Efficient and Robust Spike-Driven Deep Convolutional Neural Networks Based on NOR Flash Computing Array

Yachen Xiang, Peng Huang, *Member, IEEE*, Runze Han, Chu Li, Kunliang Wang, Xiaoyan Liu, and Jinfeng Kang, *Senior Member, IEEE*

***Abstract*— In this article, we propose an efficient and robust spike-driven convolutional neural network (SCNN) based on the NOR flash computing array (NFCA), which is mapped by the pretrained convolutional neural network with the same structure. The spike-driven system eliminates the additional analog-to-digital/digital-to-analog (AD/DA) conversion in the NFCA-based CNN. To study the performance of the hardware implementation, an NFCA-based SCNN for the recognition of the Mixed National Institute of Standards and Technology (MNIST) data set is simulated. Simulation results illustrate that the system achieves 97.94% accuracy with the computing speed of 1×10^6 frame per second (fps). Compared with the typical mixed-signal NFCA-based CNN, the NFCA-based SCNN saves 97% area and 56% energy consumption. Moreover, the NFCA-based SCNN demonstrates great robustness to 30% image noise with less than 2% accuracy loss. The impact of random telegraph noise (RTN) is also greatly reduced in which less than 1% accuracy decrease can be achieved at the 32-nm technology node.**

***Index Terms*— In-memory computing, nor flash memory, spike-driven convolutional neural network (SCNN).**

I. Introduction

DEEP convolutional neural network (CNN) has showcased the unprecedented computing power [1] and achieved beyond human-level accuracy in terms of speech recognition, image recognition, and machine translation [2]–[4]. However, it is still a great challenge to process the gigabyte-level data [5] in CNN efficiently owing to the "memory wall" of von Neumann architecture [6]. Hence, researchers are exploring strategies to improve the computing speed and the energy efficiency of the CNN based on von Neumann processors [7]–[9]. Among the possible solutions, the in-memory computing architecture [8], [9] based on nonvolatile memory (NVM) is promising because the vector–matrix multiplication (VMM), which accounts for more than 90% of the computation in the CNN [10], can be efficiently carried out in the NVM with the crossbar structure.

Manuscript received February 15, 2020; revised April 7, 2020; accepted April 8, 2020. Date of publication April 29, 2020; date of current version May 21, 2020. This work was supported by the National Key Research and Development under Grant 2019YFB2205102, Grant 2018YFA0701501, and Grant 2018YFE0203801. The review of this article was arranged by Editor C. Monzio Compagnoni. *(Corresponding authors: Peng Huang; Jinfeng Kang.)*

The authors are with the Institute of Microelectronics, Peking University, Beijing 100871, China (e-mail: phwang@pku.edu.cn; kangjf@pku.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TED.2020.2987439

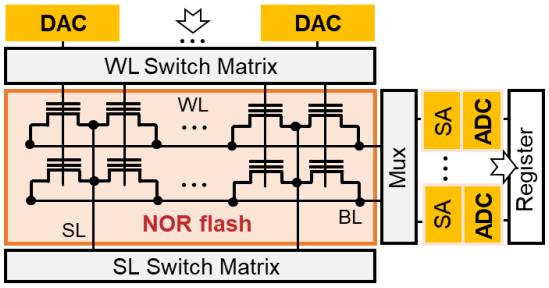

Fig. 1. Schematic of the mixed-signal NOR flash memory-based CNN.

The NOR flash array has a crossbar-like structure, which shows great potential for the implementation of the VMM. Compared with the novel NVM technologies [11]–[13] such as resistive random access memory (RRAM), the flash memory can be fabricated by the mature mass production technique [14], which promotes the large-scale integration with the peripheral CMOS circuitry. Moreover, the flash memory has an additional advantage over resistance switching memories in terms of the current gain because the floating gate (FG) transistor is the active component [15], [16]. The typical mixed-signal NOR flash memory-based CNN is shown in Fig. 1. The VMM is carried out by the NOR flash memory [15], [16]. The transconductance (gm) of FG transistors represents the matrix, and the input digital voltage represents the vector. The multiply and addition operations are realized using Ohm's law and Kirchhoff's current law, respectively. To convert the analog operation result of the previous layer to the binary input of the subsequent layer, analog-to-digital/digital-to-analog (AD/DA) converters are essential [17]. In this case, the complicated peripheral CMOS circuit with AD/DA converters dominates the hardware cost and the energy consumption (>85%) [18], which becomes one of the major concerns about improving the efficiency gains of the NOR flash memory-based CNN. For instance, the 8-bit AD conversion (ADC) in [19] occupies an area of 126.75 μm^2 at the 65-nm technology and the power consumption is 35 mW.

Recently, the spike neural network (SNN) has been proposed at the algorithm level [20], [21], in which the

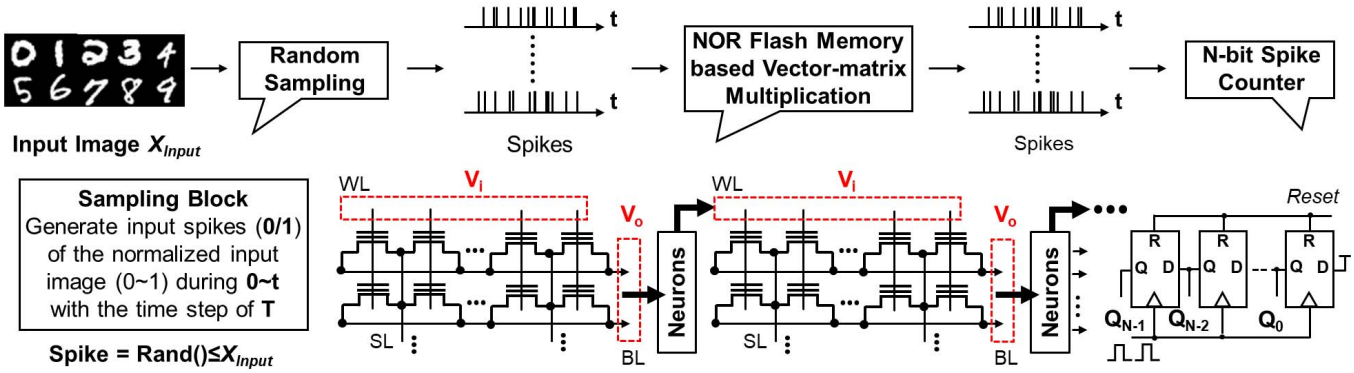

Fig. 2. Schematic of the NFCA-based SCNN.

information is coded using spikes (1/0). Accordingly, the AD/DA conversion in the NOR flash memory-based CNN is no longer needed. However, the training of the SNN is mainly achieved using the biology-like unsupervised learning rules such as spiking time/rate dependent plasticity (STDP/SRDP) [22]–[24], which makes it difficult to support complex practical cognitive applications. Different from these biological network, NOR flash computing array (NFCA)-based spike-driven CNN (SCNN) we proposed in this article is essentially a CNN trained with back-propagation (BP) algorithm [25]. The trained CNN is coded using spikes (1/0) instead of numerical values. In other words, the NOR flash memory-based SCNN is mapped by the CNN with the same structure. It promotes the scalability of the network and provides a promising hardware implementation of cognitive applications in large-scale neural networks.

This article is organized as follows. In Section II, we first introduce the principle of the NFCA-based SCNN. In Section III, we introduce the detailed implementation scheme of the hardware neural network. In Section IV, the solutions for addressing the capacitor discharge of the neuron are developed and the simulation results are shown. The evaluation method and the performance of the NFCA-based SCNN are provided in Section V. The conclusion is presented in Section VI.

II. Principle of NOR Flash-Based SCNN

Fig. 2 shows the working principle of the proposed NOR flash memory-based SCNN. When the system starts to work, the sampling block takes the samples of the normalized input image X_{Input} (0 ~1) using the random sampling method and generates input spikes. At time t_0, input spikes in the first layer are determined by

$$\text{Spike} = (\text{Rand}() \leq X_{\text{Input}}) \tag{1}$$

where Rand() generates random numbers with distribution on (0, 1). If the number generated by Rand() is smaller than the corresponding value of X_{input}, then the sampling block produces a spike (1); otherwise, no spike is produced (0). Therefore, the sampling block generates input spikes (0/1) of the input image at each sampling cycle (T). Fig. 3 exemplarily shows the results of first, second, third, and fourth samplings. Then the input spikes are applied to the word lines (WLs) of the NOR flash memory and the VMM operation is executed.

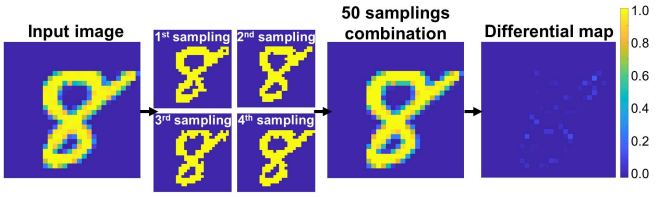

Fig. 3. Multiple samplings ($N = t/T$) within the duration ($0 \sim t$) guarantees the integrity of the image acquisition.

Each NOR flash cell multiplies the stored weight and the input spike. The accumulation is achieved by the bitline (BL) according to the Kirchhoff's current law. Therefore, the accumulated drain currents of different NOR flash cells in the same BL denote the operation results. Next, the neuron circuit integrates the current from the NOR flash memory. When the integrated voltage (V_{in}) exceeds the preset threshold voltage (V_{TH}), the neuron would generate a high-voltage level ("fire") and trigger the spike generation circuit to produce a spike to the subsequent layer. Then the neuron is reset to the initial state. If V_{in} is lower than V_{TH}, the neuron cannot be triggered and the voltage stored by the neuron is maintained at V_{in} until the next sampling ($t+T$, where T denotes the time step of the sampling). The working principle of the neuron is expressed as

$$V_{\text{in}}(t+T) = \begin{cases} 0, & \text{for} V_{\text{in}} \geq V_{\text{TH}} \\ V_{\text{in}}(t) + V_{\text{in}}(t \sim t+T), & \text{for} V_{\text{in}} < V_{\text{TH}}. \end{cases} \tag{2}$$

In the SCNN, the output of each layer is binary and in the form of spikes (0/1). Therefore, the AD/DA conversion in the NOR flash memory-based CNN (Fig. 1) is eliminated and replaced by the neuron integration. Note that the rectified linear unit (ReLU: max (0, x)) is adopted as the activation function in this article, mainly for the reason that the output of ReLU in the CNN is equivalent to the number of spikes produced by the neuron in the SCNN. It enables the neuron to perform the function of ReLU and simplifies the hardware implementation. In order to guarantee the integrity of the image acquisition and enhance the performance of the network, multiple samplings ($N = t/T$) are essential and the sampling–operation–integrate procedure would be repeated within the duration ($0 \sim t$). As shown in Fig. 3, 50 samplings of the input image are combined and normalized. The differential map is defined as the difference between the input image and the combination of 50 samplings. The near-zero

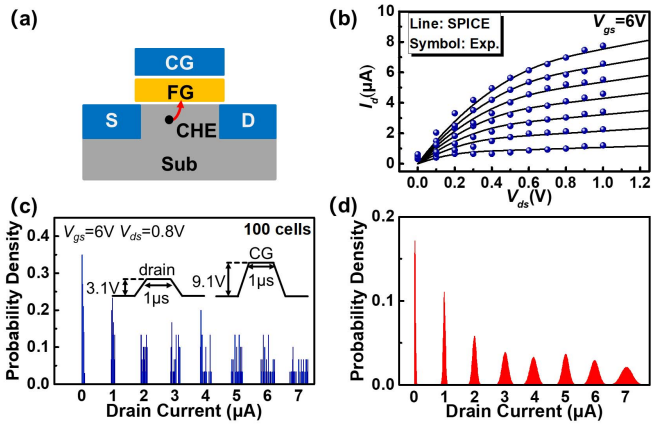

Fig. 4. (a) Schematic of the floating-gate transistor. (b) Measured transfer characteristics and the calibrated SPICE model of different states. (c) Measured drain current distribution of multilevel (3-bit) states. (d) Extracted distribution from the measured data in (c).

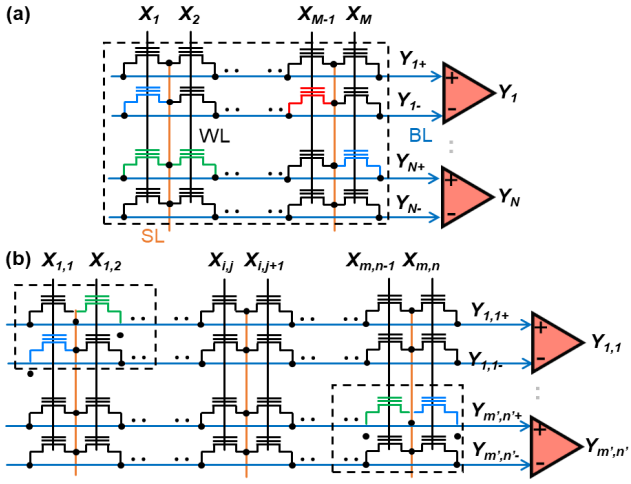

Fig. 5. Schematic of the weight mapping principle of the NFCA for (a) fully connected and (b) convolution operations.

differential map proves the validity of the random sampling method. N-bit shift registers (spike counters) [26] in the output layer count the number of spikes of each output neuron. The maximum output is regarded as the recognition result of the NOR flash memory-based SCNN.

III. Hardware Implementation

A. NFCA

Fig. 4(a) shows the schematic of the FG transistor, and the threshold voltage (V_{th}) of the NOR flash memory cell can be tuned by tunneling erase and hot-electron injection write [14]. It is possible to write and store multilevel values in the FG by controlling the number of the trapped electrons [27]–[29]. In this article, the value of the weight stored by the NOR flash memory cell is defined as the ratio of the drain current (I_d) to the reference current ($I_{\text{ref}} = 1\ \mu\text{A}$) at $V_{\text{gs}} = 6$ V and $V_{\text{ds}} = 0.8$ V. Different from the definition of weight as reported in [30] and [31], in which the weight of a memory transistor operated as a synaptic transistor is related to the subthreshold operation of the device, the FG transistors in this article work in the linear or saturation region. The experiments are performed to verify the multilevel storage characteristic within a fully processed 128-Mb NOR flash memory which is fabricated in the 65-nm technology node [32]. Although the mainstream incremental step pulse programming (ISPP) algorithm [33] enables place cell V_{th} within a certain range, there is a high probability of over programing when the interval between adjacent states is small (e.g., 1 μA). In this case, the over-programed cells should be erased and reprogrammed. However, the structure of NOR flash memory determines that the cells of the same block would be erased at the same time. To achieve efficient and precise programming of multibit NOR flash, the fixed pulse programming method with lower V_g is adopted in this article. During the programming, the source is connected to the ground, while the control gate (CG) and the drain are applied with 9.1 V (1 μs) and 3.1 V (1 μs) voltage pulses, respectively. The program-and-verify algorithm [34] is adopted. Fig. 4(b) shows the measured transfer characteristics of different states. The measured data demonstrate that the NOR flash cell can store multilevel values (3-bit) through multiple erase and write operations, which is fit well with the calibrated BSIM 3v3 model [35] in SPICE. The measured I_d distribution of 3-bit states (100 cells) is shown in Fig. 4(c). Despite device-to-device variations, the tight I_d distribution of different states can also be achieved experimentally. The mean (μ) and the standard deviation (σ) of the measured data are extracted and the extracted distribution from the measured data [Fig. 4(d)] is applied to the simulation.

NFCAs with multilevel storage characteristics are adopted to carry out the VMM including the convolution and the fully connected operations. The weight mapping principle is shown in Fig. 5. The weight matrix splits into two matrices which represent positive values (mapped to odd bit lines) and negative values (mapped to even bit lines), respectively. In other words, the differential pairs are adopted to store the weights. If we assume that G_+ and G_- denote the conductance of the two flash cells, the value w stored by the pair can be described as $w = k * (G_+ - G_-)$, where k denotes the coefficient of weight mapping from the algorithm to the NOR flash memory. When the spikes ($0/V_{\text{gs}}$) that represent the input vector are applied to WLs, the differences of currents between adjacent odd and even BLs denote the operation result. The key to achieving different operations is the programming region (black dotted box in Fig. 5). Specifically, the weight matrix ($M \times N$) is directly mapped to the NFCA with the size of $M \times N \times 2$ for the fully connected operation [Fig. 5(a)]. However, since the kernel matrix F with size of s is convoluted with the submatrix ($s \times s$) of the input matrix X ($m \times n$), only $s \times s$ NOR flash memory cells of every two BLs are mapped according to the mapping rule illustrated in [15]. The parallel convolution is achieved by setting up redundant NOR flash memory cells with the redundancy rate of $(m \times n - s \times s)/m \times n$.

B. Neuron

The schematic of the neuron is shown in Fig. 6. The analog part of the neuron is implemented by the integration circuit, which integrates the result of the VMM (V_{vmm}) from the NFCA. The output of this accumulated integral V_{in} at time t_2 is closely related to V_{in} (t_1) ($T = t_2 - t_1$), expressed as

$$V_{\text{in}}(t_2) = -1/RC \cdot \int_{t_1}^{t_2} V_{\text{vmm}} dt + V_{\text{in}}(t_1). \tag{3}$$

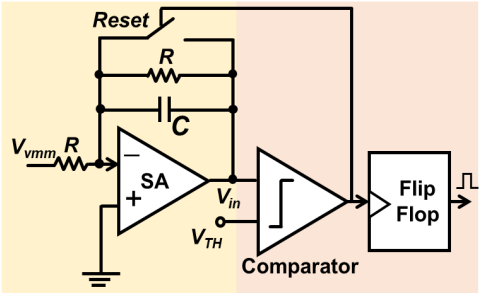

Fig. 6. Schematic of the neuron circuit.

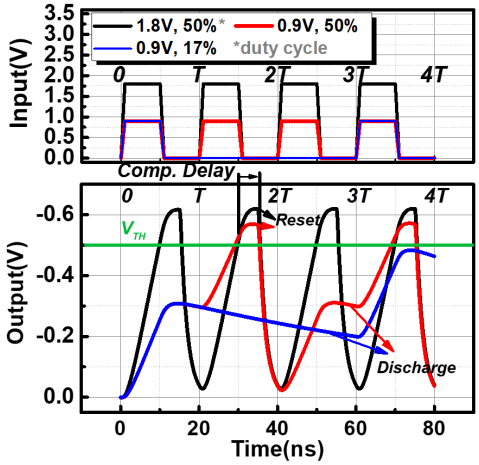

Fig. 7. Simulation results of the neuron circuit. When V_{in} exceeds V_{TH}, the neuron fires and then is reset after 2.5 ns.

The digital part of the neuron is made of a dynamic latch comparator and a D flip-flop. Whenever V_{in} exceeds the fixed threshold (V_{TH}), the output of the comparator would be set to a high-voltage level and the positive edge triggers the flip-flop to produce a spike. At the same time, the neuron is reset to the initial state by the feedback signal from the comparator. All the neurons are reset before processing a newer input image.

The speed of the integration circuit is determined by the time constant τ ($\tau = RC$). A large τ is usually required to coordinate with the frequency response of other components. Considering that large capacitors are very expensive in terms of area [36], R and C are set as 250 kΩ and 1 pF, respectively. The delay of the dynamic latch comparator is 2.5 ns in our design. The duration of the output spike of neurons is 10 ns. The sampling frequency is set as 50 MHz ($T = 20$ ns, duty cycle: 50%) so that there is enough time for the computing in the NFCA and the postprocessing in the neuron. The simulation results of the neuron circuit are shown in Fig. 7. When V_{in} exceeds V_{TH}, the neuron is reset after the comparator outputs the result (2.5 ns). However, if V_{in} is lower than V_{TH}, it remains an issue that the voltage stored by the neuron cannot be maintained owing to the discharge of the capacitor. The details of the problem and the solutions will be shown in Section IV.

IV. Design Exploration

The network used in the evaluation is LeNet-5 for the Mixed National Institute of Standards and Technology (MNIST) recognition [37], as shown in Table I.

TABLE I
HARDWARE IMPLEMENTATION OF LENET-5 (SCNN)

Layer	Dimension	Hardware
Input	<28×28>	Sampling Block
Conv. 1	<5×5> kernel → 6 <24×24>	6 NFCAs + Neurons
Pool 1	<2×2> Avg. → 6 <12×12>	6 NFCAs + Neurons
Conv. 2	<5×5> kernel → 12 <8×8>	12 NFCAs + Neurons
Pool 2	<2×2> Avg. → 12 <4×4>	12 NFCAs + Neurons
FC	192×10	1 NFCA + Neurons
Output	10	N-bit Spike Counter

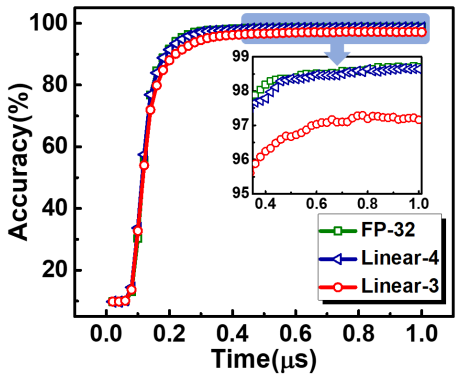

Fig. 8. Recognition accuracy of the five-layer NFCA-based SCNN with different quantization levels. The sampling frequency and the duration are set as 50 MHz and 1 μs, respectively.

A. Weight Quantization

On the one hand, limited by the resolution of different storage states coming from the program algorithm [34] and the physical limit of the electron number in the FG [38], each NOR flash cell can only store finite values ($0 \sim N$). On the other hand, low precision weights are sufficient for inference applications. For example, 4b-log quantized AlexNet shows only marginal accuracy degradation compared with 32-bit FP [39]. Therefore, the trained weights of LeNet-5 (32-bit FP) are quantized and mapped into the NFCAs. Here, the impact of the weight quantization precision on the accuracy of the network is evaluated at the algorithm level. Simulation results in Fig. 8 show that the recognition accuracy of the network with 3-bit linear-quantized weights is comparable with the FP-32 network (1.15% accuracy loss at 1 μs). To reduce the hardware cost and the energy consumption, the weight quantization precision is set as 3-bit in this article because the multilevel storage characteristic of the NOR flash memory fulfills the requirement of the SCNN exactly.

B. Neuron Discharge and Solutions

As illustrated in Section II, if V_{in} is lower than V_{TH}, V_{in} at t_1 is maintained until t_2 ($T = t_2 - t_1$) theoretically. Actually, V_{in} of the proposed neuron circuit would change because of the discharge of the capacitor, expressed as

$$V_{in}(t_2) = V_{in}(t_1) \cdot e^{-T/RC}. \quad (4)$$

The impact of the discharge problem on the neuron is shown in Fig. 7. When a pulse with an amplitude of 0.9 V (duty cycle: 50%) is continuously applied to the input of the neuron (0–80 ns, red line), V_{in} would exceed V_{TH}

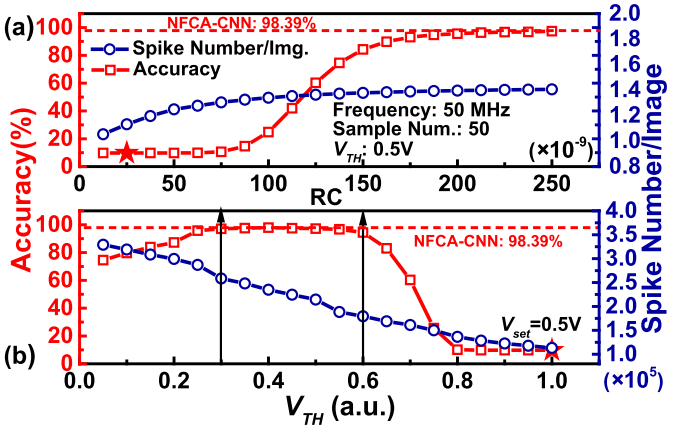

Fig. 9. Solutions for the discharge problem of the neuron circuit. (a) Increasing *RC*. (b) Decreasing V_{TH}. The red baseline denotes the accuracy of the NFCA-based CNN.

(V_{set} = 0.5 V) at t = 30 ns/70 ns and trigger the flip-flop to generate a spike after the comparator delay (2.5 ns). In that case, the function of the neuron is not affected by the discharge of the capacitor. However, if the time interval between adjacent voltage pulses is 40 ns (blue line), the spike cannot be generated at t = 70 ns. The discharge problem is destructive for the NFCA-based SCNN, as shown in Fig. 9.

From (4) and Fig. 9(a), we find that increasing *RC* can weaken the discharge of the capacitor and increase the accuracy of the network. However, the number of spikes generated in the inference is increasing gradually, leading to additional energy consumption. Moreover, larger *RC* means longer integration time and increasing area overhead induced by the capacitance. Hence, there is a limit to the increase of *RC*. Another solution is to compensate for the discharge by decreasing the threshold (V_{TH}). As shown in Fig. 9(b), the approach performs well without introducing additional area consumption and time delay. As V_{TH} shrinks, the number of spikes together with the energy consumption required for the image recognition task increases. The tradeoff between the accuracy and the energy is needed in the wide range (0.3 V_{set} ∼ 0.6 V_{set}). By decreasing V_{TH}, the NFCA-based SCNN achieves 97.94% recognition accuracy for the MNIST test set, which is comparable with the accuracy of the NFCA-based CNN.

V. Performance Evaluation

A. Area and Energy

LeNet-5 for the MNIST recognition is served to evaluate the performance of the NFCA-based SCNN and the mixed-signal NFCA-based CNN. The computing units and peripheral circuits are considered. For a reasonable comparison, we adopt the 4-bit ADC in the CNN to achieve the AD conversion between neighboring layers. The 4-bit ADC adopts the design of [40] with 2-mW power consumption, 105 × 110 μm^2 active area, 1.2 GS/s sampling rate at the 65-nm technology node. The sense amplifier (SA) in [19] is also used in the evaluation, in which the power and the area are 0.25 mW and 244 T ($T = W/L \times F^2$, F: technology node), respectively. The details of other components are depicted in Section III. Specifically,

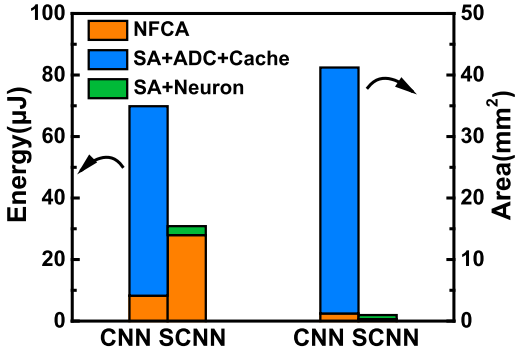

Fig. 10. Area and energy distribution of the NFCA-based CNN/SCNN.

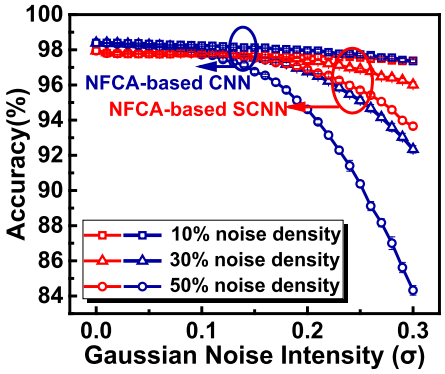

Fig. 11. Recognition accuracy as a function of Gaussian noise intensity (σ) with noise occurrence probability of 10%, 30%, and 50%.

the MIM capacitance density is about 10 to 30 fF/μm^2 [27]. In the worst case, the area of the 1 pF capacitor is 100 μm^2.

The energy and area distribution of the NFCA-based CNN/SCNN for the MNIST recognition is shown in Fig. 10. It is found that most of the energy and area are dissipated on ADCs in the CNN. As the output of each layer in the SCNN is in the form of spikes (binary: 0/1), the number of NFCAs is determined by the number of kernels of each layer, and the size of the NFCA is reduced by four times compared with the CNN. More importantly, the significant overhead induced by ADCs in the CNN is reduced greatly, although multiple samplings increase the energy consumption in terms of the NFCA in the SCNN.

B. Noise Tolerance

The image noise and the random telegraph noise (RTN) are considered in the evaluation. We adopt the Gaussian distributed noise to analyze the impact of the image noise on the proposed hardware system, in which the noise intensity and density are defined as the standard deviation (σ) of the Gaussian noise and the noise occurrence probability, respectively. In the simulation, the Gaussian distributed noise ($0, \sigma$) is generated randomly and added to the original image. According to the simulation results in Fig. 11, the NFCA-based SCNN shows great robustness to the image noise of 30% noise density (σ = 0.3) with less than 2% accuracy loss. Even if the noise density (σ = 0.3) reaches 50%, the recognition accuracy is still above 93%, while the 15% accuracy decrease is observed in the NFCA-based CNN.

As the dimension of the flash memory device continuously shrinks, the increasing V_{th} fluctuation (ΔV_{th}) induced

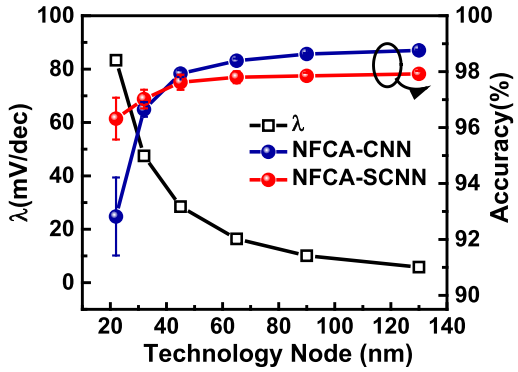

Fig. 12. Accuracy of the NFCA-based CNN/SCNN at different technology nodes.

by RTN [41] is becoming one of the major issues to be considered in the NFCA-based SCNN which requires severe V_{th} control. The origin of RTN is attributed to the electron trapping/detrapping near the substrate (Si)/oxide (SiO_2) interface [42]. The change of V_{th} of a cell due to RTN depends on the statistics of the amplitude of RTN fluctuations and the probability that a cell is affected by the change of V_{th} within a certain stretch of time [43]. In this article, we focus on the study of the impact of the statistics behavior of RTN on the performance of the NFCA-based CNN and SCNN. The statistical cumulative distribution of $|\Delta V_{\text{th}}|$ in [44] shows a clear exponential behavior, and its slope (λ, unit: mV/dec) in the semilogarithmic scale is the critical parameter describing RTN in the flash memory, which is expressed as

$$\lambda = \frac{d|\Delta V_{\text{th}}|}{d\lg(p)} = \frac{1}{\upsilon} \tag{5}$$

$$p = 1 - F(|\Delta V_{\text{th}}|) = 10^{-|\Delta V_{\text{th}}|\cdot\upsilon} \tag{6}$$

where F is the cumulative distribution of $|\Delta V_{\text{th}}|$. The dependence of λ on the technology ranging from 130 to 22 nm is shown in Fig. 12. The numerical value of λ at the 32 nm technology is calculated according to the data in [45]. The values of λ at other technology nodes are derived according to the model in [44]. It is found that λ ($|\Delta V_{\text{th}}|$) increases with the technology scaling. The V_{th} distribution of different technology nodes is extracted according to λ and added to the simulation of the network. Fig. 12 shows the MNIST recognition accuracy of the CNN/SCNN at different technology. The tolerance of RTN is effectively enhanced in the SCNN. The accuracy loss is less than 1% at the 32 nm technology.

In summary, simulation results indicate that the NFCA-based SCNN shows greater tolerance to the random noise, which is attributed to the design principle described in Section II.

1) For the NFCA-based CNN, the image noise or RTN directly affects the output of each layer. The computing errors are accumulated layer by layer and then fed into the linear classifier, leading to the misjudgment.
2) In the NFCA-based SCNN, the output of each layer is determined by the relative value of the integrated voltage V_o and V_{TH}, which is in the form of spikes (0/1). As long as the sign of $(V_o - V_{\text{TH}})$ is not changed, tiny computing errors have little impact on the output result, thus the accuracy of the SCNN can be maintained to the utmost.

TABLE II
COMPARISON BETWEEN NFCA-BASED CNN AND SCNN

	CNN	SCNN
Technology (nm)	65	65
Frequency (MHz)	104	50
Input/WL	Digit	Spike
Accuracy/MNIST (%)	98.39	97.94
Array Size (Mb)	27.69	6.92
Area (mm^2)	41.21	0.98
Time Delay/Image	385ns	1.1μs
Energy/Image (μJ)	69.84	30.85
Noise Tolerance	Low	High

C. Summary

The comparison between the NFCA-based CNN and SCNN is summarized in Table II. Regarding the MNIST recognition, SCNN achieves 42.1 times area and 2.3 times energy savings at the cost of negligible accuracy loss (0.45%). The synchronous clock is adopted in the design of SCNN. Therefore, the delay of SCNN is determined by sampling frequency (T_1), number of samplings (N_1) for each input image, and network scale (N): $\text{Delay}_1 = (N + N_1) * T_1$. Similarly, the delay of CNN is expressed as $\text{Delay}_2 = N * (1 + N_{\text{ADC}}) * T_2$, where T_2 is the clock cycle of CNN and N_{ADC} denotes the number of clock cycles required by ADCs for AD conversion. Note that the speed of SCNN is worse compared with CNN, which is partly induced by multiple samplings. More importantly, the estimated delay of CNN assumes that summed currents along different BLs are processed by ADCs in parallel. However, limited by the area and power consumption of ADC [17], it is impossible to arrange thousands of ADCs on the single chip. Therefore, the time-division multiplexing of ADCs in CNN is essential, which would lead to an increase in delay. On the other hand, according to the simulation results in Fig. 8 (Linear-3: $t = 0.6$ μs, 97.08% accuracy; $t = 1$ μs, 97.16% accuracy), the accuracy becomes stable (almost unchanged) when $t > 0.6$ μs. Therefore, the evaluation of the time delay of SCNN ($T = 20$ ns, 50 samplings, layers, delay: 1.1 μs) is conservative. By reducing the number of samplings on the premise of ensuring the recognition accuracy, and redesigning the neuron circuit using more advanced CMOS process to reduce RC constant and integration delay, the speed of SCNN can be further improved. Furthermore, for large-scale neural networks, the 4-bit ADC for the evaluation of the NFCA-based CNN is far from enough for complex cognitive benchmarks such as CIFAR-10 or ImageNet. ADCs with higher precision mean higher N_{ADC}, leading to the increase of Delay_2. In this case, the speed of NFCA-based SCNN may even be an advantage for applications with large-scale neural networks.

VI. CONCLUSION

In this article, the SCNN with high energy/area efficiency, great tolerance for image noise, and RTN is proposed based on the NFCA for the first time. The overhead of peripheral circuits is effectively minimized and the AD/DA converters needed in the typical mixed-signal NFCA-based CNN are eliminated. Regarding the MNIST recognition, the proposed hardware

implementation achieves 97% area and 56% energy savings compared with the NFCA-based CNN. More importantly, the great scalability promotes the applications of the NFCA-based SCNN in large-scale neural networks, which is promising for complex cognitive tasks in artificial intelligence (AI).

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[2] C.-C. Chiu *et al.*, "State-of-the-Art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.

[3] B. Moons and M. Verhelst, "A 0.3–2.6 TOPS/W precision-scalable processor for real-time large-scale ConvNets," in *Proc. IEEE Symp. VLSI Circuits (VLSI-Circuits)*, Jun. 2016, pp. 1–2, doi: 10.1109/VLSIC.2016.7573525.

[4] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*. [Online]. Available: http://arxiv.org/abs/1609.08144

[5] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient machine learning for big data: A review," *Big Data Res.*, vol. 2, no. 3, pp. 87–93, Sep. 2015.

[6] M. M. Waldrop, "More than Moore," *Nature*, vol. 530, no. 7589, pp. 144–147, Feb. 2016.

[7] P. Huang *et al.*, "Compact model of HfO_X-based electronic synaptic devices for neuromorphic computing," *IEEE Trans. Electron Devices*, vol. 64, no. 2, pp. 614–621, Feb. 2017, doi: 10.1109/TED.2016.2643162.

[8] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature Nanotechnol.*, vol. 8, no. 1, pp. 13–24, Jan. 2013.

[9] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proc. IEEE*, vol. 106, no. 2, pp. 260–285, Feb. 2018.

[10] J. Emer, "Design efficient deep learning accelerator: Challenges and opportunities," in *Proc. VLSI Short Course*, 2017, pp. 1–121.

[11] J. F. Kang *et al.*, "Oxide-based RRAM: Requirements and challenges of modeling and simulation," in *IEDM Tech. Dig.*, Dec. 2015, pp. 5.4.1–5.4.4, doi: 10.1109/IEDM.2015.7409634.

[12] G. W. Burr *et al.*, "Experimental demonstration and tolerancing of a large-scale neural network (165000 synapses) using phase-change memory as the synaptic weight element," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498–3507, Nov. 2015.

[13] S.-W. Chung *et al.*, "4Gbit density STT-MRAM using perpendicular MTJ realized with compact cell structure," in *IEDM Tech. Dig.*, Dec. 2016, p. 27, doi: 10.1109/IEDM.2016.7838490.

[14] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," *Proc. IEEE*, vol. 91, no. 4, pp. 489–502, Apr. 2003, doi: 10.1109/JPROC.2003.811702.

[15] R. Han *et al.*, "A novel convolution computing paradigm based on NOR flash array with high computing speed and energy efficiency," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 5, pp. 1692–1703, May 2019, doi: 10.1109/TCSI.2018.2885574.

[16] X. Guo *et al.*, "Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology," in *IEDM Tech. Dig.*, Dec. 2017, pp. 6.5.1–6.5.4.

[17] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *IEDM Tech. Dig.*, Dec. 2017. pp. 6.1.1–6.1.4.

[18] B. Li, L. Xia, P. Gu, Y. Wang, and H. Yang, "Merging the interface: Power, area and accuracy co-optimization for RRAM crossbar-based mixed-signal computing system," in *Proc. 52nd Annu. Design Autom. Conf. (DAC)*, 2015, pp. 13–18.

[19] T. Tang, L. Xia, B. Li, Y. Wang, and H. Yang, "Binary convolutional neural network on RRAM," in *Proc. 22nd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 782–787.

[20] 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, doi: 10.1007/s11263-014-0788-3.

[21] 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.

[22] S. Song, K. D. Miller, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, Sep. 2000.

[23] G. Malavena, M. Filippi, A. S. Spinelli, and C. M. Compagnoni, "Unsupervised learning by spike-timing-dependent plasticity in a mainstream NOR flash memory array—Part I: Cell operation," *IEEE Trans. Electron Devices*, vol. 66, no. 11, pp. 4727–4732, Nov. 2019.

[24] G. Malavena, M. Filippi, A. S. Spinelli, and C. M. Compagnoni, "Unsupervised learning by spike-timing-dependent plasticity in a mainstream NOR flash memory array—Part II: Array learning," *IEEE Trans. Electron Devices*, vol. 66, no. 11, pp. 4733–4738, Nov. 2019.

[25] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 4, Apr. 2009.

[26] D. Kadetotad *et al.*, "Parallel architecture with resistive crosspoint array for dictionary learning acceleration," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 5, no. 2, pp. 194–204, Jun. 2015.

[27] F. M. Bayat, X. Guo, M. Klachko, N. Do, K. Likharev, and D. Strukov, "Model-based high-precision tuning of NOR flash memory cells for analog computing applications," in *Proc. 74th Annu. Device Res. Conf. (DRC)*, Jun. 2016, pp. 1–2.

[28] G. J. Hemink, T. Tanaka, T. Endoh, S. Aritome, and R. Shirota, "Fast and accurate programming method for multi-level NAND EEPROMs," in *Symp. VLSI Technol. Dig. Tech. Papers*, 1995, pp. 129–130.

[29] C. Calligaro, A. Manstretta, A. Modelli, and G. Torelli, "Technological and design constraints for multilevel flash memories," in *Proc. 3rd Int. Conf. Electron., Circuits, Syst.*, 1996, pp. 1003–1008.

[30] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[31] C. Mead, *Analog VLSI and neural systems*. Reading, MA, USA: Addison-Wesley, 1989.

[32] G. Servalli *et al.*, "A 65nm NOR flash technology with $0.042\mu m^2$ cell size for high performance multilevel application," in *IEDM Tech. Dig.*, Aug. 2005, pp. 849–852.

[33] K.-D. Suh *et al.*, "A 3.3 v 32 mb NAND flash memory with incremental step pulse programming scheme," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995.

[34] M. Grossi, M. Lanzoni, and B. Ricco, "Program schemes for multilevel flash memories," *Proc. IEEE*, vol. 91, no. 4, pp. 594–601, Apr. 2003.

[35] B*SIM3V3.3 MOSFET Model User's Manual*, Univ. California, Berkeley, CA, USA, 2005.

[36] Y. Jiang *et al.*, "Design and hardware implementation of neuromorphic systems with RRAM synapses and threshold-controlled neurons for pattern recognition," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 9, pp. 2726–2738, Sep. 2018, doi: 10.1109/TCSI.2018.2812419.

[37] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," Tech. Rep., 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[38] B. Eitan, R. Kazerounian, A. Roy, G. Crisenza, P. Cappelletti, and A. Modelli, "Multilevel flash cells and their trade-offs," in *IEDM Tech. Dig.*, Dec. 1996, pp. 169–172.

[39] K. Ueyoshi *et al.*, "QUEST: A 7.49TOPS multi-purpose log-quantized DNN inference engine stacked on 96MB 3D SRAM using inductive-coupling technology in 40nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2018, pp. 1–2.

[40] Y. M. Tousi and E. Afshari, "A miniature 2 mW 4 bit 1.2 GS/s Delay-Line-Based ADC in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 46, no. 10, pp. 2312–2325, Oct. 2011.

[41] X. Yang *et al.*, "Gate bias dependence of complex random telegraph noise behavior in 65-nm NOR flash memory," *IEEE Electron Device Lett.*, vol. 36, no. 1, pp. 26–28, Jan. 2015, doi: 10.1109/LED.2014.2367104.

[42] S. H. Gu *et al.*, "Read current instability arising from random telegraph noise in localized storage, multi-level SONOS flash memory," in *IEDM Tech. Dig.*, May 2006, pp. 487–490, doi: 10.1109/IEDM.2006.346820.

[43] G. Malavena, S. Petro, A. S. Spinelli, and C. Monzio Compagnoni, "Impact of program accuracy and random telegraph noise on the performance of a NOR flash-based neuromorphic classifier," in *Proc. 49th Eur. Solid-State Device Res. Conf. (ESSDERC)*, Sep. 2019, pp. 122–125.

[44] A. Ghetti, C. Monzio Compagnoni, A. S. Spinelli, and A. Visconti, "Comprehensive analysis of random telegraph noise instability and its scaling in deca–nanometer flash memories," *IEEE Trans. Electron Devices*, vol. 56, no. 8, pp. 1746–1752, Aug. 2009.

[45] A. Ghetti, S. M. Amoroso, A. Mauri, and C. Monzio Compagnoni, "Impact of nonuniform doping on random telegraph noise in flash memory devices," *IEEE Trans. Electron Devices*, vol. 59, no. 2, pp. 309–315, Feb. 2012.