EMBRACING LOW-POWER SYSTEMS WITH IMPROVEMENT IN SECURITY AND ENERGY-EFFICIENCY

by

Pramesh Pandey

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Approved:

Sanghamitra Roy, Ph.D. Major Professor	Koushik Chakraborty, Ph.D. Committee Member
Jacob Gunther, Ph.D. Committee Member	Reyhan Baktur, Ph.D. Committee Member
Vicki H Allan, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Interim Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2021

ABSTRACT

Embracing Low-Power Systems with Improvement in Security and Energy-Efficiency

by

Pramesh Pandey, Doctor of Philosophy

Utah State University, 2021

Major Professor: Sanghamitra Roy, Ph.D.
Department: Electrical and Computer Engineering

The stagnation of Moore's Law and huge demand in the performance brought about by economies around the world based on computing, the necessity of low power design is becoming inevitable. As a result of energy inefficiencies in conventional architectures while performing AI computations, the computing industry has already invited the use of specialized computing architectures, such as Tensor Processing Unit (TPU).

Among many research efforts in increasing the energy efficiency of the computing systems, Near-Threshold Computing (NTC) has been a prominent low power design paradigm offering a quadratic reduction in power consumption through aggressive underscaling of the chip supply voltage, in comparison to the conventional Super-Threshold Computing (STC). However, the extreme sensitivity to manufacturing process variation (PV) and inherent slow down of the speed in the transistor operated in this regime, result to serious reliability and performance problems. This is causing a bottleneck to the adoption of NTC paradigm in mainstream semiconductor system designs. In this work, two disparate implementations (viz. SRAM Physical Unclonable Funtions (SPUF) and TPU) in NTC are assessed for their security and performance characteristics respectively. This dissertation improves the security properties of the NTC SPUFs by reforming the reliability and uniformity characteristics. Next, $2\times - 3\times$ higher performance is unlocked in the NTC TPU by the

providing predictive timing error resilience. Also, novel power saving opportunities are identified in the baseline STC TPU with rigorous mathematical analysis on the usage pattern of the TPU systolic array. The opportunities are exploited through dynamic dataflow adaptive power gating to curtail the wasteful leakage power, to attain $3.5\times - 6.5\times$ higher energy efficiency.

(87 pages)

PUBLIC ABSTRACT

Embracing Low-Power Systems with Improvement in Security and Energy-Efficiency

Pramesh Pandey

As the economies around the world are aligning more towards usage of computing systems, the global energy demand for computing is increasing rapidly. Additionally, the boom in AI based applications and services has already invited the pervasion of specialized computing hardware architectures for AI (accelerators). A big chunk of research in the industry and academia is being focused on providing energy efficiency to all kinds of power hungry computing architectures. This dissertation adds to these efforts.

Aggressive voltage underscaling of chips is one the effective low power paradigms of providing energy efficiency. This dissertation identifies and deals with the reliability and performance problems associated with this paradigm and innovates novel energy efficient approaches. Specifically, the properties of a low power security primitive have been improved and, higher performance has been unlocked in an AI accelerator (Google TPU) in an aggressively voltage underscaled environment. And, novel power saving opportunities have been unlocked by characterizing the usage pattern of a baseline TPU with rigorous mathematical analysis.

To my dearest grandfather Kedar, mother Pabitra and sister Shilpa, who all rest in heaven and mystically guide me towards a content life.

ACKNOWLEDGMENTS

I would like to remember and offer my sincere gratitude to several persons, who have helped me in their own ways throughout the Ph.D. journey. I would like to thank my major advisor Dr. Sanghamitra Roy, and my co-advisor Dr. Koushik Chakraborty for their continual advice, encouragement, and feedback that have helped me to mold my curiosities and general apprehension towards engineering to methodical research aptitude. Their contribution fluidly extends outside of academia with their cordial hospitality towards me and my wife. I thank my Ph.D. committee members Dr. Jacob Gunther, Dr. Reyhan Baktur and Dr. Vicki Allan, for their valuable insights and feedbacks on my research. I have so much to thank Tricia Brandenburg, my graduate program coordinator for bearing the burden of my institutional formalities and advising me so gracefully. I also appreciate the efforts of Diane, Kathy and Brady from the department for easing my journey. I thank Patrick Cuevas, Luke Faber and Betty Rosado from Qualcomm for gracefully introducing and guiding me to the semiconductor industry, during my internships.

I am extremely thankful for my colleagues at the BRIDGE lab. I thank Prabal, whose personality inspired me to approach things rationally both in life and research; Chidham, for reminding the blissful fundamentals of my life as a human; Rajesh, for always being there for me, helping to effortlessly integrate my personal and professional life; Asmita, Sourav and Shamik for being my very dear friends, with whom I could relive my fun undergrad days; Tahmoures for showing the alternate understandings of life in terms of the struggle and perseverance; Aatreyi for being there like a strict sister and inspiring me with her tactical research aptitude; Noel for being a great research partner and always keeping me in his prayers.

I thank my dear wife Padma, for being my unconditional life partner throughout the journey, bearing with my Ph.D. induced rationalism, and continually pushing and micromanaging me towards goals. I thank my family; my dear parents Ramesh, Pabitra, Puspa and grandparents for always nurturing me to this point and beyond; my brother Mahesh

for being my best friend and second father; my sisters Shila and Seema for holding and cherishing me in their heart forever; sister in-law Preeza, brothers-in-law Sunil, Bhim, Hemant and Narayan, mother-in-law Sita for always believing and motivating me. I am grateful to nephews Ayden, Seasun, Bibhusan, neice, Samridhi and my little friend Deep for enlightening me with their smiles, and making me hopeful for the future; my cousins and their families in US, Jay Nepal, Himal, Bidhan, Shisir, Prativa, Sandeep, Sanju, Saru and Gopal for extending my home in the US. Finally, I am very grateful for my Nepali family in Logan for giving me a heartfelt homely warmth throughout the Ph.D. journey.

Pramesh Pandey

CONTENTS

Page

LIST OF FIGURES

ACRONYMS

TPU	Tensor Processing Unit
GPU	Graphics Processing Unit
CPU	Central Processing Unit
STC	Super-Threshold Computing
NTC	Near-Threshold Computing
NTV	Near-Threshold Voltage
VLSI	Very Large Scale Integration
MAC	Multiply Accumulate Unit
PE	Processing Element
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
NMOS	N-channel Metal-Oxide Semiconductor
PMOS	P-channel Metal-Oxide Semiconductor
FinFET	Fin Field-Effect Transistor
SPICE	Simulation Program with Integrated Circuit Emphasis
FO4	FanOut-of-4
STA	Statistical Timing Analysis
PTM	Predictive Technology Model
RTL	Register Transfer Level
DRAM	Dynamic Random Access Memory
SRAM	Static Random Access Memory
PUF	Physical Unclonable Function
SPUF	SRAM Physical Unclonable Function
6TSRAM	6-Transistor Static Random Access Memory
8TSRAM	8-Transistor Static Random Access Memory
RFID	Radio-Frequency IDentification
IoT	Internet of Things

PUC	Percentage of Unreliable Cells
BSIM-CMG	Berkeley Short-channel IGFET Model – Common Multi-Gate
FPGA	Field Programmable Gate Array
CUBIT	Current Suppression with Biasing Technique
CUSIT	Current Suppression with Sizing Technique
RLT	Reliability Loss Threshold
ULT	Uniformity Loss Threshold
AI	Artificial Intelligence
NN	Neural Network
ML	Machine Learning
DNN	Deep Neural Network
SA	Systolic Array
ELT	Error Log Table
BCU	Boost Control Unit
GTR	GreenTPU Reactive
GTL	GreenTPU Lite
SeMU	Sequence Monitor Unit
TECU	Timing Error Control Unit
ESU	Error Sensing Unit
BCR	Boost Control Register
VBE	Voltage Boost Energy
TRU	True Resource Usage
MAR	Maximum Available Resource
RUR	Resource Usage Ratio
UPTPU	Underutilization based Power-gating paradigm for TPU
SPG	Systolic Power Gating
ZWPG	Zero Weight Power Gating
NVM	Non Volatile Memory
STT-MRAM	Spin Transfer Torque Magnetic Random-Access Memory
ZS	Zero-Skip

CHAPTER 1

INTRODUCTION

From mundane livelihood of individuals to modern economies around the world, computing industry has touched almost every aspect of 21st century. Andrae et al. have projected that global computing systems will consume about 21% of the world's electrical energy by the year 2030 [1]. This can be attributed to the spikes in the energy demands in data centers and the rapid rise of portable and IoT devices at the edge. Furthermore, the rise in performance demand from slow paced hardware development (characterized by stagnated Moore's Law), has forced the computing infrastructure to operate at very tighter thermal bounds. The latest boom in the AI is also demanding huge pool of extremely low-power and battery powered and smart edge devices. This calls for low-power design paradigms to be adapted into mainstream computing industry. However, the severe drop in low power system's performance along with associated reliability and security risks are rendering the adaptation very slow.

The total power consumption in VLSI is composed of switching or dynamic power and idle or static power. The dynamic power is quadratically dependent on the supply voltage. Near Threshold Computing (NTC) is one of the design paradigms which exploits this fundamental property which promises to significantly decrease the power consumption. NTC operates its devices at a supply voltage close and slightly higher than the devices' switching threshold voltage. This operation, while dramatically reduces the power consumption, invites many performance and reliability concerns. The devices fundamentally operate slower when operated at lower voltages, and the delay variability due to extreme sensitivity to process and environmental variations cause reliability concerns. The practical adaptation of NTC can only be successful by various circuit-architectural innovations that can deal with these performance and reliability concerns.

Two bodies of work in the dissertation investigate and innovate in NTC's security and

performance characteristics through two disparate computation implementations. Chapter 3 explores the security characteristics of 8TSRAM Physical Unclonable Functions (PUF) operating at NTC on the metrics of reliability and uniformity. Chapter 4 addresses the performance issues of a NTC Tensor Processing Unit (TPU) by providing it adequate timing error resilience, so that it can perform at $2\times - 3\times$ faster than its NTC operation.

The third body of work in the dissertation is devoted to providing energy efficiency to TPU by preventing the large bulk of the wasteful idle power. Chapter 5 presents this work which mathematically showcases the vast amount of leakage power and prevents it with systolic powergating. Chapter 2 performs literature review of the research efforts in the academics pertinent to the all the contributions in this dissertation. Chapter 6 concludes the works of this dissertation. Section 1.1 presents the formal contributions of the works in this disserations to the academia through several journals and conference publications.

1.1 Contributions of This Dissertation

The works presented in this dissertation have been published in several conference proceedings and journal articles, including 2016 and 2020 IEEE/ACM Design Automation Conference (DAC), 2018 International Symposium on Low Power Electronics and Design (ISLPED), 2020 IEEE Transactions on Very Large Scale Integration Systems (TVLSI), 2020 Journal of Low Power Electronics and Applications (JLPEA). Details of the publications are listed below:

1.1.1 Conference Papers

- UPTPU: Improving Energy Efficiency of a Tensor Processing Unit through Underutilization Based Power-Gating. Pramesh Pandey, Noel Daniel Gundi, Koushik Chakraborty and Sanghamitra Roy. Accepted for publication in *IEEE/ACM Design Automation Conference (DAC), 2021.*

- GreenTPU: Improving Timing Error Resilience of a Near-Threshold Tensor Processing Unit. Pramesh Pandey, Prabal Basu, Koushik Chakraborty and Sanghamitra Roy.

IEEE/ACM *Design Automation Conference (DAC)*, 2019.

- Reliability and Uniformity Enhancement in 8T-SRAM based PUFs operating at NTC. Pramesh Pandey, Asmita Pal, Koushik Chakraborty, Sanghamitra Roy. *International Symposium on Low Power Electronics and Design (ISLPED)'18*.

1.1.2 Journal Articles

- Challenges and Opportunities in Near-Threshold DNN Accelerators around Timing Errors. Pramesh Pandey, Noel Daniel Gundi, Prabal Basu, Tahmoures Shabanian, Mitchell Patrick, Koushik Chakraborty, Sanghamitra Roy. *Journal of Low Power Electronics and Applications 2020, 10(4), 33*

- GreenTPU: Predictive Design Paradigm for Improving Timing Error Resilience of a Near-Threshold Tensor Processing Unit. Pramesh Pandey, Prabal Basu, Koushik Chakraborty, Sanghamitra Roy. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 7, pp. 1557-1566, July 2020*

CHAPTER 2

LITERATURE REVIEW

This chapter presents an extensive literature review on the research efforts, pertinent to the works in this dissertation. The works include the areas around SRAM PUFs, low power computing, DNN accelerators and power gating. Section 2.1 discusses the fundamental works embracing energy efficiency through near threshold computing. Section 2.2 discusses the introductory works on SRAM PUFs. Section 2.3 points out the alternate SRAM designs targeted for low power operation. Section 2.4 details the works which improve the quality of SRAM PUFs. Section 2.5 discusses and classifies the works around energy efficiency of DNN accelerators components. Section 2.6 discusses works on the powergating approach used to improve system energy efficiency.

2.1 Works on Near Threshold Computing (NTC)

- **Dreslinski et al. [2]:** This work serves as the modern day designer guide for NTC systems and also highlights the inability of 6T-SRAM to be used as a reliable memory device at NTC. They highlight that SRAM is a site for high yield requirements and the aggressive sizing of the SRAMs result in very high sensitivity to local variation. The combination of global and local variations at NTC leads to several functional read/write failures.
- **Pinckney et al. [3]:** This work explores on how the cessation of Dennard scaling can be dealt with by a near-threshold voltage operation of a chip multiprocessor. Utilizing the inherent parallelism of the applications is presented as the saving grace. With the parallelization overhead, their NTC operation provides 4× improvement in the CPU performance across 6 commercial technology nodes.
- **Hsu et al. [4]:** This work proposes a reconfigurable single instruction multiple data vector permutation engine that can work at the NTC region, while tolerating process

variation. The ultra-low voltage optimizations drop down the power to 109 micro-watt at 0.28V, achieving 9× higher energy efficiency.

- **Marković et al. [5]:** The authors explore the near threshold operation of systems with supply voltage variations and transistor sizing. The authors introduce a pass-transistor based logic family with only sub-threshold leakage while operating at the near-threshold region. The work uses the ultra-low power design in the design synthesis.

2.2 SRAM PUF Implementations

- **Suh et al. [6]:** Suh et al. introduce PUFs as critical and low overhead security primitives for device authentication and secret key generation. They present PUF designs that exploit inherent delay characteristics within the wires and transistors of the IC. They describe how PUFs can be made from these characteristics that differ among chip of same design/function. They showcase the generation of volatile secret keys for cryptographic operations and chip identification.
- **Holcomb et al. [7]:** Holcomb et al. give the first comprehensive basis for using the initial power up state of SRAMs as electronic fingerprint for devices with SRAMs already in it. They also extend the use of non reliable bitcells to develop true random number generator from power up state of SRAMs.
- **Selimis et al. [8]:** The authors successfully evaluate low power 90nm commercial 6T-SRAMs of Wireless Sensor Networks (WSN) at different environmental, electrical, and ageing conditions, for operation as PUF primitives. They extend SRAM PUF implementation with fuzzy extractor to generate unique cryptographic keys.
- **Kaseem et al. [9]:** The authors introduce a sub-threshold PUF based on the 10T-SRAM cell as a suitable low-power solution for secure devices. This parameters of reliability and uniformity have not been addressed at sub-threshold operation for the 10T-SRAM PUF. The work weakly evaluates 10TSRAM to be a primitively viable option for PUF implementation.

2.3 Alternate SRAM configurations

- **Chang et al. [10]:** The authors propose an eight transistor SRAM cell architecture to improve variability tolerance and low-voltage operation within high-speed SRAM caches. They design 32 kb 8TSRAM array without significant area penalty by modifying traditional 6T-SRAM techniques.
- **Calhoun et al. [11]:** Calhoun et al. describe the specifics of design of a ten transistor SRAM cell that can operate below 400mv. The design achieves 2.25× lower leakage power and 2.25× lower active energy than its 6T counterpart at 0.6V.
- **7T-SRAM et al. [12]:** This work proposes a seven transistor SRAM cell that improves read stability and write ability of the conventional 6T SRAM cell at low voltages. The authors improve read stability and write ability of the conventional 6T SRAM cell by separating read and write access transistors in this cell.

2.4 SRAM PUF Improvements

- **Garg et al. [13]:** This work addresses uniformity and reliability improvement of SRAM PUFs by utilizing aging effects. , like application of NBTI stress. This technique, based on 6TSPUF cannot control the gate leakage current which is primarily responsible for degraded uniformity, in an 8T-SPUF.
- **Bhargava et al. [14]:** This work demonstrates the efficacy and associated costs of directed accelerated aging, multiple evaluations, and activation control, three SRAM PUF's reliability enhancing techniques. They base their evaluation of a 65nm custom PUF chip and measure a 40%-71% improvement in reliability with these techniques.
- **Chellapa et al. [15]:** Chellapa et al. propose an alternative SRAM cell power-up strategy, by raising the wordline and decoder voltage higher than array voltage. They also show an extensive mathematical proof to how their fingerprinting extraction technique is better than conventional SRAM powering up method. However, the voltage raise defeats the key purpose of NTC operation.

- **Chang et al. [16]:** The authors argue that the design methods usable for PUFs to maximize the mismatches between the transistors in SRAMs, act badly when those cells have to be used as memory elements, by giving more read/write failures. They propose several voltage scaling/biasing and sizing strategies enhance SPUF reliability and embrace dual mode use of expensive SRAM cells.
- **Elshafiey et al. [17]:** Elshafiey et al. model the effect of power supply ramp time on SRAM PUFs with a binary classification of Vdd Ramp up time regions, based on either threshold variations only dominate or both capacitive and threshold variations dominate.
- **Simons et al. [18]:** Simons et al. , This work acknowledge the importance of voltage ramp up times on the reliability of SRAM based PUFs, in addition to the conventional temperature and voltage based reliability. They argue that Vdd can influence the stability of PUF responses. They advise to keep the faster ramp-up time of PUF primitives.

To the best of our knowledge, the work in the dissertation is the first one which has explored reliability and uniformity characteristics for 8T-SPUFs operated at NTC and adopt efficient design strategies to overcome their adverse effects.

2.5 Improving energy efficiency of DNN accelerators

Several efforts have been made to improve energy efficiency of components around DNN accelerators. Section 2.5.1 discusses the innovations around architectural elements. Section 2.5.2 discusses the works improving the energy efficiency through innovations around memory. Section 2.5.3 reviews the innovations on the analog and mixed signal components of the DNN accelerators.

2.5.1 Architectural Enhancements

- **Li et al. [19]:** This work demonstrates that by providing appropriate precision and numeric range to values in each layer, the failure rate can be reduced by 200x. In

each layer of DNN, this technique uses a 'symptom based fault detection' scheme to identify the range of values and adds a 10% guard-band.

- **Libano et al. [20]:** This work proposes a scheme to design and apply triple modular redundancy selectively to the vulnerable NN layers to effectively mask the faults.
- **TE-Drop [21]:** Zhang et al. proposed a timing speculation approach that enables an aggressive voltage underscaling in DNN accelerators without compromising the classification accuracy [21]. The authors expect a timing error at a MAC, detect it and drop the computation to isolate the damage the errant computation can bring. They use the inherent error tolerance of the DNN implementations.
- **Choi et al. [22]:** Choi et al. proposed error resilient techniques to enable aggressive voltage scaling by exploiting the variable error resilience exposed by different components of DNN The authors approximate variable weight sensitivity by using Taylor expansion and assign the highly sensitive weights to robust MACs and weakly sensitive weights to underpowered and variation prone MACs.
- **Zhang et al. [23]:** This work evaluates the drop of classification accuracy in the presence of faults in TPU systolic array and proposes design of fault-tolerant, systolic array based DNN accelerators for high defect rate technologies in case of permanent hardware faults. Their proposal is based on fault-aware pruning and combination of fault-aware pruning and retraining. They show that their techniques can tolerate upto 50% in the TPU.
- **Chen et al. [24]:** The authors analyze how dataflow plays a very important role in energy efficiency optimization in DNN accelerators and provide guidelines on future DNN accelerator designs. They propose an optimal MAC operation mapping rule, called Row-Stationary dataflow, that optimizes the data movement inside a deep CNN, resulting in a superior system-level energy efficiency.
- **Minerva [25]:** This work demonstrates an automated co-design approach across the algorithm, architecture, and circuit to offer a staggering 8.1$\times$ power reduction over a baseline DNN accelerator, without compromising the accuracy. The authors take

holistic approach in optmizing and combining gains from different granularities of DNN hardware, such as algorithm, architecture, and circuit.

- **Lin et al. [26]:** Lin et al. This work presents a statistical error compensation technique to correct process variation induced timing errors in CNNs, operating under near-threshold condition. The authors use a ripple carry adder to show the exacerbated delay variation at NTC and with their technique, achieve an 11x improvement in variation tolerance when comparison to a conventional CNN.
- **Whatmough et al. [27, 28]:** This work has incorporated several techniques like curbing unwanted computations, providing algorithmic error tolerance, timing violation tolerance and so on to come up with an extremely energy efficient DNN SoC, in actual hardware. They provide the timing error tolerance by complementing Razor with their time borrowing techniques in [29, 30].
- **Hegde et al. [31]:** propose a predictive scheme to tackle timing errors coming as a result of critical undervolting in DSP architectures They compensate the errors with algorithmic noise-tolerance schemes. They use a prediction-based error-control scheme to improve the performance of the filtering algorithm.
- **Karakonstanis et al. [32]:** This work proposes a undervolting enabled discrete cosine transform architecture to demonstrate higher energy savings. Their architecture puts the long paths for the operations which have less influence on the quality of the final output, so that the impact of low voltage variation sensitivity can be reduced.

2.5.2 Enhancements around Memory

- **Kim et al. [33]:** This work analyzes the bit-level SRAM errors and isolate the contribution of total energy spent in SRAMs in several DNN accelerators. The authors utilize the motivation to present memory adaptive training with in-situ canaries, that enables aggressive voltage scaling of DNN-accelerator weight memories to improve the overall energy-efficiency.
- **DRIS-3 [34]:** This work demonstrates that a significant accuracy loss is caused by

certain bits during faulty DNN operations and using this fault analysis, proposes a fault tolerant reliability improvement scheme— DRIS-3, to mitigate the faults during DNN operations.

- **Chandramoorti et al. [35]:** This work presents a technique of low-voltage neural network acceleration with application-aware SRAM architecture. Undervolting causes errors in SRAMs. They evaluate low voltage SRAM errors specifically for ML applications and incorporate an application aware voltage boosting framework, that runs deep into the SRAM banks, to enhance the overall energy efficiency for ML application.
- **Parana [36]:** This work evaluates thermal issues in a NN accelerator 3D memory and propose a ”3D + 2.5D” integration processor named Parana, which integrates 3D memory and the NPU. Parana tackles the thermal problem by lowering the number of memory accesses and changing the memory access patterns.
- **Salami et al. [37]:** This work performs a thorough analysis of the NN accelerator components and devise a strategy to appropriately mask the MSBs, to recover the corrupted bits, thereby enhancing the efficiency by mitigating the faults.
- **Nguyen et al. [38]:** This work presents innovation in error resilience around DRAM accesses to increase the energy efficiency of DNN applications. The authors exploit that the DNN classification accuracy is not affected equally by all the bits fetched from memory. By studying the trade-off, the authors devise an adaptive DRAM refreshing technique, eliminating unnecessary refresh energy spent on insignificant bits.

2.5.3 Analog/Mixed-Signal Enhancements

- **Eshraghian et al. [39]:** This work for ReRAM based DNN accelerators, utilizes the frequency dependence of *v-i* place hysteresis to relieve the limitation on the single-bit-per-device and allocating the kernel information to the device conductance and partially to the frequency of the time-varying input.
- **BIHIWE [40]:** Ghodrati et al. propose a technique BIHIWE for mixed signal DNN

accelerators, to address the issues in mixed-signal circuitry due to restricted scope of information encoding, noise susceptibility and overheads due to Analog to Digital conversions. BIHIWE, bit-partitions vector dot-product into clusters of low-bitwidth operations executing in parallel and embedded across multiple vector elements.

- **ISAAC [41]:** This work demonstrates a scheme ISAAC, by implementing a pipelined architecture with each neural network layer being dedicated specific crossbars and heaping up the data between pipe stages using eDRAM buffers. ISAAC also proposes a novel data encoding technique to reduce the analog-to-digital conversion overheads and performs a design space inspection to obtain a balance between memristor storage/compute, buffers and ADCs on the chip.
- **Mackin et al. [42]:** This work proposes the usage of crossbar arrays of NVMs to implement MAC operations at the data location and demonstrates simultaneous programming of weights at optimal hardware conditions and exploring its effectiveness under significant NVM variability.

To the best of our knowledge, the work on this dissertation is the first one to exploits the data-driven delay variance in the systolic array of MACs, to predict timing errors in TPUs, operating under near-threshold condition.

2.6 Power Gating Implementations

- **Tschanz et al. [43]:** This work advocated for the need of active leakage control techniques in VLSI. The authors employ dynamic sleep transistors and body bias with clock gating to provide active leakage control on an execution core in 130-nm CMOS technology. They use PMOS sleep transistors, and are able to reducing the power consumption by 8%.
- **Shi et al. [44]:** This work outlines the challenges and opportunities in optimal sleep transistor design in different configurations. Most of the aspects like the design and implementation of header and/or footer switch, the actual distribution of the sleep

transistors, dimensions of the sleep transistor, and possibilities of optimization through bias are discussed in depth.

- **Hu et al. [45]:** This work provides an extensive analytical model of the idleness in CPU. The sections that are to be powergated need to be idle for sufficient number of cycles so that the sleep/wake-up overheads of powergate implementation don't outweigh the benefits of leakage power savings. They show that the floating point units can be powergated for upto 28% of the cycles for a performance loss of 2%.

To the best of our knowledge, the work in the dissertation is the first one in the DNN accelerator domain to explore the severe, yet predictable resource underutilization and propose power-gating strategies to extract a staggering gain in energy efficiency.

CHAPTER 3

RELIABILITY AND UNIFORMITY ENHANCEMENT IN 8T-SRAM PUFs

3.1 Background and Contributions of This Work

SRAM-based PUFs (SPUFs) have emerged as a viable security choice in resource constrained systems [16]. This can be attributed to their obviation for dedicated circuitry and elimination of the overheads of complex encryption mechanisms [6]. Instead, SPUFs rely on inherent physical characteristics, that originate from manufacturing process variations (PV), to enable chip security [7]. Likewise, Near-Threshold Computing (NTC) has transpired as a promising energy-efficient design paradigm, as compared to Super Threshold Computing (STC). SPUFs operating at NTC exhibit quadratic energy gains making them more pervasive among low power systems [46]. For example, battery-operated systems, batteryless RFIDs running on inductive coupling, Internet of Things (IoT) applications, sensor networks, wearable gadgets, low power embedded systems and so on.

However, the supply voltage reduction is also accompanied by increasing effects of PV. Herein, it is important to highlight a unique property of SPUFs to reproduce the same chip signature every time it is attempted to be authenticated, often referred to as SPUF *reliability* [7]. Consequently, it becomes extremely challenging to reliably deploy SPUFs at NTC. Thus, it remains an intriguing research question whether NTC operation of SPUFs brings about any degradation in reliability, due to exacerbated PV sensitivity.

The read instability introduced by low voltage operation thus makes 6T-SRAMs an unfavorable design choice at NTC. Researchers have proposed 8T-SRAM and 10T-SRAM models for sub-threshold computing systems [11, 47]. 8T-SRAMs have been used for NTC operation, in this work, owing to their relatively lower area overhead. The addition of extra read transistors in 8T-SRAM introduces a schematic asymmetry, quite contrary to a symmetrical 6T-SRAM. It is observed that, this leads to an asymmetric start-up current,

which when sensitized by varying system noise and temperature, leads to a degradation in SPUF reliability.

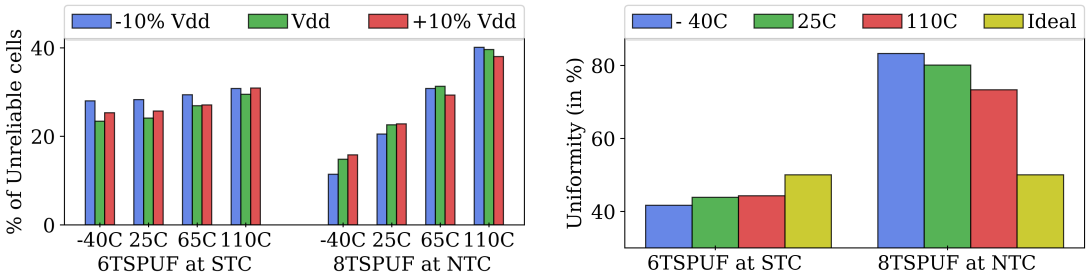
In addition to reliability concerns, this chapter finds that the shift to NTC SPUFs challenges *ideal SPUF uniformity* owing to the differences in device geometry. SPUF uniformity depicts how uniformly the 0's and 1's are distributed in the SPUF signature; with better uniformity corresponding to a more randomized distribution, making it unfathomable for the attacker to recreate [13]. It is observed that the schematic asymmetry exhibited by 8T-SPUFs leads to an imbalanced distribution of the start-up current within the cell, giving rise to decreased uniformity. To preserve the energy efficiency at NTC, this chapter analyzes the impact of device asymmetry on reliability and uniformity of 8T-SPUF and propose *CUBIT: Biasing based strategies* and *CUSIT: Sizing based strategies*.

Our contributions in this chapter are as follows:

- It is observed that there is a marked degradation in reliability and uniformity for 8T-SPUFs operating at NTC, in comparison with STC-operated 6T-SPUFs (Section 3.2).
- By analyzing the impact of device asymmetry on reliability and uniformity characteristics, this chapter proposes CUBIT: biasing based design strategy, and CUSIT: sizing based design strategy (Section 3.3).
- In comparison to state-of-the-art technique by Chang et al. [16], our proposed design strategies exhibit a comprehensive enhancement in both reliability and uniformity, with more than 55% improvement in percentage of unreliable cells, and 82% progression in the ballpark of ideal uniformity over the Baseline NTC 8T-SPUF array (Section 3.4).

3.2 Background and Motivation

In this section, the metrics of reliability (Section 3.2.1) and uniformity (Section 3.2.2), which are key determinants of SPUF behavior, are quantified. Using the methodology discussed in Section 3.2.4, a radical change in these characteristics in an 8T-SPUF operated at NTC (Section 3.2.3) is observed. Further, it is demonstrated that although 8T-SPUFs ex-



(a) *PUC comparison w.r.t. temperature and V_{dd} fluctuations* **(b)** *Uniformity(%) comparison w.r.t. temperature variations*

Fig. 3.1: *Reliability and Uniformity characteristics for STC-operated 6T-SPUF versus NTC-operated 8T-SPUF*

hibit an energy efficient solution [46], their naive implementation on low power platforms come with massive degradation in reliability and uniformity (Section 3.2.5).

3.2.1 Estimating SPUF Reliability

SPUF Reliability is a measure of repeatability of SPUF array signature. The reliability is threatened when the start-up states of cells in the unique signature of SPUF array are flipped by noise and environmental variations, leading to *unreliable cells* [7]. A larger proportion of *unreliable cells* in the SPUF array, indicates a lower SPUF reliability. In Equation (3.1), Bit_Flips gives the number of times an SPUF cell c powers up to a different bit value, relative to the noiseless iteration. n is the number of power-ups of the same SPUF cell in the presence of system noise.

$$Bit\_Flips\ (BF_c) = \sum_{i=0}^{n} |Bit\_value\ _{noiseless} - Bit\_value\ _i| \tag{3.1}$$

$Bit\_Reliable$ is a binary thresholder, determining whether the c-th SPUF cell is reliable. t is the threshold of number of allowed bit flips to still mark the cell as reliable.

$$Bit\_Reliable\ (BR_c) = \begin{cases} 1 & t \leq BF_c = 0 \\ 0 & BF_c > t \end{cases} \tag{3.2}$$

Finally, *PUC*, gives the percentage of unreliable cells in the SPUF with m cells, at supply voltage V and temperature T.

$$PUC(\%\ of\ Unreliable\ Cells\ _{(V,\ T)}) = \frac{1}{m}\sum_{c=0}^{m} BR_c \times 100\% \tag{3.3}$$

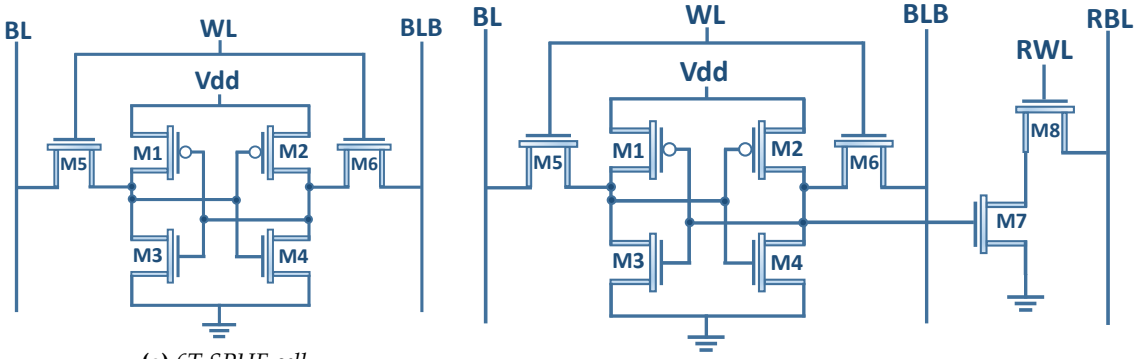
For assessing reliability in this chapter, the values used are t=0, n=10, m=1000.

3.2.2 Estimating SPUF Uniformity

Uniformity depicts the randomness of the SPUF array's signature. Uniform distribution of '1's and '0's in the SPUF signature ensures a strongly random key, which is difficult to be replicated by an attacker [13]. For a m-bit SPUF, Maiti et al. have defined uniformity as the percentage Hamming Weight (HW) of the m-bits [48], given by Equation (3.4).

$$Uniformity\ (\%) = \frac{1}{m}\sum_{c=0}^{m} Bit\_value_c \times 100\% \tag{3.4}$$

where Bit_value_c is the power up state of the cth SPUF cell ('0' or '1') of SPUF with m cells. The ideal SPUF with even distribution of bits '0' and '1' produces 50% uniformity. Hence, a uniformity proximal to 50% translates to more randomness in the SPUF. In uniformity analysis of the skewness of a bit to '0' or '1', Bit_value_c is considered to be '1'('0') if SPUF cell powers to '1'('0') more than 50% of the time.



(a) *6T-SPUF cell* **(b)** *8T-SPUF cell*

Fig. 3.2: *Schematic Representation of a SPUF cell*

3.2.3 Threats to SPUFs at NTC

It is discussed how 8T-SPUFs operating at NTC bring reliability and uniformity concerns in following arguements.

- Although PV is responsible for the generation of a unique SPUF signature, aggravated PV sensitivity at NTC impairs the repeatability of the signature. Under variation in environmental conditions, such as system noise, the inherent skewness of the SPUF cell is overriden, instigating an increase in *PUC* (Equation 3.3).
- 8T-SRAM (Figure 3.2b) used for NTC execution exhibits a schematical asymmetry with respect to a STC-operated 6T-SRAM (Figure 3.2a). The addition of two extra transistors introduces a current sharing on one half of the SRAM cell. The asymmetric current causes the SRAM cell to skew towards an uneven number of 0s or 1s, in a sequence of power-ups, making the uniformity sway further from its ideal value (Equation 3.4).
- The current induced due to asymmetry of 8T-SPUF, experiences further imbalance as a consequence of exacerbated PV sensitivity at NTC. This current variation degrades the reliability of a NTC-SPUF. Hence, it is a fair deduction that reliability and uniformity characteristics of an SPUF undergo cataclysmic changes when operated at NTC.

3.2.4 Methodology

To estimate the reliability and uniformity of an SPUF array, 6T-SRAM and 8T-SRAM cells are modeled using Predictive Technology Model (PTM) for 32nm [49] based on BSIM-CMG [50]. 1000 unique cells are instantiated to build an SPUF array by Monte Carlo simulations for Threshold voltage (V_{th}), Length (L) and Width (W), with PV of 9% for V_{th} and 4.5% for W and L. The noise in time-domain is modeled using all the noise sources defined in BSIM-CMG. For the frequency-domain noise modeling, the base and maximum frequency are set to F_{min}=10^4 and F_{max}=10^9 respectively [51]. The SPUF array are then simulated at supply voltage ranging from -10% V_{dd} to +10% V_{dd} and temperature from -40°C to 110°C.

3.2.5 Results and Significance

Figure 3.1a shows the comparison of *PUC* in STC 6T-SPUF and NTC 8T-SPUF array. It is observed that the PUC variation across V_{dd} is under 5% for both the STC and NTC SPUF. However, the window of variation of PUC across temperatures for NTC 8T-SPUF shoots up to 30%, as opposed to 6% in STC 6T-SPUF. This means that if a key is devised at same temperature, for both STC and NTC SPUFs, and then attempt to reconstruct the response at a higher temperature, the number of unreliable cells is much higher for NTC 8T-SPUF. Fuzzy extractor and other error correcting mechanisms would then have to cover a larger spread of unreliable cells across all corners of environmental variations, which leads to excessive power and area overheads [18]. The increased overheads will disrupt the entire SPUF ecosystem, which is primarily targeted for low cost security primitives. Hence, shifting from STC to NTC, SPUFs subjected to temperature variation, are plagued by decreased reliability.

Figure 3.1b compares the uniformity of STC 6T-SPUF and NTC 8T-SPUF at different temperatures. It is seen that the worst case deviation of 6T-SPUF from ideal uniformity is under 9%, as compared to the glaring deviation of 33.2% for 8T-SPUF. This anomaly can be attributed to the exorbitant skewness of the bits to a particular state for the NTC 8T-SPUF. Therefore, it is imperative to realise that the 8T-SPUFs are plagued by decrement in uniformity, making them more vulnerable to attacks. To recuperate this atrophy in reliability and uniformity characteristics, design strategies are proposed, for SPUFs operated at NTC in Section 3.3.

3.3 Design

In this section, the impact of schematic asymmetry in 8T-SPUFs amalgamated with the increased effect of PV are first discussed, to justify the governing principle of our design (Section 3.3.1). Following which, design strategies are proposed, *CUBIT: **C**urrent S**u**ppression with **Bi**asing **T**echnique* (Section 3.3.2) and *CUSIT: **C**urrent S**u**ppression with **Si**zing **T**echnique* (Section 3.3.3), to tackle the glaring degradation in reliability and uniformity

characteristics of 8T-SPUFs at NTC.

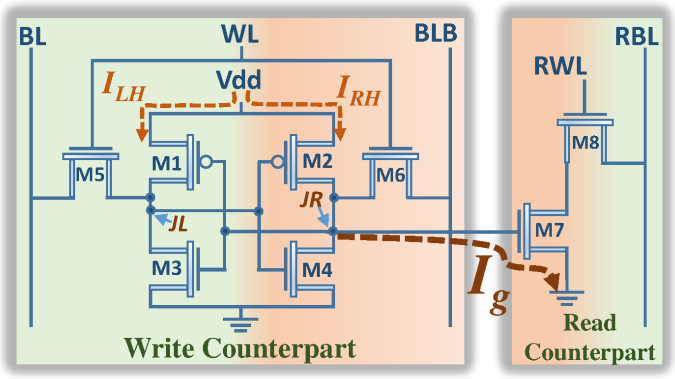
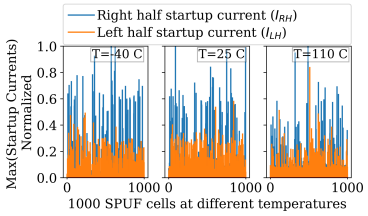
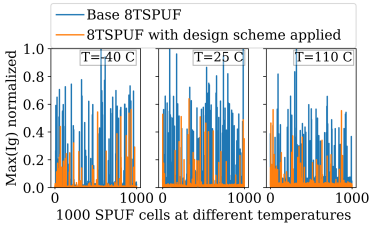


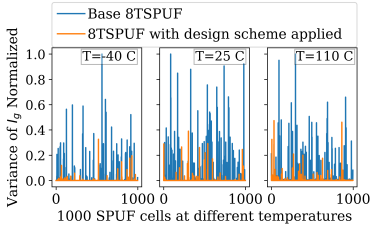
Fig. 3.3: *Current Ig is shared from only right junction* JR *of the 8T-SPUF cell, rendering the current in the right half, I_{RH} asymmetric to left half current I_{LH}.*



(a) *Asymmetric Distribution of startup current*



(b) *Suppression of Ig Magnitude*



(c) *Suppression of Ig Variance*

Fig. 3.4: *Fig. 3.4a : Plot of the maximum supply currents distributed to right and left half of 8T-SPUF cell. Fig. 3.4b : Effective suppression of* Ig *by biasing techniques. Fig. 3.4c : Effective supression of Variance of* Ig *by biasing techniques. Maximum and variance are calculated among the maximum currents until trip point, of 10 different noisy startups*

3.3.1 Impact of Schematic Differences

The schematic difference between a 6T-SPUF and 8T-SPUF cell is the addition of the two NMOS's for read access. Schematically, as the read access transistors are only connected to one half of the cell (Fig. 3.2), there is an asymmetry in right half and left half supply current, I_{RH} and I_{LH} (Fig.3.3).

The maximum of currents, I_{RH}, I_{LH} are compared among 10 different noisy startups,

until the trip point [7], from where the voltages at *BL* and *BLB* diverge to their final states. Figure 3.4a shows that the current I_{RH} dominates the current I_{LH}. This non-uniformity in current distribution among the SPUF cells is brought about by gate leakage current, I_g flowing from right junction *JR* towards the gate of transistor M7 as shown in Figure 3.3. I_g tries to drag down the voltage rise at junction *JR*. This phenomenon leads to more number of SPUF cells ending up to a final states '1' than '0' at *JL*, .i.e degraded uniformity. In addition, it is observed that this current is very sensitive to temperature change and random system noise. Due to this increased sensitivity, the chances of degraded reliability increases manifold. Hence, the suppression of this current and its variation opens doors to better reliability and uniformity in NTC 8T-SPUF.

3.3.2 CUBIT: Biasing based Techniques

CUBIT improves the reliability and uniformity of the NTC 8T-SPUF by biasing Read Counterpart of NTC 8T-SPUF (Fig. 3.3) in different ways, targeting the suppression of current Ig. In Figure 3.4b, the supression of maximum of current Ig in SPUF cells by one of our biasing technique is shown, which improves both reliability and uniformity.

Similarly, in Figure 3.4c, the massive reduction of statistical variance of maximum of current Ig among the noisy startups for SPUF cells for the same technique is shown. Higher magnitude and variance contributes towards decreased reliability and non uniformity respectively. Different biasing techniques are discussed in 3.3.2, as different steps of an algorithm, devised to comprehensively improve both reliability and uniformity.

CUBIT Algorithm

An algorithm for finding the best combination among the different ways of biasing Read Counterpart (Fig. 3.3) is proposed. At the end of the algorithm, a stack of improved uniformity and reliability figures and the respective moves which cause it are achieved. Top of the stack is the best combination according to priority constraints given. User can also select sub optimal solutions as a tradeoff with the overheads in actual implementa-

tion. It is started by applying minimum possible moves first, and then move to different combinations of moves. Table 3.1 lists the terminologies and the objective of our algorithm 1.

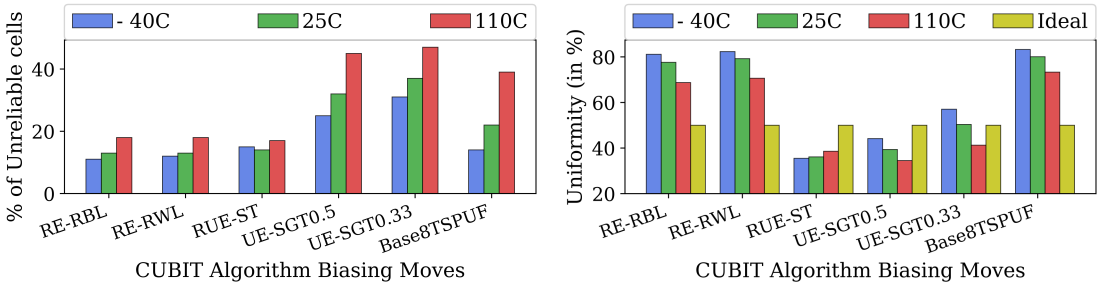
T	{-40°C, 25°C, 110°C}
R	*Max(% of Unreliable Cells (PUC))*, across ***T***
U	*Max(Uniformity-Ideal Uniformity)*, across ***T***
RLT	***R***eliability ***L***oss ***T***hreshold: The maximum allowed Loss of R (in %) for gaining U.
ULT	***U***niformity ***L***oss ***T***hreshold: The maximum allowed Loss of U (in %) for gaining R.
Objective	Minimize **U** and **R**

Table 3.1: *Terminologies and Objective for Algorithm 1*

- ***T***= {-40C, 25C, 110C}
- ***R***= *Max(% of Unreliable Cells (PUC))*, across ***T***.
- ***U***= *Max(Uniformity-Ideal Uniformity)*, across ***T***.
- ***Reliability Loss Threshold (RLT)***: The maximum allowed Loss of R (in %) for gaining U.
- ***Uniformity Loss Threshold (ULT)***: The maximum allowed Loss of U (in %) for gaining R.
- ***Objective:*** Minimize **U** and **R**

CUBIT Moves The moves of the algorithm that bias the Read Counterpart of 8T-SPUF cell in different ways are outlined, targeting the supression of magnitude and/or variation of *Ig*, which are successful in improving in reliability and/or uniformity.

1. **RE-RBL:** This is a **R**eliability **E**nhancer move, where the Read Bit Line (RBL) of the 8T-SPUF cell is biased to Biasing voltage (VB), as shown in Fig. 3.6a. The SPUF array logic can be customized to provide a logic high through precharging in RBL line at the startup of the 8T-SPUF array. Simulation results in Fig. 3.5, show that, with VB=Vdd, this move can decrease ***R*** (Fig. 3.5a), by 55%, but cannot decrease ***U*** (Fig. 3.5b).



(a) *Comparison of Reliability Improvement* **(b)** *Comparison of Uniformity Improvement*

Fig. 3.5**:** *Improvement in Reliability (Fig. 3.5a) and Uniformity (Fig. 3.5b) obtained by different biasing schemes. Individual biasing schemes cannot always address comprehensive improvement in both reliability and uniformity.*

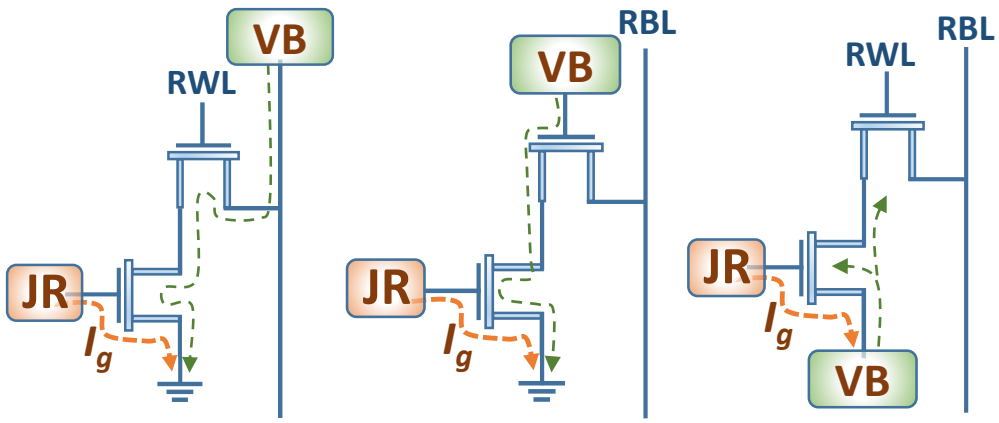
2. **RE-RWL:** This is also a RE move, where the Read Word Line (RWL) of the 8T-SPUF cell is biased to VB as shown in Fig. 3.6b. The SPUF array logic can be customized to enable RWL, also at the startup. Simulation results in Fig. 3.5, show that, with VSB=Vdd, this move can decrease $\boldsymbol{R}$ (Fig. 3.5a), by 53%, but cannot decrease $\boldsymbol{U}$ (Fig. 3.5b).

3. **RUE-SGT** and **UE-SGTf:** Biasing the Source Ground Terminal (SGT) of M7 with voltage VB=Vdd, as shown in Fig. 3.6c, at the startup of the 8T-SPUF cell, gives us Reilability and Uniformity Enhancer (RUE) move. As the sink of the current Ig is SGT, rising its potential from ground supresses the Ig very effectively. Simulation results in Fig. 3.5 show that Bias of VSB=Vdd at SGT is able to reduce $\boldsymbol{R}$ and $\boldsymbol{U}$ by 55.8% and 56% respectively. RUE-SGT improves the uniformity by aggressively turning around the population of '1' skewed cells to from 83% (67% above ideal) to 35% (30% below ideal). Hence, by reducing the degree of supression with lowering VB from Vdd, the uniformity can be brought closer to ideal uniformity. This give us Uniformity Enhancer (UE) moves, UE-SGTf, where f is VB as fraction of cell Vdd (VB/Vdd). Simulation results in Fig. 3.5b shows improvement of $\boldsymbol{U}$ with decrease of VB. However, as the VB goes on decreasing, reliability decreases severely as shown in Fig. 3.5a. Hence, true benefit of UE-SGTfs can be extracted only by coupling with REs.

Terminologies

Below are some terminologies and equations used in the algorithm 1

- The set of RE moves combinations: ***REcombinations***={*RE-RWL, RE-RBL,*{*RE-RWL, RE-RBL*}}
- The set of RUE moves: ***RUEmoves***={*RUE-SGT*}
- The set of UE moves: ***UEmoves***={*UE-SGT1, UE-SGT2, and so on*}
- ***R***= *Max(Percentage of Unreliable Cells (PUC))*, across all temperatures.
- ***U***= *Max(Uniformity-Ideal Uniformity)*, across all temperatures.
- ***Reliability Loss Threshold (RLT)***: is the the maximum allowed Loss of R in percentage for gaining U.
- ***Uniformity Loss Threshold (ULT)***: is the the maximum allowed Loss of U in percentage for gaining R.
- ***Objective:*** Minimize **U** and **R**



(a) *Biasing RBL with VB* **(b)** *Biasing RWL with VB* **(c)** *Elevating GND to VB*

Fig. 3.6**:** *Moves to effectively supress the magnitude and/or variation of the current* Ig, *which bias a voltage* VB *at different terminals of Read Section of 8T-SPUF cell in CUBIT Algorithm.*

Algorithm 1 CUBIT

REcombinations={*RE-RWL, RE-RBL,*{*RE-RWL, RE-RBL*}}
RUEmove={*RUE-SGT*}
UEmoves={*UE-SGTf1, UE-SGTf2,....UE-SGTfn*}
procedure MAIN
    Apply RUE_move
    Push move and (R, U) to stack
    for (r in *REcombinations*) **do**:
        Apply r
        if (RU-IMPROVED(R,U)==1) **then**
            Push r and (R, U) to stack
        end if
     ENHANCE UNIFORMITY
    end for
end procedure

procedure ENHANCE UNIFORMITY
    for u in *UEmoves* **do**:
        Apply u
        if RU-IMPROVED(R,U)==1 **then**
            Push u and (R, U) to stack
        end if
    end for
end procedure

procedure RU-IMPROVED(R_i, U_i)
    $(R_{i-1}, U_{i-1}) \leftarrow Top\ value\ of\ stack$
    $RG \leftarrow \frac{(R_i - R_{i-1})}{R_{i-1}} \times 100\%$
    $UG \leftarrow= \frac{(U_i - U_{i-1})}{U_{i-1}} \times 100\%$
    if $(RG > 0$ **and** $UG > 0)$ **then**
        return 1
    else if $(RG > 0$ **and** $-ULT \leq UG \leq 0)$ **then**
        return 1
    else if $(UG > 0$ **and** $-RLT \leq RG \leq 0)$ **then**
        return 1
    else
        return 0
    end if
end procedure

3.3.3 CUSIT: Sizing based Techniques

CUSIT is a sizing based technique to supress the effects of current I_g, for better reliability and uniformity. CUSIT scales the sizes of the transistors of I_g source (Write counterpart) and sink (Read counterpart) in such a way that I_g, relative to supply current, $I(V_{dd})$ is decreased. This scaling ensures that the impact of I_g on reliability and uniformity is curtailed. Figure 3.7 shows the variation in I_g for upscaling write transistors relative to read transistors, with six different scaling factors. I_g is normalized to $I(V_dd)$ and observed till the time a 8T-SPUF reaches its trip point. It is evident from the figure that I_g decreases with an increase in the upscaling factor. To establish the adaptive nature of CUSIT in the light of varying currents in the transistor, two different approaches to transistor resizing are presented. First, scaling down the size of read transistors (relative to write transistors), and second, scaling up the size of write transistors (relative to read transistors). In light of implementation feasibility, the Read and Write Counterparts of 8T-SPUF cells (Fig. 3.3) can be sized independent of each other, unlike the meticulous sizing constraints of conventional 6T-SPUF cells [52].

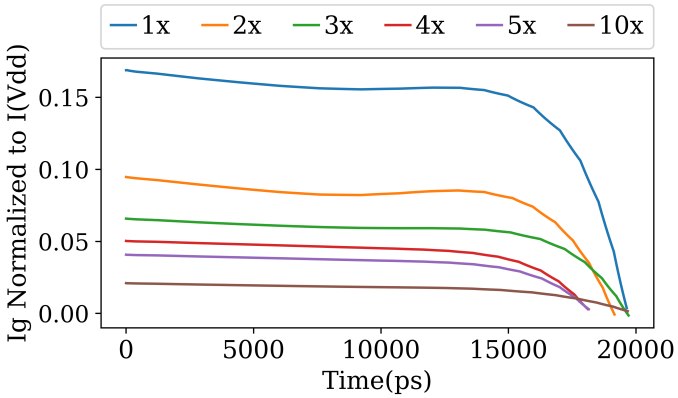


Fig. 3.7: *Supression of normalized current* Ig *with different size upscaling factors of CUSIT*

3.4 Results

In this section, the results for enhancement of reliability and uniformity given by our

design strategies are presented. Section 3.4.1 presents the results obtained by applying CUBIT and Section 3.4.2 presents results obtained by aplying CUSIT, comparing it with a comparative sizing scheme in literature.

3.4.1 CUBIT Results

In this section, the results of applying CUBIT algorithm are presented and analyzed. Table 3.2 shows the final combination of RE/RUE/UE moves which minimize the (R,U) with different priority constraints. It is possible to obtain an excellent enhancement of more than (51%, 76%) in (R, U) across the Baseline NTC 8T-SPUF across all priority constraints. It is observed that the optimal combination of biasing techniques for best Reliability and Uniformity is a trade-off. If Reliability (Uniformity) is favored, 55.8% (82.86%) enhancement in R(U) is achieved. Also, even by only using a single voltage source which eliminates the need of additional low voltage source, (54.9%, 76.8%) enhancement in (R, U) is got as compared to Baseline 8T-SPUF.

Priority	**RLT**	**ULT**	**Bias Set**	**PUC(%)**			**Uniformity(%)**			**Enhancement**
				-40°C	25°C	110°C	-40°C	25°C	110°C	**(R, U) (in %)**
Reliability Favoring	5%	10%	{RE-RBL, RE-RWL, UE-SGT0.17}	17.4	16.7	17.5	49.09	47.16	43.77	(55.8%, 81.35%)
Uniformity Favoring	10%	5%	{RE-RBL, UE-SGT0.33}	18.8	18.9	18.4	56.50	51.21	44.27	(52.27%, 82.86%)
Single Voltage Source	10%	10%	{RE-RWL, RUE-SGT}	17.8	16.2	17.9	42.29	42.84	44.17	(54.79%, 76.8%)

Table 3.2: *Different (R, U) enhancements over base line NTC 8TSPUF, obtained from application of different priority constraints in the CUBIT algorithm.*

Comparative Schemes	**Size Upscaling**	**(Area,Power)**	**PUC(%)**			**Uniformity(%)**			**Enhancement**
			-40°C	25°C	110°C	-40°C	25°C	110°C	**(R, U) (in %)**
CUSIT	Baseline	(1x,1x)	14.8	22.6	39.6	83.27	80.10	73.32	(-, -)
	Relative X2	(1.75x, 2.12x)	13.3	19.0	30.0	71.02	67.67	63.50	(24.24, 36.79)
	Relative X3	(2.5x, 3.23)	13.0	17.7	24.4	65.11	63.04	60.24	(38.38, 54.56)
	Relative X4	(3.25x, 4.31x)	12.0	15.6	21.1	61.70	60.52	57.89	(46.71, 64.80)
	Relative X5	(4x, 5.45x)	10.7	14.4	19.5	59.09	58.56	56.09	(50.75, 72.67)
VCTS	Baseline	(1x,1x)	14.8	22.6	39.6	83.27	80.10	73.32	(-, -)
	Holistic X2	(2x, 2.15x)	9.9	15.0	27.1	82.76	79.31	72.00	(31.57, 1.51)
	Holistic X3	(3x, 3.26x)	8.4	12.7	21.8	82.88	79.09	71.50	(44.95, 1.16)
	Holistic X4	(4x, 4.38x)	6.7	11.2	17.3	82.76	79.29	71.14	(56.31, 1.50)
	Holistic X5	(5x, 5.51x)	6.5	10.4	16.3	82.53	78.67	70.79	(58.83, 2.21)

Table 3.3: *Comparative analysis of Enhancement of Reliability and Uniformity*

3.4.2 CUSIT Results

In this section, the improvement in reliability obtained with CUSIT is presented . As a sizing technique for 8T-SPUF's quality improvement in literature is not available, the comparison is of CUSIT is made with a sizing technique [16] demonstrated for 6T-SPUFs which holistically upscales the size of all transistors of SPUF cell targeting Vth variation (**VCTS**-**V**th **C**entric **T**ransistor **S**izing). For CUSIT, the size of transistors in write counterpart relative to the size of transistors in read counterpart is upscaled, targeting supression of current *Ig*. Table 3.3 shows that reliability improvement with size upscaling is similar in both techniques, achieving improvement in *PUC* by more than 50% when upscaling factor reaches 5. However, CUSIT vastly outperforms VCTS in terms of comprehensive enhancement of both reliability and uniformity as CUSIT achieves more than 72% enhancement in proximity to ideal uniformity, compared to an insignificant 3% improvement by VCTS. VCTS cannot achieve uniformity improvement because, holistic scaling of all the transistors with a common factor can't reduce the effect of asymmetric current *Ig*. Hence, it is imperative to deduce that CUSIT is a superior comprehensive technique than VCTS for NTC 8T-SPUFs.

3.4.3 Overhead Analysis

The techniques in CUBIT can be realized in SPUF array at the granularity of array logic and column. Like most of other circuit level techniques [16], area and power overheads of CUBIT relative to the entire SPUF array is insignificant because the area and power consumed by the SPUF cells hugely dominates the resources consumed by SPUF's array logic [53] [16].

For CUSIT, it is observed that CUSIT's relative sizing over VCTS's holistic sizing achieves linear savings in transistor's active area and power consumtion with size upscaling factors (Table 3.3). Although overheads in CUBIT inrease linearly with upscaling factors (Table 3.3), they can be amortized with respect to conventional 6T-SPUFs by adopting future technology nodes in design. This is because the size of 8T-SPUFs can be smaller

than 6T-SPUFs along the future technology nodes, as 6T-SPUF's further shrinkability with future technology nodes is already limited by their cell stability issues [54, 55].

CHAPTER 4

IMPROVING PERFORMANCE OF A NEAR-THRESHOLD TENSOR PROCESSING UNIT WITH TIMING ERROR RESILIENCE

4.1 Background and Contributions of This Work

The cessation of Dennard's scaling, accompanied with the diminishing throughput from the growing number of on-chip cores, has led to the adoption of power-efficient domain-specific architectures. With the recent confluence of artificial intelligence (AI) and high performance computing, the domain-specific computing paradigm is already on the uprise, as evident by the success of the deep neural-network (DNN) accelerators [24,25,33,56]. Among the multitude of such ad-hoc AI architectures, the Google Tensor Processing Unit (TPU) is at the forefront, claiming 15× - 30× faster inference, compared to the top of the line CPUs and GPUs [57]. However, the unprecedented growth in the DNN workloads (e.g., speech recognition in Google Assistant [57,58]) portends a rapid increase in the overall power consumption of the Google data-centers. With a view to heavily curtailing the power consumption while sustaining a high inference accuracy, we envision a near-threshold (NTC) operation of the TPUs. However, operating a TPU at the NTC condition, can significantly dwindle the inference accuracy due to a high rate of timing errors [21,46]. This chapter aims to exploit the inherent architectural artifacts of the TPUs, to predict and tackle the timing errors at NTC, thus promoting a reliable and energy-efficient low-power TPU design paradigm.

The high delay sensitivity to voltage and process variation (PV) at NTC necessitates a relaxed clock constraint to ensure an error-free execution. On the other hand, hardware accelerators like TPUs are designed to offer a high throughput in niche applications. So, in order to embrace the NTC design paradigm for TPUs, it is needed to adopt a better-than-worst case design strategy that can efficiently tolerate the timing errors in its systolic array

architecture (Section 4.2.1). Prior research efforts delve into the challenges and solutions of tackling timing errors in conventional CPU and contemporary TPU architectures [21, 59]. Next, it is discussed why such existing techniques are not effective in an NTC TPU.

Razor—one of the most popular timing error detection and recovery schemes—employs a double sampling flip-flop to detect timing violations inside a pipeline stage [59]. The erring instruction is replayed at a reduced clock frequency to prevent a subsequent timing error. Adopting Razor in TPUs will negatively impact the performance, as the global timing error rate rapidly grows with the dimension of the systolic array. Hence, any recovery penalty, associated with correcting an erroneous computation, will significantly bloat the execution time of the inference. Zhang et al. have recently proposed TE-Drop—where an erring multiplier-and-accumulator (MAC) in a TPU, steals a clock cycle from its downstream MAC to correct the error, and bypasses the downstream MAC's update [21]. However, this approach cannot tackle any timing error in the last row of MACs, without incurring a significant performance penalty at NTC. As the partial sums grow towards the bottom of the systolic array, the impact of timing errors in the last row of MACs is the most crucial. Moreover, as the rate of the timing error increases significantly at NTC, bypassing the update of some MACs will greatly diminish the inference accuracy.

In the light of such shortcomings of the existing timing error mitigation techniques, we propose a novel *timing error prediction strategy*, exploiting the wavefront propagation of the data in a TPU systolic array. It is observed that only few activation sequences are more likely to cause timing violations in the MACs (Section 4.2.3). As the activation data streams through all the MACs in a row, an error-causing activation sequence, can serve as an excellent predictor to avoid subsequent errors in the rest of the MACs in the same row. This early error prediction scheme is combined with a low-complexity voltage boosting mechanism to propose GreenTPU—a new frontier in the design of reliable and low-power TPU. Following are the key contributions of our work:

- It is observed that only few input data sequences cause timing violations in MACs. Consequently, they serve as an efficient predictor for impending timing errors (Section

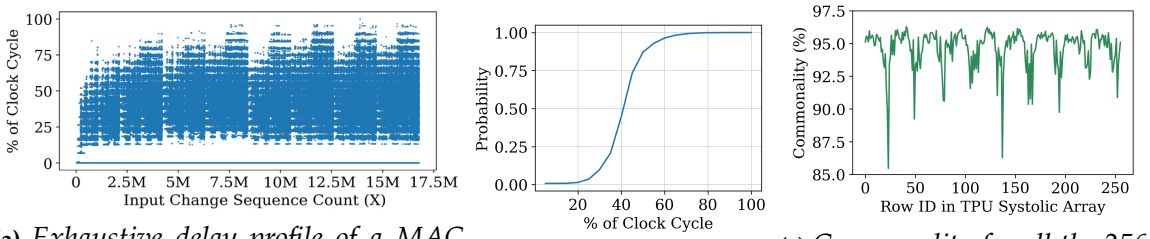
4.2).

- A heuristic to group several input sequences with similar delay characteristics into a family in order to predict future timing errors in a hardware-efficient manner is proposed. (Section 4.3.2).
- GreenTPU—a low-overhead NTC TPU design paradigm is proposed that predicts impending timing violations in its systolic array, and precludes them using a novel voltage boosting mechanism (Section 4.3).
- Combining with our in-house statistical timing analyzer tool, a TPU systolic array simulator in C++ is developed. An end-to-end integration is supported, by interfacing our simulator with Keras [60], so as to closely emulate a real-life TPU-accelerated inference eco-system for contemporary DNN applications (Section 4.4).
- It is demonstrated that GreenTPU provides two orders of magnitude reduction in timing errors at NTC, with respect to TE-Drop [21]—a cutting-edge timing error mitigation technique for TPUs (Section 4.5).
- Compared to TE-Drop, GreenTPU offers **2X–3X** higher performance (TOPS) in an NTC TPU, in 7 out of 8 DNN datasets, with only 3% average loss in the inference accuracy. Estimated from synthesis, place and route of a TPU systolic array RTL, augmented with GreenTPU, the area, power, and wire-length overheads are found to be ∼1.8%, ∼2.2%, and ∼4.1%, respectively (Section 4.5).

4.2 Motivation

In this section, the opportunity of employing a predictive mechanism to tackle timing errors in NTC TPUs is demonstrated. Section 4.2.1 provides a background on the TPU systolic array. Using a cross-layer methodology (Section 4.2.2), the data-driven delay variance in the systolic array of MAC units are analyzed (Section 4.2.3), and motivate the need for a timing error prediction scheme in NTC TPUs (Section 4.2.4).

4.2.1 Background



(a) *Exhaustive delay profile of a MAC unit.*

(b) *CDF of the delay profile.*

(c) *Commonality for all the 256 rows.*

Fig. 4.1: *Figure 4.1a shows the plot of the sensitization delays for all possible weights and input changes for a MAC unit. The variance in the input data can bring about ample delay variance. However, there are only few input sequences that can sensitize the longest delay paths, as depicted by the CDF plot in Figure 4.1b. Figure 4.1c exhibits a very high % of Commonality (Equation 4.2) in the error causing input sequences for all the rows, during the inference of the MNIST dataset.*

TPU Systolic Array

Matrix multiplication is the most expensive operation in the *inference* phase of the DNN applications. The usage of the systolic array of MAC units, has been recognized as a promising direction to accelerate the matrix multiplication. TPU—a DNN accelerator—employs a 256×256 systolic array of MAC units, to multiply the weight matrix with the activation (also referred to as *input*) matrix, maintaining a precision of 8-bit integer [57]. The weights are pre-loaded into the MACs. The activations stream from the left to the right columns of the array at successive clock cycles. The partial sums from the rows of MACs move downstream. Unlike CPUs and GPUs, *a TPU boasts a distinctly homogeneous architecture with a highly predictable data-flow pattern*.

Hazards and Opportunities of NTC TPUs

Operating a TPU at the NTC condition ideally contributes to a quadratic saving in the energy consumption. However, the performance of the TPU heavily declines due to a large delay experienced by the circuits at an NTC voltage [46]. Moreover, a high delay sensitivity to PV and voltage variation at NTC, demands the clock frequency to be heavily relaxed, compared to a super-threshold operation. Hence, in order to operate with an aggressive clock constraint at NTC, a TPU needs to efficiently tolerate a high rate of timing violations.

Furthermore, due to a very deep pipelined architecture of the systolic array (Section 4.2.1), even a small rate of timing error aids to a severe drop in the inference accuracy of the DNN applications [21].

Fortunately, the architectural homogeneity and a predictable data-flow pattern in TPUs, offer a unique opportunity to efficiently tackle timing errors at NTC. Owing to a fixed 8-bit precision in the arithmetic operations, a finite state space of different sensitized path delays, experienced by the MAC units is achieved. *Isolating the subset of the relatively high delays, and correlating that subset with the concerned data patterns, can facilitate the prediction of the impending timing errors in the TPU systolic array*.

4.2.2 Methodology

A MAC unit is synthesized at an NTC operating condition (Section 4.5), by using the 15-nm FinFET library from NanGate [61]. In-house statistical timing analysis tool is employed to study the delay distributions of the sensitized paths for different inputs to the MAC unit. For a conservative estimate, PV-induced delays, obtained from VARIUS-NTV [62] are considered, in randomly chosen 2% of the gates in the MAC circuit [63]. Our cross-layer methodology is further elaborated in Section 4.4.

4.2.3 Results and Significance

The multiplier block of a MAC unit has a relatively deeper logic depth, compared to the accumulator. Hence, the delay distribution of the MAC is modeled, as a function of the change in inputs to the multiplier, i.e., the activation sequence, and the weight. An exhaustive set of 8-bit activation sequences is created for all possible 8-bit weights, leading to a total of 16,777,216 unique combinations.

Figure 4.1a shows the delay profile of a PV-affected MAC unit at NTC, obtained by providing all the aforementioned combinations of weights and input changes. A value of X in Figure 4.1a, corresponds to a specific input change sequence, for a specific weight W,

as expressed in equation 4.1.

$$\begin{aligned} Weight\,(W) \;&=\; \lfloor \frac{X}{65536} \rfloor,\; S = X \bmod 65536 \\ Input\ change &: \lfloor \frac{S}{256} \rfloor \rightarrow (S \bmod 256) \end{aligned} \tag{4.1}$$

The delay profile shows ample variation, resulting from the variance in the input data. This delay variation is statistically shown as a CDF plot of the delay values in Figure 4.1b, where the maximum delay to be the clock period is conservatively attributed. The key observations from Figure 4.1a and 4.1b are: (a) no paths are sensitized when the same activation sequence is applied in two consecutive cycles, and (b) a majority of the multiplication operations sensitize paths with low delays. For instance, it is noticed that the set of delays with more than 60% of the clock cycle is only 3.6% of the entire state space of delays. This sparse sensitization of the higher delay paths, eases the prediction of the recurring timing errors from the same input sequence. Next, the insight to our proposed design of GreenTPU is discussed.

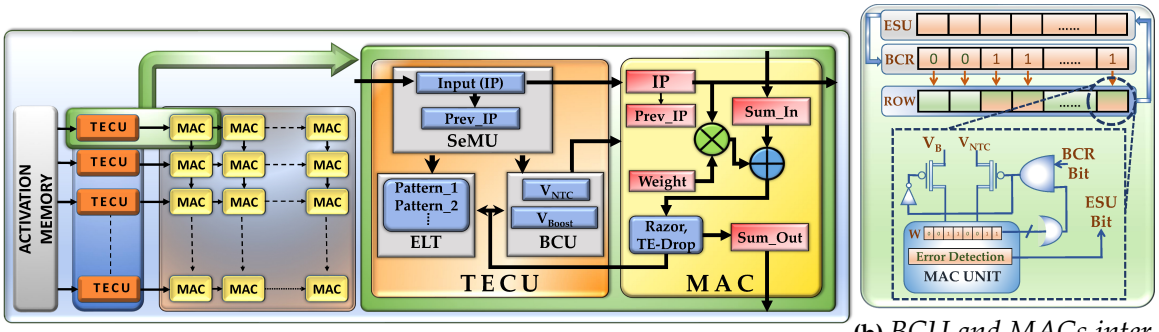
4.2.4 Timing Error Prediction in TPUs

It is aimed to systematically study the likelihood of an error-causing input sequence in a MAC, to produce timing errors in the subsequent MACs, belonging to the same row. In this pursuit, a *Commonality* metric is proposed in Equation 4.2.

$$Commonality_i(\%) = 100 * \left(1 - \frac{\bigcup_{j=0}^{255} UES_j}{\sum_{j=0}^{255} UES_j} \right) \tag{4.2}$$

where, UES_j is the set of unique input sequences that cause timing errors in the j^{th} MAC unit of the i^{th} row.

Figure 4.1c shows a plot of the *Commonality*(%), measured across all the 256 rows during the inference of 1000 test inputs of the MNIST dataset. It is observed that, for all the rows, the commonality of the error-causing input sequences is more than 85%. This result



(a) *A conceptual block diagram of GreenTPU.*

(b) *BCU and MACs interaction.*

Fig. 4.2: *Figure 4.2a shows that the TECUs are pipelined between the activation memory and the rows of the systolic array of MACs. A timing error inside a MAC unit is detected and tackled using Razor and TE-Drop techniques, respectively. A TECU comprises an ELT, an SeMU, and a BCU. ELT stores the error-causing input patterns. SeMU, on the other hand, monitors the input data stream and queries the ELT, to identify potential error-causing input sequences. The BCU (Figure 4.2b), comprising two 256-bit registers—ESU and BCR—prevents future timing errors by boosting the operating voltage of the MACs in a row.*

indicates a *landslide effect* of timing errors in the systolic array of a TPU. In other words, if an input sequence causes a timing error in a MAC unit, that sequence is very likely to cause timing errors in the subsequent MACs, until the sequence is alive in the row. Hence, predicting errors based on the input sequences, and adopting a row-wise control strategy can greatly reduce the number of timing errors in a TPU. With this insight, GreenTPU—our proposed energy-efficient TPU systolic array design, for a near-threshold operation is discussed next.

4.3 GreenTPU

GreenTPU is a novel low-power TPU design paradigm, that dynamically predicts and tackles timing errors in the systolic array of MAC units. Section 4.3.1 outlines the design overview. The details of the components of GreenTPU are elaborated in Section 4.3.2 through 4.3.5.

4.3.1 Design Overview

Figure 4.2a depicts the top-level design overview of GreenTPU. The heart of GreenTPU is the Timing Error Control Unit (TECU). TECU is responsible for predicting and preventing timing errors in the MAC units. In order to maintain a low-complexity circuit design while incurring a negligible performance overhead, one TECU per row of MACs is dedicated, pipelined between the activation memory and the systolic array. A TECU has three main components, viz., Error Log Table (ELT), Sequence Monitor Unit (SeMU), and Boost Control Unit (BCU). When a timing error occurs in any MAC unit of a row, the ELT logs the timing error causing input sequence pattern. Simultaneously, the BCU is alerted to boost the operating voltage of the subsequent MACs in the row, in order to prevent any future timing error. The SeMU monitors the sequence of inputs, and tries to find a matching family representing the pattern in the ELT in every clock cycle. If a match is found, SeMU communicates with the BCU to preclude future timing errors in all the MAC units of a row.

4.3.2 Heuristic for Determining Input Sequence Family

As timing error prediction lies at the heart of GreenTPU design, an efficient heuristic for storing and matching the input sequences responsible for producing higher delays is formulated.

It is observed that the input sequences with similar delay characteristics can be grouped into a family. The input sequences within a family have similar characteristics of bit flips among them, which are responsible to produce delays that are close to each other. Storing families of the high delay causing sequences, rather than storing each sequence as a different entry is thus a hardware efficient strategy to realize our prediction based design.

The correlation between the input sequences and the delays is analyzed to group several input sequences into families. The changes of bits in an input sequence are divided into three different groups of bit changes with their respective contribution in producing specific delay or its vicinity. a) Dynamic bit positions, which have the highest domination and are required to flip, b) Static bits positions, which are required to remain static and not

flip, c) Insignificant bits positions, whose flipping is insignificant. One input sequence can thus virtually represent a family of numerous input sequences that produce similar delays, by virtue of different combinations of bits in Dynamic and Insignificant bit positions.

Algorithm 2 methodizes the hardware inexpensive heuristic to store and match the input sequences. In our heuristic, storage of a new input sequence as a family is done by its bit wise XOR, which can inherently reflect dynamic and static bits positions. Matching is the process of determining if there exists a family in the stored entries, which can encapsulate the input sequence under consideration. Search and incorporation of the insignificant bits is done by loosening the static bit positions by a threshold (Line 1 in Algorithm 2). Then the domination of dynamic bit positions in the contribution towards the delay is maintained (Line 9 of Algorithm 2). Consequently, a match is declared, if an entry is found whose static bit positions are same in more than a threshold percentage of positions with the input sequence under consideration (Line 6-16 in Algorithm 2). The threshold serves as a trade-off between the storage efficiency and the grouping efficiency of the timing error prone input sequences. Lower threshold enables a family to represent higher number of input sequences, but decreases the grouping efficiency, leading to unwanted voltage boosting. Similarly, a higher threshold more accurately groups the error causing sequences, while storing more family entries.

4.3.3 Error Log Table (ELT)

ELT is a look-up table which stores the patterns of the input sequence that lead to timing errors in a MAC unit. A timing error in each MAC unit is sensed using a double-sampling flip-flop at the output, similar to Razor [59]. An erroneous computation is prevented from the timing error by employing TE-Drop [21], where the errant MAC steals a clock cycle from its downstream MAC to correctly finish its own update. Each MAC unit is augmented with the capability to store the previous clock cycle's activation input, thus enabling it to infer the input sequence responsible for the timing error. The sequence is then stored as an 8-bit family, as per the STORE procedure of Algorithm 2 (line 2 to 4) in

Algorithm 2 Pattern Storing/Matching Heuristic

```
1:  TH ← pattern_match_threshold
2:  procedure STORE(current_activ, previous_activ)
3:      xor_pattern ← current_activ ⊕ previous_activ
4:      Store(xor_pattern)
5:  end procedure
6:  procedure MATCH(current_activ, previous_activ)
7:      new_pattern ← current_activ ⊕ previous_activ
8:      for all saved_pat ∈ saved_patterns do
9:          similarity ← saved_pat | new_pattern
10:         num_zeros_sim ← num_reset_bits(saved_pat)
11:         num_zeros_new ← num_reset_bits(similarity)
12:         if num_zeros_new > ⌊TH × num_zeros_sim⌋ then
13:             return match_found
14:         end if
15:     end for
16: end procedure
```

the ELT, while the correct output is being computed parallelly. Also, the BCU is signalled with the errant MAC unit's position in the row, to prevent further timing errors in the MAC units, located to the right of the errant MAC. The ELT is implemented as a content addressable memory that enables a fast lookup. When the ELT is full, a pseudo-LRU-based eviction policy is used (not shown in Algorithm 2) to replace an existing pattern with the new incoming pattern. The size of the ELT is a trade-off between the hardware overhead and prediction accuracy, which is discussed in Section 4.5.

4.3.4 Sequence Monitor Unit (SeMU)

SeMU identifies the possibility of a recurring timing error. The input activation data, coming to each row, is intercepted by SeMU, as the TECU is placed in pipeline between the activation memory and the systolic array. For a given activation sequence coming from the activation memory, SeMU checks if a corresponding family for that sequence is already present in the ELT, as per the MATCH procedure of Algorithm 2 (line 6-16). If a match is found, the BCU is alerted to boost the operating voltage of some of the MACs in the row (Section 4.3.5). This action is taken in order to prevent the timing errors that would have been caused by the input sequence. Due to its pipelined architecture, SeMU adds a

negligible performance overhead.

Algorithm 3 BCU Algorithm

```
1: resolution ← no_of_bits_to_set
2: BCR ← Boost_Control_Register
3: ESU ← Error_Sensing_Unit
4: procedure BOOST_REACTIVE(ESU_mac_i)
5:     start ← mac_i + 1
6:     while start < 256 do
7:         wait_for_clock_cycles(resolution)
8:         if start ≠ mac_i then
9:             BCR[start − resolution ... start − 1] ← 0
10:        end if
11:        BCR[start ... min(start + resolution − 1, 255)] ← 1
12:        start ← start + resolution
13:    end while
14: end procedure
15: procedure BOOST_PROACTIVE(semu_signal)
16:    start ← 0
17:    while start < 256 do
18:        wait_for_clock_cycles(resolution)
19:        if start ≠ 0 then
20:            BCR[start − resolution ... start − 1] ← 0
21:        end if
22:        BCR[start ... start + resolution − 1] ← 1
23:        start ← start + resolution
24:    end while
25: end procedure
```

4.3.5 Boost Control Unit (BCU)

BCU is responsible for boosting the operating voltage of the MAC units, in order to prevent timing errors. As shown in the Figure 4.2b, a BCU houses two 256-bit registers: Boost Control Register (BCR) and Error Sensing Unit (ESU). Each bit of these registers corresponds to each MAC unit in a row. Timing error in a MAC is reflected by the setting of the corresponding bit in ESU. The boosting technique proposed in [64] is adopted, where every MAC unit has access to two voltage rails, V_{NTC} and V_B, representing a near-threshold and a boost voltage, respectively. The reset (set) value in any bit of the BCR, indicates the

corresponding MAC unit to operate with the V_{NTC} (V_B) voltage. In our experiments, V_{NTC} and V_B are set to 0.45V and 0.65V, respectively. Employing the transition infrastructure of [64], it is noticed that the switching between V_{NTC} and V_B can be performed within one clock cycle of the NTC TPU. Also, It is observed that if the pre-loaded weight of a MAC unit is zero, it is unlikely to encounter a timing error. Hence, a MAC with weight zero can disable the voltage boost for itself, to conserve energy.

The boost control procedures are illustrated in Algorithm 3. Whenever a timing error occurs in any MAC unit (ESU_{mac_i}), a certain number of bits of BCR (indicated as *resolution* (Line 1)) that are located to the right of that position, are periodically set (Line 11) and unset (Line 9) as per BOOST_REACTIVE procedure (Line 4). One bit in the resolution also reflects one clock cycle to wait for the next boost period (Line 7). As a result, the MAC units, specific to those set bits in the BCR, will be boosted in the subsequent cycles, precluding any probable timing violations.

On the other hand, if the SeMU (Section 4.3.4) sends a signal (*semu_signal*) as a result of finding an errant pattern, BCU starts to periodically boost the entire row, starting from the column 0. Boosting of the MAC units will happen periodically in response of the set bits in BCR. Again, a certain number of BCR bits are set at a time, defined by *resolution* as per the BOOST_PROACTIVE procedure (Line 15). The *resolution* is empirically ascertained. The choice is guided by the energy budget of the GreenTPU implementation, and the noise margin of the TPU systolic array at NTC, so as to trade-off between the tolerable timing errors, and the energy overhead.

4.3.6 GreenTPU Variants

Three different variants of GreenTPU are outlined to better understand the implication of different architectural artifacts of GreenTPU design.

GreenTPU

GreenTPU is the variant, which includes every details of the architecture discussed

so far. It includes a full fledge predictive engine, which can store many error causing input sequence families to facilitate prediction of any imminent timing errors from the input sequences represented by those families, as defined by pattern-match threshold in Algorithm 2. This variant helps us to better understand the efficacy of a full blown predictive appraoch for maintaining the DNN accuracy in a high timing error prone environment.

GreenTPU Reactive (GTR)

GTR is a variant without any form of predictive capabilities of GreenTPU design paradigm. When the timing error in a MAC unit is detected as a result of an error causing sequence, the MAC units to the right in the row are prevented from potential timing errors from the same sequence. Architecturally, GTR omits SeMU and ELT altogether and only uses BOOST_REACTIVE procedure (Line 3 of Algorithm 3) for BCU. GTR helps to better understand the extent of efficacy than can be provided by a purely reactive approach in maintaining DNN accuracy in a high timing error prone environment.

GreenTPU Lite (GTL)

GTL is a variant which introduces a hint of predictiveness over GTR. It does so by including the minimum possible predictive engine for the most basic prediction scheme. Only one error causing input sequence is used as the basis of prediction and that entry is constantly replaced on the introduction of a new timing error. Hence, architecturally, the entire ELT is replaced by a 8-bit register to store the XOR of error causing sequence. The pattern-match threshold in Algorithm 2 is set to 100% to simplify the prediction scheme, thereby flagging a match only upon an exact match between stored XOR pattern and the XOR pattern of the incoming input sequence. GTL helps to better understand the efficacy added by basic introduction of predictiveness over a reactive timing error correction scheme for DNN accelerators.

4.4 Methodology

In this section, the extensive cross-layer methodology, used to implement and evaluate GreenTPU variants is explained.

4.4.1 Device Layer

The NTC energy consumptions are estimated by performing HSPICE simulations on the basic logic gates (viz., NAND, NOR and Inverter). The 31-stage FO4 inverter-chain is used as a representative of various combinational logics in a TPU. The simulation parameters are obtained from the 16-nm Predictive Technology Model [65]. The impact of the PV at NTC is incorporated using the VARIUS-NTV [66] model. The FinFET characteristics are obtained using the VARIUS-TC model [67]. The delays of the basic gates are used in the circuit layer (Section 4.4.2) to ascertain the sensitized path delays in a MAC.

4.4.2 Circuit Layer

The Verilog RTL description of a systolic array are developed, and augment it with the GreenTPU components. The RTLs are synthesized using the Synopsys Design Compiler, at various operating conditions. Place and route of the synthesized netlist is performed using Cadence SoC Encounter, and estimate the area, power, and wirelength overheads at the NTC operating condition. Using both synthetically generated, as well as, real dataset driven inputs, the sensitized path delays in the MAC array are obtained with our in-house statistical timing analysis (STA) tool. Based on a library of the delay files of the basic logic gates at different operating voltages, the STA tool reports the delays of the sensitized paths of the MAC circuit.

4.4.3 Architecture Layer

Based on the architectural description detailed in [57], a cycle-accurate TPU systolic array simulator—TPU-Sim— is developed in C++, and the GreenTPU components are implemented in TPU-Sim. The STA tool (Section 4.4.2) is integrated with TPU-Sim, to accurately model timing errors in the MACs, based on real data-driven sensitized path delays.

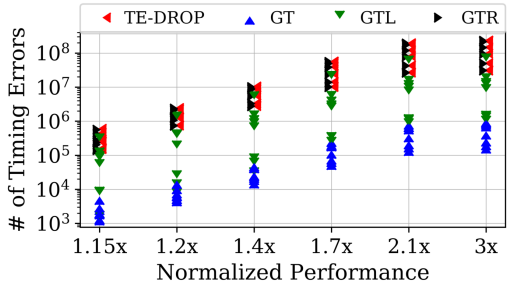


Fig. 4.3: *Number of timing errors encountered in different comparative schemes across 8 DNN datasets.*

A real TPU-based inference eco-system is created by conjoining TPU-Sim with Keras [60]. First, several DNN applications (viz., MNIST [68], Reuters [69], CIFAR-10 [70], IMDB [71], SVHN [72], GTSRB [73], FMNIST [74], FSDD [75]) are trained using Keras, running TensorFlow in the back-end. Each layer's activation inputs and trained model weights are extracted, and pre-processed them into multiple 256×256 8-bit-integer matrices. TPU-Sim is invoked with each pair of the pre-processed input and weight matrices. The output matrices from the TPU-Sim are combined to evaluate the inference accuracy. The framework for handling large amount of test data is parallelized using Python Multiprocessing.

4.5 Experimental Results

In this section, the efficacy of different timing error-resilient schemes, when a TPU operates at a better-than-worst-case scenario are evaluated. Our baseline NTC operating condition (0.45V, 67MHz) guarantees an error-free execution of the TPU. Section 4.5.1 describes the comparative schemes. Section 4.5.2 elaborates the timing error resilience of the schemes. Section 4.5.3 presents the inference accuracy and the energy consumption of the TPU under different schemes. Finally, Section 4.5.4 discusses the hardware overheads of GreenTPU.

4.5.1 Comparative Schemes

- **TE-Drop (TD):** This is a recently proposed technique that can tackle timing errors in the systolic array of a TPU [21]. The errant MAC steals the next clock cycle from its

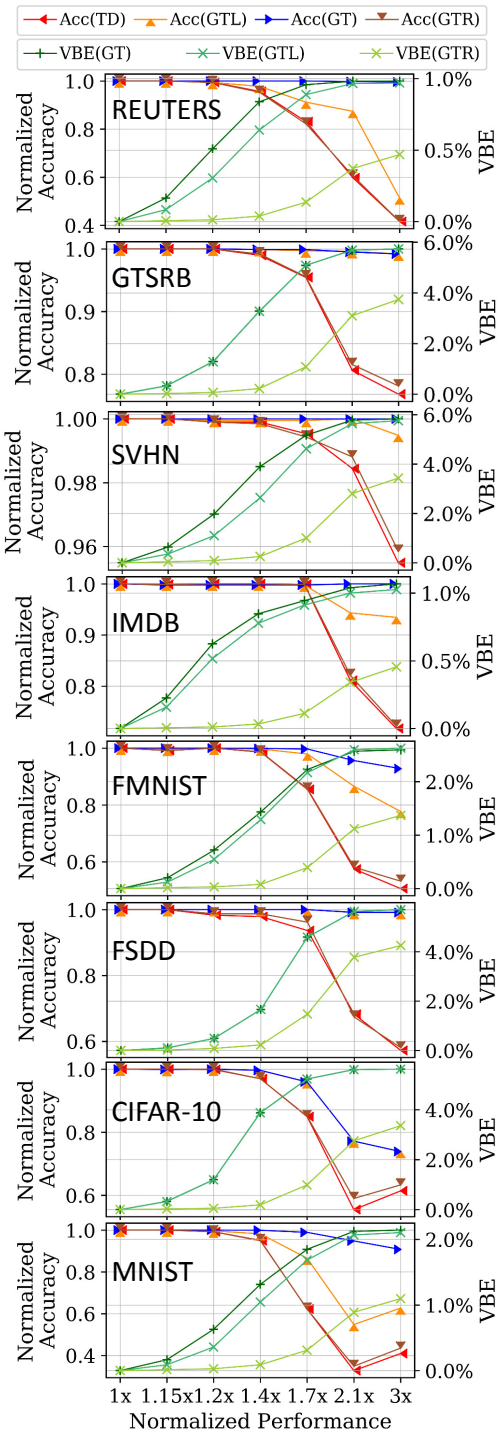


Fig. 4.4: *Normalized inference accuracy (Acc), and voltage boost energy (VBE) from the comparative schemes, at different normalized performance levels, across 8 DNN datasets.*

downstream MAC to correct the error, while the downstream MAC bypasses its own operation.

- **GreenTPU (GT):** This is our proposed design strategy that stores the error-causing patterns in order to predict any imminent timing errors from those patterns (Section 4.3). This variant has a full fledge predictive engine in place (Section 4.3.6). For the experiments, the pattern-match threshold of 90%, and an ELT size of 10 is chosen.
- **GreenTPU-Lite (GTL):** This is a lighter variant of GreenTPU, with the most basic predictive engine capable of storing only one error-causing pattern (Section 4.3.6).
- **GreenTPU-Reactive (GTR):** This is a variant of GreenTPU with the predictive capability taken off (Section 4.3.6). The prevention of timing errors thus only occurs reactively after a timing error in the row has occured.

4.5.2 Timing Error Resilience

Figure 4.3 depicts the number of timing errors encountered during the inference of the DNN datasets under different schemes, when the TPU operates at a higher frequency, compared to the baseline. However, at all frequency—denoted by the X-axis—the operating voltage is kept constant at 0.45V. The Y-axis values are represented in a logarithmic scale. It is noticed that, *on an average, GT encounters two orders of magnitude less timing errors, with respect to TD, across all the datasets, at any higher performance level.* GT, boasting a full blown prediction engine attributes to this huge reduction. GT is seen capable of predicting most of the timing errors and preventing them from occurring. Although equipped with a preliminary prediction scheme, GTL is still seen to substantially reduce the number of timing errors compared TD and GTR. On the other hand, as GTR does not have any form of prediction mechanism, the reduction in number of timing errors in the log scale is almost negligible. These results demonstrate the importance of predictive approaches, when it comes to an exponential reduction of the timing errors. It shows that predictive approaches are the way to go when it is required to aggressively scale up the performance of a massively parallel architecture like NTC TPU.

4.5.3 Inference Accuracy and Energy

Figure 4.4 presents the variations in the inference accuracy at different performance points (Section 4.5.2), under various comparative schemes (Section 4.5.1), for 8 DNN datasets. The accuracy values of the datasets are normalized to the corresponding error-free accuracy (IMDB: 0.90, CIFAR-10: 0.77, MNIST: 0.98, REUTERS: 0.80, FSDD: 0.92, FMNIST: 0.89, GTSRB: 0.97, SVHN: 0.94) from the baseline NTC TPU. Figure 4.4 also shows the voltage boosting energy (VBE), associated with the boosting mechanism in GT and GTL. VBE is calculated as a percentage of the energy consumption of the baseline NTC systolic array with no augmentation. It is seen that the accuracy curves (left Y-axis) fall from the normalized maximum at different rates, due to varied timing error resilience of different schemes. Also, the VBE curves (right Y-axis) rise from the minimum, reflecting the different rates of increase in the number of voltage boosting events necessary to provide the required timing error resilience for different schemes.

Up to 1.4× the baseline performance, all the schemes can efficiently prevent the impact of timing errors from affecting the inference accuracy. However, as the performance is further increased, GT and GTL offer considerably better accuracies with respect to TD and GTR, for all the datasets. This is due to the high timing error resilience of GT and GTL (Section 4.5.2). The pattern matching capability, along with a larger ELT, makes GT a more effective scheme, compared to GTL. *Our baseline NTC TPU, augmented with GT, can be operated at* 2×–3× *the baseline frequency, with only 3% average loss in the inference accuracy for 7 out of 8 DNN datasets.* For CIFAR-10, GT is only as effective as GTL. This anomaly is attributed to the extreme variance in the activation patterns of CIFAR-10. GTR performs better or equal to TD in all the cases, however, noticeable increase in accuracy relative to TD is not seen. It is seen that even the most basic prediction scheme added to a reactive approach (GTR to GTL), can have huge impact in maintaining the inference accuracy. This clearly shows that the maintenance of DNN inference accuracy at aggressively higher performance points can only be achieved by near-exponentially reducing the potential timing errors. Remaining in-line with the capability of predictive schemes to be able to exponentially reduce the number of timing errors (Section 4.5.2), predictive schemes GT and GTL

maintain the accuracy far better than the non predictive and reactive approaches like GTR and TD.

The VBE of GTL—owing to its lower hardware footprint and infrequent boosting—is usually less than the VBE of GT. However, for CIFAR-10, both the schemes trigger the boosting mechanism for the same number of times, thus incurring similar energy overheads. GTR incurs the lowest VBE of all the GreenTPU variants as it has to perform boosting for the least number of times, provided that the GTR is not a predictive scheme. Despite a monotonic increase with the performance, VBEs of our proposed schemes are limited to $\sim$6% of the baseline NTC TPU energy consumption. This result is due to the sporadic occurrence of the boosting. Due to the sparsity of need for boosting, it can be concluded that, rather than selectively undervolting the MACs in a TPU operating at Super Threshold Voltage, it is highly beneficial from an energy perspective, to operate the TPU at Near Threshold Voltage and selectively boosting the MACs in a TPU. The performance loss coming from this setting can be effectively uplifted by GT to yield a highly energy efficient TPU. *Hence, GT serves as an extremely error-resilient and energy-efficient design paradigm that can unlock a high performance in future low-power NTC TPUs.* Furthermore, as GT is based on the hardware level data-delay relationship at the basic granularity of a MAC unit, it can scale well to systolic array dimension, bit width of activation/weight, and the size of the DNN applications.

4.5.4 Implementation Overheads

The hardware overheads of GreenTPU come from the TECU components, the additional voltage rail, and the augmentation of each MAC with the Razor capability. The area overhead of GreenTPU is estimated to be $\sim$1.8%. This small footprint is attributed to the fact that the systolic array occupies only 24% of the overall TPU die area [57]. GreenTPU incurs a power overhead of $\sim$2.2%, compared to the vector-less power consumption of the systolic array. From the detailed route reports, GreenTPU's wire-length overhead is estimated to be $\sim$4.1%.

CHAPTER 5

IMPROVING ENERGY EFFICIENCY OF A TENSOR PROCESSING UNIT THROUGH UNDERUTILIZATION BASED POWER-GATING

5.1 Background and Contributions of This Work

Artificial intelligence (AI) is predicted to contribute up to $15.7 trillion to the global economy by 2030 [76]. In line with the AI evolution, the computing industry has already embraced specialized AI accelerators, as conventional CPUs and GPUs are no longer able to match up the required throughput [24, 25, 33, 56]. Google's Tensor Processing Unit (TPU) [57], a representative Systolic Array (SA) based architecture, has been widely employed throughout Google data centers to meet the excessive performance demand in its Deep Neural Network (DNN) inference computations. The unprecedented growth in DNN workloads demands huge energy efficiency in these architectures. Additionally, the scarce energy resources coupled with limited hardware cost in battery powered edge-AI applications necessitates an extremely energy efficient inference architecture.

Researchers have demonstrated impressive energy savings from power-gating in CPU and GPU architectures. However, the savings are being limited due to the relatively unpredictable idleness pattern (from general purpose applications) and performance loss considerations due to the sleep and wakeup cycles of sleep transistors [43, 44, 77].A well structured and massive hardware underutilization problem in TPU SA is observed and parametrized, and a much larger opportunity of improving energy efficiency through power-gating the idle hardware resources is uncovered.

DNN inference through TPU SA is carried out by performing a matrix multiplication between input matrix and weight matrix, in the array of 256×256 Multiply-And-Accumulate (MAC) units. Weight matrix is preloaded into the SA and the input vectors can be grouped and sent as batches of different sizes. Major chunk of the SA energy con-

sumption comes from the dynamic energy consumed by the computationally active MAC units and leakage energy consumed by idle MAC units. It is observed that the number of computationally active MACs on an average for a batch computation period decreases with the batch size. It is found that, the MAC units are computationally active only for less than 40% of the time for practical batch sizes, incurring a substantial leakage loss. By parameterizing the activity and idleness pattern of the MAC units, UPTPU: an intelligent and adaptive power-gating paradigm for SAs, is formulated. Moreover, UPTPU prevents any performance or accuracy loss by a smart control around circuit level tolerances of the sleep transistors.

Furthermore, as the weights remain static for a batch computation lifecycle and any computation with a zero weight is just an energy overkill, the same sleep transistor resources are reused to powergate the zero weight holding MACs, further inflating the energy efficiency. More importantly, as the share of leakage energy in the total energy becomes more prominent for lower technology nodes [43], the effectiveness of UPTPU magnifies with the future technology scaling.

Prior works have enhanced energy efficiency from optimizations in different granularities of DNN accelerator spanning around dataflow, algorithm, memory, undervolting and so on [21, 25, 33, 56, 78–80], which either reduce the number of computations, or skip unwanted computations or reduce the energy per computation. *However, this work is the first one that improves the energy efficiency of TPUs by saving leakage loss in the under-utilized SA resources which are idly waiting for computations.* Following are the key contributions of this work:

- The cycle accurate activity and idleness profile of MAC units inside a TPU systolic array on different batch sizes of inputs is parametrized (Section 5.2.2).
- It is established that the MACs in TPU systolic array remain computationally idle for 40-90% of the time, depending on server or edge Inference applications (Section 5.2.3).
- UPTPU - a low overhead power-gating paradigm, adaptive of batch size, sleep transistor's tolerances and zero-weight computations is proposed.(Section 5.3).

- Combined with Zero-Skip [25], UPTPU offers a staggering 3.5 × −6.5× energy efficiency for eight DNN applications, with zero sacrifice on the performance and inference accuracy (Section 5.5).

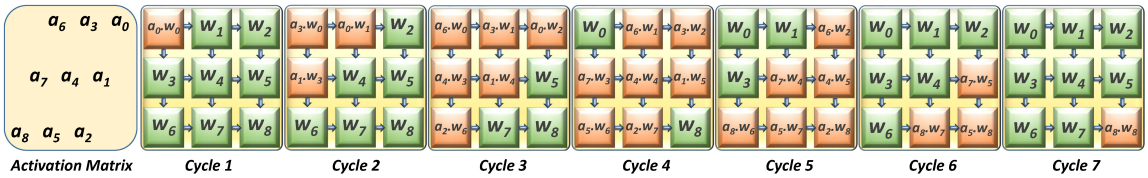


Fig. 5.1: *Cycle accurate representation of matrix multiplication between Activation and Weight matrices. Orange represents computationally active MAC unit (only multiplication shown for space constraints), whereas, green represents idle MAC unit, waiting for activation stream.*

5.2 Motivation

In this section, the opportunities of drastically reducing the energy consumption in TPU systolic arrays are demonstrated. Section 5.2.1 provides a background and architectural overview of the TPU systolic array. Section 5.2.2 presents a rigorous mathematical analysis to parametrize the computation and dataflow pattern. Finally, Section 5.2.3 presents new insights in energy saving opportunities.

5.2.1 TPU Systolic Array

Matrix Multiplication is the most crucial part of computation in the inference phase of DNN applications. TPU hosts a weight stationary 256 × 256 systolic array of MAC units to perform Matrix multiplication between 8-bit-quantized activation inputs and pre-trained weights. Weights are fetched from weight FIFO and pre-loaded to each MAC unit. Unified Buffer stores the activation inputs, which are streamed into the corresponding row of the systolic array, to be multiplied by all the weights in the row. Partial sums move downstream, adding themselves to the multiplication outputs at each row. Activation matrix is transposed and sent to the the systolic array as a (systolic) diagonal wavefront, creating a predictable dataflow.

Figurative representation of the scaled down TPU systolic array dataflow can be seen in Figure 5.1. A 3×3 Activation matrix ($[a_0, .., a_8]$) is streaming into 3×3 systolic array, with Weight matrix ($[W_0, .., W_8]$) preloaded into it. A distinct systolic pattern can be seen in the computation activity and idleness of a MAC with green and orange colors, respectively. Seeking a general and exhaustive outlook on the pattern, the accurate analytical model of the usage of hardware resources are presented and analyzed in Section 5.2.2.

5.2.2 Mathematical Parametrization

In this section, the usage of hardware resources is accurately parametrized for a general $B \times N$ Activation matrix multiplied by $N \times N$ Weight matrix in an N-dimension systolic array. Different metrics are defined in Table 5.1 and illustrate them in Equations 5.1-5.3.

T_C	Architectural lifetime (clock cycles) of the matrix multiplication of $B \times N$ activation and $N \times N$ weight matrix.
$U(n)$	No. of computationally active MACs in n^{th} clock cycle.
TRU	*True Resource Usage* : Number of computationally active MACs over matrix multiplication lifecycle, T_C.
MAR	*Maximum Available Resource* : Maximum number of MACs ideally available for computation over T_C.
RUR	*Resource Usage Ratio* : Percentage of the MAC resources used for the matrix multiplication over T_C.

Table 5.1: *Definitions of Metrics used for parametrization.*

Through rigorous mathematical analysis, it is found that $U(n)$ (Equation 5.2) can be described accurately with eight distinct quadratic and linear arithmetic sequences of n, as an artifact of varied dependence on B and N. Six regions (viz. $R_1 - R_3$ and $R_6 - R_8$ in Equation 5.2) annotate the rise and fall in the number of actively used MACs at the beginning and towards the end of T_c and two regions (viz. R_4 and R_5) describe the connection between the rising and falling regions. R_1 (and R_8) includes quadratic rise (fall) from (to) the minimum of a single active MAC . They are followed (and preceded) by linear regions R_2 (and R_7) for $B < N$ or quadratic regions R_3 (and R_6) for $B > N$. Finally, R_4 connects

R_3 and R_6 for large batch sized Activation matrix ($B > 2N-2$) with capping constant value (N^2) representing the usage of all possible MACs. And quadratic region R_5 connects regions $R1(R8)$ or $R2(R7)$ or $R3(R6)$ for medium B's ($2 < B < 2N-2$).

TRU aggregates $U(n)$ over T_c. MAR amounts to all N^2 MAC units available for T_C clock cycles. Finally, RUR, reflecting the actual resource usage ratio manifests itself as a function of N and B (Equation 5.3). Next, this parametrization is used to better understand the hardware utilization scenario in TPU systolic array with N=256.

$$\textit{Total Computation Clock Cycles } (T_C) = 2N + B - 2 \tag{5.1}$$

$$U(n) = \begin{cases} \frac{n^2}{2} + \frac{n}{2}, & R_1\ [1 \le n \le min(N,B)] \\ Bn - \frac{B(B-1)}{2}, & R_2\ [min(N,B) < n \le N] \\ -\frac{n^2}{2} + \frac{(4N-1)n}{2} - N(N-1), & R_3\ [N < n \le max(N, min(B, 2N-2))] \\ N^2, & R_4\ [max(N, min(B, 2N-2)) < n <= max(N,B)] \\ -\frac{n^2}{2} + (B+2N-1)n - \frac{B(B-1)+2N(N-1)}{2}, & R_5\ [max(N,B) < n <= max(B, (N+min(N,B)-2))] \\ -\frac{n^2}{2} + \frac{(2B-1)n}{2} - \frac{B(B-1)-2N^2}{2}, & R_6\ [max(B, (N+min(N,B)-2)) < n <= (T_C - N)] \\ -Bn + \frac{B^2}{2} + \frac{(4N-1)B}{2}, & R_7\ [(T_C - N) < n <= (T_C - min(N,B))] \\ \frac{n^2}{2} - \frac{(2B+4N-1)n}{2} + \frac{B^2 + B(4N-1) + 2N(2N-1)}{2}, & R_8\ [(T_C - min(N,B)) < n <= T_C] \end{cases} \tag{5.2}$$

$$TRU = \sum_{n=1}^{T_C} U(n) = N^2 \times B, \quad MAR = N^2 \times T_C, \quad RUR(\%) = \frac{TRU}{MAR} \times 100\% = \frac{100B}{(2N+B-2)} \tag{5.3}$$

5.2.3 TPU Hardware Resource Utilization

Using Equation 5.2, the distribution of computationally active MAC units, $U(n)$, in the TPU SA ($N = 256$), for different batch sized (B) activation matrices is plotted in Figure

5.2. The correctness of this distribution are experimentally verified on the systolic array simulator (Section 5.4). Following are the important observations from the figure. Number of active MACs reach the peak, only during the mid of the multiplication lifecycle. During the start and end of the multiplication, maximum MACs are unused. Spatially, all the available 65536 MACs can be used simultaneously (region R_4 in Equation 5.2), if a batch (B) is supplied with more than 510 inputs, and that peak utilization period can then be sustained for the difference of $(510 - B)$ cycles. For any batch size, the number of active MACs at the beginning and the end of multiplication lifecycle follows a rise and fall leading to a massive underutilized hardware components. Furthermore, with smaller batch sizes, the distance from the maximum usage keeps on increasing to result an almost entirely unused SA.

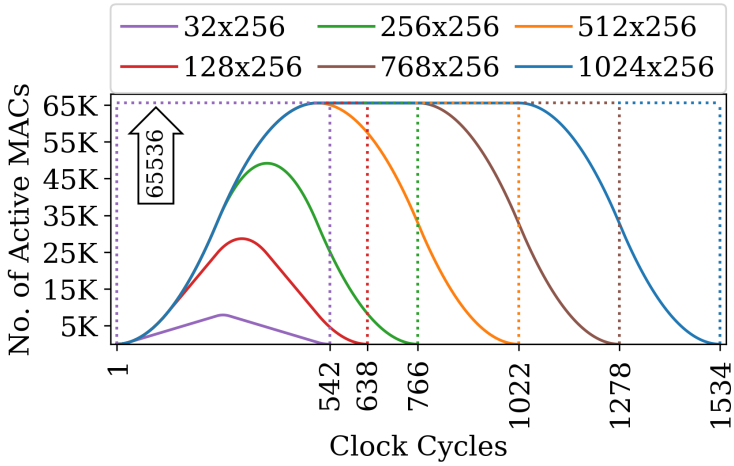


Fig. 5.2: *Distribution of computationally active MACs over all the clock cycles for different* $B \times 256$ *input matrices multiplied to* 256×256 *weight matrix. X-axis labels show the respective ends of* T_c.

Figure 5.3 plots the Resource Usage Ratio (RUR) (Equation 5.3 and Table 5.1). It exhibits the stark dependence of the RUR on the batch size. It is seen that even for a batch size of $4\times$ the SA dimension, the SA resource usage is under 65%, while smaller batch sizes result in very poor resource utilization. There are practical limits to the batch size and thus the RUR, as outlined below.

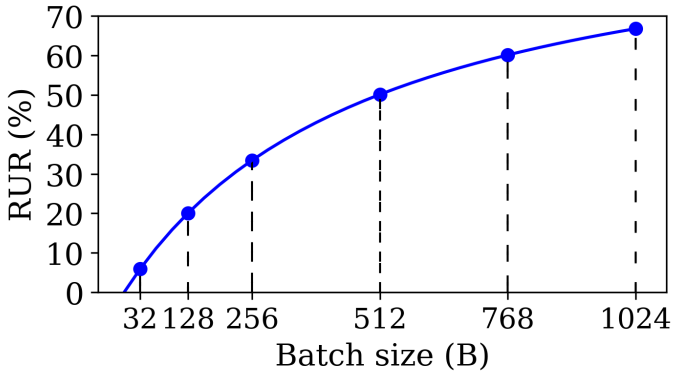


Fig. 5.3: *Resource Usage Ratio (%) for different batch sized input in TPU Matrix Multiplier Unit.*

- Very expensive high speed (typically on-chip SRAM) and higher size unified buffer is required to stream the large batch sizes of inputs and handle the consequent large output matrix.

- The inference decision from a batch of inputs can only be completed after matrix multiplications with different weights from all the layers of DNN. Hence, larger batch sizes introduce real time inference response latency [57]. *Consequently, although TPU can accommodate batch sizes upto 2048 from its 24 MiB SRAM, Google workloads' latency requirements limit the batch sizes to only around 30-200 ($RUR = 5 - 30\%$) [57].*

- Edge Inference applications are limited to handful of batches (as low as one) due to real time need of low response latencies, and are bound by energy and cost budget for streaming inputs and holding the outputs of larger batch sizes [81, 82].

This analysis points that any *RUR* less than 100% denotes that 100-*RUR*% MACs are consuming wasteful leakage energy while waiting for computation and holding the weight value. This scenario provides us with the unique opportunity to make the TPU highly energy efficient by capping the wastage energy, while not interfering with the computation throughput and accuracy. Section 5.3 details UPTPU, an intelligent power gating paradigm to carefully exploit this opportunity.

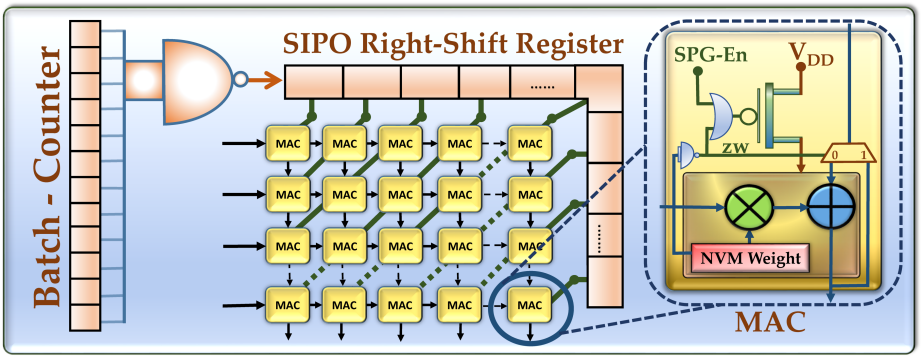


Fig. 5.4: *UPTPU design overview*

Algorithm 4 SPG Control Algorithm

1: $Batch\ Counter(BC) \leftarrow Batch\_Size - 1$
2: $Systolic\ Array\ Dimension(N) \leftarrow 256$
3: $T_w \leftarrow wake\_up\_tolerance$
4: $sleep\_bit \leftarrow 1, wake\_bit \leftarrow 0$
5: $SR[(2N-2).....(2N-2-T_w)] \leftarrow wake\_bit$
6: $SR[(2N-2-T_w).....0] \leftarrow sleep\_bit$
7: **while** $BC > 0$ **do**
8: $\quad SR \leftarrow SR \gg 1$
9: $\quad SR[2N-2] \leftarrow wake\_bit$
10: $\quad BC \leftarrow BC - 1$
11: **end while**
12: **while** $current\_batch$ **do**
13: $\quad SR \leftarrow SR \gg 1$
14: $\quad SR[2N-2] \leftarrow sleep\_bit$
15: **end while**

Line 3: T_w refers to the number of clock cycles alloted for the MAC diagonals to completely power up from sleep-state and be ready for error free computation. This enables a **zero performance (TOPS) overhead.**

Lines 5-6: SR is initialized to T_w $wake\_bits$ followed entirely by $sleep\_bits$. Right Shifting of SR along with these T_w $wake\_bits$ ensures that **all the idle MAC diagonals will start switching on T_w clock cycles in advance of the computation scheduled in that diagonal.**

Lines 7-11: SR shifts right by injecting $wake\_bit$ to the MSB, **waking-up only one additional diagonal at a time**, until BC counts to zero.

Lines 12-15: After BC cycles, SR starts shifting right by injecting $sleep\_bit$ at the MSB, **sleeping only one additional diagonal at a time.**

5.3 UPTPU Design

In this section, ***UPTPU***: ***U****nderutilization based* ***P****ower-gating Paradigm for* ***TPU***, a low overhead design paradigm is presented which curbs almost entire wasteful leakage energy coming from severely underutilized MAC units, through intelligent power gating. Section 5.3.1 presents the batch size aware gating control strategy. Section 5.3.2 delves into replacing the volatile storage units of MAC unit with NVMs. Finally, Section 5.3.3 discusses the circuit level specifics on the power gating design choices.

5.3.1 Power-Gating Control Strategy

Systolic Power Gating (SPG)

Section 5.2 exposes the stark dependence of the underutilization of the TPU systolic array on batch size. The batch of activation inputs is supplied to the systolic array in a diagonal fashion so that the wave of the *activity* in MACs advance diagonally to use up the idle MACs one diagonal each clock cycle. After the clock cycles equal to the batch size, the wave of *idleness* advances one diagonal each cycle. The phenomenon can be seen in Figure 5.1 with orange (*activity*) and green (*idleness*) colors. With the advance knowledge of batch size, an accurate assessment of which diagonal of MAC units are active and idle in any computation clock cycle can be achieved. It is assumed that the software can provide batch size along with the data.

Figure 5.4 shows the design overview of UPTPU. Following the systolic-diagonal trend of activity and idleness, A $2N-1$ bit Serial-In-Parallel-Out (SIPO) Right Shift Register (SR) is included whose bits are physically mapped onto $2N-1$ diagonals of MACs. The parallel right-shifting bit values serve as systolic wake/sleep signals for each diagonal. A Batch Counter is conceived which will be loaded with the binary representation of $Batch\_Size-1$, the value of which will stream $sleep\_bit$ and $wake\_bit$. Algorithm 4 and its respective description explain how the batch size information is used to map bits in the SR, while ensuring a *zero overhead in performance*.

Zero Weight Power Gating (ZWPG)

It is observed that significant amount of the computations in the MAC units for the practical DNN datasets see *'zero computations'* involving either zero activation or zero weight. Figure 5.6 shows the percentage distribution of *Zero Activation or Weight Computations (ZAWC)* (average 75%) and *Zero Weight Computations (ZWC)* (average 26%) for different DNN datasets. These zero computations can occur either naturally from the activation and trained weights or from zero padding of some activation and weight matrices to fit into the 256×256 TPU Systolic Array. However, only the MACs with zero weights are practical to be powergated as the weights remain static over the batch computation lifecycle, T_c (Equation 5.1) and the sleep transistors won't have to be woken up until T_c.

Zero Weight Power Gating (ZWPG) is proposed to extend the applicability of the SPG sleep transistor to curb the energy consumptions from zero weight MACs. As seen in figure 5.4, the zero weight (zw) signal coming from weights stored in NVM (Section 5.3.2) puts the MAC unit to sleep irrespective of SPG-En signal. Unlike a MAC gated with SPG, a MAC gated with ZWPG should be able to route the upstream data through itself. A demux is used to route the upstream MAC's data by bypassing the MAC powergated with ZWPG.

5.3.2 Usage of NVMs

As each MAC unit stores a weight value in its associated volatile SRAM cells, it is needed to preserve the weight values during powergating for seamless computation on wake-up. Classic powergating employs retention cells, which call for further leakage [77], given that there is a SRAM register in each of the 256×256 MACs. The replacement of leaky weight holding SRAM cells with non-leaky STT-MRAM Non-Volatile Memory (NVM) to solve both the hurdles of volatility and memory leakage is envisioned. STT-MRAMs also provide other compelling advantages for the niche of weight stationary systolic arrays. STT-MRAMs boast about $20\times$ reduction in leakage energy and about $4\times$ increment in packing density with respect to SRAMs and they have comparable read characteristics to SRAMs [83].

The unique write pattern in TPU systolic array facilitates the adoption of STT-MRAMs although they suffer from a 3× write speed and 20× energy penalty for writes [83]. The weights in the MACs are only written once for an entire batch of computation, which gives the delay overhead of less than 0.5%, even for the smallest of batch sizes. More importantly, the spread of 20× savings in leakage power over the batch computation lifecycle, diminishes the once-per-batch 20× write-energy-increase. Thus, usage of NVM overturns the otherwise overkill in both energy and area consumption coming from SRAMs, and also facilitates the sleep of entire MAC unit without retention leakage. In attempt to replace the SRAMs with STTMRAMs in CPU/GPU caches, researchers have compensated the write penalties by packing manyfold MRAMs in the same area footprint of SRAM cache [83], [84]. The area savings with STT-MRAM contributes to amortizing the area overhead due to sleep transistors.

5.3.3 Circuit Level Considerations for Power-Gating

Sleep transistor design is a challenging VLSI domain because of the difficulty in optimization around its various effects on design performance, area, routability, overall power dissipation, and signal/power integrity [44]. One sleep transistor per MAC is conceived, which receives a per-diagonal power gating control signal. Although it might seem intuitive to house one sleep transistor per diagonal, per-MAC strategy eliminates several circuit-level complications. As the diagonals represent diverse switching loads pertaining to the varying number of connected MACs (1-256), design of graded sleep transistors adds huge design complexity. In addition, the usage of sleep transistor at the granularity of a MAC facilitates ZWPG (Section 5.3.1) and eliminates the need of having bulkier power lines running through each diagonal, ultimately improving routability.

As the sleep and wake-up happens at a granularity of just one diagonal at a time during computation (Algorithm 4), noise and current crowding issues are minimized and the average sleep time is maximized. Moreover, the decrease in area penalty, noise, and the high power-on rush current is favored by compromising the switching speed of the

sleep transistor. The system wide performance is not affected by slower sleep transistors because of the wake-up tolerance included in the gating control strategy (T_w in Algorithm 4). The 6% area overhead of PMOS sleep transistors [43], combined with the overheads from control hardware, dilutes to only around 3.4% area overhead with respect to the entire TPU die.

5.4 Methodology

In-house cycle accurate TPU systolic array simulator is used, which is built upon [85], with architectural details from [57], as an architectural simulator for the cycle accurate assessment of computation data and resource utilization pattern. First, eight DNN applications (viz., MNIST [68] , Reuters [69] , CIFAR-10 [70] , IMDB [71] , SVHN [72] , GTSRB [73] , FMNIST [74] , FSDD (Audio-MNIST) [75]) are trained using Keras with TensorFlow backend and extract the weights from the trained model. The 8-bit quantized activation input is streamed from the datasets in several batch sizes to the simulator to be multiplied with the weight matrices stored in SA. The output matrices from the simulator are combined to evaluate the inference accuracy.

The energy efficiency model is developed by conjoining the architectural outcomes of the datasets with estimations of dynamic and leakage energy from CAD tools. The RTL description of SA MAC units is synthesized with different design augmentations, through Synopsys Design Compiler followed by place and route through Cadence SoC Encounter using 45nm standard cell library, to estimate the area and energy (dynamic and leakage) consumption and associated overheads. The leakage energy is found to be 20% of the dynamic energy. The wake-up tolerance (T_w in Algorithm 4) is set to three clock cycles, inline with the prior power gate implementations [43], [44], [45] . The switching energy overhead is embedded in the model with break even clock cycles, as suggested by [45].

5.5 Experimental Results

In this section, the efficacy of different schemes are evaluated on increasing the energy efficiency of a 256 $\times$ 256 TPU systolic array. Section 5.5.1 presents the comparative schemes. Section 5.5.2 compares and describes the energy efficiency coming from different schemes.

5.5.1 Comparative Schemes

- **Zero-Skip (ZS):** This is a widely used technique for drastically improving the energy efficiency of DNN Accelerators [25, 86, 87], where the computation in MAC is entirely skipped if activation input or weight is equal to zero. Zero skipping gets rid of the dynamic energy for those MAC units which hold zero weight or receive zero activation.
- **UPTPU-LITE:** This is an extension to ZS, with application of *Z*ero *W*eight *P*ower *G*ating (ZWPG). All the MAC units holding the weight value of zero are power gated for the computation lifecycle of a batch of activation inputs. In addition to the dynamic energy savings from ZS, this scheme prevents the leakage power from the zero weight holding MACs.
- **UPTPU:** UPTPU includes the Systolic Power Gating (SPG) of unutilized MAC units, in addition to the benefits provided by UPTPU-LITE. It intelligently powergates almost all the idle MAC units arising from TPU underutilization on different batch sizes.

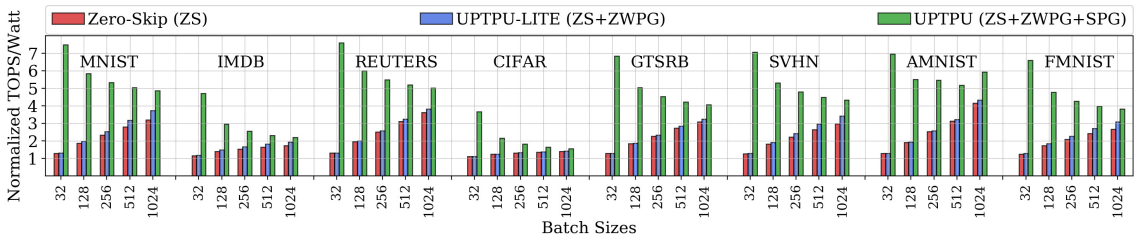


Fig. 5.5: *Normalized TOPS/Watt of eight DNN datasets computed on a TPU systolic array with different batch sizes brought about by the comparative schemes.*

5.5.2 Interpretation of Energy Efficiency

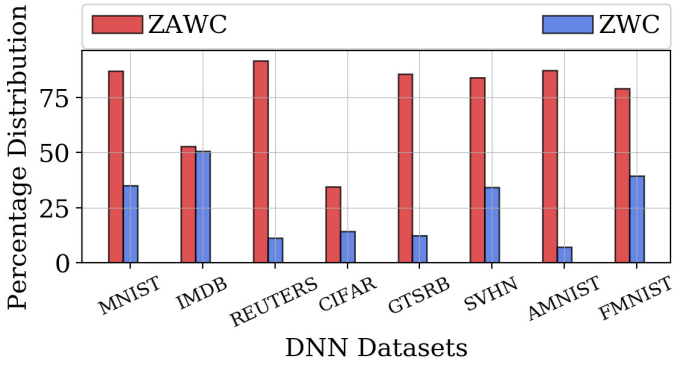


Fig. 5.6: *Zero Activation or Weight Computations (ZAWC) and Zero Weight Computations (ZWC) expressed as percentage of total computations for different DNN datasets.*

The gains in energy efficiency are simulated for eight DNN datasets, when the computation is performed in different batch sizes. Figure 5.5 presents the gain in Tera Operations Per Second per Watt (TOPS/Watt) normalized with base TPU SA for eight DNN datasets, for different comparative schemes. Figure 5.6 presents the batch-size independent Zero *A*ctivation or *W*eight Computations (ZAWC) and Zero *W*eight Computations (ZWC) among the total MAC computations pertinent to the ZS and ZWPG schemes respectively. Various trends are seen in energy efficiency gains for different datasets and schemes. In general, the maximum average gain for any dataset (Figure 5.5) is dictated by the percentage of ZAWC (Figure 5.6). Higher ZAWC gives many opportunities for ZS embedded in all comparative schemes. The datasets with relatively lower ZAWC (viz. IMDB and CIFAR) have relatively lower energy efficiency gains.

A minimal benefit in UPTPU-LITE (ZS+ZWPG) is seen in comparison to ZS, as the extra ZWPG scheme adds the small additional leakage savings coming from the small subset (ZWC-Figure 5.6) of dynamically skipped MACs. The relatively smaller subsets (viz. REUTERS, AMNIST, GTSRB) result in minimal benefit addition to gains. However, more importantly, the gains from UPTPU-LITE (ZS+ZWPG) decrease for lower batch sizes. As the RUR decreases with lower batch sizes (Section 5.2), the constant benefits coming from ZS and ZWPG are progressively diluted by the increasing leakage energy consumption in unutilized MACs.

Finally, UPTPU (ZS+ZWPG+SPG) is able to achieve much higher gains, because of the

addition of Systolic Power Gating (SPG) which intelligently power gates the unutilized MACs. In addition to higher average gain, a complementing effect to ZS and ZWPG is also achieved, pronounced by the increase of the energy efficiency with the decrease in the batch size. As the batch sizes decrease, SPG gets increasing opportunities from decreasing RUR to give massive gain in TOPS/Watt. UPTPU achieves, on a average of $3.5\times - 6.5\times$ gain in TOPS/Watt for batch sizes $1024 - 32$. This shows that UPTPU can achieve staggering energy efficiency gains throughout the range of both highest and lowest ends of the batch sizes. The performance and inference accuracy is not compromised at all, because of the dataflow adaptive intelligent power gating (Algorithm 4).

CHAPTER 6
CONCLUSION

This dissertation proposes design methodologies to improve the security and performance in a near-threshold implementation of SRAM PUFs and TPU, while also significantly improving energy efficiency of TPU operating at nominal voltage. The enhancement in SRAM PUF security is shown through significant improvement in the uniformity and reliability metrics. Higher performance is unlocked in NTC TPU by substantially elevating the timing error resilience at near-threshold voltages. The prominent energy efficiency in the STC TPU is extracted by identifying and carefully masking the sizeable dataflow guided leakage energy through powergating.

Various threats to reliability and uniformity characteristics of NTC-operated SPUF are analyzed. Leveraging the impact of device asymmetry on these characteristics, the current suppression techniques (viz. CUBIT and CUSIT) are crafted. The principles governing CUBIT and CUSIT schemes are based on biasing and sizing various read and write counterparts of a 8T-SRAM PUF respectively. CUBIT and CUSIT adaptively mitigate the accentuated effects of PV on reliability and uniformity, by giving a comprehensive improvement of more than 82% in reliability and 55% in uniformity metrics with negligible overheads. With improved reliability and uniformity, NTC SPUFs are presented as viable alternatives in security primitives to the conventional power hungry 6T-SRAM PUFs.

The unprecedented growth of the DNN workloads in the recent years, requires an energy-efficient DNN accelerator design paradigm, that can offer an optimal inference accuracy at a high performance. In this dissertation, we present GreenTPU—an energy-optimized systolic array design for Google TPU—a state-of-the-art DNN accelerator is presented. Operating at the NTC condition, GreenTPU can efficiently predict and prevent the imminent timing errors in its systolic array of MACs, thus offering close to an error-free accuracy with a high performance. It is also established that predictive approaches to error

resilience, have the required potential to maintain DNN inference accuracy in aggressively performance scaled DNN accelerator platforms. Compared to a recently proposed timing error mitigation strategy for TPUs, GreenTPU enables 2×–3× higher performance (TOPS) in an NTC TPU, with a minimal loss in the prediction accuracy, and minor hardware footprints. GreenTPU paves a way towards adoption of low power design paradigms like NTC in the mainstream computing industry with an elevated confidence in their system performance, owing to a more greener AI future.

This dissertation also attempts to significantly improve the energy efficiency of the TPU at the granularity of STC (nominal) operating voltage. A huge hardware underutilization problem is parametrized in the weight stationary systolic array with rigorous mathematical analysis. The leakage energy spent in the systemic underutilization is then masked through intelligent powergating layer, which dynamically adapts to the dataflow and batch size, bestowing a $3.5\times - 6.5\times$ gain in energy efficiency, when combined with other energy efficient schemes. The scheme can be superimposed on top of other existing architectural or circuit level techniques to inflate the energy efficiency, without any compromise in the inference accuracy or performance. More generally, due to a predictable data-flow pattern in the AI workload, this work opens up newer avenues for exploration of power-gating based energy efficient solutions for all forms of AI accelerators.

In conclusion, this dissertation embraces the application, adaptation and proliferation of low power systems in mainstream computing, by putting forward innovations and design methodologies, to solve the reliability and performance problems in existing low power design paradigms and providing energy efficiency to existing designs. It is hoped that this dissertation adds significant contribution to the academia and design practices in semiconductor industry.

REFERENCES

[1] A. S. Andrae and T. Edler, "On global electricity usage of communication technology: trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, 2015.

[2] R.G.Dreslinski, M.Wieckowski, D. Blaauw, D.Sylvester, and T.Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," in *Proc. IEEE*, Feb. 2010.

[3] N. Pinckney, K. Sewell, R. Dreslinski, D. Fick, T. M. udge, D. Sylvester, and D. Blaauw, "Assessing the performance limits of parallelized near-threshold computing," in *DAC*, 2012, pp. 1143–1148.

[4] S. Hsu, A. Agarwal, M. Anders, S. Mathew, H. Kaul, F. Sheikh, and R. Krishnamurthy, "A 280mv-to-1.1v 256b reconfigurable SIMD vector permutation engine with 2-dimensional shuffle in 22nm CMOS," 2012, pp. 178–180.

[5] D. Markovic, C. C. Wang, L. P. Alarcon, T.-T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 237–252, 2010.

[6] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," ser. DAC '07, 2007, pp. 9–14.

[7] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Computers*, pp. 1198–1210, 2009.

[8] G. Selimis, M. Konijnenburg, M. Ashouei, J. Huisken, H. de Groot, V. van der Leest, G. J. Schrijen, M. van Hulst, and P. Tuyls, "Evaluation of 90nm 6t-sram as physical unclonable function for secure key generation in wireless sensor nodes," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, 2011, pp. 567–570.

[9] M. Kassem, M. Mansour, A. Chehab, and A. Kayssi, "A sub-threshold sram based puf," in *2010 International Conference on Energy Aware Computing*, 2010, pp. 1–4.

[10] L. Chang, R. Montoye, Y. Nakamura, K. Batson, R. Eickemeyer, R. Dennard, W. Haensch, and D. Jamsek, "An 8t-sram for variability tolerance and low-voltage operation in high-performance caches," vol. 43, no. 4, pp. 956–963, 2008.

[11] B. H. Calhoun and A. Chandrakasan, "A 256kb sub-threshold sram in 65nm cmos," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, 2006.

[12] K. Mehrabi, B. Ebrahimi, and A. Afzali-Kusha, "A robust and low power 7t sram cell design," in *2015 18th CSI International Symposium on Computer Architecture and Digital Systems (CADS)*, 2015.

[13] A. Garg and T. T. Kim, "Design of sram puf with improved uniformity and reliability utilizing device aging effect," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 1941–1944.

[14] M. Bhargava, C. Cakir, and K. Mai, "Reliability enhancement of bi-stable pufs in 65nm bulk cmos," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, 2012, pp. 25–30.

[15] S. Chellappa, A. Dey, and L. T. Clark, "Improved circuits for microchip identification using sram mismatch," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 2011, pp. 1–4.

[16] C.-H. Chang, C. Q. Liu, L. Zhang, and Z. H. Kong, "Sizing of sram cell with voltage biasing techniques for reliability enhancement of memory and puf functions," *Journal of Low Power Electronics and Applications*, vol. 6, no. 3, 2016.

[17] A. T. Elshafiey, P. Zarkesh-Ha, and J. Trujillo, "The effect of power supply ramp time on sram pufs," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 946–949.

[18] P. Simons, E. van der Sluis, and V. van der Leest, "Buskeeper pufs, a promising alternative to d flip-flop pufs," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, 2012.

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

[20] F. Libano, B. Wilson, J. Anderson, M. Wirthlin, C. Cazzaniga, C. Frost, and P. Rech, "Selective hardening for neural networks in fpgas," *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 216–222, 2018.

[21] J. Zhang, K. Rangineni, Z. Ghodsi, and S. Garg, "Thundervolt: Enabling aggressive voltage underscaling and timing error resilience for energy efficient deep neural network accelerators," *arXiv preprint arXiv:1802.03806*, 2018.

[22] W. Choi, D. Shin, J. Park, and S. Ghosh, "Sensitivity based error resilient techniques for energy efficient deep neural network accelerators," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: ACM, 2019, pp. 204:1–204:6. [Online]. Available: http://doi.acm.org/10.1145/3316781.3317908

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

[24] Y.-H. Chen, J. Emer, and V. Sze, "Using dataflow to optimize energy efficiency of deep neural network accelerators," *IEEE Micro*, vol. 37, no. 3, pp. 12–21, 2017.

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

[26] Y. Lin, S. Zhang, and N. R. Shanbhag, "Variation-tolerant architectures for convolutional neural networks in the near threshold voltage regime," in *Signal Processing Systems (SiPS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 17–22.

[27] *A 28nm SoC with a 1.2GHz 568nJ/prediction sparse deep-neural-network engine with>0.1 timing error rate tolerance for IoT applications*, 2017.

[28] P. N. Whatmough, S. K. Lee, D. Brooks, and G. Wei, "Dnn engine: A 28-nm timing-error tolerant sparse deep neural network processor for iot applications," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 9, pp. 2722–2731, Sep. 2018.

[29] P. N. Whatmough, S. Das, and D. M. Bull, "A low-power 1ghz razor fir accelerator with time-borrow tracking pipeline and approximate error correction in 65nm cmos," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2013, pp. 428–429.

[30] P. N. Whatmough, S. Das, D. M. Bull, and I. Darwazeh, "Circuit-level timing error tolerance for low-power dsp filters and transforms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 6, pp. 989–999, June 2013.

[31] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 9, no. 6, p. 813–823, 2001.

[32] G. Karakonstantis, N. Banerjee, and K. Roy, "Process-variation resilient and voltage scalable dct architecture for robust low-power computing," *IEEE Trans. Very Large Scale Integr. Syst.*, p. 1461–1470, 2010.

[33] S. Kim, P. Howe, T. Moreau, A. Alaghi, L. Ceze, and V. S. Sathe, "Energy-efficient neural network acceleration in the presence of bit-level memory errors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, no. 99, pp. 1–14, 2018.

[34] J.-S. Kim and J.-S. Yang, "Dris-3: Deep neural network reliability improvement scheme in 3d die-stacked memory based on fault analysis," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[35] N. Chandramoorthy, K. Swaminathan, M. Cochet, A. Paidimarri, S. Eldridge, R. Joshi, M. Ziegler, A. Buyuktosunoglu, and P. Bose, "Resilient low voltage accelerators for high energy efficiency," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2019, pp. 147–158.

[36] S. Yin, S. Tang, X. Lin, P. Ouyang, F. Tu, J. Zhao, C. Xu, S. Li, Y. Xie, S. Wei *et al.*, "Parana: A parallel neural architecture considering thermal problem of 3d stacked memory," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 1, pp. 146–160, 2018.

[37] B. Salami, O. S. Unsal, and A. C. Kestelman, "On the resilience of rtl nn accelerators: Fault characterization and mitigation," in *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. IEEE, 2018, pp. 322–329.

[38] D.-T. Nguyen, N.-M. Ho, and I.-J. Chang, "St-drc: Stretchable dram refresh controller with no parity-overhead error correction scheme for energy-efficient dnns," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: ACM, 2019, pp. 205:1–205:6. [Online]. Available: http://doi.acm.org/10.1145/3316781.3317915

[39] J. K. Eshraghian, S.-M. Kang, S. Baek, G. Orchard, H. H.-C. Iu, and W. Lei, "Analog weights in reram dnn accelerators," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2019, pp. 267–271.

[40] S. Ghodrati, H. Sharma, S. Kinzer, A. Yazdanbakhsh, K. Samadi, N. S. Kim, D. Burger, and H. Esmaeilzadeh, "Mixed-signal charge-domain acceleration of deep neural networks through interleaved bit-partitioned arithmetic," *arXiv preprint arXiv:1906.11915*, 2019.

[41] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.

[42] C. Mackin, H. Tsai, S. Ambrogio, P. Narayanan, A. Chen, and G. W. Burr, "Weight programming in dnn analog hardware accelerators in the presence of nvm variability," *Advanced Electronic Materials*, vol. 5, no. 9, p. 1900026, 2019.

[43] J. Tschanz, S. Narendra, Y. Ye, B. Bloechel, S. Borkar, and V. De, "Dynamic-sleep transistor and body bias for active leakage power control of microprocessors," in *2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC.*, Feb 2003, pp. 102–481 vol.1.

[44] K. Shi and D. Howard, "Challenges in sleep transistor design and implementation in low-power designs," 2006, pp. 113–116.

[45] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose, "Microarchitectural techniques for power gating of execution units," 2004, pp. 32–37.

[46] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. N. Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proc. of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[47] L. Chang, Y. Nakamura, R. K. Montoye, J. Sawada, A. K. Martin, K. Kinoshita, F. H. Gebara, K. B. Agarwal, D. J. Acharyya, W. Haensch, K. Hosokawa, and D. Jamsek, "A 5.3ghz 8t-sram with operation down to 0.41v in 65nm cmos," in *2007 IEEE Symposium on VLSI Circuits*, 2007.

[48] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," *IACR Cryptology ePrint Archive*, vol. 2011, 2011.

[49] ASU, *Predictive Technology Models (PTM) ASU*, http://ptm.asu.edu.

[50] W. Liu and C. Hu, "Bsim4 and mosfet modeling for ic simulation," 2011.

[51] Synopsis, *HSPICE ® User Guide: Advanced Analog Simulation and Analysis*, 2013.

[52] S. Birla, N. K. Shukla, K. Rathi, R. K. Singh, and M. Pattanaik, "Analysis of 8t SRAM cell at various process corners at 65 nm process technology," *Circuits and Systems*, pp. 326–329, 2011.

[53] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of sram array for yield enhancement in nanoscaled cmos," vol. 24, no. 12, pp. 1859 – 1880, dec. 2005.

[54] Y. Morita, H. Fujiwara, H. Noguchi, Y. Iguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto, "Area optimization in 6t and 8t SRAM cells considering vth variation in future processes," *IEICE Transactions*, pp. 1949–1956, 2007.

[55] K. Ishibashi, *Low power and reliable SRAM memory cell and array design*. Berlin New York: Springer, 2011.

[56] V. Gokhale, A. Zaidy, A. X. M. Chang, and E. Culurciello, "Snowflake: An efficient hardware accelerator for convolutional neural networks," in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–4.

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

[58] Ok google, siri, alexa, cortana; can you tell me some stats on voice search? https://edit.co.uk/blog/google-voice-search-stats-growth-trends/.

[59] D. Ernst, N. S. Kim, S. Das, S. Pant, R. R. Rao, T. Pham, C. H. Ziesler, D. Blaauw, T. M. Austin, K. Flautner, and T. N. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," 2003, pp. 7–18.

[60] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[61] NanGate, http://www.nangate.com/?page_id=2328.

[62] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Varius:a model of process variation and resulting timing errors for microarchitects," vol. 21, pp. 3 –13, 2008.

[63] T. Shabanian, A. Bal, P. Basu, K. Chakraborty, and S. Roy, "Ace-gpu: Tackling choke point induced performance bottlenecks in a near-threshold computing gpu," 2018.

[64] T. N. Miller, X. Pan, R. Thomas, N. Sedaghati, and R. Teodorescu, "Booster: Reactive core acceleration for mitigating the effects of process variation and application imbalance in low-voltage chips," in *HPCA*, 2012, pp. 1–12.

[65] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm early design exploration," vol. 53, no. 11, pp. 2816 –2823, 2006.

[66] U. R. Karpuzcu, K. B. Kolluru, N. S. Kim, and J. Torrellas, "Varius-ntv: A microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages," 2012, pp. 1–11.

[67] S. K. Khatamifard, M. Resch, N. S. Kim, and U. R. Karpuzcu, "Varius-tc: A modular architecture-level model of parametric variation for thin-channel switches," 2016, pp. 654–661.

[68] Y. LeCun and C. Cortes, "MNIST handwritten digit database," http://yann.lecun.com/exdb/mnist/, 2010.

[69] "Reuters-21578 dataset," http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html, 2021.

[70] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[71] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis." Association for Computational Linguistics, 2011, pp. 142–150.

[72] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. [Online]. Available: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[73] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, no. 0, pp. –, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608012000457

[74] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017. [Online]. Available: http://arxiv.org/abs/1708.07747

[75] "Free spoken digit dataset (fsdd)," https://github.com/Jakobovski/free-spoken-digit-dataset, 2021.

[76] "Ai will add 15 trillion to the world economy by 2030," https://www.forbes.com/sites/greatspeculations/2019/02/25/ai-will-add-15-trillion-to-the-world-economy-by-2030/, 2019.

[77] Y.Wang, S.Roy, and N.Ranganathan, "Run-time power-gating in caches of gpus for leakage energy savings," in *Proc. of DATE*, March 2012.

[78] P. Pandey, P. Basu, K. Chakraborty, and S. Roy, "Greentpu: Improving timing error resilience of a near-threshold tensor processing unit," 2019, pp. 173:1–173:6.

[79] V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 g-ops/s mobile coprocessor for deep neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, p. 696–701.

[80] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "Shidiannao: Shifting vision processing closer to the sensor," in *2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, June 2015, pp. 92–104.

[81] J. Hanhirova, T. Kamaainen, S. Seppaa, M. Siekkinen, V. Hirvisalo, and A. Yla-Jaaski, "Latency and throughput characterization of convolutional neural networks for mobile computer vision," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18, 2018, p. 204–215.

[82] Z. Jiang, "Efficient deep learning inference on edge devices," in *SysML COnference*, 2018.

[83] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen, "Circuit and microarchitecture evaluation of 3d stacking magnetic ram (mram) as a universal memory replacement," in *2008 45th ACM/IEEE Design Automation Conference*, June 2008, pp. 554–559.

[84] J. Zhang, M. Jung, and M. Kandemir, "Fuse: Fusing stt-mram into gpus to alleviate off-chip memory access overheads," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2019, pp. 426–439.

[85] "Ucsb archlab opentpu project," https://github.com/UCSBarchlab/OpenTPU.

[86] J. Albericio, P. Judd, T. Hetherington, T. Aamodt, N. E. Jerger, and A. Moshovos, "Cnvlutin: Ineffectual-neuron-free deep neural network computing," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 1–13.

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

CURRICULUM VITAE

Pramesh Pandey

Journal Articles

- Challenges and Opportunities in Near-Threshold DNN Accelerators around Timing Errors. Pramesh Pandey, Noel Daniel Gundi, Prabal Basu, Tahmoures Shabanian, Mitchell Patrick, Koushik Chakraborty, Sanghamitra Roy. *Journal of Low Power Electronics and Applications 2020, 10(4), 33*

- GreenTPU: Predictive Design Paradigm for Improving Timing Error Resilience of a Near-Threshold Tensor Processing Unit. Pramesh Pandey, Prabal Basu, Koushik Chakraborty, Sanghamitra Roy. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 7, pp. 1557-1566, July 2020*

- TITAN: Uncovering the Paradigm Shift in Security Vulnerability at Near-Threshold Computing. Prabal Basu, Pramesh Pandey, Aatreyi Bal, Chidhambaranathan Rajamanikkam, Koushik Chakraborty and Sanghamitra Roy. *IEEE Transactions on Emerging Topics in Computing (TETC), vol. 1, pp. 1-1, 2018.*

- FIFA: Exploring a Focally Induced Fault Attack Strategy in Near-Threshold Computing. Prabal Basu, Chidhambaranathan Rajamanikkam, Aatreyi Bal, Pramesh Pandey, Trevor Carter, Koushik Chakraborty and Sanghamitra Roy. *IEEE Embedded Systems Letters (ESL), vol. 10, issue. 4, pp. 115-118, 2018.*

Conference Papers

- UPTPU: Improving Energy Efficiency of a Tensor Processing Unit through Underutilization Based Power-Gating. Pramesh Pandey, Noel Daniel Gundi, Koushik Chakraborty

and Sanghamitra Roy. Accepted for publication in *IEEE/ACM Design Automation Conference (DAC), 2021.*

- GreenTPU: Improving Timing Error Resilience of a Near-Threshold Tensor Processing Unit. Pramesh Pandey, Prabal Basu, Koushik Chakraborty and Sanghamitra Roy. IEEE/ACM *Design Automation Conference (DAC)*, 2019.

- EFFORT: Enhancing Energy Efficiency and Error Resilience of a Near-Threshold Tensor Processing Unit. Noel Daniel, Tahmoures Shabanian, Prabal Basu, Pramesh Pandey, Koushik Chakraborty, Sanghamitra Roy, Zhen Zhang, *Asia and South Pacific Design Automation Conference (ASPDAC)'20*.

- Reliability and Uniformity Enhancement in 8T-SRAM based PUFs operating at NTC. Pramesh Pandey, Asmita Pal, Koushik Chakraborty, Sanghamitra Roy. *International Symposium on Low Power Electronics and Design (ISLPED)'18*.