Towards Reliable and Energy-Efficient RRAM based Discrete Fourier Transform Accelerator

Jianan Wen*, Andrea Baroni*, Eduardo Perez*†, Max Uhlmann*, Markus Fritscher*†,
Karthik KrishneGowda*, Markus Ulbricht*, Christian Wenger*† and Milos Krstic*‡
*IHP - Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany
†Brandenburgische Technische Universität Cottbus-Senftenberg, Cottbus, Germany
‡Universität Potsdam, Potsdam, Germany

***Abstract*—The Discrete Fourier Transform (DFT) holds a prominent place in the field of signal processing. The development of DFT accelerators in edge devices requires high energy efficiency due to the limited battery capacity. In this context, emerging devices such as resistive RAM (RRAM) provide a promising solution. They enable the design of high-density crossbar arrays and facilitate massively parallel and in situ computations within memory. However, the reliability and performance of the RRAM-based systems are compromised by the device non-idealities, especially when executing DFT computations that demand high precision. In this paper, we propose a novel adaptive variability-aware crossbar mapping scheme to address the computational errors caused by the device variability. To quantitatively assess the impact of variability in a communication scenario, we implemented an end-to-end simulation framework integrating the modulation and demodulation schemes. When combining the presented mapping scheme with an optimized architecture to compute DFT and inverse DFT(IDFT), compared to the state-of-the-art architecture, our simulation results demonstrate energy and area savings of up to 57% and 18%, respectively. Meanwhile, the DFT matrix mapping error is reduced by 83% compared to conventional mapping. In a case study involving 16-quadrature amplitude modulation (QAM), with the optimized architecture prioritizing energy efficiency, we observed a bit error rate (BER) reduction from 1.6e-2 to 7.3e-5. As for the conventional architecture, the BER is optimized from 2.9e-3 to zero.**

***Index Terms*—Discrete Fourier Transform, resistive RAM, in-memory computing, energy-efficient systems, variability.**

I. Introduction

The DFT is a fundamental digital signal processing algorithm with a broad range of applications, such as audio and image processing, telecommunication, medical imaging [1], and accelerating convolutional operations [2]. The DFT and its inverse form IDFT convert the data between time and frequency domains. In order to address the high computational complexity of DFT and achieve efficient design, the fast Fourier Transform (FFT) is proposed, which scales the complexity from $O(N^2)$ down to $O(N\log N)$. However, the FFT processors deployed at the edge are still challenged by the requirement to deliver high performance and energy efficiency.

Emerging memristive technologies such as RRAM have been actively investigated due to the benefits of compact cell area, high storage density, and CMOS compatibility [3]. As CMOS technology nears the physical limits, the in-memory computing paradigm, powered by these emerging technologies, offers a way to enhance system performance and energy efficiency by preventing massive, costly data transfer intrinsic to the von Neumann architecture.

RRAM crossbar arrays enable efficient vector-matrix multiplications (VMMs). The values in the matrix are stored as the conductance in the memory array, and the computations can be performed locally. Several works designed and fabricated the RRAM systems to accelerate deep neural networks (DNNs) [4], which achieve significant performance and energy efficiency improvement and indicate promising potential. Similar to DNN accelerators, the efficient DFT design relies heavily on realizing the low-powered VMM, where the RRAM crossbar arrays emerge as competitive candidates for hardware implementations.

Previous works studied the design of DFT/IDFT accelerators with emerging devices for various applications. The architectures of memristive crossbar-based DFT and multiple-input multiple-output (MIMO) detectors for baseband processing were proposed in [5], and the simulation results indicated the DFT implementation based on memristive devices outperforms the conventional CMOS-based FFT in terms of latency and energy efficiency. However, some detailed hardware information was not described, such as the type of used devices, DFT matrix bitwidth, and parameters of the sensing circuitry. In [6], a circuit design realizing Discrete Cosine Transform (DCT) based on memristive devices was presented to perform the image compression. The simulation results implied the feasibility of utilizing emerging technologies to achieve high-quality image compression with improved computational speed, but the system hardware overhead was not discussed. Moreover, neither of the aforementioned research studies utilized the device features or considered the effects of variability derived from characterized devices for evaluations. This abstraction potentially compromises the hardware implementation results.

The first hardware implementation of the RRAM-based DFT accelerator featuring medical image reconstruction was reported in [1]. To address the high energy consumption caused by the peripherals, a matrix concatenation method was proposed. However, the positive and negative parts of the DFT coefficients are stored in differential device pairs to enable the signed computation, which sacrifices energy efficiency and area footprint.

Based on the previous work in [1], we propose a novel RRAM crossbar-based architecture to compute DFT/IDFT that integrates the differential pairs. The feasibility and re-

979-8-3503-4859-0/DATE24/© 2024 EDAA

liability of this architecture are bolstered by the proposed adaptive variability-aware crossbar mapping scheme, which substantially reduces the mapping inaccuracies stemming from quantization and device variabilities. An end-to-end simulation framework is implemented to assess the influence of inaccuracies in communication systems and the system hardware overhead. In a case study with 128-point DFT and 16-QAM, the proposed architecture reduces area utilization and energy consumption by 18% and 57%, respectively. With the complementary mapping scheme, the BER is optimized from 1.6e-2 to 7.3e-5.

II. RRAM Crossbar Array for DFT Computation

A. 1T1R Devices and Measurements

RRAM is a memristive technology that stores information as device conductance under stimulating electrical stress. With its high integration density, low energy consumption, and robust data retention, RRAM is a promising choice to promote the performance of storage and computing systems [3].

To scale up the single device to crossbar arrays, each RRAM device is typically connected to an NMOS transistor, enabling device selection, known as 1T1R structure, as depicted in Fig. 1. By applying the positively or negatively polarized pulses between the bitline (BL) and sourceline (SL), the device can be programmed to the high resistance state (HRS) and to the low resistance state (LRS) by respectively destroying and growing the conductive filaments. As a consequence, the conductivity in the metal-insulator-metal (MIM) stack is altered. During the transition from HRS to LRS, by assigning different gate voltages on the wordline (WL) to manipulate the compliance currents, the devices can be programmed to distinct LRSs achieving the multilevel storage [7].

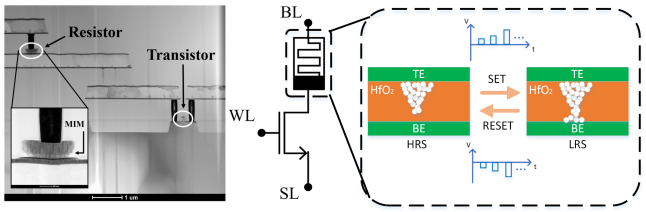


Fig. 1. Cross-sectional TEM image of fabricated 1T1R device (left), the schematic (middle), and the state transition between HRS and LRS (right).

In this study, to fully consider the device properties and non-idealities in system-level evaluation, we characterize the RRAM devices fabricated at IHP. The RRAM devices are monolithically integrated with the standard 130 nm CMOS technology in the back-end-of-line (BEOL) process. Four pages with a total amount of 256 devices are randomly selected and experimentally characterized from a packaged 4k-bit RRAM memory chip [7]. Fig. 2 displays the distribution of 256 devices programmed to HRS, LRS1, LRS2, and LRS3, corresponding to 2-bit storage. During the programming process, the write-verify algorithm is deployed to precisely control the device conductance and minimize the device-to-device variability [7].

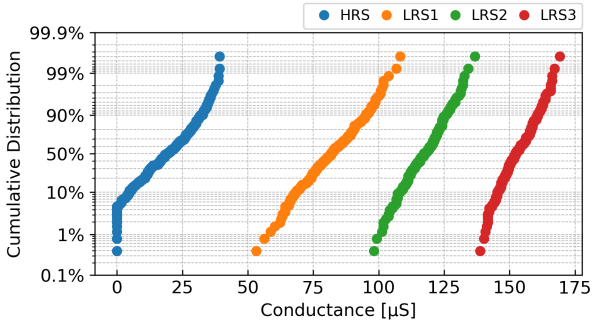


Fig. 2. Cumulative conductance distribution of 256 1T1R devices programmed to four resistance states.

B. Discrete Fourier Transform with RRAM Crossbar

The DFT algorithm converts the discrete time-domain signals $\mathbf{x}$ with N complex numbers into the corresponding frequency-domain representations $\mathbf{X}$ with the same size. The computation can be described as:

$$\mathbf{X} = \mathbf{W}\mathbf{x}, \tag{1}$$

where $\mathbf{W}$ is the DFT transformation matrix with the size of $N \times N$, which can be expressed as:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{N-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \cdots & w^{(N-1)(N-1)} \end{bmatrix} \tag{2}$$

and w is defined as $e^{-\frac{2\pi i}{N}}$. To accelerate the DFT computation with RRAM crossbars, the coefficients are mapped as the conductance of devices. Since DFT coefficients contain real and imaginary parts, $\mathbf{W}$ is decomposed to two matrices $\mathbf{W}_{\mathrm{Re}}$ and $\mathbf{W}_{\mathrm{Im}}$ that consist of either only real or imaginary parts, where $\mathbf{W} = \mathbf{W}_{\mathrm{Re}} + \mathbf{W}_{\mathrm{Im}}$. Besides, both positive and negative input values should be computed by the DFT/IDFT modules in the digital communication systems. Therefore, Eq. 1 can be rewritten as:

$$\begin{bmatrix} \mathbf{X}_{\mathrm{Re}} \\ \mathbf{X}_{\mathrm{Im}} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{\mathrm{Re}} & -\mathbf{W}_{\mathrm{Re}} & -\mathbf{W}_{\mathrm{Im}} & \mathbf{W}_{\mathrm{Im}} \\ \mathbf{W}_{\mathrm{Im}} & -\mathbf{W}_{\mathrm{Im}} & \mathbf{W}_{\mathrm{Re}} & -\mathbf{W}_{\mathrm{Re}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_{\mathrm{Re}+} \\ \mathbf{x}_{\mathrm{Re}-} \\ \mathbf{x}_{\mathrm{Im}+} \\ \mathbf{x}_{\mathrm{Im}-} \end{bmatrix} \tag{3}$$

Associating the mathematical formulation with the implementation of crossbar arrays, the DFT can be computed with electrical quantities as:

$$\begin{bmatrix} \mathbf{I}_{\mathrm{Re}} \\ \mathbf{I}_{\mathrm{Im}} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{\mathrm{Re}} & -\mathbf{G}_{\mathrm{Re}} & -\mathbf{G}_{\mathrm{Im}} & \mathbf{G}_{\mathrm{Im}} \\ \mathbf{G}_{\mathrm{Im}} & -\mathbf{G}_{\mathrm{Im}} & \mathbf{G}_{\mathrm{Re}} & -\mathbf{G}_{\mathrm{Re}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{V}_{\mathrm{Re}+} \\ \mathbf{V}_{\mathrm{Re}-} \\ \mathbf{V}_{\mathrm{Im}+} \\ \mathbf{V}_{\mathrm{Im}-} \end{bmatrix} \tag{4}$$

$\mathbf{V}$ are the input voltages applied to the crossbar arrays. $\mathbf{G}$ are the corresponding conductance values mapped to the crossbar arrays. $\mathbf{I}$ are the accumulated currents, which are the results of DFT.

IDFT computes the conversion of discrete frequency-domain signals to the time domain. The transformation matrix for IDFT is the conjugate transpose of the DFT matrix. Additionally, to ensure the recovery of the original signals after performing both transforms sequentially without scaling,

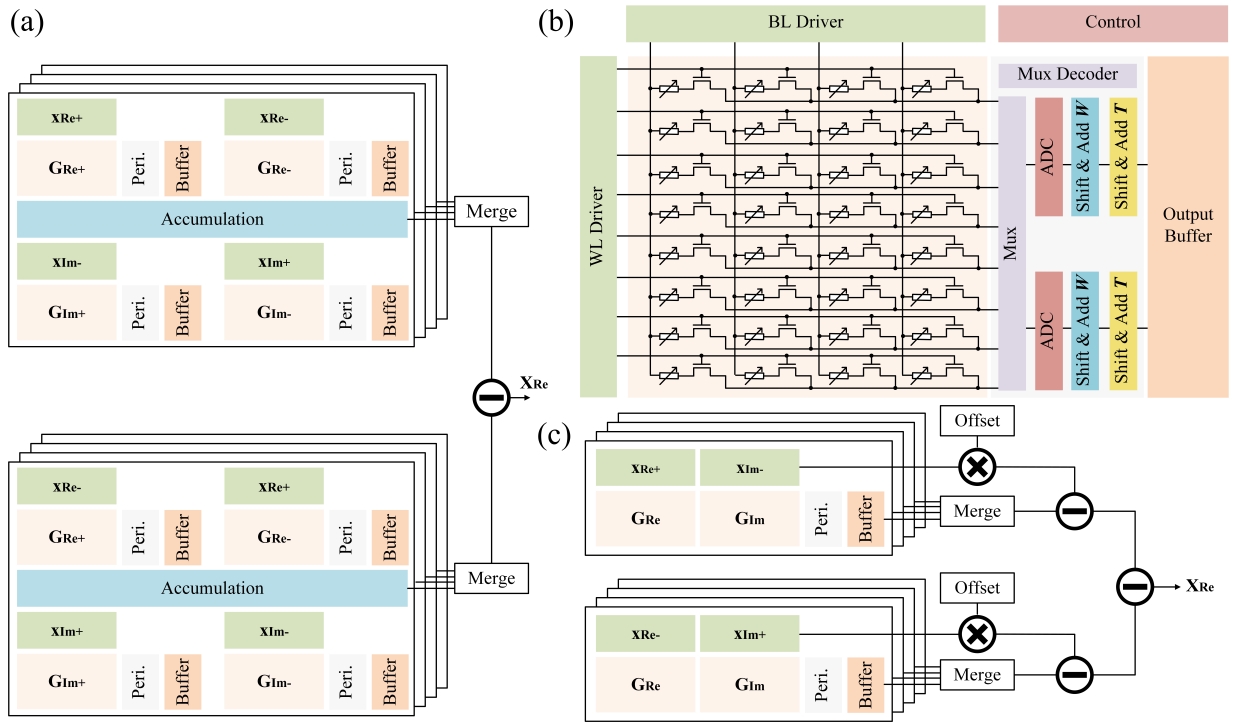


Fig. 3. System architecture of the RRAM-based DFT/IDFT accelerator. (a) The conventional architecture where the signed coefficients are realized with differential device pairs (only the computation for $\mathbf{X}_{\mathrm{Re}}$ is shown). (b) RRAM crossbar with peripherals to compute analog VMMs in accelerators. (c) The proposed system architecture integrates positive and negative coefficients on the same crossbars, and the peripherals are shared by two crossbars to save the hardware cost.

a normalization factor N is introduced to the IDFT matrix. The IDFT can be computed with RRAM crossbar arrays as:

$$\begin{bmatrix}\mathbf{I}_{\mathrm{Re}}\\ \mathbf{I}_{\mathrm{Im}}\end{bmatrix} = \frac{1}{N}\begin{bmatrix}\mathbf{G}_{\mathrm{Re}} & -\mathbf{G}_{\mathrm{Re}} & \mathbf{G}_{\mathrm{Im}} & -\mathbf{G}_{\mathrm{Im}}\\ -\mathbf{G}_{\mathrm{Im}} & \mathbf{G}_{\mathrm{Im}} & \mathbf{G}_{\mathrm{Re}} & -\mathbf{G}_{\mathrm{Re}}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}_{\mathrm{Re}+}\\ \mathbf{V}_{\mathrm{Re}-}\\ \mathbf{V}_{\mathrm{Im}+}\\ \mathbf{V}_{\mathrm{Im}-}\end{bmatrix} \tag{5}$$

III. Proposed Architecture and Algorithm

A. System Architecture

DFT coefficients comprise positive and negative values. In prior works, signed coefficients are either directly mapped to the separated crossbar arrays [5] or represented with a form of differential pairs [1]. After performing VMMs in the analog domain, the outputs from the crossbars are subtracted either in the digital or analog domain.

To be mapped on differential pairs, the complex coefficients $\mathbf{G}_{\mathrm{Re}}$ and $\mathbf{G}_{\mathrm{Im}}$ are decomposed further as $\mathbf{G}_{\mathrm{Re}+}$, $\mathbf{G}_{\mathrm{Re}-}$, $\mathbf{G}_{\mathrm{Im}+}$ and $\mathbf{G}_{\mathrm{Im}-}$, where $\mathbf{G}_{\mathrm{Re/Im}} = \mathbf{G}_{(\mathrm{Re/Im})+} - \mathbf{G}_{(\mathrm{Re/Im})-}$. The Eq. 4 can be formulated as:

$$\begin{aligned}\begin{bmatrix}\mathbf{I}_{\mathrm{Re}}\\ \mathbf{I}_{\mathrm{Im}}\end{bmatrix} = &\begin{bmatrix}\mathbf{G}_{\mathrm{Re}+} & \mathbf{G}_{\mathrm{Re}-} & \mathbf{G}_{\mathrm{Im}-} & \mathbf{G}_{\mathrm{Im}+}\\ \mathbf{G}_{\mathrm{Im}+} & \mathbf{G}_{\mathrm{Im}-} & \mathbf{G}_{\mathrm{Re}+} & \mathbf{G}_{\mathrm{Re}-}\end{bmatrix}\cdot\mathbf{V}\\ &-\begin{bmatrix}\mathbf{G}_{\mathrm{Re}-} & \mathbf{G}_{\mathrm{Re}+} & \mathbf{G}_{\mathrm{Im}+} & \mathbf{G}_{\mathrm{Im}-}\\ \mathbf{G}_{\mathrm{Im}-} & \mathbf{G}_{\mathrm{Im}+} & \mathbf{G}_{\mathrm{Re}-} & \mathbf{G}_{\mathrm{Re}+}\end{bmatrix}\cdot\mathbf{V},\end{aligned} \tag{6}$$

where $\mathbf{V} = \begin{bmatrix}\mathbf{V}_{\mathrm{Re}+} & \mathbf{V}_{\mathrm{Re}-} & \mathbf{V}_{\mathrm{Im}+} & \mathbf{V}_{\mathrm{Im}-}\end{bmatrix}^{\mathrm{T}}$.

Fig 3 (a) illustrates the conventional system architecture that employs the differential pairs approach to compute $\mathbf{X}_{\mathrm{Re}}$. The input vectors are assigned to the corresponding RRAM crossbars. Outputs from the crossbars in the upper and lower blocks corresponding to the first and second terms of Eq. 6, are aggregated and subsequently subtracted in the digital domain.

To maximize the computational speed and meet the requirement of real-time processing in the communication system, the crossbars can be replicated four times to represent each matrix entity. It is also feasible to multiplex the crossbar arrays, aiming for the optimal hardware overhead with the sacrificed latency.

Since the DFT computation demands high precision, it is essential to cluster multiple devices to represent one number with the bit slicing approach [8], particularly when the number of available resistance states is limited. The required crossbar size to map each matrix entity depends on the target length of DFT and the number of devices representing one number. Since the RRAM crossbar size is limited by the parasitic resistance [9], it is practical to utilize multiple crossbars to represent each matrix entity. For example, to compute a N-point DFT requiring the bitwidth of BW, with the crossbars sized by $M \times M$ and b-bit RRAM devices, the needed number of crossbars N_{xBar} can be derived as:

$$N_{\mathrm{xBar}} = \frac{16 \cdot N^2 \cdot BW}{b \cdot M^2} \tag{7}$$

Fig. 3 (b) illustrates the RRAM crossbar array with the peripheral circuitry that is utilized to compute VMM in DFT accelerators. The inputs are bit-sliced as binary values and encoded as voltages applied on the BLs, while the WL driver activates certain rows by biasing a voltage to the corresponding

access transistors. The weighted currents are received at the SLs and sensed by the analog-to-digital-converters (ADCs). Practically, each ADC is shared across multiple rows with the time-multiplexing way to amortize its hardware cost. The digitized outputs are fed into the shift-and-add blocks to recover the significance of bits due to the bit-sliced coefficients and inputs.

The energy efficiency of the RRAM systems is primarily limited by the power-hungry peripheral circuits that sense and convert the analog outputs to digital values [8]. To reduce the hardware cost of DFT accelerators, Zhao *et al.* [1] introduced a matrix concatenation strategy to maximize the computation in the analog domain. In the conventional architecture shown in Fig. 3, the outputs of the crossbars from the same blocks are accumulated in the digital domain. This process can be moved to the analog domain by merging the corresponding matrices. With this approach, the peripherals can be saved by half. However, it should be noted that the resolution of ADCs B_{adc} should be increased accordingly to guarantee full precision, which follows [8]:

$$B_{\mathrm{adc}} = \begin{cases} b + B_{\mathrm{in}} + \log_2 M, & \text{if } b > 1, B_{\mathrm{in}} > 1 \\ b + B_{\mathrm{in}} + \log_2 M - 1, & \text{otherwise} \end{cases} \tag{8}$$

where B_{in} is the input bitwidth.

The differential pairs approach realizes the mapping of the signed matrix with minimum efforts on data post-processing. Besides, it provides high immunity for the computations against the variability because both crossbars suffer the device-to-device variability, and the distortions can be suppressed after the subtraction [9], [10]. Nevertheless, this method also comes with some disadvantages. By comparing Eq. 4 and Eq. 6, it is evident that the number of required crossbars doubles. This increase, along with the associated peripherals, deteriorates the hardware overhead in terms of energy efficiency and area utilization. Apart from differential pairs, some other options can also be utilized to map the matrix with negative values. An alternative is to introduce an offset to shift all values in the matrix toward the positive direction before mapping and consequently eliminate all negative values [10], [11].

To further optimize the required hardware resource, instead of employing the differential pairs to represent the signed matrices, we propose a novel architecture to compute DFT that utilizes the offset approach and matrix concatenation to map the signed coefficients and optimize the utilization of peripheral circuitry, which is exhibited in Fig. 3 (c). By integrating the positive and negative matrix entities, the computation in Eq. 4 can be directly executed. The number of crossbars and the related peripheral circuits can be reduced by half. Additionally, the need for peripheral circuits is further halved due to the concatenation of $\mathbf{W}_{\mathrm{Re}}$ and $\mathbf{X}_{\mathrm{Im}}$ [1]. Compared to the conventional architecture, a post-processing stage is introduced, where the multiplications between the introduced offset and the summed inputs are subtracted from the obtained VMM outputs to recover the original results of DFT [10]. Notably, the post-processing operation is conducted in the digital domain and only requires negligible hardware resources compared to the saved crossbars and peripherals.

Mapping the signed matrices to the crossbar with the offset approach may eventually degrade the robustness of the system against the device variabilities and, as a consequence, deteriorate the computational accuracy [9], [10]. Therefore, to guarantee reliable computations with the proposed architecture, we propose an adaptive variability-aware mapping algorithm to compensate for the impact of mapping errors caused by the device non-idealities and quantization.

B. Adaptive Variability-aware Mapping Scheme

RRAM technology exhibits spatial variability across devices and temporal variability from cycle to cycle, which are induced by the nature of stochastic ion migration during the resistive switching [12]. The variability may lead to distorted computational results in RRAM systems because the actual programmed conductance on the crossbars differentiates from the expected values even with the applied write-verify algorithm.

To address this issue, several mapping algorithms, including SWIPE [13] and PRIVE [14], suggest adjusting devices that represent the same number. This approach emphasizes correcting devices representing less significant bits to compensate for errors from those for more significant bits. However, SWIPE [13] relies on the differential pairs, and PRIVE [14] only considers the binary RRAM devices. Remarkably, the reported schemes are only evaluated in the applications of DNN accelerators, which intrinsically allow some degree of randomness in the weights.

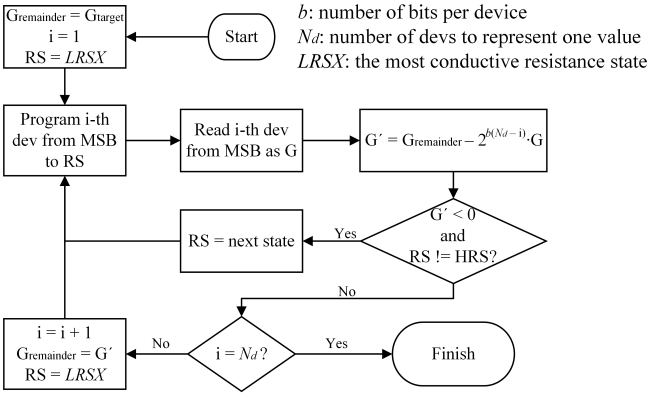


Fig. 4. The proposed adaptive variability-aware mapping scheme. The quantization errors and mapping errors caused by the devices with higher significance are mitigated by the devices with lower significance.

Fig. 4 exhibits the flowchart of the proposed variability-aware mapping scheme. We define the equivalent conductance as the conductance of the device after applying the shift operation based on bit significance, and the target conductance for a certain value W to be mapped to the devices as $\mathrm{G}_{\mathrm{target}}$. In other words, $\mathrm{G}_{\mathrm{target}}$ is the total conductance of the devices representing W after the shift-and-add operations. It is worth noting that $\mathrm{G}_{\mathrm{target}}$ may not always match the total equivalent conductance of the devices representing W with the ideal

conductance. Since the original matrix W holds a linear relationship with G_{target}, instead of quantizing W to derive G_{target}, we can directly convert W to G_{target} and start mapping. In this way, during the mapping process, any distortions in mapped conductance are incorporated into the quantization process and contribute to mitigating the quantization errors. This scheme differentiates from the conventional mapping approach, in which the original value W is firstly quantized with a specific bitwidth and mapped to the crossbar. In other words, the mapped conductance is vulnerable to two distortion sources.

The mapping is conducted progressively from the most significant bit (MSB) towards the least significant bit (LSB). Before the mapping starts, the remaining equivalent conductance $G_{remainder}$ to be mapped is assigned the value of G_{target} because no device is programmed yet. Firstly, the device corresponding to the MSB ($i = 1$) is programmed to the resistance state with the highest conductance. It should be noted that the programming is performed with the write-verify algorithm. Then, the conductance of the programmed device G is compared with $G_{remainder}$ with the consideration of bit significance, and the condition $G \cdot 2^{b(N_d-i)} \leq G_{remainder}$ should be fulfilled to terminate the programming of the current device. Otherwise, the device should be programmed to the next state with lower conductance, and the comparison is repeated. After programming the current device, the values of the remaining equivalent conductance $G_{remainder}$ is updated by subtracting the equivalent conductance of the last successfully programmed device, namely, $G_{remainder} = G_{remainder} - G \cdot 2^{b(N_d-i)}$. The remaining equivalent conductance $G_{remainder}$ is adaptively changed depending on the conductance G, which is possibly programmed with inaccuracies.

C. Simulation Framework

To assess the proposed RRAM-based DFT system architecture and adaptive mapping scheme, we develop a comprehensive framework. This framework quantitatively evaluates the system metrics, including mapping error and BER reflecting the reliability, as well as area utilization and energy consumption related to hardware overhead, as shown in Fig. 5.

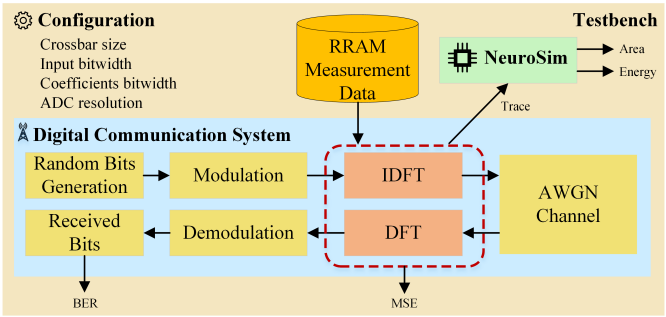


Fig. 5. The developed end-to-end simulation framework for RRAM-based DFT evaluation. A digital communication system model is included to evaluate the computational accuracy, and the NeuroSim is deployed to estimate the hardware overhead.

In the testbench, several parameters defining the system should be specified, including crossbar size, input and coefficient bitwidth, as well as ADC resolution. These parameters enable the broad design space exploration to investigate the trade-off between system reliability and hardware cost. The framework comprises two modules: the digital communication system implemented in Python to simulate the computational accuracy and NeuroSim [15] for hardware cost estimation.

In the digital communication system, we implement a signal flow for digital signal processing. The transmitted bits are randomly generated and modulated with the 16-QAM scheme. Then, the modulated frequency-domain symbols are processed by the RRAM-based IDFT accelerator and converted to the time domain. Afterward, the transmitted symbols pass through the channel and are captured by the receiver, which converts the time-domain signals back to the frequency domain by performing DFT. After demodulation, the demodulated bits are compared with the original bits to obtain the BER that indicates the system reliability degradation caused by the RRAM-based IDFT/DFT modules. To guarantee the accurate simulation of RRAM-based IDFT/DFT blocks considering the device-to-device variability, the measured RRAM data is utilized to derive the results of VMMs depending on the device indexes and input values.

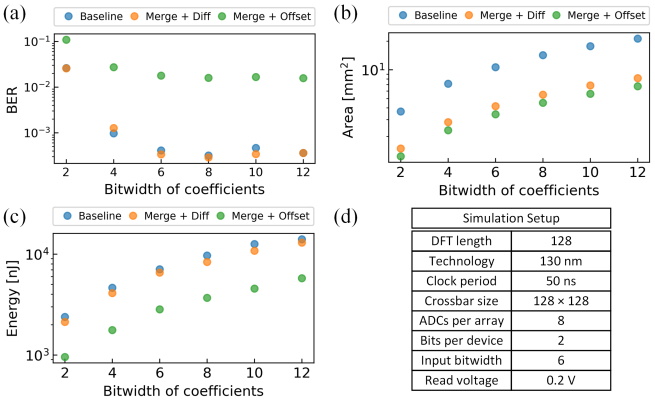
The second module adopts the NeuroSim framework [15] to estimate the hardware overhead of the RRAM crossbars and the associated peripheral circuits in the DFT accelerator. Given that the crossbars and peripheral circuits should be implemented on the same chip for high integration, we calibrate the hardware models in NeuroSim with the IHP 130 nm PDK in terms of area utilization and power consumption. Besides, the ADC model in the NeuroSim is calibrated with an ADC designed with the same technology and similar specifications [16]. The simulated traces of RRAM-based DFT/IDFT modules are conveyed to the NeuroSim to enable the trace-based simulation.

IV. Result and Analysis

A. Architecture Evaluation

We benchmark three architectures of the RRAM-based DFT/IDFT system. The architecture denoted as *Baseline* is the conventional architecture that allocates the peripherals to each crossbar, and the signed matrices are realized by differential pairs. The *Merge+Diff* represents the optimized architecture that is implemented based on the matrix concatenation approach [1] to save the hardware overhead. Our proposed architecture *Merge+Offset* implements the signed matrices with the offset approach.

Fig. 6 illustrates the simulation setup and results with the three architectures varying the bitwidth of the DFT matrix. It should be pointed out that the ADC resolution for each architecture is adaptively adjusted according to Eq. 8 because of the enlarged crossbar size after concatenating. The simulated BER indicates that the reliability of the communication system is improved with the wider bitwidth of coefficients, but it is achieved with the price of area footprint and energy consumption. Additionally, it can be observed that when the bitwidth is larger than eight bits, the BER improvement saturates.



Simulation Setup	
DFT length	128
Technology	130 nm
Clock period	50 ns
Crossbar size	128 × 128
ADCs per array	8
Bits per device	2
Input bitwidth	6
Read voltage	0.2 V

Fig. 6. Simulation results of the conventional architecture *Baseline*, architecture with matrix concatenation and differential pairs *Merge+Diff*, and the proposed offset architecture *Merge+Offset*. (a) BER in the communication system with 16-QAM. (b) Area footprint. (c) Energy consumption for each transform. (d) Simulation setup.

The architectures implemented with differential pairs share similar BERs that are much lower than the *Merge+Offset*. Besides, the *Merge+Diff* design utilizes less area and energy compared to *Baseline*. Notably, the proposed design *Merge+Offset* requires less hardware cost, especially for energy consumption.

B. Efficacy of Variability-aware Mapping Scheme

Merge+Diff and *Merge+Offset* designs are evaluated with the proposed adaptive mapping scheme to validate its effectiveness. The coefficient bitwidth is configured as eight bits.

Design	MSE			BER		
	Conv.	Proposed	Δ	Conv.	Proposed	Δ
Merge+Diff	2.3e-3	4.8e-4	79%	2.9e-3	0	100%
Merge+Offset	1.3e-2	2.2e-3	83%	1.6e-2	7.3e-5	>99%

Table I. The comparison of MSE and BER between the conventional and proposed mapping scheme.

Design	Area	Energy	BER
Merge+Diff	5.5 mm^2	8520 nJ	0
Merge+Offset	4.7 mm^2	3644 nJ	7.3e-5
Δ	18%	57%	

Table II. Area and energy utilization comparison.

As shown in Tab. I, with the proposed mapping scheme, both designs gain an extraordinary enhancement of mapping errors quantified with mean squared error (MSE) and the consequent BER compared to the conventional approach. Furthermore, as indicated by Tab. II, compared to *Merge+Diff*, the proposed *Merge+Offset* approach improves the area footprint and consumption by 18% and 57%, respectively, with an insignificant BER of 7.3e-5. For *Merge+Diff* inherently with high robustness, the BER decreases from 2.9e-3 to zero.

V. Conclusion

As an emerging technology, RRAM provides new opportunities to evolve existing computing systems by combining remarkable device properties and the in-memory computing paradigm. In this work, we propose a novel RRAM-based architecture to compute DFT/IDFT. To improve the computational accuracy, we present an adaptive variability-aware mapping scheme to efficiently compensate for the quantization and mapping errors in RRAM crossbars. With our developed end-to-end simulation framework, the simulation results emphasize that in comparison with the state-of-the-art architecture, the proposed design optimizes the area and energy consumption. Notably, by employing the proposed mapping scheme, the signal processing BERs of both designs are significantly improved. With the described strategies, both the computational accuracy and energy efficiency of RRAM-based systems are elevated.

Acknowledgement

This work was supported by the Federal Ministry of Education and Research (BMBF, Germany) as part of the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K.

References

[1] H. Zhao *et al.*, "Energy-efficient High-fidelity Image Reconstruction with Memristor Arrays for Medical Diagnosis," *Nat Commun*, vol. 14, p. 2276, 2023.
[2] T. Abtahi *et al.*, "Accelerating Convolutional Neural Network with FFT on Embedded Hardware," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 9, pp. 1737–1749, 2018.
[3] M. Lanza *et al.*, "Memristive Technologies for Data Storage, Computation, Encryption, and Radio-frequency Communication," *Science*, vol. 376, no. 6597, p. eabj9979, 2022.
[4] W. Wan *et al.*, "A Compute-in-memory Chip based on Resistive Random-access Memory," *Nature*, vol. 608, pp. 504–512, 2022.
[5] G. Yuan *et al.*, "Memristor Crossbar-based Ultra-efficient Next-generation Baseband Processors," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1121–1124, 2017.
[6] Q. Hong *et al.*, "Circuit Design and Application of Discrete Cosine Transform based on Memristor," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 2, pp. 502–513, 2023.
[7] E. Perez *et al.*, "Variability and Energy Consumption Tradeoffs in Multilevel Programming of RRAM Arrays," *IEEE Transactions on Electron Devices*, vol. 68, no. 6, pp. 2693–2698, 2021.
[8] A. Shafiee *et al.*, "ISAAC: A Convolutional Neural Network Accelerator with In-situ Analog Arithmetic in Crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pp. 14–26, 2016.
[9] T. P. Xiao *et al.*, "On the Accuracy of Analog Neural Network Inference Accelerators," *IEEE Circuits and Systems Magazine*, vol. 22, no. 4, pp. 26–48, 2022.
[10] J. Wen *et al.*, "Evaluating Read Disturb Effect on RRAM based AI Accelerator with Multilevel States and Input Voltages," in *2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pp. 1–6, 2022.
[11] M. Hu *et al.*, "Dot-product Engine for Neuromorphic Computing: Programming 1T1M Crossbar to Accelerate Matrix-vector Multiplication," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2016.
[12] M. Zhao *et al.*, "Reliability of Analog Resistive Switching Memory for Neuromorphic Computing," *Applied Physics Reviews*, vol. 7, no. 1, p. 011301, 2020.
[13] S. K. Gonugondla *et al.*, "SWIPE: Enhancing Robustness of ReRAM Crossbars for In-memory computing," in *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9, 2020.
[14] W. He *et al.*, "PRIVE: Efficient RRAM Programming with Chip Verification for RRAM-based In-memory Computing Acceleration," in *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, 2023.
[15] X. Peng *et al.*, "DNN+NeuroSim: An End-to-End Benchmarking Framework for Compute-in-memory Accelerators with Versatile Device Technologies," in *2019 IEEE International Electron Devices Meeting (IEDM)*, pp. 32.5.1–32.5.4, 2019.
[16] M. Uhlmann *et al.*, "Energy Efficient ADC for Low Fan-out MIMO Sub- THz Imaging System in SiGe: BiCMOS Technology," in *2022 52nd European Microwave Conference (EuMC)*, pp. 44–47, 2022.