IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 29, 2021, accepted January 28, 2022, date of publication February 7, 2022, date of current version March 9, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149577

Neuron Circuits for Low-Power Spiking Neural Networks Using Time-To-First-Spike Encoding

SEONGBIN OH, DONGSEOK KWON, GYUHO YEOM, WON-MOOK KANG, SOOCHANG LEE, SUNG YUN WOO, JAEHYEON KIM, AND JONG-HO LEE, (Fellow, IEEE)
Department of ECE and ISRC, Seoul National University, Seoul 08826, South Korea

Corresponding author: Jong-Ho Lee (jhl@snu.ac.kr)

This work was supported in part by the Brain Korea 21 Plus Project in 2022, and in part by the National Research and Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under 2021M3F3A2A02037889.

ABSTRACT Hardware-based Spiking Neural Networks (SNNs) are regarded as promising candidates for the cognitive computing system due to its low power consumption and highly parallel operation. In this paper, we train the SNN in which the firing time carries information using temporal backpropagation. The temporally encoded SNN with 512 hidden neurons achieved an accuracy of 96.90% for the MNIST test set. Furthermore, the effect of the device variation on the accuracy in temporally encoded SNN is investigated and compared with that of the rate-encoded network. In a hardware configuration of our SNN, NOR-type analog memory having an asymmetric floating gate is used as a synaptic device. In addition, we propose a neuron circuit including a refractory period generator for temporally encoded SNN. The performance of the 2-layer neural network composed of synapses and proposed neurons is evaluated through circuit simulation using SPICE based on the BSIM3v3 model with 0.35 μm technology. The network with 128 hidden neurons achieved an accuracy of 94.9%, a 0.1% reduction compared to that of the system simulation of the MNIST dataset. Finally, each block's latency and power consumption constituting the temporal network is analyzed and compared with those of the rate-encoded network depending on the total time step. Assuming that the network has 256 total time steps, the temporal network consumes 15.12 times less power than the rate-encoded network and makes decisions 5.68 times faster.

INDEX TERMS Neuromorphic, spiking neural networks (SNNs), hardware-based neural networks, time-to-first-spike (TTFS) coding, temporal coding, neuron circuits.

I. INTRODUCTION

Artificial Neural Networks (ANNs) have recently shown remarkable results surpassing humans in certain tasks such as pattern recognition, object detection, and natural language processing [1]–[7]. The success of ANN has been attributed to the multi-layered structure inspired by the nervous system and its ability to compute nonlinear complex transformations [8], [9]. Conventional ANN, however, has fundamentally different structures from the human brain in that time has no effect on data propagation and uses analog-valued neurons [10]. Also, software-based ANNs are far from real-time and low power processing, making computing on the edge devices is challenging. In this perspective, there are many studies on neural networks based on hardware [11], [12], especially SNNs using analog synaptic devices are regarded as an enormously competent network. In SNN, data propagates in short spikes as in the biological neural system [13], [14]. Such short pulses perform a read operation on each synaptic device, and the total current flowing in the array is integrated into the analog neuron by Kirchhoff's rule, allowing high-performance parallel computation such as Vector-by-Matrix Multiplication (VMM).

There are several methods to encode the input data of multiple resolutions into the input pulse train of SNN. Commonly, the rate of pulses can be proportional to the intensity of the input data. In the rate-encoded network, the integrate and fire behavior of the neuron is almost matched to the ReLU activation function [15]. Therefore, the weights trained by ANNs can be used directly in SNNs, and these networks have shown outstanding performance on a complex benchmark such as CIFAR [16] or ImageNet [15], [17]. However, encoding an analog input value in the form of

The associate editor coordinating the review of this manuscript and approving it for publication was Mitra Mirhassani.

a firing rate requires many spikes to express the intensity of one input data. The rate-encoding method needs to be improved for efficient computing on edge devices in power consumption and latency.

Another candidate for the encoding method is temporal encoding, where the input data is transformed to the firing time of the input spikes [18]. There are several different types of temporal encoding, phase coding, burst coding, and Time-to-First-Spike (TTFS) coding are the usual methods. First, phase coding is a method of using spikes' phase [19], [20]. Generally, the spike train corresponds to a binary representation of the input value in phase coding. Meanwhile, burst coding uses weighted spikes, but the data capacity of each time step can be dynamically controlled [21]–[23]. Last, in TTFS coding, the arrival time of the spike of the input neuron is inversely proportional to the input value [24]–[28]. TTFS encoding uses only a single spike regardless of the intensity of the input data; hence it shows the highest efficiency in terms of the number of spikes. There have been several efforts to train the networks encoded by the TTFS method. However, many works used complex synaptic functions, which are difficult to implement in hardware [24]–[27]. Also, the system of some other works is not power-efficient due to their long duration of input pulses, not a spike [28].

In this paper, we configure SNN at the circuit level, where information is carried as the firing time of a single spike by adopting the temporal encoding method. First, by using a temporal backpropagation algorithm [29], we evaluate the performance of SNN at a system-level on MNIST data sets and investigate the non-ideal issues that can occur in a hardware implementation. Afterward, we propose neuron circuit blocks to generate a refractory period for a single spike-SNN. By combining proposed neuron circuits with the synaptic device reported from our previous work, the entire network is simulated at a circuit level using HSPICE. Finally, the power consumed by each block and the latency of the network are analyzed and compared with that of a rate-encoded network with the same size.

The contributions of our work are as follows:

- The verification of the operation of the entire system at the circuit level using the results measured from the TFT device,
- Proposal for an additional circuit block with functions for the TTFS-SNN system,
- Power consumption measurements for each block in a simulation, and providing a guideline for improving the power efficiency of the proposed SNN,
- Simulation results demonstrating how much advantage it has in power efficiency compared to conventional rate coding.

II. METHODS

A. TRAINING ALGORITHM

Note that the training algorithm of this work is based on the previous work [29]. Fig. 1 depicts a schematic diagram of

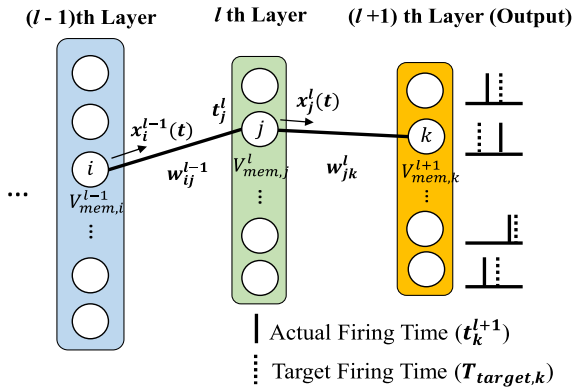


FIGURE 1. Schematic illustration of a multi-layer TTFS-SNNs.

SNN encoded by the TTFS method. In this network, the input

$$t_{\mathrm{i}}^{\mathrm{input}} = \left\lfloor \frac{I_{\max} - I_{\mathrm{i}}}{I_{\max}} T_{\max} \right\rfloor \tag{1}$$

information of the SNN is encoded using the time-to-first-spike (TTFS) method as follows:

where I_{max} is the maximum value of input data, and I_i is the input value of the i^{th} input neuron. T_{max} represent the total time step. The firing time of input neurons is inversely proportional to the input value (I_i) of each neuron [29]–[33]. The cumulative input function of the j^{th} neuron is given by:

$$S_j^l(t) = \left\{ \begin{array}{ll} 1 & (if\ t \geq t_j^l) \\ 0 & (if\ t < t_j^l) \end{array} \right\} \tag{2}$$

where t_j^l is the firing time of the j^{th} neuron in the l^{th} layer. $S_j^l(t)$ is a parameter indicating whether the neuron is a fired state at time t. The input pulses and weights are multiplied and integrated by a non-leaky IF model; thus the membrane voltage of the neuron is calculated as follows:

$$\begin{aligned} V_{mem,k}^{l+1}(t) &= V_{mem,k}^{l+1}(t-1) + \sum_j^{N^l} x_j^l(t) w_{jk}^l \\ &= \sum_j^{N^l} S_j^l(t) w_{jk}^l \end{aligned} \tag{3}$$

where $x_j^l(t)$ and w_{jk}^l are the input spikes and the weights between j^{th} and k^{th} neurons, respectively. When the membrane voltage reaches the neuron threshold (V_{th}^l), the neuron fires and generates a spike in the next layer. We assumed that each neuron could generate at most a single spike per image because of the refractory period.

In TTFS network, the output value of neuron k is expressed as the firing time (t_k^o). Accordingly, the error function of output layer is defined by:

$$\delta_k^o = e_k = \frac{T_{target,k} - t_k^o}{T_{\max}}, \tag{4}$$

so that the output neuron can fire as close as possible to the target firing time of each neuron ($T_{target,k}$). The weights are

TABLE 1. Parameters used in the system-level simulation.

Parameters	Description	Value
$\alpha_{penalty}$	Penalized term for the incorrect neuron	1
η	Learning rate	0.02
V_{th}^{l}	Threshold voltage of I&F neuron in l th layer	1.6 V
I_{init}	Initialization term	0.1

updated as follows:

$$\Delta w_{ij}^{l-1} = \begin{Bmatrix} -\eta \delta_j^l S_i^{l-1}(t_j^l) & if\ t_j^l < t_{\max} \\ 0 & otherwise \end{Bmatrix} \quad (5)$$

where η is a learning rate. Additionally, the delta values (δ_j^l) are calculated as the weighted sum of the delta values of neurons in the following layer (δ_k^{l+1}).

We also set the target firing time of output neurons as follows:

$$T_{target,k} = \begin{cases} \tau & : if\ k = target\ label \\ \tau + \alpha_{pen} & : if\ k \neq target\ label,\quad t_k^o \leq (T_{max} - \alpha_{pen}) \\ t_k^o & : if\ k \neq target\ label,\quad t_k^o > (T_{max} - \alpha_{pen}) \end{cases} \quad (6)$$

where τ is the minimum value of the firing time among the output neurons, and α_{pen} represents the penalizing term of wrongly fired neuron. Correct output neuron is encouraged to fire first among the output neurons at time τ, and output neurons fired wrongly around τ have a higher risk of responding incorrectly, so that penalizing as α_{pen}.

B. NOR-TYPE SYNAPTIC DEVICE HAVING ASYMMETRIC FLOATING GATE

On the other hand, various types of emerging memory are being reported as candidates for artificial synaptic devices, a key element for configuring SNN in hardware. In our previous work, a NOR-type flash memory device was fabricated using a conventional CMOS process [34]. Fig. 2 (a) shows that this TFT type device has a poly channel and a half-covered poly-Si floating gate (FG) that functions as a charge storage layer. The thickness of blocking SiO_2, FG, and tunneling SiO_2 are 15 nm, 80 nm, and 7 nm, respectively, and the channel length (the length between source and drain) and the width are 0.5 μm each. Input pulses are presented to each gate (WL), and the currents of the synaptic devices are summed in the common drain line (BL). Hence, the output of vector-by-matrix multiplication (VMM) can be expressed as the current of each post-neuron. In addition, since the current is controlled by three terminals in this FET-type synaptic device, it is more resistant to sneak path issues [35], [36] or off-current issues [37], [38] than two-terminal devices such as RRAM.

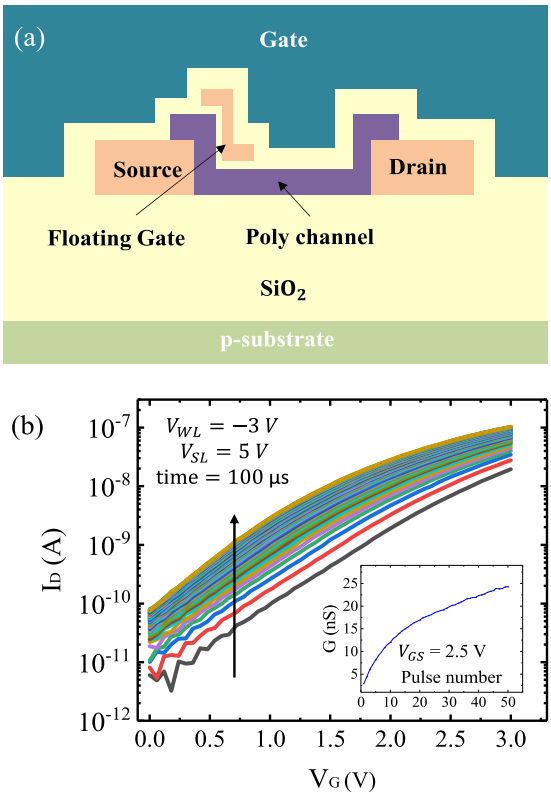


FIGURE 2. (a) Cross-sectional view of NOR-type flash memory having an asymmetric floating gate. (b) Measurement of I_D-V_G characteristics of the synaptic device by applying a consecutive erase pulse. The inset shows the change of conductance in the read condition according to the applied pulse number.

Fig. 2 (b) provides the measured I_D-V_G curves of the NOR-type flash memory device. 50 repeated erase pulses ($V_{WL} = -3$ V, $V_{SL} = 5$ V, duration $= 100$ μs) are applied, and the threshold voltage of the device decreases. The inset of Fig. 2 (b) shows the conductance change when the device is under the read conditions. The behavior of these synaptic devices is similar to the long-term potentiation (LTP) of the synapse in the nervous systems.

III. SYSTEM-LEVEL SIMULATIONS AND RESULTS

A. PERFORMANCE OF SNN ON MNIST

The system-level simulation is performed on the MNIST datasets to evaluate the performance of SNN based on the NOR-type synaptic devices. All system-level simulations in this work were performed through the python-based TTFS-SNN simulator. The edge of the image is removed and resized to 20 × 20 to reduce the size of the network. As depicted in Fig. 3 (a), a 2-layer SNN was assumed, and the simulation was conducted by increasing the number of hidden neurons to 128, 256, and 512. The parameters used in the simulation are shown in Table 1. The input data is transformed into a TTFS spike train over 64 time steps, and the $\alpha_{penalty}$ mentioned in training algorithm section is set to 1. The training batch size is 1, and the initial learning rate is set to 0.02, but it gradually drops as training continues. The threshold of neurons in all

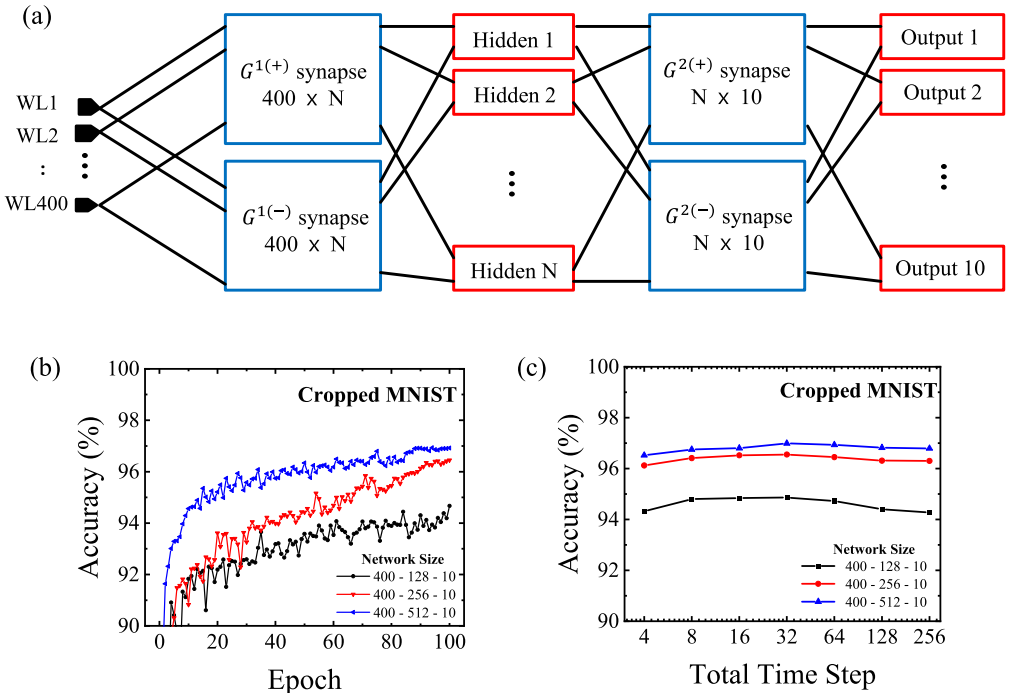


FIGURE 3. (a) Conceptual diagram of the 2-layer network with N hidden neurons. (b) Training curves of the TTFS network as a parameter of the number of neurons in the hidden layer. (c) The accuracy of the TTFS network as a parameter of the total time steps.

layers is 1.6 V, and the winner neuron of SNN is determined as the neuron that fires first among neurons in the last readout layer. However, if any output neuron does not spike until the last time step, it is evaluated by considering the membrane voltage of the output neuron at the last time step. Fig. 3(a) depicts a schematic diagram of a fully connected hardware neural network (HNN). The entire network is composed of the synaptic array and neuron circuits. The weights are initialized using the initialization method proposed by K. He [39]. The initial weight distribution is given by:

$$W^l \sim N(I_{\text{init}}, \frac{2}{n_{in}^l}) \tag{7}$$

where n_{in}^l represents the number of input nodes in the lth layer. However, by changing the mean of the normal distribution to a positive value (I_{init}) rather than 0, many hidden neurons are fired at the start of training, leading to participation in data propagation.

Weights trained by the off-chip learning rules described in training algorithm section are transferred to HNN. Before being transferred, the weights are normalized and quantized to the 101 states. The weight for one synapse in the SNN is represented by the conductance difference between two synaptic devices as follows:

$$w_{ij}^l = G_{ij}^{l(+)} - G_{ij}^{l(-)} \tag{8}$$

where G_{ij}^l is one of the measured conductance value of the asymmetric FG synaptic device ($G(1), G(2), ..G(50)$). For all the positive w_{ij}^l, $G_{ij}^{l(-)}$ is set to $G(1)$ (G_{min}), and conversely, for all the negative w_{ij}^l, $G_{ij}^{l(+)}$ is set to $G(1)$ (G_{min}). If the w_{ij}^l is 0, the conductance of both synaptic devices is set to $G(1)$ (G_{min}) [40]. The quantized target weights of each synapse can be transferred by applying the corresponding index number of pulses to devices in the synaptic array [41].

Fig. 3 (b) presents MNIST accuracy as a parameter of the width of the hidden layer in a 2-layer SNN. The accuracy for 60,000 training sets and 10,000 test sets not used for training is indicated by dotted and solid lines, respectively. As the number of hidden neurons increased, it was observed that the accuracy increased. The accuracy of the network (400 - 512 - 10) is 99.21% for the training set and 96.90% for the test set. The accuracy shows the degradation of about 1% compared to those of rate-encoded networks of similar size [42], [43]. Fig. 3 (c) shows the recognition accuracy with the total number of time steps per image. The time steps, representing the image's resolution, can be reduced to 8 without significant accuracy degradation (0.15% degradation for 512 hidden neurons).

B. EFFECTS ON ACCURACY BY VARIATION IN HARDWARE

Changes in device characteristics caused by process variations during manufacturing negatively affect the operation of synaptic devices and neuronal circuits, which reduces the recognition accuracy of the SNN. Several types of variation have been analyzed in previous studies [44]–[49]. We categorize the four major variations as follows:

1) device-to-device variation in the synaptic array [44], [45],
2) firing threshold variation in the neuron circuits [46],
3) stuck-at-off variation in the synaptic array [47]–[49], and
4) stuck-at-off variation in the neuron circuits.

Note that our network does not take into account the pulse-to-pulse variation considered in many previous studies [33], [50] since the weights obtained through off-chip learning are transferred once to the synaptic devices in the array. The recognition accuracy with the variation of device characteristics is compared with that of conventional rate-encoded networks of the same size. In the rate-encoded network, the number of input spikes follows a Poisson distribution, and the weights are also quantized of the same resolution in the same manner as in the TTFS network.

We first mathematically model the device-to-device variation in the synaptic array as follows:

$$W^l \leftarrow W^l \times N(1, \sigma^2_{weight}), \tag{9}$$

where W^l denotes the overall quantized weights, and $N\left(1, \sigma^2_{weight}\right)$ stands for the normal distribution with mean 1 and standard deviation σ_{weight}. Also, weights with a value of 0 are set to random numbers with normal distribution N(0, σ^2_{weight}). In addition, variation of neuron thresholds is also modeled by normal distribution as follows:

$$V^l_{th} \leftarrow \max(V^l_{th} \times N(1, \sigma^2_{th}), 0). \tag{10}$$

However, negative neuron thresholds are difficult to implement in hardware, so they follow a clipped normal distribution. Lastly, a stuck-at-off fault where one of the devices is not working is considered. Dead synaptic devices can cause the current to not flow even when the input pulse is applied, resulting in a fatal error in the weighted sum. Further, if the neuron block dies, there may be cases where the current cannot be integrated into the capacitor, or the spike cannot be emitted even if the membrane voltage exceeds the neuron threshold. We defined the stuck-at-off ratio as the number of dormant synaptic devices (or neurons) relative to the total number of synapses (or neurons) in the array and named it $R_{synapse}(R_{neuron})$. The conductance of the dead synapse is assumed to 0, and the input by a dead neuron is assumed to 0 regardless of the membrane voltage.

Fig. 4 shows how the accuracy for the MNIST data set changes as the device variation increases. Simulations were performed on a 2-layer network with 128, 256, and 512 hidden neurons. The simulation was repeated 10 times with the modeled variation and the results are indicated by error bars. Compared to the rate-encoded network, the TTFS network is vulnerable to variation since a single neuron can contribute only one spike in the inference process. In the TTFS network, even if only one spike disappears (or even one false spike occurs), it makes a significant error in the overall weighted sum. In particular, it is observed that the network with a small number of hidden neurons shows severe accuracy degradation due to each neuron's importance. Therefore, synaptic arrays of TTFS networks should be finely controlled so that the variation to be minimized. For example, device-to-device variation can be reduced by precision tuning using the read-write-verify scheme in the weight transfer process [51].

IV. CIRCUIT-LEVEL SIMULATIONS AND RESULTS

A. SNN ARCHITECTURE

In this section, we propose the neuron blocks suitable for TTFS encoded SNN, and simulate fully connected HNN at circuit-level using SPICE. BSIM3v3 model based on 0.35 μm technology was used in circuit-level simulation. The NOR-type synaptic device is modeled with a Voltage-Controlled Current Source (VCCS) behavioral model with 3 terminals (gate, drain, source) as shown in Fig. 5 (a) [52], [53]. The current between the drain and the source is determined by the voltage difference between the gate and the source. As shown in Fig. 5 (b), the behavior modeling of the synaptic device was based on the results measured by increasing the gate voltage from 0 V to 3 V in 60 mV steps.In addition, neuron circuits consist of a current mirror, an I&F block, and a refractory period generator. Fig. 5 (c) depicts a modeled synaptic array and a current mirror designed for summing and subtracting currents. A single wordline (WL) corresponds to an input, and the weighted sum of 400 inputs is expressed as the sum of current flowing through the bitline (BL). Currents flowing in the positive and negative synaptic arrays are copied through the respective current mirrors to integrate the net charge in the membrane capacitor. Before the input pulse is presented, V_{mem} of all neurons are initialized to V_{DD2} by V_{init}. If the initial membrane voltage of the neuron is not V_{DD2} (e.g., 0 V), the negative charge created by the inputs in the early stage of the time domain cannot be integrated into the capacitor, so the final result of the weighted sum can be distorted. Since SNN encoded by the TTFS method assumes that one neuron spikes at most once, the neurons already fired should enter the refractory period so that no more spikes are generated. To implement this, the output neuron that has already fired keeps V_{refrac} in a high state until the corresponding input ends. This causes M7 and M8 to turn off so that no more current flows through the synaptic array. Fig. 5 (d) shows the circuit of the refractory period generator (RPG). The block for generating V_{refrac} is based on the structure of the latch. Before the input pulse is presented, M12 is turned on by V_{init}, so the output node of RPG is initialized to the ground state. Then, as soon as the I&F block fires, M10 and M11 turn on and V_{refrac} goes to high state, which is maintained until a new input data is presented. Fig. 5 (e) represents the I&F block constituting the neuron [54]. If the membrane voltage exceeds the V_{th} of M14 by the integrated charge, node 1 in the high state changes to the low state. This brings node 2 to the high state. After the delay time by C_{pulse}, the voltage at node 2 turns M21 on and puts node 1 back in high state, and returns node 2 to original state. The W/L ratio of M16 acting as a resistor and the value of C_{pulse} determine the width of a spike generated in the output node. In addition, the W/L ratio

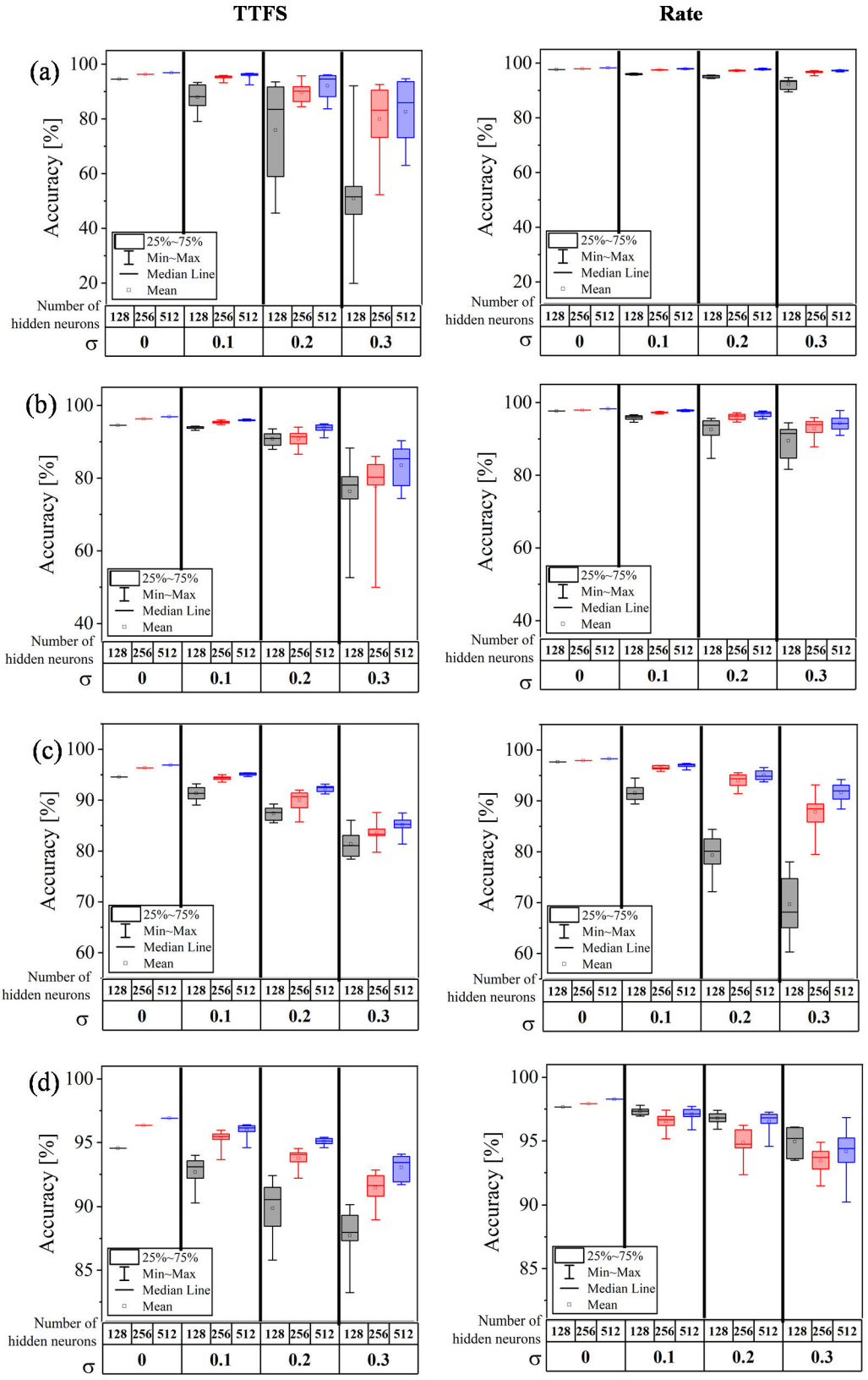


FIGURE 4. Evaluation of the TTFS (1st column) and rate-encoded (2nd column) network as a parameter of (a) device-to-device variation in the synaptic array, (b) firing threshold variation in the neuron circuits, (c) the stuck-at-off ratio in the synaptic array and (d) neuron circuits.

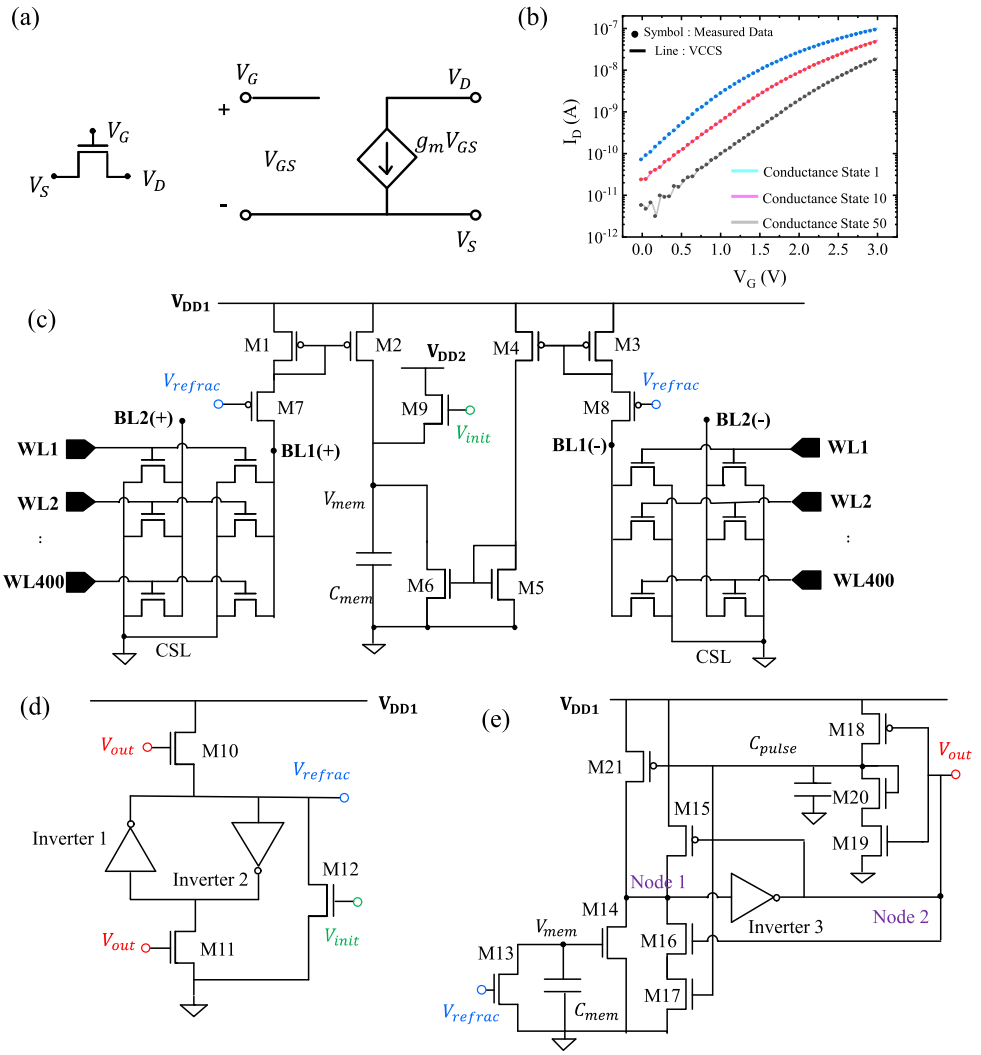


FIGURE 5. (a) Voltage-Controlled Current Source (VCCS) based synapse model in SPICE. (b) Comparison of I_{DS}-V_{GS} curves between synaptic device measurement data and VCCS model. Circuit diagram of the (c) synaptic array, current mirror, (d) refractory period generator, (e) integrate and fire block that makes up the neuron circuit.

of M14 and M21 determines the voltage of node 1, so it affects the threshold of the neuron. After I&F block fires, V_{refrac} turns M13 on to keep V_{mem} as the ground state.

B. PERFORMANCE IN CIRCUIT-LEVEL SIMULATION

Among the networks simulated in the system-level simulation section, a relatively light network, the 400-128-10 sized network is simulated using a circuit simulator (HSPICE) with predictive technology model (PTMs). In order to reduce the simulation time, the total time step is assumed to be 8, which hardly degrades the network performance. A circuit-level simulation was performed with 0.35 μm CMOS technology, and the parameters of the components in circuits are shown in Table 2. Fig. 6 provides the waveforms of some nodes in the process of inferencing MNIST data sets. Before the input pulses are presented, V_{init} is first presented to initialize the membrane capacitors in I&F block and RPG block. After that, as shown in Fig. 6 (a), all inputs are transformed into time-to-first spike pulses with a duration of 0.5 μs over 8 time steps. The interval between each time step of the input pulse is

TABLE 2. Parameters of components used in the circuit-level simulation.

Description	Components	Value
Width / Length of Transistor	M1 ~ M9, M21	0.5 µm / 2 µm
	M10 ~ M19, Inverter 1~3	0.5 µm / 0.5 µm
	M20	0.5 µm / 7 µm
Capacitance	$C_{mem,\ hidden}$	122 fF
	$C_{mem,\ output}$	77 fF
	C_{pulse}	2 pF
Supply Voltage	$V_{dd,1}$	2.5 V
	$V_{dd,2}$	0.4 V
Threshold Voltage of Neuron	$V_{th,hidden}$, $V_{th,out}$	0.92 V

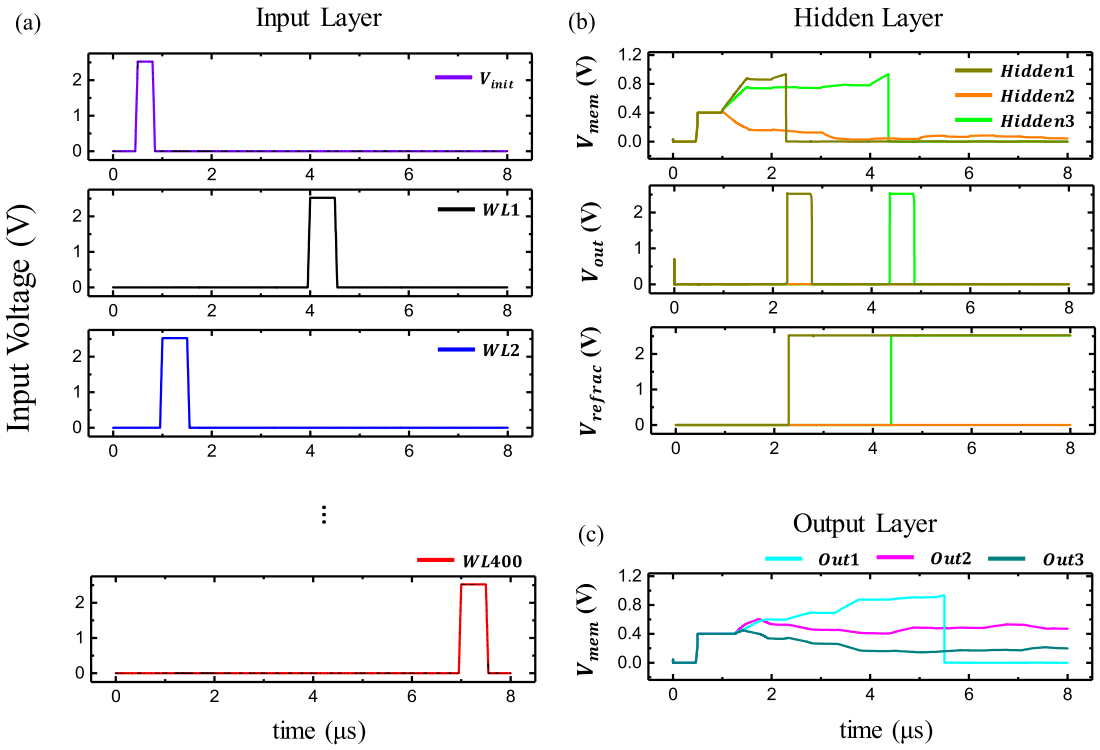


FIGURE 6. **(a) Pulses fed into the input neuron shown in the time domain. (b) (Top) Evolution of the membrane voltage of hidden neurons. (Middle) Generated output pulse and (Bottom) refractory period by the neurons in the hidden layer. (c) Evolution of the membrane voltage of output neurons. The answer predicted by the network is the class of output neuron 1. All results are simulated at the circuit-level.**

also set to 0.5 μs. The rising and falling times of input pulses are each set to 0.1 μs. Fig. 6 (b) shows transient waveforms of some nodes in hidden neurons. The currents flowing through the synaptic array by the input pulses are integrated into the capacitor of the hidden neurons, and when V_{mem} exceeds the neuron threshold, the corresponding neuron fires and presents a spike with a width of 0.5 μs to the post-layer. At the very moment the neuron fires, V_{refrac} generated by each RPG prevents further integration of charge into the fired neuron. Finally, Fig. 6 (c) represents the membrane voltage of neurons in the output layer. As in the system simulation, the earliest fired output neuron class is the result predicted by SNN. However, in rare cases when no output neuron fires, the neuron with the highest membrane voltage is considered the winner neuron.

Fig. 7 compares the results of system-level and circuit-level simulations. The network size is 400 - 128 - 10, and the number of time steps for each image is set to 8. Fig. 7 (a)-(c) shows the firing times of input, hidden, and output layers in the system-level simulation of one image. The x-axis of the three raster plots represents the time, and the y-axis stands for the index number of neurons in each layer. Fig. 7 (e) and (f) depict a raster plot of spike timing in hidden neurons and V_{mem} of output neurons in the circuit-level simulation for the same image. By comparing the firing times of hidden neurons and output neurons shown in (b), (e) and (c), (f), it is observed that the results of both simulations are similar. In addition, we also simulated the circuits for 1000 randomly selected MNIST data sets. Fig. 7(d) shows the result of comparing the firing time of the winner neuron obtained by simulations at system-level (x-axis) and circuit-level (y-axis). Since the system-level simulation was performed during 8 discrete time steps, the firing time in the system-level is a discrete value. The firing times of the two simulations are not perfectly matched, but they show almost the same tendency, which means the proposed SNN shown at the circuit level works pretty much like that at the system level. Indeed, the proposed SNN has reached 94.9% accuracy for networks having 128 hidden neurons at the circuit-level. This accuracy is only 0.1% lower than the 95.0% accuracy in a system-level simulation. The reason for the slight decrease in accuracy is that the off current in the synaptic array is not considered at the system-level. Also, calculating the weighted sum through discrete time steps in system-level simulation can cause a difference from actual circuit operation.

C. POWER MEASUREMENTS

In this section, we estimate the power consumed by the TTFS network at the circuit-level and compare the results with that of the rate-encoded network. The most significant advantage of the TTFS encoding method is that it requires much fewer pulses compared to the conventional rate encoding method, which results in lower power consumption. TTFS

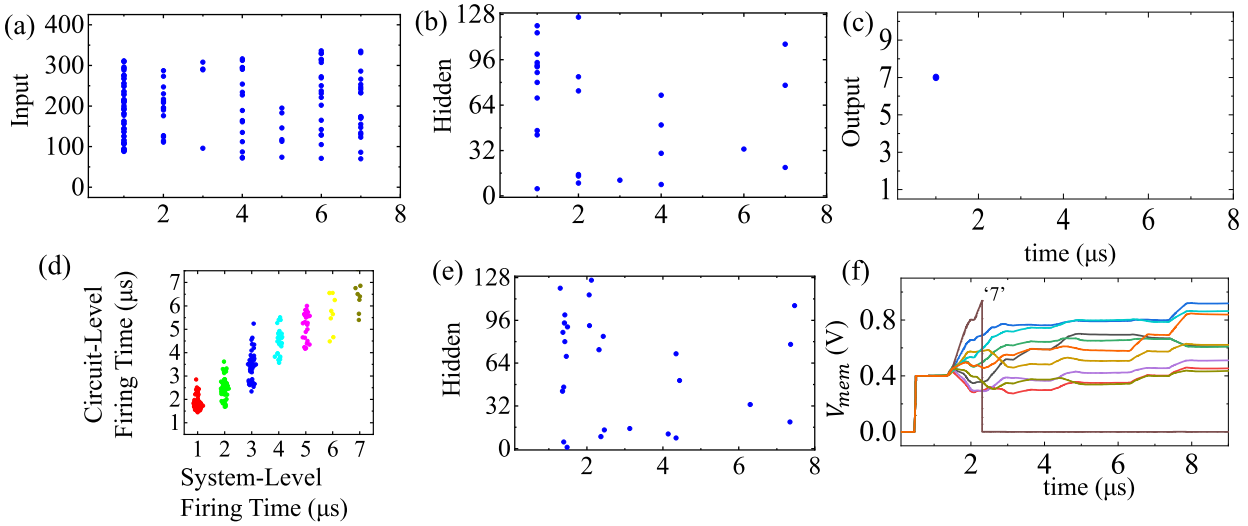


FIGURE 7. Raster plots of the spike timing of the (a) input, (b) hidden, and (c) output neuron in the 2-layer (400-128-10) SNN for randomly selected test data '7'. The x-axis represents the time in the system simulation, and the y-axis represents the index of each neuron. (d) Comparison of the firing time of the winner neuron in the system-level (x-axis) and circuit-level (y-axis) simulation. (e) Raster plots of the hidden neuron when simulated in the circuit-level for the same network size and data as (b). (f) Evolution of the membrane voltage of output neurons in the circuit-level simulation.

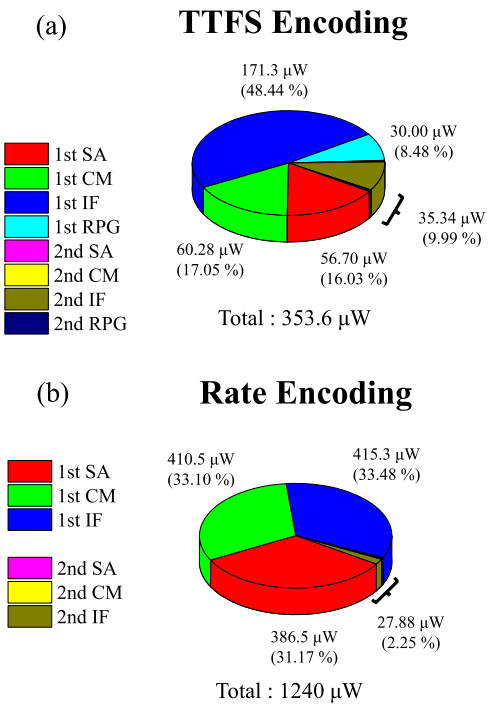


FIGURE 8. Pie chart of power consumption in (a) TTFS and (b) rate-encoded networks.

and rate-encoded networks, each with the same number (128) of hidden neurons are simulated for 100 randomly picked MNIST data sets, and the total time steps for each image are set to 8.

Fig. 8 shows the amount of power consumed by each block in the proposed SNN. The entire network consists of synapse array (SA) and neuron circuits, and specifically, the neuron is composed of a current mirror (CM), a circuit for integrate and fire (IF), and a refractory period generator (RPG). Fig. 8 (a) depicts the power consumed in the inference process of the TTFS network. The entire network consumes 353.6 μW, and it is observed that most ($\sim$90%) of the power is consumed by the components in the 1st layer. In particular, I&F block accounts for a remarkable proportion of power consumption. This is because not only power is consumed to generate the pulse, but also subthreshold leakage current flows due to the membrane voltage of the neuron below V_{th}^{l}. In the I&F block depicted in Fig. 5 (f), even if V_{mem} does not exceed V_{th}, M14 can be finely turned on if V_{mem} is a positive value. This creates a leakage path through M14 and M15, allowing current to flow even the neuron is not fired. Since the number of spikes in the TTFS network is small, this standby power occupies a relatively large portion as much as the power required to generate spikes. Improving the structure of I&F circuits to deal with this issue can be a topic for further study. Fig. 8 (b) represents the power consumed in the rate-encoded networks. In the circuit-level simulation, each input spike of the rate-encoded network is filled from the last time steps [55]. Compared to the TTFS encoding method, the rate-encoding method requires more pulses to represent an image, so the currents in the synapse array and current mirror are enormous. Likewise, the higher the number of spikes generating in each layer, the greater the power consumed by I&F block. Unlike the TTFS network, the rate-encoded network does not require a refractory period generator, but the power that can be saved is very small ($\sim$2%). It is obtained that the entire network consumes 1240 μW of power, which is 3.5 times more than that of the TTFS network.

The power consumption ratio of the two networks increases as the total time steps per image increases. Fig. 9 shows the required number of spikes and consumed energy as a function of time step. The solid line represents the average value of the spike numbers required to compute

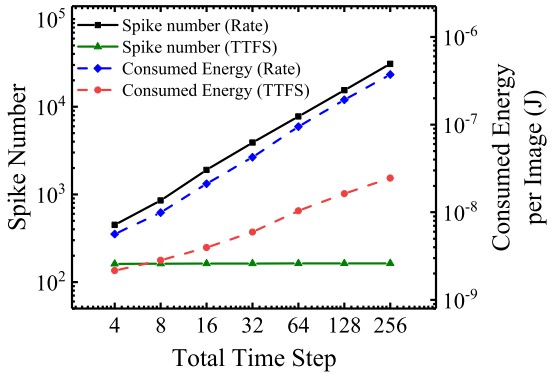


FIGURE 9. Comparison between TTFS and rate-encoded network in terms of spike number (system-level) and consumed energy per image (circuit-level). Consumed energy was measured at various time steps on 100 randomly selected MNIST test sets. The simulation was conducted at various time steps.

an image at the system-level. The required spike number is the sum of spikes generated in all layers. The number of pulses in the TTFS network is only counted until the winner neuron of the output layer fires, and the number in the rate-encoded network is counted until the final time step. When the input is converted to a spike rate, the number of spikes required to express the same values increases as the time step increases. On average, if the total time steps are 4, only 495 spikes are required, whereas 30793 spikes are needed when the time step reaches 256. On the other hand, the number of spikes in the TTFS encoding method is nearly constant at about 162 regardless of the total time steps. Therefore, as the resolution of input data increases, the difference between the required spike numbers of the two networks increases.

Meanwhile, the dashed lines represent the average energy required to compute an image as a result of circuit-level simulation. Since the time required to compute the image depends on the time step, the energy is compared between two networks. On average, the rate-encoded network consumes 5.65 nJ of energy per image at a time step number of 4 and 372 nJ at 256. On the other hand, the TTFS network consumes 2.16 nJ at a time step number of 4 and 24.6 nJ at a time step number of 256 to compute one image. As the total time step of the TTFS network grows, the number of spikes is not changed, but the consumed energy is increased. This is because the amount of energy consumed by the leakage path in I&F block is proportional to the time for processing an image. Rate-encoded networks are also affected by this leakage, but the relative proportion of the leakage in total energy consumption is less than that in the TTFS network due to a large number of spikes. Hence, the consumed energy of the rate encoded network is almost proportional to the required spike number. Meanwhile, the TTFS network uses a small number of spikes, tends to increase the consumed energy even if the required spike number is constant. Nevertheless, the superiority of the TTFS network in terms of power-efficiency is increased as the time step increases compared to the rate encoded network. Finally, the ratio of power efficiency of the TTFS networks to rate-encoded networks reaches 15.75 at a time step number of 256.

The TTFS network also has an advantage in terms of the latency, the time it takes to infer the answer. The latency of the TTFS network was calculated as the average value of the time until the emission of the first spike at the output layer. It is observed that the TTFS network with 128 hidden neurons can make a decision about 5 times faster than rate-encoded network of the same size.

V. CONCLUSION

In this study, we have evaluated the performance of the SNN consisting of NOR-type asymmetric FG synaptic devices and neuron circuits at the system-level and circuit-level. Input data was encoded as the time of the input spikes (time-to-first spike: TTFS), and the network was trained by temporal backpropagation, a learning method suitable for networks applying the TTFS encoding method. The neural network with 512 hidden neurons showed a competitive accuracy of 96.90 % for the cropped MNIST data sets. We also investigated the impact of the non-ideal characteristics of the synaptic array and neuron circuits on accuracy. These results can be a guideline that informs which level of variation is allowed in the TTFS network. In addition, we proposed a neuron circuit for inferencing temporal data and modeled the synapse device to demonstrate the operation of the entire network. Simulating an SNN with 128 hidden neurons in SPICE gives 94.9% accuracy for 1000 MNIST data sets, almost no degradation compared to the system-level simulation. We also analyzed the power consumed in the inference process by each block in SNN. When using 8 time steps in a 400-128-10 size network, the TTFS network showed approximately 3.5 times higher power efficiency compared to the rate-encoded network. At the same network size, the TTFS networks showed significantly lower energy consumption and shorter latency than rate-encoded networks. The difference in energy consumption between the two networks increases as the number of time steps increases.

As a further study, more realistic circuit-level simulation can be conducted. In fact, as the crossbar array becomes large and the unit cell scales down, the effect of parasitic resistance and parasitic capacitance on network performance may increase. A more effective training algorithm that can overcome the performance decrease considering hardware the non-ideality can also be studied further.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural. Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[3] T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," *Frontiers Neurosci.*, vol. 10, pp. 1–13, Jul. 2016, doi: 10.3389/fnins.2016.00333.

[4] S. Ambrogio, P. Narayana, H. Tsai, R. M. Shelby, I. Boybat, C. Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. P. Farinha, B. Kileen, C. Cheng, Y. Jaoudi, and G. W. Burr, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, pp. 60–67, Jun. 2018, doi: 10.1038/s41586-018-0180-5.

[5] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 160–167, doi: 10.1145/1390156.1390177.

[6] F. Jiang, L. Dong, and Q. Dai, "Designing a mixed multilayer wavelet neural network for solving ERI inversion problem with massive amounts of data: A hybrid STGWO-GD learning approach," *IEEE Trans. Cybern.*, early access, May 20, 2020, doi: 10.1109/TCYB.2020.2990319.

[7] F. Jiang, L. Dong, K. Wang, K. Yang, and C. Pan, "Distributed resource scheduling for large-scale MEC systems: A multi-agent ensemble deep reinforcement learning with imitation acceleration," *IEEE Internet Things J.*, early access, Sep. 20, 2021, doi: 10.1109/JIOT.2021.3113872.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[9] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. Adv. Neural. Inf. Process. Syst. (NIPS)*, 2014, pp. 2654–2662. [Online]. Available: http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep

[10] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. M. McGinnity, "A review of learning in biologically plausible spiking neural networks," *Neural Netw.*, vol. 122, pp. 253–272, Feb. 2020, doi: 10.1016/j.neunet.2019.09.036.

[11] K. Khalil, O. Eldash, B. Dey, A. Kumar, and M. Bayoumi, "Architecture of a novel low-cost hardware neural network," in *Proc. IEEE 63rd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2020, pp. 1060–1063.

[12] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "An efficient approach for neural network architecture," in *Proc. 25th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2018, pp. 745–748.

[13] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA, USA: MIT Press, 1986.

[14] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers Neurosci.*, vol. 12, p. 774, Oct. 2018, doi: 10.3389/fnins.2018.00774.

[15] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers Neurosci.*, vol. 11, p. 682, Dec. 2017, doi: 10.3389/fnins.2017.00682.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper

[17] E. Hunsberger and C. Eliasmith, "Training spiking deep networks for neuromorphic hardware," 2016, *arXiv:1611.05141*.

[18] W. Maass and T. Natschläger, "Emulation of Hopfield networks with spiking neurons in temporal coding," *Computational Neuroscience*. Springer, 1998, pp. 221–226, doi: 10.1007/978-1-4615-4831-7_37.

[19] C. Kayser, M. A. Montemurro, N. K. Logothetis, and S. Panzeri, "Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns," *Neuron*, vol. 61, no. 4, pp. 597–608, 2009.

[20] J. Kim, H. Kim, S. Huh, J. Lee, and K. Choi, "Deep neural networks with weighted spikes," *Neurocomputing*, vol. 311, pp. 373–386, Oct. 2018.

[21] E. M. Izhikevich, N. S. Desai, E. C. Walcott, and F. C. Hoppensteadt, "Bursts as a unit of neural information: Selective communication via resonance," *Trends Neurosci.*, vol. 26, no. 3, pp. 161–167, Mar. 2003.

[22] S. Park, S. Kim, H. Choe, and S. Yoon, "Fast and efficient information transmission with burst spikes in deep spiking neural networks," in *Proc. 56th Annu. Design Autom. Conf.*, Jun. 2019, pp. 1–6.

[23] S. Park, D. Lee, and S. Yoon, "Noise-robust deep spiking neural networks with temporal information," 2021, *arXiv:2104.11169*.

[24] S. M. Bohte, J. N. Kok, and H. La Poutré "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, nos. 1–4, pp. 17–37, 2000, doi: 10.1016/S0925-2312(01)00658-0.

[25] Q. Yu, H. Tang, K. C. Tan, and H. Yu, "A brain-inspired spiking neural network model with temporal encoding and learning," *Neurocomputing*, vol. 138, pp. 3–13, Aug. 2014., doi: 10.1016/j.neucom.2013.06.052.

[26] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3227–3235, Jul. 2018, doi: 10.1109/TNNLS.2017.2726060.

[27] I. M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, and J. Alakuijala, "Temporal coding in spiking neural networks with alpha synaptic function," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8529–8533, doi: 10.1109/ICASSP40776.2020.9053856.

[28] B. Rueckauer and S.-C. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351295.

[29] S. R. Kheradpisheh and T. Masquelier, "Temporal backpropagation for spiking neural networks with one spike per neuron," *Int. J. Neural Syst.*, vol. 30, no. 6, Jun. 2020, Art. no. 2050027, doi: 10.1142/S0129065720500276.

[30] R. Vaila, J. Chiasson, and V. Saxena, "A deep unsupervised feature learning spiking neural network with binarized classification layers for EMNIST classification using SpykeFlow," 2020, *arXiv:2002.11843*.

[31] C. Lee, G. Srinivasan, P. Panda, and K. Roy, "Deep spiking convolutional neural network trained with unsupervised spike-timing-dependent plasticity," *IEEE Trans. Cogn. Devel. Syst.*, vol. 11, no. 3, pp. 384–394, Sep. 2019, doi: 10.1109/TCDS.2018.2833071.

[32] S. R. Kheradpisheha, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Netw.*, vol. 99, pp. 56–67, Mar. 2018, doi: 10.1016/j.neunet.2017.12.005.

[33] S. Oh, C.-H. Kim, S. Lee, J. S. Kim, and J.-H. Lee, "Unsupervised online learning of temporal information in spiking neural network using thin-film transistor-type NOR flash memory devices," *Nanotechnology*, vol. 30, no. 43, Oct. 2019, Art. no. 435206, doi: 10.1088/1361-6528/ab34da.

[34] C. H. Kim, S. Lee, S. Y. Woo, W. M. Kang, S. Lim, J. H. Bae, J. Kim, and J. H. Lee, "Demonstration of unsupervised learning with spike-timing-dependent plasticity using a TFT-type NOR flash memory array," *IEEE Trans. Electron Devices*, vol. 65, no. 5, pp. 1774–1780, May 2018, doi: 10.1109/TED.2018.2817266.

[35] E. Linn, R. Rosezin, C. Kügeler, and R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nature Mater.*, vol. 9, pp. 403–406, Apr. 2010, doi: 10.1038/NMAT2748.

[36] J. Liang and H.-S. P. Wong, "Cross-point memory array without cell selectors-device characteristics and data storage pattern dependencies," *IEEE Trans. Electron Devices*, vol. 57, no. 10, pp. 2531–2538, Oct. 2010, doi: 10.1109/TED.2010.2062187.

[37] P.-F. Chiu and B. Nikolić, "A differential 2R crosspoint RRAM array with zero standby current," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 5, pp. 461–465, May 2015, doi: 10.1109/TCSII.2014.2385431.

[38] S. Kim, X. Liu, J. Park, S. Jung, W. Lee, J. Woo, J. Shin, G. Choi, C. Cho, S. Park, D. Lee, E.-J. Cha, B.-H. Lee, H. D. Lee, S. G. Kim, S. Chung, and H. Hwang, "Ultrathin (<10 nm) Nb_2O_5/NbO_2 hybrid memory with both memory and selector characteristics for high density 3D vertically stackable RRAM applications," in *Proc. Symp. VLSI Technol. (VLSIT)*, Jun. 2012, pp. 155–156, doi: 10.1109/VLSIT.2012.6242508.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034. [Online]. Available: https://www.cv-foundation.org/openaccess/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper

[40] G. W. Burr, R. M. Shelby, S. Sidler, C. di Norfo, J. Jang, B. Irem, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498–3507, Jul. 2015, doi: 10.1109/TED.2015.2439635.

[41] S. Lim, D. Kwon, J. H. Eum, S. T. Lee, J. H. Bae, H. Kim, C. H. Kim, B. G. Park, and J. H. Lee, "Highly reliable inference system of neural networks using gated Schottky diodes," *IEEE J. Electron Devices Soc.*, vol. 7, pp. 522–528, 2019, doi: 10.1109/JEDS.2019.2913146.

[42] D. Kwon, S. Lim, J.-H. Bae, S.-T. Lee, H. Kim, C.-H. Kim, B.-G. Park, and J.-H. Lee, "Adaptive weight quantization method for nonlinear synaptic devices," *IEEE Trans. Electron Device*, vol. 66, no. 1, pp. 395–401, Jan. 2019, doi: 10.1109/TED.2018.2879821.

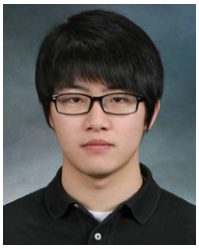
[43] E. O. Neftci, C. Augustine, S. Paul, and G. Detorakis, "Event-driven random back-propagation: Enabling neuromorphic deep learning machines," *Frontiers Neurosci.*, vol. 11, p. 324, Jun. 2017, doi: 10.3389/fnins.2017.00324.

[44] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S.-P. Wong, "A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation," *Adv. Mater.*, vol. 25, no. 12, pp. 1774–1779, Mar. 2013, doi: 10.1002/adma.201203680.
[45] D. Querlioz, O. Bichler, P. Dollfus, and C. Gamrat, "Immunity to device variations in a spiking neural network with memristive nanodevices," *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 288–295, May 2013, doi: 10.1109/TNANO.2013.2250995.
[46] Y. Sakemi, K. Morino, T. Morie, and K. Aihara, "A supervised learning algorithm for multilayer spiking neural networks based on temporal coding toward energy-efficient VLSI processor design," 2020, *arXiv:2001.05348*.
[47] Y. Li, Z. Wang, R. Midya, Q. Xia, and J. J. Yang, "Review of memristor devices in neuromorphic computing: Materials sciences and device challenges," *J. Phys. D, Appl. Phys.*, vol. 51, no. 50, Dec. 2018, Art. no. 503002, doi: 10.1088/1361-6463/aade3f.
[48] D. Kwon, S. Lim, J.-H. Bae, S.-T. Lee, H. Kim, Y.-T. Seo, S. Oh, J. Kim, K. Yeom, B.-G. Park, and J.-H. Lee, "On-chip training spiking neural networks using approximated backpropagation with analog synaptic devices," *Frontiers Neurosci.*, vol. 14, p. 423, Jul. 2020.
[49] S. N. Truong, "Single crossbar array of memristors with bipolar inputs for neuromorphic image recognition," *IEEE Access*, vol. 8, pp. 69327–69332, 2020, doi: 10.1109/ACCESS.2020.2986513.
[50] P.-Y. Chen, B. Lin, I.-T. Wang, T.-H. Hou, J. Ye, S. Vrudhula, J.-S. Seo, Y. Cao, and S. Yu, "Mitigating effects of non-ideal synaptic device characteristics for on-chip learning," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2015, pp. 194–199, doi: 10.1109/ICCAD.2015.7372570.
[51] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, Feb. 2012, Art. no. 075201, doi: 10.1088/0957-4484/23/7/075201.
[52] M. J. Sharifi and Y. M. Banadaki, "General SPICE models for memristor and application to circuit simulation of memristor-based synapses and memory cells," *J. Circuits, Syst. Comput.*, vol. 19, no. 2, pp. 407–424, Apr. 2010, doi: 10.1142/S0218126610006141.
[53] H. Kim and B.-G. Park, "Solving overlapping pattern issues in on-chip learning of bio-inspired neuromorphic system with synaptic transistors," *Electronics*, vol. 9, no. 1, p. 13, Dec. 2019, doi: 10.3390/electronics9010013.
[54] W.-M. Kang, C.-H. Kim, S. Lee, S. Y. Woo, J.-H. Bae, B.-G. Park, and J.-H. Lee, "A spiking neural network with a global self-controller for unsupervised learning based on spike-timing-dependent plasticity using flash memory synaptic devices," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, doi: 10.1109/IJCNN.2019.8851744.
[55] S. Hwang, H. Kim, J. Park, M.-W. Kwon, M.-H. Baek, J.-J. Lee, and B.-G. Park, "System-level simulation of hardware spiking neural network based on synaptic transistors and I&F neuron circuits," *IEEE Electron Device Lett.*, vol. 39, no. 9, pp. 1441–1444, Sep. 2018, doi: 10.1109/LED.2018.2853635.

SEONGBIN OH received the B.S. degree in electrical and computer engineering from Seoul National University (SNU), Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree. He is also with the Inter-University Semiconductor Research Center, SNU. His current research interests include neuromorphic systems and its application in computing.

DONGSEOK KWON received the B.S. degree in electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Seoul National University (SNU), Seoul, South Korea. His current research interests include neuromorphic systems and its application in computing.

GYUHO YEOM received the B.S. degree in electrical and computer engineering from Seoul National University (SNU), Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree. His current research interests include neuromorphic systems and its application in computing.



WON-MOOK KANG received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Seoul National University, Seoul. His current research interests include neuromorphic systems and its application in computing.

SOOCHANG LEE received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2016, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. His current research interests include neuromorphic systems and its application in computing.

SUNG YUN WOO received the B.S. degree in electrical engineering from Kyungpook National University, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Seoul National University (SNU), Seoul, South Korea. He is also with the Inter-University Semiconductor Research Center, SNU. His current research interests include neuromorphic systems and neural networks.

JAEHYEON KIM received the B.S. degree in materials science and engineering from Yonsei University (YU), in 2020. He is currently pursuing the M.S. degree with the Department of Electrical and Computer Engineering, Seoul National University, Seoul. His current research interests include neuromorphic systems and its application in computing.

JONG-HO LEE (Fellow, IEEE) received the Ph.D. degree in electronic engineering from Seoul National University (SNU), Seoul, South Korea, in 1993. He was a Postdoctoral Fellow with the Massachusetts Institute of Technology, Cambridge, MA, USA, from 1998 to 1999. He has been a Professor with the School of Electrical and Computer Engineering, SNU, since 2009. He is also a Lifetime Member of the Institute of Electronics Engineers of Korea (IEEK).