IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 21, 2022, accepted March 7, 2022, date of publication March 16, 2022, date of current version March 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3160271

On-Chip Trainable Spiking Neural Networks Using Time-To-First-Spike Encoding

JISEONG IM, JAEHYEON KIM, HO-NAM YOO, JONG-WON BAEK, DONGSEOK KWON, (Graduate Student Member, IEEE), SEONGBIN OH, JANGSAENG KIM, JOON HWANG, BYUNG-GOOK PARK, (Fellow, IEEE), AND JONG-HO LEE, (Fellow, IEEE)

Department of Electrical and Computer Engineering 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 BK21 FOUR Program of the Education and Research Program for Future Information Communication Technology (ICT) Pioneers, Seoul National University, in 2021; in part by the Technology Innovation Program under Grant 20009972 through the Ministry of Trade, Industry and Energy (MOTIE), South Korea; in part by the National Research and Development Program through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT under Grant 2021M3FA2A02037889; and in part by SK Hynix.

ABSTRACT Artificial Neural Networks (ANNs) have shown remarkable performance in various fields. However, ANN relies on the von-Neumann architecture, which consumes a lot of power. Hardware-based spiking neural networks (SNNs) inspired by a human brain have become an alternative with significantly low power consumption. In this paper, we propose on-chip trainable SNNs using a time-to-first-spike (TTFS) method. We modify the learning rules of conventional SNNs using TTFS to be suitable for on-chip learning. Vertical NAND flash memory cells fabricated by a device manufacturer are used as synaptic devices. The entire learning process considering the hardware implementation is also demonstrated. The performance of the proposed network is evaluated through the MNIST classification in system-level simulation using Python. The proposed SNNs show an accuracy of 96% for a network size of 784 – 400 – 10. We also investigate the effect of non-ideal cell characteristics (such as pulse-to-pulse and device-to-device variations) on inference accuracy. Our networks demonstrate excellent immunity for various device variations compared with the networks using off-chip training.

INDEX TERMS Spiking neural networks (SNNs), time-to-first-spike (TTFS), on-chip training, synaptic devices, NAND flash.

I. INTRODUCTION

Recently, artificial neural networks (ANNs) have demonstrated superior performance in various fields, such as classification tasks, pattern recognition, and detection tasks [1]–[3]. As the demand for ANNs increases, the limitations of the conventional von Neumann architecture, such as training speed and power consumption, have become a major concern [4]. Various researches have been conducted to overcome these issues, including digital accelerators [5] or efficient learning algorithms [6]. However, the conventional von Neumann architecture has fundamental limitations in terms of power consumption and time required for memory access [7]. As an alternative, hardware-based spiking neural networks (SNNs) that use analog synaptic devices have emerged with advantages in terms of power consumption and operation time [8]–[10].

Spiking Neural Networks mimic the behavior of the human brain, which consists of numerous neurons and synapses [11]. In the SNNs, neurons communicate with adjacent neurons by generating spikes and transmitting those via synapses. Each neuron integrates the spikes propagated from the preceding neurons as a form of its membrane potential. When the membrane potential exceeds the threshold of the neuron, a spike is generated and transmitted to the post neuron [12]. The spikes can contain information in two general forms: firing rate and firing time [13].

By using firing time as the information-carrying quantities, the network can operate with a small number of spikes. Compared to rate coding, where the spiking rate of a neuron encodes an analog value of ANNS, the temporal-based networks can be implemented more power-efficient on neuromorphic hardware since it reduces the number of spikes [14].

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

However, the temporal-based networks are not suitable for applying the conventional learning rules of ANNs, as they convert analog values of ANNs into temporal formats. To implement the temporal-based SNNs, various approaches have been proposed, such as methods using derivatives of temporal values [15], alpha synaptic function [16], and dynamic target firing time [17]. Although these attempts have achieved remarkable results in terms of network performance, an additional software-based computation is required in the learning process due to their complex learning algorithms. Therefore, these methods are unsuitable for implementing on-chip training, which trains the network by applying update pulses to synaptic devices in the hardware level [18] without ANN-to-SNN conversion. This reveals the limitations of conventional approaches in terms of power consumption.

In this paper, we propose an on-chip trainable temporal-based SNNs. While conventional training methods are complicated to implement in hardware, simplified methods such as static target firing time and constant denominator for gradient normalization are applied. The conductance characteristics of the synaptic devices are obtained from the results measured from cells in vertical NAND flash memory cell strings. The holistic process of the proposed network consists of 5 phases: 1 forward phase, 2 backpropagation phases, and 2 update phases. Schematic circuitry to generate the error value between the output spike and the teaching signal is also proposed. The performance of the proposed network is evaluated at the system level by classifying the MNIST dataset, and the power efficiency of the network is estimated through the total amount of synaptic weight updates. Furthermore, the effects of variations occurred by non-ideal characteristics of the synaptic devices are evaluated in three types: a pulse to pulse variation, a device to device variation, and a stuck-at-off ratio [19].

This paper is organized as follows. Section II contains the measured characteristics of cells in a cell string of vertical NAND flash memory and the proposed algorithms to train the network. Section III presents a scheme to implement the proposed network in hardware. Section IV provides simulation results and discussion. Finally, section V provides a summary and a conclusion of our research.

II. SYNAPTIC DEVICES AND LEARNING METHODS

A. VERTICAL NAND FLASH MEMORY

In this work, cells in cell strings of vertical NAND (VNAND) flash memories manufactured by a memory company are used as synaptic devices. Each string contains multiple word-line (WL) cells and two select-line transistors. A schematic view of the VNAND flash string and the bias condition is demonstrated in Fig. 1 (a). The center WL cell is erased and programmed to demonstrate the long-term potentiation (LTP) and long-term depression (LTD) characteristics. In the LTP process, the GIDL (Gate Induced Drain Leakage) mechanism initiated by the erase pulse provides holes to the selected WL cell and lowers its threshold voltage [20]. For the LTD process, the program pulse initiating FN-tunneling is applied to the gate of the selected WL cell.

The measured LTP/LTD characteristics of the synapse are presented in Fig. 1 (b). Both LTP and LTD were conducted for five types of the update pulse width. A total of 40 pulses, 20 program pulses, and 20 erase pulses were applied, respectively. Both erase and program processes show non-linear conductance behaviors. Each inset presents the number of required pulses to obtain the same amount of conductance change as when the unit pulse is applied 20 times. Since the pulse width and the number of pulses have an inverse proportional relationship, the amount of conductance change is proportional to the update pulse width. Therefore, weight updates can be implemented in hardware by modulating with a pulse width proportional to the delta value stored in each neuron. The detailed process is covered in III.

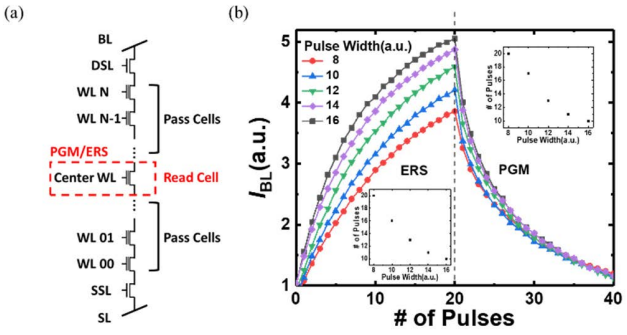

FIGURE 1. (a) Schematic view of VNAND flash string used as synaptic devices. (b) Measured LTP/LTD characteristics of the synaptic device.

The conductance behavior of synaptic devices is generally expressed as follows [21],

$$G(x) = a + \frac{1}{\beta}\ln(x + c), \tag{1}$$

where G is the conductance of the device, β is the non-linearity factor, and x is the number of the applied pulses. a and c are the fitting parameters. The conductance characteristics of the measured device are fitted with a non-linearity factor equal to 2.434 in the LTP process and 3.504 in the LTD process.

B. LEARNING METHODS

We employed time-to-first-spike (TTFS) method as the encoding rule and the modified learning methods of the previous work [17]. Consider a gray image with pixel values ranging from 0 to $I_{\max}$. The intensity of each pixel is converted to a single spike spiking at a specific time from each corresponding input neuron. Spike time of i^{th} neuron t_i is calculated from the pixel intensity I_i as follows:

$$t_i = \left| \frac{I_{max} - I_i}{I_{max}} \cdot t_{max} \right|, \tag{2}$$

where I_{max} is 255 and t_{max} is 511.

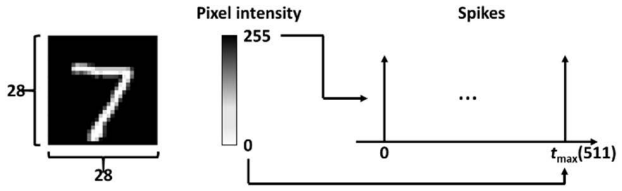

FIGURE 2. A schematic illustration for temporal encoding. Each pixel intensity is converted to a single spike. A pixel with the higher intensity corresponds to a spike that fires earlier.

Fig. 2 demonstrates the schematic description of the temporal encoding rule. Assuming each neuron only fires once per an input image, spike train S_i of i^{th} input neuron is defined as follows:

$$S_i(t) = \begin{cases} 1 & if\ t = t_i \\ 0 & otherwise. \end{cases} \tag{3}$$

Spikes generated from the input layer are multiplied by synaptic weights and integrated into the neurons in the next layer following the non-leaky I&F (Integrate and Fire) model [22]. Each neuron and synapse is a fully connected. The membrane potential V_j^l of j^{th} neuron in layer l can be defined as

$$V_j^l(t) = \sum_0^t \sum_i^{N^{l-1}} w_{ji} S_i^{l-1}(\tau), \tag{4}$$

where N^{l-1} is the number of neurons in layer $l-1$, and w_{ji} is the synaptic weight connecting j^{th} neuron in layer l and i^{th} neuron in layer $l-1$. The neuron fires when the membrane voltage exceeds its threshold. Spike train of j^{th} neuron in layer l is expressed as follows,

$$S_j^l(t) = \begin{cases} 1 & if\ V_j^l(t) > \theta_j^l\ and\ h_j^l(t) = 0 \\ 0 & otherwise, \end{cases} \tag{5}$$

$$h_j^l(t) = \sum_0^{t-1} S_j^l(\tau), \tag{6}$$

where θ_j^l is the threshold of the j^{th} neuron in layer l, and $h_j^l(t)$ indicate that the neuron did not fire before.

The neuron which fires the earliest in the output layer determines the label of the input image. To train the network that the output neuron for the right category fires first, we define the error e_i of the i^{th} neuron in the output layer and the loss as follows:

$$e_i = \begin{cases} \max\left(\dfrac{t_i - t_{target}}{t_{max}}, 0\right) & if\ i = target\ index \\ \max\left(\dfrac{t_{target} - t_i}{t_{max}}, 0\right) & otherwise, \end{cases} \tag{7}$$

$$L = \frac{1}{2}\sum_i^{N^0} e_i^2, \tag{8}$$

where t_i is the firing time of the output neuron, and t_{target} is the target firing time. The method to set the target firing time is presented at the end of this section. The synaptic weight connecting the i^{th} neuron in layer $l-1$ and the j^{th} neuron in layer l is updated as

$$w_{ji}^l = w_{ji}^l - \eta \frac{\partial L}{\partial w_{ji}^l}, \tag{9}$$

$$\begin{aligned} \frac{\partial L}{\partial w_{ji}^l} &= \frac{\partial L}{\partial t_j^l} \cdot \frac{\partial t_j^l}{\partial w_{ji}^l} \\ &= \frac{\partial L}{\partial t_j^l} \cdot \begin{cases} 0 & if\ h_j^l(t_{max}) = 0 \\ -\sum_0^{t_j^l} S_i^{l-1}(\tau) & otherwise, \end{cases} \end{aligned} \tag{10}$$

where η is the learning rate, and $\frac{\partial t_j^l}{\partial w_{ji}^l}$ in (10) is assumed following the previous work [17]. The delta value δ_j^l of the neuron is defined from the derivative of the loss as

$$\delta_j^l = \frac{\partial L}{\partial t_j^l}, \tag{11}$$

where t_i^l is the firing time of the neuron in layer l. If layer l is the output layer, the delta value is obtained as

$$\begin{aligned} \delta_j^l &= \frac{\partial L}{\partial t_j^l} = e_j^l \frac{\partial e_j^l}{\partial t_j^l} \\ &= \begin{cases} \begin{cases} e_j^l & if\ t_j^i > t_{target} \\ 0 & otherwise \end{cases} & if\ i = target\ index \\ \begin{cases} -e_j^l & if\ t_j^i < t_{target} \\ 0 & otherwise \end{cases} & otherwise, \end{cases} \end{aligned} \tag{12}$$

and when layer l is the hidden layer, the delta value is calculated by the backpropagation as follows [17],

$$\delta_j^l = \sum_k w_{jk}^{l+1} \delta_k^{l+1} h_j^l\left(t_k^{l+1}\right). \tag{13}$$

To prevent gradient exploding and vanishing, we normalize gradient by modulating the delta values. $L2$-norm is generally used as a normalization tool. However, it is challenging to implement the $L2$-norm in hardware [23]. Due to the aforementioned challenge, a novel method for normalization is proposed. Instead of $L2$-norm, predefined constant parameter r^l becomes a denominator of the normalization term. In the proposed network, r^l is the saturated value of $L0$-norm of the delta values during the training process. With a layer-wise hyperparameter r^l, the delta values are normalized as follows,

$$\delta_{j,norm}^l = \frac{\delta_j^l}{r^l}. \tag{14}$$

In (7), the functionality of the target firing time t_{target} is encouraging the target neuron to fire the first. As a result, the target neuron fires earlier at every training step, and the other neurons fire later. Various methods are utilized in previous works: cross-entropy loss [15] and the relative target firing time [17] to implement the update process. Although these attempts effectively train the network, these are not suitable for on-chip training due to the difficulty of hardware implementation. To obtain cross-entropy loss in hardware,

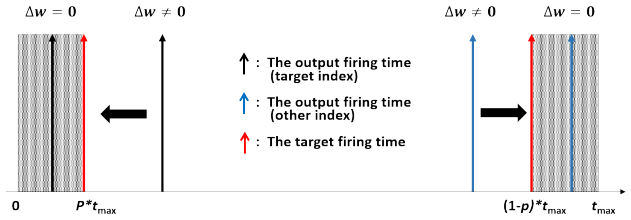

FIGURE 3. The cases of weight updates according to the position of the target firing time. Weights are not updated in the gray area.

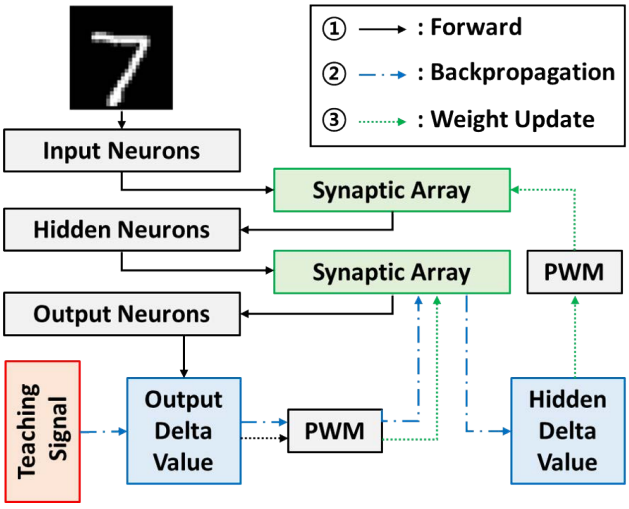

FIGURE 4. The block diagram of the hardware implementation scheme for the proposed network.

it necessitates the inclusion of circuitry for calculating exponential function [24], logarithmic function [25], and summation function [26]. If the target firing time is a dynamic parameter, additional circuits are required for sensing the firing time of the output neuron and generating the target signal according to the output spike.

In order to avoid the circuit complexity of the above methods, a constant target firing time is utilized in this paper. Given the functionality of the target firing time, the simplest way is to set the target firing time at the first time step for the target and the last time step for the others. However, it may lead the target neuron to fire earlier than most neurons in the input layer and hidden layer, resulting in information loss which degrades the performance of the network [16]. Therefore, we set the target firing time to be spaced apart from both ends of the time step by a specific interval as follows:

$$t_{target} = \begin{cases} p * t_{max} & for\ target\ index \\ (1-p) * t_{max} & otherwise, \end{cases} \tag{15}$$

where p is a hyperparameter with a value of between 0 and 1. Fig. 3 demonstrates the cases of weight update with the proposed target firing time t_{target}.

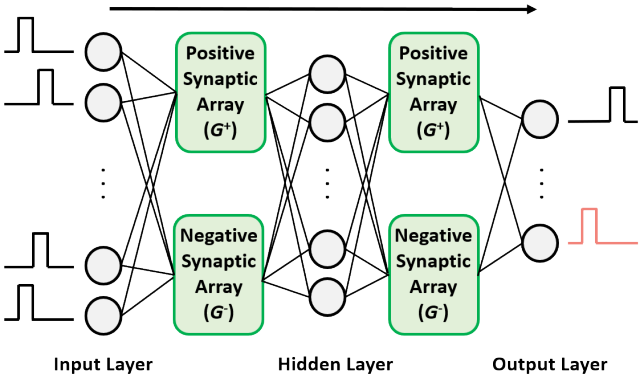

FIGURE 5. The structure of the proposed network. After the forward propagation, the label of the output neuron, which fires the fastest is considered as an answer.

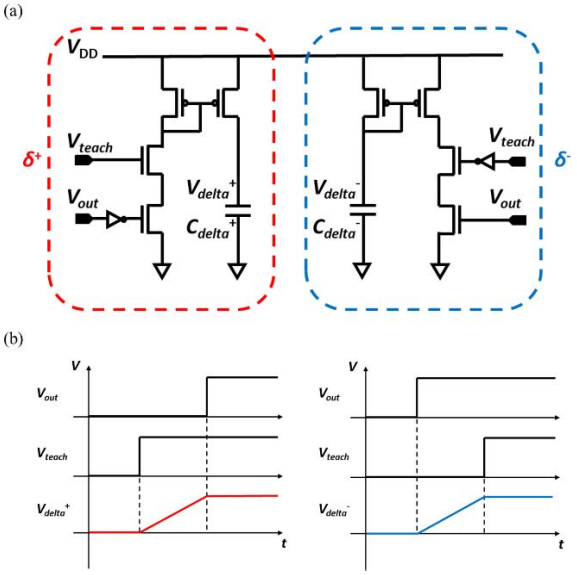

FIGURE 6. (a) The delta value generating circuit. The red dashed part stores the positive delta value, and the blue dashed part stores the negative delta value. (b) The characteristics of V_{delta} according to the output spike and the teaching signal.

III. A SCHEME FOR HARDWARE IMPLEMENTATION

Fig. 4 presents the block diagram of the hardware implementation scheme for the proposed network. The entire process consists of 5 phases: 1 forward phase, 2 backpropagation phases, and 2 update phases.

In the forward phase (① of Fig. 4), input signals encoded in the temporal form are integrated into the neurons of the input layer. When the membrane voltage of the input neuron exceeds the threshold, it fires and the spike propagates to the next layer through the synaptic array. The spikes eventually reach the output layer with this process. The synaptic array is divided into the G^+ synaptic array and the G^- synaptic array in order to represent a weight with a value from negative

to positive. The weight value w of a synapse is represented as follows,

$$w = G^+ - G^-. \tag{16}$$

The schematic diagram for the network structure is shown in Fig. 5.

At the output layer, the spikes of the output layer are transmitted to the delta value generating circuit (Fig. 6. (a)). The delta value generating circuit has two capacitors. Each capacitor is charged during the time difference between the output signal and the teaching signal. At the target firing time, the teaching signal is applied to the circuit. For the target output neuron, the teaching signal is applied to the red dashed part of Fig. 6. (a). Then, the voltage V^+_{delta} of the capacitor C^+_{deltaa} increases only if the teaching signal is prior to the output signal. For the other output neurons, the teaching signal is applied to the blue dashed part of Fig. 6. (a). The voltage V^-_{delta} of the capacitor C^-_{delta} increases when the teaching signal is posterior to the output signal. The voltage transitions across C^+_{delta} and C^-_{delta} are shown in Fig. 6. (b). These voltage values V^+_{delta} and V^-_{delta} represent the positive and the negative part of the delta value in Section II-B, respectively.

The stored delta values in the output layer are propagated to the hidden layer during the backpropagation phase (② of Fig. 4). After the forward phase, the delta value is converted to the magnitude of the voltage regardless of the sign. Since it is necessary to distinguish the sign of the delta value in the process of the backward weighted sum of the delta value, the backpropagation phase is performed twice by dividing the cases of positive and negative delta values.

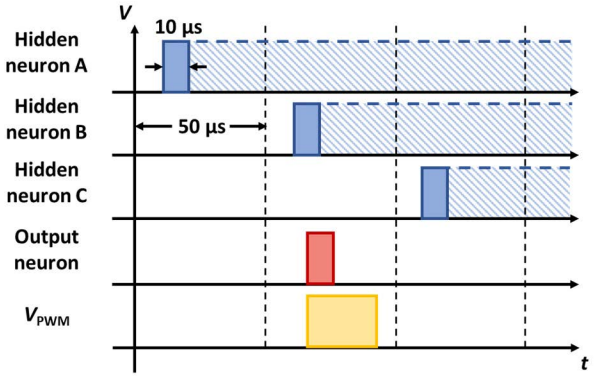

FIGURE 7. The pulse scheme for implementing a causal relationship between the hidden neurons and the output neurons in backpropagation phase. Since each hidden neuron accepts the backpropagated delta values only after it fires (blue dashed part). Therefore, hidden neurons that fire later than the output neuron do not receive the delta value from the output neuron.

In order to obtain the positive delta value of the hidden layer, the delta values of the target output neuron and the other neurons converted through pulse width modulator (PWM) are applied to the G^+ synaptic array and the G^- synaptic array, respectively. The weighted sum of the delta value should be positive in this case while the target output neuron always has the positive delta value and the others have negative delta value. In section II-B, only the hidden neurons that fire prior to the output neurons receive the delta value of the output neurons. The pulse scheme to implement this algorithm in hardware is demonstrated in Fig. 7. At the beginning of this phase, the input signal is applied again to the network. The pulse representing the delta value of the output neuron is emitted through the PWM circuit when the output neuron fires. When the hidden neuron fires, the switch connecting the delta capacitor of the hidden neuron and the synaptic array is closed. The output pulse width of the PWM circuit is 30μs when the delta value has a maximum value when the length of each time step is 50μs. According to this method, the hidden neuron receives the backpropagated delta value from the output neuron only when it fires earlier than the output neuron. In case of obtaining the negative delta value of the hidden layer, the delta values of the target output neuron and the other neurons converted through the PWM circuit are applied to the G^- synaptic array and the G^+ synaptic array, respectively. The rest of the process is the same as that of the positive delta value.

After obtaining all delta values of the hidden layer, the update phase (③ of Fig. 4) begins. Each synaptic device receives an update pulse corresponding to the delta value stored in the neuron connected backward. The update phase is divided into 2 sub-phases: weight increase sub-phase and weight decrease sub-phase, since the delta value is divided into each sign. In section II-B, weight update only occurs when the prior neuron fire earlier than the posterior neuron. Due to this property, the input signal is applied to the network to check the firing time of neurons without additional memory.

In the sub-phase for weight increase, the voltage across the capacitors that store the positive delta values of each neuron is converted to a pulse width through the PWM circuit when the neuron fire. The pulse output through the PWM circuit has a magnitude of a half of the erase bias V_{ERS} and the program bias V_{PGM}, and a width ($<30\mu$s) proportional to a delta value. These pulses are applied to the BLs and SLs of the VNAND flash cell strings in the G^+ synaptic array and the WLs of the G^- synaptic array connected in front of the neurons. At the same time, the negative pulses with a magnitude of a half of V_{ERS} and V_{PGM} are applied to the WLs of the G^+ synaptic array and the DLs/SLs of the G^- synaptic array connected behind of the neurons. In the G^+ synaptic array, the voltage across DLs/SLs and WLs generate GIDL and consequently strengthen the weights of the synapses. In the G^- synaptic array, FN-tunneling is occurred and depress the weights of the synapses. This scheme allows the synapse can be updated only when the prior neuron fires earlier than the posterior neuron. Fig. 8 demonstrates this pulse scheme. The weight increase in this sub-phase is accomplished with a potentiation of the G^+ synaptic array and a depression of the G^- synaptic array.

The sub-phase for weight decrease is conducted with the opposite mechanism that leads to a conductance decrease of

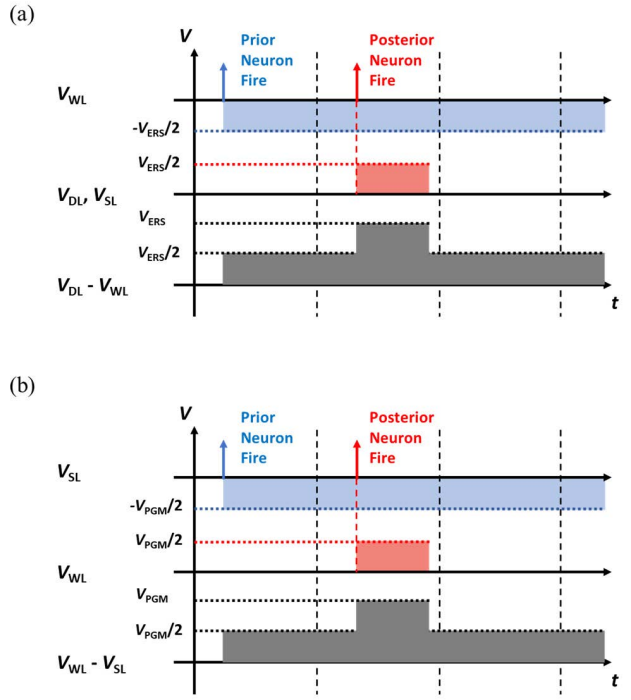

FIGURE 8. The pulse scheme for implementing a causal relationship between the neurons for (a) LTP process and (b) LTD process. If the prior neuron fires later than the posterior neuron, the synapse between the neuron is not updated.

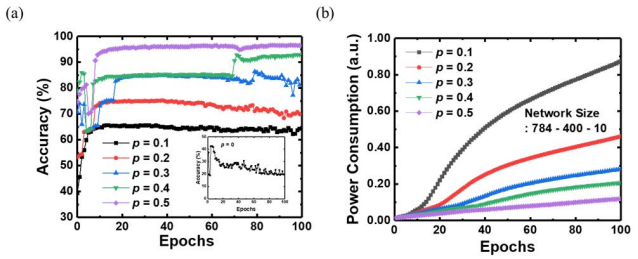

FIGURE 9. (a) The classification accuracy and (b) the approximated power consumption of the network with various p.

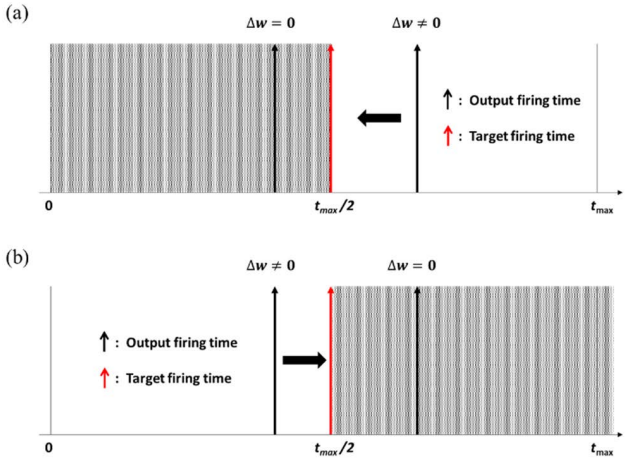

FIGURE 10. The cases of weight updates for (a) the target neuron and (b) the other neurons when p is 0.5. Note that the position of the target signal is identical in both types of neurons.

the G^+ synaptic array and a conductance increase of the G^- synaptic array. The pulses generated from the negative delta values are sequentially applied as program pulses of cells in the G^+ synaptic array (LTD process) and erase pulses of cells in the G^- synaptic array (LTP process). As a result, the weight decrease sub-phase is finished with G^+ decreases and G^- increases.

IV. RESULTS AND DISCUSSION

The performance of the proposed network is evaluated through the MNIST dataset classification. The system-level simulation for this task is conducted using Python. We implement the network with a size of 784 – 400 – 10. The threshold of a neuron is set to 100 uniformly and the learning rate is 0.2. The synaptic weights in the network reflect the measured device characteristics shown in Section II-A. The proposed learning rules for hardware implementation are adopted.

A. TARGET FIRING TIME

We evaluated the performance of the network with various p of (15). Fig. 9. (a) shows the classification accuracy of the network while p varies from 0.1 to 0.5 in 5 steps, and the inset of Fig. 9. (a) presents the case when p is 0. The inset shows that the accuracy of the network decreases with increasing training epochs, as mentioned in Section II-B. The accuracy of the network saturates to a particular value when p is non-zero. As the value of p increases, the performance of the network tends to improve. If p is greater than 0.5, there

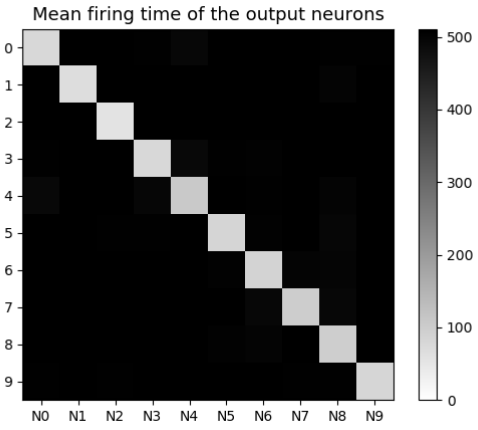

FIGURE 11. Confusion matrix of input indices and the mean firing time of the output neurons at a p of 0.5. Each neuron fires within 100 time steps only when the input signal with the correct index is applied.

is a time interval in which the firing time of the neuron of the target index overlaps the firing time of other neurons, so the simulation is not performed for this case. The best classification accuracy of the network is 96% at a p of 0.5.

Fig. 9. (b) shows the approximated power consumption of the network with various p. The power consumption is

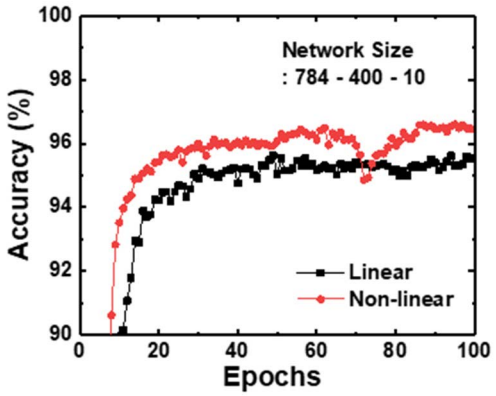

FIGURE 12. The classification accuracy of the networks with linear and non-linear characteristics of synaptic devices.

approximated from the total amount of the weight updates. When the value of p increases, the network consumes low power during the training process. As demonstrated in Fig. 9. (b), with a larger value of p, the weight update is terminated earlier, and the neurons store the smaller delta values. Therefore, in the case that p is 0.5, the network costs the least power consumption.

With the advantage in terms of classification accuracy and power consumption, we set the value of p to 0.5. Thus Fig. 3 is modified to Fig. 10. Furthermore, when p is 0.5, the target firing time of the target output neurons and the other output neurons are the same. Identical target firing time for all output neurons allows using same delta value generation circuit for all neurons, reducing circuit complexity for hardware implementation.

The confusion matrix of input indices and the mean firing time of the output neurons at a p of 0.5 is shown in Fig. 11. Each output neuron fires much earlier than the other neurons when the input signal corresponding to the index comes in, and fires later in other cases. The mean firing time of the output neurons for the input signal with the correct index is 84.5.

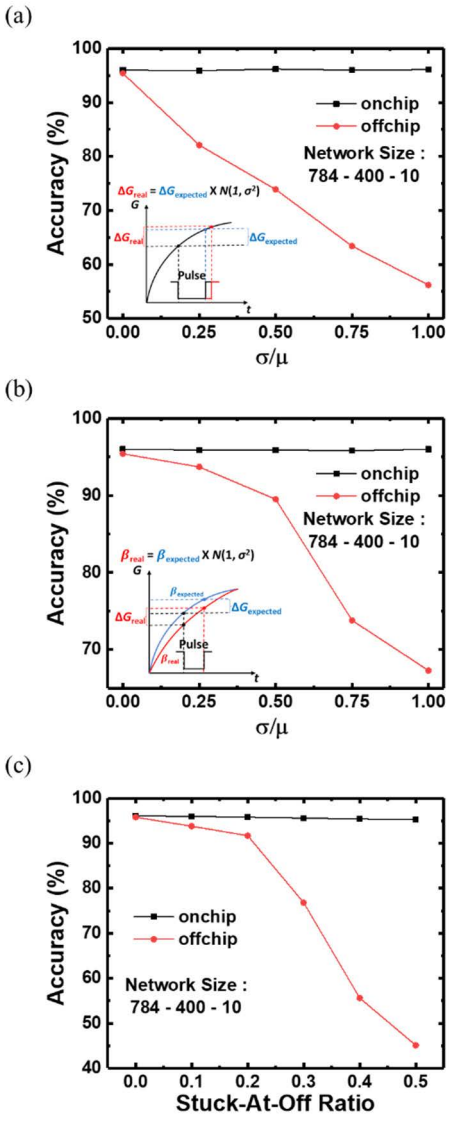

FIGURE 13. The classification accuracy of the networks under (a) the pulse-to-pulse variation, (b) the device-to-device variation, and (c) the stuck-at-off variation.

B. NON-LINEAR CHARACTERISTICS OF SINAPTIC DEVICES

The measured LTP and LTD characteristics of the synaptic device in Fig. 1 have a non-linear curve. To verify the effect of non-linear characteristics, we compare two cases of synapse characteristics: linear and non-linear, when the other conditions are identical. Fig. 12 presents the simulation results. When the synapse characteristic is non-linear, the network shows a slightly higher accuracy of 96%, compared to the accuracy of 95.4% with the linear characteristics. Due to the non-linear LTD nature, in a large weight range, the weight is updated by a relatively small amount. This kind of update can be viewed as a variant of weight decay regularization, which is well-known for improving training accuracy [27].

The conductance change of non-linear synaptic device becomes smaller as the conductance increase. This property restricts the device from reaching to excessive conductance level. From section II-B, the performance of the network can be degraded when the target neuron fires earlier than most of the input signals. The non-linear characteristic of the synaptic device may prevent this degradation by regulating weight increase.

When the non-linear LTP and LTD characteristics of VNAND cells are taken into account for the weights of the synapses, the accuracy exhibits large perturbations. According to Section III, for weight increase, G^+ synaptic device is erased and G^- synaptic device is programmed sequentially. Considering a weight is sufficiently strengthened during the training process, the G^+ synaptic device is fully erased and the G^- synaptic device is fully programmed. If weight decrease occurs in this situation, the G^+ synaptic

device needs to be programmed and the G^- synaptic device needs to be erased. Due to the LTP and LTD characteristics of VNAND cells, both updates result in large conductance changes with the fully erased G^+ synaptic device and the fully programmed G^- synaptic device. It seems to cause instability in the training curve demonstrated in Fig. 11.

C. EFFECTS OF DEVICE VARIATIONS

The intrinsic device variation is inevitable in hardware implementation [28]. Three types of the device variation are considered: pulse-to-pulse variation [29], device-to-device variation [30], and stuck-at-off variation [31]. We evaluate the tolerance of the proposed network to these variations by comparison with the off-chip learning scheme that uses ANNs-to-SNNs conversion. In the off-chip learning scheme, the values of the pre-trained weights are transferred to the conductance of devices after the training process.

The effect of pulse-to-pulse variation is shown in Fig. 13. (a). When the update pulse is applied to the synaptic device, a fluctuation of the pulse width is approximated along with a Gaussian distribution. The pulses are applied at each training step in our on-chip learning scheme, while they are applied only once at the end of the training process in off-chip learning scheme. As σ/μ is increased from 0 to 1, the network with the on-chip learning scheme shows immunity to the variation, whereas the accuracy significantly degrades with the off-chip learning scheme.

Fig. 13. (b) presents the effect of device-to-device variation. Synaptic devices in a synaptic array all have slightly different characteristics. We assume this variation by the non-linearity factor β along with a Gaussian distribution. The network with the off-chip learning scheme shows a small degradation than the case of pulse-to-pulse variation, while the network with the on-chip learning scheme still maintains its good performance.

We define the stuck-at-off ratio as the proportion of the number of stuck-at-off synaptic devices to the total number of synaptic devices. Note that 10% of synaptic devices have a conductance of 0 when the stuck-at-off ratio is 0.1. Fig. 13. (c) demonstrates the performance of the networks as the stuck-at-off ratio increases from 0 to 0.5. With the on-chip learning scheme, the performance of the network slightly degrades by 1% of accuracy when the stock-at-off ratio is 0.5. The accuracy of the network using the off-chip learning scheme decreases below 50% at the same stuck-at-off ratio. Overall, networks using the on-chip learning scheme are more tolerant of device variation than those using the off-chip learning scheme.

V. CONCLUSION

In this paper, we have proposed hardware implementable SNNs using TTFS encoding scheme. Modified learning methods, including constant target firing time and delta normalization method, were proposed considering hardware implementation. The entire process of the proposed network consists of 5 phases: 1 forward phase, 2 backpropagation phases, and 2 update phases. In the forward phase, input signals encoded by the TTFS method are applied to the network and transmitted to the next layers. Each backpropagation phase generates the positive and the negative delta values in all layers except the input layer. In update phases, the synaptic devices are updated by the delta value of the post neuron. Each update phase is responsible for the weight increase process and weight decrease process.

VNAND flash memory cells fabricated by a company were used as synaptic devices in this work. Measured characteristics of the device showed that the update pulse width is proportional to the conductance change. Based on this characteristic, we employed PWM to generate the pulses for the synaptic weight updates. The measured LTP and LTD behavior has non-linear conductance with a non-linearity factor (β) of 2.434 in the LTP process and a β of 3.504 in the LTD process.

The performance of the proposed training method was evaluated through the MNIST dataset classification. The system-level simulation using Python is conducted for 5 cases of the target firing time. The network has 784 input neurons, 400 hidden neurons, and 10 output neurons. After 100 epochs of the training process, the network presented the highest classification accuracy of 96% at a p of 0.5 (the target firing time is the middle of the total time step). As the value of p decreased, the accuracy of the network degraded. Additionally, the approximated power consumption of the network was the lowest at a p of 0.5. We assumed the power consumption of the network by the total amount of the synaptic weight updates.

We also investigated the effects of three types of non-ideal device variation: pulse-to-pulse, device-to-device, and stuck-at-off variations. The proposed on-chip trainable network was compared with the network using the off-chip learning scheme under the presence of device variation. For presented types of device variation, the proposed system showed excellent immunity compared to networks using the off-chip learning scheme.

ACKNOWLEDGMENT

(Jiseong Im and Jaehyeon Kim contributed equally to this work.)

REFERENCES

[1] S. Dodge and L. Karam, "Human and DNN classification performance on images with quality distortions," *ACM Trans. Appl. Perception*, vol. 16, no. 2, pp. 1–17, Apr. 2019, doi: 10.1145/3306241.

[2] M. Arozi, W. Caesarendra, M. Ariyanto, M. Munadi, J. D. Setiawan, and A. Glowacz, "Pattern recognition of single-channel sEMG signal using PCA and ANN method to classify nine hand movements," *Symmetry*, vol. 12, no. 4, p. 541, 2020, doi: 10.3390/sym12040541.

[3] A. Sharma, I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Syst. Appl.*, vol. 88, pp. 249–257, Dec. 2017, doi: 10.1016/j.eswa.2017.07.005.

[4] S. Petrenko, "Limitations of von Neumann architecture," in *Big Data Technologies for Monitoring of Computer Security: A Case Study of the Russian Federation*. Springer, 2018, pp. 115–173, vol. 2018, doi: 10.1007/978-3-319-79036-7_3.

[5] J. Ambrosi, A. Ankit, R. Antunes, S. R. Chalamalasetti, S. Chatterjee, I. E. Hajj, G. Fachini, P. Faraboschi, M. Foltin, S. Huang, W.-M. Hwu, G. Knuppe, S. V. Lakshminarasimha, D. Milojicic, M. Parthasarathy, F. Ribeiro, L. Rosa, K. Roy, P. Silveira, and J. P. Strachan, "Hardware-software co-design for an analog-digital accelerator for machine learning," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, 2018, pp. 1–13, doi: 10.1109/icrc.2018.8638612.

[6] J. C. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, Dec. 2009.

[7] 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.

[8] D. Querlioz, W. S. Zhao, P. Dollfus, J.-O. Klein, O. Bichler, and C. Gamrat, "Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches," in *Proc. Int. Symp. Nanosc. Archit.*, Apr. 2012, pp. 203–210, doi: 10.1145/2765491.2765528.

[9] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S.-P. Wong, "A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling," in *IEDM Tech. Dig.*, Dec. 2012, pp. 1–14, doi: 10.1109/IEDM.2012.6479018.

[10] H. Kim, S. Hwang, J. Park, and B.-G. Park, "Silicon synaptic transistor for hardware-based spiking neural network and neuromorphic system," *Nanotechnology*, vol. 28, no. 40, 2017, Art. no. 405202, doi: 10.1088/1361-6528/aa86f8.

[11] K. Liu, X. Cui, Y. Zhong, Y. Kuang, Y. Wang, H. Tang, and R. Huang, "A hardware implementation of SNN-based spatio-temporal memory model," *Frontiers Neurosci.*, vol. 13, p. 835, Jul. 2019, doi: 10.3389/fnins.2019.00835.

[12] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biol. Cybern.*, vol. 95, no. 1, pp. 1–19, 2006, doi: 10.1007/s00422-006-0068-6.

[13] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, Mar. 2019, doi: 10.1016/j.neunet.2018.12.002.

[14] S. Oh, D. Kwon, G. Yeom, W.-M. Kang, S. Lee, S. Yun Woo, J. Saeng Kim, M. Kyu Park, and J.-H. Lee, "Hardware implementation of spiking neural networks using time-to-first-spike encoding," 2020, *arXiv:2006.05033*.

[15] 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. 2017, doi: 10.1109/tnnls.2017.2726060.

[16] J. Ambrosi, A. Ankit, R. Antunes, S. R. Chalamalasetti, S. Chatterjee, I. E. Hajj, G. Fachini, P. Faraboschi, M. Foltin, S. Huang, W.-M. Hwu, G. Knuppe, S. V. Lakshminarasimha, D. Milojicic, M. Parthasarathy, F. Ribeiro, L. Rosa, K. Roy, P. Silveira, and J. P. Strachan, "Hardware-software co-design for an analog-digital accelerator for machine learning," in *Proc. Int. Conf. Rebooting Comput. (ICRC)*, 2018, pp. 1–13, doi: 10.1109/icassp40776.2020.9053856.

[17] 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, 2020, Art. no. 050027, doi: 10.1142/s0129065720500276.

[18] R. Hasan, T. M. Taha, and C. Yakopcic, "On-chip training of memristor crossbar based multi-layer neural networks," *Microelectron. J.*, vol. 66, pp. 31–40, Aug. 2017, doi: 10.1016/j.mejo.2017.05.005.

[19] 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, doi: 10.3389/fnins.2020.00423.

[20] Y. Komori, M. Kido, M. Kito, R. Katsumata, Y. Fukuzumi, H. Tanaka, Y. Nagata, M. Ishiduki, H. Aochi, and A. Nitayama, "Disturbless flash memory due to high boost efficiency on BiCS structure and optimal memory film stack for ultra high density storage device," in *IEDM Tech. Dig.*, Dec. 2008, pp. 1–8, doi: 10.1109/iedm.2008.4796831.

[21] 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.

[22] L. F. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Res. Bull.*, vol. 50, nos. 5–6, pp. 303–304, 1999, doi: 10.1016/s0361-9230(99)00161-6.

[23] M. Drakaki, G. Fikos, and S. Siskos, "Analog signal processing circuits using floating gate MOS transistors," in *Proc. Int. Conf. Technol. Autom.*, Oct. 2005, pp. 322–327.

[24] C.-H. Lin, T. C. Pimenta, and M. Ismail, "A low-voltage CMOS exponential function circuit for AGC applications," in *Proc. Brazilian Symp. Integr. Circuit Design*, Jan. 1998, p. 195, doi: 10.1109/sbcci.1998.715440.

[25] Z. Lang, H. Lamaire, and C. Han, "Integrated-circuit logarithmic arithmetic units," *IEEE Trans. Comput.*, vol. C-34, no. 5, pp. 475–483, Oct. 1985, doi: 10.1109/tc.1985.1676588.

[26] S. Franco, *Design With Operational Amplifiers and Analog Integrated Circuits*. New York, NY, USA: McGraw-Hill, 2002.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[28] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019, doi: 10.1038/s41586-019-1677-2.

[29] 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.

[30] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proc. IEEE*, vol. 106, no. 2, pp. 260–285, Feb. 2018, doi: 10.1109/jproc.2018.2790840.

[31] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Efficient and self-adaptive *in-situ* learning in multilayer memristor neural networks," *Nature Commun.*, vol. 9, no. 1, Dec. 2018, Art. no. 2385, doi: 10.1038/s41467-018-04484-2.

JISEONG IM received the B.S. degree in electrical and computer engineering from Seoul National University (SNU), Seoul, South Korea, in 2020, where he is currently pursuing the combined master's and Ph.D. degree. He is with the Inter University Semiconductor Research Center, SNU. His current research interest includes neuromorphic systems and its application in computing.

JAEHYEON KIM received the B.S. degree in materials science and engineering from Yonsei University, Seoul, South Korea, in 2020. He is currently pursuing the combined master's and Ph.D. degree with Seoul National University (SNU), Seoul. He is with the Inter University Semiconductor Research Center, SNU. His current research interest includes neuromorphic systems and its application in computing.

HO-NAM YOO received the B.S. and M.S. degrees in physics and astronomy from Seoul National University (SNU), Seoul, South Korea, in 2004 and 2009, respectively, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. He is with the Inter University Semiconductor Research Center, SNU. His current research interest includes neuromorphic systems and its application in computing.

JONG-WON BAEK received the B.S. degree in electrical and computer engineering from Seoul National University (SNU), Seoul, South Korea, in 2018, where he is currently pursuing the combined master's and Ph.D. degree. He is with the Inter University Semiconductor Research Center, SNU. His current research interest includes neuromorphic systems and its application in computing.

DONGSEOK KWON (Graduate Student Member, IEEE) 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 M.S. degree with the Department of Electrical and Computer Engineering, Seoul National University (SNU), Seoul, South Korea. His current research interest includes neuromorphic systems and its application in computing.

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 combined master's and Ph.D. degree. He is with the Inter University Semiconductor Research Center, SNU. His current research interest includes neuromorphic systems and its application in computing.

JANGSAENG KIM received the B.S. degree in electrical and computer engineering from Seoul National University (SNU), Seoul, South Korea, in 2018, where he is currently pursuing the combined master's and Ph.D. degree. He is with the Inter University Semiconductor Research Center, SNU. His current research interest includes neuromorphic systems and its application in computing.

JOON HWANG received the B.S. degree in electrical and computer engineering from Seoul National University (SNU), Seoul, South Korea, in 2020, where he is currently pursuing the combined master's and Ph.D. degree. He is with the Inter University Semiconductor Research Center, SNU. His current research interest includes neuromorphic systems and its application in computing.

BYUNG-GOOK PARK (Fellow, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University (SNU), Seoul, South Korea, in 1982 and 1984, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1990. He joined as an Assistant Professor with the Department of Electrical and Computer Engineering, SNU, in 1994, where he is currently a Professor.

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 at 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 a Lifetime Member of the Institute of Electronics Engineers of Korea (IEEK).