

Realizing In-Memory Baseband Processing for Ultra-Fast and Energy-Efficient 6G

Qunsong Zeng, Jiawei Liu, Mingrui Jiang, Jun Lan, Yi Gong, Zhongrui Wang, Yida Li, Can Li, Jim Ignowski, and Kaibin Huang

Abstract—To support emerging applications ranging from holographic communications to extended reality, next-generation mobile wireless communication systems require ultra-fast and energy-efficient baseband processors. Traditional complementary metal-oxide-semiconductor (CMOS)-based baseband processors face two challenges in transistor scaling and the von Neumann bottleneck. To address these challenges, in-memory computing-based baseband processors using resistive random-access memory (RRAM) present an attractive solution. In this paper, we propose and demonstrate RRAM-implemented in-memory baseband processing for the widely adopted multiple-input-multiple-output orthogonal frequency division multiplexing (MIMO-OFDM) air interface. Its key feature is to execute the key operations, including discrete Fourier transform (DFT) and MIMO detection using linear minimum mean square error (L-MMSE) and zero forcing (ZF), in one-step. In addition, RRAM-based channel estimation module is proposed and discussed. By prototyping and simulations, we demonstrate the feasibility of RRAM-based full-fledged communication system in hardware, and reveal it can outperform state-of-the-art baseband processors with a gain of $91.2\times$ in latency and $671\times$ in energy efficiency by large-scale simulations. Our results pave a potential pathway for RRAM-based in-memory computing to be implemented in the era of the sixth generation (6G) mobile communications.

Index Terms—In memory computing, baseband processing, resistive switching memory, 6G communications, MIMO-OFDM.

I. INTRODUCTION

While the fifth generation (5G) mobile networks are being deployed, the sixth generation (6G) is under development all over the world to provide a new infrastructure for propelling the digital economy forward and realizing Society 5.0 [1]. The performance of 6G will be unprecedented as reflected in a set of target key performance indicators (KPIs), dictating a peak data rate to go beyond 100Gb/s, having a minimum latency 0.1ms, and achieving an energy efficiency of 10^{-12} J/bit [2]–[6]. This coined the term *ultra-fast-and-energy-efficient* (UFEE) communication and will enable a wide range of emerging applications, for example, industrial automation [7], [8], tactile internet [9]–[11], holographic communications [12], [13], and digital twin [14], [15]. Hence, this provides a strong motivation for 6G researchers to explore the largely unoccupied Terahertz (THz) spectrum [2]–[6]. However, the required scaling up of baseband data rates to the

hundreds of Gbps level will dramatically increase the power consumption and complexity of baseband processing, making it challenging to realize the 6G vision [16]–[18]. This is further exacerbated by the increasingly sophisticated communication techniques required, including large-scale *multiple-input multiple-output* (MIMO), high-dimensional *orthogonal frequency division multiplexing* (OFDM), and interference management. From 2G to 5G era, baseband processing demands have been satisfied largely by shrinking transistor size as governed by Moore’s law. Accordingly, the semiconductor industry has evolved from planar bulk *Metal-Oxide-Semiconductor Field-Effect Transistors* (MOSFETs) to the recent 3D FinFETs and *Gate All Around* (GAA) architectures to improve transistor performance and density in an integrated circuit (IC) chip [19]. However, this approach is facing increasing challenges as transistor size approaches the atomic limit [20]. In view of the Moore’s Law coming to an end, we propose the new paradigm called in-memory baseband processing for the post-Moore era, which adopts the emerging in-memory computing architecture instead of relying on transistor densification, to pave the way towards realizing the 6G UFEE connectivity.

Baseband processing and computing at large face two bottlenecks: the von Neumann bottleneck and the power wall, incurring large energy and footprint overheads. The former is due to data shuttling between the physically separated processing and storage units, resulting in significant latency and high energy consumption (e.g., 100-time more than digital logical circuits). In the latter, the increasing power density of transistors as the transistor size shrinks has created a “power wall” that limits practical processor frequency to ~ 4 GHz since 2006 [21], falling far short of the requirements for THz communications. In the past decade, researchers have started to improve computing latency and energy consumption by employing an architecture that co-locates data processing and storage, so called in-memory computing. Rather than making incremental improvements to conventional systems such as parallelism or memory bandwidth, in-memory computing takes a different approach by performing calculations where the data is located, thus fundamentally changing the von Neumann architecture [22]. This method is similar to the way the human brain processes information in the networks of neurons and synapses, where there is no separation between computation and memory [23]. In contrast to traditional computing schemes, in-memory computing eliminates latency and energy usage issues associated with the memory wall. However, this new architecture requires computational memory devices that can both store data and perform cal-

Q. Zeng, J. Liu, M. Jiang, Z. Wang, C. Li, and K. Huang are with The University of Hong Kong, Hong Kong SAR. J. Liu, J. Lan, Y. Gong, and Y. Li are with Southern University of Science and Technology, Shenzhen, China. J. Ignowski is with Hewlett Packard Enterprise, United States. Q. Zeng and J. Liu contributed equally to this work. Contact: K. Huang (huangk@eee.hku.hk).

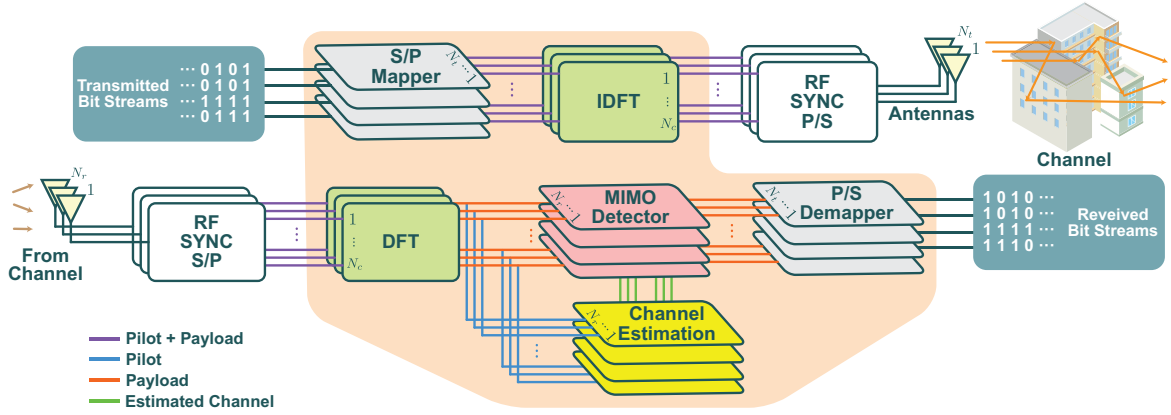


Figure 1. The architecture of RRAM-based transmitter: It consists of baseband processing modules [i.e., mapper and IDFT], RF modem, and an array of transmit antennas. Each layer represents a piece of RRAM-based circuit. The architecture of RRAM-based receiver: It is comprised of an array of receive antennas, RF front-end, and baseband processing modules [i.e., DFT, channel estimation, MIMO detection, and demapper].

culations simultaneously, usually by leveraging physical laws like Ohm's and Kirchhoff's laws in electrical circuits [24]. Emerging non-volatile memories such as *resistive random-access memory* (RRAM) is touted as one of the most potential candidates for such computational memory devices [25]. It has been reported that parallel execution of a larger number (e.g., millions) of multiply-and-accumulate (MAC) operations for matrix vector multiplications (MVM) can be accomplished with extremely high energy-efficiency and low latency [26]. This makes in-memory computing a UFEE solution for MVM intensive applications such as deep neural networks [27]–[42] and linear algebra computation [43]–[46]. Such advantages can naturally contribute to the trend of seamless integration of communication and artificial intelligence (AI) for the next-generation Internet-of-Things (IoTs). A new paradigm for communications called in-memory baseband processing, which adopts the emerging in-memory computing architecture and novel signal processing approach, are potential key factors to alleviate the challenges faced by researchers in realizing UFEE connectivity in the era of 6G.

6G will feature scaling up of different physical-layer technologies, for example, massive MIMO using large-scale antenna arrays [47] and OFDM comprising thousands of sub-carriers [4]. The resultant baseband processing will involve frequent large-scale matrix operations. This motivates us to propose the new paradigm of in-memory baseband processing, which relocates the conventional digital operations to the analogue domain to achieve UFEE processing. In this paper, we present the design of an in-memory baseband processor for MIMO-OFDM which is a dominant air-interface technology for 5G-and-beyond [2]–[5], [48]. The key novelty includes modules, namely OFDM demodulation, MIMO detection, and channel estimation, which are designed and implemented using in-memory computing approach based on Ta/TaO_x/Pt RRAM chip. The OFDM module implements the *discrete Fourier transform* (DFT) using two RRAM crossbar arrays. Using such arrays to store DFT matrix enables one-step DFT operation, cutting down the power/latency overheads in conventional CMOS-based processor significantly. Furthermore, the required channel matrix inversion for MIMO detection

is realized using a novel RRAM circuit featuring stability and easy mode switching, enabling the one-step operation. The performance of our design is evaluated using proof-of-concept prototypes for separate modules and a complete system by physical experiments, respectively, and simulations for a large-scale communication system. We show that the throughput and energy-efficiency can be boosted up to $91.2\times$ and $671\times$ respectively as compared to state-of-the-art CMOS-based baseband processors.

II. OVERVIEW OF RRAM-BASED BASEBAND PROCESSOR

In this paper, the baseband processor targets the MIMO-OFDM air interface, where a pair of multi-antenna transmitter and receiver communicate over a broadband channel. In broadband communications, frequency selective fading occurs when the channel having a coherence bandwidth is smaller than that of the signal causes its distortion. As a popular technology for coping with such fading as well as inter-symbol interference, OFDM is adopted to divide the whole bandwidth into N_c orthogonal sub-channels. As a result, each sub-channel, say the k -th sub-channel, is a narrowband channel with N_t transmit and N_r receive antennas, modelled by a MIMO-channel matrix $\mathbf{H}^{(k)} \in \mathbb{C}^{N_r \times N_t}$ that is fixed within an OFDM symbol. The input-output relation of a MIMO system over the k -th sub-channel is given as

$$\mathbf{y}^{(k)} = \mathbf{H}^{(k)} \mathbf{x}^{(k)} + \mathbf{z}^{(k)}, \quad (1)$$

where $\mathbf{x}^{(k)} \in \mathbb{C}^{N_t \times 1}$ consists of symbols at the k -th sub-carrier, $\mathbf{y}^{(k)} \in \mathbb{C}^{N_r \times 1}$ comprises the received symbols at the k -th sub-carrier, and $\mathbf{z}^{(k)}$ represents the *additive white Gaussian noise* (AWGN) in propagation.

The architectures of the RRAM-based transceiver are illustrated in Fig. 1. Before baseband processing, the receiver still needs sampling at the RF front-end. Compared with a fast analogue-to-digital converter (ADC) design for traditional digital baseband processing, the proposed baseband processor features the direct processing of analogue-valued input signals so that the ADC can be replaced with a simpler sample-and-hold circuit. The baseband (information) processing starts at

the mapper module in the transmitter that transforms bits into symbols and ends at the demapper module in the receiver that transforms the symbols back to bits. The digital modulation is chosen as 16 quadrature amplitude modulation (16-QAM) unless specified otherwise, which maps a 4-bit string to one of the 16 points on the constellation diagram. The bit stream is split into in-phase (denoted by I) and quadrature (denoted by Q) streams, associated with 0-degree and 90-degree phase shifts of the carrier wave, respectively. I and Q components are Gray encoded, i.e., neighbour points only differ in a single bit, to produce symbol points in the constellation. The system performance is evaluated by two metrics: i) The *modulation error ratio* (MER) measures the dispersion of the constellation of the received symbols. To be specific, given total M transmitted symbols, the definition of MER is

$$\text{MER} = 10 \log_{10} \left(\frac{\sum_{m=1}^M (I_m^2 + Q_m^2)}{\sum_{m=1}^M [(I'_m - I_m)^2 + (Q'_m - Q_m)^2]} \right) \text{ dB}, \quad (2)$$

where I_m and Q_m denote the in-phase and quadrature components of the m -th transmitted symbol while I'_m and Q'_m denote the in-phase and quadrature components of the received symbol. ii) The *bit error ratio* (BER) is the number of bit errors divided by the total number of transmitted bits. To be specific, during the studied time interval, the BER is given by

$$\text{BER} = \frac{\# \text{ error bits}}{\# \text{ total transmitted bits}} \times 100\%. \quad (3)$$

In this work, we focus on the baseband processing between the mapper and demapper. The module in the transmitter performs *inverse DFT* (IDFT). For the receiver, the three modules are DFT module, channel estimator, and MIMO detector. To reconcile signals and channels in the complex domain and the fact that RRAM devices store and compute real numbers, we propose to apply the mapping $\mathcal{R} : \mathbb{C}^{K \times L} \rightarrow \mathbb{C}^{2K \times 2L}$ which transforms a complex matrix $\mathbf{A} \in \mathbb{C}^{K \times L}$ into a real matrix $\mathcal{R}(\mathbf{A}) = \begin{bmatrix} \Re(\mathbf{A}) & -\Im(\mathbf{A}) \\ \Im(\mathbf{A}) & \Re(\mathbf{A}) \end{bmatrix} \in \mathbb{R}^{2K \times 2L}$. The complex vector is translated as the input voltages (or currents) for the RRAM array, with the mapping $\mathcal{T} : \mathbb{C}^{K \times 1} \rightarrow \mathbb{R}^{2K \times 1}$ transforming a complex vector $\mathbf{x} \in \mathbb{C}^{K \times 1}$ into a real vector $\mathcal{T}(\mathbf{x}) = \begin{bmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{2K \times 1}$. Such transformations enables the equivalent computation involving complex matrices and vectors.

We use the Ta/TaO_x/Pt-based RRAM arrays as the hardware accelerators for its compatibility with traditional CMOS process and reliable electrical characteristics. Details of the RRAM array fabrication and integration are described in Appendix A. The wire-bonded integrated RRAM chip that we used to implement the baseband processor modules is shown in Fig. 2(a), which contains three 64×64 RRAM crossbar arrays and one of them is shown in Fig. 2(b). The $50\text{nm} \times 50\text{nm}$ Ta/TaO_x/Pt RRAMs are integrated with back-end-of-the-line (BEOL) processing on top of the control peripheral circuits (see Fig. 2(c)). The peripheral control circuits are implemented with a commercial 180nm technology integrated chip, among

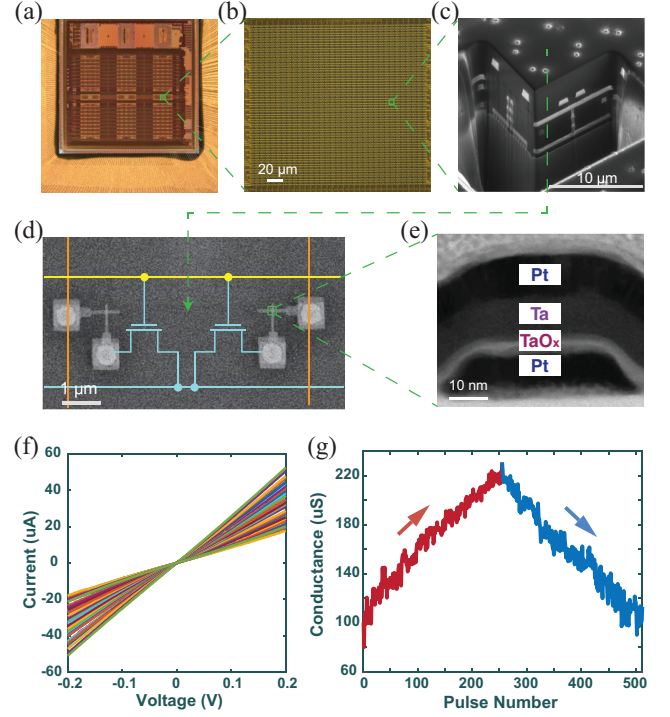


Figure 2. Integrated RRAM chip and measurements. (a) A photo of a wire-bonded integrated RRAM chip, which contains three 64×64 1T1R crossbar arrays, row MUXes, column MUXes, transimpedance amplifiers (TIA), sample-and-hold, and ADC. (b) Optical image of a 64×64 RRAM crossbar array. (c) The cross-section view of the integrated chip with CMOS circuits at the bottom, inter-connection in the middle, and metal through-hole on the surface used for RRAM and back-end process integration. (d) Top view of four cross-point RRAM devices. (e) The TEM image of the RRAM device. (f) Ohmic behaviour of RRAM devices. The linear I-V relationship is illustrated at different conductance states under different read voltages (-0.2~0.2V). (g) The conductance modulation characteristic of the RRAM device. A train of voltage pulses (pulse width 10ns) are applied for the RRAM conductance modulation measurements. The magnitude of voltages starts at 0.60V and grows to 0.70V smoothly for potentiation, while it starts at -0.50V and drops to -0.65V gradually for depression. The cycle-to-cycle variations are 4.41% during potentiation and 5.44% during depression, respectively. The conductance ranges from $79.93\mu\text{S}$ to $230.99\mu\text{S}$ in the behavioral measurement.

which the access transistors are highlighted in Fig. 2(d). Such one-transistor-one-resistor (1T1R) array architecture avoids the sneak current issue during RRAMs' conductance programming and allows each cell in the array to be accessed independently [49]. The cross-sectional transmission electron microscopy (TEM) image of the RRAM device is shown in Fig. 2(e). As a non-volatile analogue device, our RRAM device exhibits linear Ohmic behaviour (see Fig. 2(f)) to ensure accurate in-memory computing. For matrix mapping, the conductance programming of the fabricated RRAM device can be achieved by applying a train of positive pulses (0.60~0.70V/10ns) for potentiation, and continuous negative pulses (-0.50~-0.65V/10ns) for depression (see Fig. 2(g)).

III. ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING MODULE

The RRAM-based DFT module is illustrated in Fig. 3(a), where data are modulated onto non-interfering sub-carriers in the frequency domain. The transformation between the time

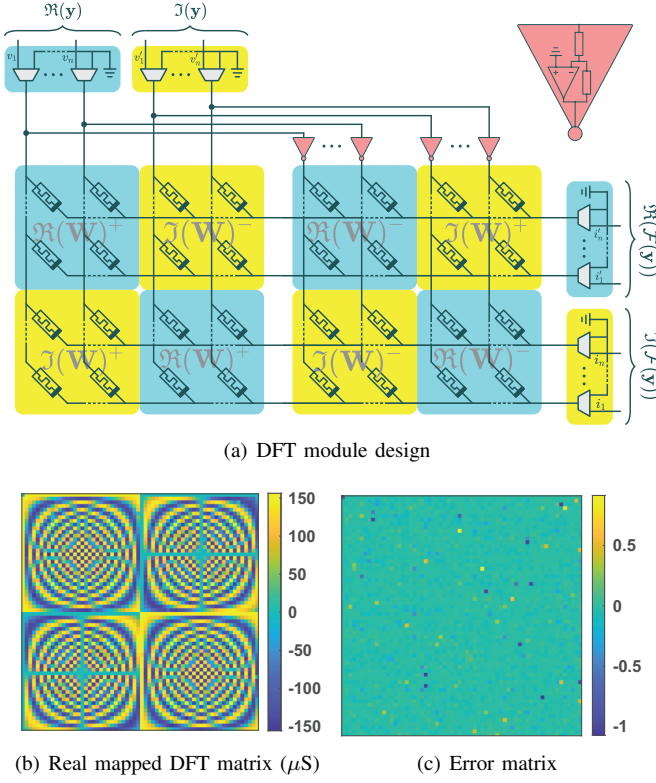


Figure 3. Orthogonal frequency-division multiplexing modules. (a) The architecture of RRAM-based DFT module. The DFT matrix \mathbf{W} is stored in the RRAM array and the input signal \mathbf{y} is translated as the voltages to be applied to the array. The elements (and signals) in real and imaginary domains are highlighted by different colours. The module performs the DFT over signal \mathbf{y} and the read drivers get the result $\mathcal{F}(\mathbf{y}) = \mathbf{W}\mathbf{y}$. The design of inverting amplifier is presented at the upper right corner. (b, c) In the experiment, the real mapping of the DFT matrix is scaled and programmed into two 64×64 RRAM arrays in our integrated RRAM chip: (b) conductance matrix and (c) corresponding error matrix, each element of which refers to the ratio (experimental conductance – target conductance)/target conductance [note: the value is not in percentage form].

and frequency domains involves IDFT/DFT operations. For the received block of symbols \mathbf{y} , the DFT of which can be represented as an N_c -length vector: $\mathcal{F}(\mathbf{y}) = \mathbf{W}\mathbf{y}$, where $\mathcal{F}(\cdot)$ denotes the DFT operation and \mathbf{W} is the DFT matrix. In the circuit design, the real mapping of DFT matrix, $\mathcal{R}(\mathbf{W})$, is scaled into the RRAM devices' conductance range and stored as the difference between two arrays. The received signal \mathbf{y} is translated to the input voltages $\mathcal{T}(\mathbf{y})$ for the array. The module computes the DFT of \mathbf{y} , and the current outputs are the scaled real vector mapping $\mathcal{T}(\mathcal{F}(\mathbf{y}))$. The detailed discussion on the hardware implementation of this module is provided in Appendix B. Compared with conventional approaches based on *fast Fourier transform* (FFT) algorithms [50], the RRAM-based design features the dramatic reduction of computational complexity of from $O(N_c \log N_c)$ for FFT to just a one-step (i.e., $O(1)$) operation. This makes it possible to overcome the bottleneck of high complexity of DFT in baseband processing for the next-generation large-scale OFDM communications.

In this section, a single-antenna OFDM system with 32 sub-carriers is demonstrated. The conductance mapping of the DFT matrix to RRAM array is scaled to fit the RRAM devices' conductance range, which are programmed into two arrays.

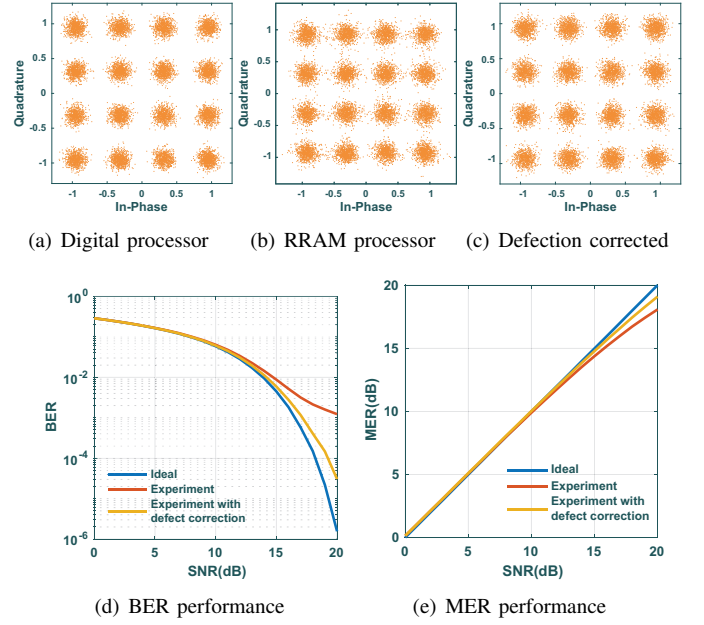


Figure 4. Experimental performance of the DFT module in the demonstrated OFDM system. At the receiver, the constellation diagram is recovered using (a) digital processor, (b) the RRAM processor experimental results, (c) RRAM processor results compensated by defective correction. (d, e) Communication performance of digital processor, RRAM-implemented DFT and deflection corrected RRAM-DFT under different channel conditions.

The subtraction of the conductance matrices of these two arrays, in the form of differential pairs, and the corresponding error matrix are presented in Fig. 3(b) and Fig. 3(c). The complete signal processing path in the prototypical RRAM-based OFDM system is described as follows. For the transmitter, a message in bits is firstly modulated into 16-QAM symbols, and then transformed from the frequency domain into the time domain by IDFT. After adding a cyclic prefix, the OFDM symbols are transmitted over the channel towards the receiver. At the receiver, after removing the cyclic prefix, the RRAM-based DFT is performed to transform the received signal back to the frequency domain, where the symbols are then demodulated into bits to recover the message. The performance of the receiver with RRAM-based DFT module is experimentally characterized over the wireless channel of receive signal-noise-ratio (SNR) being 20dB. As a benchmark, the constellation diagram recovered by the digital processor using a double-precision floating-point DFT matrix is shown in Fig. 4(a). In this case, the distortion of demodulated symbols is measured by MER 20dB at which no bit errors occur. For our RRAM-implemented DFT prototype, the measured constellation diagram is shown in Fig. 4(b) with MER dropping to 18dB while the BER growing to 0.00146. The communication performance is affected by both the channel noise and RRAM devices' imperfections. Compared with the results from a digital processor, the performance loss of our experimental RRAM-implemented DFT module comes from the imperfections of RRAM devices in the array, including defections and programming errors. To reveal the effect of the defective devices, we compensate the defections by post-processing. To be specific, we define the defection matrix $\mathbf{W}_{\text{defection}}$

as the compensatory conductance matrix of stuck-on and stuck-off RRAM devices. We then perform the multiplication operation $\mathbf{W}_{\text{defection}}\mathbf{y}$ in computer and add the result into the experimental outcome to obtain the defection-corrected result: $\mathbf{x} = \mathbf{W}_{\text{RRAM}}\mathbf{y} + \mathbf{W}_{\text{defection}}\mathbf{y}$. Leveraging this method, we rectify the experimental constellation diagram from RRAM-implemented DFT module as shown in Fig. 4(c), whose BER is ameliorated by an order of magnitude. Moreover, we explore the performance of our design with different transmission powers (i.e., different SNRs) as shown in Fig. 4(d) and Fig. 4(e). We observe that the performance differences between digital processor and our experimental RRAM-implemented DFT module are insignificant for a noisy channel. However, the differences can be noticeable for cleaner channels where the imperfections of RRAM devices in the array deteriorate the communication performance. The defective devices play a destructive role in the baseband processing and tackling this issue brings benefit to the enhancement of performance.

IV. MULTIPLE-INPUT AND MULTIPLE-OUTPUT DETECTION MODULE

The RRAM-based MIMO detection module is illustrated in Fig. 5(a) and Fig. 5(b), which spatially multiplexes multiple parallel data-streams. This scales up the system throughput since different symbols are simultaneously transmitted over different antennas. Exploiting the unique channel between each pair of transmit and receive antennas allows each transmitted symbol to be recovered through the module of MIMO detection. In practice, two linear detectors are widely used, namely *linear minimum mean square error* (L-MMSE) and *zero forcing* (ZF) detectors. They reverse the signal distortion by propagation through a MIMO channel by channel equalization. To be specific, given the channel matrix \mathbf{H} , the L-MMSE detector minimizes the mean squared error in the estimate of \mathbf{x} among all linear detectors. The recovered signal vector is given by $\hat{\mathbf{x}} = (\mathbf{H}^H\mathbf{H} + \frac{1}{\text{SNR}}\mathbf{I})^{-1}\mathbf{H}^H\mathbf{y}$, where \mathbf{y} is the received signal vector at the receiver. In hardware implementation, the equivalent real-value channel matrix $\mathcal{R}(\mathbf{H})$ is scaled and written into the RRAM arrays in the way as illustrated in Fig. 5(a), and the received signal \mathbf{y} is scaled and translated to the input voltages $\mathcal{T}(\mathbf{y})$. The output voltages are the real vector mapping $\mathcal{T}(\hat{\mathbf{x}})$, and the detailed analysis of this circuit is provided in Appendix C. To cope with heterogeneous propagation environments with different SNRs, the feedback conductance of operational amplifiers can be represented using a RRAM device as shown in Fig. 5(b). Our design also applies to ZF detection (see Appendix C) which solves the least square problem and gives the recovered signal vector as $\hat{\mathbf{x}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y}$. As shown in Fig. 5(b), the transistor dictates whether L-MMSE or ZF is applied. If the channel matrix is square, i.e., $N_t = N_r = N$, the computational complexity of conventional matrix inversion is $O(N^3)$. The complexity increases rapidly as the number of transmit/receive antennas grows. On the contrary, the proposed MIMO detection performs the computation in just a single step (i.e., $O(1)$), presenting a promising solution for efficient detection in the 6G massive MIMO communication.

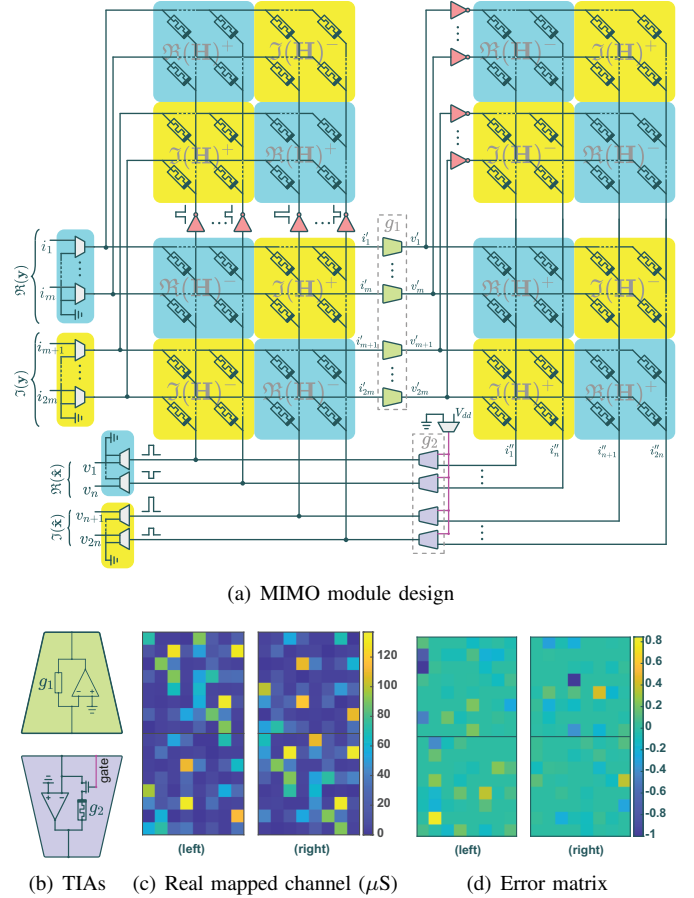


Figure 5. Multiple-input and multiple-output modules. (a) The architecture of RRAM-based MIMO detection module. The channel matrix \mathbf{H} is stored in the four RRAM arrays as marked in the figure and the input signal \mathbf{y} is scaled and translated as the input currents. The elements (and signals) in real and imaginary domains are highlighted by different colours. (b) The transistor controls whether L-MMSE or ZF modules is adopted. When the gate is grounded, the circuit performs ZF detection. Otherwise, L-MMSE is selected. In addition, to adapt to environments with different SNRs, the feedback conductance of the operational amplifiers is tuneable. (c, d) In the experiment, the real mapped channel matrix is scaled and programmed into RRAM arrays in our integrated RRAM chip: (c) conductance matrix and (d) corresponding error matrix, each element of which refers to the ratio (experimental conductance – target conductance)/target conductance.

We experimentally demonstrated the RRAM-based narrow-band MIMO system with 4 transmit antennas and 4 receive antennas. The real mapped channel matrix $\mathcal{R}(\mathbf{H})$ is scaled and programmed into the RRAM arrays (see Fig. 5(c)). The programming error is presented in Fig. 5(d). The experimental results of the constellation diagrams from L-MMSE detection for a noisy channel of SNR being 20dB. To begin with, the constellation diagram recovered by the digital processor is shown in Fig. 6(a) as a benchmark. For our RRAM-implemented MIMO detection module, the measured MER drops 4dB compared to the digital counterpart, inducing more bit errors (see Fig. 6(b)). The performance loss comes from the programming noise of RRAM devices whose effects on the circuit are twofold: i) the imprecision of the channel matrix representation; ii) the imbalance of the left and right channel matrices. To reduce the effect of the programming noise in RRAM devices, we use two RRAM devices to represent one

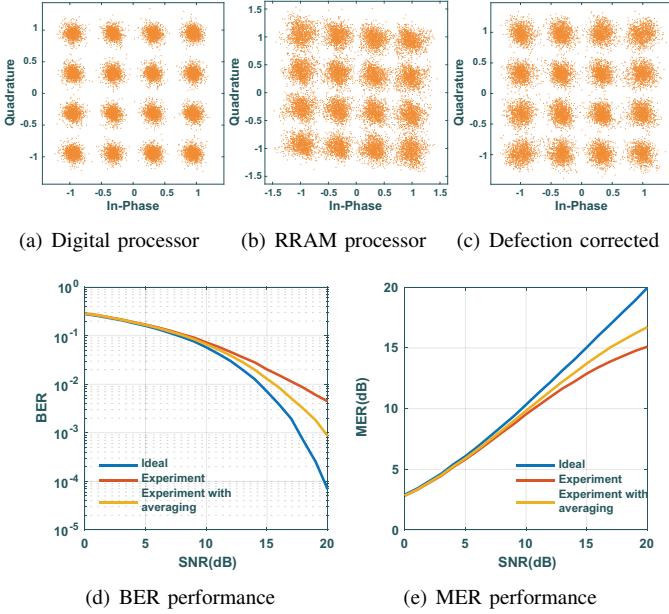


Figure 6. Experimental performance of the MIMO detection module in the demonstrated MIMO system. At the receiver, the constellation diagram is recovered using (a) digital processor, (b) RRAM-implemented MIMO detection module, (c) RRAM MIMO detection module averaged from two implementations. (d, e) Communication performance of digital processor, RRAM-implemented MIMO detection, and averaged RRAM-pair implementation under different channel conditions.

real value which suppresses the variance of random noise. By this means, we find the dispersion of constellation points (see Fig. 6(c)) becomes less severe and the BER reduces by an order of magnitude. Furthermore, as shown in Fig. 6(d) and Fig. 6(e), we study the performance differences between digital processor and our RRAM-implemented MIMO detection module under different channel conditions. We observe subtle differences between them in the low SNR regime while the divergence becomes noticeable in the high SNR regime. The performance loss of our RRAM-implemented MIMO detection module can be relieved by representing channel matrix using more RRAM devices to reduce the programming noise.

V. CHANNEL ESTIMATION MODULE

To acquire the *channel state information* (CSI) needed for MIMO detection, the channel matrix is estimated at the receiver using pilot signals that are sent by the transmitter and known a priori to the receiver. Many data symbols can be transmitted between two pilot signals separated by channel coherent time, amortizing the overhead of channel training. A larger ratio between data and pilot symbols improves the system throughput at the cost of adaptive to time-varying channels. The transmitted training matrix $\mathbf{P} \in \mathbb{C}^{N_t \times N_t}$ is known by the receiver, while the actual received matrix is $\mathbf{S} \in \mathbb{C}^{N_r \times N_t}$. By choosing the pilot signal as a unitary matrix [51], i.e., $\mathbf{P}\mathbf{P}^H = \mathbf{I}$, the channel matrix estimated by *maximum likelihood* (ML) or *least square* (LS) is given as $\hat{\mathbf{H}} = \mathbf{S}\mathbf{P}^H$. In the RRAM-based channel estimation module, the real mapped training matrix $\mathcal{R}(\mathbf{P})$ is stored in the RRAM array. Each row vector of the real mapped received

matrix $\mathcal{R}(\mathbf{S})$ is translated to the supplied input voltages. The computation can be completed by $2N_r$ read pulses while the complexity is $O(N_r N_t^2)$ for traditional processors.

When ready, the row vectors of the estimated channel matrix are sequentially written into the RRAM array implementing the MIMO detector. We evaluate the performance of different writing process in terms of system latency. To this end, a mathematical model is developed to facilitate latency analysis for programming a 1T1R array as elaborated in Appendix E. Consider the writing process using a train of pulses to program an $N \times N$ array in the row-by-row manner. It can be proved that the expected writing latencies of write-without-verification and write-with-verification schemes scale with the array size in the way no faster than $O(N\sqrt{\ln N})$ and $O(N \ln N)$, respectively. This contributes to the most latency in our design. For comparison, the computational complexity is $O(N^3)$ for traditional digital processors.

VI. PERFORMANCE EVALUATION OF THE COMPLETE SYSTEM

Recall that we consider the MIMO-OFDM air interface where a transmitter/receiver integrates the RRAM-based OFDM and MIMO modules. The modules are separately validated in previous sections. Here, we report a system-level demonstration of a MIMO-OFDM system with 32 sub-carriers and 2×2 antennas for proof-of-concept. The RF chains are physically implemented using software defined radio (SDR) platform, which provides the realistic MIMO communication links over-the-air. The digital logic on a workstation regulates data generation, executes frame synchronization algorithms, orchestrates the operation of the other two platforms, and controls the system data flow. The system schematic is presented in Fig. 7. The workflow is described as follows. The workstation randomly generates bit stream and maps bits to symbols in 16-QAM constellation diagram. The symbols are then transformed to time domain waveforms by OFDM modulation, which are fed in the SDR transmitter to radiate the wireless signals at carrier frequency into the air by the two transmit antennas. The constellation diagram of symbols in the data payload when they are emitted from the transmitter is presented in Fig. 8(a). The dispersion of constellation points results from the thermal noise in transmitter circuit and the non-ideality of the RF components (e.g., nonlinear power amplifier response). The RF signal is captured by the two receive antennas at the SDR receiver, and down-converted to baseband signal with the locally generated carrier frequency. After that, the channel matrix is estimated using pilot symbols and programmed into RRAM arrays as mentioned. The data payloads are then processed using RRAM-implemented modules (i.e., DFT and MIMO detection), and the recovered constellation diagram is presented in Fig. 8(c) with MER being 12.83dB. For comparison, the constellation points from a digital baseband processor are given in Fig. 8(b) whose MER is 17.43dB. The performance loss of our RRAM-implemented system mainly comes from the defective RRAM devices and programming errors, which are compensatory as discussed. Our demonstration proves the feasibility of system-

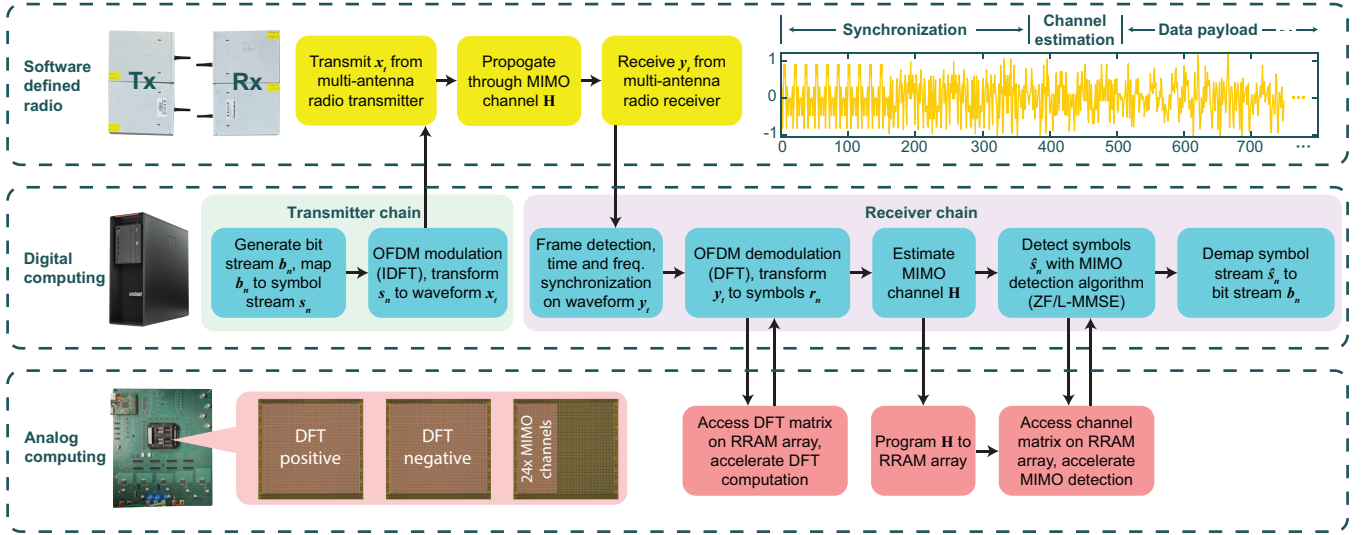


Figure 7. Proof-of-concept in-memory baseband processing experiment. A small-scale MIMO-OFDM system (32 sub-carriers, 2 transmit and 2 receive antennas) is validated in the experiment using SDRs, workstation and RRAM crossbar arrays. A schematic of the experimental system is presented. The realistic communication system is realized by SDR platform. Our in-memory computing test board provides physical measurement for the computation of DFT and MIMO detection modules. The computer bridges different platforms and controls the dataflow.

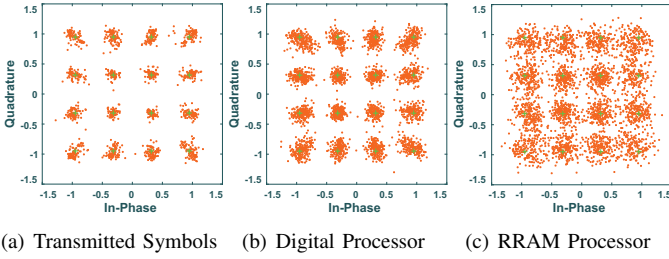


Figure 8. Proof-of-concept in-memory baseband processing experimental results: (a) Constellation diagram of transmitted symbols. (b, c) Constellation diagrams of the symbols recovered at the receiver from (b) digital baseband processing and (c) RRAM-implemented in-memory baseband processing.

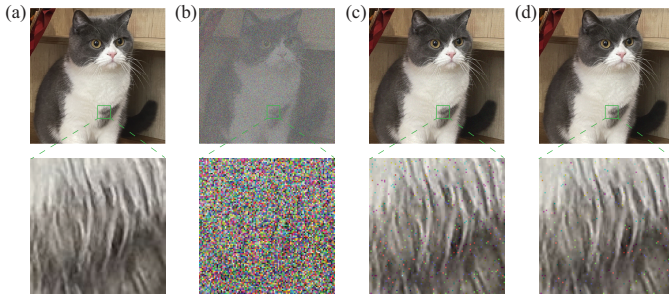


Figure 9. An illustration of the communication performance of transmitting an image over a noisy channel. (a) The original image. (b, c) The recovered images are from RRAM-based baseband processor where the RRAM arrays at the MIMO detector are programmed by (b) write-without-verification and (c) write-with-verification schemes. (d) Benchmark: software result.

level RRAM-based in-memory baseband processing in a real wireless communication system.

In the following, we perform the simulations of a large-scale RRAM-accelerated communication system corresponding to the standard of 5G new radio (NR) (see Table I). The simulation of RRAM array programming is based on the

RRAM model calibrated using the experimentally acquired device properties such as the Ohmic behaviour (see Fig. 2(f)) and the evolution of the conductance with voltage pulses (see Fig. 2(g)). Both the cycle-to-cycle variations and read noise during RRAM programming are included in our simulations. Since the transmitter is much simpler than the receiver, we focus on the RRAM-based receiver in the remainder of this section. To visually demonstrate the performance of our designed in-memory baseband processor, we consider the specific task of uncoded transmission of an image as shown in Fig. 9(a). The image recovered at the receiver are presented in Fig. 9(b) and Fig. 9(c) where RRAM devices are programmed using writing without and with verification schemes, respectively. As a benchmark for comparison, the image resulting from a digital baseband processor is shown in Fig. 9(d). One can observe that the performance of the write-without-verification scheme is poor while the other scheme with verification performs similarly as the ideal processor. To quantify the performance, we present the relation between MER (and BER) and SNR for both schemes as shown in Fig. 10(a) (and Fig. 10(b)). The simulation results are aligned with the earlier observation and show that write-with-verification scheme outperforms the other in terms of communication performance. From the perspective of latency and energy efficiency, the performance is compared in Fig. 10(c) and Fig. 10(d). As shown in Table II, the throughput and energy-efficiency of the proposed RRAM-based in-memory baseband processing exceed those of any reported CMOS-based digital processors [52]–[56]. For example, under the specifications in Table I, the throughput and energy efficiency can achieve up to 160.8Gb/s and 4637Gb/J, exceeding state-of-the-art digital counterparts in the literature by more than $329\times$ and $671\times$, respectively. Moreover, by reasonable conversion, our design is estimated to supports a throughput $91.2\times$ higher than one of the state-of-the-art commercial modems, i.e., Qualcomm Snapdragon

Table I
PARAMETERS FOR LARGE-SCALE MIMO-OFDM SYSTEM SIMULATION

Parameter	# sub-carriers	# Tx antennas	# Rx antennas	# Pilot symbols	# Symbols/frame
Notation	N_c	N_t	N_r	N_t	M
Value	1024	4	4	4	14×160

Table II
COMPARISON WITH CMOS-BASED DIGITAL PROCESSORS

Processor	Technology	Latency (ms)	Energy (mJ)	Communication Throughput (Gb/s)	Energy Efficiency (Gb/J)
Qualcomm Snapdragon X65 [52]	4 nm	<10	N/A	<5	N/A
TMS320C6678 8-core digital signal processor [53]	20 nm	589.9	6548	0.0621	0.0056
Domain adaptive processor 16×DAP in literature [54]	12 nm	74.95	6547	0.4888	0.0056
Combined digital baseband modules: FFT in [55] and MIMO detector in [56]	65 nm	50.17	5.3024	0.7303	6.9091
Proposed RRAM-based Baseband Processor	-	0.2278	0.0079	160.8	4637

Table III
PARAMETERS OF MEMRISTORS' BEHAVIORAL MODELS FOR THE SIMULATIONS

Memristor Device	Mechanism	Pulse width	State number	Cycle-to-cycle variation	G_{\max}/G_{\min}	Operation voltages
Ta/TaO _x /Pt our RRAM	Filament	10 ns	256	4.41% (P) 5.44% (D)	230.99/79.93 μS	0.65/-0.575 V
TiN/HZO/SiO ₂ /Si FeFET [57]	FeFET	75 ns	32	0.5%	1.79/0.04 μS	3.65/-2.95 V
Ag/PZT/Nb:SrTiO ₃ FTJ [58]	FTJ	10 ns	256	2.06%	80/1 μS	1.675/-3.5 V
		630 ps	150	3.65%	27.5/1 μS	4/-5 V

X65. Underpinning the improvements is the ultra-fast one-step baseband processing after channel estimation such that the baseband latency mostly comes from programming the RRAM arrays of MIMO detection module at the beginning of the frame. In contrast, for CMOS-based digital processors, data symbols are processed by executing the DFT (or FFT) and MIMO-detection algorithms using digital logic circuits, both suffering from high complexity as discussed. Next, there exists a tradeoff between communication performance and latency, i.e., higher performance requires better programming accuracy and thus longer latency. On the one hand, the write-without-verification scheme shows lower latency but poor communication performance in terms of BER and MER, a result of the intrinsic stochasticity of RRAM. On the other hand, RRAM with more states can achieve higher precision but possibly more pulses are needed to reach the target conductance value.

VII. DISCUSSION

This work demonstrates the feasibility of UFEE MIMO-OFDM baseband processing by leveraging the emerging in-memory computing technology based on RRAM arrays. The processing latency and energy are mostly contributed by the programming of the RRAM arrays for MIMO detection due to periodic channel estimation, while the following processing of data symbols can be completed in one-step. These advantages promise a feasible approach for realizing UFEE baseband processing. It shall be also emphasized that the proposed in-memory baseband processing not only works on RRAM but can be readily applied to other emerging in-memory

computing technologies including phase change, ferroelectric and magnetoresistive memories, as detailed in Table III which lists the device features of our experimental RRAM devices and other types of memristor. There are some observations from the simulation results in Fig. 11. First, we compared two different schemes for updating memristor arrays: writing with and without verification, elucidating the importance of verification and low cycle-to-cycle variation in ensuring the accuracy of the operations. To ensure satisfactory communication performance, write-with-verification is suggested for updating the memristor arrays even if the cycle-to-cycle variation is relatively small (e.g., $\sim 0.5\%$) as shown in the simulation result of programming ferroelectric FET (FeFET) [57] without verification. Second, memristor can be further improved using ultra-narrow pulse width along with relatively large number of states to achieve ultra-fast conductance updates without compromising the communication performance. For example, the simulation results in Fig. 11 show that the UFEE requirements can be met using ferroelectric tunnel junction (FTJ) which is reported for high precision attainable using sub-nanosecond pulses [58]. Leveraging the behavioural model of such memristors, the latency of our in-memory baseband processing system can be reduced to the scale of several microseconds and the energy consumption to the scale of several micro-Joules, which meets the UFEE requirements of 6G communications. Furthermore, in-memory baseband processing is more effective for applications with less stringent precision requirements. For example, if the transmitted messages, such as images, are inputs to the downstream neural networks for inference,

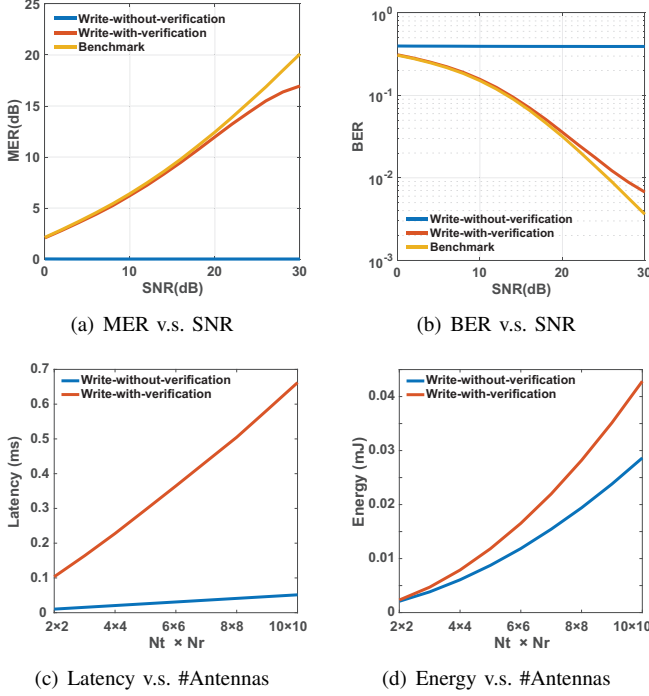


Figure 10. Performance evaluation of in memory MIMO-OFDM baseband processing using experimental RRAMs. The simulations target a large-scale MIMO-OFDM system of 1024 sub-carriers, 4 transmit antennas and 4 receive antennas unless specified otherwise. The behavioural model of RRAM devices comes from the experimental testing of our fabricated RRAM devices. The simulation curves are averaged over 200 trials to eliminate the randomness of channel and RRAM devices. (a, b) Under different channel conditions, the resultant (a) MER and (b) BER from the three schemes. (c, d) For writing with and without verification schemes, the (c) latency and (d) energy are evaluated in terms of different MIMO sizes.

the models' robustness against programming noise can ensure high classification accuracy. Overall, developing the proposed in-memory baseband processing into a versatile technology is believed to provide a feasible approach for realizing the 6G vision on supporting future services and applications with extremely low latency and high energy-efficiency.

APPENDIX

A. RRAM Device Fabrication and Integration

The integrated chip platform is comprised of three 64×64 Ta/TaO_x/Pt RRAM crossbar arrays, together with digital control and analogue sensing circuits to realize in-memory computing. The driving and sensing analogue circuits are taped out with TSMC's 180nm technology node. After the integration of the RRAM devices with the CMOS circuits, the chip is wire-bonded in a package. The RRAM devices have a lateral dimension of $50\text{nm} \times 50\text{nm}$, fabricated in house with back-end-of-line (BEOL) processes. The layers of the RRAM materials stack (Ta/TaO_x/Pt) are deposited with room temperature sputtering, and the electrodes are patterned with electron-beam lithography. The deposited TaO_x, serving as the switching layer, has a thickness of $\sim 2\text{nm}$.

B. DFT/IDFT Circuit

Consider the circuit with DFT matrix $\mathbf{W} \in \mathbb{C}^{N_c \times N_c}$. The real mapping of the DFT matrix $\mathcal{R}(\mathbf{W}) \in \mathbb{R}^{2N_c \times 2N_c}$ is scaled

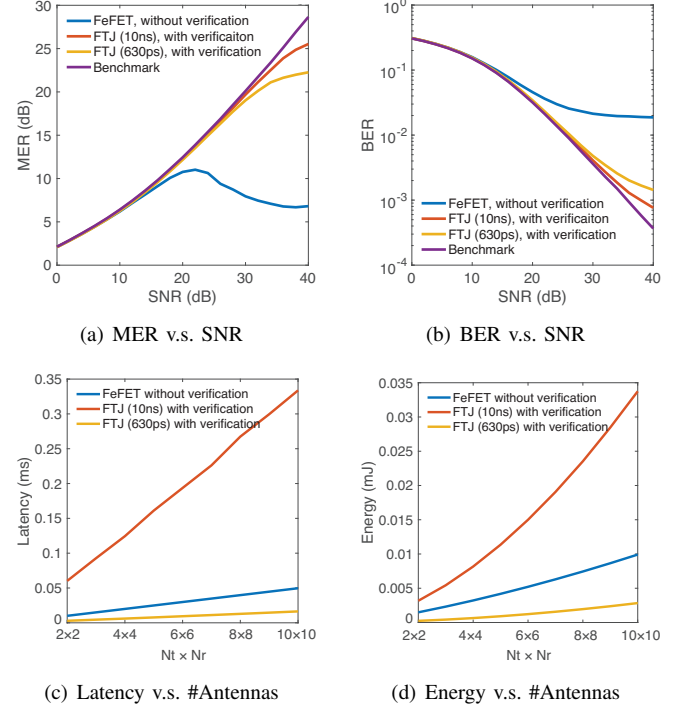


Figure 11. Performance evaluation of in memory MIMO-OFDM baseband processing using memristors in the literature. For the three memristor behavioural models, the (a) latency and (b) energy are evaluated in terms of different MIMO sizes. Under different channel conditions, the resultant (c) MER (in dB) and (d) BER are shown.

into the RRAM devices' conductance range by a scaling factor α , giving the conductance matrix $\mathbf{G} = \alpha \mathcal{R}(\mathbf{W}) \in \mathbb{R}^{2N_c \times 2N_c}$. The real conductance matrix \mathbf{G} is implemented by the difference between a pair of conductance arrays, $\mathbf{G}^+ - \mathbf{G}^-$, with the utilization of inverting amplifier to invert the voltages. The received signal $\mathbf{y} \in \mathbb{C}^{N_c \times 1}$ is translated to the input voltages with the real vector mapping, such that $\mathbf{v} = \mathcal{T}(\mathbf{y}) \in \mathbb{R}^{2N_c \times 1}$. Leveraging Ohm's law (i.e., current = conductance \times voltage), the multiplications $\{G_{kl}^+ v_l\}$ and $\{G_{kl}^- v_l\}$ are achieved. Then, Kirchhoff's current law sums these contributions along each row line and the read circuit integrates all the signals, giving the current at the k -th column $i_k = \sum_{l=1}^L (G_{kl}^+ - G_{kl}^-) v_l$. Therefore, the output currents at the read circuit give the result: $\mathbf{i} = (\mathbf{G}^+ - \mathbf{G}^-) \mathbf{v}$, which gives the DFT result $\alpha \mathcal{T}(\mathbf{x}) = \alpha \mathcal{R}(\mathbf{W}) \mathcal{T}(\mathbf{y})$. Since DFT matrix is unitary, i.e., $\mathbf{W}^{-1} = \mathbf{W}^H$, IDFT module circuit is the same as that of DFT when we replace $\mathcal{R}(\mathbf{W})$ with $\mathcal{R}(\mathbf{W})^T$.

C. L-MMSE/ZF MIMO Detector Circuit

Consider the L-MMSE detection circuit with channel matrix $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$. The real mapped channel matrix $\mathcal{R}(\mathbf{H}) \in \mathbb{R}^{2N_r \times 2N_t}$ is scaled into the RRAM devices' conductance range by a scaling factor α , giving the conductance matrix $\mathbf{G} = \mathbf{G}^+ - \mathbf{G}^- = \alpha \mathcal{R}(\mathbf{H}) \in \mathbb{R}^{2N_r \times 2N_t}$ which is implemented as the difference between two RRAM arrays. The real vector mapping of the received signal $\mathcal{T}(\mathbf{y}) \in \mathbb{R}^{2N_r \times 1}$ is translated to input currents. To make the voltages in the circuit within a reasonable range, the input currents are also scaled as

$\mathbf{i} = \alpha \mathcal{T}(\mathbf{y}) \in \mathbb{R}^{2N_r \times 1}$. The two arrays at the left-hand side constitute the conductance matrix $-\mathbf{G} = \mathbf{G}^- - \mathbf{G}^+$ with voltages \mathbf{v} supplied at the bottom of the nether array. The Kirchhoff's current law sums the output currents from the left RRAM array pair, $-\mathbf{G}\mathbf{v}$, and the input currents, \mathbf{i} , such that the input currents at the operational amplifiers are $\mathbf{i}' = -\mathbf{G}\mathbf{v} + \mathbf{i}$. Hence, the output voltages that supplied to the right RRAM array pair are $\mathbf{v}' = -\frac{\mathbf{i}'}{g_1} = \frac{\mathbf{G}\mathbf{v} - \mathbf{i}}{g_1}$, where g_1 is the feedback conductance of the TIAs. Then, the right RRAM array pair, whose conductance matrix is represented by $\mathbf{G}^\top = (\mathbf{G}^+ - \mathbf{G}^-)^\top$, performs the MVM computation and outputs the current vector $\mathbf{i}'' = \mathbf{G}^\top \mathbf{v}' = \mathbf{G}^\top \frac{\mathbf{G}\mathbf{v} - \mathbf{i}}{g_1}$. The currents are applied to the other set of TIAs, so that $\mathbf{i}'' = -g_2 \mathbf{v}$, where g_2 is the feedback conductance of the TIAs in this set. Accordingly, one can observe the relation: $\mathbf{G}^\top \frac{\mathbf{G}\mathbf{v} - \mathbf{i}}{g_1} = -g_2 \mathbf{v}$, which gives the output voltages $\mathbf{v} = (\mathbf{G}^\top \mathbf{G} + g_1 g_2 \mathbf{I})^{-1} \mathbf{G}^\top \mathbf{i}$. By setting the SNR as $\alpha^2 (g_1 g_2)^{-1}$, the designed L-MMSE circuit outputs the desired vector: $\mathcal{T}(\hat{\mathbf{x}}) = (\mathcal{R}(\mathbf{H})^\top \mathcal{R}(\mathbf{H}) + \frac{1}{\text{SNR}} \mathbf{I}_{2N_t \times 2N_t})^{-1} \mathcal{R}(\mathbf{H})^\top \mathcal{T}(\mathbf{y})$. When the feedbacks of the TIAs in the second set are open, i.e., $g_2 = 0$, the output voltages of the circuit are $\mathbf{v} = (\mathbf{G}^\top \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{i}$. This computes the ZF and gives the desired vector $\mathcal{T}(\hat{\mathbf{x}}) = (\mathcal{R}(\mathbf{H})^\top \mathcal{R}(\mathbf{H}))^{-1} \mathcal{R}(\mathbf{H})^\top \mathcal{T}(\mathbf{y})$.

D. Implementing Complex Matrices and Vectors

Both the channel entries and signals are in the complex domain while employing differential pairs of RRAM devices can only represent real numbers. To address this issue, we propose to apply the equivalent matrices and vectors of real entries instead. Inspired by the matrix representation of complex numbers, i.e., the mapping $a + bj \rightarrow \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$ is a ring isomorphism from the field of complex numbers to the ring of these matrices, we extend the method to complex matrices and define the mappings as follows.

Definition 1. (Real Matrix Mapping). Define the mapping $\mathcal{R} : \mathbb{C}^{K \times L} \rightarrow \mathbb{R}^{2K \times 2L}$, which transforms a complex matrix $\mathbf{A} = \Re(\mathbf{A}) + j\Im(\mathbf{A}) \in \mathbb{C}^{K \times L}$ into a real matrix $\mathcal{R}(\mathbf{A}) \in \mathbb{R}^{2K \times 2L}$:

$$\mathcal{R}(\mathbf{A}) = \begin{bmatrix} \Re(\mathbf{A}) & -\Im(\mathbf{A}) \\ \Im(\mathbf{A}) & \Re(\mathbf{A}) \end{bmatrix}. \quad (4)$$

The defined mapping preserves the basic operations of matrices (see Lemma 1), making it a feasible method for in-memory baseband processing implementation.

Lemma 1. (Properties of Equivalent Real Matrices). Some basic properties of the mapping \mathcal{R} defined in Definition 1 are described as follows. For any matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{K \times L}$, $\mathbf{C} \in \mathbb{C}^{L \times N}$,

$$\mathcal{R}(\mathbf{A}) + \mathcal{R}(\mathbf{B}) = \mathcal{R}(\mathbf{A} + \mathbf{B}), \quad (5)$$

$$\mathcal{R}(\mathbf{A})\mathcal{R}(\mathbf{C}) = \mathcal{R}(\mathbf{A}\mathbf{C}), \quad (6)$$

$$\mathcal{R}(\mathbf{A}^\text{H}) = \mathcal{R}(\mathbf{A})^\top. \quad (7)$$

The proof involves straightforward calculations of matrices and thus omitted for brevity. It can be inferred from the equation (6) that $\mathcal{R}(\mathbf{A}^{-1}) = \mathcal{R}(\mathbf{A})^{-1}$ if \mathbf{A} is invertible. Given the mapping, the complex matrices can be written into the

RRAM arrays without specific changes or auxiliary circuits. On the other hand, the proposed method can be applied to complex vectors as well, where one complex vector $\mathbf{x} \in \mathbb{C}^{K \times 1}$ is mapped to a real matrix $\mathcal{R}(\mathbf{x}) \in \mathbb{R}^{2K \times 2}$. Then the *matrix-vector multiplication* (MVM) can be achieved following equation (6). However, it takes two steps to complete the operation since $\mathcal{R}(\mathbf{x})$ is a matrix with two columns. To further improve the computational efficiency, we propose to implement the complex vector, which is usually the input voltages/currents for RRAM array, using the following transformation.

Definition 2. (Real Vector Mapping). Define the mapping $\mathcal{T} : \mathbb{C}^{K \times 1} \rightarrow \mathbb{R}^{2K \times 1}$, which transforms a complex vector $\mathbf{x} \in \mathbb{C}^{K \times 1}$ into a real vector $\mathcal{T}(\mathbf{x}) \in \mathbb{R}^{2K \times 1}$:

$$\mathcal{T}(\mathbf{x}) = \begin{pmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{pmatrix}. \quad (8)$$

The proposed mappings in Definition 1 and 2 make it possible to realize one-shot MVM computation as shown below.

Lemma 2. (One-Shot MVM Operation Between Equivalent Real Matrix and Vector). For any matrix $\mathbf{A} \in \mathbb{C}^{K \times L}$ stored in RRAM array and vector $\mathbf{x} \in \mathbb{C}^{L \times 1}$ translated as supply voltages, one-shot MVM is realized by the relation:

$$\mathcal{T}(\mathbf{A}\mathbf{x}) = \mathcal{R}(\mathbf{A})\mathcal{T}(\mathbf{x}). \quad (9)$$

It is proved by checking the following two relations:

$$\Re(\mathbf{A}\mathbf{x}) = \Re(\mathbf{A})\Re(\mathbf{x}) - \Im(\mathbf{A})\Im(\mathbf{x})$$

and

$$\Im(\mathbf{A}\mathbf{x}) = \Im(\mathbf{A})\Re(\mathbf{x}) + \Re(\mathbf{A})\Im(\mathbf{x}).$$

Based on Lemmas 1 and 2, the following two useful equations can be obtained:

$$\mathcal{T}((\mathbf{A}^\text{H}\mathbf{A})^{-1}\mathbf{A}^\text{H}\mathbf{x}) = (\mathcal{R}(\mathbf{A})^\top \mathcal{R}(\mathbf{A}))^{-1} \mathcal{R}(\mathbf{A})^\top \mathcal{T}(\mathbf{x}). \quad (10)$$

$$(\mathbf{A}^\text{H}\mathbf{A} + \lambda \mathbf{I})^{-1}\mathbf{A}^\text{H}\mathbf{x} = (\mathcal{R}(\mathbf{A})^\top \mathcal{R}(\mathbf{A}) + \lambda \mathbf{I})^{-1} \mathcal{R}(\mathbf{A})^\top \mathcal{T}(\mathbf{x}). \quad (11)$$

where λ is a constant. The above equations correspond to MIMO detection implementation.

E. Latency Analysis

We aim at quantifying the latency of writing a MIMO channel matrix into a RRAM array in the row-by-row manner. In particular, the channel is assumed as Rayleigh fading while the RRAM array is comprised of 1T1R cells. Two writing schemes are analyzed: write-without-verification and write-with-verification.

1) *Communication model:* The input-output relation of a MIMO system with channel matrix $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is described as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}, \quad (12)$$

where $\mathbf{x} \in \mathbb{C}^{N_t \times 1}$ and $\mathbf{y} \in \mathbb{C}^{N_r \times 1}$ denote the transmit and receive symbols, respectively. $\mathbf{z} \sim \mathcal{CN}(0, \sigma_n^2 \mathbf{I}_{N_r})$ represents the AWGN in propagation. We consider i.i.d. Rayleigh fading model, where the entries of the channel matrix \mathbf{H} follow i.i.d.

zero-mean complex Gaussian distribution. For any (i, j) -th element in matrix \mathbf{H} , we have $H_{ij} \sim \mathcal{CN}(0, 2\sigma_h^2)$ where $\Re(H_{ij}) \sim \mathcal{N}(0, \sigma_h^2)$ and $\Im(H_{ij}) \sim \mathcal{N}(0, \sigma_h^2)$.

The elements of channel matrix $\{H_{ij}\}$ vary in the whole real domain. We need to scale them into the feasible conductance range for RRAM devices. For each RRAM device, the maximum and minimum values of working conductance are denoted by G_{\max} and G_{\min} , respectively. For convenience, we assume $G_{\min} = 0$ such that a differential pair of RRAM devices can represent a real number in the interval $[-G_{\max}, G_{\max}]$. In order to scale the channel matrix into this range, we apply the three-sigma rule:

$$\Pr(-3\sigma_h \leq \Re(H_{ij}) \leq 3\sigma_h) = 99.73\%, \quad (13)$$

$$\Pr(-3\sigma_h \leq \Im(H_{ij}) \leq 3\sigma_h) = 99.73\%. \quad (14)$$

Through this rule, we can guarantee the feasibility of representing channel matrix by a RRAM array with high probability. Accordingly, the variance of the channel elements as mentioned becomes $\sigma_h = G_{\max}/3$.

2) *Model of RRAM device*: To begin with, we focus on the RRAM device whose behavioral model is shown in Fig. 2(f) and Fig. 2(g). Consider the conductance update process using a train of write pulses. The pulse width is denoted as Δt_w which is a minuscule value. Cycle-to-cycle variation σ_c refers to the variation in conductance change at every write pulse. It is expressed as the percentage of the entire conductance range in the existing literature [59]–[62]. In other words, the standard variance of the per-cycle write noise, $\mathcal{N}(0, \sigma_c^2)$, is presented by $\sigma_c = \gamma G_{\max}$ where $\gamma \in (0, 1)$ denotes the percentage. Then, we can characterize the per-pulse conductance change as follows:

$$\Delta G = \frac{G_{\max} - G_{\min}}{N_p} + \frac{\sigma_c}{\sqrt{\Delta t_w}} \Delta W, \quad (15)$$

where $\Delta G = G(t + \Delta t_w) - G(t)$ is the conductance change by applying one write pulse over $G(t)$, N_p is pulse number corresponding to programming conductance from G_{\min} to G_{\max} , and $\Delta W = W(t + \Delta t_w) - W(t)$ with $W(t)$ being a Wiener process: $\Delta W \sim \mathcal{N}(0, \Delta t_w)$. From (15), we know the following knowledge of the RRAM device's state after one pulse:

$$\mathbb{E}[G(t + \Delta t_w)|G(t)] = G(t) + \frac{G_{\max} - G_{\min}}{N_p \times \Delta t_w} \Delta t_w, \quad (16)$$

$$\text{Var}[G(t + \Delta t_w)|G(t)] = \left(\frac{\sigma_c}{\sqrt{\Delta t_w}} \right)^2 \Delta t_w. \quad (17)$$

By introducing the slope parameter $\mu \triangleq \frac{G_{\max} - G_{\min}}{N_p \times \Delta t_w}$ and the diverting variance $\sigma \triangleq \frac{\sigma_c}{\sqrt{\Delta t_w}}$, the evolution of conductance state $G(t)$ is characterized by the stochastic differential equation (SDE) given the initial state G_0 (i.e., the conductance at the time $t = 0$):

$$dG(t) = \mu dt + \sigma dW(t), \quad G(0) = G_0. \quad (18)$$

The solution of (18) gives an Itô process:

$$G(t) = G_0 + \mu t + \sigma \int_0^t dW(s). \quad (19)$$

3) *Performance of RRAM device*: Let $p(g, t|G_0)$ represent the conditional probability density of $G(t)$ given initial state $G(0) = G_0$. For the SDE specified in (18), the probability density of the solution satisfies the forward Kolmogorov equation (also known as Fokker-Planck equation) with the initial condition as follows:

$$\begin{aligned} \frac{\partial p(t, g)}{\partial t} &= -\mu \frac{\partial p(t, g)}{\partial g} + \frac{\sigma^2}{2} \frac{\partial^2 p(t, g)}{\partial g^2}, \\ p(0, g) &= \delta(g - G_0), \end{aligned} \quad (20)$$

where $\delta(\cdot)$ is Dirac function. By solving the partial differential equation in (20), we obtain the following lemma.

Lemma 3. (Probability Density). The probability density for the conductance evolution $G(t)$ is

$$p(t, g|G_0) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \exp\left(-\frac{(g - \mu t - G_0)^2}{2\sigma^2 t}\right), \quad (21)$$

which gives the Gaussian distribution $(G(t)|G_0) \sim \mathcal{N}(G_0 + \mu t, \sigma^2)$.

The process $\{G(t), t \geq 0\}$ is time homogeneous with independent increments. Without loss of generality, we assume the target conductance \bar{G} is larger than the initial state, i.e., the increment of conductance is positive: $\Delta G \triangleq \bar{G} - G_0 \geq 0$. The writing time is denoted as T , which aims at increasing the conductance by the amplitude of ΔG .

- **Write-without-verification scheme**

Given target conductance \bar{G} and the increment ΔG , the write latency of the RRAM device is determined by

$$T = \frac{\Delta G}{\mu}. \quad (22)$$

Meanwhile, the achieved conductance state $G(T)$ is inaccurate, giving that

$$G(T) \sim \mathcal{N}\left(\bar{G}, \frac{\sigma^2 \Delta G}{\mu}\right). \quad (23)$$

- **Write-with-verification scheme**

To achieve the target conductance \bar{G} , the read pulse is applied after each write pulse to monitor the evolution of conductance state $G(t)$. We model it as the first passage time, which refers to the first time when the conductance state $G(t)$ achieves the target value \bar{G} ,

$$T \triangleq \inf\{t \geq 0 : G(t) = \bar{G}\}. \quad (24)$$

By adding an absorbing boundary $p(t, \bar{G}) = 0$ to the partial differential equation (together with the initial condition) in (20), we obtain the probability density of the first passage time as the solution of the boundary value problem.

Lemma 4. (First Passage Time Probability Density). The probability density of the first passage time with the target increment conductance ΔG is given by

$$p(T|\Delta G) = \frac{\Delta G}{\sqrt{2\pi\sigma^2 T^3}} \exp\left(-\frac{(\Delta G - \mu T)^2}{2\sigma^2 T}\right), \quad (25)$$

which gives the inverse Gaussian distribution $(T|\Delta G) \sim \text{IG}(\Delta G/\mu, (\Delta G/\sigma)^2)$.

4) *Latency of RRAM array programming*: We consider the latency of writing the real mapped channel matrix $\mathcal{R}(\hat{\mathbf{H}})$ onto the 1T1R array in the row-by-row manner. For simplicity, we assume the array has been fully reset, i.e., all the RRAM devices are initialized with $G_{ij}(0) = 0, \forall(i, j)$. In this setting, G_{ij}^+ will be updated if $H_{ij} \geq 0$, or G_{ij}^- will be updated otherwise. The conductance change of one device, e.g., ΔG_{ij} for the (i, j) -th device, follows the i.i.d. half-normal distribution over indices $\forall(i, j)$, giving that

$$\Delta G_{ij} \sim |\mathcal{N}(0, G_{\max}^2/9)|, \quad i = 1, \dots, N_t, \quad j = 1, \dots, N_r. \quad (26)$$

The matrix $\mathcal{R}(\hat{\mathbf{H}})$ has $2N_r$ rows and it consists of two identical processes each with updating N_r rows. Thus, the expected latency of writing this matrix is

$$T_{\text{write}} = 2 \times \mathbb{E} \left[\sum_{i=1}^{N_r} T_i^{\text{row}} \right] = 2N_r \times \mathbb{E}[T_i^{\text{row}}], \quad (27)$$

where the expectation is taken over channel entries. One-row latency, say the i -th row, is determined by the RRAM device consuming the largest write time, that is

$$T_i^{\text{row}} = \max_{1 \leq j \leq 2N_t} T_{ij}, \quad (28)$$

where latencies $\{T_{ij}\}$ refer to writing $\Re(H_{ij'})$ and $\Im(H_{ij'})$, $j' = 1, \dots, N_t$ into the i -th row of RRAM array. Hence, latencies $\{T_{ij}\}_{j=1}^{2N_t}$ are $2N_t$ i.i.d. random variables.

• Write-without-verification scheme

Recall that the write time of one RRAM device using write-without-verification scheme is determined by $T = \Delta G/\mu$. Thus, for the i -th row, the write time of the RRAM device at the j -th column follows the half-normal distribution, that is

$$T_{ij} \sim |\mathcal{N}(0, \frac{G_{\max}^2}{9\mu^2})|, \quad j = 1, \dots, 2N_r. \quad (29)$$

where $|\mathcal{N}(\cdot, \cdot)|$ denotes the half-normal distribution.

Theorem 1. (*Expected Latency of Write-Without-Verification*). Consider writing the scaled real mapped channel matrix into an 1T1R array row-by-row using write-without-verification scheme. The expected write latency is upper bounded by

$$T_{\text{write}} \leq \frac{2\sqrt{2}G_{\max}}{3\mu} N_r \left(\sqrt{\ln N_t} + \frac{1}{\sqrt{\pi \ln N_t}} \right). \quad (30)$$

Proof: Since $T_i^{\text{row}} = \max_{1 \leq j \leq 2N_t} T_{ij}$, the probability of T_i^{row} satisfies

$$\begin{aligned} \Pr(T_i^{\text{row}} > t) &= 1 - \Pr\left(\max_{1 \leq j \leq 2N_t} T_{ij} \leq t\right) \\ &= 1 - (1 - \Pr(T_{ij} > t))^{2N_t}. \end{aligned} \quad (31)$$

Applying Bernoulli's inequality, we know that

$$(1 - \Pr(T_{ij} > t))^{2N_t} \geq 1 - 2N_t \Pr(T_{ij} > t). \quad (32)$$

Thus, for the non-negative random variable T_i^{row} , its expectation is expressed as

$$\begin{aligned} \mathbb{E}[T_i^{\text{row}}] &= \int_0^\infty \left(1 - (1 - \Pr(T_{ij} > t))^{2N_t}\right) dt \\ &\leq \varepsilon + \int_\varepsilon^\infty \left(1 - (1 - \Pr(T_{ij} > t))^{2N_t}\right) dt \\ &\leq \varepsilon + 2N_t \int_\varepsilon^\infty \Pr(T_{ij} > t) dt, \end{aligned} \quad (33)$$

where the inequality holds for any positive constant $\varepsilon > 0$. To obtain the probability $\Pr(T_{ij} > t)$, we introduce the *cumulative distribution function* (CDF) of T_{ij} , i.e., $F_{T_{ij}}(t) = \text{erf}\left(\frac{3\mu t}{\sqrt{2}G_{\max}}\right)$. To ease the notation, we denote the parameter $\bar{\sigma} \triangleq \frac{G_{\max}}{3\mu}$. Then the probability is given by

$$\Pr(T_{ij} > t) = 1 - F_{T_{ij}}(t) = 2Q(t/\bar{\sigma}), \quad (34)$$

where $Q(\cdot)$ is the Q-function. Leveraging the inequality $Q(x) \geq \frac{1}{x\sqrt{2\pi}} \left(1 - \frac{1}{x^2}\right) e^{-x^2/2}$ for $x > 0$, we have

$$\begin{aligned} \mathbb{E}[T_i^{\text{row}}] &\leq \varepsilon + 4N_t \int_\varepsilon^\infty Q\left(\frac{t}{\bar{\sigma}}\right) dt \\ &= \varepsilon + 2\sqrt{\frac{2}{\pi}} N_t \bar{\sigma} \exp\left(-\frac{\varepsilon^2}{2\bar{\sigma}^2}\right) - 4N_t \varepsilon Q\left(\frac{\varepsilon}{\bar{\sigma}}\right) \\ &\leq \varepsilon + 2\sqrt{\frac{2}{\pi}} N_t \bar{\sigma}^3 \exp\left(-\frac{\varepsilon^2}{2\bar{\sigma}^2}\right). \end{aligned} \quad (35)$$

Substituting $\varepsilon = \bar{\sigma}\sqrt{2\ln N_t}$ into the inequality for $N_t > 1$, we obtain

$$\begin{aligned} \mathbb{E}[T_i^{\text{row}}] &\leq \bar{\sigma}\sqrt{2\ln N_t} + \bar{\sigma}\sqrt{\frac{2}{\pi}} \frac{1}{\ln N_t} \\ &= \frac{\sqrt{2}G_{\max}}{3\mu} \left(\sqrt{\ln N_t} + \frac{1}{\sqrt{\pi \ln N_t}} \right). \end{aligned} \quad (36)$$

Finally, according to (27), the expected latency of updating the whole RRAM array is upper bounded by

$$T_{\text{write}} \leq \frac{2\sqrt{2}G_{\max}}{3\mu} N_r \left(\sqrt{\ln N_t} + \frac{1}{\sqrt{\pi \ln N_t}} \right). \quad (37)$$

This completes the proof. \square

• Write-with-verification scheme

Recall that the latency of writing one RRAM device, say the (i, j) -th device, in this scheme is determined by the first passage time with the compound distribution as follows:

$$\begin{aligned} T_{ij}|\Delta G_{ij} &\sim \mathcal{IG}(\Delta G_{ij}/\mu, (\Delta G_{ij})^2/\sigma^2), \\ \Delta G_{ij} &\sim |\mathcal{N}(0, G_{\max}^2/9)|. \end{aligned} \quad (38)$$

where $\mathcal{IG}(\cdot, \cdot)$ and $|\mathcal{N}(\cdot, \cdot)|$ denote the inverse Gaussian distribution and half-normal distribution, respectively.

Theorem 2. (*Expected Latency of Write-With-Verification*). Consider writing the scaled real mapped channel matrix into an 1T1R array row-by-row using

write-with-verification scheme. The expected write latency is upper bounded by

$$T_{\text{write}} \leq 2N_r \times \min \left\{ \frac{2\sqrt{2}G_{\max}}{3\mu} \sqrt{\ln(4N_t)}, \frac{2\sigma^2}{\mu^2} \ln(4N_t) + \frac{G_{\max}^2}{9\sigma^2} \right\}. \quad (39)$$

Proof: Recall that $T_i^{\text{row}} = \max_{1 \leq j \leq 2N_t} T_{ij}$ with i.i.d. $\{T_{ij}\}_{j=1}^{2N_t}$. Applying Jensen's inequality,

$$\begin{aligned} \mathbb{E}[\exp(\varepsilon T_i^{\text{row}})] &\leq \mathbb{E}[\exp(\varepsilon T_i^{\text{row}})] \\ &= \mathbb{E} \left[\max_{1 \leq j \leq 2N_t} \exp(\varepsilon T_{ij}) \right] \\ &\leq \sum_{j=1}^{2N_t} \mathbb{E}[\exp(\varepsilon T_{ij})] \\ &= 2N_t \mathbb{E}[\exp(\varepsilon T_{ij})], \end{aligned} \quad (40)$$

where the expectation follows the rule of compound distribution, that is

$$\mathbb{E}[\exp(\varepsilon T_{ij})] = \mathbb{E}_{\Delta G_{ij}} [\mathbb{E}[\exp(\varepsilon T_{ij}) | \Delta G_{ij}]]. \quad (41)$$

Given ΔG_{ij} , the latency T_{ij} follows an inverse Gaussian distribution, $T_{ij} \sim \mathcal{IG} \left(\frac{\Delta G_{ij}}{\mu}, \frac{(\Delta G_{ij})^2}{\sigma^2} \right)$, whose *moment-generating function* (MGF) is

$$\mathbb{E}[\exp(\varepsilon T_{ij}) | \Delta G_{ij}] = \exp \left\{ \frac{\mu \Delta G_{ij}}{\sigma^2} \left(1 - \sqrt{1 - \frac{2\sigma^2 \varepsilon}{\mu^2}} \right) \right\}. \quad (42)$$

To ease the notation, we define

$$x(\varepsilon) \triangleq \frac{\mu}{\sigma^2} \left(1 - \sqrt{1 - \frac{2\sigma^2 \varepsilon}{\mu^2}} \right), \quad (43)$$

which is an increasing function for $\varepsilon \in (0, \frac{\mu^2}{2\sigma^2}]$. Furthermore, we denote the standard variance of ΔG_{ij} as $\tilde{\sigma}$ whose value is $\tilde{\sigma} = \frac{G_{\max}}{3}$. Then, leveraging the MGF of the half-normal distribution, $\Delta G_{ij} \sim |\mathcal{N}(0, \tilde{\sigma}^2)|$, we have

$$\mathbb{E}[\exp(x(\varepsilon) \Delta G_{ij})] = \exp \left(\frac{\tilde{\sigma}^2 x(\varepsilon)^2}{2} \right) \left(1 + \operatorname{erf} \left(\frac{\tilde{\sigma} x(\varepsilon)}{\sqrt{2}} \right) \right). \quad (44)$$

From the relation $\exp(\varepsilon \mathbb{E}[T_i^{\text{row}}]) \leq 2N_t \mathbb{E}[\exp(\varepsilon T_{ij})] = 2N_t \mathbb{E}[\exp(x(\varepsilon) \Delta G_{ij})]$, we have

$$\begin{aligned} \mathbb{E}[T_i^{\text{row}}] &\leq \frac{1}{\varepsilon} \ln(2N_t) + \frac{\tilde{\sigma}^2 x(\varepsilon)^2}{2\varepsilon} \\ &\quad + \frac{1}{\varepsilon} \ln \left(1 + \operatorname{erf} \left(\frac{\tilde{\sigma} x(\varepsilon)}{\sqrt{2}} \right) \right). \end{aligned} \quad (45)$$

It is not hard to prove the following facts: $x(\varepsilon) \leq \frac{2}{\mu} \varepsilon$ and $\operatorname{erf}(\cdot) \leq 1$, and thus we have

$$\mathbb{E}[T_i^{\text{row}}] \leq \frac{1}{\varepsilon} \ln(4N_t) + \frac{2\tilde{\sigma}^2}{\mu^2} \varepsilon, \quad (46)$$

which holds for any $\varepsilon \in (0, \frac{\mu^2}{2\sigma^2}]$. This indicates the expected one-row latency $\mathbb{E}[T_i^{\text{row}}]$ is upper bounded by the minimum of the right-hand side of inequality (46).

1). If $\ln(4N_t) \leq \frac{\mu^2}{2\sigma^2} \left(\frac{\tilde{\sigma}}{\sigma} \right)^2$, the minimum value of the right-hand side of (46) is

$$\begin{aligned} \frac{1}{\varepsilon} \ln(4N_t) + \frac{2\tilde{\sigma}^2}{\mu^2} \varepsilon &\geq 2\sqrt{\frac{1}{\varepsilon} \ln(4N_t) \cdot \frac{2\tilde{\sigma}^2}{\mu^2} \varepsilon} \\ &= \frac{2\sqrt{2}G_{\max}}{3\mu} \sqrt{\ln(4N_t)}, \end{aligned} \quad (47)$$

and thus, we have

$$\mathbb{E}[T_i^{\text{row}}] \leq \frac{2\sqrt{2}G_{\max}}{3\mu} \sqrt{\ln(4N_t)}. \quad (48)$$

2). If $\ln(4N_t) > \frac{\mu^2}{2\sigma^2} \left(\frac{\tilde{\sigma}}{\sigma} \right)^2$, the minimum value of the right-hand side of inequality (46) is achieved at $\varepsilon = \frac{\mu^2}{2\sigma^2}$, so that

$$\begin{aligned} \frac{1}{\varepsilon} \ln(4N_t) + \frac{2\tilde{\sigma}^2}{\mu^2} \varepsilon &\geq \frac{2\sigma^2}{\mu^2} \ln(4N_t) + \left(\frac{\tilde{\sigma}}{\sigma} \right)^2 \\ &= \frac{2\sigma^2}{\mu^2} \ln(4N_t) + \frac{G_{\max}^2}{9\sigma^2}, \end{aligned} \quad (49)$$

and thus, we have

$$\mathbb{E}[T_i^{\text{row}}] \leq \frac{2\sigma^2}{\mu^2} \ln(4N_t) + \frac{G_{\max}^2}{9\sigma^2}. \quad (50)$$

By combining the results in the two cases, we have

$$\mathbb{E}[T_i^{\text{row}}] \leq \begin{cases} \frac{2\sqrt{2}G_{\max}}{3\mu} \sqrt{\ln(4N_t)}, & \ln(4N_t) \leq \frac{\mu^2}{2\sigma^2} \left(\frac{G_{\max}}{3\sigma} \right)^2, \\ \frac{2\sigma^2}{\mu^2} \ln(4N_t) + \frac{G_{\max}^2}{9\sigma^2}, & \ln(4N_t) > \frac{\mu^2}{2\sigma^2} \left(\frac{G_{\max}}{3\sigma} \right)^2, \end{cases} \quad (51)$$

which can be easily verified that it is equivalent to

$$\mathbb{E}[T_i^{\text{row}}] \leq \min \left\{ \frac{2\sqrt{2}G_{\max}}{3\mu} \sqrt{\ln(4N_t)}, \frac{2\sigma^2}{\mu^2} \ln(4N_t) + \frac{G_{\max}^2}{9\sigma^2} \right\}. \quad (52)$$

Finally, according to (27), the expected latency of updating the whole RRAM array is upper bounded by

$$T_{\text{write}} \leq 2N_r \times \min \left\{ \frac{2\sqrt{2}G_{\max}}{3\mu} \sqrt{\ln(4N_t)}, \frac{2\sigma^2}{\mu^2} \ln(4N_t) + \frac{G_{\max}^2}{9\sigma^2} \right\}. \quad (53)$$

This completes the proof. \square

F. Comparison with Digital CMOS Counterpart

In this note, we provide the comparisons with the state-of-the-art (SOTA) digital CMOS counterparts, which is important to highlight the advantages of our RRAM-based baseband processor in terms of latency (or throughput) and energy efficiency. The calculations are based on the parameters in the Table IV unless specified otherwise.

Table IV
PARAMETERS FOR COMPARISON WITH CMOS-BASED DIGITAL PROCESSORS

Parameter	DFT Size	# Slots/Frame	Symbols/Slot	Modulation	Number of Antennas
Value	1024 2048*	160	14	16-QAM	4×4 $2 \times 2 + 4 \times 4^*$

*2048 and $2 \times 2 + 4 \times 4$ are only used for comparison with Qualcomm Snapdragon X65.

1) *Comparison with SOTA commercial modem:* SOTA commercial modems are fabricated using the latest technology node, targeting the 5G signal processing. Take the popular Qualcomm Snapdragon X65 modem as an example. It is fabricated using the TSMC 4nm process according to the public data [52]. The maximum download speed can achieve up to 10 Gb/s over mmWave and sub-6 carrier aggregation. Note that the modulation used in this product is 256-QAM (8 bits/symbol) while we used 16-QAM (4 bits/symbol) in this work for the demonstration of our RRAM-based design. Therefore, the top speed is reduced by half as we match the QAM order of the two processors for a fair comparison, resulting in a peak throughput of 5 Gb/s as benchmark. In addition, the throughput of X65 modem is the combination of two separate bands and specifications, i.e., mmWave (2×2 MIMO) and sub-6 GHz (4×4 MIMO). For the same communication overhead, the baseband processing latency of our design with the same parameters (2048-point DFT, 2×2 MIMO + 4×4 MIMO) is only 0.2409 ms by re-simulation. It means our design can support a communication throughput as large as 455.8 Gb/s, which is $91.2 \times$ higher than that of the SOTA (i.e., Snapdragon X65 modem). However, it is hard to scale the throughput to fit the other group of settings specified in Table 1 which are used for other baselines, so that we just summarize it as < 5 Gb/s in Table II, making it appear different from the values in other designs. On the other hand, the energy efficiency comparison is not feasible since the X65 modem is an integrated system-on-chip (SoC) that contains RFIC, control processor, digital baseband processor, and other units. Unfortunately, there is no detailed energy efficiency data related to baseband processing in this modem available in the public domain.

2) *Comparison with multi-core digital signal processor (DSP) reported in literature:* Next, we compare the proposed design with the reported powerful multi-core DSP in the literature, namely TMS320C6678 from Texas Instruments [53]. The performance achieves up to 128 GOPS while the average power consumption for the optimized digital processing function is 11.1 Watts. For fair comparison, we just take the power of this 8-core DSP into consideration while neglect the power of the development board TMDSEVM6678LE. The baseband processing workload is estimated by the algorithmic computation complexity, giving the value of 75.5 GOPs. Accordingly, the latency and energy consumption are calculated as 589.9 ms and 6.548 J, respectively. Meanwhile, ignoring the power of peripheral circuits, our RRAM-based design shows the latency and energy consumption being 0.2278 ms and 0.0079 mJ, and thus the equivalent computational throughput and energy efficiency are calculated as 331.4 TOPS and 9557 TOPS/Watt which outperforms this DSP by 10^3 and 10^5 times,

respectively.

3) *Comparison with digital baseband processor reported in literature:* Moreover, we compare the proposed design with the reported digital baseband processor in the literature. Since the design of traditional digital baseband processor is considered a matured area, there are few recent publications on the complete baseband system design. As a compromise, we provide the following two comparisons: a) one with the SOTA domain adaptive processor (DAP) for wireless communication; b) the other with “virtually assembled” digital baseband processor by combining the SOTA designs of isolated digital baseband modules (i.e., DFT, MIMO detection, etc.) collected from the recent literatures.

- a) *Comparison with SOTA digital adaptive processor (DAP):* The accelerator presented in the reference [54] has been fabricated by a 12nm technology node and specialized for wireless communication workloads. Its peak performance reaches 264 GOPS at power consumption of 272 GOPS/Watt. One can observe it has a noticeable performance gain compared with commercial DSPs [53]. However, the algorithms for computing FFT and MMSE in this design are different from the discussions in Supplementary Note 9, so it is not reasonable to directly compare the computational performance. Hence, we evaluate the performance of the baseline in terms of communication throughput and energy efficiency in the way as follows. The measurement results of this DAP show the throughput and energy efficiency are 4.41G samples/s and 53.96 nJ/FFT for 256-point FFT. For a fair comparison, we consider the joint utilization of 16 DAPs to complete the 1024-point DFT by decomposing this large-scale DFT into 16 small-scale 256-point FFTs. Accordingly, the latency is saved at the cost of more area and energy consumption. On the other hand, the results from the measurements on MMSE MIMO detection reveal the throughput and energy efficiency being 1.95M matrices/s and 178.5 nJ/matrix, respectively. Moreover, there are additional reprogram times in OFDM demodulation ($0.5\mu\text{s}$) and MIMO detection ($0.2\mu\text{s}$). Accordingly, the latency of the assembled 16 DAPs is $14 \times 160 \times 4 / (4.41 \times 10^9) + (14 \times 160 - 4) \times (1024 / (1.95 \times 10^6 \times 16) + (0.5 + 0.2) \times 10^{-6}) = 0.0750\text{s}$, and the energy consumption is $14 \times 160 \times 4 \times 53.96 \times 10^{-9} \times 16 + (14 \times 160 - 4) \times 1024 \times 178.5 \times 10^{-9} \times 16 = 6.547\text{J}$. In comparison, our RRAM-based design is $329.2 \times$ faster and $> 10^5$ more energy efficient than this baseline (i.e., 16-DAPs).
- b) *Comparison with the combination of isolated digital baseband modules:* For the DFT module, we use the high-throughput FFT processor proposed in [55]. The pro-

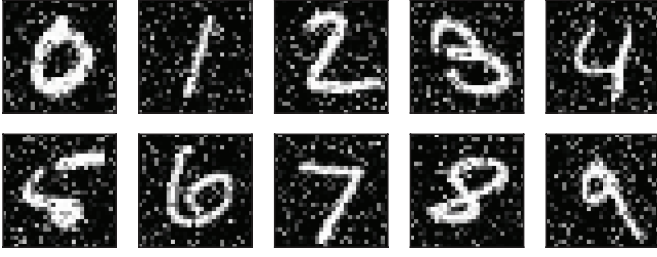


Figure 12. An illustration of noisy test images. In this example, the test images are uncoded transmitted over a noisy channel (SNR = 15dB) and recovered by RRAM-based baseband processing.

cessor is fabricated using 65nm CMOS technology and is designed to support 5G high-speed requirements. The clock frequency is 250MHz. The processing latency and energy consumption for one 1024-point FFT operation are estimated using the given values of 1080-point FFT (688 clock cycles) and 1200-point FFT (2.07 FFTs/ μ J), respectively.

For the L-MMSE MIMO detection module, consider the design presented in [56] which was fabricated using the 65nm CMOS technology. The clock frequency is 625MHz. The digital MIMO detector is based on the lower-upper (LU) decomposition which is a two-step algorithm. To estimate the computational workload of solving the related linear equations, the classical Gaussian elimination algorithm is adopted, whose complexity is contributed by two parts: the forward elimination and the backward substitution. Using the terminology, we count the forward substitution step once for channel matrix on all N_c sub-carriers and backward substitution up to $(M - N)$ times for all received data symbols on each sub-carrier. As for the energy consumption, a measurement value of 19.2pJ/b is presented in the literature, which is equivalent to $19.2 \times 8 = 153.6$ pJ per MIMO detection.

For the combined baseband processing system, the sum processing latency of the two modules with respective 250MHz and 625MHz clocks for DFT and MIMO modules is given by: $688 \times 14 \times 160 / (250 \times 10^6) + 1024 \times (24 + 12 \times (14 \times 160 - 4)) / (625 \times 10^6) = 0.0502s$. In contrast, the latency of our design completing the operations under the same parameters is only 0.2321ms, approximately $220.4 \times$ faster than the digital CMOS counterpart.

On the other hand, the energy consumption of the digital CMOS design is estimated as follows: $(14 \times 160 / 2.07) \times 10^{-6} + 153.6 \times 12 \times (14 \times 160 - 4) \times 1024 \times 10^{-12} = 0.0053J$, which is approximately $670.9 \times$ more than our proposed design. It is worth mentioning that the processing latency of the considered digital processor does not meet the stringent requirement of 5G. Traditionally, in order to reduce the processing time to below 10 ms, multiple MIMO detection units should be employed to enable parallel processing, with the sacrifices in chip area and energy consumption.

The discussions on comparison with CMOS-based digital processors are summarized in the Table II.

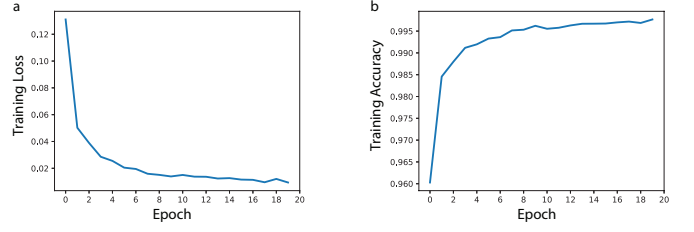


Figure 13. The history of training process. (a) Training loss versus epoch. (b) Training accuracy versus epoch.

G. Neural Network Application

In this note, we demonstrate that in-memory baseband processing is more effective for applications with less stringent precision requirements. For example, if the transmitted messages, such as images, are inputs to the downstream neural networks for inference, the models' robustness against programming noise can ensure high classification accuracy. The simulation is divided into two steps: 1) image transmission; 2) neural network inference. The two procedures are simulated in C++ and Python (TensorFlow), respectively.

1) *Noisy Images*: Consider the scenario that images are transmitted from sender to destination over a noisy wireless channel. The receiver is equipped with the proposed RRAM-based baseband processor. The images then are fed into a neural network for classification. Due to channel and circuit noise, bit errors occur in the received images. In the simulation, we transmit the test dataset (10,000 images) of MNIST database over noisy channels (characterized by SNRs) using MIMO-ODFM air interface. Then the receiver decodes the images through RRAM-based baseband processing. The parameters follow the standard of 5G NR as well as the behavioral model of our fabricated RRAM device. The examples of resultant noisy test images are shown in Fig. 12.

2) *Neural Network*: The neural network is located at the receiver side to perform classification tasks. The architecture is described as follows: The classifier model is implemented by a 5-layer convolutional neural network (CNN) that consists of two 3×3 convolutional layer, each followed with a 2×2 max pooling, two fully connected layers (the first with 512 units, the second with 64) with ReLU activation, and a final softmax output layer. The CNN classifier is trained using noiseless training dataset (60,000 images) of MNIST database. The objective of training is to minimize the loss function, which is chosen as cross entropy, using the Adam optimizer. The training process in terms of loss and accuracy is illustrated in Fig. 2. After training, the training and testing accuracy achieve up to 99.73% and 98.74%, respectively.

3) *Inference Task*: After transmission and baseband processing, the noisy images (i.e., test dataset) are fed into the trained CNN classifier for inference. We perform simulation under noisy channels with different SNRs. The inference performance is compared to the accuracy from noiseless test dataset, and the performance losses are calculated. The results are listed in Table V.

Table V
PERFORMANCE OF RRAM-BASED BASEBAND PROCESSING FOR INFERENCE TASK

Channel Condition SNR (dB)	Digital Baseband Processing Inference Accuracy (%)	RRAM-based Baseband Processing Inference Accuracy (%)	Performance Loss (%)
30	98.69	98.58	0.11
25	98.48	98.34	0.14
20	97.91	97.48	0.43
15	94.11	93.82	0.29

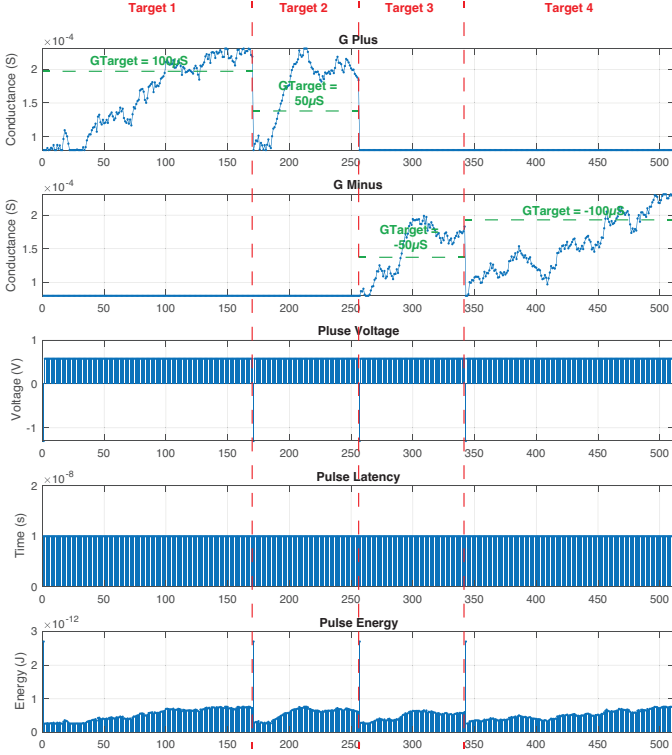


Figure 14. Conductance tuning process of a differential RRAM pair using write-without-verification scheme. (a) The trace of conductance changes of the RRAM device representing the positive value in the differential pair, denoted as G Plus. (b) The trace of conductance changes of the RRAM device representing the negative value in the differential pair, denoted as G Minus. (c) The voltages of write pulses applied to G Plus or G Minus. Each write pulse causes a corresponding conductance change of G Plus or G Minus. (d, e) The latency and energy induced by applying the corresponding write pulses.

H. Simulations of Conductance Programming

In this note, we discuss the modeling of RRAM's writing process in our simulations. The simulation codes are developed in C++, compiled in Clang++-13, and run on Linux system. The simulation is based on the measurement of conductance programming characteristic of our fabricated RRAM device. We considered two schemes: 1) "write-without-verification", which pre-determines the number and magnitude of write pulses without monitoring the conductance change during the programming process; 2) "write-with-verification", which used a read pulse after each write pulse to ensure the conductance of the RRAM device can be set to high accuracy. For comparison and clarification, we summarize the two writing schemes as follows:

- Write-without-verification scheme: The number of pulses

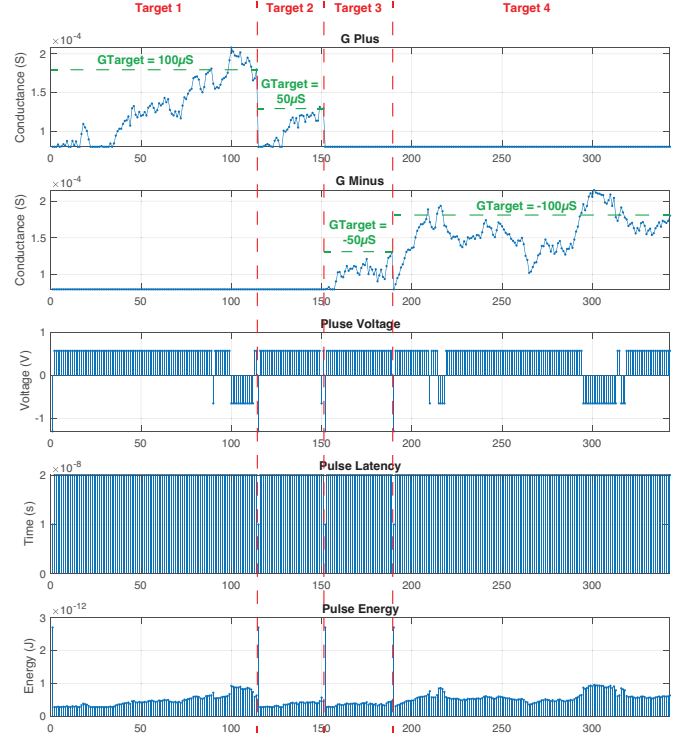


Figure 15. Conductance tuning process of a differential RRAM pair using write-with-verification scheme. (a) The trace of conductance changes of the RRAM device representing the positive value in the differential pair, denoted as G Plus. (b) The trace of conductance changes of the RRAM device representing the negative value in the differential pair, denoted as G Minus. (c) The voltages of write pulses applied to G Plus or G Minus. Each write pulse causes a corresponding conductance change of G Plus or G Minus. (d, e) The latency and energy induced by applying the corresponding write pulses.

and the voltage direction are determined based on the fabricated RRAM behavioral model shown as the curve presented in Fig. 2(g). Then the write pulses are applied to the RRAM device regardless of its intermediate states, i.e., no read operation is applied after each write pulse.

- Write-with-verification scheme: A read pulse is applied after each write pulse to get the RRAM device's conductance. In this way, the state of RRAM is monitored in real-time during the writing process. Based on the read-out conductance value, the controller determines whether the target is met, or additional positive (or negative) pulse should be applied.

We emphasize that the simulation is more authentic than the model in theoretical analysis which is simplified to obtain the theoretical scaling law. The simulation results come from the more accurate modeling of RRAM's behaviour and com-

plicated programming algorithm. To be specific, the pulses supplied are discrete and thus the conductance change may exceed the threshold. Then the negative pulses will be applied to fine tune the conductance of RRAM. The process may involve multiple stages of positive pulses and negative pulses before convergence to the target value. Moreover, the read noise is included in our simulations ($\sim 1\mu S$), which also affects the precision of conductance programming and thus deteriorate the performance. We present Fig. 14 and Fig. 15 to illustrate the evolution of conductance values, supplied voltage pulses, latency, and energy consumption during updating a RRAM differential pair from initial state to conductance targets. For convenience, we name the RRAM device representing the positive value in the differential pair as G Plus, and the other one as G Minus. The conductance programming follows the exact procedures described as follows:

- Step 1: Initialization. A negative pulse with high voltage (-1.5V) is applied to both G Plus and G Minus, such that the two RRAM devices are fully reset to the minimum conductance state.
- Step 2: Tuning. The write pulses are applied to either G Plus or G Minus, depending on the conductance target. If the target is positive, only G Plus is tuned while G Minus remains unchanged during the writing process, vice versa. For write-without-verification scheme, the pre-determined number of pulses are supplied to the RRAM device. For write-with-verification scheme, the iterations between write pulse (0.65V/-0.575V) and read pulse (0.15V) proceed until the conductance of the RRAM device is close enough to the target value within a tolerable error range.

Our simulations are validated by the congruence between the conductance tuning trace depicted in Fig. 14 and Fig. 15, and the corresponding real measurement in the experiments.

I. Mapper and Demapper Modules

In this note, we present the designs for mapper and demapper modules. The mapper module is a binary to Gray code converter, which converts binary code at the beginning of baseband processing at the transmitter side. On the contrary, the demapper module is a Gray to binary code converter, which converts Gray code back to binary code at the end of baseband processing at the receiver side.

1) *Gray Code*: The Gray code, also known as reflected binary code, is defined as an ordering of the binary numeral system such that the two adjacent values only differ in one bit. Table VI shows the relation between decimal numbers, binary and Gray codes.

2) *Mapper Module*: As for binary to Gray code conversion, several observations can be made from Table VI as follows: 1) The first bit of Gray code is equal to the most significant bit of binary code; 2) The second bit of Gray code is the XOR of the first and second bits of the binary code; 3) The third bit of Gray code is the XOR of second and third bits of the binary code; 4) The fourth bit of Gray code is the XOR of third and fourth bits of the binary code. Based on these observations, the logical circuit for binary to Gray code converter (i.e.,

Table VI
THE CONVERSION BETWEEN BINARY CODE TO GRAY CODE AND DECIMAL TO GRAY CODE

Decimal Numbers	Binary Code	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

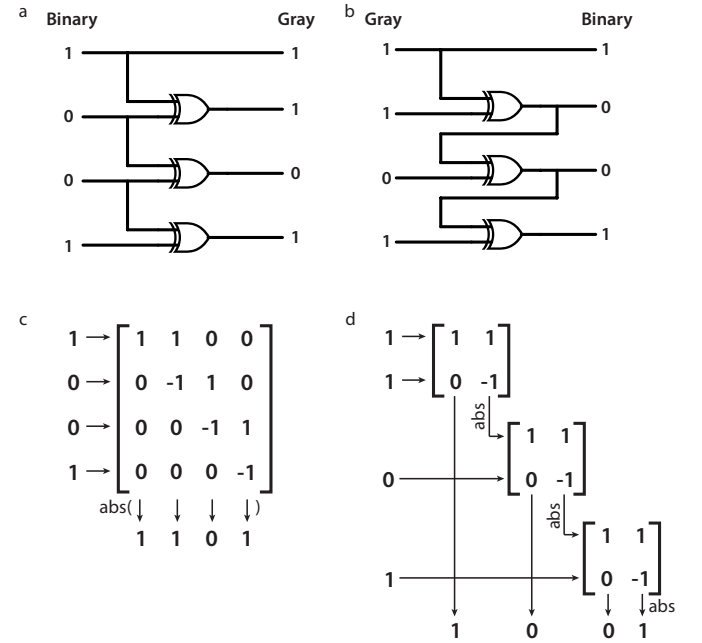


Figure 16. An illustration of the implementations of mapper and demapper. (a) At the transmitter, the logical circuit maps binary bits to Gray codes. After that, they are transformed to analogue values using DAC. (b) At the receiver, the outputs from ADC are Gray coded. The logical circuit maps Gray code back to binary bits. (c) At the transmitter, the codebook as shown in the matrix form is stored in RRAM array. The binary bits are translated as input voltages for the RRAM array. The magnitudes of outputs give the Gray codes. (d) At the receiver, the matrix as shown is stored in three RRAM arrays with the connection between them. The Gray codes are translated as input voltages for them, and the outputs give the binary bits.

mapper) is presented in Fig. 16(a). To implement the mapper with RRAM array(s), we transform the operations of binary to Gray code converter into matrix-vector multiplication(s). An illustration of the principle of RRAM-based mapper module is shown in Fig. 16(c). The circuit architecture is presented in Fig. 17(a).

3) *Demapper Module*: As for Gray to binary code conversion, several observations can be made from Table VI as follows: 1) The most significant bit of binary code is equal to the first bit of Gray code; 2) If the second bit of Gray code is

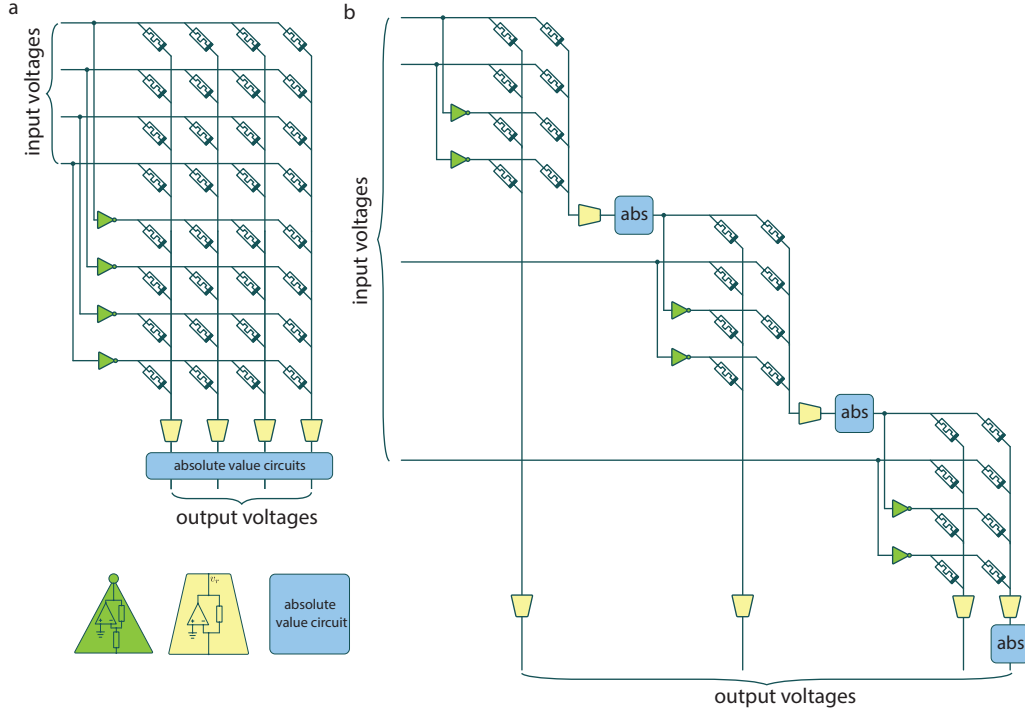


Figure 17. The circuit design of RRAM-based mapper and demapper. (a) The architecture of RRAM-based mapper module. (b) The architecture of RRAM-based demapper module.

0, the second bit of binary code is the same as the previous bit. If the second Gray bit is 1, the second binary bit is the opposite of the previous bit; 3) The operations in 2) continues for the remaining bits. Based on these observations, the logical circuit for Gray to binary code converter (i.e., demapper) is presented in Fig. 16(b). To implement the demapper with RRAM array(s), we transform the operations of Gray to binary code converter into matrix-vector multiplication(s). An illustration of the principle of RRAM-based demapper module is shown in Fig. 16(d). The circuit architecture is presented in Fig. 17(b).

REFERENCES

- [1] T. Ghosh, R. Saha, A. Roy, S. Misra, and N. S. Raghuvanshi, "AI-based communication-as-a-service for network management in society 5.0," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4030–4041, 2021.
- [2] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?" *Nat. Electron.*, vol. 3, no. 1, pp. 20–29, 2020.
- [3] M. W. Akhtar, S. A. Hassan, R. Ghaffar, H. Jung, S. Garg, and M. S. Hossain, "The shift to 6g communications: vision and requirements," *Human-Centric Comput. Inf. Sci.*, vol. 10, no. 1, pp. 1–27, 2020.
- [4] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6g wireless systems: Vision, requirements, challenges, insights, and opportunities," *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, 2021.
- [5] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, "Cellular, wide-area, and non-terrestrial iot: A survey on 5g advances and the road towards 6g," *IEEE Commun. Surveys Tuts.*, 2022.
- [6] N. Rajatheva, I. Atzeni, E. Bjornson, A. Bourdoux, S. Buzzi, J.-B. Dore, S. Erkucuk, M. Fuentes, K. Guan, Y. Hu *et al.*, "White paper on broadband connectivity in 6g," *arXiv:2004.14247*, 2020.
- [7] A. H. Sodhro, S. Pirbhulal, Z. Luo, K. Muhammad, and N. Z. Zahid, "Toward 6g architecture for energy-efficient communication in iot-enabled smart automation systems," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5141–5148, 2020.
- [8] P. Popovski, Č. Stefanović, J. J. Nielsen, E. De Carvalho, M. Angelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless access in ultra-reliable low-latency communication (urllc)," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5783–5801, 2019.
- [9] O. Holland, E. Steinbach, R. V. Prasad, Q. Liu, Z. Dawy, A. Aijaz, N. Pappas, K. Chandra, V. S. Rao, S. Oteafy *et al.*, "The ieee 1918.1 'tactile internet' standards working group and its standards," *Proc. IEEE*, vol. 107, no. 2, pp. 256–279, 2019.
- [10] N. Promwongsa, A. Ebrahimzadeh, D. Naboulsi, S. Kianpisheh, F. Belqasmi, R. Glitho, N. Crespi, and O. Alfandi, "A comprehensive survey of the tactile internet: State-of-the-art and research directions," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 472–523, 2020.
- [11] E. Steinbach, M. Strese, M. Eid, X. Liu, A. Bhardwaj, Q. Liu, M. Al-Ja'afreh, T. Mahmoodi, R. Hassen, A. El Saddik *et al.*, "Haptic codecs for the tactile internet," *Proc. IEEE*, vol. 107, no. 2, pp. 447–470, 2018.
- [12] A. Clemm, M. T. Vega, H. K. Ravuri, T. Wauters, and F. De Turck, "Toward truly immersive holographic-type communication: Challenges and solutions," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 93–99, 2020.
- [13] K. Wakunami, P.-Y. Hsieh, R. Oi, T. Senoh, H. Sasaki, Y. Ichihashi, M. Okui, Y.-P. Huang, and K. Yamamoto, "Projection-type see-through holographic three-dimensional display," *Nat. Commun.*, vol. 7, no. 1, pp. 1–7, 2016.
- [14] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital twin for 5g and beyond," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 10–15, 2021.
- [15] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6g," *IEEE Commun. Surveys Tuts.*, 2021.
- [16] H. Sarieddeen, M.-S. Alouini, and T. Y. Al-Naffouri, "An overview of signal processing techniques for terahertz communications," *Proc. IEEE*, 2021.
- [17] P. Skrimponis, S. Dutta, M. Mezzavilla, S. Rangan, S. H. Mirfarshbafan, C. Studer, J. Buckwalter, and M. Rodwell, "Power consumption analysis for mobile mmwave and sub-thz receivers," in *6G Wireless Summit*. IEEE, 2020, pp. 1–5.
- [18] K. Rikkinen, P. Kyosti, M. E. Leinonen, M. Berg, and A. Parssinen, "Thz radio communication: Link budget analysis toward 6g," *IEEE Commun. Mag.*, vol. 58, no. 11, pp. 22–27, 2020.
- [19] N. G. Orji, M. Badaroglu, B. M. Barnes, C. Beitia, B. D. Bunday, U. Celano, R. J. Kline, M. Neisser, Y. Obeng, and A. Vldar, "Metrology for the next generation of semiconductor devices," *Nat. Electron.*, vol. 1, no. 10, pp. 532–547, 2018.

- [20] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez, and T. B. Schardl, "There's plenty of room at the top: What will drive computer performance after moore's law?" *Science*, vol. 368, no. 6495, p. eaam9744, 2020.
- [21] B. Van Straalen, *Method of Local Corrections Solver for Manycore Architectures*. University of California, Berkeley, 2018.
- [22] M. Di Ventra and Y. V. Pershin, "The parallel approach," *Nature Physics*, vol. 9, no. 4, pp. 200–202, 2013.
- [23] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.
- [24] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nat. Electron.*, vol. 1, no. 6, pp. 333–343, 2018.
- [25] Z. Wang, H. Wu, G. W. Burr, C. S. Hwang, K. L. Wang, Q. Xia, and J. J. Yang, "Resistive switching materials for information processing," *Nat. Rev. Mater.*, vol. 5, no. 3, pp. 173–195, 2020.
- [26] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nat. Nanotechnol.*, vol. 15, no. 7, pp. 529–544, 2020.
- [27] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1864–1878, 2014.
- [28] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. Di Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. Farinha *et al.*, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, 2018.
- [29] G. W. Burr, R. M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti *et al.*, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498–3507, 2015.
- [30] B. Yan, Q. Yang, W.-H. Chen, K.-T. Chang, J.-W. Su, C.-H. Hsu, S.-H. Li, H.-Y. Lee, S.-S. Sheu, M.-S. Ho *et al.*, "Rram-based spiking nonvolatile computing-in-memory processing engine with precision-configurable in situ nonlinear activation," in *Symposium VLSI Technol.* IEEE, 2019, pp. T86–T87.
- [31] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, 2020.
- [32] X. Zhang, J. Lu, Z. Wang, R. Wang, J. Wei, T. Shi, C. Dou, Z. Wu, J. Zhu, D. Shang *et al.*, "Hybrid memristor-cmos neurons for in-situ learning in fully hardware memristive spiking neural networks," *Sci. Bulletin*, vol. 66, no. 16, pp. 1624–1633, 2021.
- [33] S. Woźniak, A. Pantazi, T. Bohnstingl, and E. Eleftheriou, "Deep learning incorporating biologically inspired neural dynamics and in-memory computing," *Nat. Mach. Intell.*, vol. 2, no. 6, pp. 325–336, 2020.
- [34] X. Zhang, Y. Zhuo, Q. Luo, Z. Wu, R. Midya, Z. Wang, W. Song, R. Wang, N. K. Upadhyay, Y. Fang *et al.*, "An artificial spiking afferent nerve based on mott memristors for neurorobotics," *Nat. Commun.*, vol. 11, no. 1, pp. 1–9, 2020.
- [35] Q. Duan, Z. Jing, X. Zou, Y. Wang, K. Yang, T. Zhang, S. Wu, R. Huang, and Y. Yang, "Spiking neurons with spatiotemporal dynamics and gain modulation for monolithically integrated memristive neural networks," *Nat. Commun.*, vol. 11, no. 1, pp. 1–13, 2020.
- [36] X. Li, J. Tang, Q. Zhang, B. Gao, J. J. Yang, S. Song, W. Wu, W. Zhang, P. Yao, N. Deng *et al.*, "Power-efficient neural network with artificial dendrites," *Nat. Nanotechnol.*, vol. 15, no. 9, pp. 776–782, 2020.
- [37] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.
- [38] W.-H. Chen, C. Dou, K.-X. Li, W.-Y. Lin, P.-Y. Li, J.-H. Huang, J.-H. Wang, W.-C. Wei, C.-X. Xue, Y.-C. Chiu *et al.*, "Cmos-integrated memristive non-volatile computing-in-memory for ai edge processors," *Nat. Electron.*, vol. 2, no. 9, pp. 420–428, 2019.
- [39] M. Mahmoodi, M. Prezioso, and D. Strukov, "Versatile stochastic dot product circuits based on nonvolatile memories for high performance neurocomputing and neurooptimization," *Nat. Commun.*, vol. 10, no. 1, pp. 1–10, 2019.
- [40] F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, and W. D. Lu, "A fully integrated reprogrammable memristor-cmos system for efficient multiply-accumulate operations," *Nat. Electron.*, vol. 2, no. 7, pp. 290–299, 2019.
- [41] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang *et al.*, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nat. Commun.*, vol. 9, no. 1, pp. 1–8, 2018.
- [42] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong *et al.*, "Face classification using electronic synapses," *Nat. Commun.*, vol. 8, no. 1, pp. 1–8, 2017.
- [43] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, and D. Ielmini, "Solving matrix equations in one step with cross-point resistive arrays," *Proc. National Academy Sci.*, vol. 116, no. 10, pp. 4123–4128, 2019.
- [44] Z. Sun, G. Pedretti, A. Bricalli, and D. Ielmini, "One-step regression and classification with cross-point resistive memory arrays," *Sci. Adv.*, vol. 6, no. 5, p. eaay2378, 2020.
- [45] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou, "Mixed-precision in-memory computing," *Nat. Electron.*, vol. 1, no. 4, pp. 246–253, 2018.
- [46] M. A. Zidan, Y. Jeong, J. Lee, B. Chen, S. Huang, M. J. Kushner, and W. D. Lu, "A general memristor-based partial differential equation solver," *Nat. Electron.*, vol. 1, no. 7, pp. 411–420, 2018.
- [47] G. Zhu, K. Huang, V. K. Lau, B. Xia, X. Li, and S. Zhang, "Hybrid beamforming via the kronecker decomposition for the millimeter-wave massive mimo systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2097–2114, 2017.
- [48] H. He, X. Yu, J. Zhang, S. Song, and K. B. Letaief, "Cell-free massive mimo for 6g wireless communication networks," *J. Commun. Inf. Netw.*, vol. 6, no. 4, pp. 321–335, 2021.
- [49] S. Yu and P.-Y. Chen, "Emerging memory technologies: Recent trends and prospects," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 2, pp. 43–56, 2016.
- [50] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the fast fourier transform?" *Proc. IEEE*, vol. 55, no. 10, pp. 1664–1674, 1967.
- [51] J. R. Hampton, *Introduction to MIMO communications*. Cambridge university press, 2013.
- [52] "Snapdragon x65 breaks record with download speeds exceeding 10 gbps through 5g mmwave spectrum," Qualcomm Technologies, Inc., Jun 2021. [Online]. Available: <https://www.qualcomm.com/news/onq/2021/06/snapdragon-x65-breaks-record-download-speeds-exceeding-10-gbps-through-5g-mmwave>
- [53] T. Fryza and R. Mego, "Power consumption of multicore digital signal processor: Theoretical analysis and real applications," in *IEEE Int. Symposium Ind. Electron. (ISIE)*. IEEE, 2014, pp. 1894–1898.
- [54] K.-Y. Chen, C.-S. Yang, Y.-H. Sun, C.-W. Tseng, M. Fayazi, X. He, S. Feng, Y. Yue, T. Mudge, R. Dreslinski *et al.*, "A 507 gmacs/j 256-core domain adaptive systolic-array-processor for wireless communication and linear-algebra kernels in 12nm finfet," in *IEEE Symposium VLSI Techn. Circuits*. IEEE, 2022, pp. 202–203.
- [55] S. Liu and D. Liu, "A high-flexible low-latency memory-based fft processor for 4g, wlan, and future 5g," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 3, pp. 511–523, 2018.
- [56] W. Tang, C.-H. Chen, and Z. Zhang, "A 2.4-mm² 130-mw mmse-nonbinary ldpc iterative detector decoder for 4×4 256-qam mimo in 65-nm cmos," *IEEE J. Solid-State Circuits*, vol. 54, no. 7, pp. 2070–2080, 2019.
- [57] M. Jerry, P.-Y. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu, and S. Datta, "Ferroelectric fet analog synapse for acceleration of deep neural network training," in *IEEE Int. Electron Dev. Meeting (IEDM)*. IEEE, 2017, pp. 6–2.
- [58] Z. Luo, Z. Wang, Z. Guan, C. Ma, L. Zhao, C. Liu, H. Sun, H. Wang, Y. Lin, X. Jin *et al.*, "High-precision and linear weight updates by subnanosecond pulses in ferroelectric tunnel junction for neuro-inspired computing," *Nat. Commun.*, vol. 13, no. 1, pp. 1–11, 2022.
- [59] S. Yu, "Neuro-inspired computing with emerging nonvolatile memories," *Proc. IEEE*, vol. 106, no. 2, pp. 260–285, 2018.
- [60] P.-Y. Chen, X. Peng, and S. Yu, "Neurosim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3067–3080, 2018.
- [61] P.-Y. Chen and S. Yu, "Technological benchmark of analog synaptic devices for neuro-inspired architectures," *IEEE Des. Test*, vol. 36, no. 3, pp. 31–38, 2018.
- [62] X. Sun and S. Yu, "Impact of non-ideal characteristics of resistive synaptic devices on implementing convolutional neural networks," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 9, no. 3, pp. 570–579, 2019.