2022

MR-PIPA: An Integrated Multi-level RRAM (HfOx) based Processing-In-Pixel Accelerator

Minhaz Abedin

Arman Roohi

Maximilian Liehr

Nathaniel Cady

Shaahin Angizi

MR-PIPA: An Integrated Multi-level RRAM (HfO_x) based Processing-In-Pixel Accelerator

Minhaz Abedin, *Student Member, IEEE,* Arman Roohi, *Member, IEEE,* Maximilian Liehr, *Student Member, IEEE,* Nathaniel Cady, *Member, IEEE* and Shaahin Angizi, *Member, IEEE*

***Abstract*—This work paves the way to realize a processing-in-pixel accelerator based on a multi-level HfO_x RRAM as a flexible, energy-efficient, and high-performance solution for real-time and smart image processing at edge devices. The proposed design intrinsically implements and supports a coarse-grained convolution operation in low-bit-width neural networks leveraging a novel compute-pixel with non-volatile weight storage at the sensor side. Our evaluations show that such a design can remarkably reduce the power consumption of data conversion and transmission to an off-chip processor maintaining accuracy compared with the recent in-sensor computing designs. Our proposed design, namely MR-PIPA, achieves a frame rate of 1000 and efficiency of ∼1.89 TOp/s/W, while it substantially reduces data conversion and transmission energy by ∼84% compared to a baseline at the cost of minor accuracy degradation.**

***Index Terms*—Resistive random-access memory (RRAM), processing-in-pixel, accelerator, non-volatile memory, CNN**

I. Introduction

Internet of Things (IoT) devices are expected to reach $1100B in revenue by 2025, with a web of interconnections estimated to consist of approximately 75+ billion IoT devices, including wearable devices as well as smart cities and industries [1], [2]. Artificial Intelligence of Things (AIoT) nodes are composed of a variety of sensors, which are used to collect and process data from the environment and people. There is usually a great deal of redundant and unstructured sensory data captured. The conversion and transmission of large raw data to a backend processor at the edge are energy-intensive, and high-latency [1], [3]. Those issues can be addressed by shifting computing architecture from a cloud-centric way of thinking to a thing-centric (data-centric) perspective, where IoT nodes process sensed data. Despite such challenges, artificial intelligence tasks that require hundreds of layers of Convolutional Neural Networks (CNNs) have severe computational and storage constraints. There have been considerable advancements in both software and hardware to improve CNN efficiency by mitigating the "power and memory wall" bottleneck.

This work is supported in part by the National Science Foundation under Grant No. 2216772 and 2216773.

M. Abedin, M. Liehr, and N. Cady are with the College of Nanoscale Science and Engineering, SUNY Polytechnic Institute, Albany, NY E-mail: abedin@sunypoly.edu, liehrmw@sunypoly.edu, cadyn@sunypoly.edu.

A. Roohi is with School of Computing, University of Nebraska–Lincoln, Lincoln NE, USA. E-mail: aroohi@unl.edu.

S. Angizi is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. E-mail: shaahin.angizi@njit.edu.

From the software point of view, exploration of shallower but wider CNN models, quantizing parameters, and network binarization [4] are widely accomplished. A recent development is reducing computing complexity and model size using low-bit-width weights and activations. By converting the multiplication-and-accumulate (MAC) operation into the corresponding AND-bitcount operations in [4], the authors performed bit-wise convolution between the inputs and the low-bit-width weights. Binarized convolutional neural networks (BNN), as an extreme quantization method, have achieved acceptable accuracy on both small [5] and large datasets [4] after removing some high precision requirements. By binarizing the weight and/or input feature map, they offer a promising solution to mitigate the aforementioned bottlenecks in storage and computation.

From the hardware point of view, the underlying operations should be realized using efficient mechanisms. The conventional processing elements are designed to work with a von-Neumann computing model involving separate memory and processing blocks interconnected via buses, which poses serious problems, such as long memory access latency, limited memory bandwidth, and energy-hungry data transfer, which limit the edge device's efficiency and working time [2]. Additionally, this presents several significant issues at the upper level, including bandwidth congestion and security concerns. The concept of instant image pre-processing with smart image sensors has therefore been extensively investigated [2], [6]–[8] as a potential remedy. By using an on-chip processor, the digital output from pixels can be accelerated where the sensor is located, paving the way for enhanced sensor paradigms such as Processing-Near-Sensor (PNS) as depicted in Fig. 1(b). Other promising alternatives are a Process-in-Sensor (PIS) platform [7], [9], as shown in Fig. 1(c), that processes pre-Analog-to-Digital Converter (ADC) data and a hybrid PIS-PNS [1] platform to incorporate vision sensors and eliminate redundant data output. Generally, PIS units process images before transmitting them to an on-chip processor for further processing. Typical designs rely on this type of data transfer (from CMOS image sensors to memory), which reduces the speed of feature extraction. With this PIS unit, a computation core can (i) significantly reduce the power consumption of converting photocurrents into pixel values used for image processing, (ii) accelerate data processing, and (iii) alleviate the memory bottleneck problem [1], [2].

This paper develops a new efficient Processing-in-

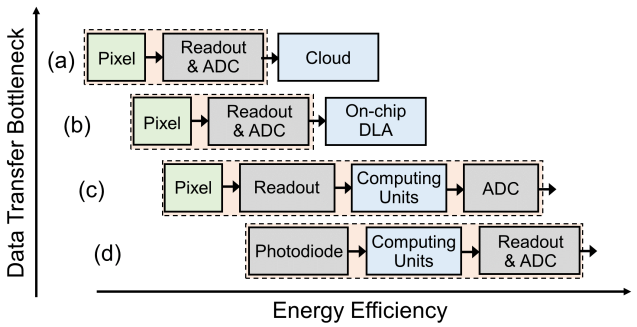

Figure 1: Visual systems with different architectures; (a) Conventional, (b) PNS, (c) PIS, and (d) PIP architectures, where green and pink (the outer box) boxes indicate the pixel and the sensors, respectively, and blue boxes represent where the computing is performed.

Pixel (PIP) paradigm, as shown in Fig. 1(d), named MR-PIPA, co-integrating always-on sensing and processing capabilities for image sensors. The main contributions of this work are as follows. (1) We experimentally demonstrate an integrated two-bit-per-cell RRAM-based weight storage unit. As low resistance states of the RRAM devices can lead to high power consumption, we run extensive device-level experiments on the fabricated device to achieve multi-level high resistive states; (2) The MR-PIPA architecture is developed based on a set of innovative microarchitectural and circuit-level schemes optimized to process the 1^{st}-layer of Quantized Neural Networks (QNN) using non-volatile RRAM components to store weights offering energy-efficiency and speed-up; (3) We present a solid bottom-up evaluation framework and a PIP assessment simulator to analyze the whole system's performance; and (4) MR-PIPA's performance and energy-efficiency are thoroughly evaluated and then compared with the recent IoT sensory platforms.

II. Background & Motivation

Systematic integration of computing and sensor arrays has been widely studied to eliminate off-chip data transmission and reduce ADC bandwidth, known as PNS [8]; combining sensor and processing elements so-called PIS [9]–[11]; and integrating pixels and computation unit, known as PIP [7], [8]. In [8], photo-currents are converted into pulse-width modulation signals, and a dedicated analog processor is used to perform feature extraction, reducing the amount of power consumed by the ADC. To run spatiotemporal image processing, 3D-stacked column-parallel ADCs, and processing elements are implemented and utilized in [2]. The CMOS image sensor with dual-mode delta-sigma ADCs described in [12] is designed to process 1^{st}-convolutional (Conv.) layer of binarized-weight neural networks (BWNN). Charge-sharing tunable capacitors are used by RedEye [13] to implement the convolution operation. By sacrificing accuracy in favor of energy savings, this design reduces energy consumption compared to a CPU/GPU. However, for high accuracy computation, the required energy per frame increases dramatically by 100×. As a PIS

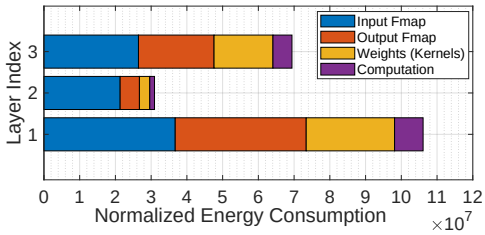

Figure 2: Energy consumption for a 3-layer MLP.

platform, MACSen [7] processes the 1^{st}-Conv. layer of BWNNs with the correlated double sampling procedure and achieves speeds of 1000fps in computation mode. This method, however, suffers from an expansive area overhead and high-power consumption. In this work, *we are motivated mainly by three observations* to develop a PIP accelerator for the first layer of QNNs. *First*, from the accuracy point of view, in most QNN accelerators, the first and the last layers of the networks remain in the full-precision, i.e., the floating-point domain. This is translated to a performance bottleneck in different hardware/software co-design accelerators and requires excessive memory and processing resources [14]. The continuous-valued inputs can be readily handled as fixed points with n bits of precision. To verify this, we utilize the deep neural network energy estimation tool developed by MIT [15] to assess the energy requirements. Figure 2 depicts the breakdown of normalized energy consumption of a 3-layer Multi-Layer Perceptron (MLP). As observed, the first layer consumes considerably higher energy than the other layers for computation (purple block) and data movement (the other three blocks). It is worth noting that this figure could be varied for different neural network architectures. *Second*, in conventional image sensors, most of the power (>96% [16]) is consumed by processing and converting pixel values. That means pixel circuits consume only 4% of power to perform photovoltaic conversions, whereas signal amplification, Digital-to-Analog Conversion (DAC), and data transmission consume most of the power. *Third*, almost all the PNS/PIS/PIP systems are hardwired, so the functionalities are limited to simple pre-processing tasks such as 1^{st}-layer BWNN computation.

III. Proposed RRAM-based Multi-bit Storage

Resistive Random Access Memory (RRAM) is a two-terminal Non-Volatile Memory (NVM) that stores data in varying resistive states by creating and rupturing a conductive filament within the metal oxide insulator, as shown in Fig. 3(a). Figure 3(b) illustrates a Transmission Electron Micrograph (TEM) of the fabricated TiN/Ti/HfO_2/TiN RRAM device integrated with CMOS n-channel Field-Effect Transistor (nFET) in 65nm CMOS technology to realize a 1T1R unit cell as a primary storage element in the proposed PIP accelerator. In the set phase, the conductive filament connects the top and bottom electrodes, leading to a Low Resistance State (LRS), whereas in the reset phase, the filament breaks, and the resistance of the device increases, yielding a

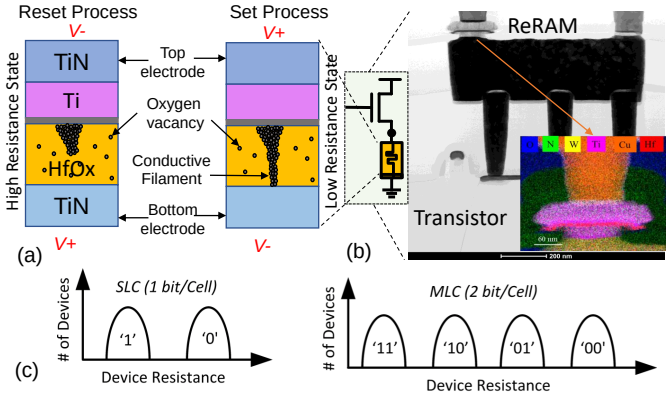

Figure 3: (a) Operating principles of RRAM device in reset and set phases, (b) 1T1R RRAM memory cell and the TEM of the fabricated RRAM device, and (c) Device resistance in single and multi-level cell (MLC).

High Resistance State (HRS), as shown in Fig. 3(a). Switching between LRS and HRS allows RRAM to operate as binary storage/memory elements. Leveraging different switching schemes enable RRAM devices to store multi-level resistance states (Fig. 3(c)) for multi-bit per cell storage [17]. The most commonly used ways to produce multi-level resistance states are modulating the compliance current at lower resistant states and the reset voltage amplitude to reach multiple high resistance states [18], [19]. The first approach results in an increased cell current due to low resistance and consequently increases overall system power consumption, while the latter results in higher HRS variability. Therefore, we propose a promising device-to-system level co-design approach to reduce overall system power consumption aiming at multiple well-defined HRS levels. Figure 4(a) shows the experimental results for switching voltage pulse widths across RRAM and gate voltages on the transistor (Fig. 3(b)). The device-level switching experiments are performed using a semi-automated Suss Microtech probe station with a high-precision semiconductor device analyzer B1500. A switching pulse width of 100ns to 1ms and a gate voltage during switching on 15 devices with 1000 cycles for each condition are considered. The median resistance values at the HRS state range from 80kΩ to 200kΩ. This approach shows much higher resistances compared to low resistance levels, ranging from 3kΩ to 30kΩ [20]. To reduce HRS variability, we adopted a read-write-verify approach to achieve resistances in a specific window, as shown in Fig. 4(b) [17]. The selected experimental resistance states will then serve as the potential memory states for MR-PIPA. We confirmed that the read-write-verify strategy employed requires a minimal amount of programming cycles. The box plots in Fig. 4(b) show that the required median programming cycles are as low as 20.

IV. MR-PIPA Architecture

We propose an energy-efficient and high-performance solution for real-time and smart image processing for

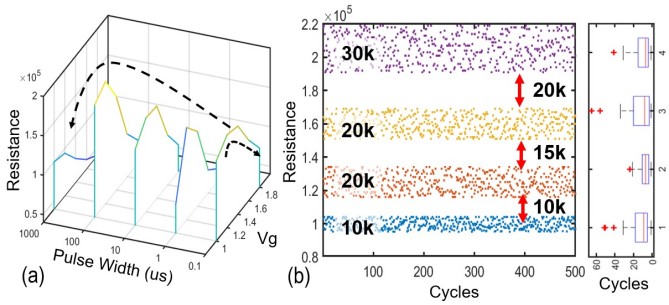

Figure 4: (a) The experimental results of the median values of HRS for pulse width and V_g, (b) Four distinguishable resistance levels programmed into 1T1R cell.

AIoT devices. MR-PIPA will integrate sensing and processing phases and can intrinsically implement a coarse-grained convolution operation required in a wide variety of image processing tasks such as classification by processing the 1^{st}-layer in QNNs. Once the object is roughly detected, MR-PIPA will switch to a typical sensing mode to capture the image for a fine-grained convolution.

A. Microarchitecture

At the architecture-level, the MR-PIPA's array consists of an $m \times n$ Compute Focal Plane (CFP), Row and Column controllers (Ctrl), command decoder, sensor timing ctrl, and sensor I/O operating in two modes, i.e., sensing and processing, as shown in Fig. 5(a). The CFP is designed to co-integrate sensing and processing of the 1^{st}-layer of QNNs targeting a low-power and coarse-grained classification. To enable this, the conventional pixel unit is upgraded to a Compute Pixel (CP). The *Ri* (Row) signal is controlled by the Row Ctrl and shared across pixels located in the same row to enable access during the row-wise sensing mode. The core part of MR-PIPA is the CP unit consisting of a pixel connected to v NVM elements, as shown in Fig. 5(b). A Sense Bit-line (SBL) is shared across pixels on the same column connected to the sensor I/O for sensing mode. Moreover, CPs share v Compute Bit-lines (CBL), each connected to a sense amplifier for processing, as indicated by the purple line in Fig. 5(a). The 1^{st}-layer weight corresponding to each pixel is pre-stored into RRAM conductance, and an efficient coarse-grained MAC operation is then accomplished in a voltage-controlled crossbar fashion. Figure 6(a) depicts a sample MLP, wherein $CP_{1,1}$-$CP_{m,n}$ are linked to out1 via NVM_1's weight. Similarly, every pixel is connected to out2-out$v$. To maximize MAC computation throughput and fully leverage MR-PIPA's parallelism, we propose a hardware mapping scheme and a connection configuration between CP elements and corresponding NVM add-ons shown in Fig. 6(b) to implement the target neural network.

B. Pixel Design

1) Basic Pixel Structure: A basic three-transistor (3T) pixel structure is depicted in Fig. 7(a) [21]. It comprises a Photodiode (PD) as the primary sensing component, a reset transistor, a source-follower transistor, and a

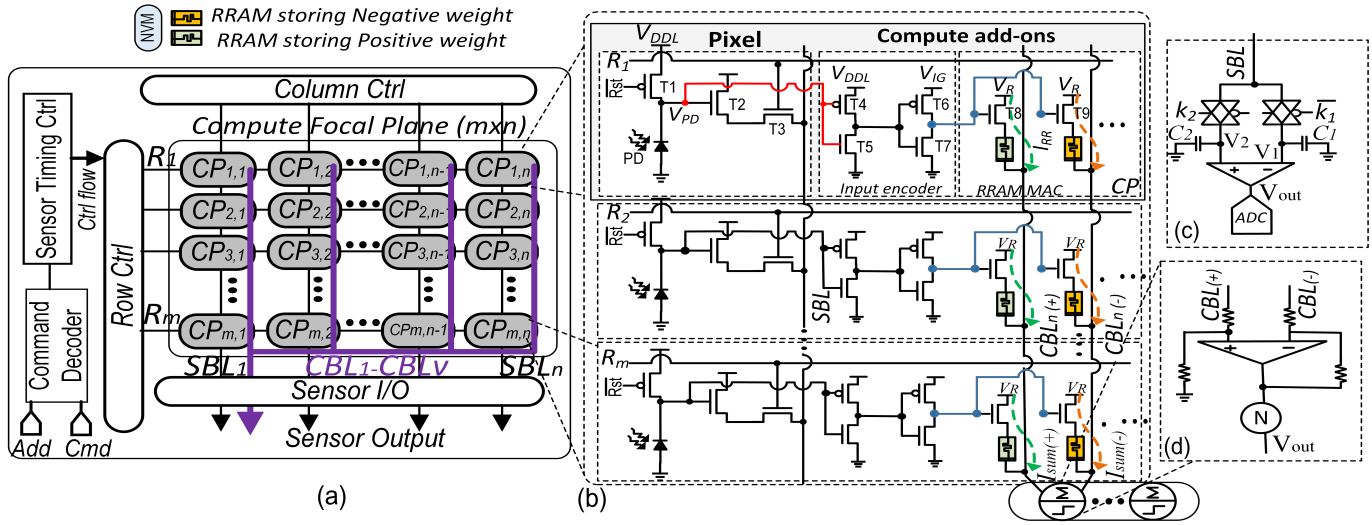

Figure 5: (a) The MR-PIPA architecture, (b) a m×1 CFP array in processing mode with compute pixels (CPs), (c) CP's read and conversion circuits in sensing mode, and (d) Differential amplifier design.

transfer transistor. PD is a semiconductor sensor which generates the photo-current (I_{PH}), proportional to the brightness of incident light or the number of photons. A simplified equivalent circuit of the PD is shown in Fig. 7(a) [22]. During exposure, the PD functions as a leaky capacitance while the leakage rate proportionally depends on the illumination [23]. The photo-current, I_{PH}, generated from PD can be calculated from the active PD area (A_{PD}), responsivity (R), and input I_{RR}adiance (E_{in}) as $I_{PH} = A_{PD} \times R \times E_{in}$. As shown in Fig. 7(b), during the bright illumination phase, the capacitor discharges faster and decreases the voltage across the PD quicker. During low illumination, I_{PH} is low, which results in a low voltage drop across the PD. The source follower (SF) operates as a voltage buffer between the sensing element PD and replicates the voltage for readout.

2) Compute add-on: The compute add-on structure depicted in Fig. 5(b) consists of two functional blocks, i) an input encoder followed by ii) 1T1R cells. The input encoder converts input from the basic pixel circuit to the input of the 1T1R cell. The 1T1R cell (part of the 1T1R array) acts as an analog multiplier unit for column-wise MAC operation. The input encoder unit consists of four transistors, of which T4-T5 are logic transistors with an operating voltage of 1.2V while T6-T7 are thick oxide 1.8V transistors. The 1T1R devices are integrated with thick gate-oxide transistors, T8 and T9. These transistors' maximum operating voltage is 3.3V, allowing them to form and program high voltages for the RRAM cells. The proposed design follows three critical considerations (**C**s) as follows. ***C1. Location of RRAM devices:*** The thin oxide transistors require a smaller area and are suitable for low power applications; as they have a low safe operating voltage, e.g., 1.2V. On the other hand, the thick oxide transistors can withstand large operating voltage, e.g., 3.3V, but suffers from higher power and area consumption. Hence, to reduce power and area, the pixel circuit is typically designed using low operating voltage thin oxide transistors. However, RRAM devices require high forming and programming voltages (∼3.3V). If the RRAM devices are connected directly across PD or pixel circuit transistors, during the forming/programming voltages will far exceed their operating voltages (Fig. 8(a)), which can damage the low voltage devices. MR-PIPA separates the pixel sensing and computing modules by transferring the signal from the pixel circuit through input encoders to the gate of the thick oxide transistors, as shown in Fig. 5(b). We then used thick oxide transistors with RRAM, allowing it to be formed or programmed at the required higher voltage. ***C2. Compute add-on output:*** The following

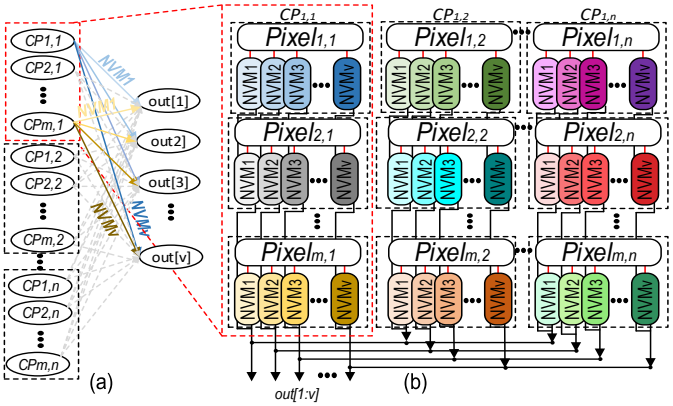

Figure 6: (a) An example of a fully-connected network with v output, (b) The mapping scheme for a m×n CFP.

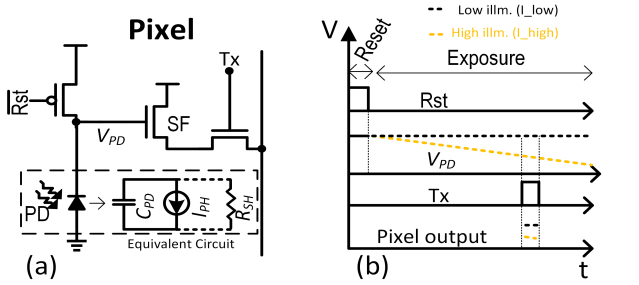

Figure 7: (a) Basic 3T pixel circuit, (b) Operation timing of the pixel.

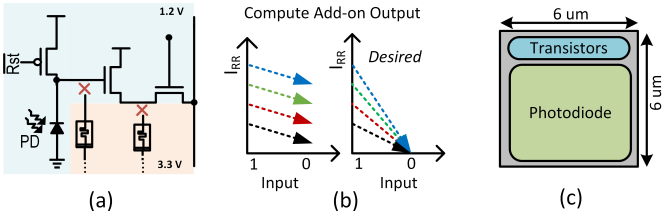

Figure 8: (a) Impractical design of an RRAM-based CP, (b) Desired output of ADD-on cell for MAC. The colors represent output currents from 4 levels of resistance states (high to low). (c) The total pixel area consisted of the PD-based sensing area and the transistors area.

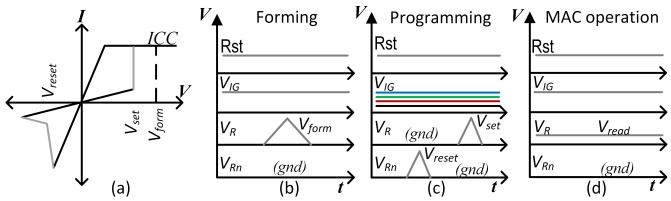

Figure 9: (a) Current vs. Voltage (I-V) for RRAM. Operation timing diagram for (b) Forming, (c) Programming, and (d) MAC (note: ICC stands for compliance current.)

subtle but critical consideration focuses on input encoding for RRAM cells, converting PD voltage to input for RRAM cells. In the standard RRAM-based matrix multiplication, for binary input x, which can be either 0 or 1, each RRAM cell current can be expressed by $I = x \cdot (V_R/R)$ [24]. Here, V_R is the applied voltage across the device, and R is the RRAM resistance of the cell. It can be realized from the given equation that for input zero, the RRAM-based compute unit should result in ideally zero current. Due to improper input encoder for the PIP circuit, the RRAM cell can result in non-zero cell current I_{RR} when the input is 0 (Fig. 8(b)). In our design, we follow the conventional RRAM-based in-memory crossbar operations for neural network inference as shown in Fig. 8(b). ***C3. Fill-factor:*** The pixel is fabricated on silicon for hardware deployment using the CMOS fabrication process. Typically for imaging applications, a larger sensing area is preferred. The ratio of PD sensing area to total pixel area is defined as the fill factor. It is optimal to increase the PD area, which results in increasing the fill factor. However, depending on the application and add-on pixel capability, such as in-pixel digital processing, the fill-factor-feature trade-off is chosen. Since the RRAM devices are fabricated at the back of the line, no large silicon area is consumed, as shown in Fig. 8(c). Although the fill factor is unaffected by RRAM, its access transistors can affect the fill factor.

C. Operational Modes

To initialize the MR-PIPA, the proposed pixel circuit requires to go through forming and programming of the RRAM devices for weight storage. The filament (Fig. 3(a)), required for resistive switching, can be formed by applying V_R=3.3V across the RRAM one-time. *Forming* can be performed by turning on transistor T1; this results in the input encoder output to be V_{IG}. As the input encoder is followed by RRAM cells, V_{IG} is applied to the gate of T8 and T9 integrated into series with RRAM (Fig. 9(b)). As for the multilevel programming, different 1T1R gate voltage is required from 1V to 1.8V (Fig. 4(a)), it can be possible with similar approach applying different V_{IG}=(1V to 1.8V) (Fig. 9(c)). As we utilize a bipolar RRAM, it requires opposite polarity voltages for set and reset operations as shown in Fig. 3(a). This can be accomplished by applying positive voltages across opposite electrodes of the RRAM as shown in Fig. 9(c).

In the *sensing mode*, initially setting *Rst*='high', the reverse biased PD is charged to V_{DDL}=1.2V (Fig. 7(a,b)) [21]. In this way, turning on the access transistor T3 and the k_1 switch at the shared ADC (Fig. 5(c)) allows the C_1 capacitor to fully charge through SBL. By turning off T1, PD generates a I_{PH} based on the external light intensity, which leads to a voltage drop (V_{PD}) at the gate of T2. Once again, by turning on T3, and this time the k_2 switch, C_2 is selected to record the voltage drop. Therefore, the voltage values before and after the image light exposure, i.e., V_1 and V_2 in Fig. 5(c), are sampled. The difference between two voltages is sensed with an amplifier, while this value is proportional to the voltage drop on V_{PD}. In other words, the voltage at the cathode of PD can be read at the pixel output.

During *object-detection mode*, we leverage the efficient crossbar MAC with 1T1R array. As RRAM cells store data as resistive states, the resultant cell current $I_{RR} = V_R/R$ when V_R is the voltage applied across the cell (see Fig. 5(b)). The voltage applied across the 1T1R cell, also known as read voltage V_R, is chosen as low as 0.2V such that it does not alter the programmed state of the device (e.g., the voltage required to set or reset the device is $\geq$0.7V). Here, the T8/T9 transistor gate voltage controls the output of the input encoder (Fig. 5(b)). If T8/T9's gate voltage is larger than the threshold voltage (0.7V), it allows the current to pass through; as a result, the cell current is $I_{RR} = V_R/R = V_R \cdot G$. Here R is one of the four resistive states representing the weighted state, and G represents the conductance of the cell. If the T8/T9 transistor gate voltage is 0, the transistor blocks the current, resulting in no cell current (I_{RR}= 0A).

As discussed previously, in high illumination, the voltage across PD, V_{PD} is low and vice versa (Fig. 7(b)). The proposed input encoder converts the V_{PD} so that output is logic '1' during low illumination (dark pixel) and logic '0' for high illumination (bright pixel). The first inverter (T4, T5) of the input encoder operates at 1.2V and converts the V_{PD} to 0 or 1.2V output for the second inverter. The second inverter consists of thick oxide 1.8V-transistors (T6, T7) which allow the 0-1.8V gate voltage for multi-level programming (Fig. 9(c)). As the threshold voltage of T6 and T7 transistors is below 0.7V, the output of second inverter results in (V_{IG}, 0V) (Fig. 5(b)). The resultant output of the input encoder is V_{IG} and 0V for low/dark illumination and high/bright illumination, respectively. Accordingly, the resultant cell's current for low and high illumination are $I_{RR} = V_R/R$ and 0, respec-

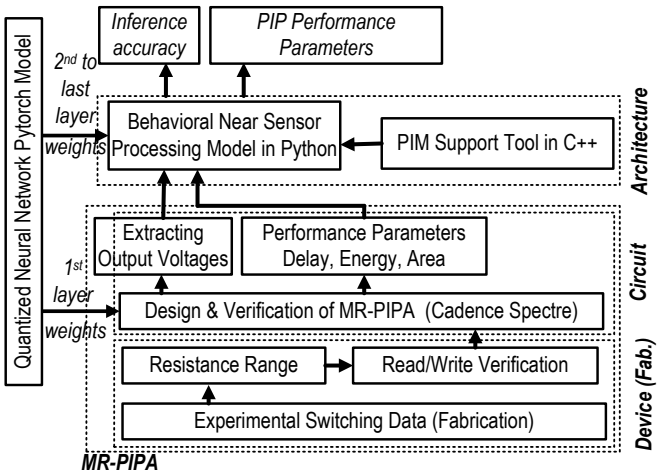

Figure 10: Evaluation framework.

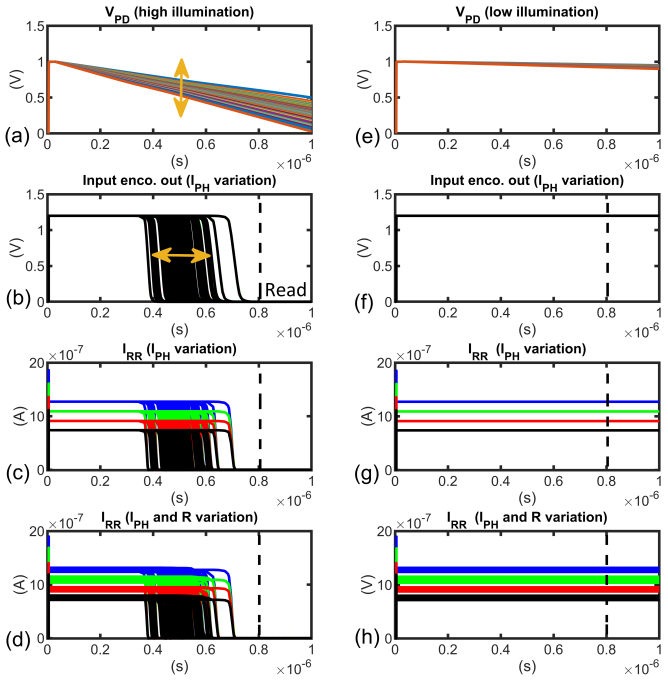

Figure 11: Proposed circuit response for high and low input illumination showing (a) voltage across PD, (b) input encoder output, (c) RRAM CP's current with only I_{PH} variation and no RRAM variation, and (d) RRAM CP's current with both I_{PH} and RRAM variation. Panels (e-h) show the circuit responses at low illumination.

tively. Then, to combine and quantify the currents from both positive and negative weight connections, we constructed a differential amplifier (Fig. 5(d)). Input currents into the operational amplifier in each column pair consist of two columns of the positive and the negative weights (Fig. 5(a)). Each column current is the summation current from each 1T1R cells, e.g., the positive weight current for $j-th$ column can be described as $\sum_{i=1}^{M} V_R \cdot G_{i,j}^{+}$. The resultant output voltage of the operational amplifier will be proportional to $\sum_{i=1}^{M}\left((V_R)\cdot(G_{i,j}^{+}-G_{i,j}^{-})\right)$ where $G_{i,j}^{+}, G_{i,j}^{-}$ is the conductance of the RRAM cell indexed by i and j storing the positive and negative weights, respectively. From a programmer's standpoint, MR-PIPA is a third-party accelerator rather than a memory unit. Thus, for general-purpose parallel execution, an ISA and virtual machine will be needed. With this, any user-level program can be translated at install time to the MR-PIPA's hardware instruction set to support MAC.

V. Performance Evaluation

A. Framework & Methodology

To assess the performance of the proposed design, we developed a simulation framework from scratch consisting of three main components as shown in Fig. 10. First, at the device-level, we fabricated the proposed RRAM device and extracted the switching data, and resistance ranges experimentally. Second, at the circuit-level, we fully implemented MR-PIPA with peripheral circuity with IBM 65nm CMOS10LPe PDK in Cadence to achieve the performance parameters. We trained a PyTorch QNN model inspired by [4] extracting the 1^{st}-layer weights. MR-PIPA's RRAM elements are then programmed at the circuit-level by the quantized 2-bit weights. Third, after the 1^{st}-layer computation, the results are recorded and fed into a behavioral-level in-house simulator to simulate the whole network at the architecture-level and extract the performance parameters and inference accuracy.

B. Device-to-circuit Level results

The proposed CP was designed at a 65nm process node. The pixel's PD was simulated as a parallel capacitor, and the photo-current represented the illumination. The capacitance value (13 fF) was calculated from the doping concentration of the 65nm CMOS process and the PD area (section IV B). To demonstrate the lowest case for high illumination (/bright pixel) was considered as ∼13k lux and the highest case for low illumination/dark pixel was considered as ∼130 lux. The resultant I_{PH}s used for the simulations are 10nA and 0.1nA for high and low illumination, respectively.

We simulated both high and low illumination with 10% variation and observed the response at different points of the circuit. First, the voltage response across the PD shows expected high voltage drop and low voltage drop over time respectively (Fig. 11 (a,e)). It also confirms that the add-on compute unit does not affect the pixel sensing operation. Figures 11 (b,f) show the input encoder output. As the proposed input encoders are inverters, the inverters tend to switch to rail voltages 0V and 1.2V during MAC operation. The switching from 1.2V to 0V occurs before the 'read' operation (at 0.8×10^{-6}s). We observe that the proposed design is immune to I_{PH}'s variation as any V_{PD} during high illumination and low illumination are converted as rail to rail 0V and 1.2V (Fig. 11 (f)). As the input encoder output acts as an input for the thick oxide transistor integrated with RRAM, the 0V and 1.2V voltages fall below and above the transistor's threshold voltage of 0.7V. As a result, no current flows through the RRAM cell during 0V input encoder output or during high illumination. On the other hand, during low illumination,

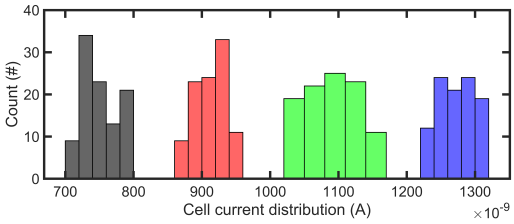

Figure 12: The histogram shows the in-pixel compute cell current. Blue, green, red, and black represent the cell current from the compute cell with RRAM resistance levels 1, 2, 3, and 4 (low to high), respectively.

the input encoder output becomes 1.2V, and the cell output current according to Ohm's law is $I_{RR} = V_R/R$. It is noteworthy that the RRAM cell output current (Fig. 11(b,f)) is independent of I_{PH}'s variation. The immunity to I_{PH}'s variation is a result of using inverters for input encoding. As for an analog voltage range above and below $V_{DDL}/2$, the output of an inverter is 0V or V_{DDL} respectively. Figure 11(d,h) show that when I_{PH} and RRAM resistance variation are present, the output RRAM cell current is only dependent on the RRAM resistance variation. The RRAM cell current for four different resistance levels shown in Fig. 12. Even with variations considered, the cell currents are distinguishable for different resistance/weight stored.

C. Circuit-to-architecture Level results

We limited the weight precision to four resistance levels. This can be readily used to map and accelerate binary, ternary, and quaternary neural networks. Table I compares the literature's structural and performance parameters of selective processing-in-pixel and sensor designs. As different designs are developed for specific domains, for an impartial comparison, we estimated and normalized the power consumption when all units executed the similar task of processing the 1^{st}-layer of CNN. Our cross-layer simulation results show that the MR-PIPA achieves a frame rate of 1000. This comes from the massively-parallel CPs. However, the design in [6] achieves the highest frame rate, and the design in [2] imposes the least pixel size enabling in-sensor computing. As for the area, our simulation results reported in Table I show the proposed MR-PIPA's compute-pixel occupies $\sim 6\times 6\mu m^2$ in 65nm. As we do not have access to the other layouts' configurations, it is almost impossible to have a fair comparison between area overheads. However, we believe a rough assessment can be made by comparing the number of transistors in previous SRAM-based designs and MR-PIPA's lower-overhead compute add-on. We re-implemented MACSen [7] at the circuit-level as the only CNN accelerator developed with the same purpose. Our evaluation showed that MR-PIPA consumes $\sim$74% less power consumption compared with MACSen performing the same task. Compared to [6], MR-PIPA substantially reduces data conversion and transmission energy by $\sim$84%. While Table I focuses on various PIS architectures (close-to-pixel computation) primarily supporting CNNs in the binary domain, recent architectures show a systolic neural CPU fusing the operation of a traditional CPU and a systolic CNN accelerator [26]. Compared with our work, the design in [26] shows a systolic neural CPU fusing the operation of a traditional CPU and a systolic CNN accelerator. It converts 10 CPU cores into an 8-bit systolic CNN accelerator showing a comparable performance (1.82 TOPS/W @65nm vs. 1.89 TOPS/W @65nm in MR-PIPA) but provides higher flexibility and bit-width (up to 8-bit). Putting everything together, MR-PIPA offers: 1) a low-overhead, dual-mode and reconfigurable design to keep the sensing performance and realize a processing mode to remarkably reduce the power consumption of data conversion and transmission; 2) single-cycle in-sensor processing mechanism to improve image processing speed; 3) highly parallel in-sensor processing design to achieve ultra-high-throughput; 4) exploiting NVM reduces standby power consumption during idle time and offers instant wake-up time and resilience to power failure to achieve high performance.

D. Accuracy

An image classification task is selected to demonstrate the benefits of MR-PIPA design. In the original BWNN topology, all the layers, except the first and last, were implemented with quantized weights [27]. However, in these tasks, the number of input channels is relatively lower than the number of internal layers' channels, so the required parameters and computations are small, and converting the input layer will not be a significant issue [27]. Therefore, in almost all previously developed 3T and 4T -pixel PIP designs, the first layer is implemented with quantized weights, realizing BWNN [7]. Then an identical NN accelerator can be used to accelerate the remaining layers after the first layer has been computed.

Datasets: We conducted experiments on several datasets, including MNIST [28], Fashion-MNIST [29], MCFD [30] and SVHN [31]. MNIST is leveraged as a gray-scale dataset that contains 70,000 28×28 images of handwritten digits from 0 to 9, 60,000 images for training, and 10,000 images for testing sets. Similar to MNIST, Fashion-MNIST consists of 28×28 gray-scale images but includes 10,000 images for each training and testing set to form ten fashion categories. MCFD face recognition database contains face images of 10 subjects, where each image is normalized to 20×20 pixels. Training data consists of 6,977 images, while testing data consists of 24,045 images. Finally, we also exploit SVHN with 73,257 training digits, 26,032 testing digits, and 531,131 additional digits for extra training data. The images are pre-processed to 20×20 from the original 32×32 cropped version and fed to the model.

NN Architecture: In order to evaluate our design and perform a fair comparison, we developed two networks, including a 2-layer MLP and a CNN with 3 convolutional and 3 FC layers, which are equivalently imple-

Table I: Performance comparison of various PIP and PIS units.

Designs	Technology (nm)	Purpose	Comput. Scheme	Memory	NV*	Pixel Size (μm^2)	Array Size	Frame Rate ($frame/s$)	Power (mW)	Efficiency ($TOp/s/W$)
[25]	180	2D optic flow est.	raw-wise	Yes	No	28.8×28.8	64×64	30	0.029	0.0041
[8]	180	edge*/blur/sharpen/ 1st layer DNN	raw-wise	No	No	7.6×7.6	128×128	480	sensing: 0.077 processing: 0.091	0.777
[2]	60/90	STP†	raw-wise	Yes	No	3.5×3.5	1296×976	1000	sensing: 230 processing:363	0.386
[7]	180	1st layer BNN	entire-array	No	No	110×110	32×32	1000	0.0121	1.32
[6]	180	edge*/TMF‡	raw-wise	Yes	No	32.6×32.6	256×256	100,000	1230	0.535
MR-PIPA	65	1st layer QNN	entire-array	Yes	Yes	6×6	256×256	1000	sensing:0.021 processing:0.088	1.89

* Edge extraction. †Spatial Temporal Processing. ‡Thresholding Median Filter.

Table II: Classification accuracy (%).

Configuration	MNIST	FashionMNIST	MCFD	SVHN
BWNN [27]	98.6	90.02	–	97.47
PIP [7]	96.0	83.17	90.67	–
PISA [32]	95.12	–	–	90.35
MR-PIPA	97.26	85.68	92.30	91.05

*BWNN is a software implementation and is considered a baseline. While the PIP and PISA designs are hardware-based techniques.

mented by convolutional layers. Herein, the 1^{st}-layer is performed at the device-level, and its outputs are then fed into the second layer of the algorithm, which is implemented in Python. The comparison of classification accuracy is summarized in Table II. The results show that higher accuracy can be achieved using our MR-PIPA architecture, which can handle four analog values (2-bit quantized) rather than two (1-bit).

VI. Conclusion

This work presents a PIP accelerator that intrinsically implements and supports a coarse-grained convolution operation in low-bit-width quantized neural networks leveraging a novel compute-pixel with non-volatile weight storage at the sensor side. We demonstrate four distinct high resistance levels in order to decrease overall system power consumption. Our results demonstrate acceptable accuracy on various data sets, while MR-PIPA achieves the frame rate of 1000 and the efficiency of ∼1.89 TOp/s/W.

References

[1] T.-H. Hsu *et al.*, "Ai edge devices using computing-in-memory and processing-in-sensor: from system to device," in *IEDM*. IEEE, 2019, pp. 22–5.

[2] T. Yamazaki *et al.*, "4.9 a 1ms high-speed vision chip with 3d-stacked 140gops column-parallel pes for spatio-temporal image processing," in *ISSCC*. IEEE, 2017, pp. 82–83.

[3] J. H. Ko *et al.*, "A single-chip image sensor node with energy harvesting from a cmos pixel array," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 9, pp. 2295–2307, 2017.

[4] S. Zhou *et al.*, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv:1606.06160*, 2016.

[5] C. Matthieu et al., "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[6] S. J. Carey *et al.*, "A 100,000 fps vision sensor with embedded 535gops/w 256× 256 simd processor array," in *2013 Symposium on VLSI Circuits*. IEEE, 2013, pp. C182–C183.

[7] H. Xu *et al.*, "Macsen: A processing-in-sensor architecture integrating mac operations into image sensor for ultra-low-power bnn-based intelligent visual perception," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 2, pp. 627–631, 2020.

[8] T.-H. Hsu *et al.*, "A 0.5-v real-time computational cmos image sensor with programmable kernel for feature extraction," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 5, pp. 1588–1596, 2020.

[9] H. Xu *et al.*, "Senputing: An ultra-low-power always-on vision perception chip featuring the deep fusion of sensing and computing," *IEEE TCASI: Regular Papers*, 2021.

[10] M. Abedin *et al.*, "A processing-in-pixel accelerator based on multi-level hfox reram," in *CASES*. IEEE, 2022.

[11] S. Angizi *et al.*, "Integrated sensing and computing using energy-efficient magnetic synapses," in *ISQED*. IEEE, 2022, pp. 1–4.

[12] W.-T. Kim *et al.*, "An on-chip binary-weight convolution cmos image sensor for neural networks," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7567–7576, 2020.

[13] R. LiKamWa *et al.*, "Redeye: analog convnet image sensor architecture for continuous mobile vision," *ACM SIGARCH Computer Architecture News*, vol. 44, pp. 255–266, 2016.

[14] F. Muñoz-Martínez *et al.*, "Stonne: Enabling cycle-level microarchitectural simulation for dnn inference accelerators," in *IISWC*. IEEE, 2021, pp. 201–213.

[15] T.-J. Yang *et al.*, "A method to estimate the energy consumption of deep neural networks," in *2017 51st asilomar conference on signals, systems, and computers*. IEEE, 2017, pp. 1916–1920.

[16] J. Choi *et al.*, "An energy/illumination-adaptive cmos image sensor with reconfigurable modes of operations," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 6, pp. 1438–1450, 2015.

[17] B. Q. Le *et al.*, "Resistive ram with multiple bits per cell: Array-level demonstration of 3 bits per cell," *IEEE Transactions on Electron Devices*, vol. 66, pp. 641–646, 2019.

[18] F. Zahoor *et al.*, "Resistive random access memory (rram): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications," *Nanoscale Research Letters*, vol. 15, p. 90, 12 2020.

[19] Z. Zhang *et al.*, "Memory materials and devices: From concept to application," *InfoMat*, vol. 2, pp. 261–290, 3 2020.

[20] M. Liehr *et al.*, "Impact of switching variability of 65nm cmos integrated hafnium dioxide-based reram devices on distinct level operations," in *IIRW*. IEEE, 2020, pp. 1–4.

[21] S. Henker *et al.*, "Active pixel sensor arrays in 90/65nm {CMOS}-technologies with vertically stacked photodiodes," *Proc. IEEE International Image Sensor Workshop IIS07*, pp. 16–19, 2007.

[22] W. Hernandez, "Input-output transfer function analysis of a photometer circuit based on an operational amplifier," *Sensors*, vol. 8, 2008.

[23] J. Wilson *et al.*, "Optoelectronics: An introduction," *American Journal of Physics*, vol. 52, pp. 479–479, 5 1984.

[24] A. Sebastian *et al.*, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, pp. 529–544, 7 2020.

[25] S. Park *et al.*, "7.2 243.3 pj/pixel bio-inspired time-stamp-based 2d optic flow sensor for artificial compound eyes," in *ISSCC*. IEEE, 2014, pp. 126–127.

[26] Y. Ju and J. Gu, "A 65nm systolic neural cpu processor for combined deep learning and general-purpose computing with 95% pe utilization, high data locality and enhanced end-to-end performance," in *ISSCC*, vol. 65. IEEE, 2022.

[27] I. Hubara *et al.*, "Binarized neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[28] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, 1998.

[29] H. Xiao *et al.*, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[30] C. for Biological, C. L. at MIT, and MIT. (2000) Cbcl face fatabase. [Online]. Available: http://cbcl.mit.edu/software-datasets/FaceData2.html

[31] Y. Netzer *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS*, 2011.

[32] S. Angizi *et al.*, "Pisa: A binary-weight processing-in-sensor accelerator for edge image processing," *arXiv preprint arXiv:2202.09035*, 2022.