TYPE Original Research
PUBLISHED 30 April 2025
DOI 10.3389/felec.2025.1567562

SPIKA: an energy-efficient time-domain hybrid CMOS-RRAM compute-in-memory macro

Khaled Humood*, Yihan Pan, Grahame Reynolds, Mohammed Mughal, Shiwei Wang, Alexander Serb and Themis Prodromakis

The Centre for Electronics Frontiers, Institute for Integrated Micro and Nano Systems, School of Engineering, University of Edinburgh, Edinburgh, United Kingdom

Check for updates

OPEN ACCESS

EDITED BY
Sumeet Kumar Gupta,
Purdue University, United States

REVIEWED BY
Abhronil Sengupta,
The Pennsylvania State University (PSU),
United States
Jeffry Victor,
Purdue University, United States

*CORRESPONDENCE
Khaled Humood,
✉ K.M.M.Humood@sms.ed.ac.uk

RECEIVED 27 January 2025
ACCEPTED 15 April 2025
PUBLISHED 30 April 2025

CITATION
Humood K, Pan Y, Reynolds G, Mughal M, Wang S, Serb A and Prodromakis T (2025) SPIKA: an energy-efficient time-domain hybrid CMOS-RRAM compute-in-memory macro. *Front. Electron.* 6:1567562. doi: 10.3389/felec.2025.1567562

The increasing significance of machine learning (ML) has led to the development of circuit architectures suited to handling its multiply-accumulate-heavy computational load such as Compute-In-Memory (CIM). A big class of such architectures uses resistive RAM (RRAM) devices, typically in the role of neural weights, to save power and area. In this work, we introduce SPIKA, a novel RRAM-based ML accelerator implemented in 180nm CMOS technology. The design features a 64×128 crossbar array, supports 4-bit inputs, ternary weights, and 5-bit outputs. Post-layout analysis indicates a remarkable performance of the proposed system compared to state-of-the-art with a peak throughput of 1092 GOPS and energy efficiency of 195 TOPS/W. The key innovation of SPIKA lies in its natural signal domain crossing, which eliminates the need for power-hungry data converters. Specifically, digital input signals are converted to pulse-width modulated (time-domain), then applied on the RRAM weights that convert them to analog currents, and then aggregated into digital values using a simple switch capacitor read-out system.

KEYWORDS

in-memory-computing, RRAM, IMC, CIM, accelerators, VMM, analog-computing

1 Introduction

The deployment of neural networks (NNs) in machine learning (ML) applications such as computer vision, speech recognition and natural language processing has grown exponentially in the past few decades (Hertel et al., 2015; Graves et al., 2013; Bahdanau et al., 2015; Humood et al., 2023b). The biggest challenge in implementing such algorithms is the constant data movement between the compute units and memory units (Yu et al., 2021). Today's computing systems, primarily built based on the von Neumann architecture where data must be moved to a processing unit, have shown inefficiency in implementing ML algorithms (Amirsoleimani et al., 2020). The latency and energy associated with this bottleneck present a key performance concern for a range of applications in artificial intelligence (AI) workloads. For example, the cost of multiplying two numbers is orders of magnitude lower than accessing them from the memory at 45 nm CMOS technology (Sebastian et al., 2020). Another key challenge is that NNs carry out copious calculations of Multiply and Accumulate (MAC) operations which require high-performance GPUs, consuming a great amount of power. Thus innovation in computing architectures is expected to play a major role in the future of ML hardware.

Recently non-volatile compute-in-memory (nvCIM) technology has shown prominent results in solving the data movement and MAC operation bottleneck of ML algorithms and running parallel analog vector matrix multiplication (VMM) operations in memory arrays. This is achieved by configuring the physical characteristics of the memory devices, the array level organizations, the peripheral circuitry and the control logic (Yu et al., 2021; Sebastian et al., 2020). RRAM-based VMM engines in particular have attracted considerable attention by directly using Ohm's law for multiplication and Kirchhoff's law for accumulation, an RRAM array is capable of implementing parallel in-memory MAC operations with greatly improved throughput and energy efficiency over digital computing approaches (Mittal, 2019; Amirsoleimani et al., 2020; Yu et al., 2021). For example, the ISAAC structure (baseline nv-CIM) has demonstrated a 14.8× increase in throughput and a 5.5× improvement in energy efficiency compared to DaDianNao (Shafiee et al., 2016).

Previous works on RRAM-based nvCIM architectures (Musisi-Nkambwe et al., 2021; Mittal, 2019; Yao et al., 2020; Marinella et al., 2018; Cai et al., 2019; Bayat et al., 2018; Sahay et al., 2020; Liu et al., 2020; Mochida et al., 2018; Shafiee et al., 2016; Xue et al., 2020; Prezioso et al., 2018; Ankit et al., 2017; Wang et al., 2015; Narayanan et al., 2017; Kadetotad et al., 2015; Hung et al., 2021; Tang et al., 2017; Li et al., 2015; Ming et al., 2017; Chen et al., 2019; Su et al., 2017; Xia et al., 2016; Chi et al., 2016; Li et al., 2021; Khaddam-Aljameh et al., 2022; Wan et al., 2022; Jiang et al., 2023) have been proposed to accelerate NN. These works can primarily be divided into 2 approaches: Current-domain (CD) designs and Time-domain (TD) designs. The majority of the reported VMM engines are CD approaches (31, 29, 50, 6?, 24, 30, 38, 47, 33, 3, 44, 32, 20, 42, 22, 28, 8, 41, 45, 9) in which the inputs of the neural network are mapped as voltages with different amplitudes using digital to analog converts (DACs) and are applied at the row of the RRAM crossbar. After that, the current passing in each bit-line is converted to a digital value using general-purpose analog-to-digital converters (ADCs) such as SAR ADC, Flash ADC or sigma delta ADC. In such approaches, the precision of the VMM engine and the array size is very limited as higher precision requires power-hungry and expensive DACs and ADCs at each row/column. For example, the ADCs in ISAAC structure account for 58% of total power and 31% of total Area (Shafiee et al., 2016).

Nevertheless, a recent approach proposes a time-domain encoding scheme where the inputs of the neural network are time-modulated by applying fixed amplitude voltage pulses but with varying duration, then, the charge in the bit-line is integrated by a switched capacitor and converted to digital using ramp ADCs or current integrators, such an approach is referred as TD approach (Amirsoleimani et al., 2020). Reported TD designs (Marinella et al., 2018; Cai et al., 2019; Sahay et al., 2020; Hung et al., 2021; Alemdar et al., 2017) have demonstrated great potential in reducing the cost of area and energy consumption of encoding the inputs and overcoming the I-V non-linearity of RRAM devices which affects the output accuracy significantly (Amirsoleimani et al., 2020). However, current approaches include complex output circuity including high-resolution ramp ADC and current integrators (Marinella et al., 2018) or high-resolution accumulators (Sahay et al., 2020) which limits the area and energy efficiencies of these approaches.

In this work, we present SPIKA, an energy-efficient TD RRAM-nvCIM macro designed for accelerating inference tasks. SPIKA includes a passive modified 1T1R crossbar and is fully integrated with all the necessary interface and communication circuitry using a commercial 180 nm process. SPIKA has a crossbar size of 64 × 128 and supports a 4-bit/ternary/5-bit (input/weight/output) resolution. Every 2 columns share one output, where one column holds the positive weights and the one column holds the negative weights, thus, a total of 64 outputs per core. Two novel circuit techniques are implemented in SPIKA to significantly improve energy and throughput efficiency. First, the *efficient conversion of the input signal to the output signal* featuring 3 domain crossings: digital-time, time-analog and analog-digital using minimum area and energy overhead. This is achieved by introducing the "clicking mechanism" in which during the VMM process, the column capacitor voltage resets every time it reaches a threshold voltage and the number of clicks/resets per column represents the digital output that is tracked by a digital counter. In contrast to the switched capacitor design by Sahay et al. (2020), the clicking mechanism in SPIKA effectively reduces the size of the switched capacitor in each column and simplifies the readout circuitry, thereby improving both density and power consumption.

The second technique is *the negative weight representation*. The majority of prior nv-CIM approaches that support real weight representation position positive and negative weights in adjacent columns/rows, where the contribution of negative weights is subtracted from positive weights after the analog to digital conversion leading to additional subtracting units at each output which adds more energy and area overhead to the design (Chen et al., 2019; Su et al., 2017; Yu et al., 2016; Xia et al., 2016; Chi et al., 2016). Other works use differential input encoding, in which the row drivers send input voltage pulses with different polarities for positive and negative weights. This would require two different drivers, one for positive weights and one for negative weights which decrease the power efficiency of the system (Wan et al., 2022). On the other hand, SPIKA places positive and negative weights in adjacent columns but applies the same pulse voltage (both in magnitude and polarity). Each pair of columns shares a single counter (output circuit), and the subtraction of the positive and negative weights occurs inside the counter naturally.

Ultimately, the key linchpin of SPIKA is that it leverages the low-resolution niche it addresses to allow each domain to play to its strengths (time-domain for fixed-voltage, multi-level input encoding, analog for power and space efficient computation and digital for reliable communications) whilst using simple and efficient domain converters. This makes for a highly functional and simultaneously energy and area-efficient implementation. Circuit simulations reveal that the SPIKA core, operating on a 180 nm process, achieves a peak normalized throughput of 1092 GOPS and an energy efficiency of 195 TOPS/W, competing with prior works running on more advanced technology nodes. A detailed comparison between SPIKA and previous designs is provided in Table 6.

The remainder of this paper is organized as follows: Section 2 provides a system overview of the SPIKA core architecture. Section 3 describes the methodology of the clicking mechanism and the transition of the signal from input to output along with design constraints and design parameters. Post-layout simulation results and performance evaluation are provided in Section 4. Section 5

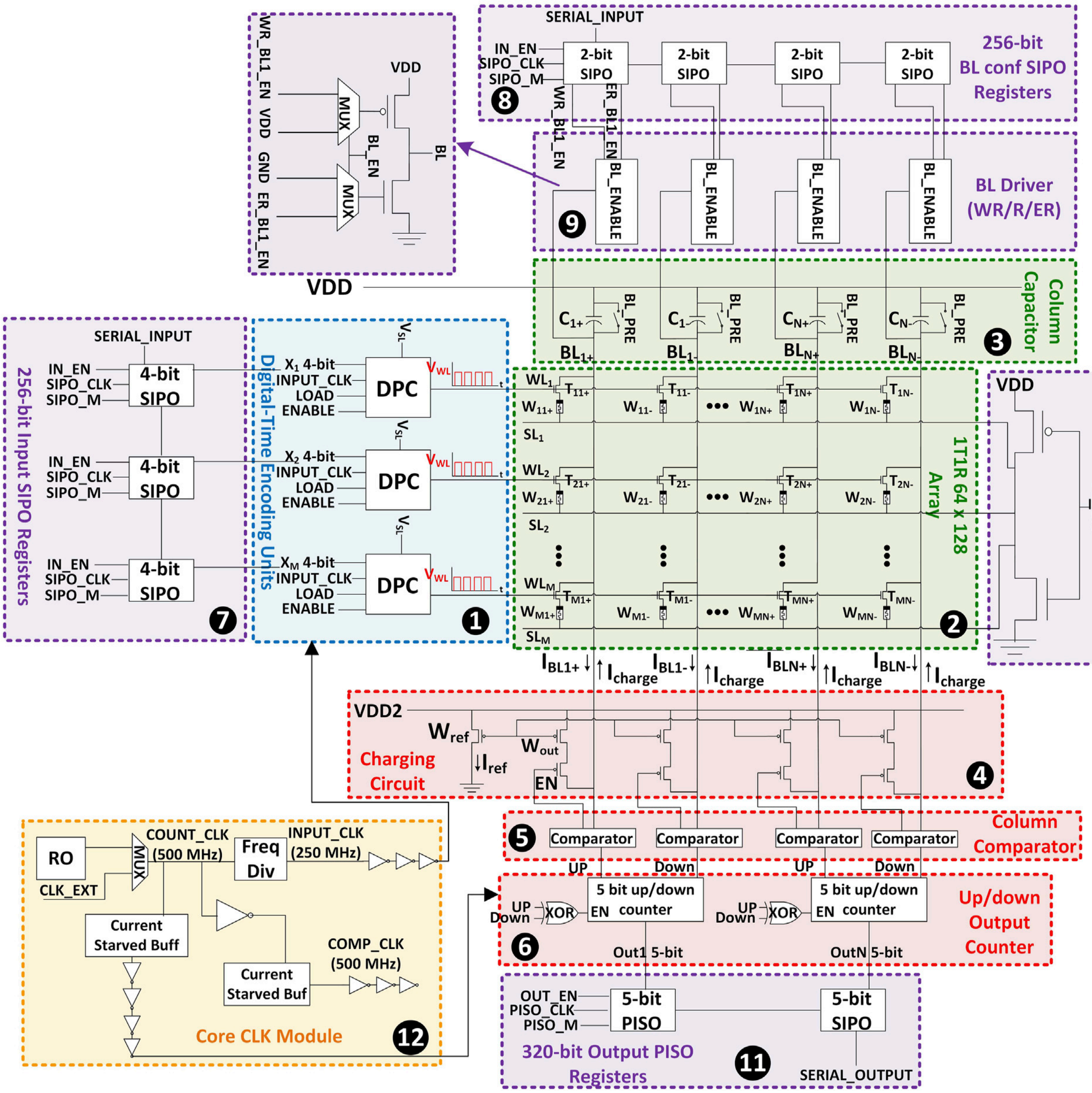


FIGURE 1
SPIKA macro architecture showing the input circuit (blue dashed rectangle), 1T1R array (green dashed rectangle, output circuit (red dashed rectangle), clock module (yellow dashed rectangle) and I/O interface and driving (purple dashed rectangle).

provides circuit level bench-marking and comparisons with the state of the art, followed by conclusions in Section 6.

2 System overview

The SPIKA circuit architecture is summarized in Figure 1. The SPIKA core can be divided into 5 parts, namely, the input circuit, the 1T1R crossbar array and column capacitors, the output circuit, the I/O interface and driving units and the clock module. SPIKA is a fully integrated core with all the necessary interface and communication circuitry using a commercial 180 nm process. RRAM cells designed in this work are based on an experimental model developed by our research group (Maheshwari et al., 2021a; Maheshwari et al., 2021b). SPIKA has a crossbar size of 64 × 128 and supports a 4-bit input resolution (per row), ternary weights and 5-bit output resolution (per column). All system components except the charging circuit are powered with a 1.8 V power supply rail (VDD). In this section, we provide a description of the system components and circuits, along with their respective functionalities. More detailed information about the system operations is discussed in Section 3.

2.1 Input circuit

Here, row-wise digital-pulse-converter (DPC) units (label 1, Figure 1) convert incoming 4-bit digital inputs to a

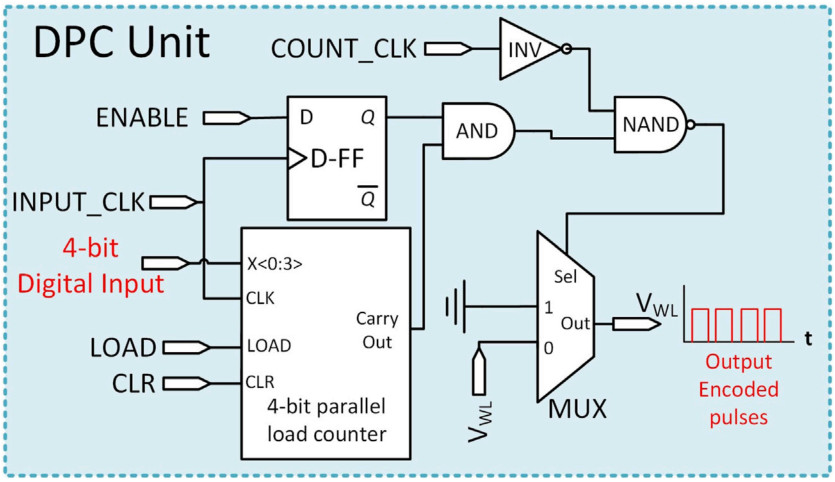


FIGURE 2
Digital-pulse-converter (DPC) circuit schematic.

corresponding number of regularly-spaced voltage pulses at fixed-level V_{WL}, as set by an external power supply. The circuit schematic of one DPC unit is shown in Figure 2. The counter utilized in the DPC unit is a parallel load counter, capable of initiating any desired counting sequence once the inputs are loaded (via the LOAD enable signal) (Humood et al., 2023a). The D-FF used in the DPC unit is based on the true single-phase-clock FF (TSPC-FF) (Ji-Ren et al., 1987). The number of output pulses is determined by the value of the digital input. The duration of the pulse is set by the input clock ($INPUT\_CLK$). V_{SL} once designed is fixed and it is applied to the word-lines (WLs) of the 1T1R array and thus controls the voltage drop on the RRAM device. This configuration allows the design to utilize a low-power row driver instead of a high-resolution DAC circuit. For more information on DPC circuit used in this work, please refer to our previous study (Humood et al., 2024).

2.2 1T1R array and column capacitors

The weights of the NN are mapped as resistances in the RRAM cells. In SPIKA, we use a ternary encoding scheme for the weights where positive and negative weights are stored in adjacent columns. The high resistive state (HRS) represents weight 0 and the low resistive state (LRS) represents weight 1 on positive columns and −1 on negative columns. Note that while the weight range is ternary, the RRAM devices are only required to be able to assume 2× states in total; a deliberately very loose requirement intended to lower the entry bar for various RRAM technologies being developed around the world. The LRS is selected to be 40 KΩ and the HRS is selected to be 3 MΩ.

A modified structure of the conventional 1T1R crossbar (Humood et al., 2019) is configured in SPIKA (label 2, Figure 1), this allows the crossbar to act as a current sink to the column capacitors (label 3, Figure 1). WLs are shared across the row and connect to the gate terminal of the access transistor in the 1T1R cell. Source-lines (SLs) are connected to a driver that determines the memristor (MR) mode of operation (writing/reading/erasing). Bit-lines (BLs) are shared across the columns and connect the column capacitors with the drain of the access transistors and the output circuits. Table 1 provides a summary of the array mode of operation in relation to the WL/BL and SL voltages. In addition, the table includes the voltage across the RRAM device in each mode and the peak current passing through the device.

2.3 Output circuit

2.3.1 Charging circuit

The charging circuit in SPIKA (label 4, Figure 1) is used to charge the column capacitors (C_{1-N}) after a clicking operation. The charging circuit consists of a current mirror supplying constant currents to all the BL branches of the 1T1R array. The sizing of the transistors W_{ref}, W_{out} and EN were designed to supply a charging rate of 0.6 V/ns under a bias of I_{ref} = 0.35 μA.

2.3.2 Comparator

A low-power double-tail dynamic comparator presented by Babayan-Mashhadi and Lotfi (2014) is designed in SPIKA (label 5, Figure 1). The comparator schematic is shown in Figure 3A. The sizing of the comparator was optimized for area, power, offset and decision time. Figure 3B summarizes the comparator performance generated by post-layout and Monte-Carlo simulations. The comparator in SPIKA tracks the BL voltage and issues requests for clicking once the capacitor voltage falls below the threshold value.

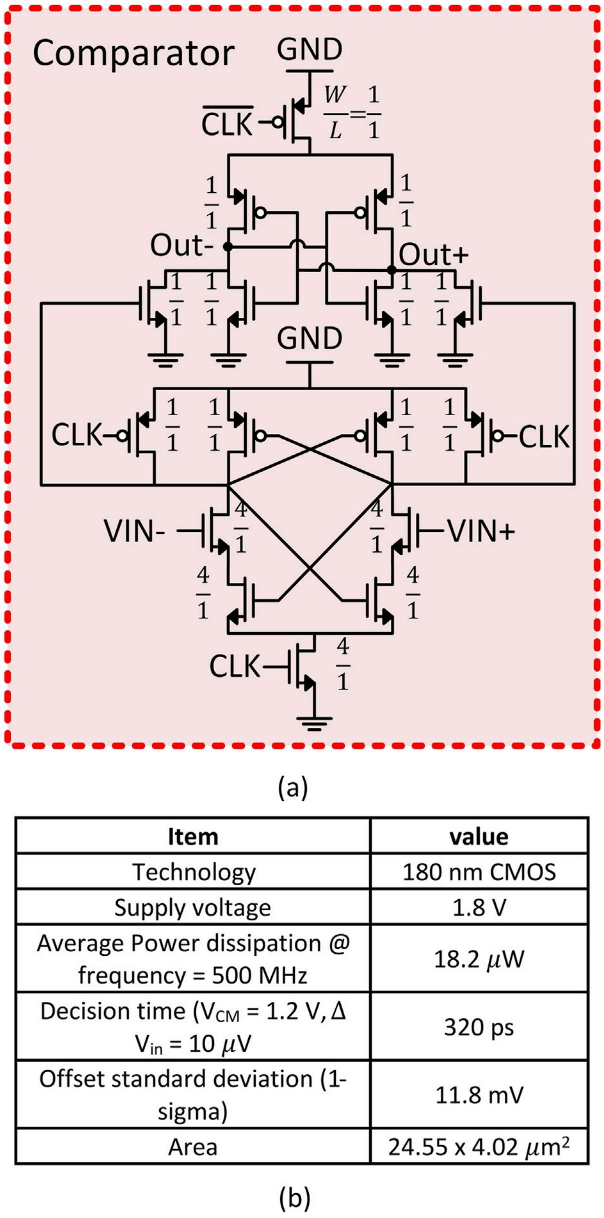
2.3.3 5-Bit UP/DOWN counter

The output analog to digital conversion in SPIKA is realized through synchronous 5-bit UP/DOWN counters (label 6, Figure 1) without the need for power-hungry ADCs as a result of the novel clicking mechanism implemented in this work. Every 2 adjacent columns in the array share one counter, when a column requests a click, the counter value is incremented or decremented whether it is the positive weight column or the negative weight column that

TABLE 1 RRAM mode of operations with relation to the WL/SL/BL voltages. X denotes do not care.

Mode	WL voltage	SL voltage	BL Prechage	BL voltage	VMR (LRS/HRS)	IMR (LRS/HRS)
Reading/VMM	VDD	GND	VDD	V_{Cap}	0.12 V/0.251 V	3.3 uA/82 nA
Writing	VDD	GND	X	VDD	0.8 V/1.06 V	19 uA/350 nA
Erasing	VDD	VDD	X	GND	−0.962 V/-1.15 V	−24 uA/−0.4 uA
OFF	GND	X	X	X	0	0

issued the click. If both columns issue a click at the same cycle, the counter remains at its value which is implemented by clock gating. The final output of the counter is a signed 5-bit digital value.



Item	value
Technology	180 nm CMOS
Supply voltage	1.8 V
Average Power dissipation @ frequency = 500 MHz	18.2 μW
Decision time (V_{CM} = 1.2 V, ΔV_{in} = 10 μV	320 ps
Offset standard deviation (1-sigma)	11.8 mV
Area	24.55 x 4.02 μm^2

FIGURE 3
(a) SPIKA double tail dynamic comparator schematic, the ratios on the transistors (W/L) are in comparison with minimum size transistor. **(b)** Summary of the comparator performance.

2.4 I/O interface, registers and drivers

SPIKA is a fully integrated macro and includes several registers, drivers and control units for I/O communication and configurations as shown in Figure 1 (purple color rectangle box). There are 4 I/O units used in SPIKA. First, from the input circuit, a 256-bit serial in parallel out (SIPO) register (label 7, Figure 1) is designed to stream the digital inputs of the neural network to the input circuit units. Besides, the input registers can also be used to activate selected WLs for writing or erasing operations of the RRAM cell. Second, the BL configuration SIPO registers and BL driver units (label 8 and label 9, Figure 1) are used to configure the selected BLs of the 1T1R crossbar for writing/reading or erasing operations. Third, the SL driver unit (label 10, Figure 1) is used to configure the SLs of the 1T1R crossbar to either writing/reading or erasing modes. Finally, a 320-bit parallel in serial out (PISO) register (label 11, Figure 1) is used to stream out the outputs of the column counters.

2.5 SPIKA clock module

In SPIKA, the clock distribution can be divided into two sections: first, an external clock supply for the I/O interface and serial registers which runs at a lower frequency (e.g., 10 MHz) as the control configuration signals are loaded once during the setup time. The other clock module (core clock module) is generated internally and it supplies the core computation elements (label 12, Figure 1). The core clock module is shown in Figure 4 and can be divided into three parts. The first part is defined as *CLK Generation* where three clock signals are generated from a single ring oscillator (RO) that feeds different blocks in the core. The RO is powered by an external DC current source (I_{CLK}) and produces a clock with a frequency of 500 MHz under a bias of 8.84 μA. The second part is *CLK Synchronization*, where the three clock edges are synchronized by current-starved buffers that are controlled by two external current sources I_{DELAY1} and I_{DELAY2}. The third part is called *CLK Buffer Tree* where the three clock signals are buffered by increasing size buffers before being fed to the core elements.

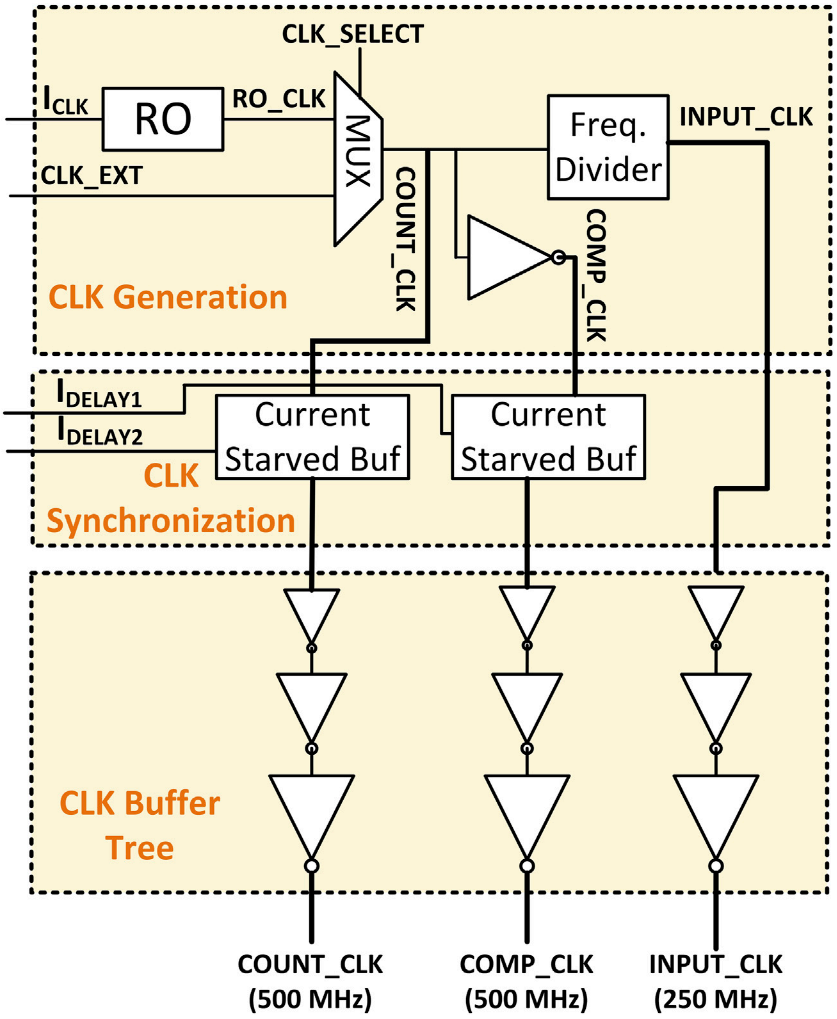


FIGURE 4
SPIKA core clock generation and distribution module.

3 Operation and design methodology

3.1 VMM process and clicking mechanism

The VMM process is the main operation mode in NNs. In order to achieve a highly efficient system, it is important to perform low-power and low-latency VMM operations. Figure 5 presents a simplified view of SPIKA architecture highlighting the VMM process by showing the necessary blocks to compute one 5-bit output (O_j). For an NxM VMM engine, the output vector (O_j) can be expressed as in (Equation 1) where x_i and W_ij are the digital inputs and weights of the NN, respectively.

$$O_j = \sum_{i=0}^{N} x_i W_{ij}. \tag{1}$$

In SPIKA, the VMM process is realized through a memory read operation. Once the inputs are loaded, the VMM operation is enabled where the digital inputs are modulated as discrete pulses through the DPC blocks. Then, the capacitor starts discharging through the activated 1T1R rows across the same column (integration time). The discharge rate of the capacitor is directly related to the combination of the inputs and weights where stronger inputs (higher number of pulses) and stronger weights (lower column resistance) lead to a higher discharge rate. When the BL voltage V_{BLj} reaches a threshold voltage V_{TH1}, the comparator issues a clicking request. This means that the column BL_j will only issue a charging request when its voltage drops below the comparator threshold, and not for every individual one-bit input multiplied by one-bit weight. Note that although the column requested a click, the clicking only occurs during periodic clicking time intervals. This allows for clicking synchronization from different columns and leads to a lower latency than asynchronous clicking as during charging time, the integration of inputs needs to be paused. More information about the column capacitor's charging and discharging process and design choices of threshold voltages is discussed in Section 3.2. Columns that requested a click increment or decrement the output counter. Additionally, the charging circuit is enabled for those columns and the capacitor is charged to refill the BL. Processes - are

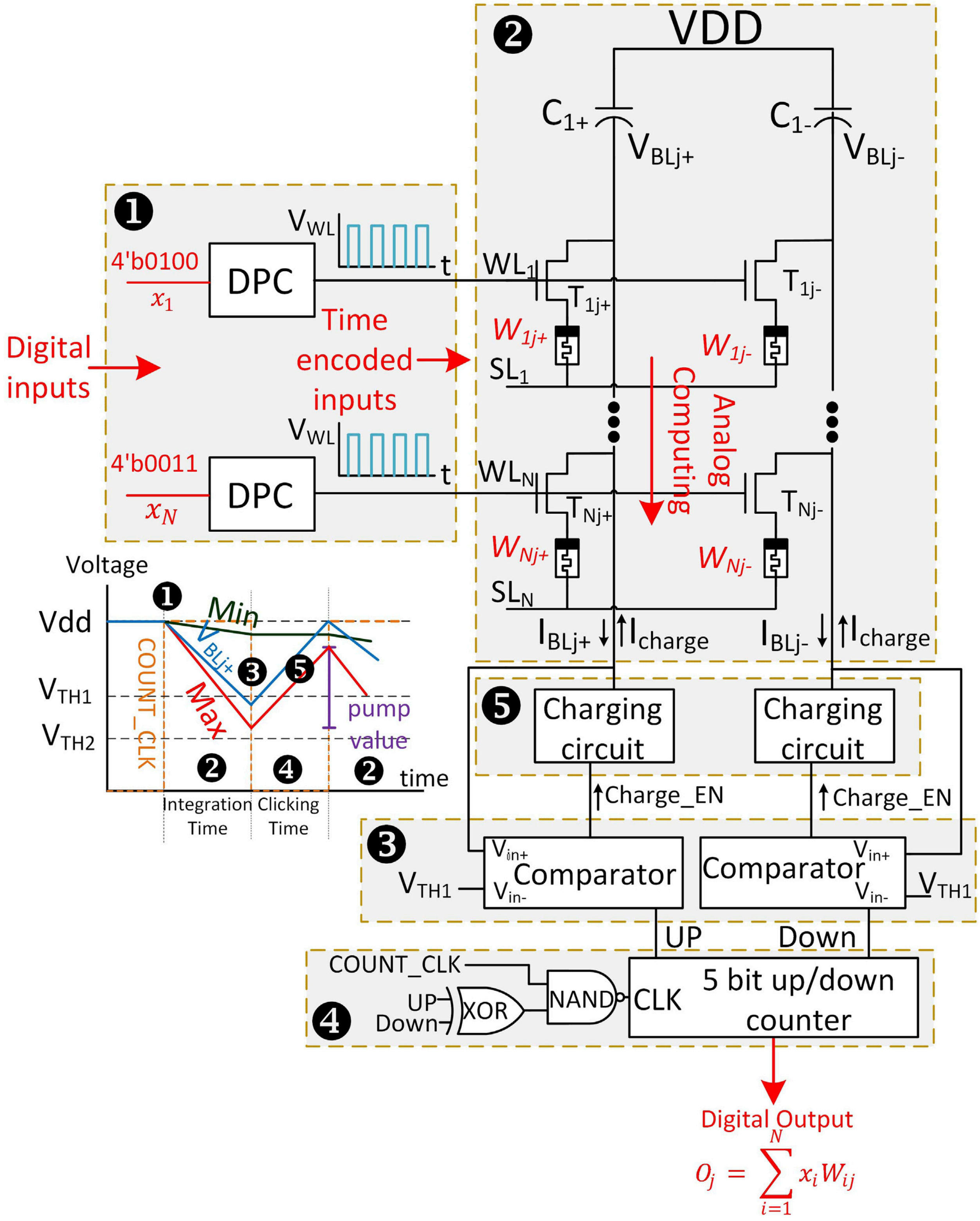


FIGURE 5
SPIKA simplified circuit schematic highlighting the VMM process and clicking mechanism.

repeated until all input pulses are consumed. Once the VMM process is over, the outputs of the counters represent the multiplication output of the VMM process.

The timing diagram in Figure 5 shows the discharge rate of column j (BL_j) with respect to the maximum and minimum discharge rates (input boundary cases). The maximum discharge

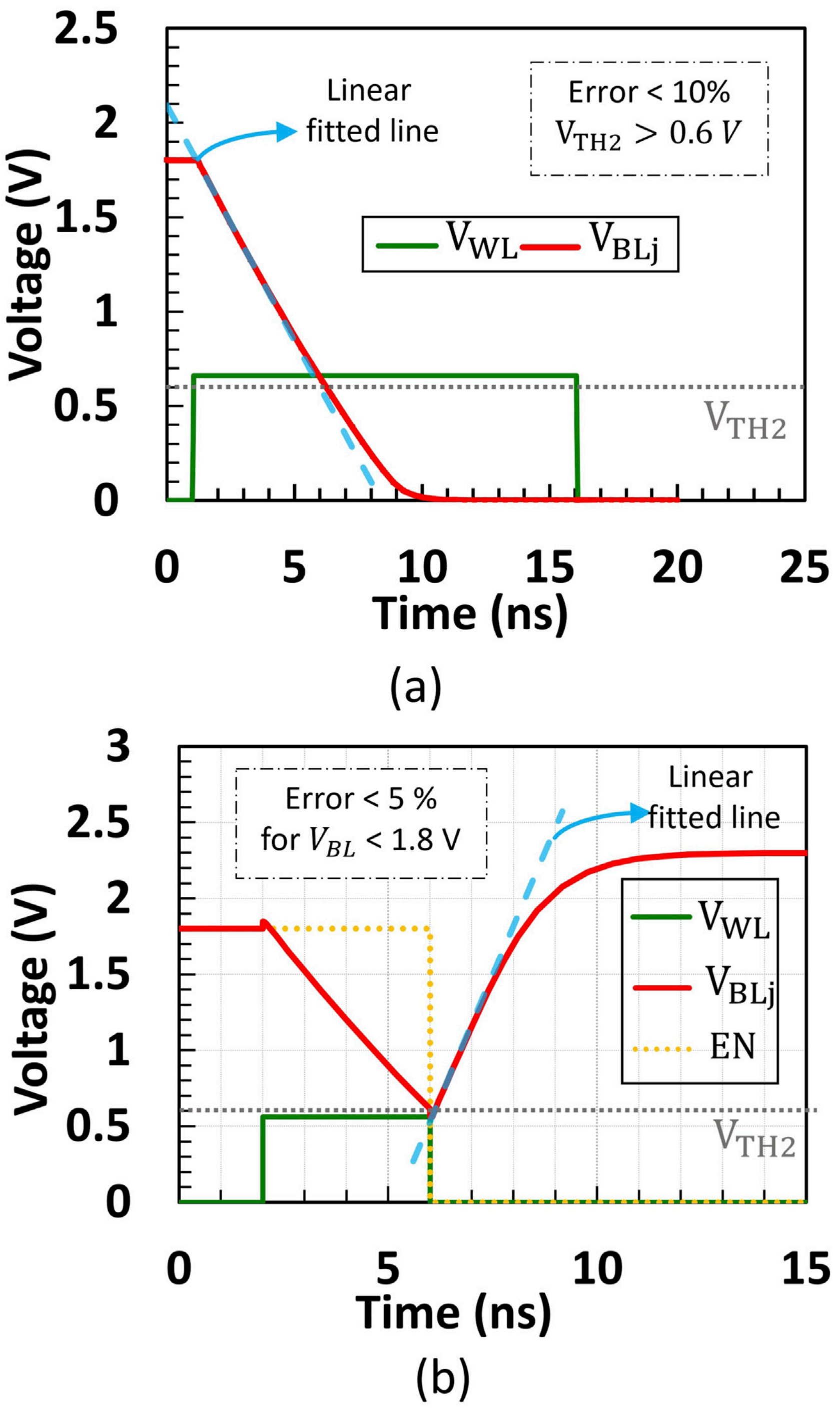


FIGURE 6
Transient simulation of the BL voltage (V_{BLj}) during **(a)** Capacitor discharge operation **(b)** Capacitor charge operation.

rate corresponds to the maximum output (15 clicks) which happens when all inputs (64 rows × 4-bit = 256-bits) are high, and all weights (64 weights per column) are in the (LRS) state. Conversely, the minimum discharge rate corresponds to the minimum output (0 clicks) where all inputs are high and all weights are in the HRS state. More details about input boundary cases are discussed in Section 3.2.4.

In SPIKA, every two adjacent columns share one counter. While one column stores positive weights and increments the counter, the other column stores the negative weights and decrements the counter once it requests a click. Hence, there is no need for additional subtractor units to implement the negative weights. The latency of one VMM operation in SPIKA is 60 ns and it is divided equally between integration time (active time) and clicking time (inactive time).

3.2 SPIKA constraints and design parameters

The key design objective in the VMM process is to ensure that the discharge rate of each column capacitor can be assumed linear, leading to linearly map the matrix product: Output = Inputs × Weights. Similarly, with an optimized current mirror design, the charging rate of the capacitor can also be assumed linear, leading to approximately lossless conversion. This section provides a summary of the design parameters and constraints in SPIKA.

3.2.1 WL voltage and RRAM read voltage

The WL voltage (V_{WL}) is supplied to the rows of the 1T1R array and controls the RRAM read/write voltage. To overcome the I-V non-linearity of the RRAM cell and reduce energy, the RRAM desired read voltage is selected to be between 0.1 V (LRS) and 0.2 V (HRS). Circuit simulations show that is achieved when (V_{WL}) is 0.525 V.

3.2.2 Capacitor-discharge operation and V_{TH2}

The discharge rate of the switched-based capacitor in this work controls the number of total clicks where higher discharge rates lead to a higher number of clicks. During integration time, the BL voltage of output j can be expressed by Equation 2.

$$V_{BLj} = -\frac{N \times V_{READ} \times t_{active}}{R \times C} \tag{2}$$

where N is the number of activated rows, V_{READ} is the RRAM read voltage, t_{active} is the total active time of the combined pulses, R is the RRAM resistance and C is the column capacitance. Among the design parameters, V_{READ} is controlled by V_{WL} which is an external voltage source. This allows for balancing the discharge rate even post-chip fabrication. Figure 6A shows a transient simulation of the BL voltage during integration time where the capacitor is fully discharged. The discharge rate of the capacitor during integration time can be assumed linear (with a maximum error percentage ~10% compared to a fitted line) approximately until the BL voltage fall below V_{TH2} = 0.6 V, thus, we use this value as the lower operating bound for V_{BL}, i.e., columns should click before V_{BL} falls below it.

3.2.3 Capacitor charge operation

The current mirror shown in Figure 1 is responsible for charging the capacitor after a click. An external DC current reference is fed to the current mirror to provide appropriate current to the columns of the 1T1R array. Hence, the charging rate can be controlled even post-chip fabrication. Similar to the discharge operation, the design parameters of the current mirror must be precisely designed to ensure that the charging rate of the capacitor is linearly approximated. By ensuring this, the capacitor will be charged with the same amount of charge regardless of the voltage it clicked at. Since the BL pre-charged is at 1.8 V, linearity needs to be presumed up to 1.8 V. This is implemented by powering the current mirror at a voltage higher than 1.8 V. In SPIKA, the minimum power supply required to maintain linearity (with a maximum error percentage ~5% compared to a fitted line) is found to be 2.3 V (VDD2) as shown in Figure 6B. This arrangement means that after a successful click, the voltage on the column capacitor increases by a very precise amount, underpinning lossless conversion.

3.2.4 Input boundary cases and system balance

In order to balance the system and the number of clicks, three boundary cases highlighted in Figure 7 need to be considered. The first case is denoted as Max input Max weight case (Case 1, Figure 7). In SPIKA each output is a 5-bit signed output which means that the maximum number of clicks a column can request is 15 clicks. In addition, since the number of rows is 64, the maximum number of active rows (N) is 64. The maximum row active time (30 ns) is achieved when the digital input is maximum, i.e. $4'$b1111. The second case is denoted as Min input Min weight or Min input Max weight (Case 2, Figure 7). In this case, the inputs are 0 (N = 0), thus, the column capacitor will not discharge through the 1T1R cells regardless of the values of the weights. The third case is denoted as Max input Min weight case (Case 3, Figure 7). In this case, all inputs are activated (N = 64) for the maximum active time (t_{active} = 30 ns) with all weights in the HRS. Ideally, the result of this case should be 0, but, due to the finite resistance of the RRAM cell, the capacitor will discharge through the 1T1R cells and the rate will depend on the HRS value of the RRAM cell. Hence, it is important in this case to have a high RRAM off-resistance and low V_{TH1} enough to ensure no clicks (i.e., the BL voltage never falls below V_{TH1}).

Thus, with the discharging and the charging rates linearity assumed earlier, balancing the system around the boundary cases leads to a linear mapping of the in-between cases. In SPIKA, the discharge rate is balanced by the external voltage supply V_{WL} and the charging rate is balanced by the external current source feeding the charging circuit (I_{CHARGE}). Post-layout, Monte-Carlo and parametric analysis simulations show that the system is balanced with (V_{WL}) = 0.660 V and (I_{CHARGE}) = 0.45 μA.

3.2.5 Upper clicking threshold voltage (V_{TH1})

V_{TH1} is the BL threshold voltage at which the column requests a click. The choice of V_{TH1} is very important as it affects the minimum HRS needed, total active time and number of clicking requests needed to balance the boundary cases of SPIKA. Lowering V_{TH1} reduces the required off-resistance but increases the total active time, hence, higher latency. In this work, V_{TH1} is selected to be 1.2 V.

3.2.6 SPIKA resolution and scalability

The choice of using 4-bit input/5-bit output resolutions in SPIKA is driven by the need to maximize power efficiency while maintaining acceptable classification accuracy. The 4-bit input resolution represents an optimal balance for the time-encoding scheme. The selection is driven by the well-known challenge of linear time encoding, where increasing from 4-bit to 5-bit encoding doubles the maximum time required (Serb and Prodromakis, 2019). We note that time-domain encoding tends to be most competitive in the 2-4-bit range. The use of ternary weight representation is based on the fact that RRAM devices only need to support two states, a relaxed requirement that broadens compatibility with various RRAM technologies being developed worldwide. Research has shown that ternary representation works well for most CNN classifiers (Yang et al., 2023). However, future work will explore multi-bit RRAM devices to enhance scalability further. To scale the system for larger networks, the approach could involve splitting

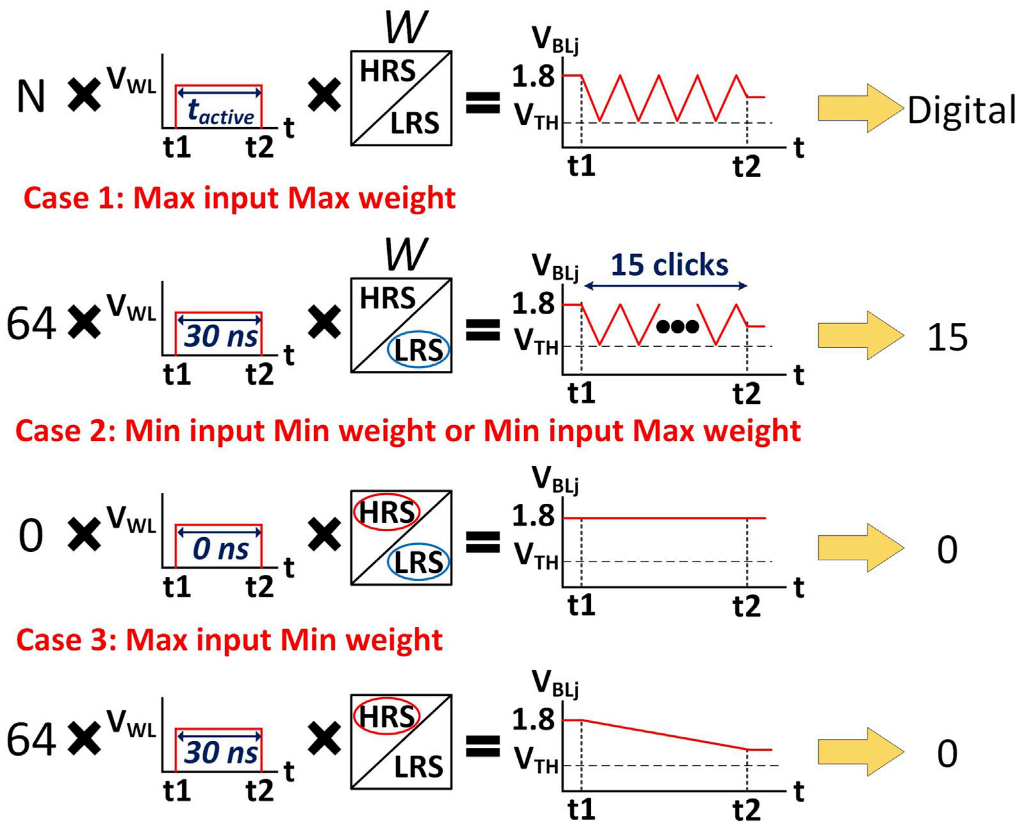

N
V_{WL}
t_{active}
t1
t2
t
W
HRS
LRS
V_{BLj}
1.8
V_{TH}
t1
t2
t
Digital
Case 1: Max input Max weight
64
30 ns
15 clicks
15
Case 2: Min input Min weight or Min input Max weight
0
0 ns
0
Case 3: Max input Min weight
64
30 ns
0

FIGURE 7
Input boundary cases of the clicking process. *N* is the number of activated rows, t_{active} is the total active time of the pulses, *t1* is the start time, *t2* is the end time and *W* is the weights of the NN.

high-resolution inputs, weights, and outputs into multiple adjacent ones rather than increasing the resolution. For example, an 8-bit input can be divided across two rows, a method that has been successful in system-level nvCIM designs like ISAAC (Shafiee et al., 2016) and PRIME (Chi et al., 2016).

4 Results

In this work, SPIKA macro has been validated in Cadence circuit tools by performing simulations and analysis on extracted post-layout views, including resistance and capacitance parasitics using a commercial 180 nm process and experimental RRAM models (Maheshwari et al., 2021a; Maheshwari et al., 2021b).

4.1 Transient simulation of input boundary cases

Figure 8 presents the SPICE post-layout simulation of SPIKA under the Max input Max weight case (Case 1, Figure 7) and Max input Min weight case (Case 3, Figure 7) showing the output result for two columns (BL_{1+} and BL_{1-}) corresponding to a positive weight column and negative weight column and shares one output (as highlighted in Figure 5). In this case, all positive weights are set to 1 (RRAM resistance = LRS) and all negative weights are set to 0 (RRAM resistance = HRS). Figure 8 shows the three clock signals that are generated from SPIKA clock module, namely,: $COUNT\_CLK$ (500 MHz), $COMP\_CLK$ (500 MHz) and $INPUT\_CLK$ (250 MHz). The BL voltage of the 2 columns BL_{1+} and BL_{1-} are also shown in Figure 8 where BL_{1+} requests 15 clicks and BL_{1-} requests 0 clicks. Finally, the 5-bit outputs of the digital counter (Q0-Q3 for magnitude and Z for sign) indicate the correct digital output (+15 at the end of conversion in this case).

4.2 CMOS process corner analysis

Table 2 provides a summary of the SPIKA process corner analysis conducted at 27 °C across five different CMOS process corners (nmos-pmos). The analysis is simulated under four distinct input (IN)/weight (W) combinations, where a higher percentage signifies stronger inputs and weights. For example, in case 1 (100% IN 100% W) all inputs are activated for the maximum duration and all weights are in the LRS. As we recall from Section 3.2.4, this case corresponds to the boundary case of the maximum input and maximum weight (15 clicks at the output). Table 2 shows the least significant bit (LSB) deviation between the simulated output and expected output for each corner and case. A deviation of 0 LSB indicates a correct conversion, highlighted in green in the table. Red-highlighted cells indicate incorrect conversions, along with the extent to which the simulated output deviates from the expected output.

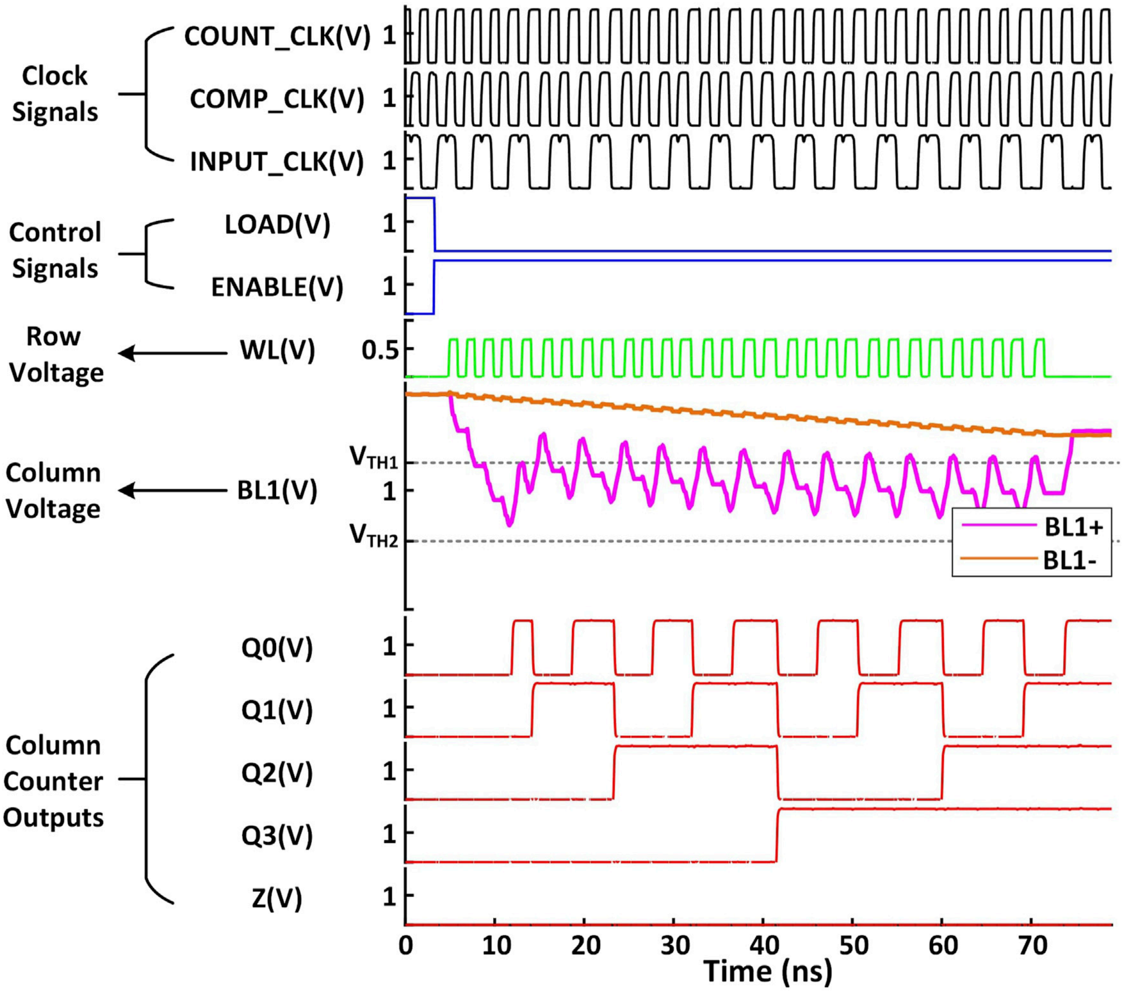


FIGURE 8
SPICE post-layout simulation of SPIKA for the Max input Max weight case (BL1+) and the Max input Min weight case (BL1-).

The results anticipated in Table 2 are divided into two segments, before balancing and after balancing. Recall that balancing in SPIKA is controlled by two external sources, V_{WL} and I_{CHARGE} which regulate the discharging and charging processes, respectively. The before-balancing segment in Table 2 shows the corner process variation on the output when using the V_{wL} optimized in Section 3.2.1. The results exhibit variations from case to case and corner to corner, with more active cases showcasing higher deviations. It is observed that the influence of corners is predominantly linked to the speed of nmos devices, wherein slower nmos transistors entail a lower discharge rate, leading to fewer clicks, while faster nmos transistors exhibit a higher discharge rate, resulting in more clicks. This variation is attributed to the fluctuation in the 1T1R voltage drops, wherein the access transistor is an nmos transistor that impacts the discharging rate.

Nevertheless, the inherent balancing mechanism in SPIKA, regulated by independent external sources, allows for the adjustment of the discharging rate to accommodate different corners. Parametric simulations demonstrated that adjusting only V_{WL} is sufficient to achieve this balance in SPIKA for all corners and cases. The outcomes are presented in the after-balancing segment of Table 2. As depicted, utilizing three distinct V_{WL} values enables the attainment of 0 LSB deviation in conversion for all corners and cases. It is noteworthy that this balancing strategy can be applied post-chip fabrication, highlighting SPIKA’s flexibility in operating across various process corners.

4.3 Temperature variation analysis

Similar to the process corner analysis, the SPIKA core was simulated across a temperature range spanning from −55°C to 125 °C and typical process corner. The outcomes for various IN/W cases are summarized in Table 3. The table comprises two sections, similar to the process corner analysis: before balancing and after balancing segments. In the before-balancing segment, the results indicate that lower temperatures lead to a lower discharge rate, resulting in fewer clicks, while higher temperatures result in a higher discharge rate and more clicks. As demonstrated before, the SPIKA system can be adjusted to function in various temperature ranges by optimizing the V_{WL} source. The use of four different V_{WL} values demonstrates the capability to achieve 0 LSB deviation in

TABLE 2 SPIKA process corner analysis conducted at 27°C for different input (IN)/weight (W) combinations. The least significant bit (LSB) provides an indication of the deviation between the simulated output and the expected output.

Case	Before balancing	Corner (nmos-pmos)					After balancing	Corner (nmos-pmos)				
		tt	ss	sf	fs	ff		tt	ss	sf	fs	ff
100% IN 100% W		0 LSB	−6 LSB	−5 LSB	4 LSB	3 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
50% IN 100% W		0 LSB	−4 LSB	−4 LSB	2 LSB	2 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
50% IN 50% W		0 LSB	−2 LSB	−1 LSB	1 LSB	2 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
100% IN 0% W		0 LSB	0 LSB	0 LSB	0 LSB	1 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
VWL		525 mV	525 mV	525 mV	525 mV	525 mV		525 mV	585 mV	585 mV	485 mV	485 mV

TABLE 3 SPIKA temperature variation analysis conducted at TT corner for different input (IN)/weight (W) combinations. The least significant bit (LSB) provides an indication of the deviation between the simulated output and the expected output.

Case	Before balancing	Temperature						After balancing	Temperature					
		−55 C	0 C	25 C	50 C	100 C	125 C		−55 C	0 C	25 C	50 C	100 C	125 C
100% IN 100% W		−7 LSB	−3 LSB	0 LSB	0 LSB	3 LSB	5 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
50% IN 100% W		−4 LSB	−1 LSB	0 LSB	0 LSB	2 LSB	3 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
50% IN 50% W		−2 LSB	−1 LSB	0 LSB	0 LSB	1 LSB	1 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
100% IN 0% W		0 LSB	0 LSB	0 LSB	0 LSB	1 LSB	1 LSB		0 LSB	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB
V_{WL}		525 mV	525 mV	525 mV	525 mV	525 mV	525 mV		580 mV	540 mV	525 mV	525 mV	475 mV	475 mV

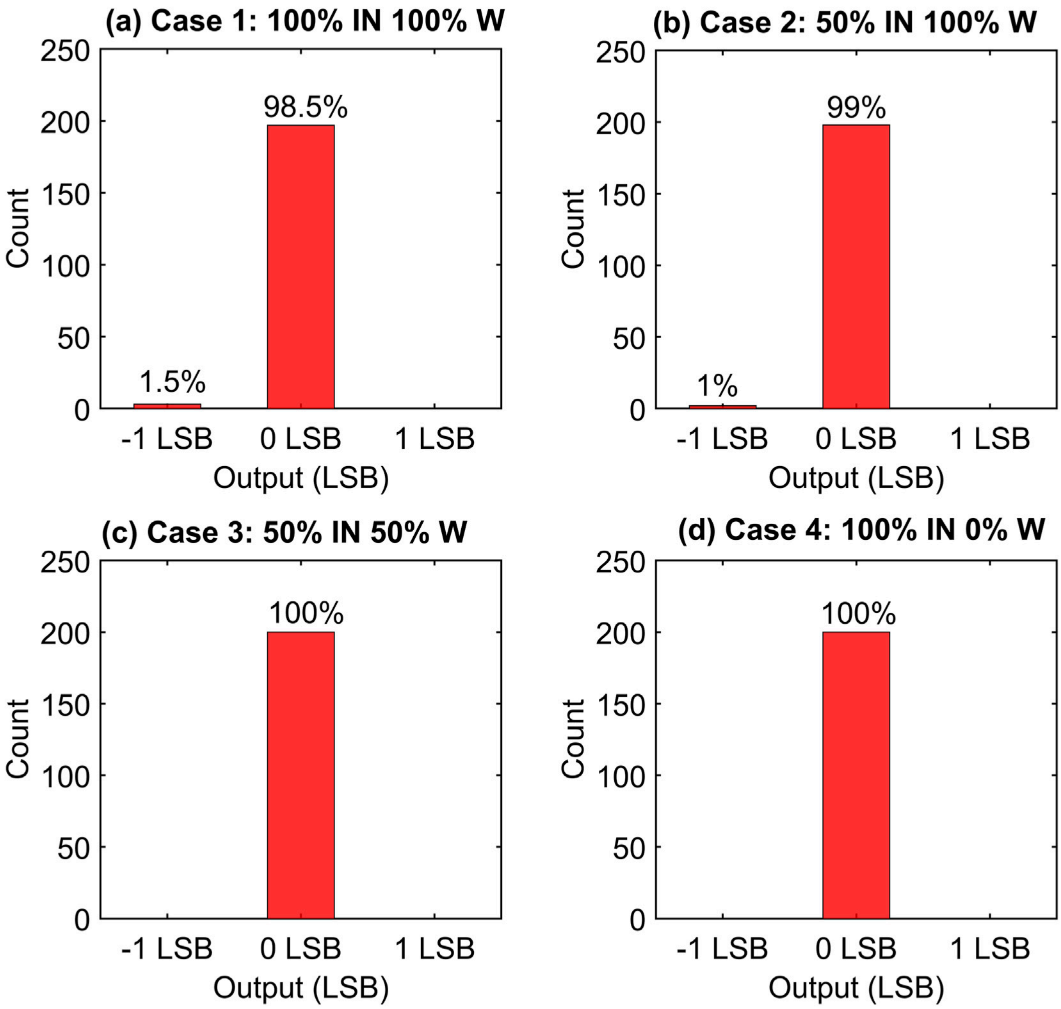


FIGURE 9
(a–d) Histogram of Monte Carlo process mismatch simulated at the TT Corner and room temperature for four distinct input/weight combinations. The correct output is represented by 0 LSB.

TABLE 4 SPIKA RRAM variation analysis conducted at 27°C. The least significant bit (LSB) provides an indication of the deviation between the simulated output and the expected output. The low resistive state (LRS) is set to 40 kΩ, while the high resistive state (HRS) is set to 3 MΩ.

Resistance	Deviation							
	−20%	−15%	−10%	−5%	5%	10%	15%	20%
LRS	2 LSB	1 LSB	0 LSB	0 LSB	0 LSB	0 LSB	−1 LSB	−2 LSB
HRS	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB	0 LSB

conversion across the temperature range. This underscores the flexibility of SPIKA in adapting to different temperature conditions.

4.4 Monte Carlo analysis

SPIKA core have been simulated using Monte Carlo analysis to investigate the impact of CMOS mismatch. Figures 9A–D presents the histogram results for four distinct input (IN)/weight (W) combinations. The LSB represents the difference between the simulated output and the expected output, with 0 LSB indicating a correct conversion. As depicted in Figure 9, the lowest success rate is seen in case 1 (Figure 9A) with a 98.5% success rate. This outcome aligns with expectations, as more active input/weight combinations involve a higher number of active 1T1R cells, introducing more transistor mismatch effect. Conversely, the success rate is found to

TABLE 5 SPIKA macro area and power breakdown.

Module	Average power consumption (mW)	Area (mm^2)
Input circuit (64 DPC units)	1.93	0.0384
1T1R array	0.20	0.2300
Charging circuit	0.33	0.0150
Comparators	3.10	0.0120
Output counters	0.20	0.0340
Total SPIKA core	5.76	0.3294
Clock generation and drivers	3.33	0.0012
Input registers	2.54	0.0380
Output registers	2.88	0.0412
BL conf registers and drivers	2.24	0.0537
SL drivers	0.10	0.0017
Total SPIKA chip	16.85	0.4652

be 100% in cases 3 and 4, where the number of activated input/weight cells is lower.

4.5 RRAM variation analysis

As mentioned previously in Section 2.2, we implement a ternary weight encoding scheme in SPIKA, where the RRAM devices are only required to support a total of 2× states. This deliberately relaxed requirement is designed to lower the barrier to entry for a wide range of emerging RRAM technologies currently under development worldwide. Since the resistance of RRAM devices can be trimmed and reprogrammed after fabrication, the impact of process variations is less critical compared to CMOS circuitry. Numerous studies have demonstrated that verify-write techniques, which iteratively program and verify the resistance state until the desired value is achieved, enable precise tuning of RRAM conductance. This approach effectively mitigates process-induced variations, enhancing overall device reliability and performance (Zhang et al., 2019; Shim et al., 2020).

However, Table 4 demonstrates that even in the absence of a successful write-verify mechanism, the SPIKA system remains robust against variations in both LRS and HRS values. In this analysis, RRAM LRS and HRS resistances were swept from −20% to +20%, and any deviation from the correct conversion accuracy was quantified in terms of LSB error. It is important to highlight that these simulations represent a worst-case scenario, where all RRAM devices in the array are subjected to identical variation. The results indicate that noticeable conversion deviation only begins when LRS variations exceed 15%, with the maximum error observed being 2 LSBs at a 20% variation. These findings suggest that such high levels of variation—15%–20% across all RRAM devices simultaneously—represent an extremely unlikely worst-case scenario, and the resulting deviations remain minimal, further highlighting the robustness of the SPIKA system to device-level variability.

4.6 Power and area breakdown

Table 5 presents the average power consumption and area breakdown of each component in SPIKA including the I/O and control circuitry. The metrics in Table 5 are based on post-layout extracted simulations and views and a randomized set of inputs and weights. The power and area overhead of the input and output circuits in SPIKA are 57% and 26%, respectively, which is an improvement of 22% and 6% of the power and area overhead of the ISAAC structure (Shafiee et al., 2016) running at 32 nm CMOS technology. The computational core consumes an average of ~5.6 mW running under a 500 MHz clock frequency and 1.8 V power supply. It is worth mentioning that the I/O registers and drivers within SPIKA were not optimized in any manner and were only incorporated as functional elements to facilitate testing procedures.

5 Circuit-level metrics and benchmarking

5.1 SPIKA energy efficiency and throughput and comparison with baseline nv-CIM

The energy per operation for a given task is a key metric to benchmark hardware accelerators (Seo et al., 2022). The computational blocks in SPIKA consume an average power of 5.6 mW during a single VMM process, with an array size of 64 × 128 and a randomized sequence of 4-bit inputs and 1-bit weights (Section 4.5). The latency for a single VMM process in SPIKA is 60 ns at a clock frequency of 500 MHz. Thus, the throughput of SPIKA, defined as the number of operations per second (where each MAC = 2 ops), is calculated as follows: 64 × 128 ×2ops÷60ns = 273 GOPS or "bit-normalized" throughput of 1092 GOPS. By "bit-normalized" we mean a reduction to 1-bit input × 1-bit/analog weights (Jiang et al., 2023). The energy efficiency of the system is found by dividing the throughput by the power consumption, resulting in 48.75 TOPS/W (4b × 1b) or 195 TOPS/W bit-normalized (1b × 1b). Similar to previous work in this field, the reported throughput and energy efficiency represent their peak values when the CIM array utilization is 100%, and do not include time and energy spent on buffering and moving intermediate data. Supplementary Figure S1 in the Supplementary Material illustrates the energy efficiency across various design choices, including input resolution, output resolution, and array size.

To benchmark SPIKA against the baseline nv-CIM, we refer to the ISAAC core reported by Shafiee et al., which utilizes DACs for input encoding and ADCs for output conversion, performing multiplications in the current domain (Shafiee et al., 2016). The ISAAC core consumes average power of 27.5 mW during a single VMM process with a 128 × 128 array size. The latency for a single VMM process is 100 ns. Thus, the throughput is calculated as 128 × 128 ×2ops÷100ns = 327.68 GOPS or "bit-normalized" throughput of 655.36 GOPS. Additionally, the core's energy efficiency is 11.9 TOPS/W (1b × 2b) or 23.8 TOPS/W bit-normalized (1b × 1b). Compared to ISAAC, the proposed SPIKA

TABLE 6 Comparison table with state-Of-The-Art CIM macro designs. norm.: Bit-Normalized. EE, Energy efficiency.

Work	ISCA'16 Shafiee et al. (2016)	VLSI'17 Su et al. (2017)	CICC'21 Li et al. (2021)	JCCS'22 Khaddam-Aljameh et al. (2022)	Nature'22 Wan et al. (2022)	TCAS1'23 Jiang et al. (2023)	TCAS1'23 Xuan et al. (2023)	ESSERC'24 Yao et al. (2024)	ISSCC'24 Spetalnick et al. (2024)	SPIKA
Approach[a]	CD	CD	CD	TD	TD	TD	TD	CD	TD	TD
CMOS technology	32 nm	150 nm	40 nm	14 nm	130 nm	40 nm	180 nm	28 nm	40 nm	180 nm
Implementation	Sim	Chip	Chip	Chip	Chip	Chip	Post-Layout Sim	Chip	Chip	Post-Layout Sim
Array Size	128 × 128	64 × 64	128 × 128	256 × 256	256 × 256	256 × 256	256 × 64	512 × 512	256 × 256	64 × 128
Bits Resolution (In/W/O)	1/2/8	1/1/3	1/8/3	8/1/8	8/4/10	8/2/8	4/4/14	4/4/8	N/A	4/1/5
Read voltage	N/A	N/A	N/A	N/A	0.5 V	0.9 V	1.8 V	0.9 V	1.1 V	0.2 V
Frequency	1.5 GHz	20 MHz	100 MHz	1 GHz	N/A	100 MHz	N/A	150 MHz	80 MHz	500 MHz
Power consumption (per core)	27.5 mW	22 mW	N/A	N/A	N/A	N/A	N/A	N/A	N/A	5.6 mW
VMM Latency	100 ns	0.1 m	N/A	N/A	N/A	N/A	370 ns	100 ns	N/A	60 ns
Throughput (GOPS)	327.68	0.082	20.96	1,008	2,135	13.93	N/A	2084	268.8	273
EE (TOPS/W)[b]	11.9	3.73	36.39	10.5	43	26.97	10.8	N/A	0.84	48.75
Bit-norm. Throughput (GOPS)[c]	655.36	0.082	20.96	8,064	8,540	222.88	N/A	N/A	N/A	1,092
Bit-norm. EE (TOPS/W)	23.8	3.73	36.39	84	172	431.52	172.8	308.8	53.76	195
Bit-norm. EE SPIKA improvement	8.19×	52.27×	5.36×	2.32×	1.13×	0.45×	1.12×	0.63×	3.63×	1×
EE (TOPS/W) at 14 nm[d]	62.12	428.16	297.06	84	3,707.65	220.16	1782	N/A	17.01	8,054.19
Bit-norm. EE (TOPS/W) at 14 nm	124.24	428.16	297.06	84	14829	3,522.61	28,612	1,235.2	1,008.64	32,216

(Continued on following page)

TABLE 6 (*Continued*) Comparison table with state-Of-The-Art CIM macro designs. norm.: Bit-Normalized. EE, Energy efficiency.

Work	ISCA'16 Shafiee et al. (2016)	VLSI'17 Su et al. (2017)	CICC'21 Li et al. (2021)	JCCS'22 Khaddam-Aljameh et al. (2022)	Nature'22 Wan et al. (2022)	TCAS1'23 Jiang et al. (2023)	TCAS1'23 Xuan et al. (2023)	ESSERC'24 Yao et al. (2024)	ISSCC'24 Spetalnick et al. (2024)	SPIKA
SPIKA EE improvement at Bit-norm. 14 nm	259×	75.29×	108.3×	390×	2.15×	9.15×	1.12×	26×	31.94×	1×

[a] CD:Current-Domain, TD, Time-domain.

[b] Takes only the analog computing cell into account, if not stated in the reference, the value is calculated back.

[c] Normalized to 1-bit input x 1-bit/analog weight.

[d] Energy Efficiency Normalized at 14 nm using Dennard scaling.

core demonstrates a 1.67× improvement in throughput and an 8.19× enhancement in energy efficiency.

5.2 Comparison with state of the art

Table 6 provides a performance summary of SPIKA CIM macro compared to prior works. The throughput and energy efficiency are determined for the computing cells only so that system-level related effects are not taken into account. Where reported values were system-level, they were recalculated specifically for CIM macros. Compared with the core proposed by Cai running at a similar technology node (180 nm) and array size (Cai et al., 2019), SPIKA provides a 12× improvement in power. The total latency of 1 VMM operation (60 ns) is a 2× improvement vs. ISAAC (Shafiee et al., 2016) and 6× vs. Marinella et al. (2018).

The SPIKA CIM macro, implemented with a 180 nm technology node, demonstrates an energy efficiency of 48.75 TOPS/W (for 4b input x 1b weight), which is comparable to state-of-the-art implementations at advanced technology nodes. To provide further perspective we also show bit-normalised throughput and energy efficiency. The SPIKA core ranks highest in raw energy efficiency in the table and third-highest in bit-normalized energy efficiency (Jiang et al., 2023). and (Yao et al., 2024) topped the bit-normalized energy efficiency. We attribute this to a combination of using a more advanced technology node and including no output circuits, where the outputs of the CIM macro remain in analog form and are not converted to digital until later at the system level.

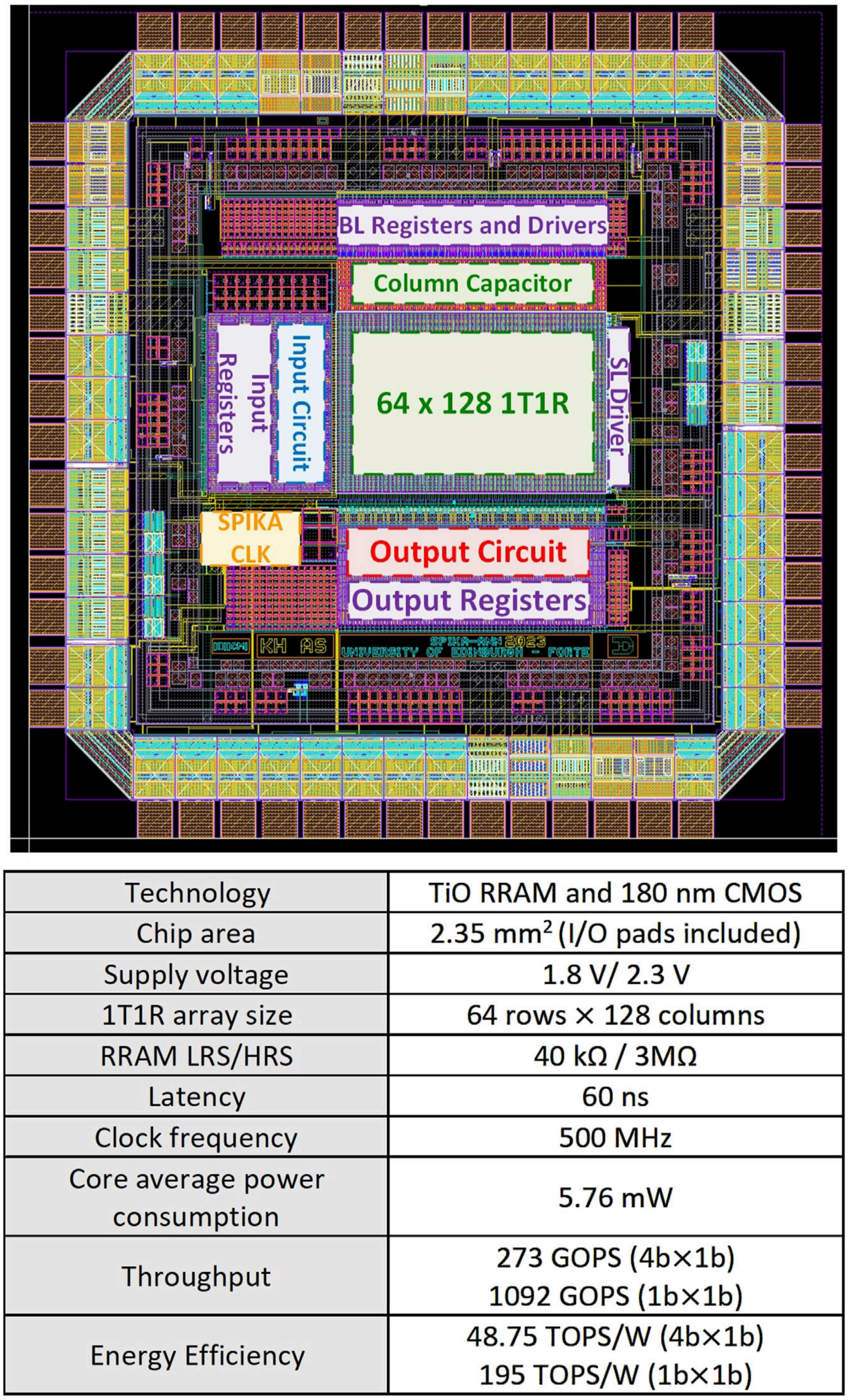
For further perspective, we also projected the energy efficiency of all designs to 14 nm using Dennard scaling assumptions (normalization factor of $(\frac{CMOSnm}{14})^2$) (Dennard et al., 1974), as presented in Table 6. The proposed system exhibits an estimated improvement in normalized energy efficiency at 14 nm ranging from 2.12× to 390× vs. state-of-art. We note that the scalability of SPIKA to such nodes necessitates further investigation.

5.3 Chip summary

Figure 10 presents a post-layout capture of SPIKA chip (I/O pads included) along with a specification summary. Our system performance could be further improved by using more advanced technology nodes and optimizing the computing architecture and peripheral circuits. For example, the column capacitors could be integrated beneath the RRAM array if additional metal layers are available, this could save up to 4% of the total area from savings in the core alone. In addition, having more metal layers could decrease the pitch size of the 1T1R cell and the peripheral circuits leading to a denser structure.

6 Conclusion and future work

This work proposed a novel time-domain RRAM-based non-volatile compute-in-memory (nvCIM) 64 × 128 macro, SPIKA, for neural network acceleration including all the necessary I/O interface and control circuitry. The system architecture, components and methodology are discussed in detail. The key novelty of this work is



Technology	TiO RRAM and 180 nm CMOS
Chip area	2.35 mm^2 (I/O pads included)
Supply voltage	1.8 V/ 2.3 V
1T1R array size	64 rows × 128 columns
RRAM LRS/HRS	40 kΩ / 3MΩ
Latency	60 ns
Clock frequency	500 MHz
Core average power consumption	5.76 mW
Throughput	273 GOPS (4b×1b) 1092 GOPS (1b×1b)
Energy Efficiency	48.75 TOPS/W (4b×1b) 195 TOPS/W (1b×1b)

FIGURE 10
SPIKA Post-layout chip capture and specification summary.

the efficient transition of the signal from the input to the output with the minimum overhead needed which substantially improved the performance and power consumption of the proposed in comparison to previous works. SPIKA was evaluated based on extracted post-layout simulation and analysis using 180 nm CMOS commercial process and experimental RRAM models.

The characterization results obtained in this work highlight the potential of nvCIM macros in AI edge computation. The SPIKA core, as designed here, is optimized for vector matrix multiplication (VMM) operations, a critical function across various neural network classifiers, including deep neural networks (DNNs), convolutional neural networks (CNNs), and spiking neural networks (SNNs). The decision to utilize 4-bit inputs, 5-bit inputs, and ternary weights is aimed at maximizing power efficiency while ensuring satisfactory classification accuracy. Similar configurations have demonstrated impressive classification performance, as demonstrated by Yang et al. (2023) and Zhang et al. (2024). Future work will focus on developing a multi-core system-level design for SPIKA, enabling the efficient implementation of comprehensive workloads.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

KH: Conceptualization, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review and editing. YP: Software, Visualization, Writing – review and editing. GR: Software, Validation, Writing – review and editing. MM: Software, Writing – review and editing. SW: Supervision, Writing – review and editing. AS: Investigation, Methodology, Supervision, Writing – review and editing. TP: Funding acquisition, Supervision, Writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Programme Grant EP/R024642/2 “Functional Oxide Reconfigurable Technologies (FORTE)” and the RAEng Chair in Emerging Technologies (CiET 1819/2/93).

Acknowledgments

The authors would like to thank Shady Agwa and Sachin Maheshwari for their valuable discussion and support.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/felec.2025.1567562/full#supplementary-material

References

Alemdar, H., Leroy, V., Prost-Boucle, A., and Pétrot, F. (2017). “Ternary neural networks for resource-efficient ai applications,” in 2017 International Joint Conference on Neural Networks (IJCNN), USA, 14-19 May 2017, 2547–2554. doi:10.1109/ijcnn.2017.7966166

Amirsoleimani, A., Alibart, F., Yon, V., Xu, J., Pazhouhandeh, M. R., Ecoffey, S., et al. (2020). In-memory vector-matrix multiplication in monolithic complementary metal–oxide–semiconductor-memristor integrated circuits: design choices, challenges, and perspectives. *Adv. Intell. Syst.* 2, 2000115. doi:10.1002/aisy.202000115

Ankit, A., Sengupta, A., Panda, P., and Roy, K. (2017). Resparc: a reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks, 1, 6. doi:10.1145/3061639.3062311

Babayan-Mashhadi, S., and Lotfi, R. (2014). Analysis and design of a low-voltage low-power double-tail comparator. *IEEE Trans. Very Large Scale Integration (VLSI) Syst.* 22, 343–352. doi:10.1109/TVLSI.2013.2241799

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate

Bayat, F. M., Prezioso, M., Chakrabarti, B., Nili, H., Kataeva, I., and Strukov, D. (2018). Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nat. Commun.* 9, 2331. doi:10.1038/s41467-018-04482-4

Cai, F., Correll, J. M., Lee, S. H., Lim, Y., Bothra, V., Zhang, Z., et al. (2019). A fully integrated reprogrammable memristor–cmos system for efficient multiply–accumulate operations. *Nat. Electron.* 2, 290–299. doi:10.1038/s41928-019-0270-x

Chen, W.-H., Dou, C., Li, K.-X., Lin, W.-Y., Li, P.-Y., Huang, J.-H., et al. (2019). Cmos-integrated memristive non-volatile computing-in-memory for ai edge processors. *Nat. Electron.* 2, 420–428. doi:10.1038/s41928-019-0288-0

Chi, P., Li, S., Xu, C., Zhang, T., Zhao, J., Liu, Y., et al. (2016). “Prime: a novel processing-in-memory architecture for neural network computation in reram-based main memory,” in *2016 ACM/IEEE 43rd annual international symposium on computer architecture (ISCA)*, 27–39. doi:10.1109/ISCA.2016.13

Dennard, R., Gaensslen, F., Yu, H.-N., Rideout, V., Bassous, E., and LeBlanc, A. (1974). Design of ion-implanted mosfet’s with very small physical dimensions. *IEEE J. Solid-State Circuits* 9, 256–268. doi:10.1109/JSSC.1974.1050511

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks, 6645, 6649. doi:10.1109/icassp.2013.6638947

Hertel, L., Barth, E., Käster, T., and Martinetz, T. (2015). “Deep convolutional neural networks as generic feature extractors,” in 2015 International Joint Conference on Neural Networks (IJCNN), USA, 12-17 July 2015, 1–4. doi:10.1109/ijcnn.2015.7280683

Humood, K., Hadi, S. A., Mohammad, B., Jaoude, M. A., Alazzam, A., and Alhawari, M. (2019). “High-density reram crossbar with selector device for sneak path reduction,” in 2019 31st International Conference on Microelectronics (ICM), USA, 15-18 December 2019, 244–248. doi:10.1109/ICM48031.2019.9021944

Humood, K., Pan, Y., Wang, S., Serb, A., and Prodromakis, T. (2024). Design of a low-power digital-to-pulse converter (dpc) for in-memory-computing applications. *Microelectron. J.* 153, 106420. doi:10.1016/j.mejo.2024.106420

Humood, K., Serb, A., Wang, S., and Prodromakis, T. (2023a). “Power, performance and area optimization of parallel load counters through logic minimization and tspc-ff utilization,” in 2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), USA, 04-07 December 2023, 1–5. doi:10.1109/ICECS58634.2023.10382888

Humood, K., Serb, A., Wang, S., and Prodromakis, T. (2023b). “Quicknn: Python toolbox for training and optimizing ann for hardware implementation,” in *2023 IEEE*

66th international midwest symposium on circuits and systems (MWSCAS), 531–535. doi:10.1109/MWSCAS57524.2023.10405963

Hung, J.-M., Xue, C.-X., Kao, H.-Y., Huang, Y.-H., Chang, F.-C., Huang, S.-P., et al. (2021). A four-megabit compute-in-memory macro with eight-bit precision based on cmos and resistive random-access memory for ai edge devices. *Nat. Electron.* 4, 921–930. doi:10.1038/s41928-021-00676-9

Jiang, H., Huang, S., Li, W., and Yu, S. (2023). Enna: an efficient neural network accelerator design based on adc-free compute-in-memory subarrays. *IEEE Trans. Circuits Syst. I Regul. Pap.* 70, 353–363. doi:10.1109/TCSI.2022.3208755

Ji-Ren, Y., Karlsson, I., and Svensson, C. (1987). A true single-phase-clock dynamic cmos circuit technique. *IEEE J. Solid-State Circuits* 22, 899–901. doi:10.1109/JSSC.1987.1052831

Kadetotad, D., Xu, Z., Mohanty, A., Chen, P. Y., Lin, B., Ye, J., et al. (2015). Parallel architecture with resistive crosspoint array for dictionary learning acceleration. *IEEE J. Emerg. Sel. Top. Circuits Syst.* 5, 194–204. doi:10.1109/JETCAS.2015.2426495

Khaddam-Aljameh, R., Stanisavljevic, M., Fornt Mas, J., Karunaratne, G., Brändli, M., Liu, F., et al. (2022). Hermes-core—a 1.59-tops/mm2 pcm on 14-nm cmos in-memory compute core using 300-ps/lsb linearized cco-based adcs. *IEEE J. Solid-State Circuits* 57, 1027–1038. doi:10.1109/JSSC.2022.3140414

Li, B., Lixue, X., Peng, G., Wang, Y., and Huazhong, Y. (2015). "Merging the interface: power, area and accuracy co-optimization for rram crossbar-based mixed-signal computing system," in *2015 52nd ACM/EDAC/IEEE design automation conference (DAC)*, 1–6. doi:10.1145/2744769.2744870

Li, W., Huang, S., Sun, X., Jiang, H., and Yu, S. (2021). "Secure-rram: a 40nm 16kb compute-in-memory macro with reconfigurability, sparsity control, and embedded security," in 2021 IEEE Custom Integrated Circuits Conference (CICC), USA, 25-30 April 2021, 1–2. doi:10.1109/CICC51472.2021.9431558

Liu, Q., Gao, B., Yao, P., Wu, D., Chen, J., Pang, Y., et al. (2020). "33.2 a fully integrated analog reram based 78.4tops/w compute-in-memory chip with fully parallel mac computing," in 2020 IEEE International Solid- State Circuits Conference - (ISSCC), USA, 16-20 February 2020, 500–502. doi:10.1109/ISSCC19947.2020.9062953

Maheshwari, S., Stathopoulos, S., Wang, J., Serb, A., Pan, Y., Mifsud, A., et al. (2021a). Design flow for hybrid cmos/memristor systems—part i: modeling and verification steps. *IEEE Trans. Circuits Syst. I Regul. Pap.* 68, 4862–4875. doi:10.1109/TCSI.2021.3122343

Maheshwari, S., Stathopoulos, S., Wang, J., Serb, A., Pan, Y., Mifsud, A., et al. (2021b). Design flow for hybrid cmos/memristor systems—part ii: circuit schematics and layout. *IEEE Trans. Circuits Syst. I Regul. Pap.* 68, 4876–4888. doi:10.1109/TCSI.2021.3122381

Marinella, M. J., Agarwal, S., Hsia, A., Richter, I., Jacobs-Gedrim, R., Niroula, J., et al. (2018). Multiscale co-design analysis of energy, latency, area, and accuracy of a reram analog neural training accelerator. *IEEE J. Emerg. Sel. Top. Circuits Syst.* 8, 86–101. doi:10.1109/jetcas.2018.2796379

Ming, C., Lixue, X., Zhenhua, Z., Yi, C., Yuan, X., Yu, W., et al. (2017). "Time: a training-in-memory architecture for memristor-based deep neural networks," in *2017 54th ACM/EDAC/IEEE design automation conference (DAC)*, 1–6. doi:10.1145/3061639.3062326

Mittal, S. (2019). A survey of reram-based architectures for processing-in-memory and neural networks. *Mach. Learn. Knowl. Extr.* 1, 75–114. doi:10.3390/make1010005

Mochida, R., Kouno, K., Hayata, Y., Nakayama, M., Ono, T., Suwa, H., et al. (2018). "A 4m synapses integrated analog reram based 66.5 tops/w neural-network processor with cell current controlled writing and flexible network architecture," in *2018 IEEE symposium on VLSI technology*, 175–176.

Musisi-Nkambwe, M., Afshari, S., Barnaby, H., Kozicki, M., and Sanchez Esqueda, I. (2021). The viability of analog-based accelerators for neuromorphic computing: a survey. *Neuromorphic Comput. Eng.* 1, 012001. doi:10.1088/2634-4386/ac0242

Narayanan, S., Shafiee, A., and Balasubramonian, R. (2017). "Inxs: bridging the throughput and energy gap for spiking neural networks," in 2017 International Joint Conference on Neural Networks (IJCNN), China, 14-19 May 2017, 2451–2459. doi:10.1109/IJCNN.2017.7966154

Prezioso, M., Mahmoodi, M. R., Bayat, F. M., Nili, H., Kim, H., Vincent, A., et al. (2018). Spike-timing-dependent plasticity learning of coincidence detection with passively integrated memristive circuits. *Nat. Commun.* 9, 5311. doi:10.1038/s41467-018-07757-y

Sahay, S., Bavandpour, M., Mahmoodi, M. R., and Strukov, D. (2020). Energy-efficient moderate precision time-domain mixed-signal vector-by-matrix multiplier exploiting 1t-1r arrays. *IEEE J. Explor. Solid-State Comput. Devices Circuits* 6, 18–26. doi:10.1109/JXCDC.2020.2981048

Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., and Eleftheriou, E. (2020). Memory devices and applications for in-memory computing. *Nat. Nanotechnol.* 15, 529–544. doi:10.1038/s41565-020-0655-z

Seo, J. S., Saikia, J., Meng, J., He, W., Suh, H. S., Anupreetham, H., et al. (2022). Digital versus analog artificial intelligence accelerators: advances, trends, and emerging designs. *IEEE Solid-State Circuits Mag.* 14, 65–79. doi:10.1109/MSSC.2022.3182935

Serb, A., and Prodromakis, T. (2019). "An analogue-domain, switch-capacitor-based arithmetic-logic unit," in *2019 IEEE international symposium on circuits and systems (ISCAS)*, 1–5. doi:10.1109/ISCAS.2019.8702070

Shafiee, A., Nag, A., Muralimanohar, N., Balasubramonian, R., Strachan, J. P., Hu, M., et al. (2016). "Isaac: a convolutional neural network accelerator with *in-situ* analog arithmetic in crossbars," in *2016 ACM/IEEE 43rd annual international symposium on computer architecture (ISCA)*, 14–26. doi:10.1109/ISCA.2016.12

Shim, W., sun Seo, J., and Yu, S. (2020). Two-step write–verify scheme and impact of the read noise in multilevel rram-based inference engine. *Semicond. Sci. Technol.* 35, 115026. doi:10.1088/1361-6641/abb842

Spetalnick, S. D., Lele, A. S., Crafton, B., Chang, M., Ryu, S., Yoon, J.-H., et al. (2024). 30.1 a 40nm vliw edge accelerator with 5mb of 0.256pj/b rram and a localization solver for bristle robot surveillance. *2024 IEEE Int. Solid-State Circuits Conf. (ISSCC)* 67, 482–484. doi:10.1109/ISSCC49657.2024.10454500

Su, F., Chen, W. H., Xia, L., Lo, C. P., Tang, T., Wang, Z., et al. (2017). "A 462gops/j rram-based nonvolatile intelligent processor for energy harvesting ioe system featuring nonvolatile logics and processing-in-memory," in *2017 symposium on VLSI technology*, T260–T261. doi:10.23919/VLSIT.2017.7998149

Tang, S., Yin, S., Zheng, S., Ouyang, P., Tu, F., Yao, L., et al. (2017). "Aepe: an area and power efficient rram crossbar-based accelerator for deep cnns," in *2017 IEEE 6th non-volatile memory systems and applications symposium (NVMSA)*, 1–6. doi:10.1109/NVMSA.2017.8064475

Wan, W., Kubendran, R., Schaefer, C., Eryilmaz, S. B., Zhang, W., Wu, D., et al. (2022). A compute-in-memory chip based on resistive random-access memory. *Nature* 608, 504–512. doi:10.1038/s41586-022-04992-8

Wang, Y., Tang, T., Xia, L., Li, B., Gu, P., Yang, H., et al. (2015). Energy efficient rram spiking neural network for real time classification. *GLSVLSI '15: Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, 189, 194. doi:10.1145/2742060.2743756

Xia, L., Tang, T., Huangfu, W., Cheng, M., Yin, X., Li, B., et al. (2016). "Switched by input: power efficient structure for rram-based convolutional neural network," in 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), USA, 4 Nov 2022, 1–6. doi:10.1145/2897937.2898101

Xuan, Z., Liu, C., Zhang, Y., Li, Y., and Kang, Y. (2023). A brain-inspired adc-free sram-based in-memory computing macro with high-precision mac for ai application. *IEEE Trans. Circuits Syst. II Express Briefs* 70, 1276–1280. doi:10.1109/TCSII.2022.3224049

Xue, C., Huang, T., Liu, J., Chang, T., Kao, H., Wang, J., et al. (2020). "15.4 a 22nm 2mb reram compute-in-memory macro with 121-28tops/w for multibit mac computing for tiny ai edge devices," in 2020 IEEE International Solid-State Circuits Conference - (ISSCC), USA, 16-20 February 2020, 244–246. doi:10.1109/ISSCC19947.2020.9063078

Yang, X., Zhu, K., Tang, X., Wang, M., Zhan, M., Lu, N., et al. (2023). An in-memory-computing charge-domain ternary cnn classifier. *IEEE J. Solid-State Circuits* 58, 1450–1461. doi:10.1109/JSSC.2023.3238725

Yao, P., Wei, Q., Wu, D., Gao, B., Yang, S., Shen, T.-Y., et al. (2024). "A 28 nm rram-based 81.1 tops/mm2/bit compute-in-memory macro with uniform and linear 64 read channels under 512 4-bit inputs," in 2024 IEEE European Solid-State Electronics Research Conference (ESSERC), Belgium, 09-12 September 2024, 577–580. doi:10.1109/ESSERC62670.2024.10719511

Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., et al. (2020). Fully hardware-implemented memristor convolutional neural network. *Nature* 577, 641–646. doi:10.1038/s41586-020-1942-4

Yu, S., Jiang, H., Huang, S., Peng, X., and Lu, A. (2021). Compute-in-memory chips for deep learning: recent trends and prospects. *IEEE Circuits Syst. Mag.* 21, 31–56. doi:10.1109/MCAS.2021.3092533

Yu, S., Li, Z., Chen, P. Y., Wu, H., Gao, B., Wang, D., et al. (2016). "Binary neural network with 16 mb rram macro chip for classification and online training," in *2016 IEEE international electron devices meeting (IEDM)*. doi:10.1109/IEDM.2016.7838429

Zhang, B., Saikia, J., Meng, J., Wang, D., Kwon, S., Myung, S., et al. (2024). Macc-sram: a multistep accumulation capacitor-coupling in-memory computing sram macro for deep convolutional neural networks. *IEEE J. Solid-State Circuits* 59, 1938–1949. doi:10.1109/JSSC.2023.3332017

Zhang, Y., Zhu, C., Zhang, L., and Wang, Z. (2019). "A write-verification method for non-volatile memory," in 2019 International Conference on IC Design and Technology (ICICDT), USA, 17-19 June 2019, 1–3. doi:10.1109/ICICDT.2019.8790834