

Received October 29, 2020, accepted November 25, 2020, date of publication December 2, 2020, date of current version December 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041946

Area- and Energy-Efficient STDP Learning Algorithm for Spiking Neural Network SoC

GISEOK KIM¹, (Graduate Student Member, IEEE), KIRYONG KIM¹,
SARA CHOI^{1,2}, (Graduate Student Member, IEEE),
HYO JUNG JANG^{1,2}, (Graduate Student Member, IEEE),
AND SEONG-OOK JUNG¹, (Senior Member, IEEE)

¹School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

²Samsung Electronics Company Ltd., Yongin 17113, South Korea

Corresponding author: Seong-Ook Jung (sjung@yonsei.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) under Grant 2017R1A2B2006679 and Grant 2020M3F3A2A01081918.

ABSTRACT Recently, spiking neural networks have gained attention owing to their energy efficiency. All-to-all spike-time dependent plasticity is a popular learning algorithm for spiking neural networks because it is suitable for nondifferentiable spike event-based learning and requires fewer computations than back-propagation-based algorithms. However, the hardware implementation of all-to-all spike-time dependent plasticity is limited by the large storage area required for spike history and large energy consumption caused by frequent memory access. We propose a time-step scaled spike-time dependent plasticity to reduce the storage area required for spike history by reducing the area of the spike-time dependent plasticity learning circuit by 60% and a post-neuron spike-referred spike-time dependent plasticity to reduce the energy consumption by 99.1% by efficiently accessing the memory while learning. The accuracy of Modified National Institute of Standards and Technology image classification degraded by less than 2% when both time-step scaled spike-time dependent plasticity and post-neuron spike-referred spike-time dependent plasticity were applied. Thus, the proposed hardware-friendly spike-time dependent plasticity algorithms make all-to-all spike-time dependent plasticity implementable in more compact areas while reducing energy consumption and experiencing insignificant accuracy degradation.

INDEX TERMS Spike-time dependent plasticity (STDP), time-step scaled STDP (TS-STDP), post-neuron spike-referred STDP (PR-STDP), spiking neural network (SNN).

I. INTRODUCTION

Artificial intelligence (AI) algorithms have developed rapidly in the last decade. As the reliability of these algorithms increases, many applications such as the Internet of Things [1], [2], smart factories [3], [4], and smart mobility have been presented. Some of these require immediate processing of data generated by an edge device to extract meaningful information. Previously, data obtained from an edge device was sent to the server and processed using a prelearned AI network [5]. However, a delay existed in the communication between the device and the server. In addition, learning using the data generated by the edge devices could not be performed immediately because only previously

collected data were used. Therefore, on-chip learning that enables learning and processing from the data generated by the edge devices is required. Well-known algorithms such as convolutional neural networks (CNNs) [6]–[8] and deep neural networks (DNNs) [9], [10] exhibit excellent performance but consume high power owing to the enormous number of computations. Thus, inspired by the significantly lower power consumption of the human brain compared to its computational capability [11], researchers have suggested the spiking neural network (SNN), a new algorithm that mimics the behavior of nerve cells.

SNN uses the relative time difference between spikes for computation; hence, the numerical data is transformed into the temporal information of the spikes [12]. These spikes are one-bit data, which reduces the computational workload, leading to low power consumption [13]. Stimuli, such as

The associate editor coordinating the review of this manuscript and approving it for publication was Zhipeng Cai¹.

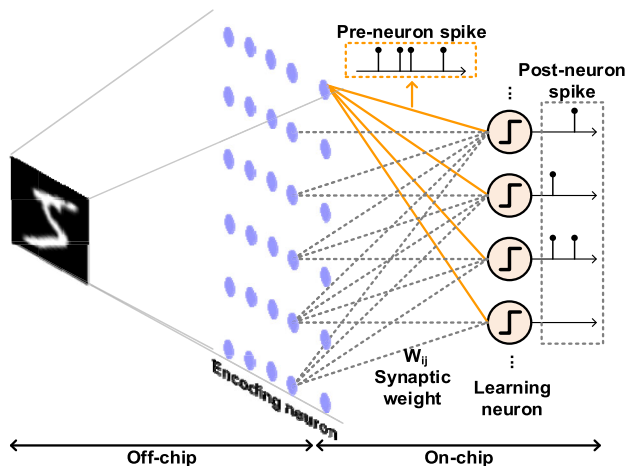


FIGURE 1. Concept of SNN and its structure.

the pixel data of an image or sound, are inputted into the encoding neuron, depicted as grey-colored neurons in Fig. 1, and encoded into a pre-neuron spike train whose rate is proportional to the intensity of the input stimuli [14]. The spike train is transmitted from the encoding neuron to the learning neuron through synapses. The product value of the spike train value and the synaptic weight is accumulated to the learning neuron potential. When the learning neuron potential exceeds the threshold, the learning neuron fires the post-neuron spike.

There are several learning algorithms in SNN [15], among which spike-time dependent plasticity (STDP) [16], [17] is a representative learning algorithm. The unsupervised learning algorithm STDP has drawn attention [18] because it can learn from data without labels (which is suitable for on-chip learning with data from the edge node) [19], [20]. STDP is a biological learning model inspired by the actual synaptic activity in the brain. It is a simple algorithm that does not require a multiplier because it considers only the time difference between pre- and post-spikes. Thus, it is preferred from the perspective of on-chip implementation. However, a large storage area is required to store the substantial time information, leading to difficulty in hardware implementation. In addition, STDP operates whenever the pre- or post-spike is fired as a trigger, which causes frequent memory accesses for updating the synaptic weight and thus leads to high power consumption. Thus, the following should be carefully considered to appropriately implement STDP on a chip: 1) reduction in the amount of time information of spikes to be stored and 2) reduction in the number of memory accesses required for updating the synaptic weight.

In this paper, we propose time-step scaled STDP (TS-STDP), which reduces the area for storing spike history by quantizing several time steps and post-neuron spike-referred STDP (PR-STDP), which reduces the number of memory accesses by using a post-neuron spike as a trigger for the learning process to save energy. The remainder of this paper is organized as follows. Section II describes the conventional

STDP with its challenges. The STDP algorithm is proposed to address these challenges in Section III. Section IV describes the circuit implementation of the conventional and proposed STDP algorithms. Simulation results on energy and accuracy with the Modified National Institute of Standards and Technology (MNIST) dataset and the estimated area improvement are discussed in Section V. Section VI provides the conclusion.

II. SPIKE-TIME DEPENDENT PLASTICITY

The basic operation of STDP is dependent on the time difference between pre- and post-neuron spikes. If the pre-neuron spike arrives at the learning neurons before the post-neuron spikes fire, then the synapses connecting the encoding neurons and learning neurons are considered to be related to the firing of the post-neuron spike. Therefore, the synaptic weights corresponding to the pre-neuron spikes that contribute to firing the post-neuron spikes are increased. However, if the pre-neuron spike arrives at the learning neurons after the post-neuron spike firing, then the synapses are not considered to be related to the firing of the post-neuron spike. Thus, the synaptic weights of these synapses are decreased. The increment and decrement in the synaptic weight is determined by the time difference between the pre- and post-neuron spikes. When the post-neuron spike is fired, the time difference between the past pre- and post-neuron spikes is compared. Accordingly, the increment in the synaptic weight is calculated using (1). This weight increasing process is called long-term potentiation (LTP). The decrease in the synaptic weight occurs when the pre-neuron spike arrives. The decrement in the synaptic weights is calculated using (2), according to the past temporal information of post-neuron spikes. This process is called long-term depression (LTD).

$$\Delta w_{LTP} = A_+ \cdot \exp\left(\frac{t_{post} - t_{pre}}{\tau}\right) (t_{post} > t_{pre}) \quad (1)$$

$$\Delta w_{LTD} = A_- \cdot \exp\left(\frac{t_{pre} - t_{post}}{\tau}\right) (t_{pre} > t_{post}) \quad (2)$$

t_{pre} and t_{post} denote the times when the pre- and post-neuron spikes are fired, respectively. A_+ and A_- denote constants representing the learning rates for LTP and LTD, respectively. The absolute values of A_+ and A_- are the same, but their signs are different. The learning rate and time constant τ can be set empirically according to the input data pattern. When the pre-neuron spike is fired before the post-neuron spike ($t_{post} > t_{pre}$), the weight change of LTP (Δw_{LTP}) is obtained using (1). Conversely, the weight change of LTD (Δw_{LTD}) is obtained using (2).

For the digital implementation, the STDP functions of LTP and LTD, described in (1) and (2), must be quantized into discrete values, as shown in Fig. 2. The quantized STDP can have several steps. However, a multi-step STDP is not necessary if the learning rate is within a reasonable range because the performance of the system does not degrade even when

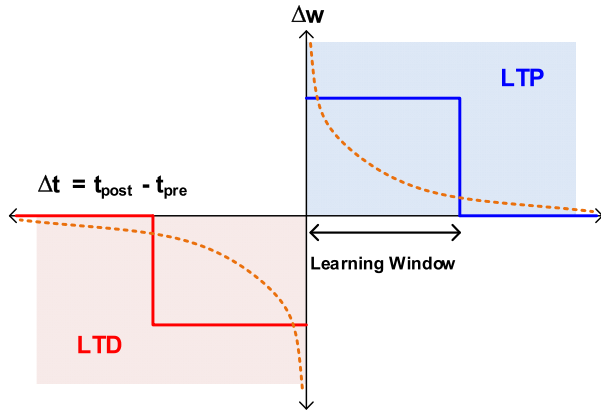


FIGURE 2. One-step quantized STDP function.

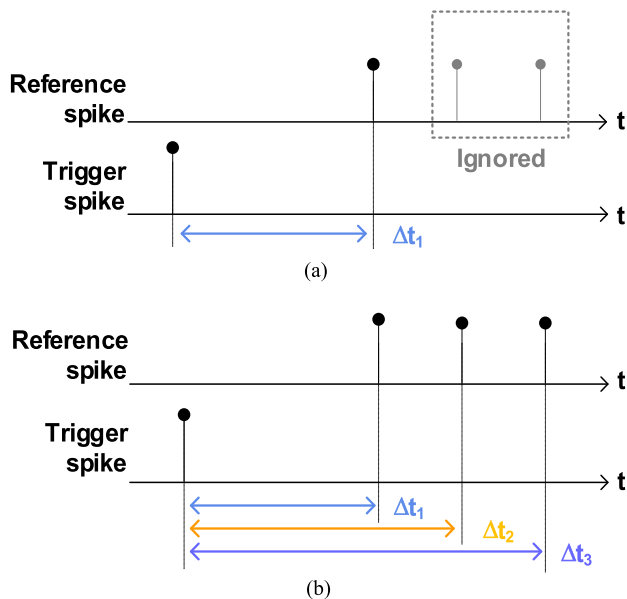


FIGURE 3. Difference of time referring of (a) nearest-neighbor and (b) all-to-all STDP.

using a one-step STDP, and furthermore, the complexity of the computation decreases [21].

The range of Δt ($= t_{post} - t_{pre}$) for nonzero Δw can be defined as the learning window. The time information of the pre- and post-neuron spikes within the learning window stored in the spike history is referred to when the other post- or pre-neuron spikes trigger the LTP or LTD process, respectively. There are two ways of referring the time information of spikes for the STDP process when calculating Δw using the stored time information: nearest-neighbor and all-to-all manners.

A. NEAREST-NEIGHBOR STDP (NN-STDP)

Nearest-neighbor STDP refers to only one temporally nearest spike at that time for updating the synaptic weight when the trigger spike (pre- or post-neuron spike that triggers LTP or LTD processes, respectively) occurs, as illustrated in Fig. 3. (a). NN-STDP requires few hardware for storage

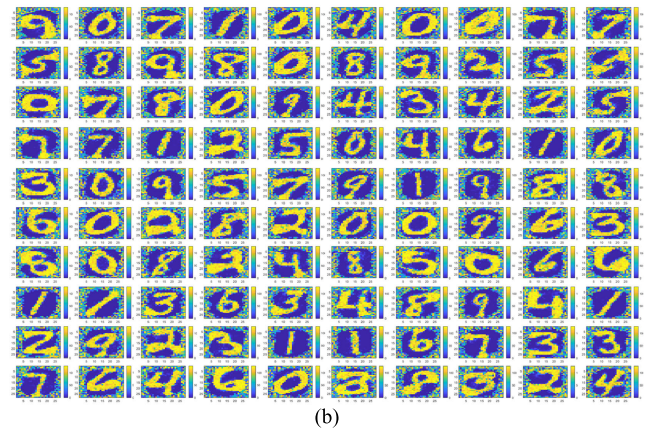
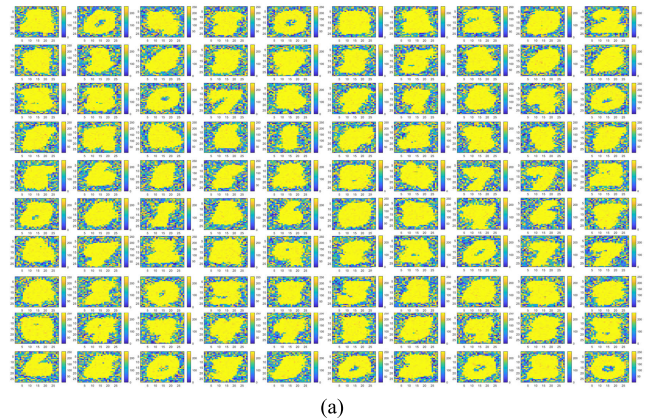


FIGURE 4. Weight reconfiguration of (a) nearest-neighbor and (b) all-to-all STDP.

because the information of only the nearest spike in the learning window is stored instead of all the spikes. However, NN-STDP is inaccurate because the loss of the temporal information of spikes is not negligible. NN-STDP does not consider the information of all the spikes in the learning window, and hence, it cannot reflect all temporal correlations. A neuron is more likely to fire when the spikes from different synapses simultaneously enter the neuron. In this situation, the synapse with a high firing rate of the pre-neuron spike has the same number of weight changes as the synapse with a low firing rate when both spikes are fired right before the trigger spike. Even though the synapse with a higher firing rate is more involved in the spike, the difference in contribution between the two synapses cannot be reflected. Therefore, the past contribution of the synapse with a higher firing rate of the pre-neuron spike is ignored when the time information is referred for the synaptic weight update.

B. ALL-TO-ALL STDP

All-to-all STDP refers to all the spikes in the learning window, as illustrated in Fig. 3. (b); hence, it accumulates the time information of all the spikes in the learning window. Thus, it can avoid the problem of missing temporal correlations that was observed in NN-STDP. The reconfigured images of the synaptic weights trained with the NN-STDP algorithm are ambiguous, as depicted by the simulated results in Fig. 4. (a).

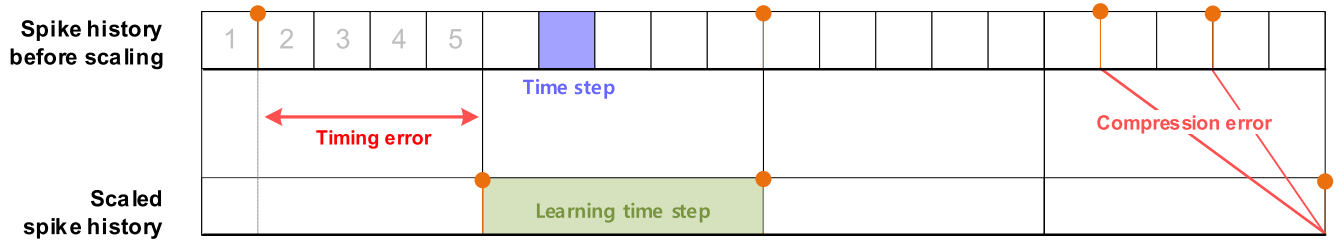


FIGURE 5. Effect of time-step scaling, timing error, and compression error when $M = 5$.

Conversely, the shapes of the synaptic weights trained with all-to-all STDP are clear.

To refer to all the temporal information, all the spike histories within the learning window must be stored. Furthermore, the learning window typically covers a wide range of time steps, and the spike pattern is irregular, which increases the hardware area for the spike history. Thus, it is far more difficult to implement all-to-all STDP than NN-STDP in hardware. Even though NN-STDP requires less storage to store only the history of the nearest spike, from an algorithm perspective, the accuracy of NN-STDP is lower than that of all-to-all STDP, and the reconfigured image shows that the neurons are not distinctively learned. Therefore, it is necessary to make all-to-all STDP hardware-friendly by reducing the area and power consumption, with insignificant accuracy degradation.

III. PROPOSED STDP ALGORITHM

We focused on two main aspects of hardware implementation of the STDP algorithm: area and energy. First, the area is mainly occupied by the storage required for the spike history. Therefore, we propose the time-step scaled STDP (TS-STDP) to reduce the area of storage for spike history. Second, to reduce the energy while learning, which is dominantly determined by the number of memory accesses, the post-neuron spike-referred STDP (PR-STDP) is proposed. PR-STDP reduces the number of memory accesses and simplifies the learning hardware.

A. TIME-STEP SCALED STDP (TS-STDP)

The number of memory elements required to store the spike history is proportional to the number of time steps of the learning window. If the learning window consists of N time steps, then N memory elements representing each time step are required to store the spike timing information in the corresponding bits for each neuron. Thus, the number of time steps of the learning window needs to be reduced to reduce the storage area. However, the size of the learning window, which is expressed by N time steps, cannot be modulated because the original performance of the overall system must be maintained. Therefore, to reduce the number of time steps while maintaining the size of the learning window, we propose TS-STDP. By quantizing M time steps into one learning time step, the learning window can be represented by M times fewer flip-flops while maintaining the original size

of the learning window. Owing to quantization, more than two spikes in the same learning time step could be treated as one spike. To prevent several situations in which multiple spikes are treated as one spike, the temporal distance between the spikes can be controlled by changing the mapping of the numerical data to rate conversion. However, the scaled timing information of spikes could have errors due to quantization. These errors could affect the STDP function and weight change. As depicted in Fig. 5, the spikes that occur within the learning time step are aligned to the edge of the learning time step when they are stored in the spike history. That is, spikes that fire between 1 to M time steps after the current time step are treated as fired in the first learning time step. Thus, the original timing information of the spike that occurs immediately before the edge of the past learning time step has an error of $M-1$ time steps compared to its original timing information when stored in a scaled spike history. In Fig. 5, the last learning time step shows that four time steps of timing error occur when the spike at the first time step is stored in the scaled spike history. If the spike arrives near the edge of the learning time step, the spike information stored in the scaled spike history is similar to the original information. This error in timing information can increase or decrease the weight change. However, owing to the use of one-step STDP and the stochastic learning property, this error in timing information does not affect the performance when M is within a certain range of values. This value is determined by the empirical method that minimizes performance degradation of the entire system based on simulation results. Thus, by suitably choosing the value of M , the hardware resources for storing the spike history can be significantly reduced with an insignificant accuracy degradation, thereby rendering all-to-all STDP implementable on a smaller area.

B. POST-NEURON SPIKE-REFERRED STDP (PR-STDP)

In the conventional STDP approach, both post- and pre-neuron spikes are used as trigger spikes of the LTP and LTD, respectively, as illustrated in Fig. 6. (a). If the trigger spikes in LTP and LTD are different, two problems occur in terms of hardware implementation. First, the synaptic weights associated with the pre- and post-neuron spikes are stored in an orthogonal direction. Thus, the memory access direction changes depending on the type of trigger spike. To support this bidirectional memory access, memory cells for the synaptic weights must be transposable or accessed row-by-row

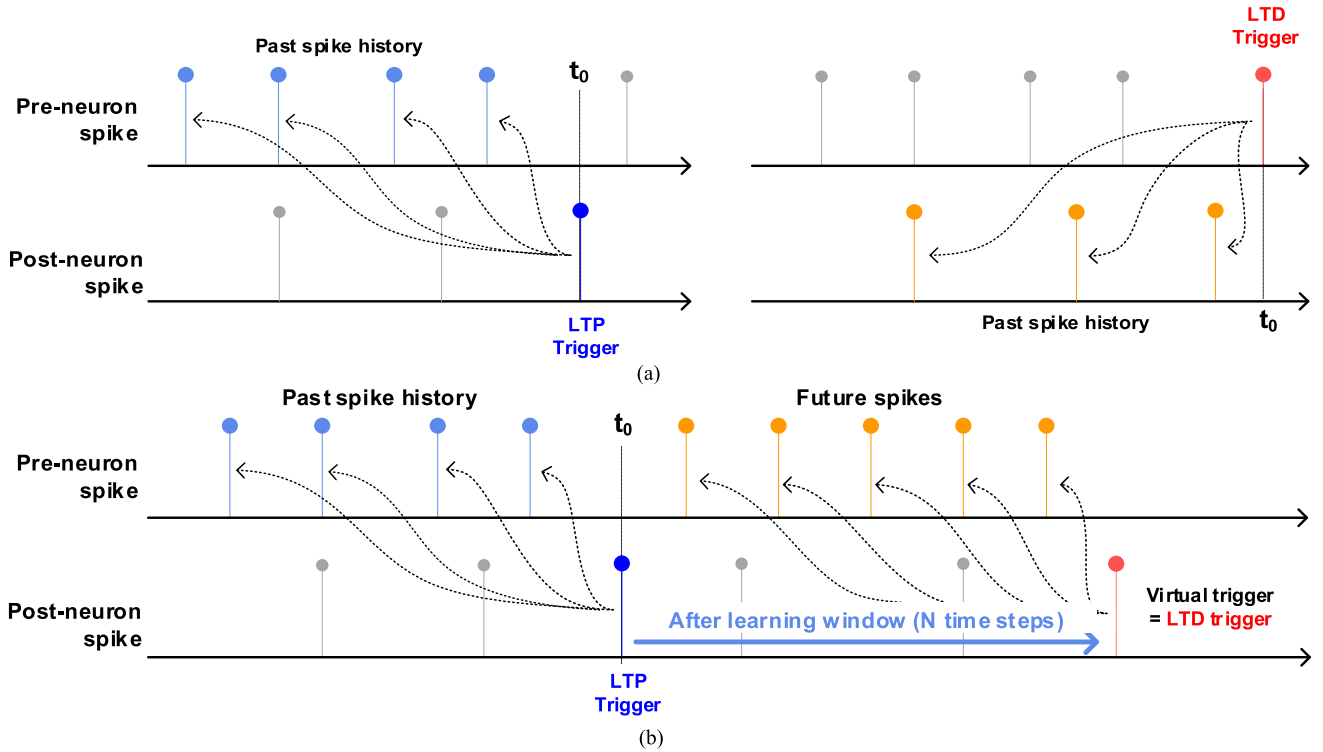


FIGURE 6. Trigger spike and timing correlation of LTP and LTD process. (a) LTP and LTD of conventional STDP. (b) Timing correlation and referencing of spike history of PR-STDP.

successively. Transposable memory [22] cells and array structures cause area overhead owing to the additional transistors and metal lines required. In addition, frequent memory access causes an increase in energy for learning a single image. The second problem is caused by the different firing rates of the pre- and post-neuron spikes. When a pre-neuron spike is used as a trigger spike, it has a higher firing rate than the post-neuron spike, causing more LTD operations [14]. This frequent memory access for LTD operations results in energy overhead. Therefore, it is suitable to use post-neuron spikes as trigger spikes in both LTP and LTD processes to achieve energy-saving hardware implementation.

The main concept of LTD is that the pre-neuron spikes entering after the post-neuron spike are irrelevant to the firing of the learning neuron. Thus, the weights of the synapses where the pre-neuron spike has entered are reduced. Therefore, the LTD process in PR-STDP can be performed by referring to the pre-neuron spike history accumulated for the learning window after the post-neuron spike is fired, as illustrated in Fig. 6. (b). The last history of post-neuron spike becomes a new virtual trigger for the LTD process, which synchronizes the memory access direction to the direction represented in blue because the synaptic weights associated with the neuron fired virtual trigger are aligned in that direction, similar to the LTP process depicted in Fig. 7. In addition, the post-neuron spikes are fired less frequently than the pre-neuron spikes, implying that the virtual spike triggers the LTD process less frequently than the baseline STDP. Therefore, using post-neuron spike in both LTP and LTD resolves

the aforementioned problems of memory access direction and frequent memory access.

In the situation where many neurons fire simultaneously, multiple cycles are required to update all synaptic weights associated with these neurons. However, the maximum number of neurons that fire at the same time step is not fixed. Thus, redundant cycles are required to find all the neurons that fire, which increases control logic complexity. In addition, updating all the neurons that fire increases the number of memory accesses for weight modifications. The neuron that has learned a certain number fires dominantly, when the input is an image of the number that the neuron has learned. Therefore, it is unusual for many neurons to fire at the same time. Thus, arbitrarily selecting a neuron to be updated at the same time step does not interrupt the learning of the dominant neuron. With arbitrary neuron selection, PR-STDP can reduce the total number of memory accesses for updating the synaptic weight by eliminating the redundant learning processes of the neuron that are less related to the input.

The proposed all-to-all PR- and TS-STDP algorithms reduce the area of the STDP circuit and the power consumption while maintaining the performance of the original all-to-all STDP algorithm.

IV. HARDWARE IMPLEMENTATION OF THE PROPOSED STDP

The system architecture of the proposed STDP is presented in Fig. 8. The synaptic weights are stored in SRAM with an array size of 1.75 K by 784. Each row of the SRAM is

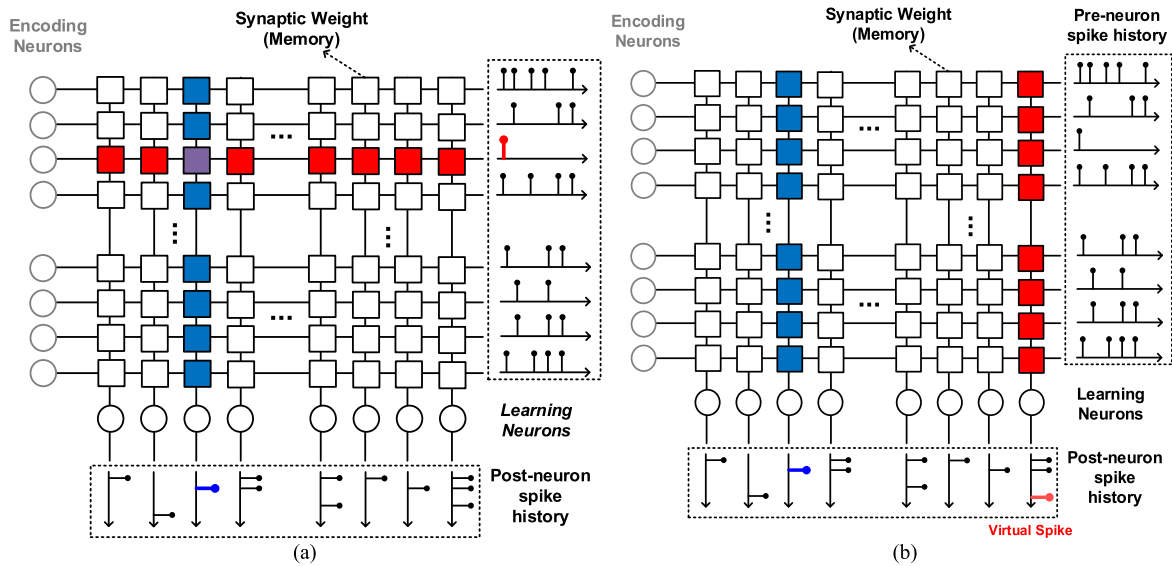


FIGURE 7. Memory access directions associated with LTP (blue cells) or LTD (red cells). (a) Orthogonal memory access direction of conventional STDP. (b) Identical memory access direction of PR-STDP.

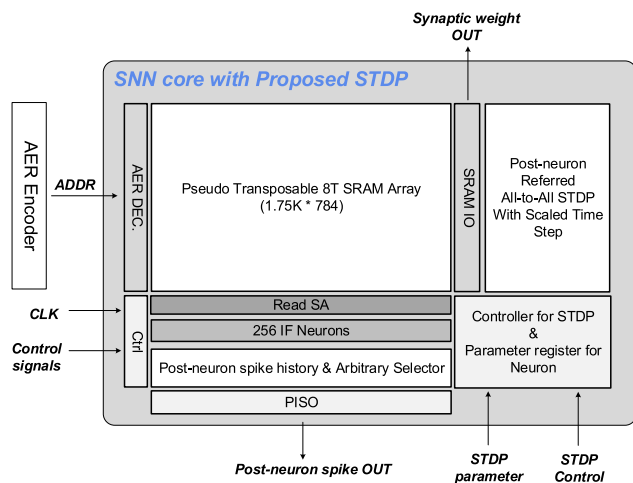


FIGURE 8. Overall chip architecture of SRAM based on-chip SNN system.

connected to the same encoding neuron; thus, 784 rows (each pixel of the 28×28 MNIST image) are required. The number of neurons is 256, and one synaptic weight is represented with 7 bits; hence, 1.75 K columns are required. The pre-neuron spikes are used as the word line (WL) signal of the SRAM array with address event representation [23]. The synaptic weights of the synapses that pre-neuron spikes have entered are accumulated to the neurons in parallel. The integrate-and-fire (IF) neuron model [24] with homeostasis [25] is used. If the accumulated neuron potential exceeds the threshold, a post-neuron spike is generated.

The STDP block is located at the right side of the SRAM array and stores the pre-neuron spike that arrives through the WL. The post-neuron spike history register is located at the bottom of the neurons to correct the timing information of each neuron. The learning signal indicates the start of

the learning mode, and thus, the controller starts the STDP process when the learning signal is enabled. The STDP block includes three functional blocks: a spike history register, Δw calculation block, and synaptic weight update block. An arbitrary selector is also used to select a neuron to be updated.

A. SPIKE HISTORY REGISTER

The spike history in the conventional STDP has been implemented in a counter-based manner [26], [27]. The counter-based approach can implement NN-STDP, but not all-to-all STDP because of its operation property. When the reference spike arrives, the counter for the spike history is reset to “0” and then starts counting. When the trigger spike arrives, the Δw value is calculated based on the current counter value, and then, the counter is reset to “0.” This approach can save the area because the counter can express the learning window of N time steps with $\log_2 N$ bits [28]. However, it is not possible to implement all-to-all STDP because the value of the previous spike history is reset and overwritten with the nearest spike information. Thus, a shift register-based approach for the spike history is required to implement all-to-all STDP.

The spike history register consists of N -bit shift registers, as illustrated in Fig. 9. The first bit of the spike history register operates asynchronously with the other bits. The spike is used as a set signal for the first flip-flop in the spike history register, and thus, the first bit is set to 1 when the spike appears. Then, at the falling edge of the learning mode signal, which indicates the learning time step, the value is shifted to the second register, and the first register is reset to 0. The other bits of the shift register operate synchronously at the falling edge of each learning time step. In the conventional method, an N -bit shift register implemented the spike history.

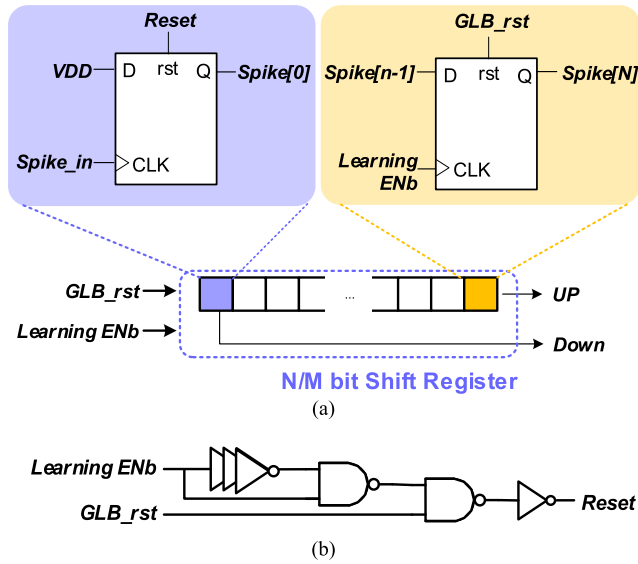


FIGURE 9. Circuit diagram of spike history register. (a) Overall circuit design of spike history register. (b) Pulse circuit for resetting the first bit at the falling edge of learning mode signal.

However, in the proposed TS-STDP, it can be implemented with an N/M bit shift register by scaling M time steps into one learning time step. With the proposed TS-STDP, the area required for the spike history register can be reduced by M times. The area of the spike history register occupies 75% of the entire learning block; hence, the decrease in the area of the spike history register significantly affects the entire area of the STDP learning circuit. When M integration and fire operations are completed, the inference signal is turned off and the learning mode signal is activated. The operation of STDP occurs M times less frequently than the baseline STDP owing to quantization, which can reduce the power consumption required for memory accesses and the number of clock cycles for STDP.

B. Δw CALCULATION BLOCK

The Δw calculation block calculates the Δw value according to the timing information stored in the spike history registers. For a simple calculation, the up/down counter is used. When the learning mode ends, if the first bit of pre-neuron spike history is “1,” the counter increases by “1.” This operation implies that the spike is in the learning window, which increases the Δw value by “1.” The spike is absent from the spike history after the learning window time has elapsed; hence, the current Δw value should be reduced by “1.” This can be carried out by checking the last bit of the spike history. When the last bit of the spike history is “1,” the counter decreases by “1.” Thus, the first and last bit values of the pre-neuron spike history are used as the UP and DOWN signals, respectively. According to these UP and DOWN signals, the counter value increases or decreases, as illustrated in Fig. 10. Δw is updated at the falling edge of the learning signal and is added to the original synaptic weight.

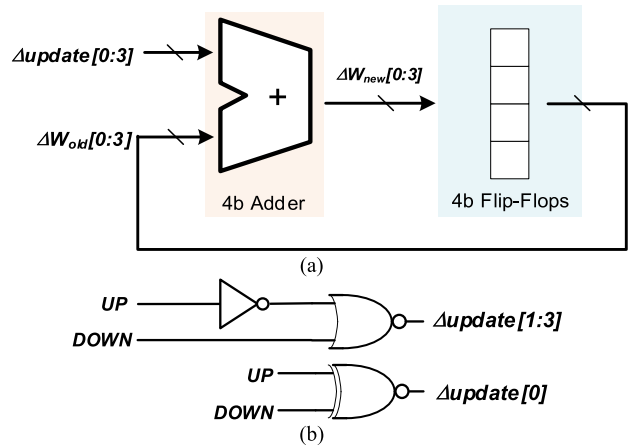


FIGURE 10. (a) Block diagram of Δw calculation block composed with 4 bit adder and 4 bit register. (b) Logic for generating the Δw update value.

C. SYNAPTIC WEIGHT UPDATE BLOCK

The synaptic weight update block operates when the learning mode is activated. This block operates in register form and update form. When the learning mode begins, the original 7-bit weight is read from the SRAM array sequentially. In this mode, the synaptic weight update block (register form) operates as a shift register, and therefore, the output of the sense amplifier is sequentially stored in the synaptic weight update block. Subsequently, according to the LTP and LTD enable signals generated at the arbitrary neuron selector, Δw is added (or subtracted) to (or from) the original weight (update form). Subsequently, the modified synaptic weights are written to the SRAM array again sequentially.

D. ARBITRARY NEURON SELECTOR

The neuron to be updated is randomly selected by an arbitrary neuron selector that includes two stages of domino NOR 16 circuits, as presented in Fig 11. One of the 16 groups of neurons is selected in the first stage of the domino NOR 16 circuit, and in the second stage of the domino NOR 16 circuit, one of the 16 neurons in the selected group at the first stage is selected. The two 4-bit counters generate an address for groups and neurons, respectively. When group detection is performed, all enable signals for all neurons are high; therefore, the output of each group is high if the fired neuron exists. After group detection, the 4-bit counter for group detection holds its value to select the group to be searched. Then, a 4-bit counter for neuron detection searches one of the neurons in the selected group. If the neuron is found, then the 4-bit counter for neuron detection holds its value. The neuron selection process is conducted with two-stage dynamic NOR gates utilizing the first and last bits of the 256-output spike history. The first and last bits of the post-neuron spike history represent whether a post-neuron spike fired at the current learning time step exists and whether the virtual spike for LTD exists, respectively. Therefore, the neuron searching process results for the first and last bits of the

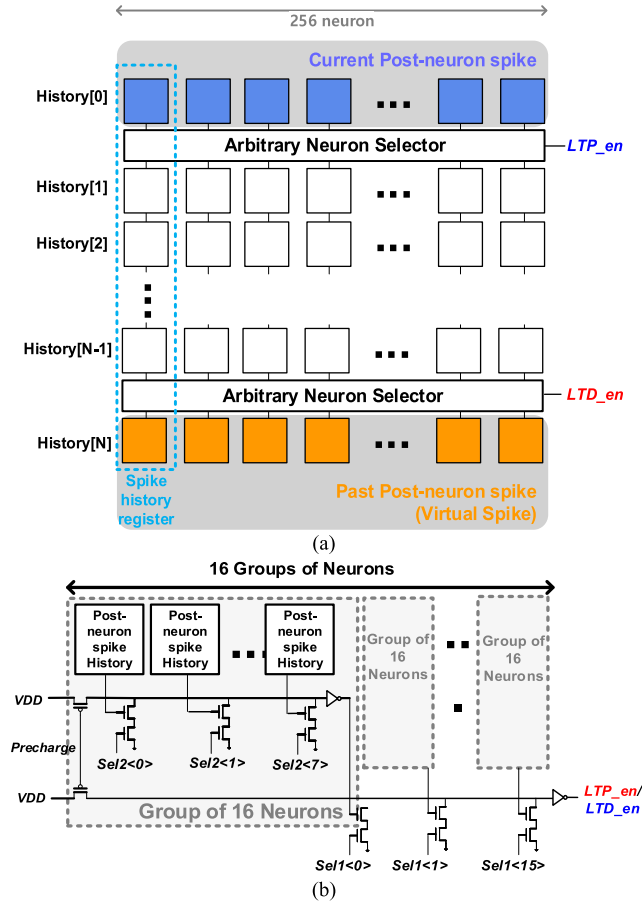


FIGURE 11. (a) Location of arbitrary neuron selector; LTP/LTD enable is detected by OR operation of first/last bit of post-neuron spike histories, respectively. (b) Circuit diagram of arbitrary neuron selector.

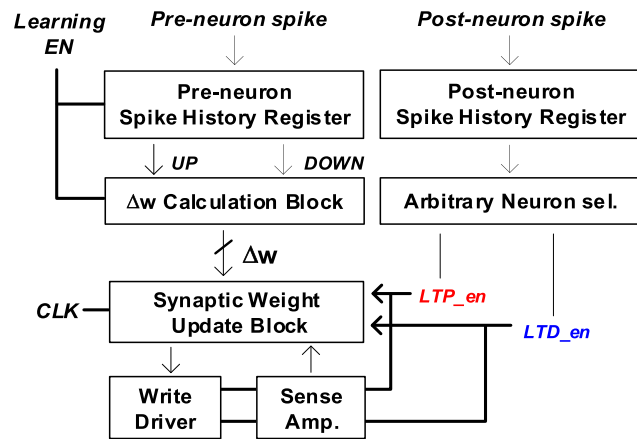


FIGURE 12. Signal flow of three blocks of STDP and memory peripheral circuits.

post-neuron spike history are used as the enable signal for the LTP and LTD processes, respectively, as demonstrated in Fig. 12.

The overall operation of PR-STDP with TS-STDP proceeds in the following order. When the learning mode is enabled, as shown in Fig. 13, in mode_s1, a group of neurons that have a fired neuron is searched for 16 cycles, and then,

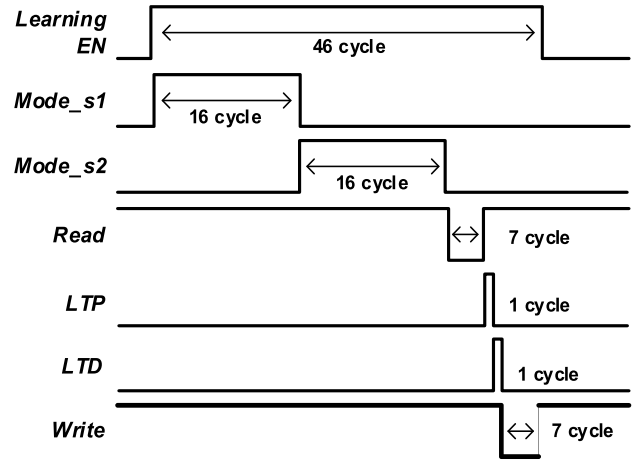


FIGURE 13. Waveform of control signals for the STDP circuit.

in mode_s2, the neurons in the group selected in mode_s1 are searched to find the neuron fired during 16 cycles. If the neuron is detected, the LTP and LTD enable signals are transferred to the STDP block. Then, the original synaptic weights associated with the neuron are read from the SRAM for 7 cycles. Subsequently, Δw is added to the original weight for one cycle if the LTP enable signal is high. After the LTP operation, Δw is subtracted from the original weight at the next cycle if the LTD enable signal is high. The modulated weight is written to the SRAM again for 7 cycles. Then, the learning mode is disabled, and the spike history register is shifted. Subsequently, a new Δw value is calculated according to the spike history.

V. SIMULATION RESULT

The proposed STDP could improve the efficiency of hardware implementation in terms of area and energy. The simulation system was based on the assumption that 784 pixels of the MNIST image were encoded into 784 pre-neuron spike trains using Poisson encoding. The 256 learning neurons produced post-neuron spikes. The performance of the overall system using the proposed STDP was evaluated according to the classification accuracy and reconfigured weight view through a MATLAB simulation. A 28-nm technology was used for circuit design to estimate the area and power.

A. PERFORMANCE

The system performance could be degraded with TS-STDP owing to the timing error. As the scaling number (M) increased, the accuracy of the MNIST classification degraded, as shown in Fig. 14. After 10 epochs of training, the system accuracy was degraded by 2.2% when $M = 6$ and by 2.8% when $M = 7$, compared to that when $M = 3$. To maintain the system performance while taking advantage of scaling, we set $M = 5$. PR-STDP with arbitrary neuron selection also does not affect the overall system performance. Previous research [12] exhibited an accuracy of 85%, the same as that exhibited by PR-STDP. When PR-STDP with arbitrary neuron selection and TS-STDP were applied, the control was

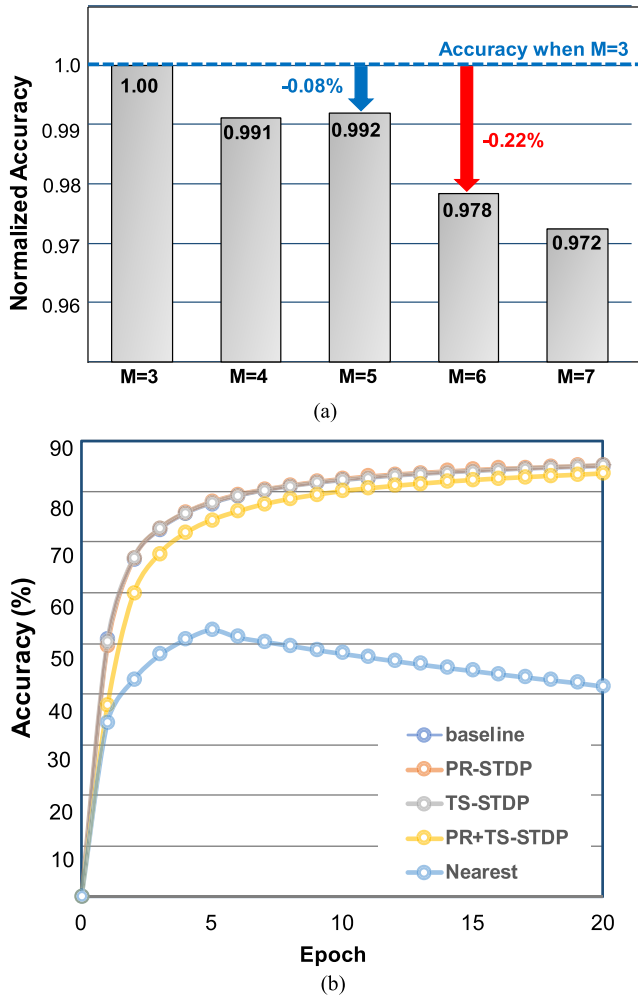


FIGURE 14. (a) Accuracy trend as scaling number M of TS-STDP increases. (b) System performance measured over 20 epochs trained with baseline and proposed STDPs.

significantly simplified, and the number of memory accesses for weight update was decreased by reducing the redundant part of STDP with an insignificant accuracy reduction of 1.5%.

1) AREA

The area of the proposed STDP block was estimated and compared to the original STDP block. With TS-STDP, the storage for the spike history was reduced by M times. Thus, the area of TS-STDP with M = 5 was decreased by 54% compared to that of the original STDP, at an insignificant accuracy degradation of 1%.

PR-STDP did not contribute to the decrease in the area of STDP block as substantially as TS-STDP did. PR-STDP reduced the STDP’s Δw calculation block and synaptic weight update block at the bottom of the SRAM array, compared to the baseline STDP. However, this reduction did not significantly reduce the overall area because the spike history block that occupied most of the area of STDP had to be retained for the virtual trigger. Thus, PR-STDP achieved only an 8% smaller area than the baseline STDP.

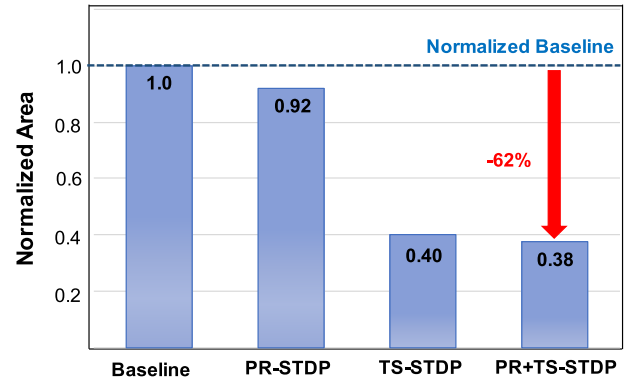


FIGURE 15. Estimated area of STDP. Proposed STDP reduced its area by 62% compared to baseline STDP when M = 5 and both TS+PR-STDP are applied.

TABLE 1. Area comparison including M = 3,4,5 cases.

	Spike History	Other Blocks*	Total	Normalized Area
Baseline	331730.7	157846.9	489577.6	1.00
PR-STDP	331730.7	118992.3	450723.0	0.92
TS-STDP (M=3)	110576.9	157846.9	268423.8	0.55
TS-STDP (M=4)	82932.7	157846.9	240779.5	0.49
TS-STDP (M=5)	66346.	157846.9	224193.0	0.46
PR+TS-STDP (M=5)	66346.1	118992.3	185338.4	0.38

Unit: μm²

* Including Δw update block and synaptic weight update block

The combination of PR- and TS-STDP reduced the total area by 62%, as shown in Fig. 15. This was mainly owing to the decrease in the area of the spike history block by TS-STDP.

In addition, PR-STDP used a pseudo-transposable 8T SRAM cell instead of a transposable 8T SRAM cell. The area of the pseudo-transposable 8T cell is 23% smaller than that of the transposable 8T cell. Therefore, synaptic weight memory that occupied approximately 62% of the total chip area could be reduced by using a pseudo-transposable 8T SRAM cell. This area reduction in both the STDP block and SRAM array helped to implement the hardware in a compact area.

B. ENERGY

A single baseline STDP consumes 0.314 pJ per operation. When the time-step scaling was applied, a single STDP block consumed 0.281 pJ. The energy reduction was approximately 10% and was from the spike history block. Because the energy consumed by memory access was more than 28 times higher than the single STDP operation energy, the reduction in the number of memory accesses was important. Table 2 shows the SRAM read and write energy for single memory access, and Fig. 16. shows the total energy consumption for learning one image including both memory access and STDP operation of the baseline and proposed STDPs. Because the energy consumption of the entire learning operation was dominantly determined by the number of memory

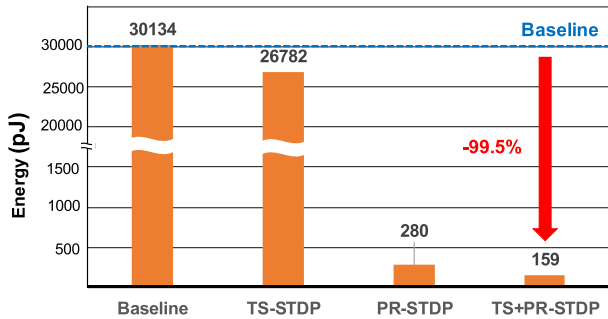
TABLE 2. Energy of single operation.

STDP	Baseline*		0.314	
	TS-STDP		0.281	
SRAM	Transposable 8T	READ	0.666	0.475**
		WRITE	0.769	0.574**
	Pseudo-transposable 8T	READ	0.621	
		WRITE	0.705	

Unit: pJ

* PR-STDP uses identical baseline STDP block

** Orthogonal direction to that of pseudo-transposable 8T

**FIGURE 16. Average energy consumption for learning an image. Average energy consumption of each STDP. TS+PR-STDP shows 99.5% improvement over the baseline STDP.**

accesses, reducing the memory access was important for energy-efficient learning. In this respect, the LTD process of the baseline STDP was triggered by a pre-neuron spike, which induced frequent memory access for LTD, resulting in high energy consumption. However, PR-STDP changed the trigger of the LTD process from pre-neuron spike to virtual spike (past post-neuron spike), which were less frequent, reducing the energy consumption by memory access. In addition, arbitrary neuron selection decreased energy by selecting a neuron to be updated. The MATLAB simulation result, which reflected the energy for memory access and STDP operation, showed an improvement in the average energy consumption for learning one image when the proposed STDP was used. Fig. 16. shows the decrease in the average energy consumption for learning one image. The energy consumption decreased by 11.1% or 99.1% when TS-STDP or PR-STDP was applied, respectively. When both schemes were applied, the energy consumption reduced by 99.5% owing to the reduction of energy from the STDP block.

Therefore, the proposed STDP reduced the area of the learning circuit by 62% and energy consumption by 99.5% with insignificant performance degradation compared to that by the baseline STDP.

VI. CONCLUSION

In this paper, we proposed two hardware-aware STDP algorithms to reduce the area and power of STDP hardware for on-chip learning. TS-STDP reduced the spike history storage by quantizing multiple steps into one learning time step. PR-STDP with arbitrary neuron selection enabled efficient memory access by synchronizing the triggers of LTP and LTD

to a post-neuron spike and eliminated redundant memory accesses. With TS- and PR-STDP, the area of the STDP block was reduced by 62% and power consumption was reduced by 99.5%. Despite the significant reduction in hardware costs, system performance exhibited insignificant accuracy degradation within 1%.

REFERENCES

- [1] F. Sandin, A. I. Khan, A. G. Dyer, A. H. M. Amin, G. Indiveri, E. Chicca, and E. Osipov, "Concept learning in neuromorphic vision systems: What can we learn from insects?" *J. Softw. Eng. Appl.*, vol. 7, no. 5, pp. 387–395, 2014.
- [2] J. Jeon, J. H. Park, and Y.-S. Jeong, "Dynamic analysis for IoT malware detection with convolution neural network model," *IEEE Access*, vol. 8, pp. 96899–96911, 2020.
- [3] N. Liu, L. Li, B. Hao, L. Yang, T. Hu, T. Xue, and S. Wang, "Modeling and simulation of robot inverse dynamics using LSTM-based deep learning algorithm for smart cities and factories," *IEEE Access*, vol. 7, pp. 173989–173998, 2019.
- [4] J. Wan, J. Yang, Z. Wang, and Q. Hua, "Artificial intelligence for cloud-assisted smart factory," *IEEE Access*, vol. 6, pp. 55419–55430, 2018.
- [5] B. Li, Z. Zhao, Y. Guan, N. Ai, X. Dong, and B. Wu, "Task placement across multiple public clouds with deadline constraints for smart factory," *IEEE Access*, vol. 6, pp. 1560–1564, 2018.
- [6] X. Lei, H. Pan, and X. Huang, "A dilated CNN model for image classification," *IEEE Access*, vol. 7, pp. 124087–124095, 2019.
- [7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
- [8] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.
- [9] X. Yang, F. Li, and H. Liu, "A survey of DNN methods for blind image quality assessment," *IEEE Access*, vol. 7, pp. 123788–123806, 2019.
- [10] Z. Chen, J. Hu, X. Chen, J. Hu, X. Zheng, and G. Min, "Computation offloading and task scheduling for DNN-based applications in cloud-edge computing," *IEEE Access*, vol. 8, pp. 115537–115547, 2020.
- [11] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proc. IEEE*, vol. 103, no. 8, pp. 1379–1397, Aug. 2015.
- [12] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.
- [13] R. V. W. Putra and M. Shafique, "FSpINN: An optimization framework for memory- and energy-efficient spiking neural networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3601–3613, Nov. 2020.
- [14] F. Ponulak and A. Kasiński, "Introduction to spiking neural networks: Information processing, learning and applications," *Acta Neurobiol. Experim.*, vol. 71, no. 4, pp. 409–433, 2011.
- [15] 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.
- [16] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, Feb. 2010.
- [17] S. Song, K. D. Miller, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, Sep. 2000.
- [18] 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.
- [19] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: Taming the beast," *Nature Neurosci.*, vol. 3, no. S11, pp. 1178–1183, Nov. 2000.
- [20] P. D. Roberts and C. C. Bell, "Spike timing dependent synaptic plasticity in biological systems," *Biol. Cybern.*, vol. 87, nos. 5–6, pp. 392–403, Dec. 2002.
- [21] A. Cassidy, A. G. Andreou, and J. Georgiou, "A combinational digital logic approach to STDP," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 673–676.
- [22] H. Cho, H. Son, K. Seong, B. Kim, H.-J. Park, and J.-Y. Sim, "An on-chip learning neuromorphic autoencoder with current-mode transposable memory read and virtual lookup table," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 161–170, Feb. 2018.

- [23] Z. Kang, L. Wang, S. Guo, R. Gong, Y. Deng, and Q. Dou, "ASIE: An asynchronous SNN inference engine for AER events processing," in *Proc. 25th IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, May 2019, pp. 48–57.
- [24] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
- [25] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.
- [26] G. K. Chen, R. Kumar, H. E. Sumbul, P. C. Knag, and R. K. Krishnamurthy, "A 4096-neuron 1M-synapse 3.8-pJ/SOP spiking neural network with on-chip STDP learning and sparse weights in 10-nm FinFET CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 992–1002, Apr. 2019.
- [27] J. Kim, J. Koo, T. Kim, and J.-J. Kim, "Efficient synapse memory structure for reconfigurable digital neuromorphic hardware," *Frontiers Neurosci.*, vol. 12, p. 829, Nov. 2018.
- [28] J. Sim, S. Joo, and S.-O. Jung, "Comparative analysis of digital STDP learning circuits designed using counter and shift register," in *Proc. 34th Int. Tech. Conf. Circuits/Syst., Comput. Commun. (ITC-CSCC)*, Jun. 2019, pp. 1–4.



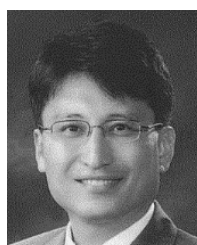
SARA CHOI (Graduate Student Member, IEEE) was born in Gwangju, South Korea, in 1990. She received the B.S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2013 and 2020, respectively.

She joined Samsung Electronics Company Ltd., Hwaseong-si, Gyeonggi-do, South Korea, in 2020. Her current research interests include peripheral circuit and memory architecture design for STT-RAM.



HYO JUNG JANG (Graduate Student Member, IEEE) was born in Seoul, South Korea, in 1994. She received the B.S. and M.S. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2018 and 2020, respectively.

She joined Samsung Electronics Company Ltd., Hwaseong-si, Gyeonggi-do, South Korea, in 2020. Her research interest is focused on SNN neuromorphic chips.

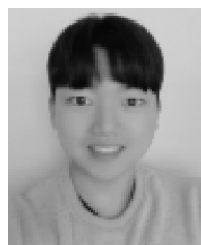


SEONG-OOK JUNG (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2002.

He was with Samsung Electronics Company, Ltd., Hwaseong, South Korea, from 1989 to 1998, where he was involved in specialty memories, such as video RAM, graphic RAM, window RAM, and merged memory logic. From 2001 to 2003, he was with T-RAM Inc., Mountain View, CA, USA, where he was the Leader of the Thyristor-Based Memory Circuit Design Team. From 2003 to 2006, he was with Qualcomm Inc., San Diego, CA, USA, where he was involved in high-performance low-power embedded memories, process variation tolerant circuit design, and low-power circuit techniques. Since 2006, he has been a Professor at Yonsei University. His current research interests include process variation tolerant circuit design, low-power circuit design, mixed-mode circuit design, and future generation memory and technology.

Dr. Jung is a Board Member of the IEEE SSCS Seoul Chapter.

• • •



GISEOK KIM (Graduate Student Member, IEEE) was born in Gunpo, South Korea, in 1996. She received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2020, where she is currently pursuing the Ph.D. degree in electrical and electronic engineering.

Her current research interests include SNN neuromorphic chip and compute-in-memory.



KIRYONG KIM was born in Yeongdong, South Korea, in 1989. He received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2014, where he is currently pursuing a Ph.D. degree in electrical and electronic engineering.

His current research interests include phase-locked loops, delay-locked loop designs, and energy harvesting interface designs.