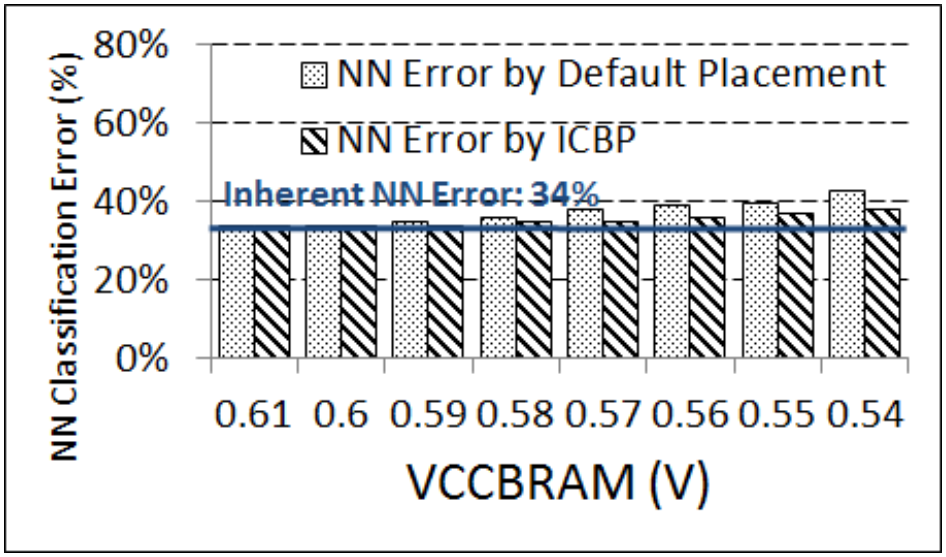
(a) Power reduction and NN classification error.



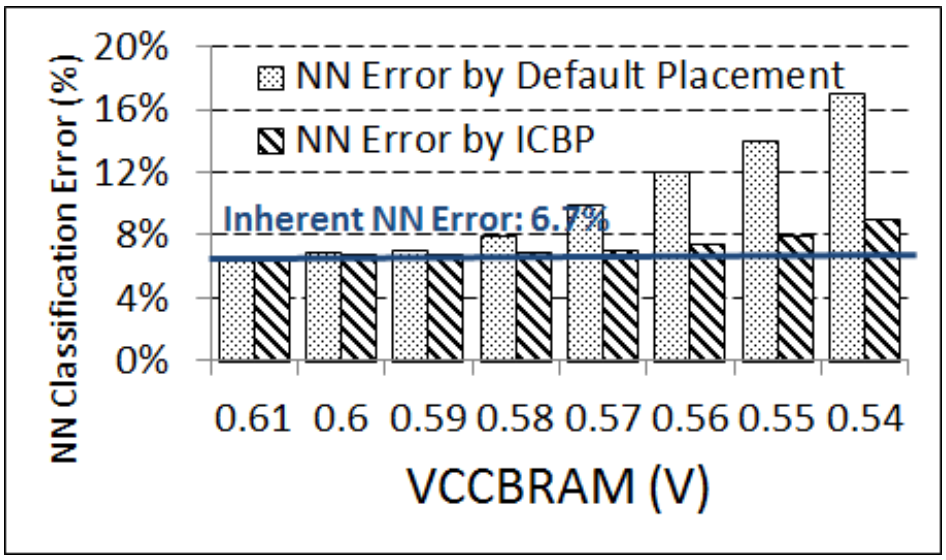
(b) The optimal V_{CCBRAM}, trading-off power and NN accuracy.

Figure 4.9: "ICBP-1" mitigation on various V_{CCBRAM}s in $[V_{min} = 0.61V, V_{crash} = 0.54V]$.

40% power savings is achieved in $V_{crash} = 0.54V$ over $V_{min} = 0.61V$, by 0.6% NN accuracy loss from the inherent fault-free classification error of 2.56%; however, the same amount of power in the default placement is dissipated by more than 3.59% NN accuracy loss.



(a) Forest [24].



(b) Reuters [25].

Figure 4.10: Efficiency of ICBP on FPGA-based NN accelerator for Forest and Reuters benchmarks on VC707.
* Different scales in y-axis.

Since V_{CCBRAM} scaling inversely impacts power consumption and NN classification error, we aim to find the optimal voltage level as the best trade-off. Toward

this goal, experimental results are analyzed in terms of a new metric, the normalized product of power consumption with NN classification error, as shown in Figure 4.13b. As can be seen, *first*, due to this trade-off $V_{CCBRAM} = 0.56V$ is the optimal voltage level for "ICBP-1", which leads to 28.1% of power saving achievements over the $V_{min} = 0.61V$, while the classification error increase up to 2.66% (0.1% overhead from the inherent fault-free classification error, 2.56%). *Second*, the efficiency of "ICBP-1" technique in comparison to the default placement technique, is relatively better manifested in lower levels of V_{CCBRAM} since in relatively higher voltage levels, the fault rate and its subsequent impact on the NN accuracy are not considerable.

Also, we repeat the similar methodology for Forest and Reuters benchmarks and as can be seen in Fig. 4.10a and Figure 4.10b, 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 impact the NN accuracy loss; however, mostly covered by ICBP.

4.4.2 Built-in ECC

BRAMs in Xilinx FPGAs are equipped with a built-in ECC with the capability of single-bit correction and double-bit detection (but not correction), *i.e.*, SECDED. Leveraging the built-in ECC has the advantage of mitigating faults without any major hardware or software modifications against other methods. For instance, Razor [50] that dynamically underscales the voltage until a fault occurs, leverages additional delay latches; and [36] presents a majorly-modified memory controller to deal with the reduced supply voltages. It is worth noting that as earlier mentioned, each row of BRAMs has two additional bits that can be either data or parity. In the previous section, we skip these two bits since experiments are performed on the

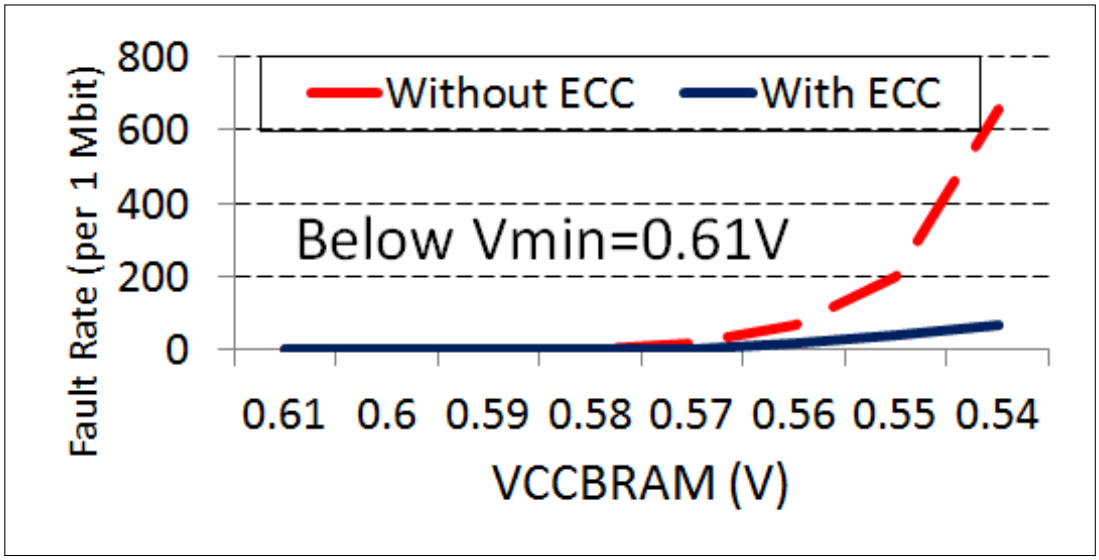


Figure 4.11: The efficiency of ECC to mitigate undervolting faults in the critical voltage regions below V_{min}

basic size BRAM without ECC capability. In this section, these bits are exploited as parity. Hence, the size of each BRAM is 1024 *18-bits.

Experimental Methodology of Built-in ECC

The built-in ECC mechanism of BRAMs uses Hamming code. When ECC is activated, parity bits are generated during each write operation and stored along with the data, at the granularity of a single row. These parity bits are used during each read operation of a row to correct single-bit faults, or to detect (but not correct) any double-bit fault, termed SECDED. In studied platforms, there are several options for BRAMs configurations. Our experimental setup is based on the following configurations:

- Configuration modes: We use simple dual-port mode BRAMs since it is the only mode that ECC can be activated.
- Soft- vs. hard-core ECC: Two types of ECC are available in Xilinx BRAMs, *i.e.*,

soft- and hard-core with the same functionality. Unlike the hard-core, in the soft-core ECC, FPGA resources such as LUTs are utilized to implement the corresponding functionality. Thus, we make our study on hard-core built-in ECC, which does not require any additional hardware.

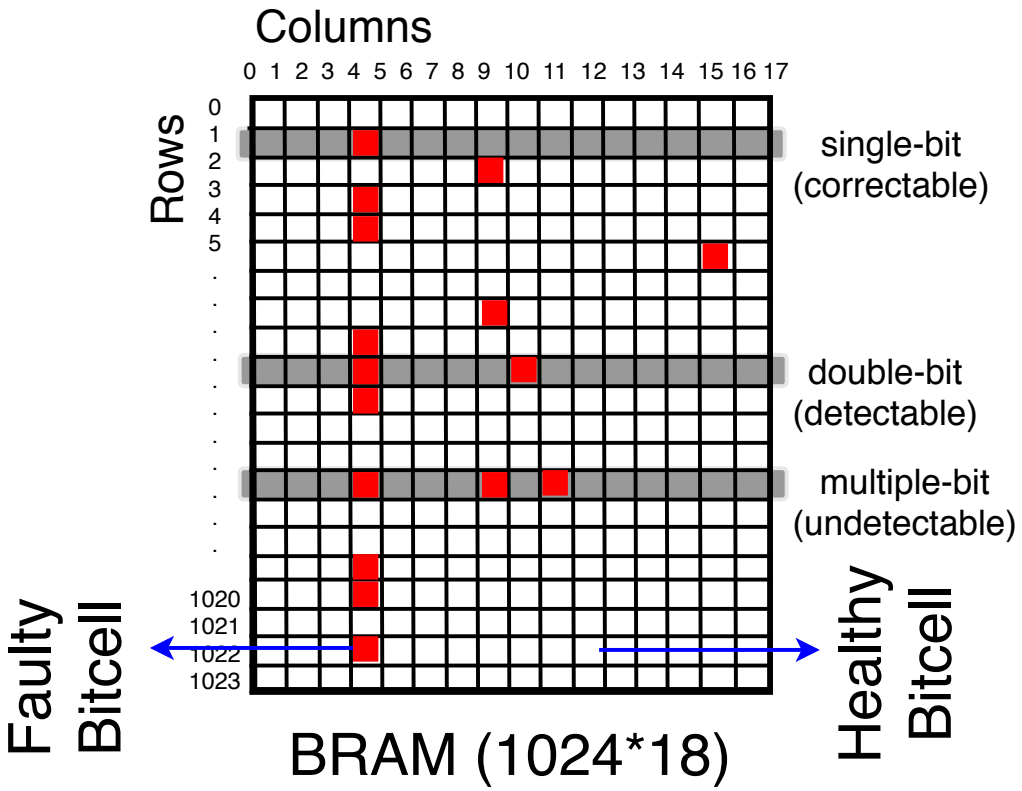
- Bit-width: Our design is based on memory bitwidth of 64-bits since the built-in ECC is optimized for memories with bitwidth $\geq$ 64-bits [9]. Note that since the basic BRAMs bitwidth is 18-bits, the memory used in our study is built by automatically cascading original BRAMs.

Efficiency of the Built-in ECC

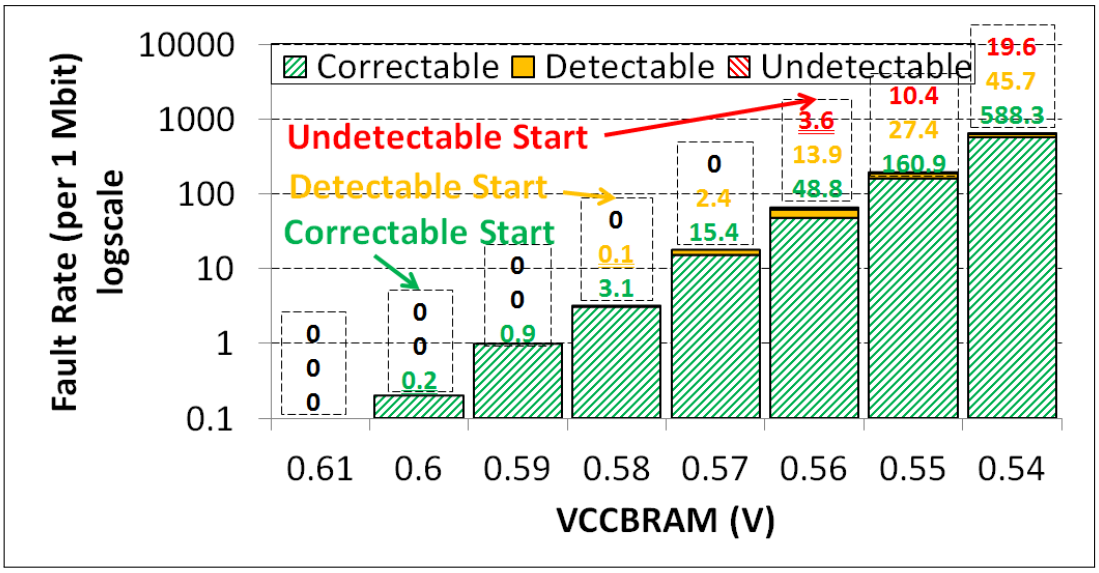
This section evaluates the efficiency and overhead of this ECC in aggressive low-voltage FPGA BRAMs, according to the behavior of undervolting faults. First, we experimentally observe that by leveraging ECC in BRAMs, the fault rate in the critical voltage region below V_{min} is significantly reduced by an average more than 90% for all platforms, as detailed for VC707 in Figure 4.11.

Due to the capability of the built-in ECC in FPGA BRAMs, we categorize faults into correctable (or single-bit), detectable (or double-bit), and undetectable (or multiple-bit) faults, as illustrated in Figure 4.12a. Figure 4.12b shows a histogram of these fault types, in different voltage levels at the critical voltage region, *i.e.*, from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$ on VC707. We observe that:

- The vast majority of these faults are correctable or detectable (but not correctable) by the built-in ECC; for instance, more than 90% and a further 7% at $V_{crash} = 0.54V$ are correctable and detectable, respectively, using the built-in ECC. This efficiency is the consequence of the inherent type of the built-in ECC, *i.e.*, SECDED, which we experimentally find that it has very good fault

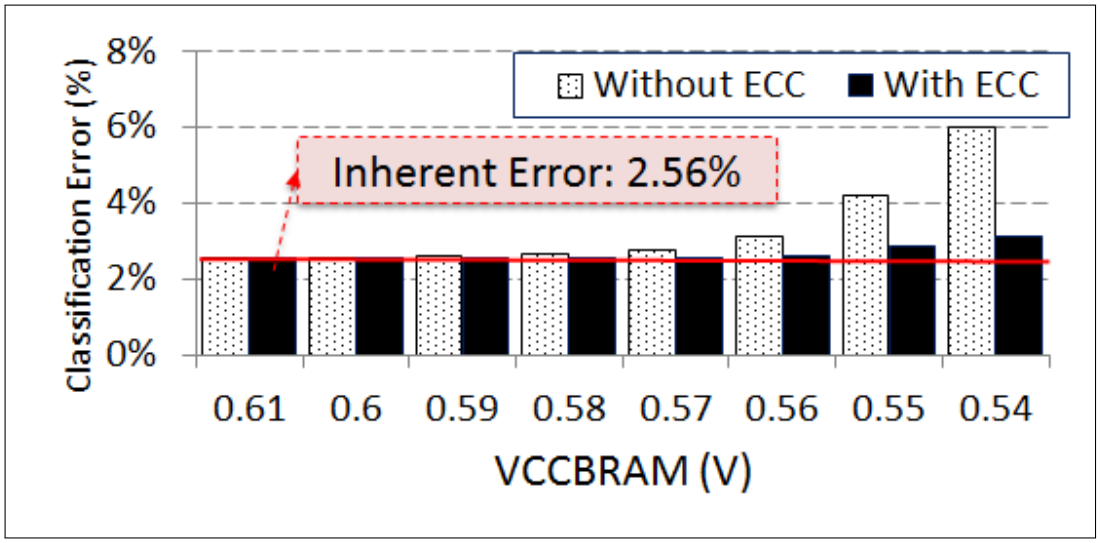


(a) Illustration (basic size BRAM with two parity bits)

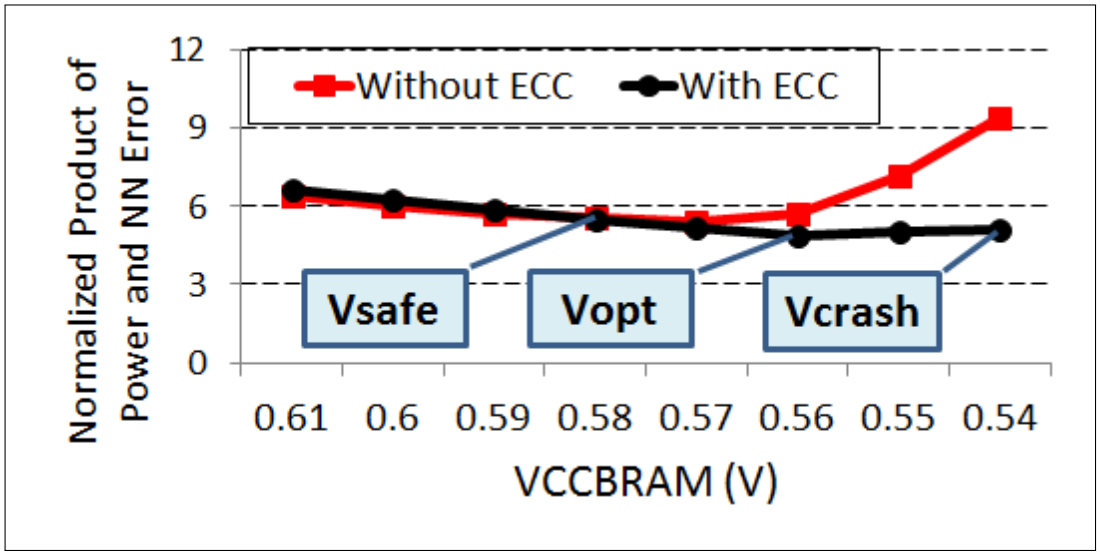


(b) Trading-off power and NN accuracy.

Figure 4.12: The behavior of ECC-activated BRAM faults in terms of fault types, when V_{CCBRAM} is scaled down from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$ for VC707.



(a) NN classification error.



(b) Trading-off power and NN accuracy.

Figure 4.13: ECC efficiency of undervolted BRAMs on FPGA-based NN accelerator.

coverage due to the relatively sparse distribution of undervolting faults, as detailed in Chapter 3.

- By further voltage underscaling, correctable faults manifest before detectable, and in turn, detectable faults manifest before undetectable faults. Through this observation, we leverage the built-in ECC to discover the minimum safe voltage of FPGA-based NN accelerator. The faults behavior mentioned above is the consequence of the FIP, as detailed in Chapter 3. It means that due to FIP, by further undervolting single-bit faults can be potentially converted to double-bit and similarly, double-bit faults can be potentially converted to multiple-bit faults.

The Overhead of the Built-in ECC

Table 4.2(a) includes the area utilization rate of the hardware design described in Section 2.1, in order to evaluate the area cost of the built-in ECC. Toward this goal, our hardware design accesses 512 memories each with the size of 1024 rows of 64-bits, which leads to a full BRAMs utilization on VC707. As can be seen, the built-in ECC does not incur considerable area cost since it is a hard-core unit and internally embedded within BRAMs structure. Also, Table 4.2(b) includes the power overhead of the built-in ECC. We report the power consumption of BRAMs at $V_{nom} = 1V, V_{min} = 0.61V$, and $V_{crash} = 0.54V$. As can be seen, the ECC power overhead is 13mW or 4.2% at $V_{crash} = 0.54V$. In other words, the power consumption of BRAMs are reduced from 0.31W to 0.211W (31.9% power reduction) with the voltage underscaling from $V_{min} = 0.61V$ to $V_{crash} = 0.54V$, by exploiting built-in ECC to cover a vast majority of faults.

Table 4.2: Power and area overheads of the built-in ECC.

a) Area Utilization (%)

	BRAM	LUT	FF
Without ECC	96%	3%	0.25%
With ECC	100%	12%	0.25%

b) BRAM Power (W)

	Vnom= 1V	Vmin= 0.61V	Vcrash= 0.54V
Without ECC	2.4	0.31	0.198
With ECC	—	—	0.211

—: Above V_{min}, since there is no fault, no need for the ECC.
Tested Memory Size: 512 * (1024 * 64-bits)

NN on ECC-enabled BRAMs

Motivated by above experiments on the efficiency of the built-in ECC, and to attain the power savings gain of the accelerator without compromising the NN accuracy, we leverage built-in ECC of FPGA BRAMs. In consequence, the NN classification error rate substantially reduces, thanks to the significant fault coverage by ECC, as shown in Figure 4.13a. For instance, the NN classification error has a 0.56% overhead, *i.e.*, the NN error of 2.56% as the inherent error rate increases to 3.12% at $V_{CCBRAM} = V_{crash} = 0.54V$, when BRAMs are equipped with built-in ECC. This overhead is 6.1X less than experiments on default BRAMs configuration without ECC, *i.e.*, 3.44% vs. 0.56%. Also, since voltage scaling inversely impacts power consumption and NN error, we analyze this trade-off at below $V_{min} = 0.61V$ voltages, by defining a new metric, *i.e.*, the normalized product of power consumption and NN classification error. As can be seen in Figure 4.13b, due to this trade-off, $V_{CCBRAM} = 0.56V$ is the optimal voltage level for ECC-activated case, which leads to 28.1% of power saving achievements over the $V_{min} = 0.61V$, while the classification error increases to 2.66% (0.1% overhead from the inherent fault-free

classification error, 2.56%).

For further analysis, the detailed power consumption and NN error rate results are summarized in Table 4.3. Below $V_{min} = 0.61V$, there are several voltage levels that are important in our analysis, *i.e.*, V_{safe} (the voltage that the first fault without correction possibility is detected by ECCs within BRAMs. There is no NN accuracy loss until this point.), V_{opt} (the voltage that leads to the least optimizing parameter on the product of the power and NN error rate), and V_{crash} (the lowest voltage that accelerator operates with the lowest power consumption). Through experiments, we measured $V_{safe} = 0.58V$, and as already discussed $V_{opt} = 0.56V$ and $V_{crash} = 0.54V$. By undervolting below $V_{min} = 0.61V$, as earlier explained, faults occur; however, correctable faults by the ECC appear earlier. Finally, at $V_{safe} = 0.58V$, the first not-correctable but detectable fault manifests. In other words, there is no NN accuracy loss until V_{safe} since all faults are corrected. Through this experimentally-confirmed property, the V_{safe} can be dynamically at the run-time determined, as is also in similar studied for modern processors [20]. In this thesis, we confirm its potential for FPGAs, as well. By further undervolting, the power consumption is more reduced; however, the NN accuracy is exponentially affected. In lower voltages, we reach the best power-accuracy trade-off at $V_{opt} = 0.56V$, as earlier discussed. Finally, further undervolting can be applied to achieve more power savings gain until the system crashes at $V_{crash} = 0.54V$; however, the NN accuracy loss is up to 0.57%.

4.4.3 Discussion on the Mitigation Techniques

We evaluated two fault mitigation technique, in which both rely on the behavior of undervolting faults that is extensively characterized in Chapter 3. In ICBP approach, we leverage the significant fault rate variability among BRAMs; also,

Table 4.3: Summary of trade-offs in NN accelerator with and without ECC capability at below V_{min} voltage level.

	$V_{min} = 0.61V$	$V_{safe} = 0.58V$	$V_{opt} = 0.56V$	$V_{crash} = 0.54V$
Fault Rate (per 1Mbit) (**Without**-ECC BRAMs)	0	3.24	66.33	653.73
Fault Rate (per 1Mbit) (**With**-ECC BRAMs)	0	0.1	4.5	64.23
NN Accuracy Loss (%)* (**With**-ECC BRAMs)	0%	0%	0.04%	0.57%
Power Saving (%)** (**With**-ECC BRAMs)	0%	17.5%	26.5%	37.7%

* Absolute distance from the inherent NN classification error, *i.e.*, 2.56%.

** Compared against the power consumption at $V_{min} = 0.61V$.

by leveraging built-in ECC the vast majority of faults that are single-bit are covered. Both techniques show good fault coverage without considerable power, performance, or area utilization overhead. However, ICBP requires a one-time pre-process phase to extract per-chip variation map, or FVM, which can be generated by the vendor, Original Equipment Manufacturer (OEM), or user. In contrast, leveraging built-in ECC does not need for this pre-process phase; however, it is optimized for relatively long bit-widths ($\geq 64 - bits$). This short cost-benefit analysis among evaluated mitigation techniques shows that the best technique can be traded-off; however, we think that researches in this area, *i.e.*, fault mitigation for undervolting FPGA faults, is not ended and more advanced mitigation techniques can be deployed according to the behavior of the faults extensively characterized in this thesis.

5 Related Work

In this section, first, we discuss the recent advances on the power and reliability of FPGAs; later on, we review the related work in the power and reliability trade-off of commuting devices and also memory systems. Then, we summarize the related work in different aspects of the voltage lowering technique in commercial devices. Finally, the recent related studies on the power efficiency and resilience of NNs are reviewed.

5.1 Power-efficient and Reliable FPGAs

To bridge the energy-efficiency gap between FPGAs and ASICs, several approaches have been studied. For instance, architecture-level optimizations are applied for Intel/Altera [3] and Xilinx [10] platforms, with the aim of more energy-efficient FPGAs. From the other approaches [41], [84], it can be mentioned to the low-power Electronic Design Automation (EDA) tools including synthesizing [40], placement [138], and routing [15]. In parallel, clock gating [126], voltage gating [95], and multi-V_{dd} [54] are other techniques that are extensively evaluated for FPGAs. This thesis has focused on the aggressive undervolting approach and evaluated its efficiency for FPGAs. Unlike the other approaches mentioned above, aggressive un-

dervolting does not need any modification on the architectures or tools.

Improvement of the reliability of FPGAs is also extensively studied in the academia and industry [150], [118]. For instance, Triple Modular Redundancy (TMR) technique is exploited for faults in FPGAs [102], and in parallel, the placement and routing tools are adapted to achieve more fault tolerant FPGA designs [116], [149]. Protecting on-chip SRAM against Single-Event Upsets (SEUs) is the other studied technique [167]. This thesis focuses on the undervolting faults and presents optimized techniques to mitigation them. Our approach to improving the reliability does not need any modification on the architectures or tools.

5.2 Power and Reliability of FPGAs versus CPUs, GPUs, and ASICs

FPGAs have collected the flexibility of CPUs/GPUs and the efficiency of ASICs. Thanks to this property, in recent years, FPGAs have brought significant attention to accelerate applications with a large size of data, *e.g.*, NNs. However, it has been shown that their energy efficiency is still the main concern in comparison to ASICs. This energy gap is experimentally studied and shown to be dependent on the architecture and applications; however, we summarize them ∼10X-∼20X by reviewing the related work. For instance, for an NN application, recent CPU, GPU, FPGA, and ASIC implementations are surveyed in [135]. As shown in this paper, ASIC-based accelerators are significantly low-power, up to 100X. As another example, in [122] and [123], by experimenting Binarized and Recurrent NNs on Aria 10 FPGA, 14-nm ASIC, software on Xeon CPU, and Nvidia Titan GPU, it has been experimentally shown that ASIC and FPGA are four and three orders of magnitude more energy-efficient than CPU, respectively. Also, ASIC is ∼11X more energy-efficient than FPGA.

5.3 Power and Reliability of FPGA BRAMs versus DRAMs and SRAMs

Since DRAMs and SRAMs (caches) are memory structures that have similar usage to BRAMs in FPGAs, techniques developed to balance power and reliability for these structures are orthogonal but relevant to the work developed in this thesis. Therefore, we briefly review the existing works as follows:

- DRAM: Main memory (DRAM) consumes as much as half of the total system power in a computer today, due to the increasing demand for memory capacity and bandwidth [56]. In addition to the relatively high power consumption of DRAMs, their reliability is also a main issue [36], [89]. To improve the power efficiency and reliability of DRAMs which trade-off accuracy for ease of design space exploration, there are many simulation-based research works [23], [136], [34]. Instead, a more accurate approach is to evaluate the power [56], [37] and reliability [36], [89], [35], [62], [64], [71], [75], [76], [77], [78], [79], [82], [88], [96], [104], [113], [114], [130] of real DRAM chips. Below we explain some of the most recent related work:

  – **Power:** [56] performed a comprehensive experimental characterization of the power consumed by modern real-world DRAM modules. Their extensive characterization of 50 DDR3L DRAM modules from three major vendors yields several key new observations about DRAM power consumption. They concluded that because state-of-the-art DRAM power models do not account for any of these key characteristics, they are highly inaccurate compared to the actual, measured power consumption of 50 real DDR3L modules. Based on their detailed analysis and characterization data, they developed the Variation-Aware model of Memory Power Informed by Real Experiments (VAMPIRE), which has the absolute percentage error of only 6.8%

compared to actual measured DRAM power. In the same line, [37] proposed a new DRAM energy reduction mechanism, called Voltron. The key idea of Voltron is to use a performance model to determine by how much they can reduce the supply voltage without introducing errors and without exceeding a user-specified threshold for performance loss. The evaluations showed that Voltron reduces the average DRAM and system energy consumption by 10.5% and 7.3%, respectively, while limiting the average system performance loss to only 1.8%, for a variety of memory-intensive quad-core workloads.

– Reliability: [36] comprehensively studied the effect of aggressive voltage underscaling on modern DRAM chips from various vendors. Similar to FPGA BRAMs in our work, they found that reducing the supply voltage below a certain point introduces bit faults in the data. According to their detailed characterization on the behavior of these faults, we observed significant similarities with undervolting faults on FPGA BRAMs such as significant fault variability and permanent behavior of faults. We also observed differences in the case of thermal behavior, which we attribute to different technology and architecture used in DRAMs versus FPGA BRAMs. They analyzed its impacts on the DRAM's access latency and reliability, by characterizing the behavior of faults and presenting effective mitigation techniques. As another recent study on the reliability of DRAMs, [89] empirically demonstrated a new form of variation that exists within a real DRAM chip, induced by the design and placement of different components in the DRAM chip: different regions in DRAM, based on their relative distances from the peripheral structures, require different minimum access latencies for reliable operation. In particular, they showed that in most real DRAM chips, cells closer to the peripheral structures could be accessed much faster than cells that are farther.

In this paper, the aforementioned phenomenon is experimentally characterized and present techniques to improve the overall system performance while ensuring reliable system operation.

- SRAM: SRAMs are architecturally different than DRAMs and mainly used as cache and low-latency registers. However, the power and reliability of these memories is also a hot research topic in recent years. For instance, [166] proposed several architectural techniques that enable microprocessor caches (L1 and L2), to operate at low voltages despite very high memory cell failure rates. During high voltage operation, the proposed schemes allow the use of the entire cache; however, during the low voltage operation, they sacrifice cache capacity by up to 50% to reduce the minimum safe voltage. In the same line, other works exists to study the power and reliability in CPU caches by different techniques [44], [44], [176], [55], [148], [48], [97]. SRAMs can also be used as on-chip memories. For instance, [174] and its later version [173] evaluated the effect of supply voltage scaling in SRAMs that they specifically fabricated. They reported that the supply voltage reduction of 310mV could save 2.9X of power consumption. Also, a similar study for SRAM cells is conducted in [52]. They observed that the ratio of faults in highly reduced voltage levels is increased exponentially.

In parallel to memory systems, there are many recent studies on the reliability and power optimizations of NAND Flash memories with the aim of improving the life time and reliability in presence of process variation and thermal stress, among others [99], [100], [29], [32], [31], [30], [98], [28], [27], [26]. Our thesis is an orthogonal approach to study the power and reliability of on-chip memories of real FPGAs through aggressive undervolting.

5.4 Aggressive Undervolting

Below, we summarize the related work in the voltage guardband, voltage underscaling below this guardband with (without faults occurring) and without (with faults occurring) frequency scaling on commercial devices.

5.4.1 Voltage Guardband

Most commercial devices are designed with a voltage guardband below the standard minimum nominal supply voltage to ensure the correct functionality in the worst case environmental and process variations. This voltage guardband is fully vendor- and system-dependent; for instance, it was measured to be 20% in modern GPUs [91] and 16% in modern DRAMs [36]. We experimentally determined the voltage guardband for Xilinx FPGA on-chip memories to be 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.

5.4.2 Simultaneous Voltage and Frequency Underscaling

Further voltage underscaling below the guardband gap, V_{min}, impacts the timing and increases the delay. In this regard, the simultaneous frequency lowering is a common approach to prevent timing violations, termed Dynamic Voltage and Frequency Scaling (DVFS). The DVFS mechanism guarantees that the design works as close to, but always above, the Critical Operating Point (COP), *i.e.*, the point where further underscaling frequency or voltage will result in observable faults [129].

DVFS is widely studied in different computing devices such as ASICs [144], [53], [101], FPGAs [119], [121], [169], GPUs [103], [47], [177], [111], [68], CPUs [81],

[164], [66], [33], heterogeneous systems [38], [128], [131], [169], as well as memory systems [45], [46], [115], [105]. For instance, a recent DVFS mechanism implemented on FPGAs, [120], showed 70% energy savings. However, the impediment of DVFS is the performance degradation as a result of the frequency lowering, which in practice, limits the efficiency and applicability of this approach. DVFS is not targeted in this thesis.

5.4.3 Aggressive Undervolting into the Critical Voltage 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, tolerating faults. Characterizing and mitigating these faults can allow better power and reliability trade-offs, without performance degradation, as is for the DVFS approach. This approach is studied in some real hardware as summarized as follows. This thesis studies the procedure mentioned above, for the first time in commercial FPGAs.

- *Modern Processors:* There are multiple studies on the voltage lowering below V_{min} in modern processors [21], [110], [108], [109], [106]. For instance, [72] revisited the microarchitecture of the processor design to be adaptable in beyond COP regions to minimize the voltage at which a soft architecture encounters the maximum allowable fault rate, and [153] presented a methodology for reliability-aware design space exploration. [127] extends aggressive undervolting to multi-core CPUs and [20] leveraged built-in ECC technique to detect and mitigate undervolting faults in Intel Itanium II.

- *GPUs*: Aggressive undervolting is also considered as a promising energy efficiency improvement technique in GPUs [158], [159]. As an example of commercial GPUs, [155] 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, [163] evaluated the Floating Point Units (FPUs) under timing violations and accordingly, presented a bit-level fault model.

Moreover, other components such as single CPU cores [57], Network On-Chips (NOCs) [18], [133], [137], caches [17], [107] as well as memory systems that is briefly surveyed in Section 5.3 are also studied by considering the aggressive undervolting effects. To detect and/or mitigate faults several general techniques are proposed in different domains such as TMR [168], Razor [50], [151], ECC using Hamming code [171], Hardware Transnational Memory (HTM) [170], 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. Unlike these relatively higher-overhead techniques, we performed our study to understand the behavior of faults under low-voltage operations comprehensively and accordingly, develop application-dependent efficient mitigation technique, more specifically experimented on FPGA-based NN accelerator. Finally, [50] proposed Razor as fault detection and mitigation technique that can be potentially exploited to deal with the timing faults in low voltage regions, as well. For instance, [92] studied the efficiency of Razor latches for FPGAs. It has been shown that the area overhead of this technique is significant. Instead of this high-overhead technique, we evaluate the built-in ECC of BRAMs to mitigate faults.

5.5 Recent Related Studies on NNs

NNs are inherently power-hungry applications, due to the computational, storage, and data movement required for the large matrices. Addressing this concern,

several application-level power-optimization techniques are proposed such as low-precision data representation model [60], node pruning [178], data compressing [61], among others. These techniques are customized for different underlying platforms such as CPUs, GPUs, FPGAs, and ASICs, as in detail surveyed in [154]. 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 [161], [132], [157], the voltage lowering can lead to significant power 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.

5.5.1 Simulation-Based Resilience Study of Low-voltage NNs

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 [161]. More specifically, aggressive 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 [135]. 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 [180]. In the same line, [179] presents an in-memory NN classifier in standard SRAM array and performs the subsequent fault study under low voltage operations; however, through Monte Carlo simulations. It is evident that this

approach lacks the exact information of the fault model under very low-voltage operations and their validation on the silicon remains a key question. Also, recently [80] studied the NN accelerator in the learning phase with low-voltage SRAM cells through circuit-level simulations. In [69], fault injection in the hardware neural network is studied by modeling the temperature and voltage variations.

5.5.2 Real Hardware-Based Resilience Study of Low-voltage NNs

There is little publicly-available work on the aggressive voltage scaling for NN applications on real-hardware; however, there are some efforts for ASICs and SRAMs, as summarized below:

– *ASICs:* There are several energy-efficient fabricated ASIC for NNs, *e.g.*, Google TPU [70], Eyeriss [42], YodaNN [16], and [165]. One of the contributions to achieving energy efficiency in these accelerators is the nominal voltage underscaling in comparison to the state-of-the-art. However, only [165] has briefly studied the behavior of NN below nominal level scaling beyond COP. They fabricated a 28nm System-On-Chip (SOC) with a programmable accelerator design for fully-connected NN, where a Razor circuit is used to detect timing faults in the datapath of aggressively reduced voltages. However, this paper targeted ASICs and did not propose a detailed fault characterization study on NNs.

– *SRAMs:* [174], [173] 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, its drawbacks are that *i*) without detailed bit-level characterization, *ii*) this study is on only input data (not weights), and *iii*) a software-level NN is used (the computations of NN are performed on MATLAB), which does not allow to apply any mitigation technique on the datapath of NN on the silicon. Also, tests are performed on specialized SRAM

cells, not on standard SRAM library cell, which makes it difficult to expand the results of this paper for real accelerators.

The key novelty of our FPGA-based NN accelerator is to experimentally study the effect of the aggressive undervolting to power and reliability of such a design. To the best of our knowledge, this thesis is the first experimental effort in this area.

6 Conclusion

We conclude the thesis, discuss findings, and look into windows opened by this thesis.

6.1 Summary and Conclusion

FPGAs are widely-used processing devices, thanks to their massively parallel architecture and for efficient streaming execution model. However, the power/energy efficiency of FPGAs is the main concern, especially when compared against ASICs. To effectively alleviate this issue, we evaluated aggressive undervolting, *i.e.*, supply voltage underscaling below the standard nominal level. Since the power consumption of digital circuits, *e.g.*, FPGAs is directly related to their supply voltage level, the aggressive undervolting approach exhibited significant potential to deliver energy saving of such devices. This thesis aims of this thesis was to study the potential of aggressive undervolting for commercial FPGAs experimentally.

Below the standard nominal voltage level, usually, large voltage guardbands are added by chip vendors to account for 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 without compromising to the performance or reliability. We experimentally evaluated this voltage guardbands for state-of-the-art commercial FPGAs. Our experiments included several representative platforms from Xilinx, a main vendor, *i.e.*, VC707 (performance-optimized architecture), ZC702 (FPGA with integrated ARM-core), and two identical samples of KC705 (power-optimized architecture) platforms. Through experimental analysis on these platforms, we found that underscaling the supply voltage until a certain level, *i.e.*, minimum safe voltage or V_{min}, does not introduce any observable fault. We experimentally confirmed the large voltage gap for all platforms that we study. For instance, for on-chip BRAMs the voltage guardband was measured to be on average 39% of the nominal level, which in turn, resulted in more than an order of magnitude of BRAMs power/energy.

However, we experimentally observed that further undervolting below the voltage guardband at V_{min}, caused reliability issues, as the result of the circuit delay increase. For this voltage region below V_{min}, we performed a detailed reliability study with preliminary focusing on BRAMs, since they play crucial roles in the FPGA-based designs and also, according to voltage distribution architecture of the studied platforms, the supply voltage of BRAMs are allowed to be independently regulated. We extensively characterized the bit-level behavior of these faults in terms of rate, location, type, the impact of the environmental conditions, etc. We exploited this detailed information for deploying fault mitigation techniques.

More specifically, we experimentally observed that by further undervolting below V_{min}, the fault rate exponentially increases; however, it varies for different studied platforms, which is the consequence of the process variation and architectural differences among them. Also, for all platforms that we study, we observed that *i)* faults are fully non-uniformly distributed over various BRAMs, *ii)* a vast majority of these faults are single-bit, *iii)* by on average 99.9% of these faults manifest

themselves as '1' to '0' bit flips, and *iv)* the location of these faults do not change over time that means to have a deterministic behavior at fixed voltage levels. Due to these behaviors, we generated a chip-dependent Fault Variation Map (FVM) that was leveraged in the further optimization of the FPGA-based accelerators. We also confirmed the Fault Inclusion Property (FIP), *i.e.*, a faulty bit at a certain voltage level will stay faulty at lower voltages as well, with a possibility to be extended for other bits of the corresponding row. As the result of the FIP phenomena, by further undervolting single-bit faults manifest before double-bit faults. This property can be efficiently considered in the design of fault mitigation techniques. Finally, we evaluated the effect of the environmental temperature on the reliability of aggressively low-voltage FPGA BRAMs. We observed that under aggressively low-voltage operations, higher temperature leads to the reduced fault rates, which confirmed the possibility of a lower V_{min} scaling at higher temperatures. This phenomenon is the consequence of the Inverse Temperature Independence (ITD).

Motivated by the above experimental results, we evaluated a typical FPGA-based NN accelerator, where BRAMs play a crucial role to achieve significant performance. We performed NN computations under aggressively low-voltage BRAMs operations and observed the potential of achieving substantial power saving gains; however, with the cost of NN accuracy loss below V_{min}. To attain power savings without NN accuracy loss, we evaluated two mitigation techniques. *First*, we developed an application-dependent BRAMs placement technique, *i.e.*, Intelligently-Constrained BRAM Placement (ICBP) that relies on the deterministic behavior of undervolting faults, and mitigates these faults by mapping the most reliability sensitive NN parameters to BRAM blocks that are relatively more resistant to undervolting faults. By adding intelligent constraints to the BRAMs placement, 28.1% BRAMs power saving gains were achieved over V_{min}, with 0.1% of NN accuracy loss and without any timing-slack overhead. *Second*, as a more general

technique, we applied the built-in ECC of BRAMs, and observed a significant fault coverage capability with a negligible power consumption overhead, thanks to its SECDED design that fits the most of faults under extremely low-voltage operations as we experimentally characterized. In consequence, the accuracy loss of the NN at low-voltage regions was prevented, while significant power reduction gain was achieved.

In consequence, further BRAM power is saved, by 28.1%, without considerable NN accuracy loss of 0.1%. Also, our experiments on the built-in ECC reveals its good fault coverage capability, thanks to its SECDED design, and by noting that a vast majority of undervolting faults are single-bit. The power consumption overhead of the built-in ECC is negligible.

6.2 Lessons Learned

This thesis is the first attempt to study the aggressive voltage underscaling for commercial FPGAs, on real devices. We experimentally found a significant potential of commercial FPGAs to safely operate below the nominal level, which in turn, leads to minimized power consumption. Although, further undervolting cause fault occurrence, through experimentally extensive characterization we discovered the possibility of correcting or at least detecting the vast majority of these faults. Hence, we practically showed that the trade-off between power consumption and reliability for commercial FPGAs could deliver significant energy savings gain.

We believe that as the first experimental study of this thesis on the FPGA aggressive undervolting and showing its significant potential for energy efficiency, it will substantially impact the future studies. The research scopes that can potentially be considered as the extension of this thesis are listed below:

- **Enterprise High-Performance Computing (HPC) Systems:** As earlier noted, FPGAs are going to be a major part of modern data centers, where, energy dissipation is a key concern. Hence, we believe that aggressive undervolting can be potentially applied to enterprise FPGA-based systems such as Microsoft Catapult [134] to improve their energy efficiency in comparison to ASIC-based systems such as IBM TrueNorth [63] and Google TPU [70].

- **Multiple FPGA vendors:** In this thesis, we concentrated on Xilinx FPGAs. However, there is another main FPGA vendor, Intel/Altera. This thesis does not include experiments on the Intel/Altera platforms; however, we briefly studied the voltage model of these platforms. We realize some differences that may impact the power and reliability trade-offs differently. For instance, Intel/Altera platforms are equipped with Smart Voltage ID (SmartVID) technology [3]. SmartVID enables the device to run at lower than default voltage while retaining the same performance level, reducing static and dynamic power. During manufacturing testing, Intel determines the optimum operating conditions for the FPGA performance. A set of voltage values corresponding to those conditions are then programmed into nonvolatile registers in the device. The contents of these registers and information about the silicon temperature control the output of the voltage regulators, minimizing power consumption. This property does not exist for Xilinx platforms. To experimentally explore the design considerations of different vendors, a promising approach is to extend the undervolting study to other vendors such as Intel-Altera, Lattice, Actel, and Quicklogic, and also to other grades such as industrial, military, and aerospace.

- **Energy-constrained Scenarios:** There are some environments, where energy-efficiency is the key metric, such as IoT and mobile applications. In this type

of environments, FPGAs can be very attractive since they inherently provide energy-efficiency and throughput, simultaneously. To achieve further energy-efficiency gains, we propose aggressive undervolting, since eliminating large voltage guardbands can deliver significant energy saving without compromising to the performance or reliability.

- **Approximate Computing:** Approximate computing is a computation technique which returns a possibly inaccurate result rather than a guaranteed accurate result. It can be used for applications where an approximate result is sufficient for its purpose such as Machine Learning, Fuzzy Systems, Signal Processing, among others. This type of applications can take advantage of aggressive undervolting even below the voltage guardband level, especially considering the possibility to mitigate the effect of undervolting faults in this voltage region. As a case study, we performed experiments on the Neural Network application; however, we expect the significant efficiency of the aggressive undervolting approach for other approximate application as well.

- **Stochastic Computing:** Stochastic computing (SC) is an unconventional method of computation that treats data as probabilities. Typically, each bit of an N-bit stochastic number (SN) X is randomly chosen to be 1 with some probability pX, and X is generated and processed by conventional logic circuits. SC has used in massively parallel systems and is very tolerant of soft errors. Its drawbacks include low accuracy, slow processing, and complex design needs. Its ability to efficiently perform tasks like communication decoding and neural network inference has rekindled interest in the field. Many challenges remain to be overcome, however, before SC becomes widespread. In other words, SC is specifically designed for the structures that are prone to bit flip faults which are the case of undervolting BRAMs in FPGAs. For example, the

coefficients of an NN may be saved in BRAMs using the format of stochastic computing. Indeed, using the following references of stochastic computing along with proposed the undervolting in FPGAs, it is possible to gain more benefits

- **Heterogeneous Computing**: With the rise of data size and the diversity of data and also due to the fundamental limitations of scaling at the atomic scale, heterogeneous systems have recently brought significant attention. There recently have been dramatically increased efforts toward heterogeneous computing, including integration of heterogeneous cores on a die (ARM), utilizing general-purpose GPUs (NVIDIA), combining CPUs and GPUs on the same die (Intel, AMD, ARM), leveraging FPGAs (Altera, Xilinx), integrating CPUs with FPGAs (Xilinx), and coupling FPGAs and CPUs in the same package (IBM-Altera, Intel-Altera). Heterogeneity aims to solve the problems associated with the end of Moore's Law by incorporating more specialized compute units in the system hardware and by utilizing the most efficient compute unit for each computation. As explained in Section 5, independent efforts exist for voltage scaling on individual computing devices; however, a holistic study on a heterogeneous system is missing. However, thanks to the potential shown in this thesis for FPGAs as well as the previous studies for other devices like GPUs, CPUs, and DRAMs, such a comprehensive study on heterogeneous systems holds promise and potentially can lead to considerable energy saving gains.

6.3 Future of Aggressive FPGAs Undervolting

We think that this thesis opens new windows for further researches in the FPGAs undervolting area. In this section, we list the major future potentials that our study

provides. These proposals can be adopted by FPGA vendors or by further academic studies to achieve better energy optimization:

- **Constraints on the voltage regulation on commercial FPGAs:** During the experiments we faced several constraints by vendors. For instance, there are many FPGA platforms from Xilinx without voltage scaling capability. For those platforms equipped with PMBus (or any similar standard), FPGA components are not fully isolated for the voltage scaling studies. We hope that by showing the potential of undervolting FPGAs in this thesis, vendors can be convinced to expose voltage margin options to users in the same manner that they exposed advanced overclocking capabilities (*i.e.*, turbo mode for CPUs) a couple of years ago.

- **Dynamic frequency scaling accompanied with voltage scaling:** During our experiments, we realized that FPGA BRAMs operate on the internal fixed clock frequency. Thus, we could not modify their operating frequency; although, unlike DVFS, undervolting delivers better energy savings gains since frequency is not scaled down. We hope that this thesis convinces vendors to provide more adjustable voltage and frequency setting of the FPGA components. Also, another constraint is that the voltage regulator is hardwired to the host. A more suitable design can be the possibility to regulate the supply voltage from itself.

- **Dynamic thermal management:** Through our experiments, we realize that at higher temperatures the reliability issues are relatively less. Thus, gradually and dynamically underscaling the supply voltage at higher temperatures can lead to even better results. This research direction needs an on-chip temperature sensor and an efficient voltage control unit to adjust the supply voltage. Note that unfortunately in most of the recent FPGA platforms including four

studied platforms, the voltage regulator is hardwired to the host (not to the FPGA). It means that voltage adjusting needs to be applied by software running at the host. This is another practical constraint that we observed, and the solution is to be able to regulate supply voltage from the FPGA logic, which can facilitate to implement dynamic voltage scaling techniques. We hope that this thesis can convince vendors to take this suggestion into account for their future board designs.

- **Analytically modeling and confirmation of the experimental studies:** We experimentally analyzed the power and reliability behavior of FPGAs as well as detailed fault characterization of BRAMs under aggressively low-voltage operations. However, to analytically modeling and in turn, confirming our experimental results, detailed circuit-level information of the internal FPGA architecture is required. Unfortunately, there is very limited publicly available information from the circuit-level structure of commercial FPGAs components, which, in turn, we did not find the chance to extend our work to analytically models, as well.

7 Publications

The work of the thesis has resulted in the following peer-reviewed publications.

7.1 Publications from the Thesis

- **Behzad Salami**, Osman S. Unsal, and Adrian Cristal Kestelman, "Comprehensive Evaluation of Supply Voltage Underscaling in FPGA on-chip Memories.", in *51st Annual IEEE/ACM International Symposium on Microarchitecture (**MICRO**)*, 2018.

- **Behzad Salami**, Osman S. Unsal, and Adrian Cristal Kestelman, "Fault Characterization Through FPGAs Undervolting.", in *28th International Conference on Field Programmable Logic & Applications (**FPL**)*, 2018.

- **Behzad Salami**, Osman S. Unsal, and Adrian Cristal Kestelman, "A Demo of FPGA Aggressive Voltage Downscaling: Power and Reliability Tradeoffs.", in *28th International Conference on Field Programmable Logic & Applications (**FPL**)*, 2018.

- **Behzad Salami**, Osman S. Unsal, and Adrian Cristal Kestelman, "UnderVolt_FNN: An Energy-Efficient and Fault-Resilient Low-Voltage FPGA-based

DNN Accelerator", in *51st Annual IEEE/ACM International Symposium on Microarchitecture (**MICRO**) ACM Student Research Competition (**SRC**)*, 2018.

7.2 Publications not Included in the Thesis

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

- **Behzad Salami**, Gorker Alp Malazgirt, Oriol Arcas-Abella, Arda Yurdakul, and Nehir Sonmez, "AxleDB: A novel programmable query processing platform on FPGA.", in *Elsevier Microprocessors and Microsystems - Embedded Hardware Design (**MICPRO**)*, vol. 51, pp. 142-164, 2017.

- Oriol Arcas-Abella, Adria Armejach, Timothy Hayes, Gorker Alp Malazgirt, Oscar Palomar, **Behzad Salami**, and Nehir Sonmez, "Hardware Acceleration for Query Processing: Leveraging FPGAs, CPUs, and Memory.", in *IEEE Computing in Science and Engineering (**CISE**)*, vol. 18(1), pp. 80-87, 2016.

- **Behzad Salami**, Oriol Arcas-Abella, Nehir Sonmez, Osman S. Unsal, and Adrian Cristal Kestelman, "Accelerating Hash-Based Query Processing Operations on FPGAs by a Hash Table Caching Technique.", in *Springer Communications in Computer and Information Science (**CCIS**), presented in Latin American Conference on High Performance Computing (**CARLA**)*, pp. 131-145, 2016.

- **Behzad Salami**, Oriol Arcas-Abella, and Nehir Sonmez, "HATCH: Hash Table Caching in Hardware for Efficient Relational Join on FPGA.", in *23th*

*IEEE International Symposium on Field-Programmable Custom Computing Machines (**FCCM**)*, pp. 163, 2015.

List of Figures

List of Tables

Bibliography

[1] (2015). Timothy Prickett Morgan, "Why Intel Might Buy FPGA Maker Altera.". https://www.nextplatform.com/2015/03/30/why-intel-might-buy-fpga-maker-altera/. 5, 87

[2] (2018). Intel, "FPGA Architecture- White Paper.". https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01003.pdf. 2, 87

[3] (2018). Intel, "Power Reduction Features in Intel Arria 10 Devices.". https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/an/an711.pdf. 5, 61, 77

[4] (2018). Power Management Bus (PMBUS). http://pmbus.org. 12

[5] (2018). Texas Instruments (TI), "CMOS Power Consumption.". http://www.ti.com/lit/an/scaa035b/scaa035b.pdf. 5

[6] (2018). Texas Instruments (TI), "Fusion Digital Power Designer". http://www.ti.com/tool/FUSION_DIGITAL_POWER_DESIGNER. 13

[7] (2018). Top500, M. Feldman, "Good Times for FPGA Enthusiasts.". https://www.top500.org/news/good-times-for-fpga-enthusiasts/. 1

[8] (2018). University of Florida, "Sparse Matrix Collection.". https://sparse.tamu.edu/. 42

[9] (2018). Xilinx, "7-Series Memory Resource.". https://www.xilinx.com/support/documentation/user_guides/ug473_7Series_Memory_Resources.pdf. 54

[10] (2018). Xilinx, "Power Analysis and Optimization.". https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_1/ug907-vivado-power-analysis-optimization.pdf. 3, 5, 17, 61, 87

[11] (2018). Xilinx, "VC707 Evaluation Board for the Virtex-7 FPGA: User Guide.". https://www.xilinx.com/support/documentation/boards_and_kits/vc707/ug885_VC707_Eval_Bd.pdf. 6, 13, 14, 15, 87

[12] (2018). Xilinx, "Xilinx Kintex-7 FPGA KC705 Evaluation Kit.". https://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html. 7

[13] (2018). Xilinx, "Xilinx Zynq-7000 SoC ZC702 Evaluation Kit.". https://www.xilinx.com/products/boards-and-kits/ek-z7-zc702-g.html. 7

[14] Alser, M., Hassan, H., Xin, H., Ergin, O., Mutlu, O. & Alkan, C. (2017). GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping. *Bioinformatics*, **33**, 3355–3363. 1, 7

[15] Anderson, J.H. & Najm, F.N. (2004). Low-power programmable routing circuitry for FPGAs. In *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, 602–609, IEEE Computer Society. 61

[16] Andri, R., Cavigelli, L., Rossi, D. & Benini, L. (2018). Yodann: An architecture for ultralow power binary-weight CNN acceleration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **37**, 48–60. 36, 70

[17] Ansari, A., Feng, S., Gupta, S. & Mahlke, S. (2009). Enabling ultra low voltage system operation by tolerating on-chip cache failures. In *Proceedings of the 2009 ACM/IEEE international symposium on Low power electronics and design*, 307–310, ACM. 68

[18] Ansari, A., Mishra, A., Xu, J. & Torrellas, J. (2014). Tangle: Route-oriented dynamic voltage minimization for variation-afflicted, energy-efficient on-chip networks. In *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, 440–451, IEEE. 68

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

[20] Bacha, A. & Teodorescu, R. (2013). Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors. In *ACM SIGARCH Computer Architecture News*, vol. 41, 297–307, ACM. 59, 67

[21] Bacha, A. & Teodorescu, R. (2014). Using ECC feedback to guide voltage speculation in low-voltage processors. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 306–318, IEEE Computer Society. 67

[22] Bertran, R., Bose, P., Brooks, D., Burns, J., Buyuktosunoglu, A., Chandramoorthy, N., Cheng, E., Cochet, M., Eldridge, S., Friedman, D. *et al.*

(2017). Very Low Voltage (VLV) Design. In *Computer Design (ICCD), 2017 IEEE International Conference on*, 601–604, IEEE. 4

[23] Binkert, N., Beckmann, B., Black, G., Reinhardt, S.K., Saidi, A., Basu, A., Hestness, J., Hower, D.R., Krishna, T., Sardashti, S. *et al.* (2011). GEM5: A multiple-ISA full system simulator with detailed memory model. *Computer Architecture News*, **39**. 63

[24] Blackard, J.A. (2000). Comparison of neural networks and discriminant analysis in predicting forest cover types. 37, 39, 42, 51

[25] Cachopo, A.M.d.J.C. (2007). Improving methods for single-label text categorization. *Instituto Superior Técnico, Portugal*. 37, 39, 42, 51

[26] Cai, Y., Haratsch, E.F., Mutlu, O. & Mai, K. (2012). Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe*, 521–526, EDA Consortium. 65

[27] Cai, Y., Haratsch, E.F., Mutlu, O. & Mai, K. (2013). Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling. In *Proceedings of the Conference on Design, Automation and Test in Europe*, 1285–1290, EDA Consortium. 65

[28] Cai, Y., Yalcin, G., Mutlu, O., Haratsch, E.F., Unsal, O., Cristal, A. & Mai, K. (2014). Neighbor-cell assisted error correction for MLC NAND flash memories. In *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, 491–504, ACM. 65

[29] Cai, Y., Luo, Y., Ghose, S. & Mutlu, O. (2015). Read disturb errors in MLC NAND flash memory: Characterization, mitigation, and recovery. In *Depend-*

able Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on, 438–449, IEEE. 65

[30] Cai, Y., Luo, Y., Haratsch, E.F., Mai, K. & Mutlu, O. (2015). Data retention in MLC NAND flash memory: Characterization, optimization, and recovery. In *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*, 551–563, IEEE. 65

[31] Cai, Y., Ghose, S., Luo, Y., Mai, K., Mutlu, O. & Haratsch, E.F. (2017). Vulnerabilities in MLC NAND flash memory programming: experimental analysis, exploits, and mitigation techniques. In *2017 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 49–60, IEEE. 65

[32] Cai, Y., Luo, Y., Haratsch, E.F., Mai, K., Ghose, S. & Mutlu, O. (2018). Experimental Characterization, Optimization, and Recovery of Data Retention Errors in MLC NAND Flash Memory. *arXiv preprint arXiv:1805.02819*. 65

[33] Calore, E., Gabbana, A., Schifano, S. & Tripiccione, R. (2018). Software and DVFS tuning for performance and energy-efficiency on Intel KNL processors. *Journal of Low Power Electronics and Applications*, **8**, 18. 67

[34] Chandrasekar, K., Akesson, B. & Goossens, K. (2011). Improved power modeling of DDR SDRAMs. In *Digital System Design (DSD), 2011 14th Euromicro Conference on*, 99–108, IEEE. 63

[35] Chandrasekar, K., Goossens, S., Weis, C., Koedam, M., Akesson, B., Wehn, N. & Goossens, K. (2014). Exploiting expendable process-margins in DRAMs for run-time performance optimization. In *Proceedings of the conference on Design, Automation & Test in Europe*, 173, European Design and Automation Association. 63

[36] Chang, K.K., Yaǎlikçi, A.G., Ghose, S., Agrawal, A., Chatterjee, N., Kashyap, A., Lee, D., O'Connor, M., Hassan, H. & Mutlu, O. (2017). Understanding reduced-voltage operation in modern DRAM devices: Experimental characterization, analysis, and mechanisms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, **1**, 10. 4, 33, 34, 52, 63, 64, 66

[37] Chang, K.K., Yaglikçi, A.G., Ghose, S., Agrawal, A., Chatterjee, N., Kashyap, A., Lee, D., O'Connor, M., Hassan, H. & Mutlu, O. (2018). Voltron: Understanding and Exploiting the Voltage-Latency-Reliability Trade-Offs in Modern DRAM Chips to Improve Energy Efficiency. *arXiv preprint arXiv:1805.03175*. 63, 64

[38] Chau, V., Chu, X., Liu, H. & Leung, Y.W. (2017). Energy Efficient Job Scheduling with DVFS for CPU-GPU Heterogeneous Systems. In *Proceedings of the Eighth International Conference on Future Energy Systems*, 1–11, ACM. 67

[39] Che, Z., Purushotham, S., Khemani, R. & Liu, Y. (2015). Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*. 7

[40] Chen, D., Cong, J. & Fan, Y. (2003). Low-power high-level synthesis for FPGA architectures. In *Proceedings of the 2003 international symposium on Low power electronics and design*, 134–139, ACM. 61

[41] Chen, D., Cong, J., Pan, P. *et al.* (2006). FPGA design automation: A survey. *Foundations and Trends® in Electronic Design Automation*, **1**, 195–330. 61

[42] Chen, Y.H., Krishna, T., Emer, J.S. & Sze, V. (2017). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, **52**, 127–138. 36, 70

[43] Cheon, Y., Ho, P.H., Kahng, A.B., Reda, S. & Wang, Q. (2005). Power-aware placement. In *Design Automation Conference, 2005. Proceedings. 42nd*, 795–800, IEEE. 5

[44] Chishti, Z., Alameldeen, A.R., Wilkerson, C., Wu, W. & Lu, S.L. (2009). Improving cache lifetime reliability at ultra-low voltages. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, 89–99, ACM. 65

[45] David, H., Fallin, C., Gorbatov, E., Hanebutte, U.R. & Mutlu, O. (2011). Memory power management via dynamic voltage/frequency scaling. In *Proceedings of the 8th ACM international conference on Autonomic computing*, 31–40, ACM. 67

[46] Deng, Q., Meisner, D., Bhattacharjee, A., Wenisch, T.F. & Bianchini, R. (2012). Coscale: Coordinating cpu and memory system dvfs in server systems. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, 143–154, IEEE Computer Society. 67

[47] Dutta, B., Adhinarayanan, V. & Feng, W.c. (2018). GPU power prediction via ensemble machine learning for DVFS space exploration. In *Proceedings of the 15th ACM International Conference on Computing Frontiers*, 240–243, ACM. 66

[48] Duwe, H., Jian, X., Petrisko, D. & Kumar, R. (2016). Rescuing uncorrectable fault patterns in on-chip memories through error pattern transformation. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on*, 634–644, IEEE. 65

[49] Elléouet, D., Savary, Y. & Julien, N. (2006). An FPGA power aware design flow. In *International Workshop on Power and Timing Modeling, Optimization and Simulation*, 415–424, Springer. 5

[50] Ernst, D., Kim, N.S., Das, S., Pant, S., Rao, R., Pham, T., Ziesler, C., Blaauw, D., Austin, T., Flautner, K. *et al.* (2003). Razor: A low-power pipeline based on circuit-level timing speculation. In *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, 7, IEEE Computer Society. 52, 68

[51] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M. & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, **542**, 115. 7

[52] Ganapathy, S., Kalamatianos, J., Kasprak, K. & Raasch, S. (2017). On characterizing near-threshold SRAM failures in FinFET technology. In *Proceedings of the 54th Annual Design Automation Conference 2017*, 53, ACM. 65

[53] Garg, D. (2018). Power Reduction using Dynamic Voltage and Frequency Scaling (DVFS) Technique. 66

[54] Gayasen, A., Lee, K., Vijaykrishnan, N., Kandemir, M., Irwin, M.J. & Tuan, T. (2004). A dual-vdd low power FPGA architecture. In *International Conference on Field Programmable Logic and Applications*, 145–157, Springer. 61

[55] Ghasemi, H.R., Draper, S.C. & Kim, N.S. (2011). Low-voltage on-chip cache architecture using heterogeneous cell sizes for high-performance processors. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, 38–49, IEEE. 65

[56] Ghose, S., Yağlikçi, A.G., Gupta, R., Lee, D., Kudrolli, K., Liu, W.X., Hassan, H., Chang, K.K., Chatterjee, N., Agrawal, A. *et al.* (2018). What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study. *arXiv preprint arXiv:1807.05102*. 63

[57] Gopireddy, B., Song, C., Torrellas, J., Kim, N.S., Agrawal, A. & Mishra, A. (2016). ScalCore: Designing a core for voltage scalability. In *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*, 681–693, IEEE. 68

[58] Gottscho, M., BanaiyanMofrad, A., Dutt, N., Nicolau, A. & Gupta, P. (2014). Power/capacity scaling: Energy savings with simple fault-tolerant caches. In *Proceedings of the 51st Annual Design Automation Conference*, 1–6, ACM. 30

[59] Guo, K., Zeng, S., Yu, J., Wang, Y. & Yang, H. (2017). A Survey of FPGA Based Neural Network Accelerator. *arXiv preprint arXiv:1712.08934*. 1, 7, 40

[60] Gupta, S., Agrawal, A., Gopalakrishnan, K. & Narayanan, P. (2015). Deep learning with limited numerical precision. In *International Conference on Machine Learning*, 1737–1746. 38, 69

[61] Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M.A. & Dally, W.J. (2016). EIE: efficient inference engine on compressed deep neural network. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on*, 243–254, IEEE. 69

[62] Hassan, H., Vijaykumar, N., Khan, S., Ghose, S., Chang, K., Pekhimenko, G., Lee, D., Ergin, O. & Mutlu, O. (2017). SoftMC: A flexible and practical open-source infrastructure for enabling experimental DRAM studies. In

2017 IEEE International Symposium on High-Performance Computer Architecture (HPCA), 241–252, IEEE. 63

[63] Hsu, J. (2014). IBM's new brain [News]. *IEEE spectrum*, **51**, 17–19. 7, 77

[64] Hwang, A.A., Stefanovici, I.A. & Schroeder, B. (2012). Cosmic rays don't strike twice: understanding the nature of DRAM errors and the implications for system design. In *ACM SIGPLAN Notices*, vol. 47, 111–122, ACM. 63

[65] Ingle, S. & Phute, M. (2016). Tesla autopilot: semi autonomous driving, an uptick for future autonomy. *International Research Journal of Engineering and Technology*, **3**. 7

[66] Jevtić, R., Le, H.P., Blagojević, M., Bailey, S., Asanović, K., Alon, E. & Nikolić, B. (2015). Per-core DVFS with switched-capacitor converters for energy efficiency in manycore processors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **23**, 723–730. 67

[67] Jia, F., Lei, Y., Lin, J., Zhou, X. & Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, **72**, 303–315. 7

[68] Jiao, Q., Lu, M., Huynh, H.P. & Mitra, T. (2015). Improving GPGPU energy-efficiency through concurrent kernel execution and DVFS. In *Proceedings of the 13th annual IEEE/ACM international symposium on code generation and optimization*, 1–11, IEEE Computer Society. 66

[69] Jiao, X., Luo, M., Lin, J.H. & Gupta, R.K. (2017). An assessment of vulnerability of hardware neural networks to dynamic voltage and temperature

variations. In *Proceedings of the 36th International Conference on Computer-Aided Design*, 945–950, IEEE Press. 70

[70] Jouppi, N.P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A. *et al.* (2017). In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1–12, ACM. 7, 70, 77

[71] Jung, M., Mathew, D.M., Rheinländer, C.C., Weis, C. & Wehn, N. (2017). A Platform to Analyze DDR3 DRAM's Power and Retention Time. *IEEE Design & Test*, **34**, 52–59. 63

[72] Kahng, A.B., Kang, S., Kumar, R. & Sartori, J. (2010). Designing a processor from the ground up to allow voltage/reliability tradeoffs. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, 1–11, IEEE. 4, 67

[73] Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J. & Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGPLAN Notices*, **52**, 615–629. 7

[74] Kara, K., Alistarh, D., Alonso, G., Mutlu, O. & Zhang, C. (2017). FPGA-accelerated dense linear machine learning: A precision-convergence trade-off. In *2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 160–167, IEEE. 1, 36

[75] Khan, S., Lee, D., Kim, Y., Alameldeen, A.R., Wilkerson, C. & Mutlu, O. (2014). The efficacy of error mitigation techniques for DRAM retention failures: A comparative experimental study. In *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, 519–532, ACM. 63

[76] Khan, S., Lee, D. & Mutlu, O. (2016). PARBOR: An efficient system-level technique to detect data-dependent failures in DRAM. In *Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on*, 239–250, IEEE. 63

[77] Khan, S., Wilkerson, C., Lee, D., Alameldeen, A.R. & Mutlu, O. (2017). A case for memory content-based detection and mitigation of data-dependent failures in DRAM. *IEEE Computer Architecture Letters*, **16**, 88–93. 63

[78] Khan, S., Wilkerson, C., Wang, Z., Alameldeen, A.R., Lee, D. & Mutlu, O. (2017). Detecting and mitigating data-dependent DRAM failures by exploiting current memory content. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, 27–40, ACM. 63

[79] Kim, J.S., Patel, M., Hassan, H. & Mutlu, O. (2018). The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices. In *High Performance Computer Architecture (HPCA), 2018 IEEE International Symposium on*, 194–207, IEEE. 63

[80] Kim, S., Howe, P., Moreau, T., Alaghi, A., Ceze, L. & Sathe, V. (2018). MATIC: Learning around errors for efficient low-voltage neural network accelerators. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018*, 1–6, IEEE. 70

[81] Kim, S.g., Eom, H., Yeom, H.Y. & Min, S.L. (2014). Energy-centric DVFS controlling method for multi-core platforms. *Computing*, **96**, 1163–1177. 66

[82] Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J.H., Lee, D., Wilkerson, C., Lai, K. & Mutlu, O. (2014). Flipping bits in memory without accessing them: An ex-

perimental study of DRAM disturbance errors. In *ACM SIGARCH Computer Architecture News*, vol. 42, 361–372, IEEE Press. 63

[83] Kuon, I. & Rose, J. (2007). Measuring the gap between FPGAs and ASICs. *IEEE Transactions on computer-aided design of integrated circuits and systems*, **26**, 203–215. 4

[84] Lamoureux, J. & Luk, W. (2008). An overview of low-power techniques for field-programmable gate arrays. In *Adaptive Hardware and Systems, 2008. AHS'08. NASA/ESA Conference on*, 338–345, IEEE. 61

[85] Lamoureux, J. & Wilton, S.J. (2003). On the interaction between power-aware FPGA CAD algorithms. In *Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, 701, IEEE Computer Society. 5

[86] Lane, N.D. & Georgiev, P. (2015). Can deep learning revolutionize mobile sensing? In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, 117–122, ACM. 7

[87] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**, 2278–2324. 37, 39

[88] Lee, D., Kim, Y., Pekhimenko, G., Khan, S., Seshadri, V., Chang, K. & Mutlu, O. (2015). Adaptive-latency DRAM: Optimizing DRAM timing for the common-case. In *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*, 489–501, IEEE. 63

[89] Lee, D., Khan, S., Subramanian, L., Ghose, S., Ausavarungnirun, R., Pekhimenko, G., Seshadri, V. & Mutlu, O. (2017). Design-induced latency

variation in modern DRAM chips: Characterization, analysis, and latency reduction mechanisms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, **1**, 26. 27, 63, 64

[90] Lei, X., Senior, A.W., Gruenstein, A. & Sorensen, J. (2013). Accurate and compact large vocabulary speech recognition on mobile devices. In *Interspeech*, vol. 1, Citeseer. 7

[91] Leng, J., Buyuktosunoglu, A., Bertran, R., Bose, P. & Reddi, V.J. (2015). Safe limits on voltage reduction efficiency in GPUs: a direct measurement approach. In *Proceedings of the 48th International Symposium on Microarchitecture*, 294–307, ACM. 4, 66

[92] Levine, J.M., Stott, E. & Cheung, P.Y. (2014). Dynamic voltage & frequency scaling with online slack measurement. In *Proceedings of the 2014 ACM/SIGDA international symposium on Field-programmable gate arrays*, 65–74, ACM. 68

[93] Li, G., Hari, S.K.S., Sullivan, M., Tsai, T., Pattabiraman, K., Emer, J. & Keckler, S.W. (2017). 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*, 8, ACM. 36, 44

[94] Li, H., Fan, X., Jiao, L., Cao, W., Zhou, X. & Wang, L. (2016). A high performance FPGA-based accelerator for large-scale convolutional neural networks. In *Field Programmable Logic and Applications (FPL), 2016 26th International Conference on*, 1–9, IEEE. 36

[95] Lin, Y., Li, F. & He, L. (2005). Routing track duplication with fine-grained power-gating for FPGA interconnect power reduction. In *Proceedings of the 2005 Asia and South Pacific design automation conference*, 645–650, ACM. 5, 61

[96] Liu, J., Jaiyen, B., Kim, Y., Wilkerson, C. & Mutlu, O. (2013). An experimental study of data retention behavior in modern DRAM devices: Implications for retention time profiling mechanisms. In *ACM SIGARCH Computer Architecture News*, vol. 41, 60–71, ACM. 63

[97] Liu, W., Wei, Z. & Du, W. (2018). A Novel Fault-Tolerant Last-Level Cache to Improve Reliability at Near-Threshold Voltage. In *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, 231–236, ACM. 65

[98] Luo, Y., Cai, Y., Ghose, S., Choi, J. & Mutlu, O. (2015). WARM: Improving NAND flash memory lifetime with write-hotness aware retention management. In *Mass Storage Systems and Technologies (MSST), 2015 31st Symposium on*, 1–14, IEEE. 65

[99] Luo, Y., Ghose, S., Cai, Y., Haratsch, E.F. & Mutlu, O. (2018). HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness. In *High Performance Computer Architecture (HPCA), 2018 IEEE International Symposium on*, 504–517, IEEE. 65

[100] Luo, Y., Ghose, S., Cai, Y., Haratsch, E.F. & Mutlu, O. (2018). Improving 3D NAND flash memory lifetime by tolerating early retention loss and process variation. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, 106–106, ACM. 65

[101] Ma, Z., Fernandes, F.C., Dong, M., Li, X. & Hwang, S. (2017). Dynamic voltage/frequency scaling for video processing using embedded complexity metrics. US Patent 9,609,329. 66

[102] McMurtrey, D., Morgan, K.S., Pratt, B. & Wirthlin, M.J. (2008). Estimating TMR reliability on FPGAs using Markov models. 62

[103] Mei, X., Wang, Q. & Chu, X. (2017). A survey and measurement study of GPU DVFS on energy conservation. *Digital Communications and Networks*, **3**, 89–100. 66

[104] Meza, J., Wu, Q., Kumar, S. & Mutlu, O. (2015). Revisiting memory errors in large-scale production data centers: Analysis and modeling of new trends from the field. In *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*, 415–426, IEEE. 63

[105] Miftakhutdinov, R., Ebrahimi, E. & Patt, Y.N. (2012). Predicting performance impact of DVFS for realistic memory systems. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, 155–165, IEEE Computer Society. 67

[106] Miller, T., Surapaneni, N. & Teodorescu, R. (2010). Flexible error protection for energy efficient reliable architectures. In *Computer Architecture and High Performance Computing (SBAC-PAD), 2010 22nd International Symposium on*, 1–8, IEEE. 67

[107] Miller, T.N., Thomas, R., Dinan, J., Adcock, B. & Teodorescu, R. (2010). Parichute: Generalized turbocode-based error correction for near-threshold caches. In *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, 351–362, IEEE Computer Society. 68

[108] Miller, T.N., Pan, X., Thomas, R., Sedaghati, N. & Teodorescu, R. (2012). Booster: Reactive core acceleration for mitigating the effects of process variation and application imbalance in low-voltage chips. In *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, 1–12, IEEE. 67

[109] Miller, T.N., Thomas, R., Pan, X. & Teodorescu, R. (2012). VRSync: Characterizing and eliminating synchronization-induced voltage emergencies in many-core processors. In *ACM SIGARCH Computer Architecture News*, vol. 40, 249–260, IEEE Computer Society. 67

[110] Miller, T.N., Thomas, R. & Teodorescu, R. (2012). Mitigating the effects of process variation in ultra-low voltage chip multiprocessors using dual supply voltages and half-speed units. *IEEE Computer Architecture Letters*, **11**, 45–48. 67

[111] Mishra, A. & Khare, N. (2015). Analysis of DVFS techniques for improving the GPU energy efficiency. *Open Journal of Energy Efficiency*, **4**, 77. 66

[112] Moons, B. & Verhelst, M. (2017). An energy-efficient precision-scalable ConvNet processor in 40-nm CMOS. *IEEE Journal of Solid-State Circuits*, **52**, 903–914. 42

[113] Mutlu, O. (2017). The RowHammer problem and other issues we may face as memory becomes denser. In *Proceedings of the Conference on Design, Automation & Test in Europe*, 1116–1121, European Design and Automation Association. 63

[114] Mutlu, O. & Moscibroda, T. (2008). Parallelism-aware batch scheduling: Enhancing both performance and fairness of shared DRAM systems. In *ACM SIGARCH Computer Architecture News*, vol. 36, 63–74, IEEE Computer Society. 63

[115] Mutlu, O. & Subramanian, L. (2015). Research problems and opportunities in memory systems. *Supercomputing frontiers and innovations*, **1**, 19–55. 67

[116] Nazar, G.L. & Carro, L. (2012). Exploiting modified placement and hard-wired resources to provide high reliability in FPGAs. In *Field-Programmable Custom Computing Machines (FCCM), 2012 IEEE 20th Annual International Symposium on*, 149–152, IEEE. 62

[117] Neshatpour, K., Burleson, W., Khajeh, A. & Homayoun, H. (2018). Enhancing Power, Performance, and Energy Efficiency in Chip Multiprocessors Exploiting Inverse Thermal Dependence. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **26**, 778–791. 9, 33

[118] Niknahad, M., Sander, O. & Becker, J. (2012). Fine grain fault tolerance- A key to high reliability for FPGAs in space. In *Aerospace Conference, 2012 IEEE*, 1–10, IEEE. 62

[119] Nunez-Yanez, J. (2015). Adaptive voltage scaling with in-situ detectors in commercial FPGAs. *IEEE Transactions on Computers*, 1–1. 66

[120] Nunez-Yanez, J. (2017). Adaptive voltage scaling in a heterogeneous FPGA device with memory and logic in-situ detectors. *Microprocessors and Microsystems*, **51**, 227–238. 67

[121] Nunez-Yanez, J.L., Hosseinabady, M. & Beldachi, A. (2016). Energy optimization in commercial FPGAs with voltage, frequency and logic scaling. *IEEE Transactions on Computers*, **65**, 1484–1493. 66

[122] Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G. & Marr, D. (2016). Accelerating binarized neural networks: comparison of FPGA, CPU, GPU, and ASIC. In *Field-Programmable Technology (FPT), 2016 International Conference on*, 77–84, IEEE. 4, 62

[123] Nurvitadhi, E., Sim, J., Sheffield, D., Mishra, A., Krishnan, S. & Marr, D. (2016). Accelerating recurrent neural networks in analytics servers: Comparison of FPGA, CPU, GPU, and ASIC. In *Field Programmable Logic and Applications (FPL), 2016 26th International Conference on*, 1–4, IEEE. 4, 62

[124] Nurvitadhi, E., Venkatesh, G., Sim, J., Marr, D., Huang, R., Ong Gee Hock, J., Liew, Y.T., Srivatsan, K., Moss, D., Subhaschandra, S. *et al.* (2017). Can FPGAs beat GPUs in accelerating next-generation deep neural networks? In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 5–14, ACM. 7

[125] Pan, J.H., Mitra, T. & Wong, W.F. (2004). Configuration bitstream compression for dynamically reconfigurable FPGAs. In *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, 766–773, IEEE. 5

[126] Pandey, B., Yadav, J., Pattanaik, M. & Rajoria, N. (2013). Clock gating based energy efficient ALU design and implementation on FPGA. In *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on*, 93–97, IEEE. 5, 61

[127] Papadimitriou, G., Kaliorakis, M., Chatzidimitriou, A., Gizopoulos, D., Lawthers, P. & Das, S. (2017). Harnessing voltage margins for energy efficiency in multicore CPUs. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, 503–516, ACM. 4, 67

[128] Park, J.G., Hsieh, C.Y., Dutt, N. & Lim, S.S. (2016). Co-Cap: energy-efficient cooperative CPU-GPU frequency capping for mobile games. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 1717–1723, ACM. 67

[129] Patel, J. (2008). Cmos process variations: A critical operation point hypothesis. In *Online Presentation*. 66

[130] Patel, M., Kim, J.S. & Mutlu, O. (2017). The reach profiler (REAPER): Enabling the mitigation of DRAM retention failures via profiling at aggressive conditions. *ACM SIGARCH Computer Architecture News*, **45**, 255–268. 63

[131] Pathania, A., Jiao, Q., Prakash, A. & Mitra, T. (2014). Integrated CPU-GPU power management for 3D mobile games. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, 1–6, IEEE. 67

[132] Phatak, D.S. & Koren, I. (1995). Complete and partial fault tolerance of feedforward neural nets. *IEEE Transactions on Neural Networks*, **6**, 446–456. 36, 69

[133] Pothukuchi, R.P., Ansari, A., Gopireddy, B. & Torrellas, J. (2017). Sthira: A Formal Approach to Minimize Voltage Guardbands under Variation in Networks-on-Chip for Energy Efficiency. In *Parallel Architectures and Compilation Techniques (PACT), 2017 26th International Conference on*, 260–272, IEEE. 68

[134] Putnam, A., Caulfield, A.M., Chung, E.S., Chiou, D., Constantinides, K., Demme, J., Esmaeilzadeh, H., Fowers, J., Gopal, G.P., Gray, J. *et al.* (2014). A reconfigurable fabric for accelerating large-scale datacenter services. *ACM SIGARCH Computer Architecture News*, **42**, 13–24. 7, 77

[135] Reagen, B., Whatmough, P., Adolf, R., Rama, S., Lee, H., Lee, S.K., Hernández-Lobato, J.M., Wei, G.Y. & Brooks, D. (2016). Minerva: Enabling low-power, highly-accurate deep neural network accelerators. In *ACM SIGARCH Computer Architecture News*, vol. 44, 267–278, IEEE Press. 4, 7, 36, 42, 62, 69

[136] Rosenfeld, P., Cooper-Balis, E. & Jacob, B. (2011). DRAMSim2: A cycle accurate memory system simulator. *IEEE Computer Architecture Letters*, **10**, 16–19. 63

[137] Sadrosadati, M., Mirhosseini, A., Aghilinasab, H. & Sarbazi-Azad, H. (2015). An efficient dvs scheme for on-chip networks using reconfigurable virtual channel allocators. In *Low Power Electronics and Design (ISLPED), 2015 IEEE/ACM International Symposium on*, 249–254, IEEE. 68

[138] Salami, B., Zamani, M.S. & Jahanian, A. (2011). VMAP: A variation map-aware placement algorithm for leakage power reduction in FPGAs. In *Digital System Design (DSD), 2011 14th Euromicro Conference on*, 81–87, IEEE. 61

[139] Salami, B., Arcas-Abella, O. & Sonmez, N. (2015). HATCH: hash table caching in hardware for efficient relational join on FPGA. In *Field-Programmable Custom Computing Machines (FCCM), 2015 IEEE 23rd Annual International Symposium on*, 163–163, IEEE. 1

[140] Salami, B., Arcas-Abella, O., Sonmez, N., Unsal, O. & Kestelman, A.C. (2016). Accelerating Hash-Based Query Processing Operations on FPGAs by a Hash Table Caching Technique. In *Latin American High Performance Computing Conference*, 131–145, Springer. 1

[141] Salami, B., Malazgirt, G.A., Arcas-Abella, O., Yurdakul, A. & Sonmez, N. (2017). AxleDB: A novel programmable query processing platform on FPGA. *Microprocessors and Microsystems*, **51**, 142–164. 1

[142] Salami, B., Unsal, O. & Cristal, A. (2018). On the Resilience of RTL NN Accelerators: Fault Characterization and Mitigation. *High Performance Machine Learning Workshop (HPML) in conjunction with 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. 44

[143] Sallab, A.E., Abdou, M., Perot, E. & Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, **2017**, 70–76. 7

[144] Santoro, G., Casu, M.R., Peluso, V., Calimera, A. & Alioto, M. (2018). Design-Space Exploration of Pareto-Optimal Architectures for Deep Learning with DVFS. In *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, 1–5, IEEE. 66

[145] Sarangi, S.R., Greskamp, B., Teodorescu, R., Nakano, J., Tiwari, A. & Torrellas, J. (2008). VARIUS: A model of process variation and resulting timing errors for microarchitects. *IEEE Transactions on Semiconductor Manufacturing*, **21**, 3–13. 27

[146] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, **61**, 85–117. 7

[147] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. *et al.* (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, **529**, 484. 7

[148] Son, Y.H., Lee, S., Seongil, O., Kwon, S., Kim, N.S. & Ahn, J.H. (2015). CiDRA: A cache-inspired DRAM resilience architecture. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, 502–513, IEEE. 65

[149] Sterpone, L. & Violante, M. (2006). A new reliability-oriented place and route algorithm for SRAM-based FPGAs. *IEEE Transactions on Computers*, 732–744. 62

[150] Stott, E., Sedcole, P. & Cheung, P.Y. (2008). Fault tolerant methods for reliability in FPGAs. In *Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on*, 415–420, IEEE. 62

[151] Stott, E., Levine, J.M., Cheung, P.Y. & Kapre, N. (2014). Timing fault detection in FPGA-based circuits. In *Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Annual International Symposium on*, 96–99, IEEE. 68

[152] Suda, N., Chandra, V., Dasika, G., Mohanty, A., Ma, Y., Vrudhula, S., Seo, J.s. & Cao, Y. (2016). Throughput-optimized OpenCL-based FPGA accelerator for large-scale convolutional neural networks. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 16–25, ACM. 36

[153] Swaminathan, K., Chandramoorthy, N., Cher, C.Y., Bertran, R., Buyuktosunoglu, A. & Bose, P. (2017). Bravo: Balanced reliability-aware voltage optimization. In *High Performance Computer Architecture (HPCA), 2017 IEEE International Symposium on*, 97–108, IEEE. 4, 67

[154] Sze, V., Chen, Y.H., Yang, T.J. & Emer, J.S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, **105**, 2295–2329. 7, 69

[155] Tan, J., Song, S.L., Yan, K., Fu, X., Marquez, A. & Kerbyson, D. (2016). Combating the reliability challenge of GPU register file at low supply voltage. In *Parallel Architecture and Compilation Techniques (PACT), 2016 International Conference on*, 3–15, IEEE. 67

[156] Tchernev, E.B., Mulvaney, R.G. & Phatak, D.S. (2005). Investigating the fault tolerance of neural networks. *Neural Computation*, **17**, 1646–1664. 36

[157] Temam, O. (2012). A defect-tolerant accelerator for emerging high-performance applications. In *Computer Architecture (ISCA), 2012 39th Annual International Symposium on*, 356–367, IEEE. 36, 44, 69

[158] Thomas, R., Barber, K., Sedaghati, N., Zhou, L. & Teodorescu, R. (2016). Core Tunneling: Variation-aware voltage noise mitigation in GPUs. In *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*, 151–162, IEEE. 67

[159] Thomas, R., Sedaghati, N. & Teodorescu, R. (2016). EmerGPU: Understanding and mitigating resonance-induced voltage noise in GPU architectures. In *Performance Analysis of Systems and Software (ISPASS), 2016 IEEE International Symposium on*, 79–89, IEEE. 67

[160] Tian, Y., Pei, K., Jana, S. & Ray, B. (2018). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*, 303–314, ACM. 7

[161] Torres-Huitzil, C. & Girau, B. (2017). Fault and Error Tolerance in Neural Networks: A Review. *IEEE Access*, **5**, 17322–17341. 69

[162] Tuan, T., Kao, S., Rahman, A., Das, S. & Trimberger, S. (2006). A 90nm low-power FPGA for battery-powered applications. In *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, 3–11, ACM. 5

[163] Tziantzioulis, G., Gok, A., Faisal, S., Hardavellas, N., Ogrenci-Memik, S. & Parthasarathy, S. (2015). b-HiVE: A bit-level history-based error model with value correlation for voltage-scaled integer and floating point units. In *Proceedings of the 52nd Annual Design Automation Conference*, 105, ACM. 68

[164] UL ISLAM, F.M.M., LIN, M., YANG, L.T. & CHOO, K.K.R. (2018). Task aware hybrid DVFS for multi-core real-time systems using machine learning. *Information Sciences*, **433**, 315–332. 67

[165] WHATMOUGH, P.N., LEE, S.K., LEE, H., RAMA, S., BROOKS, D. & WEI, G.Y. (2017). 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 *Solid-State Circuits Conference (ISSCC), 2017 IEEE International*, 242–243, IEEE. 4, 36, 70

[166] WILKERSON, C., GAO, H., ALAMELDEEN, A.R., CHISHTI, Z., KHELLAH, M. & LU, S.L. (2008). Trading off cache capacity for reliability to enable low voltage operation. In *ACM SIGARCH computer architecture news*, vol. 36, 203–214, IEEE Computer Society. 65

[167] WIRTHLIN, M., JOHNSON, E., ROLLINS, N., CAFFREY, M. & GRAHAM, P. (2003). The reliability of FPGA circuit designs in the presence of radiation induced configuration upsets. In *Field-Programmable Custom Computing Machines, 2003. FCCM 2003. 11th Annual IEEE Symposium on*, 133–142, IEEE. 62

[168] WIRTHLIN, M.J. (2004). Improving the reliability of FPGA circuits using triple-modular redundancy (TMR) & efficient voter placement. In *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*, 252–252, ACM. 68

[169] WU, Y., NUNEZ-YANEZ, J., WOODS, R. & NIKOLOPOULOS, D.S. (2014). Power modelling and capping for heterogeneous ARM/FPGA SoCs. In *Field-Programmable Technology (FPT), 2014 International Conference on*, 231–234, IEEE. 66, 67

[170] Yalcin, G., Unsal, O. & Cristal, A. (2013). FaulTM: error detection and recovery using hardware transactional memory. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*, 220–225, IEEE. 68

[171] Yalcin, G., Islek, E., Tozlu, O., Reviriego, P., Cristal, A., Unsal, O.S. & Ergin, O. (2014). Exploiting a fast and simple ECC for scaling supply voltage in level-1 caches. In *On-Line Testing Symposium (IOLTS), 2014 IEEE 20th International*, 1–6, IEEE. 4, 68

[172] Yalcin, G., Rethinagiri, S.K., Palomar, O., Unsal, O., Cristal, A. & Milojevic, D. (2016). Exploring Energy Reduction in Future Technology Nodes via Voltage Scaling with Application to 10nm. In *Parallel, Distributed, and Network-Based Processing (PDP), 2016 24th Euromicro International Conference on*, 184–191, IEEE. 4

[173] Yang, L. & Murmann, B. (2017). Approximate SRAM for Energy-Efficient, Privacy-Preserving Convolutional Neural Networks. In *VLSI (ISVLSI), 2017 IEEE Computer Society Annual Symposium on*, 689–694, IEEE. 36, 65, 70

[174] Yang, L. & Murmann, B. (2017). SRAM voltage scaling for energy-efficient convolutional neural networks. In *Quality Electronic Design (ISQED), 2017 18th International Symposium on*, 7–12, IEEE. 4, 36, 65, 70

[175] Yang, T.J., Howard, A., Chen, B., Zhang, X., Go, A., Sze, V. & Adam, H. (2018). NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications. *arXiv preprint arXiv:1804.03230*. 35

[176] Yoon, D.H. & Erez, M. (2009). Memory mapped ECC: low-cost error protection for last level caches. In *ACM SIGARCH Computer Architecture News*, vol. 37, 116–127, ACM. 65

[177] You, D. & Chung, K.S. (2015). Quality of service-aware dynamic voltage and frequency scaling for embedded GPUs. *IEEE Computer Architecture Letters*, **14**, 66–69. 66

[178] Yu, J., Lukefahr, A., Palframan, D., Dasika, G., Das, R. & Mahlke, S. (2017). Scalpel: Customizing dnn pruning to the underlying hardware parallelism. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 548–560, ACM. 69

[179] Zhang, J., Wang, Z. & Verma, N. (2017). In-memory computation of a machine-learning classifier in a standard 6T SRAM array. *IEEE Journal of Solid-State Circuits*, **52**, 915–924. 69

[180] Zhang, J., Rangineni, K., Ghodsi, Z. & Garg, S. (2018). Thundervolt: enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators. In *Proceedings of the 55th Annual Design Automation Conference*, 19, ACM. 36, 69

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

[182] Zhang, Y., Roivainen, J. & Mammela, A. (2006). Clock-gating in FPGAs: A novel and comparative evaluation. In *Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on*, 584–590, IEEE. 5

[183] Zuchowski, P.S., Reynolds, C.B., Grupp, R.J., Davis, S.G., Cremen, B. & Troxel, B. (2002). A hybrid ASIC and FPGA architecture. In *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, 187–194, ACM. 4