Tempo-CIM: A RRAM Compute-in-Memory Neuromorphic Accelerator With Area-Efficient LIF Neuron and Split-Train-Merged-Inference Algorithm for Edge AI Applications

Jingwen Jiang, Keji Zhou, *Member, IEEE*, Jinhao Liang, Fengshi Tian, *Graduate Student Member, IEEE*, Chenyang Zhao, Jianguo Yang, *Member, IEEE*, Xiaoyong Xue, *Member, IEEE*, and Xiaoyang Zeng, *Senior Member, IEEE*

***Abstract*— Spiking neural network (SNN)-based compute-in-memory (CIM) accelerator provides a prospective implementation for intelligent edge devices with higher energy efficiency compared with artificial neural networks (ANN) deployed on conventional Von Neumann architectures. However, the costly circuit implementation of biological neurons and the immature training algorithm of discrete-pulse networks hinder efficient hardware implementation and high recognition rate. In this work, we present a 40nm RRAM CIM macro (Tempo-CIM) with charge-pump-based leaky-integrate-and-fire (LIF) neurons and split-train-merged-inference algorithm for efficient SNN acceleration with improved accuracy. The single-spike latency coding is employed to reduce the number of pulses in each time step. The voltage-type LIF neuron uses a charge pump structure to achieve efficient accumulation and thus reduce the requirement for large capacitance remarkably. The split-train-merged-inference algorithm is proposed to dynamically adjust the input of each neuron to alleviate the spike stall problem. The macro measures 0.084mm^2 in a 40nm process with an energy efficiency of 68.51 TOPS/W and an area efficiency of 0.1956 TOPS/mm^2 for 4b input and 8b weight.**

***Index Terms*— Computing in memory (CIM), resistive random-access-memory (RRAM), spiking neural network (SNN), neuromorphic accelerator.**

Manuscript received 31 May 2023; revised 31 August 2023; accepted 25 September 2023. Date of publication 2 October 2023; date of current version 29 December 2023. This work was supported in part by the STI 2030—Major Projects under Grant 2022ZD0209200; in part by the National Natural Science Foundation of China under Grant 62274038, Grant 92164204, and Grant 61934002; in part by the Science and Technology Commission of Shanghai Municipality under Grant 21TS1401200 and Grant 22ZR1407100; and in part by the Opening Project of Zhejiang Laboratory under Grant 2022PF0AB01. This article was recommended by Guest Editor A. Basu. *(Corresponding author: Xiaoyong Xue.)*

Jingwen Jiang, Keji Zhou, Jinhao Liang, Chenyang Zhao, Xiaoyong Xue, and Xiaoyang Zeng are with the State Key Laboratory of Integrated Chips and Systems, School of Microelectronics, Fudan University, Shanghai 201203, China (e-mail: xuexiaoyong@fudan.edu.cn).

Fengshi Tian is with the ECE Department, The Hong Kong University of Science and Technology (HKUST), Hong Kong, SAR, China.

Jianguo Yang is with the Laboratory of Microelectronics Devices and Integrated Technology, Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JETCAS.2023.3321107.

Digital Object Identifier 10.1109/JETCAS.2023.3321107

I. Introduction

THE steady development of artificial intelligence (AI) has made the Internet of Things (IoT) a reality and led to a tremendous explosion of smart applications. However, owing to the limited energy and hardware resources on edge devices, it remains a challenge to support the operation of deep neural networks (DNN) with a huge number of parameters. Quantization and pruning are common approaches for weight compression of deep networks but often at the cost of system performance [1]. Compared with the convolutional neural network (CNN)-based algorithms, spiking neural network (SNN) has the natural advantage of being event-driven by mimicking the human brain. In addition, SNNs work with temporal data, which is sparse and further contributes to a lower expense in terms of parameter number and data transport [2], [3], [4]. For hardware implementation, the data-intensive multiply-and-accumulate (MAC) operations in CNN/SNN bring frequent data transfers between storage and processing units in traditional architecture such as CPU and GPU, resulting in serious energy and latency burdens. To deal with the problem of von Neumann bottleneck, computing in memory (CIM) has been proposed to reduce the data transfer by executing computation within or near storage arrays [5]. In typical CIM architecture, the weight parameters are stored in memory and the input activations are converted to different voltage signals. By applying the input on one side of the memory array, the voltage is passed through a memory cell to produce a current, and the obtained current sum is proportional to the result of MAC operations. Therefore, MAC operations can be performed in high parallelism within the memory array and further enables matrix-vector-multiplication (MVM). By regulating the order of the data stream, CIM can achieve large-scale reuse of stored weights [6], [7], [8]. Given the advantage mentioned above, SNN deployed on CIM architecture provides a prospective implementation for intelligent edge devices with higher energy efficiency compared with CNN/SNN deployed on conventional Von Neumann architectures [9], [10], [11]. However, challenges still exist restraining SNN CIM accelerator from practical applications, as shown in Fig. 1.

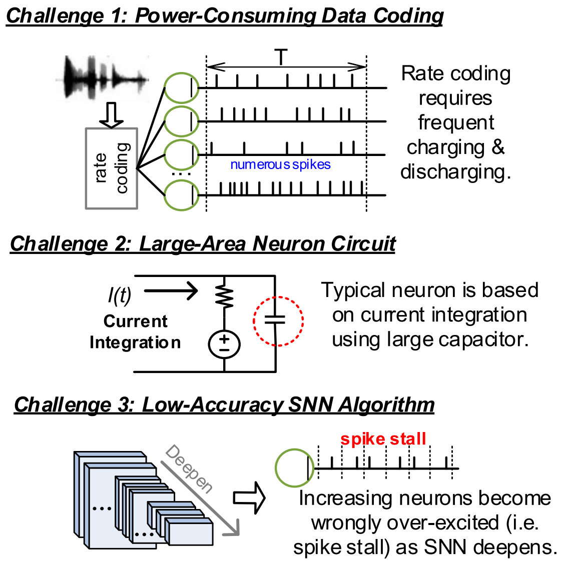

Challenge 1: Power-Consuming Data Coding
T
rate coding
numerous spikes
Rate coding requires frequent charging & discharging.
Challenge 2: Large-Area Neuron Circuit
I(t)
Current Integration
Typical neuron is based on current integration using large capacitor.
Challenge 3: Low-Accuracy SNN Algorithm
Deepen
spike stall
Increasing neurons become wrongly over-excited (i.e. spike stall) as SNN deepens.

Fig. 1. Challenges for SNN CIM accelerator in AI edge applications.

Firstly, the conventional data coding method in the time domain is power-consuming for frequent charging & discharging in the implementation circuit. Information coding converts real-world spatial signals into temporal information, requiring a trade-off between signal-to-noise ratio, sampling rate, and data density. Mainstream coding methods can be divided into rate coding and temporal coding. Most SNNs, e.g., spiking-based CNN (SCNN), use rate coding, where real values are encoded into fire rate, i.e., the average number of pulses issued in a time window [12]. This coding approach is more resistant to noise, but it produces reduplicative synaptic activities in each time step, leading to large power consumption from frequent charging & discharging. As for temporal coding schemes, the precise timing of and between spikes is required to be recorded for information storage [13], [14]. However, the registering of absolute timing in asynchronous circuits will result in large circuit complexity. Some temporal encoders use Poisson distributions of spikes, which requires a pseudo-random number generator to work at each time step, bringing tremendous power overhead [15], [16].

Secondly, there is a lack of reliable and area-efficient spiking neuron circuits. Neurons are an important computational component in SNNs to realize pre-neuron signals combination and processing [17]. The modeling of spiking neurons focuses on simulating the changes of membrane potential of neurons in the process of receiving external signals. Compared with neurons in CNN that realize summation and activation functions, the structure of the spiking neuron model is more complex due to rich biomimetic dynamic behaviors. The commonly reported spiking neuron models lie in Hodgkin-Huxley (H-H) model [18], [19], FitzHugh-Nagumo model [20], Izhikevich model [21] and leaky-integrate-fire (LIF) model [22]. In comparison, the LIF model that features low computational complexity is broadly applied in SNNs. In particular, the "integration" phase is essential for the assembling of presynaptic signals. In the digital domain, "integration" can be realized by digital counters and adders [4], [23]. Even with good noise suppression, digital implementations cause large area and power consumption [24], [25]. In contrast, analog circuits can make good use of physics laws and thus "integration" is often realized by the charge accumulation effect of capacitors. Conventional neurons also rely heavily on capacitance to store membrane potentials [26]. Although the mechanism of capacitive integration can be easily implemented on silicon-based neuron circuits, the need for large capacitors imposes high area costs for hardware resources [27]. The neuron in [28] used multiple capacitors as integrators, the largest of which is 100 fF (20 nm process), and the total area occupied by the capacitors is more than half of the neuron circuit layout. In an attempt to look for novel devices or circuit structures to replace capacitors, many efforts have focused on emerging non-volatile memories (NVMs) for achieving 'integration' operations and storing membrane potentials. For instance, the phase configuration of a chalcogenide-based phase-change memory (PCM) device has been explored to store the membrane potentials of a neuron [29]. Moreover, the conducting filaments of RRAM with built-in stochastic dynamics have been adopted to mimic the membrane capacitance of a neuron and to enable the accumulation of temporal inputs. As the membrane potential is dependent on the length of the conducting filaments, the requirement of area-consuming capacitors can be eliminated [30], [31]. However, the limited write speed of RRAMs and PCMs may restrict the upper limits of the working frequency of the system [5], [32]. The relatively low endurance for typical RRAM devices may hinder the life time of RRAM-based neurons due to the continuous updating of membrane potentials [30].

Thirdly, there is a lack of SNNs with high accuracy due to the immaturity of bio-inspired training algorithms [33]. Some SNN algorithm approaches are biologically plausible in function but with complex dynamics and the networks only have a single layer or shallow layers [34], [35], [36]. Limited by the depth of the network, shallow NN with a small number of parameters fails to achieve high accuracy on large-scale data sets. The works for deep NN fall into two main approaches. One is CNN to SNN conversion. Firstly, a modified CNN is customized according to the specified SNN structure and trained like a normal CNN with certain constraints. Then the obtained weights are mapped from the CNN to the SNN after adjustments [37], [38], [39], [40]. However, this approach makes approximations and fails to utilize biodynamics, which leads to accuracy loss from conversion. The other one for deep SNN implementation is spatio-temporal backpropagation (STBP), which combines the weight updating method with back propagation, making deep SNN be trained directly [41], [42], [43], [44], [45]. However, both methods adopt the gradient descent method, and as the network becomes deeper, the problems like gradient vanishing and gradient exploding arise [46]. Therefore, the deep SNN training algorithm is prone to the spike-stall problem featuring numerous over-excited neurons, resulting in remarkable accuracy loss.

To address these challenges, we present a 40nm RRAM CIM macro named Tempo-CIM with area-efficient LIF neuron and split-train-merged-inference algorithm for efficient

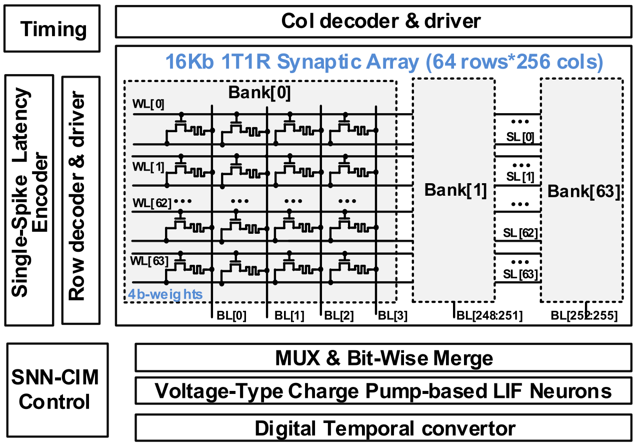


Fig. 2. Architecture of Tempo-CIM.

SNN acceleration with improved accuracy. The 16Kb 1T1R (1 transistor and 1 RRAM unit) array behaves as synapses for high-density weight storage and enables efficient in-memory computing. The single spike latency coding circuit uses a linear mapping method to efficiently convert the digital input to temporal pulses with low energy consumption. The voltage-type LIF neuron uses our patented charge pump structure to achieve efficient accumulation [47]. Instead of using single capacitance for charge accumulation and storage, the charge transport efficiency of the charge pump structure mainly depends on the capacitance ratio rather than the absolute value of the capacitance. The use of the charge pump reduces the requirement for large capacitance remarkably and avoids the huge area overhead. The split-train-merged-inference algorithm is proposed to dynamically adjust the input of each neuron and alleviate the spike stall problem. With fewer neurons to deliver invalid output signals, the accuracy of the SNN training method is remarkably improved.

The remainder of the paper is organized as follows. The design of the Tempo-CIM accelerator is described in Section II. The design and analysis of the area-efficient neuron is shown in Section III and the split-training-merged-inference algorithm is presented in Section IV. Evaluations are made in Section V, and conclusions are drawn in Section VI.

II. Architecture of Tempo-CIM

A. Accelerator Architecture

Fig. 2 shows the overall architecture of the proposed SNN CIM accelerator. It is composed of a 16Kb 1T1R array, single-spike latency encoder, voltage-type charge pump-based LIF neurons, digital temporal converter circuit, and other digital peripheral circuits. The single-spike latency encoder supports the input neural encoding. It has a total of 64 groups of encoding circuits, which process 64 4-b inputs in the time domain parallelly. The input signal is linearly mapped into a temporal square-wave pulse with a corresponding delay within each time step (to be described in Section II-B). The 1T1R array is divided into 64 banks with a size of 64 rows and 4 columns. 64 signed 4-b synaptic weights are stored as RRAM conductance in each bank. When the word line (WL) is on and the input voltage is applied on the source lines (SL), there will be voltage drops on RRAMs and a current is then generated. According to Kirchhoff's law, a sum of currents that flows through the 64 1T1R cells connected to one column will be obtained on a bit line (BL). This value theatrically represents the result of vector multiplication of the input activation and the weight. The voltage-type charge pump-based LIF neurons perform a nonlinear integration of the BL signals. The intensity of the input signal determines the accumulation speed of the membrane potential of the neuron. When the membrane potential reaches a certain threshold, the neuron issues a temporal pulse as an output and the membrane potential will be reset (to be described in Section III-A). The digital temporal converter circuit converts the neuronal output to a digital signal through a multi-bit counter, which facilitates the storage and transmission of the temporal signal (to be described in Section III-B).

B. Singe-Spike Latency Coding

Fig. 3(a) shows the circuit of singe-spike latency coding, which converts the N-bit digital input from the spatial domain to a single pulse signal in the time domain. In order to produce a delay signal with the highest possible linearity at the lowest possible cost in terms of area overhead and power consumption, a voltage-controlled delay chain circuit is used to produce a delay that is linearly and negatively correlated with the input. The voltage-controlled delay unit (VCD0) circuit generates a unit delay which can be adjusted by setting $V_{DLY\text{-}CTRL}$. The gates of transistors M3 and M8 are controlled by voltage $V_{DLY\text{-}CTRL}$, which limits the currents flowing through the inverters consisting of transistors M1 & M2 and M6 & M7. Thus, the voltage $V_{DLY\text{-}CTRL}$ regulates the propagation delay of the inverter. In addition, C1 and C2 are two small MOM capacitors with an area of only 1um$*$1um. By deploying two sets of identical delay circuits, the propagation delay of high to low (t_{pHL}) and low to high (t_{pLH}) caused by the VCD0 circuits can be basically the same. The results of 200 Monte Carlo simulations indicate that the VCD0 circuit can achieve a stable delay from 20ns to 0.5ms to enable different speed adjustments of the neuromorphic architecture. Given the effects of layout design and PVT variations, the test latency of VCD0 is not always the same as the pre-simulation latency. Fortunately, with the use of a voltage-controlled delay chain circuit, the linear relationship between input and output still holds regardless of the test unit delay. By adjusting the voltage of $V_{DLY\text{-}CTRL}$, the test unit delay can be made to match the pre-set latency. The voltage to delay (VtoD) circuit contains M sets of delay chains to process M bit binary input, and the i^{th} set of the chain has 2^{M-i} delay units in series to generate a certain delay required for the corresponding bit. By selectively connecting VCD0 circuits in series, the VtoD circuit generates a delay in linearity with the 4-b input. The Trans-detect circuit generates an output with a pulse width of a unit delay.

Fig. 3(b) shows the waveform diagram of the encoded pulse corresponding to different 4-b inputs. For example, for an input value 3, the binary input IN<3:0> equals 0011, and IN_B<3:0> equals 1100. Since IN<3> is 0, the delay units

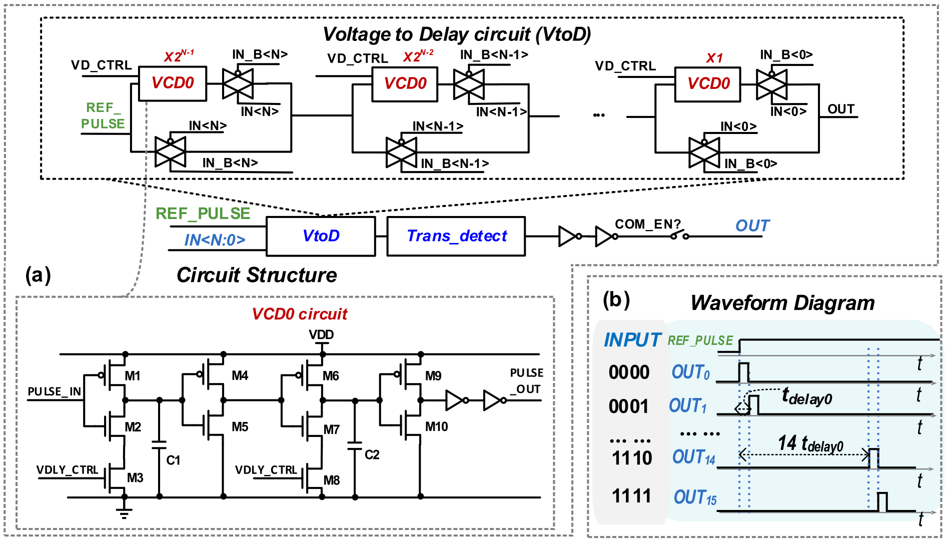


Fig. 3. (a) The structure of proposed singe-spike latency coding circuit for input conversion, (b) the waveform diagram of encoded pulse corresponding to 4bit different input data.

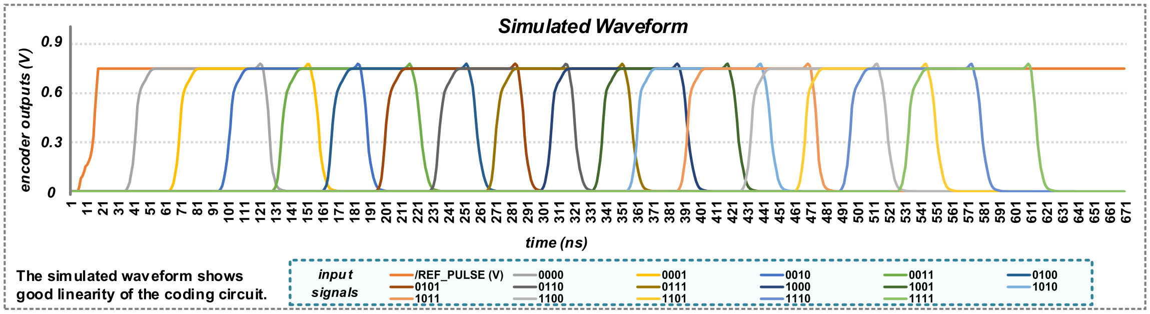


Fig. 4. Simulated waveform for the singe-spike latency coding circuit.

of the first stage are not turned on and *REF_PULSE* is passed directly backwards. Similarly, the second delay chain does not generate a delay as well. By the third stage of the delay chain, as *IN*<1> is 1, the delay units are turned on and the *REF_PULSE* signal is delayed by $2^1 = 2$ units. Similarly, when passing through the fourth delay chain, the signal is again delayed by 1 unit. The total delay for the '0011' input signal is $2^1 + 2^0 = 3$ units based on the start time of *REF_PULSE*. The proposed single-spike latency coding can be described by (1):

$$V(t) = V_0 \cdot (u(2^N - i) \cdot T_d - T_0) - V_0 \cdot (u(2^N - i - 1) \cdot T_d - T_0) \qquad (1)$$

where, T_o refers to the delay caused by gate delay, routing delay, etc. ideally equals to 0, T_d refers to the unit delay time of the VCD0 circuit, i refers the value of the N-bit input in decimal, and $u(\cdot)$ refers the step function. The simulated result of the single spike encoder is given in Fig. 4 and shows good linearity.

C. Work Flow of Tempo-CIM

The workflow of Tempo-CIM is shown in Fig. 5. The initial operation is for weights writing. The weights obtained from the training process are grouped in sequence and then written to each 1-b RRAM cell in the write-verify scheme to ensure the write accuracy, with 4 RRAM cells storing one 4-b weight.

The first phase is for reading from the input register (IR), where external data is preprocessed and stored as input activation. Due to the limited IO resources, the quantized external data will be imported serially and stored in the input register first, and be read out simultaneously in the next phase. The second phase is temporal encoding. In this phase, the input activations will be encoded into time-domain signals for SNN processing. The 4-b digital inputs read from IR will be linearly mapped into a temporal square pulse within each time step. The third phase is for CIM operation, with the *CIM_EN* control signal positive, WLs are all turned on. The encoded pulses are then applied on source lines of the synaptic array as SL voltage V_{SL}. The accessed 1T1R cells in one column are connected to one BL so that the BL current I_{BL} represents the current sum from the connected 1T1R cells, and thus the MAC result.

The fourth phase is bit merging. A bit-wise merge circuit sums the currents of bit lines from different columns based on the weight precision, which are then processed by the neurons. And the fifth phase is neuron processing. The voltage-type charge pump-based LIF neurons accumulate the current sum

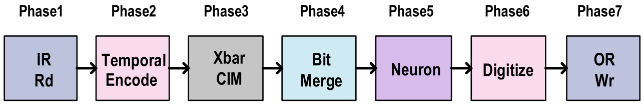


Fig. 5. SNN-CIM work flow inter-macro.

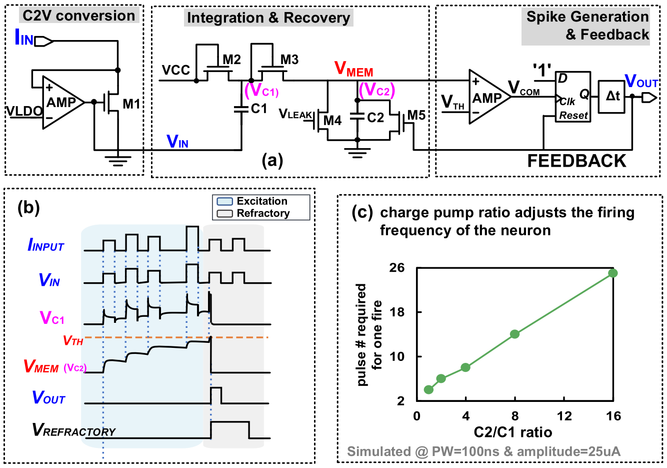


Fig. 6. (a) Proposed voltage-type charge-pump-based LIF neuron circuit; (b) waveform; (c) the influence of charge pump ratio on the firing frequency of the neuron.

and determine whether to emit an output pulse or not by comparing the integrated voltage to a reference voltage. In this phase, the BL current outputs are processed and mapped into the temporal form. The sixth phase is for temporal digital conversion. The digital temporal converter circuit converts the neuronal output to a digital signal through a multi-bit counter, which facilitates the storage and transmission of the temporal signal for further inter-macro-operations. In the last phase, the digital 4-b output will be delivered to the output register, which is available for the next macro.

III. Charge-Pump Based LIF Neuron

A. Area-Efficient Neuron Circuit

Although the bioactivities of neurons based on ion channels and membrane potentials are sophisticated, neurons based on the LIF model can be summarized into a few behaviors, integration, leakage, fire, and refractory period. To build an appropriate spiking neuronal circuit, a trade-off is required between hardware overhead and the degree of bio-fitting.

Fig. 6(a) shows the circuit structure of the proposed voltage-type charge-pump-based LIF neuron. Different from previous neurons using current-style integration by large capacitors, the proposed neuron first converts the summed currents into voltage pulses and then employs the charge pump for efficient voltage-style integration. The neuron circuit consists of a current-to-voltage conversion circuit (I2VCC), integration & recovery circuit (IRC), and spike generation & feedback circuit (SGFC). I2VCC samples the summed current from the synaptic array and converts it into a voltage pulse on V_{IN}. Then, the voltage pulse is applied on the bottom plate of capacitor C1 to cause the voltage of the pre-charged node V_{C1} to rise and transfer charges to capacitor C2 in an event-driven method. Thanks to the unidirectional conduction of transistor M3, C2 accumulates the incoming charge (“integration”) from different voltage pulses. The charge on C2 is also slowly leaking through the transistor M4 (“leakage”) and the leakage rate is controlled by the gate voltage of M4, V_{LEAK}. The membrane voltage on C2, i.e., V_{MEM}, is continuously compared with a reference voltage, i.e., threshold voltage V_{TH} by AMP in SGFC. After the output of AMP turns high, a rising edge will be transmitted to the clock input of the data flip-flop, and the output will become high. Then a spike will be generated when $V_{MEM} > V_{TH}$ (“fire”). After the retardation of the delay circuit, a pulse output signal will be generated, which will be fed back to the gate of M5 through the feedback loop. This high level will make M5 conduct, which forms a discharge path for circuit RESET. Then, the LIF neuron would enter the refractory period (“refractory”).

The waveform of the proposed voltage-type charge-pump-based LIF neuron circuit is shown in Fig. 6(b). The input currents that come from the synaptic array are square waves and differ in amplitude. The neuron works only when a current input is applied (“event”) and the neuron is not in a refractory period. Therefore, the neuron is event-driven. During the refractory period, the neuron is at rest and does not respond to external excitation, as demonstrated by the fact that the neuron membrane potential remains unchanged (shown in Fig. 6(b). The charge pump sends a portion of the charge at a time when the input comes, which enables highly efficient transfer of DC voltage at a small area cost. Considering the event-driven excitation mechanism, the charge pump is very helpful to realize the accumulation effect of neurons. In the initial state, the input voltage V_{IN} signal is 0, thus the diode-connected transistors M2 is turned on, and the voltage on C1 is $V_{C1}(t_1) = VCC - Vd2$, where *Vd1* refers to the turn-on voltage of diode-connected M2. Assume that at the moment t_1, the rising edge of V_{IN} arrives, then $V_{C1}(t_1) = V_{C1}(t_1) + VCC - Vd2$. Since V_{C1} is higher than VCC, M2 is turned off. Due to the existence of parasitic resistance, V_{C1} will slowly drop and discharge when V_{IN} is at a high level. Due to the presence of the refractory signal, the initial V_{MEM} before neuron excitation is 0, theoretically, so V_{C1} should be higher than V_{C2}. Thus, C1 continues to charge C2 until the diode-connected transistor M3 is shut down and $V_{C2}(t_2) = V_{C1}(t_2) - Vd3$. Assume that V_{IN} falls along the edge at the moment t_3, and $V_{C2}(t_{30-}) = V_{C1}(t_{30-})Vd2$. As the V_{IN} drops, VCC is higher than V_{C1} again, and VCC recharges C1 to $VCC – Vd2$.

Regardless of charge loss, the charge carried by capacitor C1 during each clock cycle is given by:

$$C1 \times \Delta V_{C1} = C2 \times \Delta V_{C2} \tag{2}$$

Therefore, the charge pump ratio(C2/C1) will affect the transfer speed and the charge volume from C1 to C2. Under the same input conditions, with a smaller C2 to C1 ratio, V_{C2} changes more and grows faster and tends to reach V_{TH} sooner, and therefore the neuron would be more likely to emit a pulse. The simulation results also confirm that the larger the charge pump ratio, the slower the transfer and the lower the neuronal

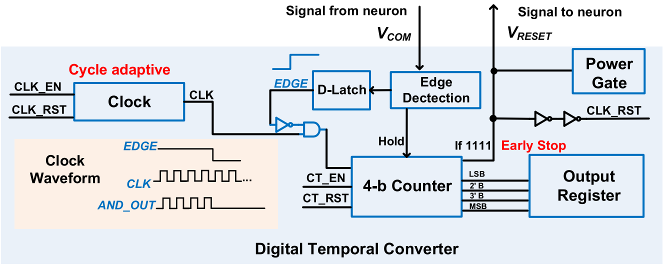


Fig. 7. Circuit structure of Digital Temporal Converter.

pulse issuing frequency (shown in Fig. 6(c)). To trade-off the ratio is set to be 4 in this work.

The use of the CtoV conversion circuit results in additional energy consumption compared to direct-charge-integration neurons. But, the large capacitors in the current-type circuits also dissipate a considerable amount of energy with frequent charging and discharging. In addition, some works have focused on optimizing neurons at the device level (e.g., RRAMs, PCMs, and FeFET), which tend to take up smaller areas due to using fewer devices. However, for system-level applications, the neurons based on emerging devices still rely on CMOS circuits to perform some essential operations, which calls for additional hardware overhead [48].

Here, we would like to discuss why the proposed charge-pump-based neuron features high area efficiency. It is known that the charging and discharging rate of a capacitor is often determined by its time constant. Thus, large capacitors are often required to prevent neurons from firing too frequently for conventional direct-current-integration neurons. In contrast, Dickson charge pumps exhibit switches arranged in the current charging/discharging pathway via unidirectional diodes. At each input cycle, the membrane potential is rapidly charged to a fraction of the voltage carried over by C1 and then discharged slowly until the rising edge of another input comes. The charge pump moves charge in and out of integrated capacitors with clocked switches and the switching frequency is the dominant determinant of the value of time constant. Thus, a large equivalent time constant can be obtained even with a small integrated capacitance. Compared with other works that restrict the transistors to work in weak-inversion regions to avoid using capacitors [17], [26], our above-threshold neuron design is more robust and compatible.

B. Processing of Neuron Outputs

We use a digital temporal converter to process the neuron output pulses for easier data transfer between macros. Considering that the amplitude of the neuron output is determined by the power, i.e., VDD, and the delay time is determined by the delay unit, the main information lies in the firing moment of pulse output. However, such a precise temporal signal is difficult to store or pass on temporarily. There is a simple way to facilitate data storage and transfer by converting the time domain signal into a digital or analog signal.

Fig. 7 shows the circuit of the digital temporal converter for signal conversion and early termination. The circuit receives the signal V_{COM} from the neuron circuit (shown in Fig. 6(a)). As mentioned above, V_{MEM} is the accumulation voltage of the synaptic inputs and represents the membrane voltage of the LIF neuron circuit. V_{COM} is the comparison result between V_{MEM} and V_{TH}. The rising edge of the neuron's output appears the moment when the V_{MEM} exceeds V_{TH}, i.e., the moment when $\mathrm{V_{COM}}$ jumps from 0 to VDD. Let this moment be donated as T_{FIRE}. Under unified clock modulation, the number of clock cycles passed before T_{FIRE} can be recorded by a multi-bit counter, so the temporal signal falls into one of the equally spaced finite time units on the time axis and is converted into a digital value.

Considering the case of low synaptic input strength of the neuron, it may result in a late output of the neuron or even not being issued within the effective time. Therefore, an ‘early termination’ mechanism is introduced. If the 4-b counter reaches the maximum value, i.e., ‘1111’, it can be considered that the neuron fails to generate a spike in this time window. Therefore, it is not necessary to wait for an output signal. A discharge signal V_{RESET} will be generated and connected to the feedback path of the neuron circuit. This signal will turn on the transistor M4 connected in parallel with C2, so as to open the discharge path and terminate the integration process of neurons in advance. Therefore, the ‘early termination’ mechanism saves the energy and the operation time of neurons.

In contrast, ReSiPE converts the neuron output from the temporal pulse into an analog signal for easy storage by using the temporal signal as the duration of the capacitor charging and discharging [49]. Therefore, the charged capacitor can temporarily store an analog value in the form of voltage. Considering that our design is based on a digital circuit, it is more area-saving. Moreover, the digital temporal converter is more reliable as there is no need to worry about capacitor leakage and the capacitance variation affected by PVT.

Although the exact timing information of neuron output is quantized to an integer value of 0-15 at the cost of quantization loss, the digital temporal converter circuit well matches the proposed linear latency coding. The neuronal outputs can be passed directly to interlayers after digital temporal conversion.

IV. Split-Training-Merged-Inference Algorithm

A. Basic Concept and Structure of STMI

To alleviate the spike-stall problem, we proposed the split-train-merged-inference (STMI) algorithm. The network structure is composed of a spike encoder, SNN layers, fully connected (FC) layers, spike neurons, and temporal voting output layers (as shown in Fig. 8(a)). Note that different network structures are employed separately for training and inference in STMI. For the training process, each main layer is accompanied by a split layer that shares the same size and the same input. The result of the split layer is subtracted with a dynamic scaling factor from that of the related main layer, and then the residual is delivered to the next layer. For the inference process, the weights of the main layers and corresponding split layers are merged through weighted subtraction, thus halving the on-chip storage capacity requirement. Fig. 8(b) summarizes the forward and backward propagation

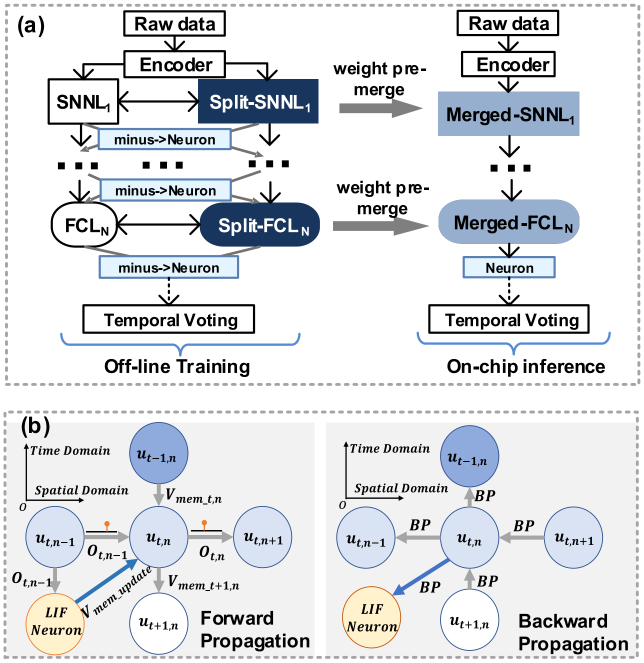


Fig. 8. (a) Network structure and components of the split- train-merged-inference (STMI) algorithm and (b) the forward and backward propagation paths of STMI.

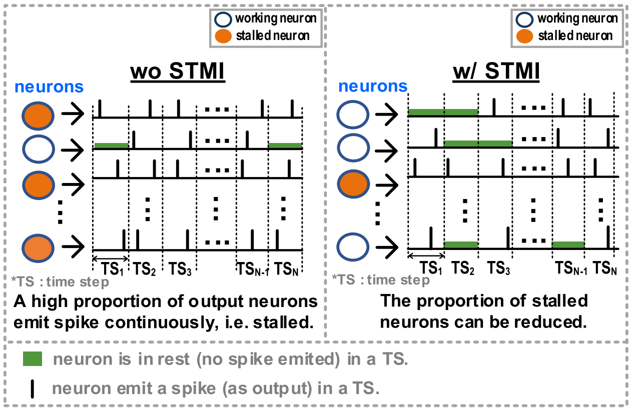

working neuron
stalled neuron
wo STMI
w/ STMI
neurons
TS1 TS2 TS3 TSN-1 TSN
*TS : time step
A high proportion of output neurons emit spike continuously, i.e. stalled.
The proportion of stalled neurons can be reduced.
neuron is in rest (no spike emited) in a TS.
neuron emit a spike (as output) in a TS.

Fig. 9. “Spike Stall” problem demonstration.

paths of the employed fully temporal direct train algorithm. The derivable square pulse enables the calculation of derivatives in the time domain and makes the direct train of SNN possible. The backward propagation path goes through both the temporal and the spatial domain, since the neuron state u at time t is determined by both its state at time $t-1$ and the synapse current from the last layer. The detailed mathematical description is illustrated in the following subsection.

In addition to reducing the hardware overhead in the inference process, the STMI method also alleviates the spike stall issue. As mentioned above, ‘Spiking Stall’ means some neurons keep constantly active and emit outputs consecutively at every time step, which implies that the neuron output is invalid for the next layer. As Fig. 9 shows, the neurons without STMI are more likely to emit spike continuously, i.e., more possible for entering the “stalled” state. In comparison, STMI allows the input excitation of the neuron to be dynamically reduced, thus partly limiting the proportion of neurons being stalled. Therefore, more effective temporal information can be delivered and a higher accuracy can be obtained through STMI.

B. Detailed Training and Inferring Methods of STMI

To perform the backpropagation in SNNs, a training algorithm utilizing both the spatial and temporal features of neurons was proposed and achieved high accuracy on several datasets [50]. As mentioned above, we proposed an advanced hardware-friendly training algorithm, named STMI, to enable the network to process single-spike signals and improve the recognition accuracy.

For a certain loss function $\boldsymbol{L}$, the gradient for weight and bias in the n-th layer can be obtained through (3):

$$\begin{aligned}\frac{\partial L}{\partial w_n} &= \sum_{t=1}^{T} \frac{\partial L}{\partial u_{t,n}} \frac{\partial u_{t,n}}{\partial w_n} = \sum_{t=1}^{T} \frac{\partial L}{\partial u_{t,n}} \frac{\partial u_{t,n}}{\partial I_0^{t,n}} \frac{\partial I_0^{t,n}}{\partial w_{t,n}} \\ &= \sum_{t=1}^{T} \frac{\partial L}{\partial u_{t,n}} o_{t,n-1}\end{aligned} \tag{3}$$

and

$$\frac{\partial L}{\partial b_n} = \sum_{t=1}^{T} \frac{\partial L}{\partial u_{t,n}} \frac{\partial u_{t,n}}{\partial b_n} = \sum_{t=1}^{T} \frac{\partial L}{\partial u_{t,n}} \tag{4}$$

where w and b denote the weight and bias, u the membrane potential of neurons, I_0 the current on the dendrite, and o the output signal. It can be observed that the results of (3) and (4) can be calculated simply through calculating $\partial L/\partial u_{t,n}$. Thus, we have:

$$\begin{aligned}\frac{\partial L}{\partial u_{t,n}} &= \frac{\partial L}{\partial o_{t,n}} \frac{\partial o_{t,n}}{\partial u_{t,n}} + \frac{\partial L}{\partial o_{t+1,n}} \frac{\partial o_{t+1,n}}{\partial u_{t,n}} \\ &= \frac{\partial L}{\partial o_{t,n}} \frac{\partial f}{\partial u_{t,n}} + \frac{\partial L}{\partial o_{t+1,n}} \frac{\partial f}{\partial u_{t,n}}\end{aligned} \tag{5}$$

$$\begin{aligned}\frac{\partial L}{\partial o_{t,n}} &= \sum_{i=0}^{l(n+1)} \frac{\partial L}{\partial o^j_{t,n+1}} \frac{\partial o^j_{t,n+1}}{\partial o_{t,n}} + \frac{\partial L}{\partial o^j_{t+1,n}} \frac{\partial o^j_{t+1,n}}{\partial o_{t,n}} \\ &= \sum_{i=0}^{l(n+1)} \frac{\partial L}{\partial o^j_{t,n+1}} \frac{\partial f}{\partial u_{t,n}} w_j + \frac{\partial L}{\partial o^j_{t+1,n}} \frac{\partial f}{\partial u_{t,n}} u_{t,n} \frac{\partial \mu}{\partial o_{t,n}}\end{aligned} \tag{6}$$

The subscript j denotes the j-th element in the layer, and $l(\cdot)$ denotes the length of one layer. The meaning of functions $f(\cdot)$ and $m(\cdot)$ was stated in the equations that described LIF functions.

Since single-spike SNNs are not as robust as those with multi-spikes, additional methods are required. The concept of STMI is illustrated in the above subsection, which improves network accuracy against disturbance. The detailed training and inference method of STMI is stated below. During training, for each neuron layer, we set a split layer with the same size. The split layer receives the same dendrite current as the main layer, and the current to the next layer will become:

$$I_{next} = \sum \left(w_{main}o - w_{split}o\right) + b_{main} + b_{split} \tag{7}$$

where subscript *main* and *split* denote the main layer and split one, respectively. STMI algorithm doubles the complexity

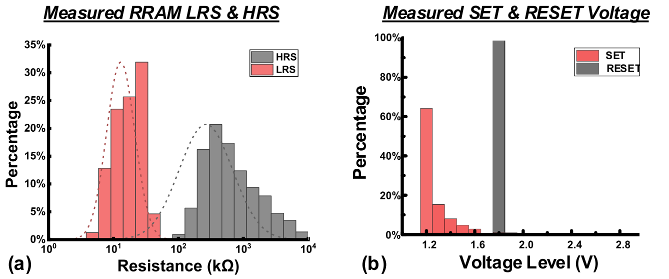


Fig. 10. (a) Measured results for LRS & HRS of RRAM; (b) measured results for SET & RESET success percentage of RRAM.

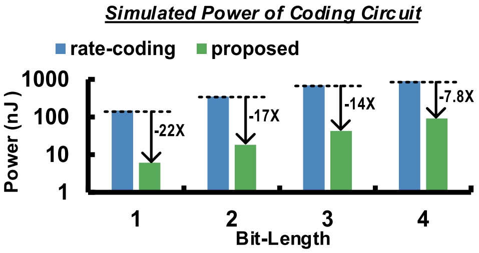


Fig. 11. Power saving for the proposed coding circuit under different input bit length.

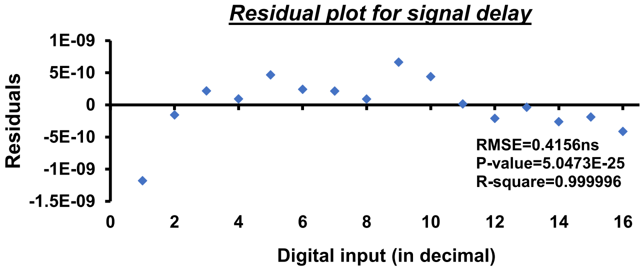


Fig. 12. Residual analysis of the single-spike latency coding circuit.

of the network when training, making the model with high learning ability. But for inferring, the weight kernels of the main and split layers are merged through subtraction, avoiding doubling the inference complexity.

The relationship between LIF neuron membrane potential, pre-synaptic current and output spike in software can be presented as:

$$u_{t+1} = u_t \mu(o_t) + I R_0 \tag{8}$$

$$o_t = f(u_t) \tag{9}$$

$$\mu(x) = \begin{cases} k_\tau, & x = 1 \\ 0, & x \neq 1 \end{cases} \tag{10}$$

$$f(x) = \begin{cases} 1, & x \geq V_{th} \\ 0, & x < V_{th} \end{cases} \tag{11}$$

$$I = \sum w o_{pre} + b \tag{12}$$

where k_t is the decay constant, R_0 is the unit resistance, V_{th} is the threshold voltage. I refers to the input current on the dendrite and $\boldsymbol{o_{pre}}$ refers to the output spike train from the last layer. Equation (8)-(11) describes the fundamental flow of network inference. We define a simulation time window T, and each pixel of the input image will fire a spike at the time given by the encoder. For the output layer, to decode the spike sequence, the output signal at time step t is defined as (13)

$$\boldsymbol{y_{N,t}} = \boldsymbol{o_N}(T - t). \tag{13}$$

The network output is obtained through (14)

$$\boldsymbol{y_N} = \sum_{t=1}^{T} \boldsymbol{y_{N,t}}. \tag{14}$$

In a time-based SNN, the recognition task is commonly performed by finding the neuron that spikes first in the output layer. Since each neuron would generate at most one spike in the time window, the position with the largest value in the output tensor spikes first.

V. Evaluation

A. Evaluation Results of Key Modules

We incorporate the architectural and algorithmic ingredients described above in an ASIC test chip implemented in a 40 nm CMOS process. The macro measures 0.084mm^2 in an area with a 16kb RRAM array.

The fabricated ReRAM test chip is measured to get real resistance and variation for RRAM devices. The write-verify method is employed for SET & RESET. The measured high resistance state (HRS) and low resistance state (LRS) of TiN/TiON/TaOx/TiN RRAM are 600 kΩ and 20 kΩ @25°C. The SET/RESET pulse width is fixed at 100 ns with a step of 0.001V. A SET voltage of 1.3V brings a >80% write success probability while a RESET voltage of 1.8V ensures a >98% erasing success probability (shown in Fig. 10).

To verify the power saving of the single-spike latency coding method, we extract the rating coding method from typical works [12], [37], [51] for 4-b inputs and take the average value. We post-simulated the layout of the encoders and carried out >200 Monte Carlo simulations to get the average value for power consumption. The simulated results are summarized in Fig.11. The single-spike latency coding significantly reduces the pulse density compared with rate coding, achieving >7X reduction in power consumption for 4-b input. For inputs of higher precision, we can divide them into several 4-bit or low-bit inputs and operate with the weight matrixes separately. Finally, the two partial products are shifted and summed to get the correct matrix-vector-multiplication (MVM) results.

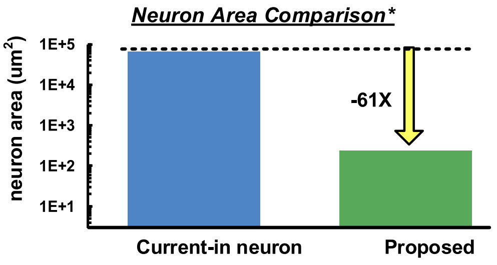
To evaluate the linearity of the single-spike latency coding method, the delay between the coding output and the reference signal is tested. We implemented extra test modules on the test chip for adjusting the bias voltages and testing the performance of the encoder. 16 4-b values from 0000 to 1111 are applied to the encoder test, respectively, and each value has been repeated 10 times to get the average value. The result is shown in Fig. 12. With residual analysis, the root mean square error of the evaluated latency is only 0.42ns. The P-value of the delay is less than 1E-20 and the difference between R square and 1 is less than 1E-05, which demonstrates good linearity.

TABLE I
NETWORK PERFORMANCE

Dataset	Network Type	Network Structure	Accuracy (%)	Add Ops	Multiply Ops
MNIST	SNN	8C3-8C3-128FC-10FC	97.92	1.3M	0
	CNN	8C3-RELU-8C3-RELU-128FC-10FC	98.50	1.3M	1.3M
Fashion-MNIST	SNN	8C3-8C3-128FC-10FC	89.14	1.3M	0
Olivetti	SNN	32C3-32C3-200FC-40FC	92.86%	16.M	0

TABLE II
NETWORK STRUCTURE FOR HARDWARE EVALUATION

Layer	Configuration	In-Channels	Output-Channels
SCNN-L1	Conv 3×3	1	8
SCNN-L1_Split	Conv 3×3	1	8
SCNN-L1	Conv 3×3	8	8
SCNN-L1_Split	Conv 3×3	8	8
FC-1	FC	6272	128
FC-1_Split	FC	6272	128
FC-2	FC	128	10
FC-2_Split	FC	128	10



*Simulated @ PW=100ns & amplitude=25uA

Fig. 13. Area saving for the proposed LIF neuron circuit.

For the proposed voltage-type charge pump-based LIF neuron, MOM capacitors are used for the critical charge pump circuit, where the transistor size of C1 is W/L=1.5um/1um while C2 uses four parallel transistors of W/L=1.5um/1um. To evaluate the area saving of the proposed voltage-type charge pump-based LIF neuron, we compare it with a direct-current-integration-type neuron that uses a single capacitor for current accumulation and membrane storage by post-layout simulation.

Given,

$$Q_C = C \times U \tag{15}$$

$$i = \frac{dq}{dt} \tag{16}$$

$$Q_I = \int i \times dt \tag{17}$$

The capacitance required for the accumulation of current-based neurons can be evaluated by (15)-(17) and the area consumed at the 40nm process can be analyzed, as summarized in Fig.13. The layout-based results illustrate that the proposed neuron circuit is 61 times smaller in area compared with the previous current-type one (with an on-off ratio equal to 4).

One of the most important metrics in neuron circuit design is the linearity between output and input. For our proposed neuron circuit based on latency coding, the linearity metric can be defined as the relationship between the output pulse frequency and the input current strength. That is, the higher the input current amplitude and input density, the shorter the output pulse interval and the faster the frequency of the neuron. To evaluate the linearity of the proposed neuron, the neuron input is set as a current square wave with a duty cycle of 50%, which has a fixed pulse width of 100ns with variable amplitude. We count the number of neuron output pulses in 5ms, to calculate the output frequency. The post-layout simulation results of the neuron output frequency statistics are summarized in Fig. 14 after 200 Monte Carlo simulations. The evaluated results show good linearity of the neuron output with the input current amplitude.

To evaluate the performance of the proposed STMI algorithm, fully temporal SNN models are constructed and verified on several image classification tasks (summarized in Table I). For MNIST dataset recognition, a SNN with two spike-based convolution layers with 8 kernels of 3 × 3 and fully-connected layers with 128 and 10 outputs (8C3-8C3-128FC-10FC) is trained and achieves an accuracy of 97.92%. A CNN with a similar network structure (8C3-RELU-8C3-RELU-128FC-10FC) can obtain an accuracy of 98.50% at the cost of double operation numbers. In addition, the SNN with the same structure (8C3-8C3-128FC-10FC) achieved 89.14% accuracy with the Fashion-MNIST dataset. Besides, another SNN model (32C3-32C3-200FC-40FC) based on the STMI achieves 92.86% accuracy on the Olivetti dataset (a small volume face recognition dataset). Thereby, it is seen that the increase in network size enables the performance of the proposed training algorithm for more complex training tasks. The above accuracies are achieved by simulation based on the NeuroSim platform [52]. In simulation, the count of time steps is set to be 10 in a time window and stalled neurons refers to those neurons whose working state (work/ rest) remains unchanged for 10 consecutive time steps. The sum of the number of stalled neurons in all the above networks is summarized in Fig. 17. By applying temporal coding and STMI, the number of stalled neurons that occurred is 10X less compared to that of rate-coding, and >14% less than that without STMI (shown in Fig. 15(a)). The recognition accuracy is also improved by 1.68% compared to that of rate coding, and by 0.8% compared to that without STMI (shown in Fig. 15(b)).

B. Performance Evaluation of SNN Accelerator

The microphotograph of the test chip is shown in Fig. 16 with key parts of the design highlighted and the chip summary is shown in Fig. 17. To verify and evaluate the performance of the proposed Tempo-CIM, we select the network for MNIST dataset from the above three pretrained SNN networks in Table I. Table II gives the network structure of training SNN

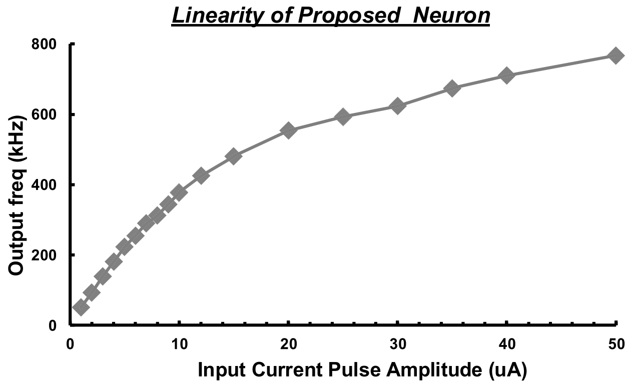


Fig. 14. Linearity analysis of the LIF neuron circuit.

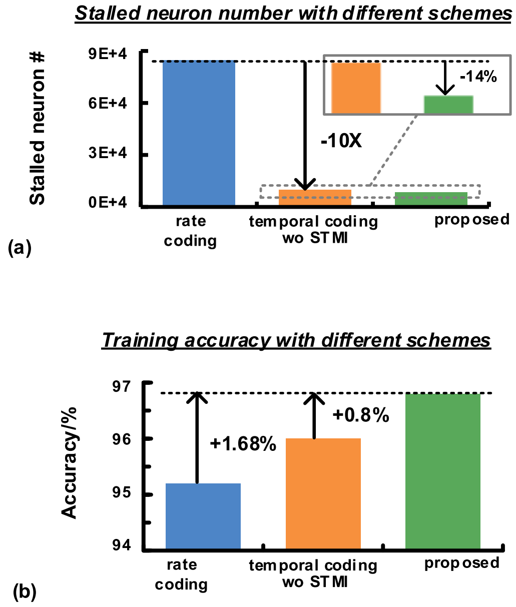


Fig. 15. Simulation results for spike stall alleviation. (a) Stall neuron number comparison with different coding method/ algorithm and (b) training accuracy comparison with different coding method/ algorithm.

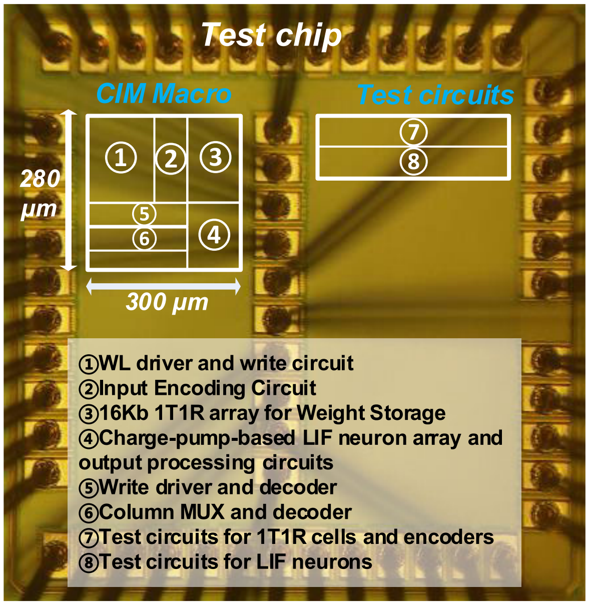

Test chip
CIM Macro
Test circuits
280 μm
300 μm
①
②
③
④
⑤
⑥
⑦
⑧
①WL driver and write circuit
②Input Encoding Circuit
③16Kb 1T1R array for Weight Storage
④Charge-pump-based LIF neuron array and output processing circuits
⑤Write driver and decoder
⑥Column MUX and decoder
⑦Test circuits for 1T1R cells and encoders
⑧Test circuits for LIF neurons

Fig. 16. Die microphotography.

Chip Summary		
Technology	40nm CMOS	
Device	Ta$_x$O RRAM	
Cell Type	1T1R	
Array Size	256 rows x 64 columns (16Kb)	
Memory Density	1.38 M/mm^2	
Macro Area	280μm x 300μm (0.084mm^2)	
Neuron Type	Voltage-type Charge-pump-based LIF neuron	
Power Supply	0.9V (Core); 2.5V (IO)	
Performance		
Throughput (GOPS)	4bIN-2bW	10.23
	4bIN-4bW	5.17
	4bIN-8bW	3.36
Energy Efficiency (TOP/W)	4bIN-2bW	330.4
	4bIN-4bW	224.8
	4bIN-8bW	117.9
Inference Accuarcy	4bIN-8bW	MNIST: 96.9%

Fig. 17. Chip summary.

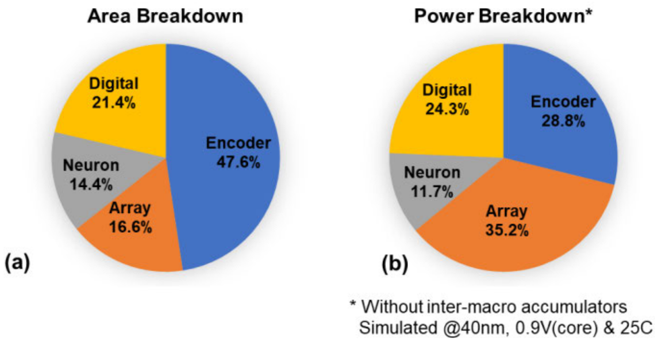


Fig. 18. (a) Area breakdown and (b) power breakdown of the 16Kb RRAM Tempo-CIM test chip (simulated).

algorithm for subsequent hardware evaluation. As previously stated, the network (8C3-8C3-128FC-10FC) achieved an accuracy of 97.72% by software simulation on the MNIST dataset. The training process is performed off-line to obtain weights, and the inference process is implemented on the test chip, with the weights merged and quantized into 4 bits, then pre-written into the 16Kb RRAM array.

Fig. 18 shows the area and power breakdown of the macro. As for area consumption, the 16Kb RRAM synaptic array only occupies less than 17%, whereas encoders take up almost 50% of the macro area. The largest area occupied by the coding circuit is attributed to the high number of coding circuits, to support high throughput in-memory operations. The digital module occupied about 20% area of the macro, including MUXs, drivers, decoders, and other necessary control circuits. Note that the proposed LIF neurons only take an area of 14%, which is affordable and are-efficient. The rest of the space contains routing and I/O (not included in the area comparison). As for power consumption, the RRAM array is dominating at 35%, and the encoder circuits use 29% of the power. Considering that there are 64 encoding circuits for

TABLE III

Comparison With State-of-the-Art Work

Metric	ISSCC'19[3]	ISSCC'23[4]	ISSCC'21[8]	ISSCC'22[53]	ISSCC'23[54]	JSSC'23[9]	This work
Technology	14nm CMOS	28nm CMOS	22nm CMOS +RRAM	22nm CMOS +RRAM	22nm CMOS +RRAM	28nm CMOS	40nm CMOS +RRAM
Architecture	Non-CIM	Non-CIM	CIM	CIM	CIM	CIM	CIM
Algorithm	SNN	CNN/SNN	CNN	CNN	CNN	SNN	SNN
Memory Capacity	4.8Kb	552Kb	4Mb (8 sub-bank)	8Mb (32 sub-bank)	4Mb (4 macro)	20Kb	16Kb
Neuron Type	Software	Digital	4-b Sense Amplifier	Cap-based Mixed-Signal	ADC based Mixed-Signal	Cap-based Mixed-Signal	Charge pump-based Analog Circuit
Supply Voltage (V)	0.8	0.7-1.1	0.8	0.8	0.7-0.8	1.1	0.9
Macro Aera (mm^2)	10.08	20.25	6(Test chip)	18(Test chip)	24.48	0.048	0.084
Precision	1bIN-4bW	4bIN-4bW	8bIN-8bW	8bIN-8bW	4bIN-4bW	4bIN-4bW	4bIN-8bW
Throughput	100k IMG/S	3279 GOPS	35.59 GOPS	14.4 GOPS	26.97 TOPS	N/A	3.36 GOPS
Energy Efficiency*[1] (TOPS/W)	3.68*[2]	31.02*[2]	3.60*[2]	18.69*[2]	73.14*[2]	60.6*[2]	117.9
Area Efficiency (TOPS/mm^2)	0.003	N/A	0.013	N/A	1.102	136.5	0.196
FoM*[3]	11.63	290.53	182.14	945.34	924.6	1454.6	3,772.80

*[1] 1 Op = 1 addition = 1 multiplication

*[2] Scaled to 40nm, assume energy$\propto$ (Tech.)^2

*[3] FoM=input precision × weight precision × reported energy efficiency (scaled to 40nm) × (supply voltage / 0.9V)^2

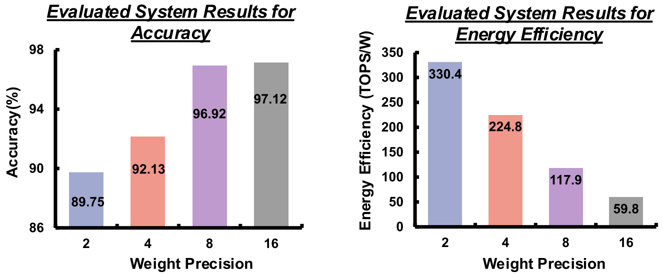


Fig. 19. (a) Evaluated system results for energy efficiency and (b) evaluated system results for accuracy.

efficient digital temporal conversion, the power consumption is actually low for one encoder. The digital modules consume 24% of the power, and the neurons occupied 12% of the power. The power consumption is evaluated by post-layout simulation.

To measure the power consumption during inference, part of the second spike-based convolution layer is implemented on the test chip for verification. Like most of the CIM-based studies have been done, the power consumed by I/O and extra test circuits has not been included. Thus, the reported power consumption of the test chip is the measured results for total power consumed in all on-chip digital logics and the Tempo-CIM macro (including power for accessing all weights, indexes, and activations). The test chip is limited in the number of input and output pads; therefore, the input image is scanned bit-by-bit into the input registers. After the scan is finished, the chip will operate with high throughput at its full speed.

For throughput, we calculated the maximum number of operations supported by Tempo-CIM in the CIM phase based on the operation frequency. The power consumption of the proposed CIM macro is obtained by measuring the average current at each power supply port during the CIM phase. For the evaluations of higher levels, we post-simulated the area and power consumption of digital control circuits and voting circuits. We also extract the hardware costs of buffers, buses, and activation units with the same evaluation method adopted in ISAAC [7]. Fig. 19 provides the results of the 40nm test chip of Tempo-CIM. The inference accuracy improves as the weight precision rises but at the cost of lower energy efficiency. The gain in accuracy becomes slower as the weight precision increases. To balance, a weight precision of 8b can provide better overall performance with a high energy efficiency of 117.9 TOPS/W. As the intermediate data is in voltage pulses rather than high-precision digital values, the area overheads for storage and peripheral circuits can be significantly reduced. Moreover, the event-driven mechanism of the proposed neuron also allows the circuit to consume less energy when the summed current from the synaptic array appears, further improving the energy efficiency.

Table III compares Tempo-CIM with state-of-the-art works. The throughput of our macro is inferior to that of the compared works because the memory capacity of our macro is limited. For a fair comparison, we scaled the energy efficiency of all the listed works to the 40nm process and used the figure of merit based on energy efficiency, input precision, weight precision and supply voltage. Thanks to the high-parallelism CIM architecture adopted, the proposed accelerator achieves an FoM improvement of at least 7X compared with the SNN accelerators based on traditional von Neumann architecture [3], [4]. For CNN-based CIM works, high-precision analog-to-digital converters (ADCs) or sense amplifiers (SAs) are often required to convert the analog MAC results obtained from the memory array into digital signals for subsequent high-accuracy digital operations. However, the use of high-precision ADCs involves tremendous area and energy consumption. Owing to the efficient LIF neuron circuit and the event-driven scheme of SNN, Tempo-CIM achieves at least 4X improvement in energy-efficiency-based FoM compared with cutting-edged CNN-based CIM works [53], [54]. Moreover, Tempo-CIM adopts the single spike coding method, which is more energy efficient in contrast with the rate coding method used in [9]. In comparison, the normalized energy efficiency is increased by 2.6X in contrast with the peak energy efficiency of the state-of-art SNN-based CIM work.

VI. Conclusion

In this work, we present a 40nm RRAM CIM macro (Tempo-CIM) with a charge-pump-based leaky-integrate-and-fire (LIF) neuron and split-train-merged-inference algorithm for efficient SNN acceleration. The single-spike latency coding reduces the number of pulses in each time step. It uses a delay chain structure to realize temporal mapping with high linearity. It achieves >7X reduction in power consumption for 4-b inputs compared with rated coding approaches. In addition, a voltage-type charge-pump-based LIF neuron is proposed to reduce the requirement for large capacitance remarkably. It saves 61 times the area in comparison with a previous current-type neuron circuit in evaluation. Next, the split-train-merged-inference algorithm is proposed to dynamically adjust the input of each neuron to prevent the occurrence of over-exciting, thus alleviating the spike stall problem. The evaluation result shows a >14% reduction in "stalled" neuron number to that without STMI and the recognition accuracy is also improved by 0.8%. The architectural and algorithmic ingredients are implemented in an ASIC test chip at 40 nm standard CMOS technology. The macro measures 0.084mm^2 with an energy efficiency of 117.9 TOPS/W and area efficiency of 0.1956 TOPS/mm^2 for 4b input and 8b weight, which achieves at least 2.6X improvement in normalized energy efficiency FoM in comparison with the state-of-the-art works.

References

[1] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 126–136, Jan. 2018.

[2] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, Nov. 2019.

[3] J. Park, J. Lee, and D. Jeon, "A 65 nm 236.5 nJ/classification neuromorphic processor with 7.5% energy overhead on-chip learning using direct spike-only feedback," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 140–142.

[4] S. Kim, S. Kim, S. Hong, S. Kim, D. Han, and H.-J. Yoo, "C-DNN: A 24.5–85.8 TOPS/W complementary-deep-neural-network processor with heterogeneous CNN/SNN core architecture and forward-gradient-based sparsity generation," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 334–336.

[5] D. Ielmini and H.-S.-P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, Jun. 2018.

[6] P. Chi et al., "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2016, pp. 27–39.

[7] A. Shafiee et al., "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2016, pp. 14–26.

[8] C.-X. Xue et al., "A 22 nm 4 Mb 8 b-precision ReRAM computing-in-memory macro with 11.91 to 195.7 TOPS/W for tiny AI edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 245–247.

[9] S. Kim, S. Kim, S. Um, S. Kim, K. Kim, and H. J. Yoo, "Neuro-CIM: ADC-less neuromorphic computing-in-memory processor with operation gating/stopping and digital-analog networks," *IEEE J. Solid-State Circuits*, vol. 58, no. 10, pp. 2931–2945, Oct. 2023.

[10] A. Agrawal, M. Ali, M. Koo, N. Rathi, A. Jaiswal, and K. Roy, "IMPULSE: A 65-nm digital compute-in-memory macro with fused weights and membrane potential for spike-based sequential learning tasks," *IEEE Solid-State Circuits Lett.*, vol. 4, pp. 137–140, 2021.

[11] Z. Zhao, Y. Wang, X. Zhang, X. Cui, and R. Huang, "An energy-efficient computing-in-memory neuromorphic system with on-chip training," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2019, pp. 1–4.

[12] J. Ding, Z. Yu, Y. Tian, and T. Huang, "Optimal ANN-SNN conversion for fast and accurate inference in deep spiking neural networks," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 2328–2336.

[13] L. Zhang, S. Zhou, T. Zhi, Z. Du, and Y. Chen, "TDSNN: From deep neural networks to deep spike neural networks with temporal-coding," in *Proc. 33rd AAAI Conf. Artif. Intell. 31st Innov. Appl. Artif. Intell. Conf. 9th AAAI Symp. Educ. Adv. Artif. Intell.*, Honolulu, HI, USA, 2019, p. 163.

[14] S. Park, S. Kim, B. Na, and S. Yoon, "T2FSNN: Deep spiking neural networks with time-to-first-spike coding," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.

[15] K. H. Long, J. D. Lieber, and S. J. Bensmaia, "Texture is encoded in precise temporal spiking patterns in primate somatosensory cortex," *Nature Commun.*, vol. 13, no. 1, p. 1311, Mar. 2022.

[16] G. K. Chen, R. Kumar, H. E. Sumbul, P. C. Knag, and R. K. Krishnamurthy, "A 4096-neuron 1M-synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 992–1002, Apr. 2019.

[17] G. Indiveri et al., "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, p. 73, Jan. 2011.

[18] M. Häusser, "The Hodgkin–Huxley theory of the action potential," *Nature Neurosci.*, vol. 3, no. S11, p. 1165, Nov. 2000.

[19] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, 1952.

[20] R. Fitzhugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophys. J.*, vol. 1, no. 6, pp. 445–466, 1961.

[21] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[22] S. Dutta, V. Kumar, A. Shukla, N. R. Mohapatra, and U. Ganguly, "Leaky integrate and fire neuron by charge-discharge dynamics in floating-body MOSFET," *Sci. Rep.*, vol. 7, no. 1, p. 8257, 2017.

[23] M. Chang et al., "A 73.53 TOPS/W 14.74 TOPS heterogeneous RRAM in-memory and SRAM near-memory SoC for hybrid frame and event-based target tracking," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 426–428.

[24] E. Painkras et al., "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013.

[25] P. A. Merolla et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[26] A. Rubino, C. Livanelioglu, N. Qiao, M. Payvand, and G. Indiveri, "Ultra-low-power FDSOI neural circuits for extreme-edge neuromorphic intelligence," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 1, pp. 45–56, Jan. 2021.

[27] N. Qiao et al., "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128 K synapses," *Frontiers Neurosci.*, vol. 9, p. 141, Apr. 2015.

[28] M. Karimi, A. S. Monir, R. Mohammadrezaee, and B. Vaisband, "CTT-based scalable neuromorphic architecture," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 13, no. 1, pp. 96–107, Mar. 2023.

[29] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, and E. Eleftheriou, "Stochastic phase-change neurons," *Nature Nanotechnol.*, vol. 11, no. 8, pp. 693–699, Aug. 2016.

[30] J. Lin and J.-S. Yuan, "Capacitor-less RRAM-based stochastic neuron for event-based unsupervised learning," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2017, pp. 1–4.

[31] J. Lin and J.-S. Yuan, "Analysis and simulation of capacitor-less ReRAM-based stochastic neurons for the in-memory spiking neural network," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 5, pp. 1004–1017, Oct. 2018.

[32] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnol.*, vol. 15, no. 7, pp. 529–544, Jul. 2020.

[33] J. K. Eshraghian et al., "Training spiking neural networks using lessons from deep learning," *Proc. IEEE*, vol. 111, no. 9, pp. 1016–1054, Sep. 2023.

[34] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: A Hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31, pp. 25–46, Jul. 2008.

[35] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing–based decisions," *Nature Neurosci.*, vol. 9, no. 3, pp. 420–428, Mar. 2006.

[36] F. Ponulak and A. Kasinski, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, Feb. 2010.

[37] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 54–66, May 2015.

[38] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[39] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Frontiers Neurosci.*, vol. 13, p. 95, Mar. 2019.

[40] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-YOLO: Spiking neural network for energy-efficient object detection," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 11270–11277.

[41] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers Neurosci.*, vol. 12, May 2018, p. 331.

[42] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers Neurosci.*, vol. 14, p. 119, Feb. 2020.

[43] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.

[44] H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. Li, "Going deeper with directly-trained larger spiking neural networks," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 12, pp. 11062–11070.

[45] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, Nov. 2019.

[46] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. 30th Int. Conf. Mach. Learn., Mach. Learn. Res.*, 2013, pp. 1310–1318.

[47] X. Xue, C. Zhao, H. Yang, J. Jiang, F. Tian, and Z. Zhang, "A charge-pump-based neuron circuit design," (in Chinese), Chin. Patent CN 110991628B, Apr. 18, 2023.

[48] T. Dalgaty et al., "Hybrid CMOS-RRAM neurons with intrinsic plasticity," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[49] Z. Li, B. Yan, and H. Li, "ReSiPE: ReRAM-based single-spiking processing-in-memory engine," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.

[50] Y. Wu et al., "Direct training for spiking neural networks: Faster, larger, better," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 1311–1318.

[51] T. Q. Nguyen, Q. T. Pham, P. C. Hoang, Q. H. Dang, D. M. Nguyen, and H. H. Nguyen, "An improved spiking network conversion for image classification," in *Proc. Int. Conf. Multimedia Anal. Pattern Recognit. (MAPR)*, Oct. 2021, pp. 1–6.

[52] X. Peng, S. Huang, H. Jiang, A. Lu, and S. Yu, "DNN+NeuroSim V2.0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 11, pp. 2306–2319, Nov. 2021.

[53] J.-M. Hung et al., "An 8-Mb DC-current-free binary-to-8b precision ReRAM nonvolatile computing-in-memory macro using time-space-readout with 1286.4–21.6 TOPS/W for edge-AI devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[54] W.-H. Huang et al., "A nonvolatile al-edge processor with 4MB SLC-MLC hybrid-mode ReRAM compute-in-memory macro and 51.4–251 TOPS/W," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 15–17.

Jingwen Jiang received the B.S. degree from Fudan University, Shanghai, China, in 2021, where she is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Chips and Systems. Her current research interests include computing-in-memory architecture and neuromorphic circuits and systems.

Keji Zhou (Member, IEEE) received the B.S. degree from the South China University of Technology, Guangzhou, China, in 2013, the M.E. degree from Ningbo University, Ningbo, China, in 2016, and the Ph.D. degree from Fudan University, Shanghai, China, in 2021. He is currently a Post-Doctoral Researcher with Fudan University. His current research interests include computing-in-memory, neuromorphic computing, and nonvolatile memory for neural networks.

Jinhao Liang received the B.S. degree from Fudan University, Shanghai, China, in 2022, where he is currently pursuing the Ph.D. degree with the Frontier Institute of Chip and System. His research interests include neuromorphic computing and RRAM-based CIM.

Fengshi Tian (Graduate Student Member, IEEE) received the B.Eng. degree in microelectronics science and engineering from Fudan University in 2021. He is currently pursuing the Ph.D. degree with the AI Chip Center for Emerging Smart System (ACCESS), The Hong Kong University of Science and Technology (HKUST). His current research interests include neuromorphic computing for edge applications, domain-specific architecture design, and computing-in-memory systems.

Chenyang Zhao received the B.S. degree from the North University of China, Taiyuan, China, in 2019. She is currently a Bachelor-Straight-to-Doctorate Student with the State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai, China. Her current research interests include the chip design of neural network accelerators based on computing-in-memory architecture.

Jianguo Yang (Member, IEEE) received the Ph.D. degree in microelectronics from Fudan University, Shanghai, China, in 2016.

In 2016, he joined the Department of Microelectronics, Fudan University, as a Post-Doctoral Research Fellow. In 2019, he joined the Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China, as an Associate Professor. His research interests include memory circuit design, hardware security, and new computing paradigm.

Xiaoyong Xue (Member, IEEE) received the Ph.D. degree in microelectronics from Fudan University, Shanghai, China, in 2011. He joined the Department of Microelectronics, Fudan University, as a Post-Doctoral Research Fellow. He is currently an Associate Professor with Fudan University. His research interests include high performance memory/storage, and in-memory computing circuit and systems.

Xiaoyang Zeng (Senior Member, IEEE) received the B.Sc. degree from Xiangtan University, Xiangtan, China, in 1996, and the Ph.D. degree (Hons.) from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2001. He is currently a Chair Professor and the Executing Director of the State Key Laboratory of Integrated Chips and Systems. His research interests include information security chips, baseband processing technologies for wireless communication, mixed-signal IC designs, and ultra-low power IC methodology.