ReRAM In Process Computing

Master of Science in Electrical Engineering
Specialization in Electronics Design and Technology

Submitted by,
Aleena Johnson
Matriculation no: 1701388

February 2025

Submitted to,
Prof. Dr.-Ing. Michael Wahl

Contents

List of Figures

Abstract

This paper presents a novel mixed-bit ReRAM-based architecture that integrates Neural Architecture Search (NAS) with Twin Range Quantization (TRQ) to achieve energy-efficient, high-performance neural network acceleration for edge AI applications. By leveraging NAS, the architecture dynamically adjusts the bit-width of each CNN layer (ranging from 1 to 8 bits), ensuring an optimal balance between accuracy and power consumption. TRQ further enhances the design by selectively optimizing ADC operations, reducing redundant A/D conversion steps and significantly lowering power usage without compromising precision. Experimental evaluations on benchmark models demonstrate that our approach maintains competitive inference accuracy while markedly improving energy efficiency. This work offers a comprehensive framework for low-power neural network processing and paves the way for future research in advanced in-memory computing architectures.

Chapter 1

Introduction to ReRAM

Resistive random-access memory (ReRAM), as a resistive switching memory, covers a broad range of memory and storage types of semiconductor devices. Generally speaking, resistive switching memory includes any devices with resistance change under external stress.An RRAM device contains a component called a memristor – a contraction of memory resistor – where the resistance varies from a high-resistance state to a low-resistance state when different voltages are imposed across it.Having introduced the basics of ReRAM and memristors, we now explore the fundamental principles that govern their operation. It operates on the principle of the memristor. Figure 1.1 illustrates how memristors can be used to perform dot-product computations. Each bitline connects to each wordline through a ReRAM cell. Let R and G denote a cell's resistance and conductance, respectively, where $G = 1/R$. Define

$$V = [V_1, V_2, \ldots, V_k]$$

as the applied voltages for each row and

$$G = [G_1, G_2, \ldots, G_k]$$

as the corresponding conductance values in a column; the total current is then calculated as the dot product [2]

$$I = V \times G,$$

, summing the products of each voltage and conductance pair. In terms of neural networks, the synaptic weights of neurons are encoded as the conductances of the ReRAM cells.Then, the total current is the output of neuron in a CNN output filter. As shown in Figure 1.1, the memristor crossbar array (MCA) achieves very high parallelism and can perform MVM in a single time step. Computers based on the von Neumann architecture physically separate computation and storage data is fetched from memory, processed in the CPU, and then stored back. In-memory computing addresses this limitation by designing systems

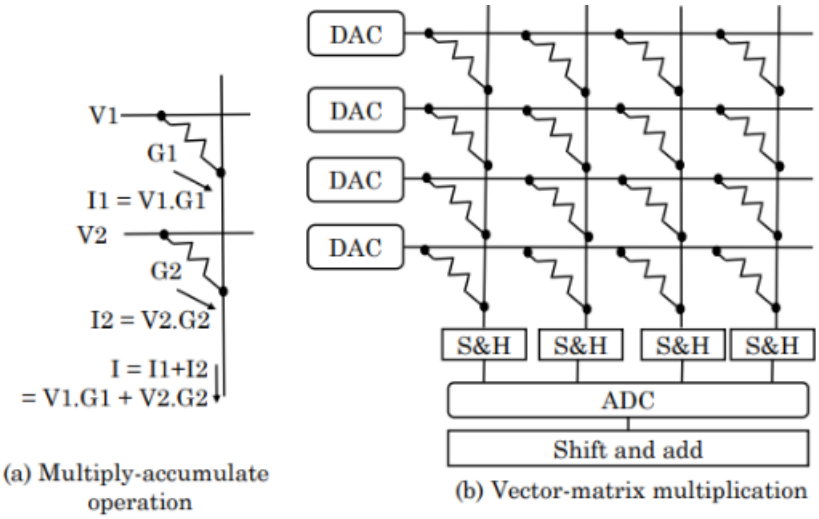

Figure 1.1: Basic Structure of the memristor[1](pictures taken from *"A Survey of ReRAM-Based Architectures for Processing-In-Memory and Neural Networks by Sparsh Mittal"*)

that perform computations within the memory itself, eliminating the energy-intensive and time-consuming data movement inherent in traditional designs. This approach, which leverages resistive switching devices, aims to radically subvert the von Neumann model by performing calculations in situ, exactly where the data resides. It is analogous to the human brain, where information is processed in interconnected networks of neurons and synapses without a distinct separation between computation and memory.

Chapter 2

Background and Existing work

In ReRAM-based accelerators, an input buffer stores incoming data, which is converted to analog voltages by DACs and processed in a ReRAM crossbar. The resulting analog currents are digitized by ADCs, accumulated via shift-and-add, and stored in an output buffer for the next layer. This flow minimizes off-chip data transfers and leverages in-memory computing for efficiency.Figure 2.1 shows a detailed design of a ReRAM based Nueral Network

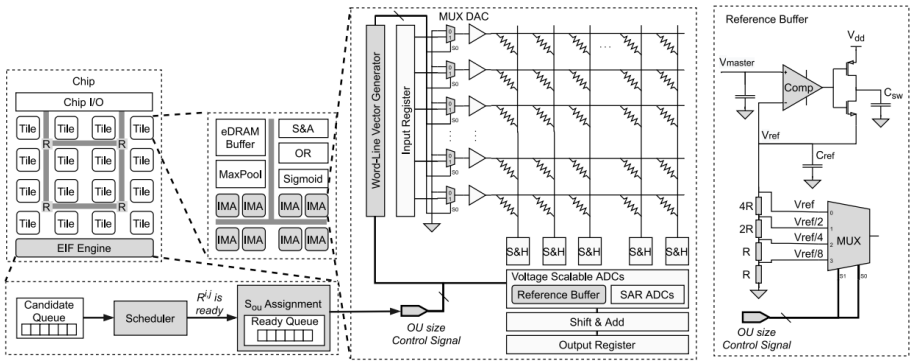

Figure 2.1: Configurable ReRAM-based NN accelerator([2](Pictures taken from *"An Energy-Efficient Inference Engine for a Configurable ReRAM-Based Neural Network Accelerator Yang-Lin Zheng, Wei-Yi Yang, Ya-Shu Chen , Member, IEEE, and Ding-Hung Han"*)

2.1 Step-by-Step Data Flow in a ReRAM-Based Accelerator

Step 1: Input Image Preprocessing

Images are quantized into fixed-bit (e.g., 8-bit) values for ReRAM compatibility, balancing reduced precision with sufficient neural network accuracy.

Step 2: Digital Input to Analog Conversion (DAC)
Quantized pixels are converted to analog voltages via DACs, with each channel (e.g., RGB) processed separately to preserve color fidelity.

Step 3: Writing Data into ReRAM Cells
Neural network weights are stored as cell conductances. Input voltages are applied to word lines, where each cell's conductance represents a weight.

Step 4: Matrix-Vector Multiplication (MVM) in the Crossbar
Ohm's and Kirchhoff's laws enable analog multiply-accumulate operations in the crossbar, where input voltages multiply weight conductances to produce output currents.

Step 5: Analog-to-Digital Conversion (ADC)
Summed currents at each column are digitized by ADCs, yielding digital outputs for further processing.

Step 6: Accumulation and Shift-and-Add Processing
Partial sums from multiple crossbars are merged in the digital domain to finalize layer outputs while preserving accuracy.

Step 7: Output Buffer and Next Layer Processing
Final outputs are stored in an output buffer and passed to the next layer until the network's final prediction is generated.

2.2 Mixed-Bit Operations in ReRAM-Based Accelerators

2.2.1 Use of Mixed-Bit Operations

Mixed-bit operations reduce energy consumption and boost speed by using lower precision when feasible. Early layers typically require higher precision, whereas later layers can tolerate lower precision, reducing power usage without sacrificing accuracy..

2.2.2 Implementation of Mixed-Bit Operations

The system dynamically adjusts the precision for weights and activations in ReRAM cells from 1 to 8 bits. Voltage levels correspond to the selected bitwidth, and each layer's precision is optimized based on its sensitivity. This strategy enables high-throughput, energy-efficient computation with minimal accuracy loss.

2.2.3 Benefits of Mixed-Bit Operations

Using lower precision for weights while selectively increasing activation precision reduces memory access and power consumption, thereby boosting throughput and accommodating various architectures (e.g., AlexNet, VGGNet). This balance maintains accuracy while enhancing energy efficiency—essential for real-time edge applications.

2.2.4 Challenges and Solutions

While mixed-bit operations can risk accuracy loss if precision is too low, Neural Architecture Search (NAS) optimizes precision on a per-layer basis to minimize performance degradation. Although hardware complexity increases with multiple bitwidths, standardizing key components (such as fixing precision in certain layers) addresses this challenge. Additionally, ReRAM variability is mitigated through error correction and carefully designed charge-based methods.

2.3 Neural Architecture Search (NAS)

Neural Architecture Search (NAS) automates network design by exploring various architectures and configurations for tasks such as image classification, effectively balancing performance and efficiency. NAS optimizes bit precision by tailoring the precision for weights and activations using lower bitwidths to save energy when possible while accepting a potential accuracy cost, and assigning higher precision where necessary. It begins by defining a search space that includes different bitwidth options for weights (e.g., 1-bit, 2-bit, 4-bit) and activations (e.g., 2-bit, 4-bit, 8-bit), along with layer-specific settings (e.g., layer 1 with 4-bit weights, layer 2 with 2-bit activations). This systematic approach identifies the optimal configuration for ReRAM-based mixed-bit accelerators while managing the energy-accuracy tradeoff.

2.3.1 How NAS is Used in the Proposed System

In the proposed mixed-bit ReRAM accelerator, NAS determines the optimal bit precision for each CNN layer. Weights, which are fixed and less sensitive to precision, typically use lower bitwidths (e.g., 1-bit) to conserve energy. In contrast, activations—being dynamic and critical for output quality—are assigned higher bitwidths (e.g., 4-bit or 8-bit) in layers where accuracy is most sensitive. By automating these layer-specific configurations, NAS ensures that each layer employs just enough precision to maintain high accuracy while minimizing energy consumption, thereby achieving a balanced tradeoff between energy savings and task accuracy.

2.4 Twin Range Quantization

Quantization reduces data precision to accelerate computations and conserve energy. In ReRAM-based accelerators, Analog-to-Digital Converters (ADCs) consume significant power at high precision. Traditional uniform quantization applies the same bitwidth across all values, which is inefficient since most data are small while only a few values are large.

2.4.1 Twin Range Quantization Mechanism

Twin Range Quantization (TRQ) divides the data domain into two segments. The **narrow range** handles small, frequently occurring values (e.g., 0–15) using higher precision (e.g., 4 bits or more) to preserve detail, while the **wide range** deals with larger, less frequent values (e.g., 16–255) using lower precision (e.g., 3 bits or less) with scaling to approximate their original magnitude. When a value is processed by the ADC, narrow range values are quantized directly at high precision, whereas wide range values are scaled down, quantized with fewer bits, and then rescaled during computation.

Example: Suppose the ADC receives two values: 10 and 200. The value 10 falls in the narrow range and is quantized with a precision of 4-bits or more, preserving its detail. In contrast, 200 is in the wide range, so it is scaled, quantized with a precision of 3-bits or less, and later rescaled to approximate its original magnitude. This dual-range approach preserves accuracy for common small values while reducing ADC overhead for infrequent large values, resulting in significant energy savings without degrading overall neural network performance.

Chapter 3

Methdology

Twin Range Quantization (TRQ) and Mixed-Bit ReRAM-Based In-Memory Computing (IMC), optimized via Neural Architecture Search (NAS), are complementary techniques that boost energy efficiency and computational accuracy in edge AI applications. Mixed-Bit ReRAM with NAS dynamically adjusts the bit-width for each CNN layer (from 1-bit to 8-bit) to balance accuracy with power efficiency, while TRQ improves ADC efficiency by categorizing analog signals into two ranges: a narrow range for small, frequent values (processed at high precision) and a wide range for large, infrequent values (processed with fewer ADC steps).
By integrating these methods, the system achieves synergistic gains in both power efficiency and computation speed, reducing ADC power consumption by 42–62% while preserving near-original model accuracy. In this hybrid approach, the ReRAM crossbar stores network weights as conductance values for in-memory matrix-vector multiplications (MVMs), and a configurable SAR-ADC, guided by TRQ, dynamically adjusts its precision. This integration minimizes redundant computations, enhances throughput, and enables ultra-low-power AI acceleration—ideal for edge AI, IoT, and autonomous systems.

3.1 NAS Optimization with TRQ-Aware Quantization

In figure 3.1 the proposed system of integration of mixed-bit ReRAM with Twin Range Quantization is shown. Neural Architecture Search (NAS) optimizes each neural network layer's bit-width based on its sensitivity, allocating more bits to layers where high precision is critical for accuracy and fewer bits to less sensitive layers to conserve energy. Twin Range Quantization (TRQ) enhances this process by incorporating ADC power efficiency into the NAS cost function, thereby minimizing unnecessary ADC operations. By extending NAS to account for both weight quantization and ADC energy consumption, the system achieves a more power-efficient model with reduced computational overhead

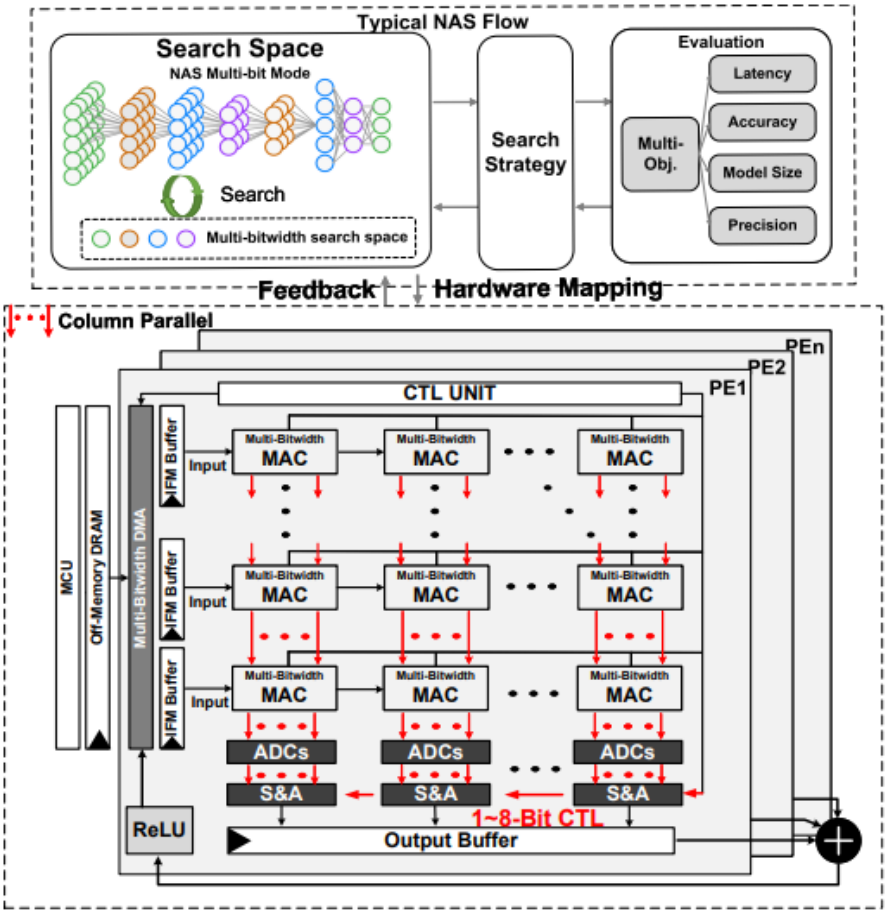

Figure 3.1: Proposed ReRAM-based CIM accelerator for mixed-bit CNN with TRQ([3](Pictures taken from*"An Energy-Efficient Mixed-Bit ReRAM-based Computing-in-Memory CNN Accelerator with Fully Parallel Readout by Dingbang Liu, Wei Mao, Haoxiang Zhou, Jun Liu, Qiuping Wu, Haiqiao Hong and Hao Yu School of Microelectronics Southern University of Science and Technology Shenzhen, China"*)

while maintaining network accuracy.

3.1.1 Implementation Steps

Step 1: Train a Mixed-Bit CNN Model Using NAS

Neural Architecture Search (NAS) automatically determines the optimal bit-width for each CNN layer by balancing accuracy and power efficiency. It assigns lower bit-widths (e.g., 2-bit or 4-bit) to shallow layers performing simpler tasks like edge detection, while deeper layers involved in classification receive higher precision (e.g., 8-bit). NAS defines a search space with options such as 1-bit, 2-bit, 4-bit, and 8-bit for each layer, and employs advanced optimization techniques—such as Differentiable NAS or Reinforcement Learning—to identify a configuration that minimizes accuracy loss while maximizing energy savings. Once the optimal architecture is determined, the model is trained using stan-

dard gradient-based methods, resulting in a power-efficient, mixed-bit CNN well-suited for edge AI applications.

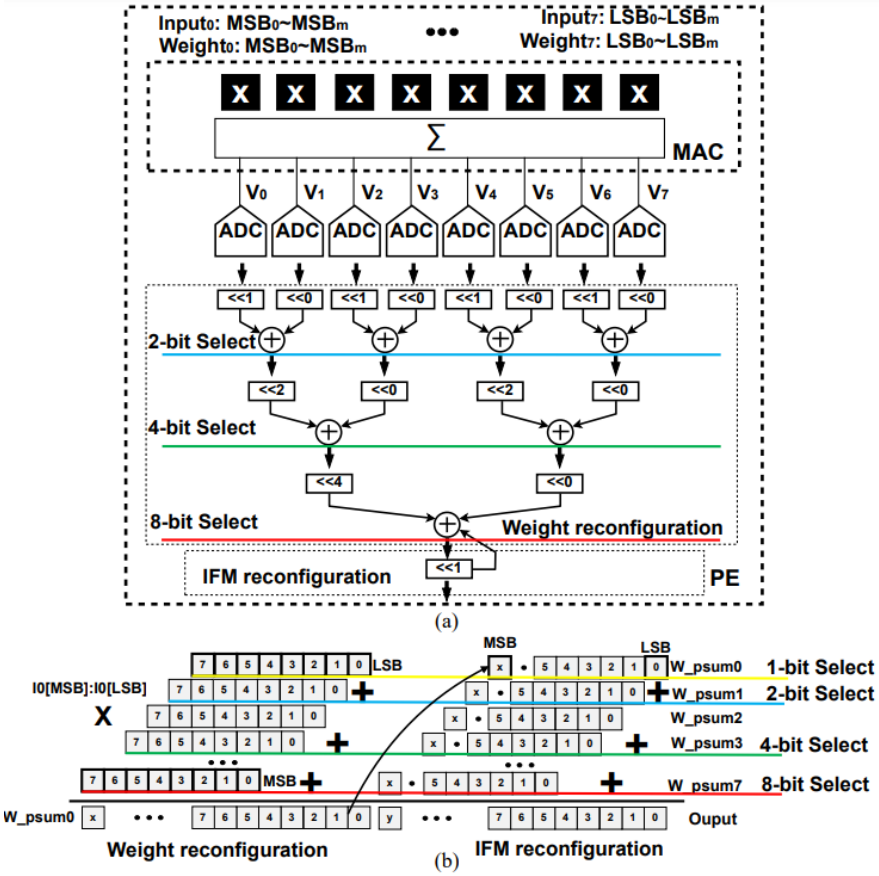

Figure 3.2: Mixed-bit operation method of the proposed system: (a) structure of mixed-bit PE and (b) mixed-bit dataflow([3]Pictures taken from *"An Energy-Efficient Mixed-Bit ReRAM-based Computing-in-Memory CNN Accelerator with Fully Parallel Readout by Dingbang Liu, Wei Mao, Haoxiang Zhou, Jun Liu, Qiuping Wu, Haiqiao Hong and Hao Yu School of Microelectronics Southern University of Science and Technology Shenzhen, China")*

Step 2: Analyze ADC Redundancy Using TRQ

n ReRAM-based systems, full-precision digitization by ADCs is often unnecessary. Twin Range Quantization (TRQ) addresses this inefficiency by classifying analog outputs into two zones:

- **R1 (High-Precision Zone):** Small, frequent values processed with high ADC resolution.
- **R2 (Low-Precision Zone):** Large, infrequent values processed with reduced ADC steps.

By analyzing the bit-line output distribution, TRQ dynamically adjusts ADC resolution, reducing power consumption by 42–62% while maintaining accuracy.

Step 3: Retrain the NAS-Optimized Model with TRQ-Aware Quantization
Once NAS determines the optimal architecture, the model is retrained using TRQ-aware quantization to further reduce ADC power. During this phase, TRQ applies early stopping for large values (R2) while preserving high precision for small values (R1). Evaluations on datasets such as ImageNet and CIFAR-10 confirm that this retraining yields significant power savings with minimal accuracy loss, fully optimizing the CNN for energy-efficient performance.

3.1.2 Mixed-Bit Mapping and Joint Hardware Implementation

Once NAS-optimized CNN layers are mapped to ReRAM crossbars, Twin Range Quantization (TRQ) splits the outputs into two zones R1 for small values and R2 for large values—reducing ADC power consumption by 42–62%. In this hybrid design, in-memory multiply-accumulate (MAC) operations, DAC inputs, and dynamically adjusted ADC precision work together to preserve accuracy while minimizing energy usage. A Mixed-Bit ReRAM architecture integrates a configurable SAR-ADC, optimized by TRQ, which selectively applies ADC steps based on the R1/R2 classification. Additionally, energy-aware hardware scheduling further enhances efficiency. This unified approach maximizes energy savings without compromising CNN performance, making it ideal for power-constrained edge AI applications.

3.2 Workflow of the Designed System

Simulation:
Design mixed-bit ReRAM cells (1–8 bits per cell) integrated with Twin Range Quantization (TRQ) to split weights and activations into high- and low-precision ranges. High-level simulations (MATLAB/Python) validate TRQ's impact on accuracy and performance. Simultaneously, device behavior and circuit-level models (including ReRAM I–V characteristics, crossbar, and peripheral circuits) are simulated to evaluate read/write times, energy consumption, and reliability under imperfections.

Validation with SPICE and Circuit Simulations:
Transistor-level simulations (Spectre/HSPICE) verify line drivers, sense amps, and ADC/DAC blocks using compact ReRAM models. Corner analysis and reliability tests across TT, SS, and FF process corners and temperatures assess read margins, set/reset thresholds, and endurance.

Digital Control and Interfaces:
Data flow for mixed-bit ReRAM and TRQ is defined in HDL (Verilog/VHDL) using FSMs for sequencing and TRQ modules for dynamic bit-width adaptation. The design

is synthesized and tested on an FPGA (e.g., Vivado/Quartus) with SoC-based firmware, verified through testbenches and real-time inference.

Prototype with PCB Design:
After FPGA validation, a custom PCB is developed to host the ReRAM array and controller. Schematic capture and layout are performed using tools like Altium Designer or Eagle, ensuring stable power rails and signal integrity (analyzed with tools such as HyperLynx). Thermal and board-level simulations finalize the design for fabrication (Gerber files and BOM).

ASIC Design and Fabrication:
The verified HDL is synthesized for ASIC implementation using Synopsys Design Compiler or Cadence Genus, followed by Place-and-Route with Cadence Innovus or Synopsys IC Compiler. Physical verification (DRC, LVS, and post-layout simulations) confirms performance, with early prototyping via MPW runs before full-scale production.

Integration and Testing:
Post-fabrication, the ASIC or PCB solution is characterized in a lab using oscilloscopes, logic analyzers, and SMUs. Calibration routines adjust write pulses and read thresholds to account for device variability. Finally, the accelerator is integrated with host processors and peripherals for end-to-end application tests (e.g., image classification, speech recognition), ensuring performance meets design specifications.

Chapter 4

Results

The design begins with an ISAAC-inspired architecture that employs 128×128 crossbar arrays of single-bit ReRAM cells as the computing engine. To support mixed-bit operations (ranging from 1 to 8 bits), each MAC cell is configured in an 8×1 arrangement, enabling flexible precision across layers. Each MAC array consists of 2048×256 units that process 8-bit inputs and weights to generate 16-bit partial sums.
Twin Range Quantization (TRQ) partitions weights and activations into high- and low-precision ranges using two parameters, α and β. An algorithmic parameter search refines the TRQ settings by analyzing layer-specific data distributions and optimizing ADC parameters to minimize quantization error while preserving accuracy.
The system is fabricated on a 28-nm process, with ReRAM devices specified at 1 MΩ (HRS) and 100 kΩ (LRS), and device variability ranging from 5% to 30%. Digital circuits, including an 8-bit SAR ADC, are synthesized using 45-nm libraries, while the ReRAM core operates at 20 MHz. All modules are integrated and simulated using Cadence AMS for system-level verification.
Calibration is performed with a small set of training images to fine-tune the ADC scaling factors for 8-bit symmetric uniform quantization. Benchmarking with NAS-optimized CNNs (e.g., on ImageNet) using DNN+NeuroSim demonstrates a maximum compression rate of 2.31 and an inference accuracy of 69.84%, highlighting the enhanced energy efficiency and performance of the design.

4.1 Algorithm Evaluation

ADC quantization bit-width is defined such that "f/f" denotes the full-precision model, while "8/f" indicates 8-bit quantization for both weights and activations. The sequence "8,7,6,5,4" represents the possible upper limits on ADC resolution, with 8 being the highest and 4 the lowest. In figure 4.1 a 28-nm implementation with ResNet18 is shown in which Twin Range Quantization (TRQ) method achieves 92.9% accuracy using a 4-bit ADC, whereas a uniform ADC requires at least 7 bits to achieve comparable accuracy.

Additionally, TRQ reduces ADC dynamic reading energy by approximately 40–60% and significantly lowers overall ADC power consumption

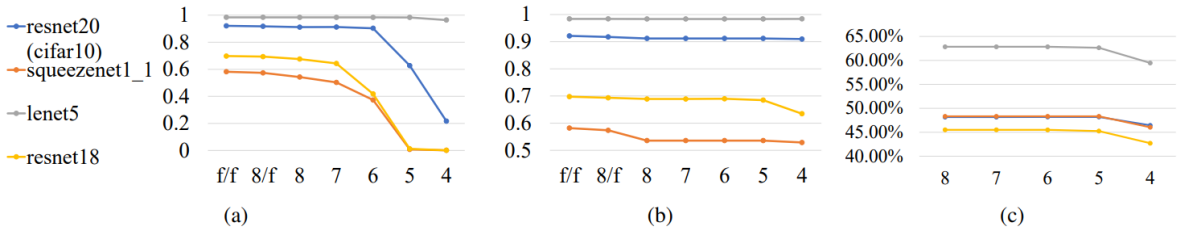

Figure 4.1: Evaluation of algorithm (a) Accuracy w.r.t. ADC resolution without TRQ and (b) with TRQ; (c) Remained A/D operations with TRQ([4](Picturs take from *"Algorithm-hardware co-design for Energy-Efficient A/D conversion in ReRAM-based accelerators Chenguang Zhang1,Zhihang Yuan1,Xingchen Li1,Guangyu Sun1, School of Integrated Circuits, Peking University, Beijing, China")*

4.2 Hardware Evaluation

Figure 4.2 shows ADC dynamic reading efficiency compared with full-precision 8-bit ADC operations, TRQ reduces ADC energy consumption by an average of 40–60% providing a 1.6 to 2.3 times improvement—while maintaining overall energy efficiency by adjusting the batch size across models. Furthermore, when comparing a 4-bit TRQ ADC to a uniform ADC that uses the minimum bit-width required for similar accuracy, TRQ significantly lowers ADC power consumption even though the original ADC bit-width remains unchanged.

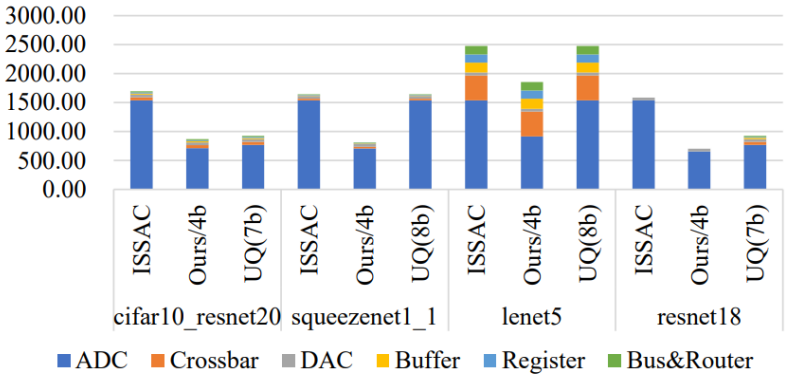

Figure 4.2: Power breakdown of ReRAM-based accelerator.([4]Picturs take from *"Algorithm-hardware co-design for Energy-Efficient A/D conversion in ReRAM-based accelerators Chenguang Zhang1,Zhihang Yuan1,Xingchen Li1,Guangyu Sun1, School of Integrated Circuits, Peking University, Beijing, China"*)

Chapter 5

Conclusion

This paper introduces an energy-efficient quantization scheme for ReRAM-based neural network accelerators that combines the TRQ algorithm with a co-design of hardware and software. Our approach targets the reduction of redundant A/D conversion operations, thereby improving power efficiency while incurring only negligible losses in prediction accuracy.

A key strength of the design is its seamless integration into existing ReRAM-based accelerators. It requires only a minor modification to the ADC's digital logic, leaving the analog circuitry unchanged. As a result, the system's original ADC resolution is maintained while benefiting from reduced power consumption. Moreover, our design is completely transparent to the deep neural network models; no retraining is needed, and there is no added overhead for encoding or decoding.

The flexibility of our approach is another significant advantage. It can adapt to a wide range of DNN architectures and complement other hardware optimizations and model compression techniques without requiring further modifications. Overall, our method provides an effective, low-cost solution for enhancing the energy efficiency of ReRAM-based neural network accelerators while ensuring robust performance across diverse applications.

Bibliography

[1] S. Mittal, "A survey of reram-based architectures for processing-in-memory and neural networks," in *Proceedings of the [Machine Learning and Knowledge Extraction]*, IEEE, 2020.

[2] Y.-L. Zheng, W.-Y. Yang, Y.-S. Chen, and D.-H. Han, "An energy-efficient inference engine for a configurable reram-based neural network accelerator," in *Proceedings of [Conference Name]*, 2020.

[3] D. Liu, W. Mao, H. Zhou, J. Liu, Q. Wu, H. Hong, and H. Yu, "An energy-efficient mixed-bit reram-based computing-in-memory cnn accelerator with fully parallel readout," in *Proceedings of [Conference Name]*, 2020. School of Microelectronics, Southern University of Science and Technology, Shenzhen, China.

[4] C. Zhang, Z. Yuan, X. Li, and G. Sun, "Algorithm-hardware co-design for energy-efficient a/d conversion in reram-based accelerators," in *Proceedings of [Conference Name]*, 2021. School of Integrated Circuits, Peking University, Beijing, China; School of Computer Science, Peking University, Beijing, China; Beijing Advanced Innovation Center for Integrated Circuits, Beijing, China.

[5] Y. He, Y. Wang, Y. Wang, H. Li, and X. Li, "An agile precision-tunable cnn accelerator based on reram," in *Proceedings of [Conference Name]*, 2020. SKLCA, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China; University of Chinese Academy of Sciences, Beijing, China; Peng Cheng Laboratory, Shenzhen, China.

[6] Y.-W. Kang, C.-F. Wu, Y.-H. Chang, T.-W. Kuo, and S.-Y. Ho, "On minimizing analog variation errors to resolve the scalability issue of reram-based crossbar accelerators," *IEEE Transactions on [Journal Name]*, 2019.

[7] P. Joshi and H. Rahaman, "A comprehensive review on reram-based accelerators for deep learning," *Journal of [Journal Name]*, 2021. Department of Information and Technology, Indian Institute of Engineering Science and Technology, Howrah, India.

[8] M. Mao, X. Sun, X. Peng, S. Yu, and C. Chakrabarti, "A versatile reram-based accelerator for convolutional neural networks," in *Proceedings of [Conference Name]*, 2020.

School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA.

[9] Y. Long, T. Na, and S. Mukhopadhyay, "Reram-based processing-in-memory architecture for recurrent neural network acceleration," in *Proceedings of [Conference Name]*, 2021.

[10] B. Li, Y. Wang, and Y. Chen, "Hitm: High-throughput reram-based pim for multimodal neural networks," in *Proceedings of [Conference Name]*, 2021. Capital Normal University, Beijing, China; Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing, China; Duke University, Durham, NC, USA.

[11] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, 2018.

[12] Y. Chen, "Reram: History, status, and future," *IEEE Spectrum*, 2017.

[13] H. Akinaga and H. Shima, *Resistive Random Access Memory (ReRAM) Based on Metal Oxides*. [Publisher Name], 2016.