Ph.D. DISSERTATION

# Hardware-based Neural Networks using gated Schottky diodes

## 게이티드 쇼트키 다이오드를 이용한 하드웨어 기반 뉴럴 네트워크

by

SUHWAN LIM

August 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

# Hardware-based Neural Networks using gated Schottky diodes

# 게이티드 쇼트키 다이오드를 이용한 하드웨어 기반 뉴럴 네트워크

지도교수 이 종 호

이 논문을 공학박사 학위논문으로 제출함

2019 년 8 월


서울대학교 대학원

전기정보공학부

**임 수 환**


임수환의 공학박사 학위논문을 인준함

2019 년 8 월

위 원 장 : 박 병 국 (인)

부위원장 : 이 종 호 (인)

위 원 : 유 승 주 (인)

위 원 : 심 재 윤 (인)

위 원 : 김 상 범 (인)

# Hardware-based Neural Networks using gated Schottky diodes

by

Suhwan Lim

Advisor: Jong-Ho Lee

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

in Seoul National University

August 2019

Doctoral Committee:

Professor Byung-Gook Park, Chair

Professor Jong-Ho Lee, Vice-Chair

Professor Sungjoo Yoo

Professor Jae-Yoon Sim

Assistant Professor Sangbum Kim

# ABSTRACT

Artificial intelligence (AI) based on neural network technology has been widely studied in various industrial fields because it can outperform human cognitive ability. However, since the von Neumann architecture, which is a conventional computing architecture, has a bottleneck problem between memory and computing unit and a very large power consumption, the necessity of neuromorphic computing is emerging. Especially, electronic devices operating as synapses can reduce a large portion of power consumption because they can perform low-power and high-speed vector-by-matrix multiplication (VMM). In this paper, we introduce the gated Schottky diode (GSD) as a synapse device. It operates as a reverse Schottky diode, so the synaptic current is low and saturates to the input voltage. In addition, the conductance response with respect to the number of pulses applied to the synapse device is linear. By considering these characteristics, we design neuron circuits that perform current summation, pulse-width modulation, and activation function. Through SPICE simulation, we evaluate the inference

accuracy of a 2-layer neural network. The classification accuracy rate of 100 images of MNIST test sets is 94%, and it is comparable to the reference accuracy obtained with software. Furthermore, we investigate the on-chip learning rule of the neural network based on the electronic synapse device. Because of the non-idealities of synapse devices, we propose a weight-updating method based on the Manhattan update rule and evaluate the learning accuracy with respect to the various non-idealities of synapse devices. When the synapse device has linear conductance response and high dynamic range, the learning accuracy can be comparable to the reference accuracy obtained with software. While off-chip training scheme is vulnerable to device variations, the on-chip training rule can mitigate this variation effect. To implement a neural network system, we fabricate a synapse device array based on GSDs. With the help of the saturation characteristics, the VMM computation is well performed without IR drop problem in metal wires. The fabricated synapse device array shows a variation ($\sigma/\mu$) of 0.34, 0.22, and 0.14 for three different synaptic weight states. The gated Schottky diode operating as synapse device, its compatible circuits, and on-chip learning rule that we have

proposed can help to implement a hardware-based neural network.

Keywords: Artificial intelligence, synapse device, gated Schottky diode, current saturation, linear conductance response, neuron circuits, on-chip learning rule.

Student number: 2015-30202

# CONTENTS

# List of Figures

# List of Tables

# Chapter 1
# Introduction

## 1.1 Background

Conventional computing systems which are based on CMOS logic and the von Neumann architecture are powerful for well-defined mathematical problems. However, problems in the real world cannot be efficiently solved using these computing systems. Therefore, biologically inspired neuromorphic systems have emerged as an attractive field of research [1]. Recently, several types of emerging electronic synapse devices such as phase change memory (PCRAM) [2]-[4], resistive change memory (RRAM) [5]-[8], ferroelectric devices [9], and FET-based devices [10]-[12] have been proposed to mimic biological synapses. Although most of these works have focused on spike-timing-dependent-plasticity (STDP, [13]) learning algorithm [14], the learning performance using STDP is still in its early stage [15], [16]. Unlike the approach in which STDP is used, electronic synapse devices ([15], [17], [18]) can also be applied to deep neural networks (DNNs) with

well-studied back-propagation (BP) algorithms [19]. Many pioneering studies that implement neural networks using electronic synapse devices have focused on fundamental computations, which are vector-by-matrix multiplications. The vector-by-matrix multiplication of forward and backward propagations accounts for a large portion of the computational tasks of a SW-DNN and thus is the main cause of enormous amounts of power consumed. However, when the input signal and the weight are replaced by the input voltage and the conductance of the electronic synapse device, respectively, the output signal is simply expressed as the current flowing from the electronic synapse device array. Therefore, the use of an electronic synapse device can greatly reduce power consumption and improve the speed [20]. In this approach, all computations including propagation should be implemented by hardware (HW) using electronic circuits compatible with the electronic synapse device array. If the BP algorithm is calculated by a SW operation, the communication between the SW and the electronic synapse device array can be the dominant bottleneck, greatly reducing the advantages of HW-DNNs. In addition, process, voltage, and temperature (PVT) variation can degrade the performance of

neural networks because they cannot be taken into account in training process [21]. That is, all computations of SW-based deep neural networks, including forward propagation, backward propagation, and weight updating, should be replaced by calculations using electronic synapse device arrays and electronic circuits. The implementation of full HW-DNNs is challenging owing to the complex computations of activation functions and their derivatives [22]. When the HW includes electronic synapse devices, HW implementation is much more complicated because the weights should be represented using the conductance of a non-ideal synapse device. Therefore, appropriate electronic synapse devices and electronic circuits compatible with the electronic synapse devices are needed. Moreover, an adaptive learning rule which especially enables hardware implementation using electronic synapse devices and electronic circuits is necessary.

The electronic synapse devices for HW-DNNs should have the following characteristics: an extremely operating low power, high scalability (density), high repeatability and reliability, the ability to combine storage and computations [14], and linear and symmetric conductance responses [15]. In large neural networks

based on electronic synapse devices, large synapse array can increase total occupied area and power consumption. Therefore, each electronic synapse device should be scalable and operated in a very low current regime. Furthermore, the electronic synapse device should offer good repeatability, reliability, and uniformity between devices. Otherwise, neural networks (NNs) will diverge because the weights cannot be accurately updated with reference to the target value. If electronic synapse devices have nonlinear and asymmetric conductance response, NNs will no longer learn or diverge because the learning rate is too small or too large depending on the conductance states. In other words, the linear and symmetric conductance response of the electronic synapse device ensures a constant learning rate and convergence of NNs [15]. Note that a nonlinear conductance response can also degrade the STDP learning algorithm [23]. There have been attempts to obtain linear conductance responses [24]-[27], and such attempts should not burden the external circuit. Thus, basically, electronic synapse devices should have linear, symmetric, and repeatable conductance responses for given identical pulses.

Electronic circuits should control the electronic synapse device array and

efficiently provide an activation function [28], [29]. To increase the integration density of synapses, emerging electronic synapse devices should be adopted. However, electronic synapse devices mentioned above have a nonlinear and finite conductance response, and have a variation in conductance between the devices fabricated on the same substrate. Thus, electronic circuits are required to control the electronic synapse array appropriately depending on the device characteristics. These control circuits are able to apply the required voltage to the electronic synapse device to update the weight to the target value. The activation function must be designed according to the device characteristics. These electronic circuits should be designed efficiently so as not to offset the benefits of the electronic synapse devices.

Above all, learning rules based on a BP algorithm must be applicable to the HW used, including the electronic synapse device array and the electronic circuits. In HW-DNNs which use electronic synapse devices, unlike SW-DNNs, the weights are represented by their conductance values, meaning that the weights have discrete and limited values. These characteristics of the weights can degrade the learning

accuracy because it becomes impossible to update the weights precisely to the target values. Therefore, we need an appropriate learning rule to minimize the degradation of the learning accuracy while maintaining the advantages of the electronic synapse device array.

In this paper, we investigate the necessary elements for implementing HW-DNNs, such as synapse device, electronic circuits compatible with synapse device, and adaptive on-chip learning rules.

## 1.2 Software and hardware-based deep neural networks

The software-based deep neural networks (SW-DNNs) with well-studied BP algorithm have shown excellent performance. As shown in Fig. 1, a SW-DNN consists of the input layer, the hidden layers, and the output layer. Each node of the layer is known as a neuron, and a node connection between adjacent layers is called a synapse. The strength of a synapse is the synaptic weight, or simply the weight. By using the BP algorithm, we can find the optimal values of the weights to minimize the training and generalization errors. In the process of finding the

optimal weights, the vector-by-matrix multiplication (VMM) of forward and backward propagations accounts for a large portion of the computational tasks. However, HW-DNNs can perform this VMM with very low power and high speed because the result of the VMM is simply the current of the electronic synapse device array, which is the product of the input voltage and conductance. There are two main approaches for HW-DNNs: on-chip training and off-chip training. We define on-chip training that the weights are updated within the synapse device array for each iteration. For on-chip training, the hardware, including the synapse device array, should perform forward and backward propagation and weight updates. On the other hand, off-chip training means the weight updates are performed by software, and then the calculated weights are transferred to the synapse device array. In this case, the synaptic array is only used for the VMM for forward propagation after training, which is also called inference or dot-product engine. Table 1.1 shows off-chip and on-chip training rule of a HW-DNN compared to a SW-DNN. In a HW-DNN, two identical electronic devices are required to represent a unit synapse, because the weights ($W_{ij}$ for the weight of the synapse between the $i^{th}$ neuron in the

$l-1$ layer and the $j^{th}$ neuron in the $l$ layer) of the unit synapse in NNs should have both positive and negative values. The input signal ($a_i^{l-1}$) for the $i^{th}$ neuron in the $l-1$ layer) and the weight ($W_{ij}$) can be represented by the applied voltage ($V_i^{l-1}$) and the conductance difference of the unit synapse ($G_{ij}^{+}-G_{ij}^{-}$), respectively. The positive and negative values of the weights can be expressed by subtracting the output current from a pair of synapses ($W_{ij} = G_{ij}^{+}-G_{ij}^{-}$). By connecting all unit synapses in the $l-1$ layer connected to the $j^{th}$ neuron in the $l$ layer, the currents from each unit synapse are summed ($\sum_i^N \left(G_{ij}^{+} - G_{ij}^{-}\right) V_i^{(l-1)}$). This weighted sum value ($s_j^l$) is then converted to the input signal of the next layer ($a_i^l$) using an activation function ($f$), which is implemented by electronic circuits. For off-chip training, synapse device array is responsible for this forward propagation. Furthermore, for on-chip training, backward propagation and weight updates should be implemented using synapse device array. In addition to the forward propagation, to compute the backward propagation, the backward-weighted sum should be performed after the weight matrices are transposed. In other words, the postsynaptic neurons during forward propagation should act as the function of the presynaptic neurons in the backward

propagation, and vice versa. That is, the synapse device array should be transposable to implement backward propagation. The input signal in the backward direction ($\delta_j^l$ for the $i^{th}$ neuron in the $l$ layer) and the weight ($W_{ij}$) are represented by applied voltage ($V_j^l$) and the conductance difference in the unit synapse ($G_{ij}^+ - G_{ij}^-$), respectively. In this way, the backward-weighted sum ($\sum_j^M W_{ij}\,\delta_j^l$) can be performed by connecting all unit synapses in the $l$ layer connected to the $i^{th}$ neuron in the $l-1$ layer. Then, we can obtain the error delta value of the $i^{th}$ neuron in the $l-1$ layer ($\delta_j^{l-1}$) by multiplying the derivative value of the activation function $f'(s_i^{l-1})$ by the backward-weighted sum value. After the error delta values of all layers excluding input layer are obtained through these process, the weights can be updated according to the product of the learning rate ($\eta$), the error delta value of the postsynaptic neuron ($\delta_j^l$), and the activated value of the presynaptic neuron ($f(s_i^{l-1})$). In this paper, we focus on on-chip training scheme because it is immune to device variation and is advantageous for power consumption. Therefore, we will discuss how to efficiently update the weights based on hardware synapse array, in a later section.

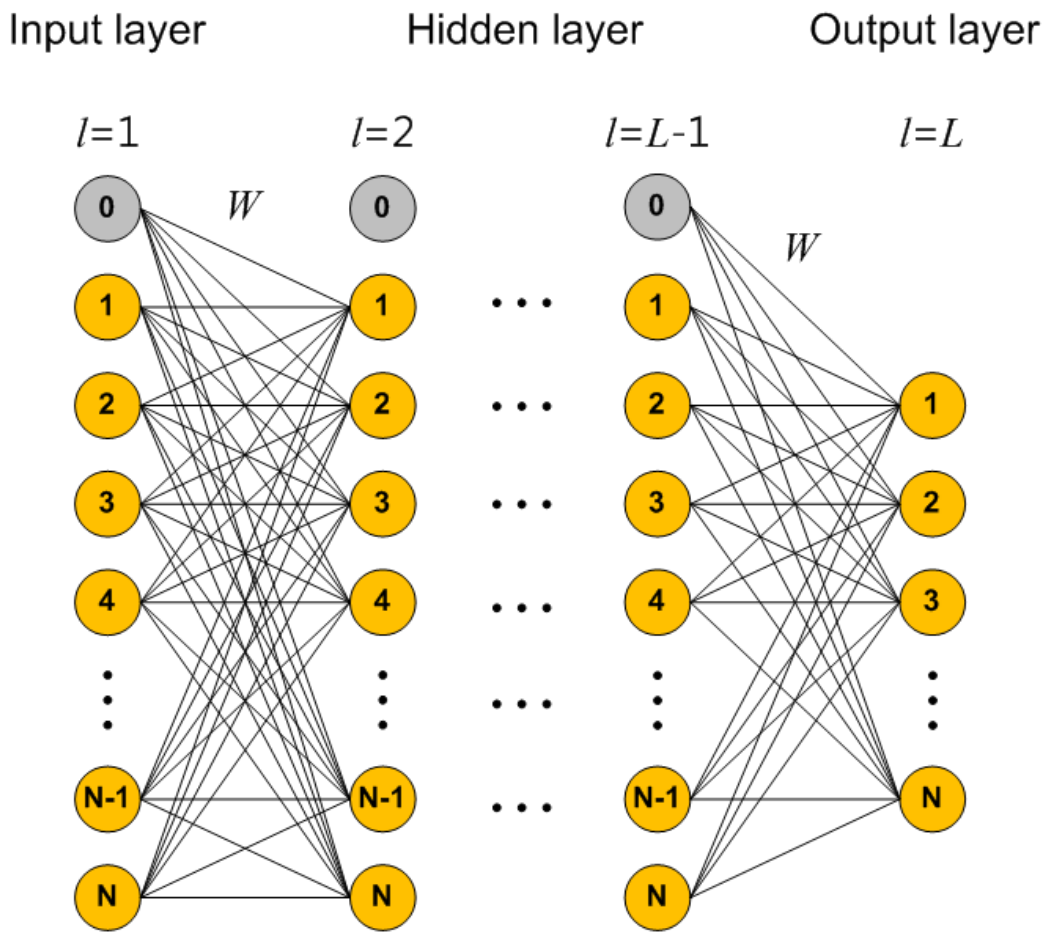



Fig. 1.1. Typical structure of the deep neural networks composed of input layer, hidden layers, and output layer.

Table 1.1. Off-chip and on-chip training rule of hardware-based neural networks.

| Target | Software-based | Hardware-based | |
|---|---|---|---|
| | | Off-chip | On-chip |
| **Weights** $W_{ij}$ | $W_{ij}$ | $G_{ij}^{+} - G_{ij}^{-}$ | $G_{ij}^{+} - G_{ij}^{-}$ |
| **Forward propagation** $S_j^{(l)}$ | $\sum_{i}^{N} W_{ij} a_i^{(l-1)}$ | $\sum_{i}^{N} (G_{ij}^{+} - G_{ij}^{-}) V_i^{(l-1)}$ | $\sum_{i}^{N} (G_{ij}^{+} - G_{ij}^{-}) V_i^{(l-1)}$ |
| **Activated value** $a_j^{(l)}$ | $f\left(s_j^{(l)}\right)$ | $f\left(s_j^{(l)}\right)$ | $f\left(s_j^{(l)}\right)$ |
| **Backward propagation** $\delta_i^{(l-1)}$ | $\sum_{j}^{M} W_{ij} \delta_j^{(l)} \cdot f'\left(s_i^{(l-1)}\right)$ | | $\sum_{j}^{M} (G_{ij}^{+} - G_{ij}^{-}) V_j^{(l)} \cdot f'\left(s_i^{(l-1)}\right)$ |
| **Weight updates** $\Delta W_{ij}$ | $-\eta \cdot \delta_j^{(l)} \cdot f\left(s_i^{(l-1)}\right)$ | | $-\eta \cdot \delta_j^{(l)} \cdot f\left(s_i^{(l-1)}\right)$ |

# Chapter 2
# Synapse device

## 2.1 Device structure and fabrication

When implementing a neural network using electronic synapse devices, the characteristics of synapse device should be considered. In the earlier work, we reported the gated Schottky diode (GSD) as an electronic synapse device [27]. The reverse current of the GSD, which operates as a synaptic current, is modulated by the stored charges in the charge storage layer by applying program or erase pulses. In this section, we devise a bias scheme to eliminate the forward current in this GSD. When performing a VMM using a synapse device array, the current of the GSDs should flow in the reverse direction only. Otherwise, the synaptic current summed by the Kirchhoff's current law (KCL) becomes unintended current, which results in malfunction of the neural networks [30]. When the GSDs in the synapse array and the circuit for the current sum, subtraction, and activation functions are connected to form a system, some of the GSDs in the array can be biased forward. At this time,

the forward-biased GSDs should not flow current in the forward direction to enable proper VMM. We disscuss the device structure and operation scheme to prevent the forward current of Schottky diode. Fig. 2.1 shows a device structure and Fig. 2.2 shows its SEM/TEM images. Whereas the occupied area for single synapse device is $6F^2$ in previous work, the modified synapse device in this paper occupies $12F^2$ to represent single synapse device. Otherwise, the device structure is the same. Key fabrication steps are shown in Fig. 2.3. A layer of $n^+$-doped poly-Si was formed on a $SiO_2$ grown on 6-inch Si substrate and followed by the deposition of $Si_3N_4$ layer (Fig. 2.3 (a)). Both $Si_3N_4$ and poly-Si layers were patterned. A 25.6 nm-thick $SiO_2$ was grown along the sidewall of the patterned poly-Si to isolate bottom-gates (b). A layer of $n^+$-doped poly-Si was deposited again (c), and CMP was performed, followed by removing $Si_3N_4$ layer and CMP for trimming (d). An oxide/nitride/oxide (O/N/O, 12/6/6 nm) gate insulating stack for charge storage was formed, and a 20 nm-thick poly-Si active layer were deposited (e). Then the poly-Si was patterned and SiO2 was grown, which reduces the thickness of the poly-Si to ~16 nm. The contact holes for bottom-gates, anode, and cathode were formed,

and Al electrodes were formed by thermal evaporation and lift-off process (f). Fig. 2.3 (f) is schematic cross-sectional view cut along a dash-dotted line in Fig. 2.2 (a). Although an aluminum electrode is formed on the silicon active layer above the $BG_C$, this electrode is not used for synapse device operation, so we omitted this electrode in Fig. 2.1.

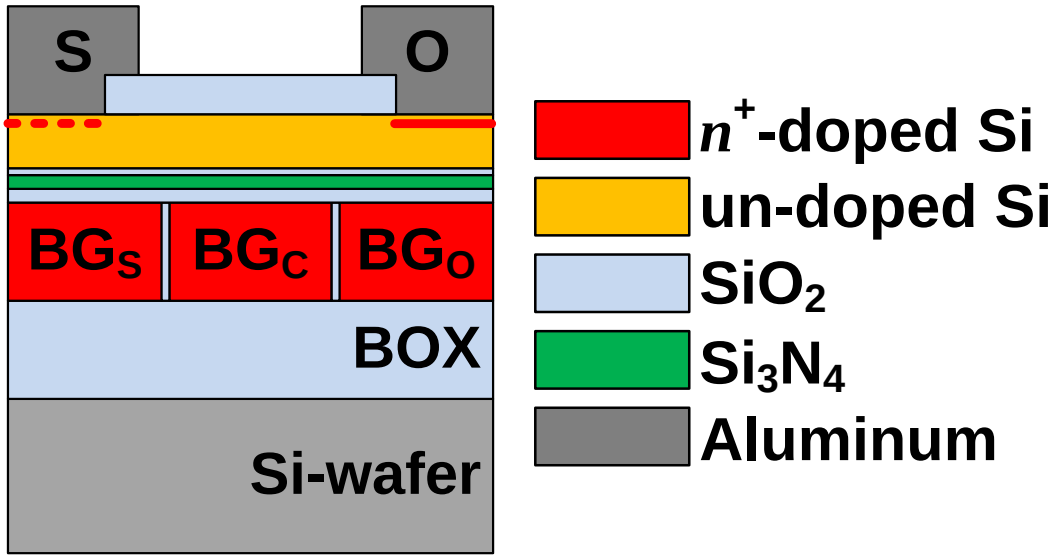



Fig. 2.1. Device structure.

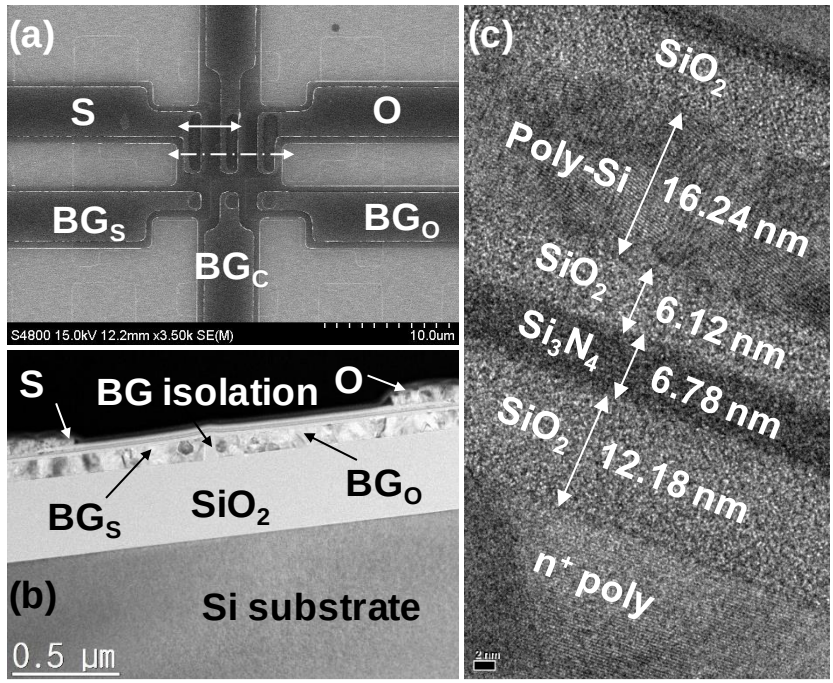



Fig. 2.2. (a) Top SEM view. (b) Cross-sectional TEM images cut along the solid line in (a) and (c) its magnified views.

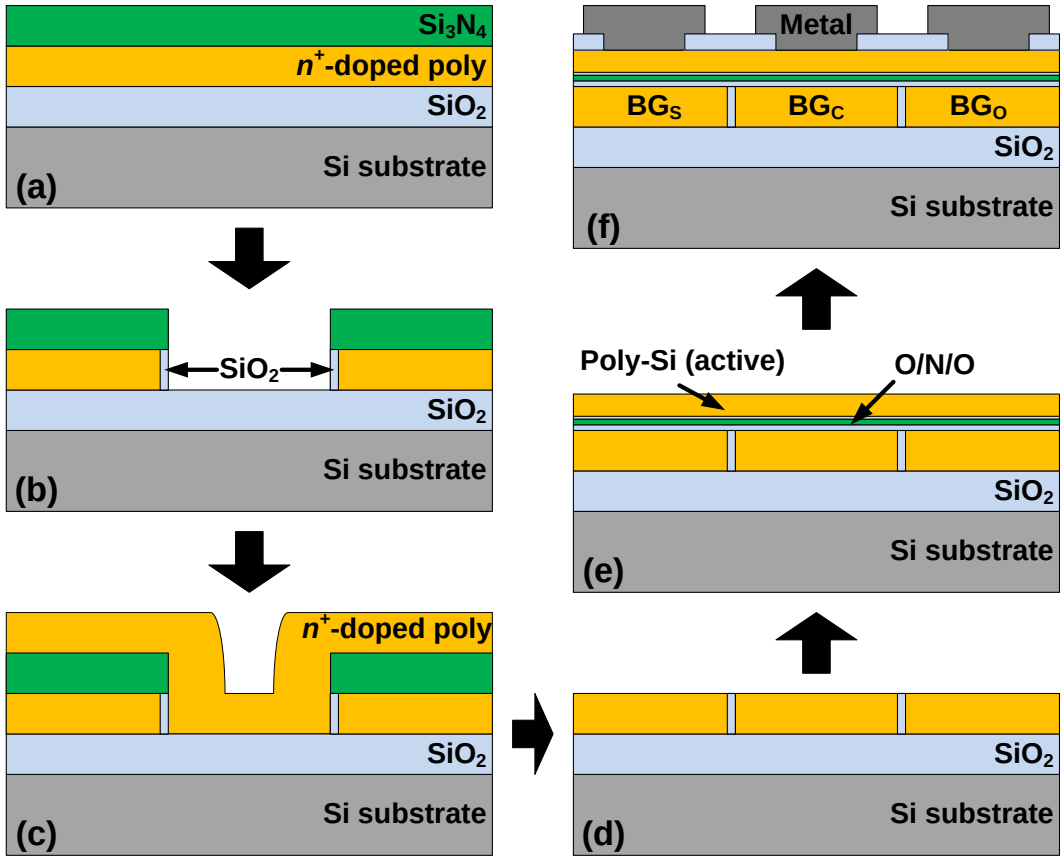



Fig. 2.3. Schematic cross-sectional views of key fabrication process steps of reconfigurable device.

## 2.2 Device operation as a synapse

S and O represent the electrodes formed of aluminum for the Schottky junction and ohmic-like junction, respectively. Input voltage is applied to the O node and the current flows through the un-doped silicon channel between the S and O nodes. Bottom gates ($BG_S$ and $BG_O$) under S and O are used for modulating the Schottky barrier height between the electrode and the un-doped Si channel. Depending on the modulated Schottky barrier height, the junction between the electrode and the Si channel can be a Schottky junction (S) or an ohmic-like junction (O). The bottom gate between $BG_S$ and $BG_O$ (expressed as $BG_C$ in Fig. 2.1) is used for preventing the forward current of the Schottky diode. By applying bias or pulses to these bottom gates, the electrons or holes are induced in the Si channel. The bias condition for operating synapse device is shown in Fig. 2.4. In case of *n*-type GSD, the Si channel above the $BG_O$ becomes an electrically $p^+$-doped region by applying −4 V to the $BG_O$ and the Si channel above the $BG_C$ is electrically $n^+$-doped region by applying 4 V to the $BG_C$. Then, the PN junction is formed internally between the Si channel above the $BG_C$ and the Si channel above the $BG_O$. Therefore, when a

negative bias (−2 V) is applied to the O, there is no current flow due to a reverse biased PN junction, which means that the forward current of the GSD is effectively blocked. On the other hand, when a positive bias (2 V) is applied to the O, the internally formed PN junction is forward biased, so that the reverse current of the Schottky diode flows. This reverse current of the Schottky diode can be modulated by the charge stored in the $Si_3N_4$ layer by applying program or erase pulses to the $BG_S$ node. In this way, the operation of the *p*-type GSD can also be explained. The Si channel above the $BG_O$ becomes an electrically $n^+$-doped region by applying 4 V to the $BG_O$ and the Si channel above the $BG_C$ is electrically $p^+$-doped region by applying −4 V to the $BG_C$. Then, the PN junction is formed internally between the Si channel above the BGC and the Si channel above the $BG_O$. Therefore, when a positive bias (2 V) is applied to the O, there is no current flow due to a reverse biased PN junction, which means that the forward current of the GSD is effectively blocked. On the other hand, when a positive bias (−2 V) is applied to the O, the internally formed PN junction is forward biased, so that the reverse current of the Schottky diode flows. The device symbols for *n*- and *p*-type GSD consisting of

Schottky diode and PN junction are shown in Fig. 2.4. Fig. 2.5 shows the simulated band diagrams cut along the red line A-B-C-D when *n*-/*p*-type GSDs are in operating and cutoff modes. The PN junction is formed near point C. Because of this PN junction, the forward current of *n*- and *p*-type GSD cannot flow in cutoff mode. Fig. 2.6 shows the diode current with respect to the number of applied $e^-$ programming pulses, in case of the *p*-type GSD. As shown in Fig. 2.6 (a), when −4 V is applied to the $BG_O$, the forward current of the Schottky diode flows at a positive $V_O$. However, the forward current of the Schottky diode is blocked when 4 V is applied to the $BG_O$ (Fig. 2.6 (b)). The reverse current of the Schottky diode is modulated by applying program pulses (8.5 V for 10 μs) to the $BG_S$. The effect of pulse width is described in Fig. 2.7. In addition, the reverse current of the Schottky diode starts to saturate when the $V_O$ increases negatively to −1.5 V. The saturated reverse current of the Schottky diode increases linearly with increasing number of applied program pulses as shown in the inset. The physical mechanisms of linear conductance response and saturation characteristics were described in previous works [27], [31]. Firstly, the amount of stored charge logarithmically increases with

the number of pulses for the storage of electrons and holes [32]. The previously stored charges will prevent from additional storing by charge repulsion, so the slope is decreased by the iterative pulses. However, the reverse current of Schottky diode exponentially increases with the channel charge induced by the stored holes. Therefore, the reverse current of Schottky diode increases linearly with the number of applied pulses [27]. In addition, the reverse current of Schottky diode is saturated because of the pinch-off of the parasitic MOSFET [31]. The saturation behavior and linear conductance response are taken into account when designing forward propagation. These characteristics have several advantages and are discussed later.

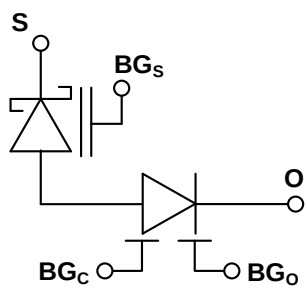
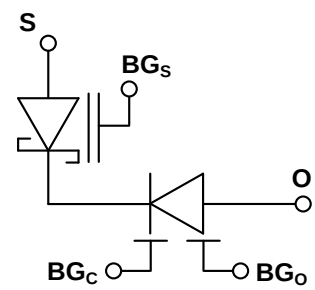
| | ***n*-type** | ***p*-type** |
|---|---|---|
| $V_{BGS}$ | + (variable) | − (variable) |
| $V_{BGC}$ | 4 V | –4 V |
| $V_{BGO}$ | –4 V | 4 V |
| $V_O$ **(Input)** | 2 V (operating)<br>–2 V (cutoff) | –2 V (operating)<br>2 V (cutoff) |
| $V_S$ **(Output)** | 0 V | 0 V |
| **Symbol** |  |  |

Fig. 2.4. Bias scheme to prevent forward current of Schottky diode.

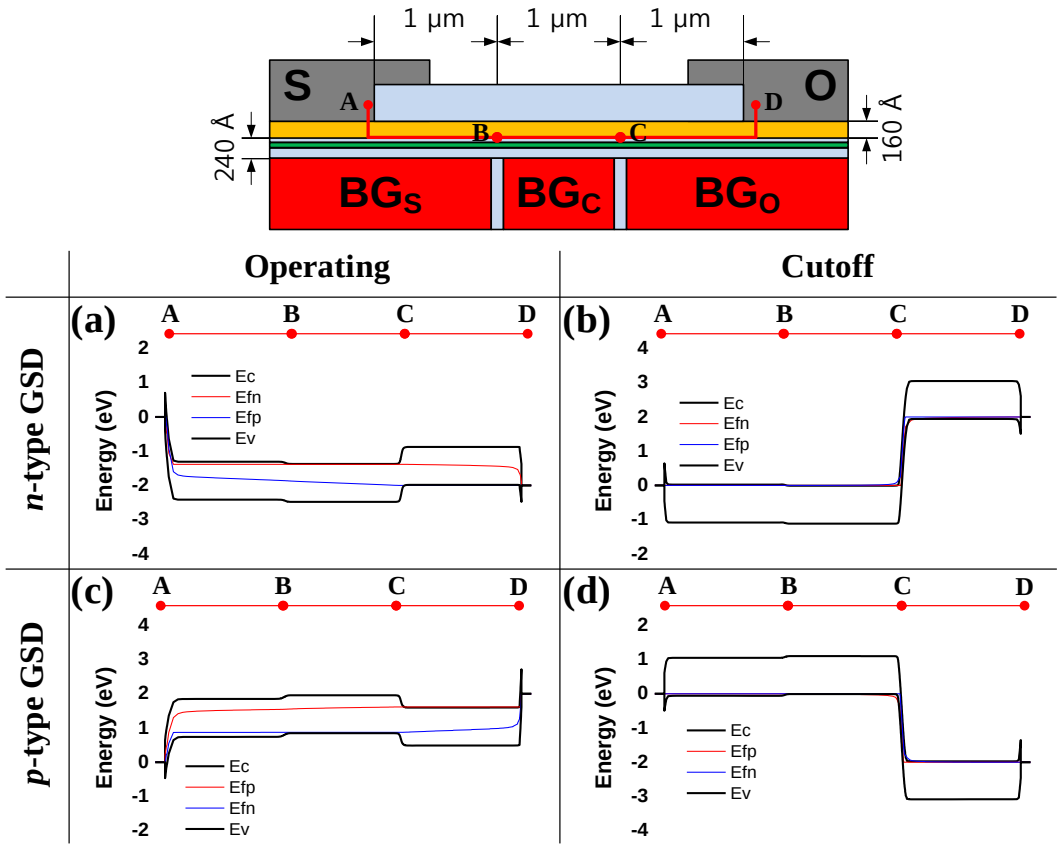



Fig. 2.5. Simulated band diagrams cut along the red line A-B-C-D when *n*-type GSD is (a) operating and (b) cutoff, and when *p*-type GSD is (c) operating and (d) cutoff.

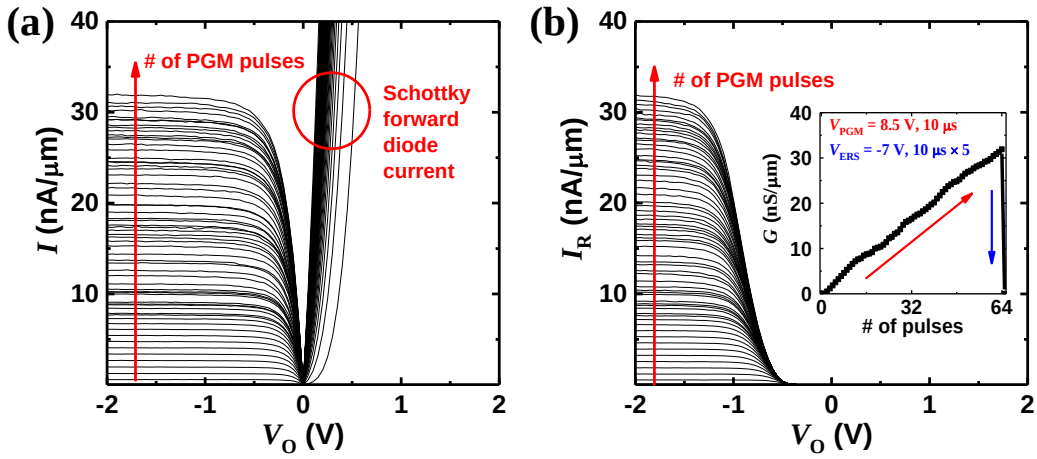



Fig. 2.6. Measured *I-V* curves of a *p*-type GSD obtained by applying $e^-$ programming pulses to the $BG_S$ (a) when −4 V is applied to the $BG_O$ and (b) 4 V is applied to the $BG_O$. Inset shows the conductance response with respect to the number of applied pulses.

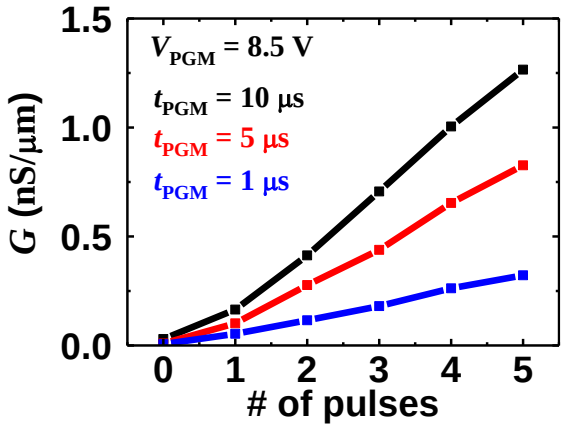



Fig. 2.7. Measured I-V curves of a p-type GSD when pulse width is 10 μs, 5 μs, and 1 μs.

# Chapter 3

# Neuron circuits

## 3.1 Necessity for input mapping

It is commonly assumed that the *I-V* characteristic of an electronic synapse device is linear in a hardware-based neural network composed of electronic synapse devices [33]. If an electronic synapse device shows linear *I-V* characteristics, the product of weight and input data can be simply expressed as the product of conductance and input voltage of the synapse device, which is the synaptic current. The difference in conductance between a pair of synapse devices is required to represent the negative weight. Then the product is given by

$$W \cdot x = \left(G^{+} - G^{-}\right) \cdot V = I^{+} - I^{-} \tag{1}$$

where $W$ and $x$ are weight and input of the neural network, and $G^{\pm}$, $V$, and $I^{\pm}$ are conductance, input voltage, and synaptic current of synapse device, respectively. In other words, since the conductance of a synapse device is independent of the input voltage, the synaptic current is the product of the input voltage and the conductance

of the synapse device. In this case, the various input data can be linearly mapped to the amplitude of the input voltage of synapse devices. However, unfortunately, most electronic devices exhibit nonlinear *I*-*V* characteristics, i.e., the conductance of synapse device depends on input voltage, as expressed in (2)

$$I^{\pm} = G^{\pm}(V) \cdot V. \tag{2}$$

In (2), the synaptic weight, represented by the conductance of synapse device, can change with respect to the input voltage in the propagation phase. If so, simply expressing the synaptic current as the product of the input voltage and the conductance can bring about error depending on the *I*-*V* nonlinearity as shown in Fig. 3.1. Therefore, the input data should be converted to the voltage that will be properly applied to the electronic synapse devices. To do this, there have been attempts to apply nonlinear mapping, nonlinear synaptic transmission function, or even improve the *I*-*V* linearity of the device [33]–[35]. Those are based on the amplitude modulation of the input voltage. In another approach, the time modulation of input voltage was reported [36]. While the methods that use nonlinear mapping or nonlinear synaptic transmission are only applied to specific

synapse device, the time-modulating method can be applied to any synapse device and can be easily implemented with electronic circuits.

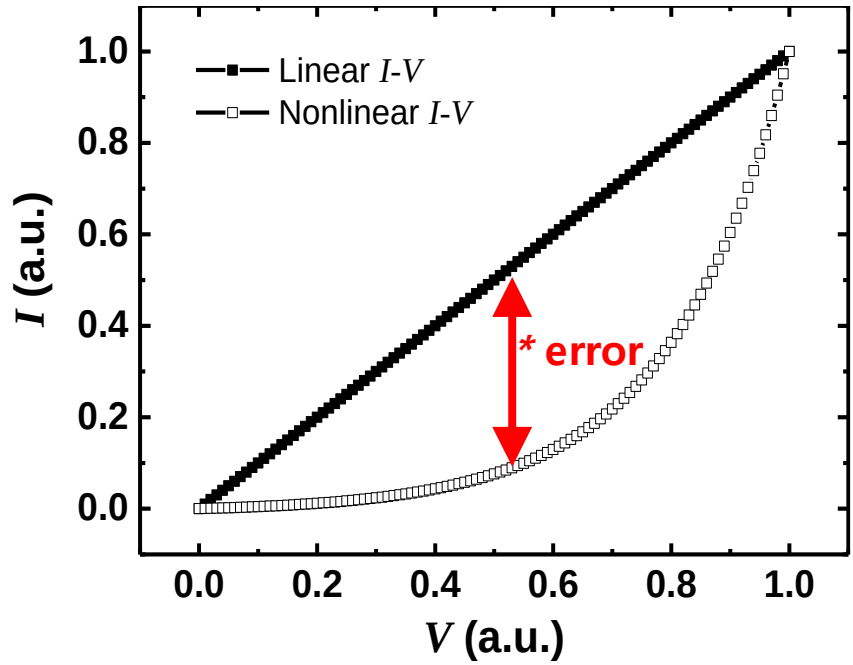



Fig. 3.1. Error caused by nonlinear *I-V* characteristics.

## 3.2 Pulse width modulation

To implement on-chip training scheme, the circuit elements to calculate how much weight to be updated and to update the weights using conductance of synapse devices are required. However, in order to do this, a method of updating the weights using hardware should be studied, so we will first discuss the circuit that enables forward propagation. The circuit elements for performing forward propagation are also required for backward propagation. To perform forward propagation using electronic circuits with synapse device array, the device characteristics of synapse device are fully considered. As shown in Fig. 2.6, our GSD has saturation region in *I-V* characteristic. In the saturation region where near linear conductance response with respect to the number of applied pulses is observed, the synaptic current as well as the conductance are independent of the input voltage. In this case, the input signal can be converted into a pulse, modulated with only the pulse width while maintaining the same amplitude. That is, the synapse output is the product of the pulse width and the unit synaptic current, as expressed in

$$Q^{\pm} = V_{\text{const.}} \cdot G^{\pm} \cdot t = I_{\text{unit}}{}^{\pm} \cdot t \tag{3}$$

where $Q^{\pm}$ and $t$ are the charges and the pulse width while a voltage pulse is applied, respectively. The unit synaptic current, which represents a specific conductance value (equivalently, synaptic weight), can only be changed if a program or erase pulse is applied to the $BG_S$ in the weight update phase. However, it is fixed in the propagation phase, so time modulation of input voltage pulse is required. For this purpose, we design the circuits for pulse-width modulation (PWM) for the inference system using GSDs. Note that the rate modulation of input information instead of the PWM can also be applied because both are the time modulation of the input information. Fig. 3.2 (a) shows a simple circuit for PWM. For the sake of the circuit simplicity, the operating voltage is a positive voltage and the circuit symbols of reverse-biased *n*-type GSD are used. The amplitude of input voltage applied to the X ($V_X$), which is an analog value, is compared with the amplitude of sawtooth wave ($V_{sawtooth}$), and the difference between the input voltage and the sawtooth wave is amplified. Then, by using level shifter, we can obtain a voltage pulse ($V_{in,s}$) with a modulated pulse width and a desired amplitude. Fig. 3.2 (b) shows voltage pulses with modulated pulse width obtained from the PWM circuits when the input

voltages are 0.6 V and 0.8 V, as an example. Since the max amplitude and the pulse width of the sawtooth wave are 1 V and 10 μs, respectively, the modulated pulse widths for 0.6 V and 0.8 V are 6 μs and 8 μs, respectively. Then, the synaptic current flows while the voltage pulse ($V_{in,s}$) is applied, as shown in Fig. 3.2 (c). A voltage pulse from the PWM circuit is applied to the input node of the synapse device, which is repeated in all layers of the neural network.

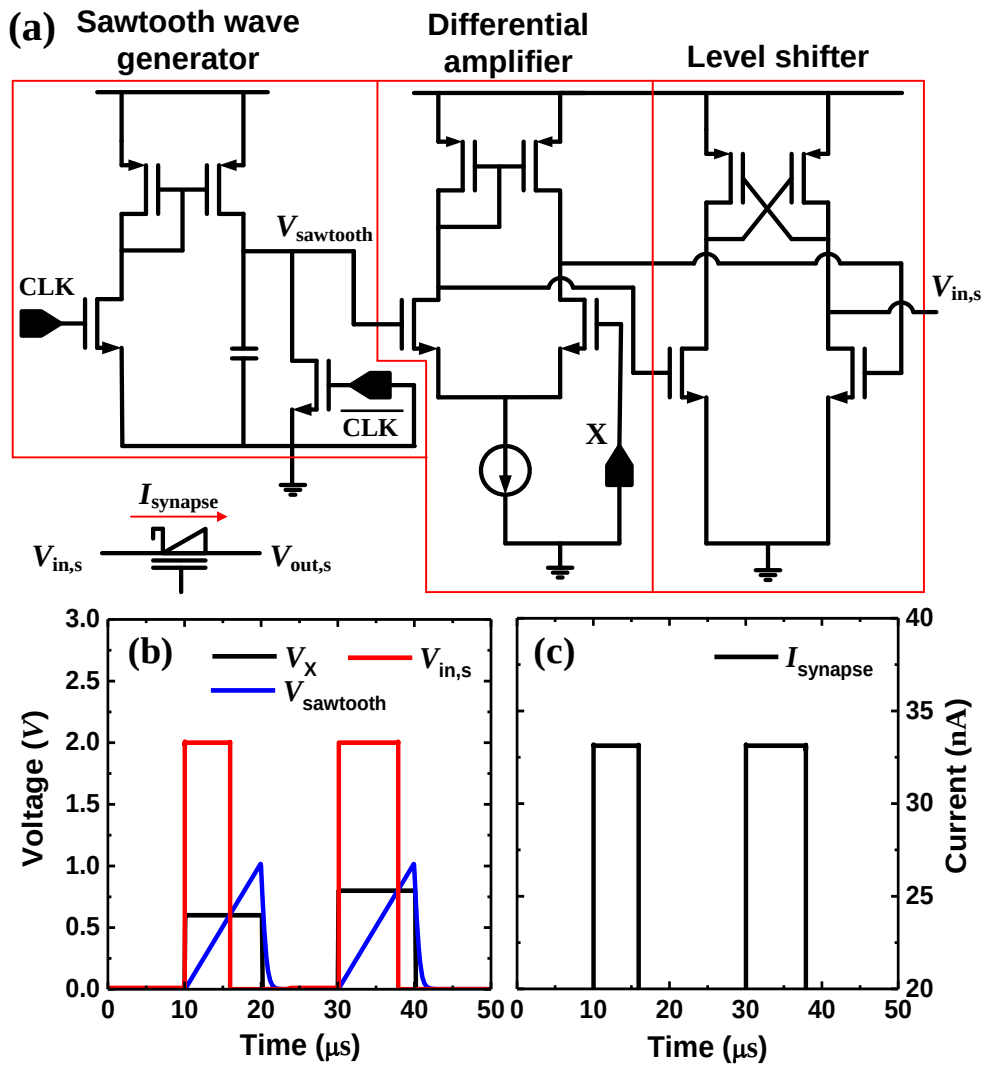



Fig. 3.2. (a) A pulse-width modulation (PWM) circuit designed with the assumption of *n*-type GSDs. (b) Voltage pulses ($V_{in,s}$) with different pulse widths modulated by the PWM circuit. (c) Synaptic currents corresponding the modulated voltage pulses.

## 3.3 Current sum and activation function

The VMM in software-based deep neural networks is accompanied by a large amount of power consumption. When this VMM is performed by a large electronic synapse device array, it is important to accurately represent the current sum of the synapse devices connected to a neuron. Since the current sum at the output of connected synapse devices changes over time, the output node voltage of the synapse devices varies with time. As a result, the synaptic current changes if the current is not saturated while the output node voltage is changing. Therefore, the change of the output node voltage results in degradation of the inference accuracy [37]. To handle this problem, the use of operational amplifier with a high gain can be a simple solution [38]. However, an operational amplifier with high gain generally requires large area and high power consumption. If the current at each synapse is constant (saturated) with the voltage change at the output of the synapses, operational amplifier is unnecessary. Because our GSD has a constant current even if the voltage across the diode varies over the operating voltage range, we can use a simple 2-stage current mirror circuit for current summation and subtraction. Note

that such a simple current mirror circuit cannot be used when the change in output node voltage changes the synaptic current. Fig. 3.3 (a) shows the current mirror circuits for the current summing, subtraction, and activation function. 785 synapses are connected to a single neuron and each synapse consists of two GSDs to represent both positive and negative weights. Note the *n*MOSFETs in current mirror circuit have enough current drivability (W/L=5) to accommodate the current from all synapse devices connected to one neuron. As mentioned above, the output node voltage ($V_{out,s}$) of synapse device can be changed with time, but the synaptic current does not change because the synaptic current is saturated. Fig. 3.3 (b) shows that the current of a synapse device is kept constant with respect to the voltage difference between the input node voltage ($V_{in,s\_i}$) and the output node voltage ($V_{out,s}^{+}$), as an example. Input node voltage ($V_{in,s\_i}$) of a synapse device is 2 V with a modulated pulse width, but the output node voltage ($V_{out,s}^{+}$) changes from about 0.3 V to 0.6 V. In other words, as the synaptic current flows, the voltage across the synapse device ($V_{in,s\_i}-V_{out,s}^{+}$) changes with time, but the synaptic current ($I^{+}$) does not change. It is an important issue when performing the VMM with analog operations using

electronic synapse device arrays. Furthermore, when the synapse device array is configured as a crossbar array, the IR drop along metal wires can cause the inaccurate VMM computation [39]. However, since our GSD operates in the saturation region, the noise of the input and output node voltages and IR drop along metal wires do not affect the current in the synapse device, so the VMM can be accurately performed. We use an integration capacitor (1.5 pF) for converting the current to the voltage, so the activation function is the hard-sigmoid function. Unlike sigmoid or hyperbolic tangent functions, this piece-wise linear function can be implemented by using a single capacitor without additional circuitry. Fig. 3.3 (c) shows the result of a hard-sigmoid function on the summation of charges ($Q_{tot}^{+}-Q_{tot}^{-}$) by synaptic currents ($I_{tot}^{+}$ and $I_{tot}^{-}$). The output value of hard-sigmoid function is again applied to the X node of the PWM circuit of the next layer in order to generate a voltage pulse having a modulated pulse width.

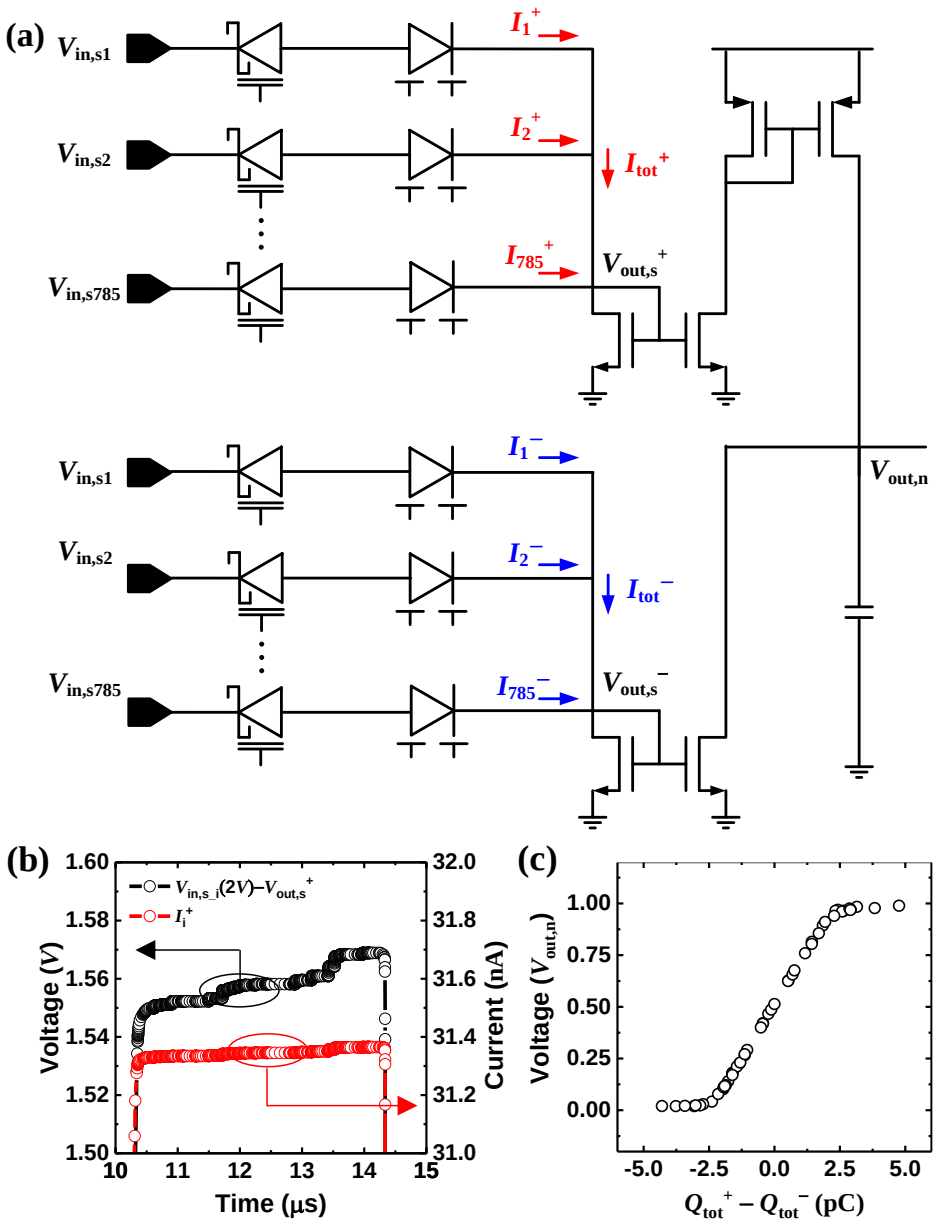



Fig. 3.3. (a) Current mirror circuits for current summing, subtraction and activation function. (b) Unchanged synaptic current with output voltage. (c) Activation function obtained from the current mirror circuits.

## 3.4 Verification of designed neuron circuits

We design a neural network consisting of 784 input, 50 hidden, and 10 output neurons to evaluate the designed electronic circuits including GSDs. For training in software, 60000 training images are applied, and the mini-batch size, the training epochs, and the learning rate are 100, 10, and 0.5, respectively. The classification accuracy rate for 10000 test images is 96.33%. After the weights are calculated using the algorithm in software, these weights are normalized and linearly quantized with discrete weights of 64 levels. After quantization of the weights, the classification accuracy rate for test sets is 96.30%. To map the pre-trained weights to synapse devices, the 64-level quantized weights are transferred to the synapse device array. Because our GSD shows linear conductance response in the potentiation, the linearly quantized weights can be applied to the synapse device array. As shown in Fig. 3.4, after normalization and quantization, we can obtain an index number representing how many pulses should be applied to the synapse device for the target weight. Although the read-write-verify scheme can also be used for precise tuning in off-chip training [40], it is more complicated method compared

to the method using an index number [41]. Since the conductance response is linear, the only thing to consider is the index number that represents the number of pulses to be applied. Based on these index numbers, all synapse devices have their own conductance values.

The designed neural network is evaluated by SPICE simulation. The designed circuit that performs the activation function and PWM can be considered as a neuron circuit. We obtained the MNIST classification accuracy for 100 randomly selected images from the test images. Fig. 3.5 shows, as an example, the output voltage obtained after the hard-sigmoid function of the last layer when the images from '0' to '9' are applied over time to the input neurons one by one. When the input image is '0', the output voltage of the neuron corresponding to '0' of the neurons in the last layer is the largest. Likewise, it is shown that the same results are obtained for other images, and the classification accuracy rate for 100 test images is 94%, which is similar to the software-based results. The designed neural network is evaluated only by 100 test images because of the time required for SPICE simulation, but we can confirm that the designed neural network works well.

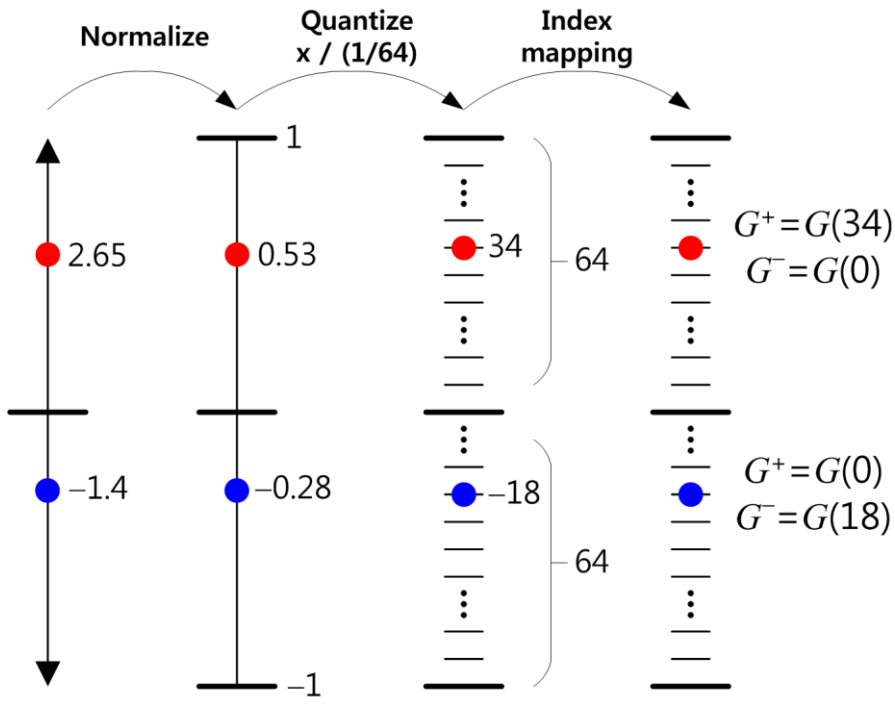



Fig. 3.4. Linear quantization of calculated weights and their mapping.

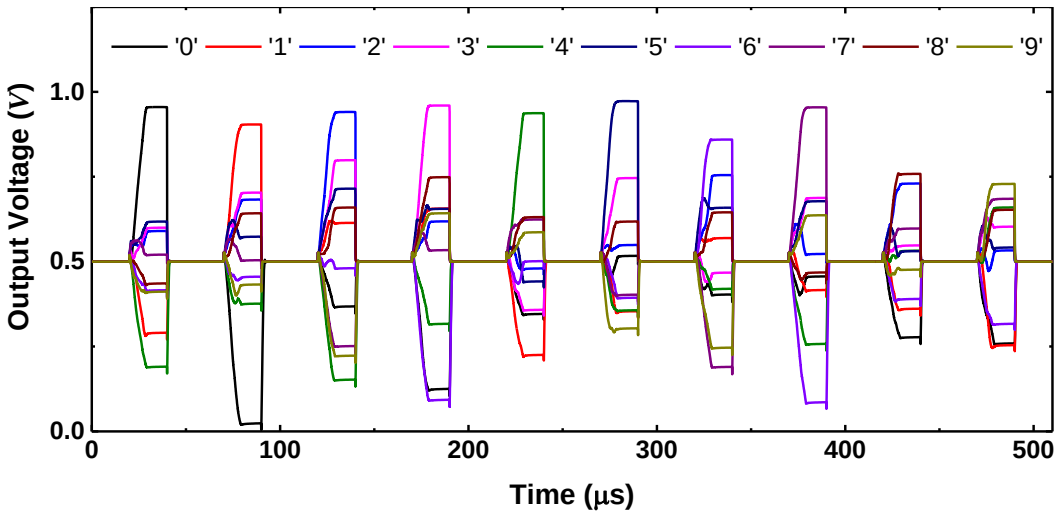



Fig. 3.5. Classification result for 10 images randomly selected from ‘0’ to ‘9’ obtained by SPICE simulation. Each image is input with an interval of 50 μs.

# Chapter 4
# On-chip learning rule

## 4.1 Manhattan update rule

We compared SW-DNN and HW-DNN, as shown in Table 1.1. However, in case of on-chip training in HW-DNN, it is difficult to compute a derivative value of the activation function and determine how much the weight changes. Since the conductance response of synapse device is finite, discrete and nonlinear, representing the weight by conductance of synapse device with high precision is almost impossible. In addition, a continuous derivative value of activation function cannot be efficiently implemented by using analog electronic circuit. Therefore, for ease of HW implementation, we used a hard-sigmoid function with derivative values of either 0 or $1/(2c)$. The modified learning rule for on-chip training in HW-DNN is described in Table 4.1. The activated value ($f(s_j)$) equals 0 when the weighted sum value ($s_j$) is less than $-c$ and equals 1 when the weighted sum value ($s_j$) is greater than $c$. Otherwise, the activated value ($f(s_j)$) represents a linear

function ($s_j/(2c)$), where $c$ is a positive constant. The parameter $c$ must be chosen by considering the training data sets and the number of electronic synapse devices connected to a post-neuron. Thus, this activated value ($f(s_j)$) has an output value (0 or 1 or $s_j/(2c)$) and an appropriate derivative value (0 or $1/(2c)$) for HW simplicity. With this help, the backward propagation can be easily performed. We should obtain the delta value of the $i^{\text{th}}$ neuron in the $l$–1 layer ($\delta_i^{(l-1)}$) by multiplying the derivative value of the activation function ($f'(s_i^{(l-1)})$) by the backward-weighted sum value ($\sum_j^M W_{ij}\delta_j^{(l)}$). Given that the hard-sigmoid function is used as the activation function, the derivative value is 0 or $1/(2c)$. In this case, we can replace the value of $1/(2c)$ with 1 without any logical error, indicating that the delta value of the $i^{\text{th}}$ neuron in the $l$–1 layer ($\delta_i^{(l-1)}$) is identical to the backward-weighted sum value ($\sum_j^M W_{ij}\delta_j^{(l)}$) or 0. Most important is how much the weight change. In SW-DNNs, the expression of $\Delta W_{ij}$ is the product of the learning rate ($\eta$), the delta value of the post-neuron ($\delta_j^{(l)}$), and the activated value of the pre-neuron ($f(s_i^{(l-1)})$). However, in HW-DNNs, the weight, which is expressed by the conductance of the electronic synapse device, shows discrete and limited

characteristics, indicating that the learning rate depends on the conductance step size. As the number of steps from minimum conductance to maximum conductance is limited in practical electronic devices, the learning rate ($\eta$) is not needed because the weight of a unit synapse is changed by one small conductance step size (one step) per iteration. In this case, it becomes necessary to consider the sign of the delta value of the post-neuron ($\mathrm{sgn}(\delta_j^{(l)})$) and the sign of the activated value of the pre-neuron ($\mathrm{sgn}(f(s_i^{(l-1)}))$), which is also referred to as Manhattan update rule [42]. There has been a study of applying such Manhattan update rule to perceptron networks implemented by a memristor crossbar array [17]. However, we suggest a rule for a more general neural network with hidden layers, taking into account the non-ideal properties of synapse devices and the ease of HW implementation. As the activated value of the pre-neuron is 0 or has a positive value, the sign of $\Delta W_{ij}$ ($\mathrm{sgn}(\Delta W_{ij})$) is 0 or the sign of the delta value of the post-neuron ($\mathrm{sgn}(\delta_j^{(l)})$). Therefore, we can replace the value of $1/(2c)$ with 1 without affecting the sign of the delta value of the post-neuron ($\mathrm{sgn}(\delta_j^{(l)})$). Since the derivative value can be considered as 1 in the linear region of activation function, the NNs can be expanded

more deeply. If the derivative of activation function is less than 1, the weights in the front layers cannot be updated because the gradients of the front layers decrease as the number of layers increases. This phenomenon is simply called the vanishing gradient problem. Similarly, in SW-DNNs, linear activation functions such as rectified linear unit (ReLU) is commonly used to avoid the vanishing gradient problem. Note that the hard-sigmoid can be considered as ReLU if the weighted sum value ($s_j$) is always less than $c$ by controlling the current flowing in the electronic synapse device. Above all, calculations using these simple gradient values can be implemented without difficulty on the HW. Identical pulses should then be generated by an electronic circuit according to $\mathrm{sgn}(\Delta W_{ij})$ and applied to the electronic synapse devices to change the conductance. Lastly, updating the weight of each training sample, which is a type of online learning, can easily be implemented on HW. If we adopt batch learning, computation in which the delta value of the post-neuron ($\delta_j^{(l)}$) is multiplied by the activated value of the pre-neuron ($f(s_i^{(l-1)})$) is an additional vector-by-matrix multiplication. This vector-by-matrix multiplication is not a product of the input signal and the weight and therefore

cannot be performed by an electronic synapse device array. Consequently, to make hardware implementation easier, online learning is appropriate.

Table. 4.1. Learning rule of SW- and HW-DNNs

| **Target** | **Software-based** | **Hardware-based** |
|---|---|---|
| Weights<br>$W_{ij}$ | $W_{ij}$ | $G_{ij}^{+} - G_{ij}^{-}$ |
| Forward<br>propagation<br>$S_j^{(l)}$ | $\sum_{i}^{N} W_{ij} a_i^{(l-1)}$ | $a_i^{(l-1)} \rightarrow V_i^{(l-1)}$<br>$\sum_{i}^{N} (G_{ij}^{+} - G_{ij}^{-}) V_i^{(l-1)}$ |
| Activated value<br>$a_j^{(l)}$ | $f\left(s_j^{(l)}\right)$ | $\begin{cases} 0 & \text{if } s_j^{(l)} < -c \\ 1 & \text{if } s_j^{(l)} > c \\ s_j^{(l)} & \text{else} \end{cases}$ |
| Backward<br>propagation<br>$\delta_i^{(l-1)}$ | $\sum_{j}^{M} W_{ij} \delta_j^{(l)} \cdot f'\left(s_i^{(l-1)}\right)$ | $\delta_j^{(l)} \rightarrow V_j^{(l)}$<br>$\begin{cases} \sum_{j}^{M} (G_{ij}^{+} - G_{ij}^{-}) V_j^{(l)} \cdot 0 \\ \sum_{j}^{M} (G_{ij}^{+} - G_{ij}^{-}) V_j^{(l)} \cdot 1 \end{cases}$ |
| Weight<br>updates<br>$\Delta W_{ij}$ | $-\eta \cdot \delta_j^{(l)} \cdot f\left(s_i^{(l-1)}\right)$ | $\begin{cases} \lvert \Delta G_{ij}^{+} \rvert & \text{if } \delta_j^{(l)} < 0 \\ -\lvert \Delta G_{ij}^{-} \rvert & \text{if } \delta_j^{(l)} > 0 \\ 0 & \text{if } f\left(s_i^{(l-1)}\right) = 0 \end{cases}$ |

## 4.2 Weight-updating methods

HW-DNNs using electronic synapse devices need a modified weight-updating method, unlike a SW-based BP algorithm. It is assumed that the weight is updated by changing the conductance of the electronic devices ($G^+$ or $G^-$) by one step. If multiple pulses are required to be applied to an electronic synapse device to arrive at a target conductance, it becomes necessary to check the current conductance of the device and determine the number of pulses required to obtain the target conductance. Because these procedures are needed for all electronic synapse devices, the external circuit can be a major burden. Furthermore, it will be much more difficult if the electronic synapse devices have nonlinear conductance responses. In other words, the use of multiple pulses for precise weight updates is impractical in actual electronic devices [16]. Therefore, we propose a weight-updating method based on a BP algorithm for HW-DNNs in which the amount of the weight change (equivalently, the learning rate in SW-DNNs) is determined by the amount of the conductance change of the electronic synapse devices.

We only need to consider the sign of the delta value of the post-neuron

($\mathrm{sgn}(\delta_j^{(l)})$) and the sign of the activated value of the pre-neuron ($\mathrm{sgn}(f(s_i^{(l-1)}))$) to determine whether the weight is changed or not. Because the activated value is positive, except when it is 0, only the sign of the delta value of the post-neuron ($\mathrm{sgn}(\delta_j^{(l)})$) is required to determine whether the weight is increased or decreased. Although this HW-based BP algorithm can be less accurate than a SW-based BP algorithm, relieving the burden on the external circuitry can be more helpful for power-efficient HW-DNNs. After determining whether the weight is to be increased or decreased depending on the sign of the delta value of the post-neuron ($\mathrm{sgn}(\delta_j^{(l)})$), the method by which the weight is changed by changing the conductance of the synapse device is important. As two identical electronic devices ($G^+$ and $G^-$) are necessary for a unit synapse device, the change in the weight can be expressed by the change in the conductance of each device. In other words, we can obtain the same result of increasing the weight by only increasing $G^+$ or decreasing $G^-$. However, as this is slightly redundant work, we choose only to increase the conductance to change the weight. The detailed weight-updating methods are described in Fig. 2. If $G^+ < G_{\max}$, and $\Delta W > 0$ is necessary, $G^+$ should be

increased by one step. Likewise, $G^-$ should be increased under a condition of $G^- < G_{\max}$, with $\Delta W < 0$. We simply need to determine whether $G^+$ or $G^-$ is to be increased to update the weight. However, when $G^+$ or $G^-$ reaches $G_{\max}$, there can be three different weight-updating methods because we cannot update the weight any more by increasing the conductance of $G^+$ or $G^-$. As an example, a case when $G^+ = G_{\max}$ and $\Delta W > 0$ is given, with both $G^+$ and $G^-$ initialized, with a subsequent increase in $G^+$ a possible solution according to the update method described in Fig. 4.1 (a). This update method has been reported [43]. For the same case, it is also possible to initialize $G^-$, followed by increasing $G^-$ to a conductance level lower by one step than the previous value, as shown in Fig. 4.1 (b). These two weight-updating methods can be implemented by electronic synapse devices which have unidirectional conductance responses and which can be reset. Although the increase in the iterative conductance after initializing can be tedious, it is not a frequent occurrence and is therefore not a major burden. In a bidirectional device, the one-step depression of $G^-$ is equivalent to an increase in the weight by one step, which has a degree of freedom compared to that of a unidirectional device

(Fig. 4.1 (c)). Additionally, it is necessary to initialize both $G^+$ and $G^-$ when they reach $G_{\max}$, because a lower conductance level results in less power consumption (Fig. 4.1 (d)). If the target $W$ is zero, either $G^+ = G^- = G_{\min}$ or $G^+ = G^- = G_{\max}$ results in the same zero $W$. We can then significantly reduce the power consumption by taking, for example, $G^+ = G^- = G_{\min}$ instead of $G^+ = G^- = G_{\max}$. Because there are many synapses in arrays for large neural networks, high conductance of electronic synapse devices can bring about high power consumption. As these update methods can control the electronic synapse devices which have discrete and limited conductance responses, they can be applied to HW-DNNs capable of running a BP algorithm.

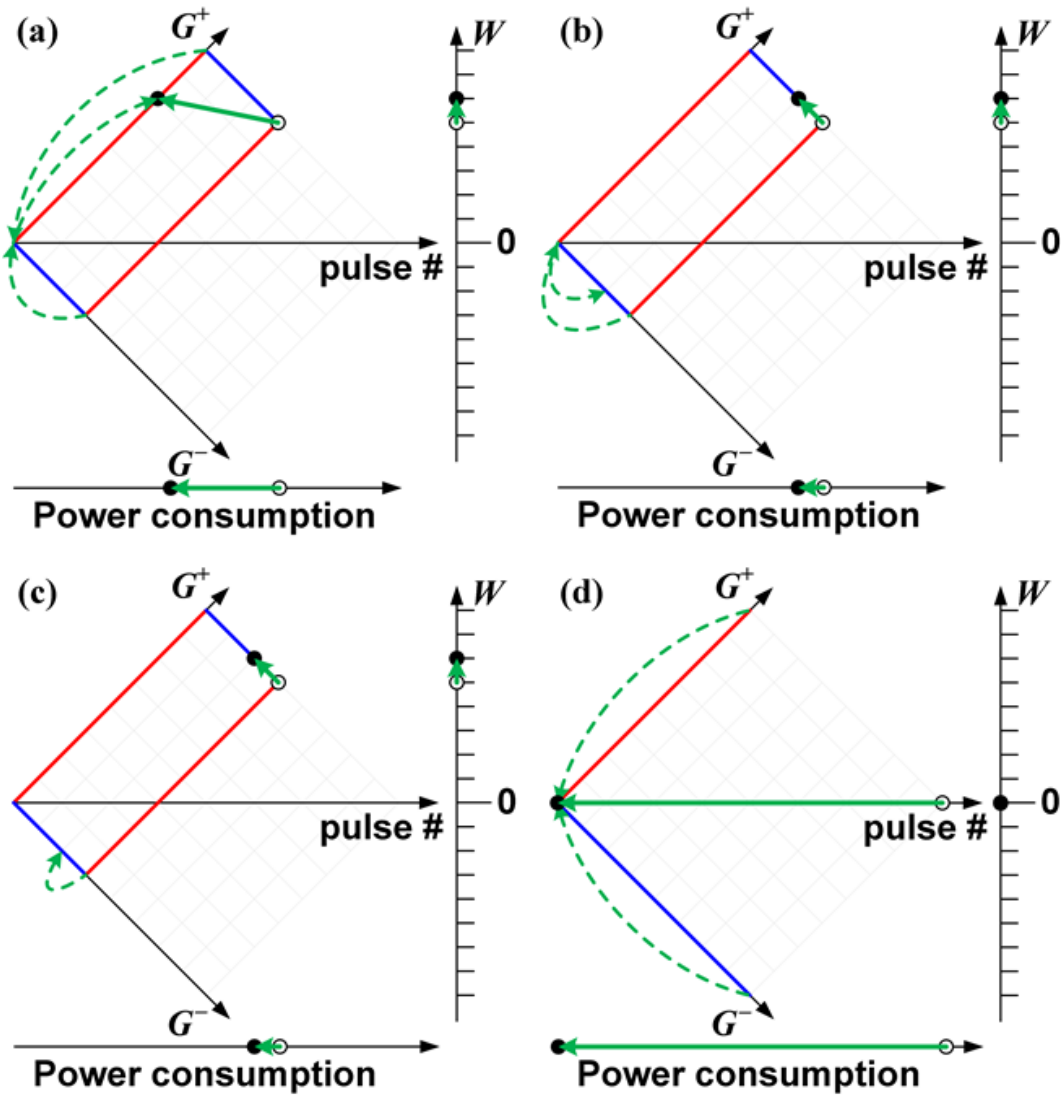



Fig. 4.1. (a)–(c) Weight-updating methods when $G^+$ reaches $G_{\max}$: (a) reported unidirectional update method [43], (b) proposed unidirectional update method, (c) conventional bidirectional update method, and (d) initialization method when both $G^+$ and $G^-$ reach $G_{\max}$

## 4.3 Evaluation of learning rule

To evaluate the learning rule mentioned above, we designed a neural networks for MNIST classification. In case of Manhattan update rule, we need to define the error tolerance between the output value of the last layer (the value of output neuron where the activation function is softmax function) and the target value to avoid unnecessary weight updates. In other words, if the error between the output value and the target value (0 or 1) is within a small range, the output value can be regarded as the 0 or 1. Otherwise, the weight updates based on the sign of delta value ($\mathrm{sgn}(\delta_j^{(l)})$) are consistently performed even if the error is small, and it can result in degradation of learning performance. That is, when the error is small, the discrete weight updating step can be excessive. In this paper, the error tolerance is set within 10% of the target value. For example, if the output value is higher than 0.9, it is regarded as 1. In addition, to evaluate the accuracy with the non-ideal characteristics of an electronic synapse device, a behavioral model from the literature [44] shown in (1) and (2) is used.

$$G_p(n+1) = G_p(n) + \alpha_p \exp\left(-\beta_p \frac{G_p(n) - G_{\min}}{G_{\max} - G_{\min}}\right) \quad (1)$$

$$G_d(n+1) = G_d(n) - \alpha_d \exp\left(-\beta_d \frac{G_{\max} - G_d(n)}{G_{\max} - G_{\min}}\right) \quad (2)$$

where $\alpha_p$ and $\beta_p$ are the fitting parameters of the potentiation characteristic. Likewise, $\alpha_d$ and $\beta_d$ are fitting parameters for the depression characteristic. Moreover, $G(n)$ denotes the conductance of the electronic synapse device when $n$ pulses are applied, and $G_{\max}$ and $G_{\min}$ are maximum and minimum conductance, respectively. If $n_{\max}$ pulses are needed to progress from $G_{\min}$ to $G_{\max}$, $n_{\max}$ can be defined as the dynamic range. These equations indicate how the conductance changes with different current conductance values. Fig. 4.2 shows the normalized conductance according to the number of pulses with respect to the parameter $\beta$. The parameter $\beta$ represents the nonlinearity of the electronic synapse device, and the parameter $\alpha$ determines the step size of the conductance change at given level of nonlinearity ($\beta$). A larger $\beta$ means greater nonlinearity. Commonly used electronic devices as synapse devices such as RRAM and PCRAM exhibit asymmetric nonlinearity with a high rate of increase at low conductance and a highly decreasing rate at high conductance. Therefore, we evaluate only the

asymmetric nonlinear shape, as mentioned above. Additionally, Fig. 4.2 shows that the conductance of each device ranges from the minimum value to the maximum value with $n_{\max}$ number of pulses, known as the dynamic range. Because an actual electronic device has a limited dynamic range, this dynamic range is an important factor related to the learning performance. In this work, we evaluate the accuracy with respect to the weight-updating methods when the nonlinearities are 0, 1, 2, and 3 and the dynamic ranges are 32, 64 and 128 (equivalently 5, 6, and 7 bits). Figs 4.3 and 4.4 represent the classification accuracies with respect to the non-idealities of synapse devices. Methods a, b, and c denote the weight-updating methods in Fig. 4.1 (a)-(c), respectively. The classification accuracy using a SW-based BP algorithm (continuous weight step) is displayed in the figures with a black symbol for reference. The red, green, and blue symbols in the figures represent the classification accuracies for the HW-based BP algorithm with weight-updating methods a, b, and c, respectively.

First, the linear conductance responses show the highest and most stable accuracy. For a better analysis of the nonlinearity effect in Fig. 4.3, the classification

accuracy with respect to the nonlinearity ($\beta$) is shown in Fig. 4.2 as a parameter of weight-updating method. The dynamic range ($n_{\max}$) is 64, and the mini-batch size is 1 (online learning). Given that the accuracy fluctuates due to the discrete weight steps, we depict the error bar obtained with the maximum and minimum accuracy levels in the last 100 iterations. The accuracy when using the SW-based BP is shown as a reference (black circle symbols); it fluctuates slightly as well due to the use of online learning. This figure indicates that the accuracy decreases when the nonlinearity increases. If the conductance response is linear, the conductance step size is identical regardless of the current conductance state, and the variation in $|\Delta W|$ at each weight update is zero. If the nonlinearity is increased, then the variation in $|\Delta W|$ increases, resulting in the degradation of the accuracy. It should be noted that when using method b there is relatively less degradation in accuracy compared to methods a and c. The average accuracy rates for method b are 95.36%, 95.59%, 94.80%, and 93.71%, with nonlinearities ($\beta$) of 0, 1, 2 and 3, respectively. Note the degradation of the accuracy in method a is the most significant among three methods. As the nonlinearity increases, not only the average accuracy

degradation but also the accuracy fluctuation becomes serious. The behavior of the accuracy with weight-updating methods will be given in the explanation of Fig. 4.5.

Secondly, accuracy increases and stabilizes as the dynamic range of conductance increases. Fig. 4.4 shows the classification accuracy with respect to the dynamic range ($n_{\max}$) as a parameter of weight-updating method. The nonlinearity ($\beta$) is fixed at 2, and the mini-batch size is 1. A high dynamic range means high resolution (small step) in weight update. Consequently, the higher accuracy can be obtained with the better resolution. Similar to that shown in Fig. 4.3, method b shows higher accuracy and smaller accuracy fluctuation compared to methods a and c as the dynamic range ($n_{\max}$) increases from 32 to 128. The average accuracy rates for method b are 92.96%, 94.80%, and 94.71%, for dynamic ranges ($n_{\max}$) of 32, 64, and 128, respectively.

Why the accuracy depends on the weight-updating method is explained in Fig. 4.5. Note that the nonlinearity evaluated here is asymmetric. That is, it shows a highly increasing rate at a low conductance level and a greatly decreasing rate at a high conductance level. We must consider two cases for a proper analysis. The first

case is when $G^+ = G_{\max}$ and $\Delta W > 0$ are given (Fig. 11 (a)-(c)). The second case is when $G^+ = G_{\max}$ and $\Delta W < 0$ are given (Fig. 11 (d)). We assume that both cases have same weights before the update, but the target weights in the two cases are in the opposite directions. The weights in the first and second cases should be increased and decreased, respectively. Applying method a in the first case, both $G^+$ and $G^-$ are initialized and then $G^+$ increases to the target weight in the low conductance region, resulting in a relatively high increasing rate of the weight (Fig. 4.5 (a)). However, a low decrease in the weight rate is obtained in the second case (Fig. 4.5 (d)). Thus, it is clear that the asymmetry between the weight increase and decrease is high when method a is applied. Similarly, if method c is applied in the first case, $G^-$ must be decreased rather than initialization taking place, thus resulting in an increase of $W$ (Fig. 4.5 (c)). For an asymmetric shape, decrease in the rate of the conductance differs from the increase in the rate of the conductance except when the current conductance is in the middle range. This difference is relatively high in the high $G^-$ region. Therefore, we can also confirm that the asymmetry between the weight increase and decrease is significant when method c

is applied. However, the accuracy obtained using method c is better than the accuracy obtained using method a. In method c, when $G^-$ decreases, $G^-$ gains a new conductance value that is not on the increase curve. Better accuracy can be achieved by effectively providing more dense steps. If method b is applied in the first case, $G^-$ must be initialized, and subsequently it increases (Fig. 4.5 (b)). Compared to the second case (Fig. 4.5 (d)), it can be seen that the difference between the weight increase and decrease is relatively small because the conductance $G$ changes by only one step from the current state. Therefore, we hold that the proposed method b can implement the closest weight change to symmetry. For this reason, the highest and the most stable accuracy rate is obtained when method b is applied. Because most synapse devices have nonlinear and asymmetric conductance responses, the proposed weight-updating method (b) will be suitable for HW-DNNs. We can say that just the unidirectional conductance response of the electronic synapse device is sufficient when update method b is applied.

Although we used three-layer perceptron networks, the proposed learning rule can also be applied to deeper neural networks with 3 and 5 hidden layers because

there is no vanishing gradient problem. Typically, architectures consisting of multiple nonlinear hidden layers are called deep architectures. Fig. 4.6 shows the classification accuracy with respect to the number of hidden layers, and the number of neurons in each hidden layer is 200. The dynamic range is 64 and the mini-batch size is 1. When the nonlinearity ($\beta$) is 0 and 2, the accuracy is evaluated using the weight-updating method b. The accuracy rates of the SW-based BP and the HW-based BP ($\beta$ = 0, 2) are represented by the black circle, and the red and green square symbols, respectively. When the number of hidden layers are 3 and 5, we applied 3 training epochs to change a large number of weights to optimal values. The average accuracy rates for the SW-based BP are 93.83%, 95.53%, and 95.13%, when the number of hidden layers is 1, 3, and 5, respectively. Here, the neural network with 3 or 5 hidden layers requires a smaller learning rate than the neural network with 1 hidden layer. The average accuracy rates for the HW-based BP ($\beta$ = 0) are 95.36%, 95.37%, and 94.45%, when the number of hidden layers is 1, 3, and 5, respectively. Unlike the SW-based BP, the learning rate in HW-based BP is fixed because the weights are changed by one conductance step. With three hidden layers, both SW-

based and HW-based improve average accuracy compared to using one hidden layer in neural networks. However, the use of 5 hidden layers does not guarantee better results. When applying random initialization, networks deeper than neural networks with one or two hidden layers are generally found to be not better [45]. Thus, a neural network with many hidden layers needs regularization methods for performance improvement [45, 46], but it is beyond the scope of this paper. Above all, it was verified that the accuracy rate of the HW-based BP is comparable to that of the SW-based BP. If the regularization methods are applied, our learning rule can be suitable for neural networks that have many hidden layers to increase the accuracy rate.

Another important non-ideal characteristic of an electronic synapse device is the device-to-device variation. Even if we can find a physical mechanism that allows electronic synapse devices to have a linear and high dynamic range of the conductance response, we cannot avoid the device-to-device variation which arises during the fabrication process. Therefore, it is needed to investigate the learning accuracy rate in HW-DNNs when there is a variation between devices.

Fig. 4.7 shows the classification accuracy with respect to the standard deviation (σ). Here the nonlinearity ($\beta$), the dynamic range, and the mini-batch size are 2, 64, and 1, respectively. The weight-updating method b is applied. We compared the accuracy rates of on-chip learning and off-chip learning to evaluate the effect of device variations. When we perform off-chip learning, all of the synaptic weights are calculated assuming there is no variation between the devices, with the calculated weights then transferred to an electronic synapse array with the given standard deviation (σ). In HW-DNNs, it is impractical precisely to map from calculated weights to the electronic conductance while taking into account variations in the entire electronic synapse device array. Whereas when performing on-chip learning, the problems due to the device-to-device variation can be solved. Because HW-DNNs can perform backward propagation using an electronic synapse array, whether the weight should be increased or decreased can be determined based on the current conductance of each imperfect electronic synapse device. Therefore, we can obtain results showing that the accuracy of on-chip learning is not affected by device variations, whereas the accuracy of off-chip learning is significantly

reduced when device variations increase. The average accuracy rates of the on-chip learning are 94.92%, 94.81%, and 94.01%, when the standard deviations ($\sigma$) are 0, 0.5, and 1, respectively. Whereas, the average accuracies for the off-chip learning are 94.67%, 79.24%, and 57.34%, when the standard deviations ($\sigma$) are 0, 0.5, and 1, respectively. The error bars of the off-chip learning are obtained by repeating 10 simulations. In an actual electronic synapse device, a standard deviation ($\sigma$) of 0.5 is quite possible as an electronic synapse device should represent multiple analog conductance values with a high dynamic range, unlike digital memory cells having only a 0 or 1 state.

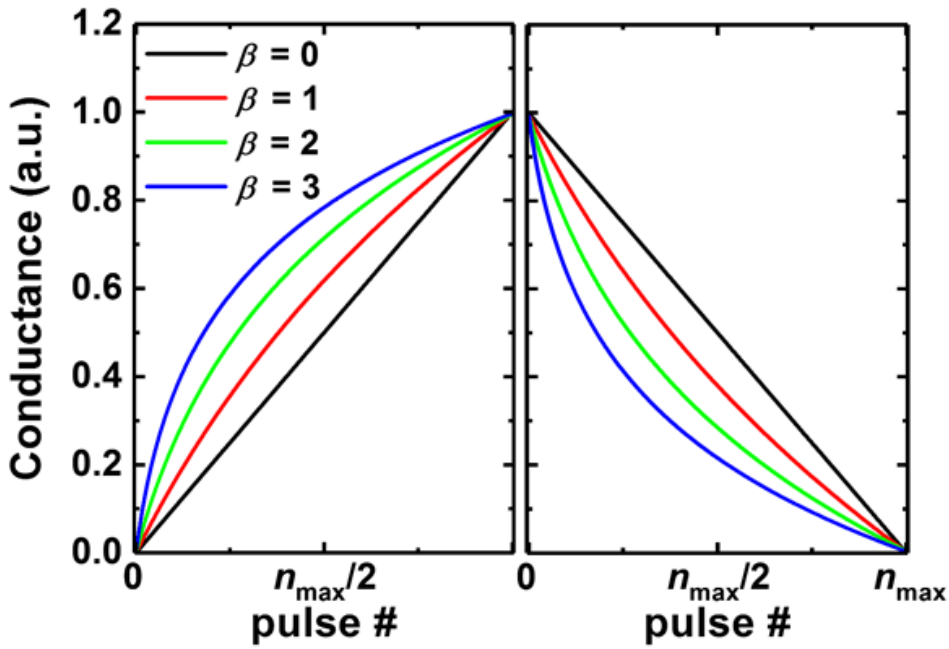



Fig. 4.2. Potentiation and depression characteristics with respect to the parameter $\beta$.

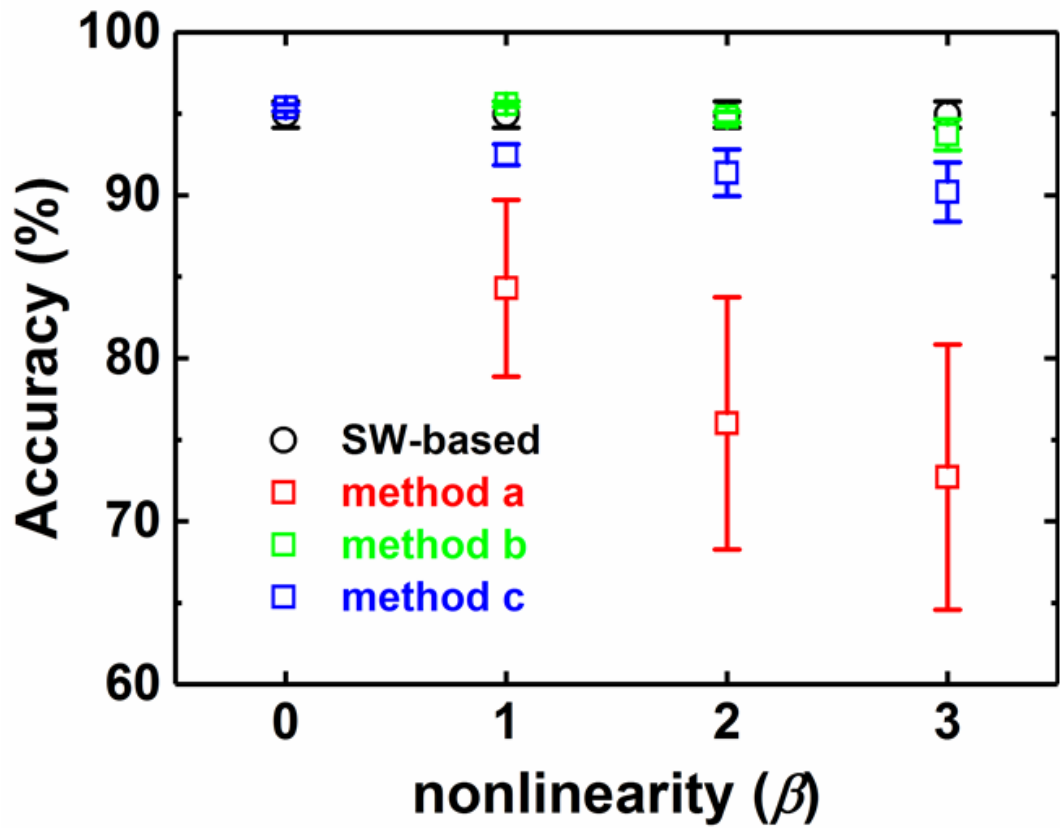



Fig. 4.3. Classification accuracy with respect to the nonlinearity ($\beta$) and weight-updating methods when the dynamic range ($n_{\max}$) is 64 and the mini-batch size is 1.

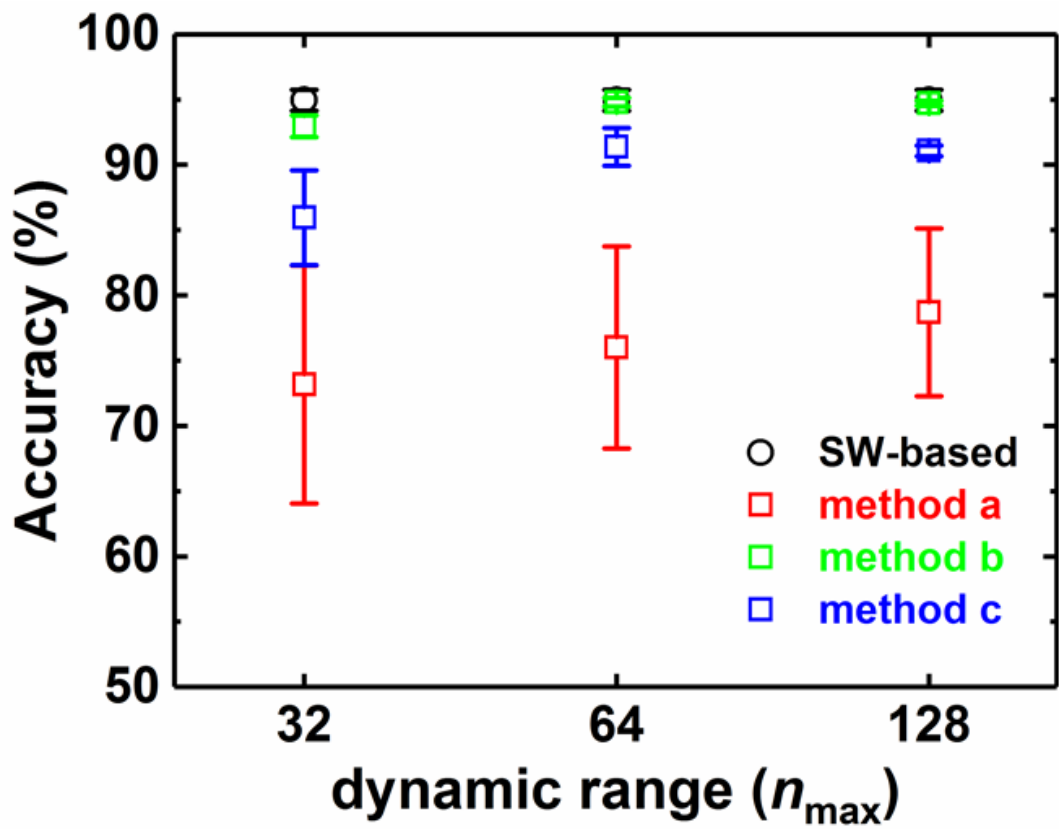



Fig. 4.4. Classification accuracy with respect to the dynamic range ($n_{\max}$) and weight-updating methods when the nonlinearity ($\beta$) is 2 and the mini-batch size is 1.

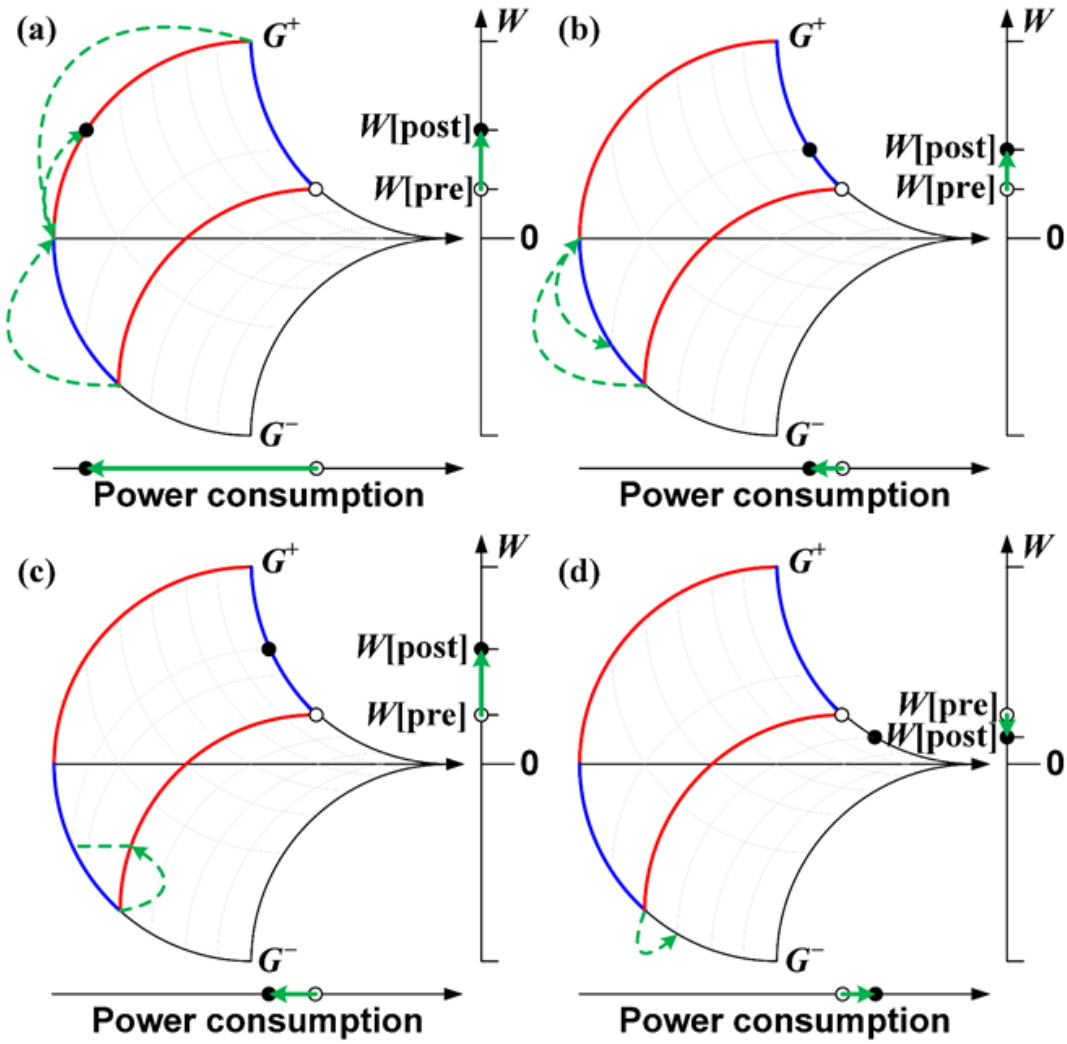



Fig. 4.5. (a)–(c) Weight updates when $G^+ = G_{\max}$ and $\Delta W > 0$ are given: (a) method a, (b) method b, and (c) method c. (d) Weight updates when $G^+ = G_{\max}$ and $\Delta W < 0$ are given.

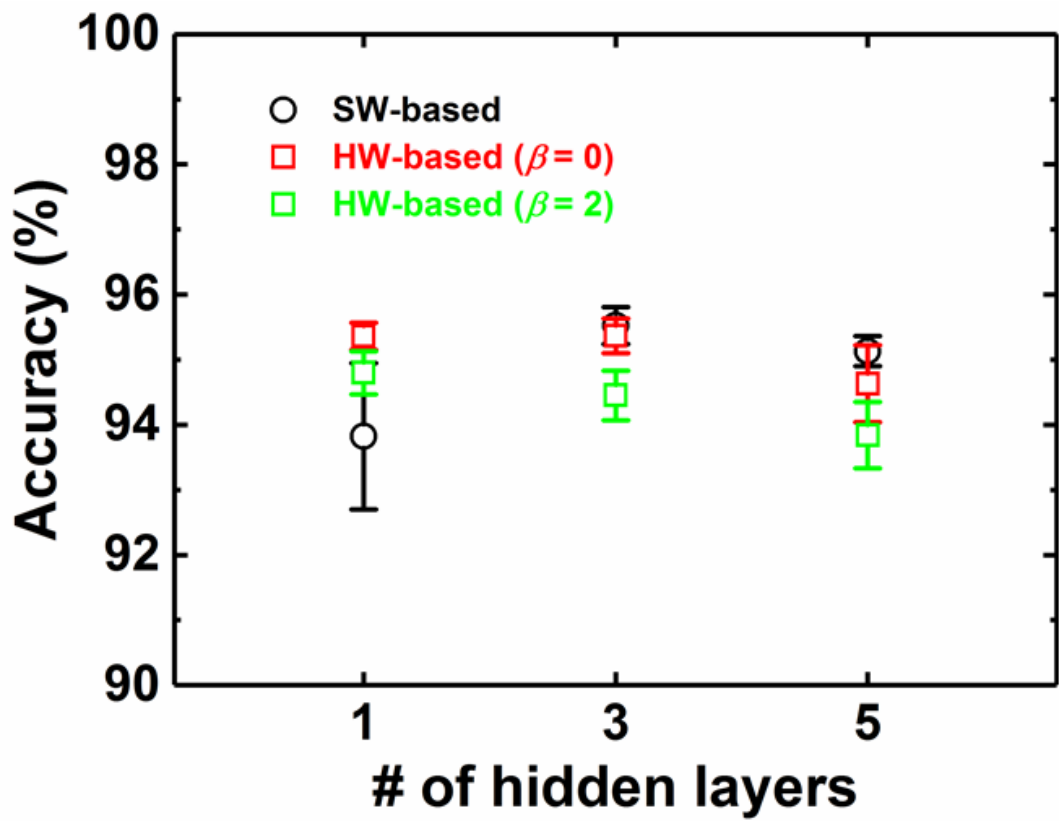



Fig. 4.6. Classification accuracy with respect to the number of hidden layers when the nonlinearity ($\beta$) is 0, the dynamic range ($n_{\max}$) is 64, the mini-batch size is 1, and method b is used.

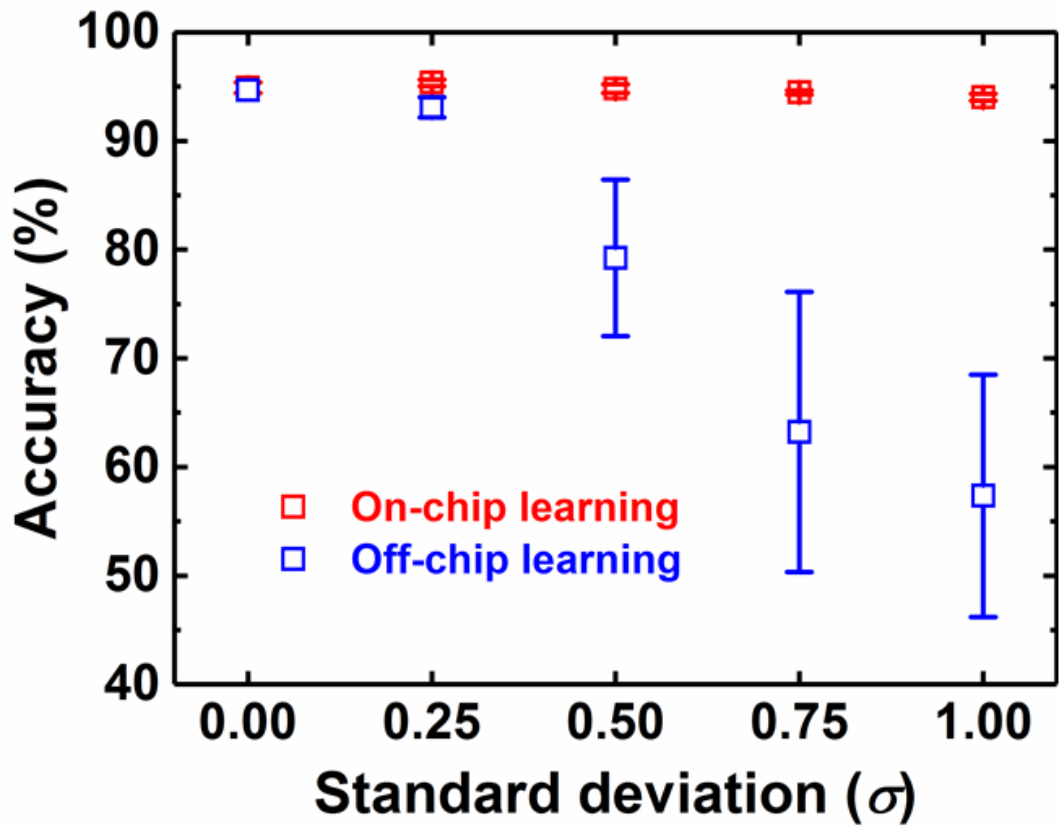



Fig. 4.7. Classification accuracy with respect to the standard deviation (σ) when the nonlinearity ($\beta$) is 2, the dynamic range ($n_{\max}$) is 64, and method b is used.

# Chapter 5
# Hardware implementation of neural networks

## 5.1 GSD for synapse array

We have successfully demonstrated a GSD that works as a synapse device in previous works [27], [36]. The reverse current of Schottky diode modulated by the Schottky barrier is used as the synaptic current. In addition, the reverse current of Schottky diode is saturated with respect to the input voltage, so that the synaptic current can be consistently maintained even with the input and output noise [31], [36]. Furthermore, we devised the bias scheme to prevent the forward current of Schottky diode, in Chapter 2. The GSD described in Chapter 2 consists of five electrodes to represent one synapse, but there are some electrodes that are not needed. The role of the $BG_O$ is only to determine whether the device type of GSD is *n*-type or *p*-type device, so we have merged the O and $BG_O$ nodes in the modified structure. The role of $BG_C$ can be replaced by applying an appropriate bias to the $BG_O$ node. Therefore, we can omit two electrodes without losing device

characteristics. The modified device structure is shown in Fig. 5.1. Two bottom gates are formed by $n^+$-poly silicon, and $SiO_2/Si_3N_4/SiO_2$ layers are deposited. The $Si_3N_4$ layer is used for the charge storage. Then, the un-doped silicon, which is the active layer of the GSD is deposited on the $SiO_2/Si_3N_4/SiO_2$ stack. After defining the contact holes on the active layer, Ti/TiN/Al/TiN are deposited consecutively by radio frequency (RF) sputtering and patterned using etch process. By applying program or erase pulses to the $BG_S$ and $BG_O$ to modulate the Schottky barrier height, the contact between the metal and the silicon active layer can be a Schottky and ohmic contact. The modified GSD makes the array configuration simple and operates like the previous GSD. For the sake of simplicity, this modified GSD is referred as GSD below. The bias condition of the *n*-/*p*-type GSDs is shown in Fig. 5.2. First, when the GSD operates as a synapse device, positive and negative input voltages (6/−6 V) are applied to the O node of *n*- and *p*-type GSDs, respectively. The Schottky barrier height of the S node is modulated by $V_{BGS}$ (2/−2 V), and electrons or holes flow from the S node to O node through this modulated the Schottky barrier height. These tunneling carriers determine the reverse current of

Schottky diode, which is the synaptic current. Note that the effect of $V_{BGS}$ can be replaced with the stored charge by applying program pulses to the $BG_S$. As the band diagram is shown in Fig. 5.3, when *n*-/*p*-type GSDs are in operating mode, the electrons or holes injected from the S node flow to the O node by the electric field. The diode current is saturated by pinch-off region, which is the B region in Fig. 5.3. On the other hand, when negative and positive voltages (−6/6 V) are applied to the O node of *n*- and *p*-type GSDs, respectively, the GSDs are in the cutoff region (Fig. 5.3). In case of the *n*-type GSD, the negative bias applied to the O node deplete electrons in the Si active layer on the O node, so that the forward current of Schottky diode cannot flow from the S node to the O node. In case of the *p*-type GSD, the positive bias applied to the O node accumulate electrons in the Si active layer on the O node, which results in the formation of reverse biased P-N junction. Because of this reverse biased P-N junction, the forward current of Schottky diode cannot flow from the O node to the S node. The NMOS of the *n*-type GSD and the P-N diode of the *p*-type GSD, which are intrinsically formed, can suppress the Schottky forward current and are represented by the circuit symbol in Fig. 5.2. Based on these

simulation results, the measured *I*-*V* characteristics of the GSDs shown in Fig. 5.4 can be explained. The reverse current of Schottky diode represented by $I_R$ according to the input voltage ($V_O$) is modulated by the $V_{BGS}$ (Fig. 5.4 (a)). The red and blue lines represent the synaptic current of the *n*-type and *p*-type GSD, respectively. The reverse current of Schottky diode, the synaptic current, is saturated as the input voltage ($V_O$) increases. The forward current of Schottky diode is suppressed by the device connected to the O node in Fig. 5.2 when the sign of $V_O$ changes. As mentioned before, the rectification is important for synapse array configuration. In addition, the current saturation is immune to the noise of the input and output nodes and the IR drop problem in crossbar array. When the synaptic current is saturated with respect to the input voltage, various input values can be accommodated by modulating the width of the pulse applied to the synapse device [36]. Fig. 5.4 (b) shows the conductance behavior with respect to the number of applied program pulses, in case of the *p*-type GSD. By applying a positive voltage pulse to the $BG_S$, electrons are stored in the charge storage layer ($Si_3N_4$). These stored electrons induce the holes in the Si active layer, so that the holes lower the Schottky barrier

height. Note that a negative voltage pulse is required for the increase of the reverse current of Schottky diode, in case of the *n*-type GSD. When 16 (4-bit) consecutive identical pulses are applied to the $BG_S$, the reverse current of Schottky diode consistently increases. Such modulated reverse current of Schottky diode is regarded as the synaptic weight. Therefore, the $BG_S$ node is used to control the synaptic weight, and O/S nodes are used for input/output, respectively.

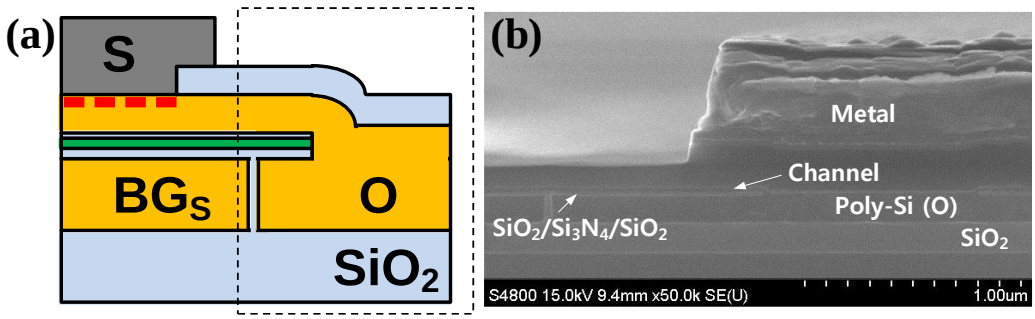



Fig. 5.1. Schematic and SEM cross-sectional views of modified GSDs.

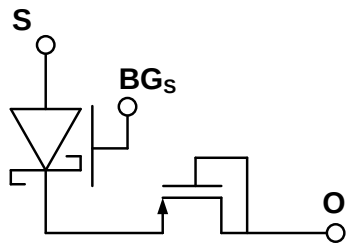
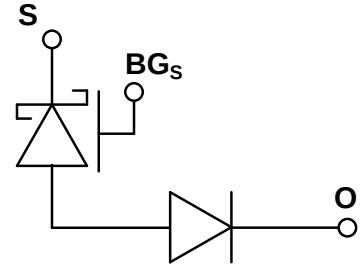
| | *n*-type | | *p*-type | |
|---|---|---|---|---|
| | Operating | Cutoff | Operating | Cutoff |
| $V_{BGs}$ | 2 V | 2 V | -2 V | -2 V |
| $V_O$ (Input) | 6 V | -6 V | -6 V | 6 V |
| $V_S$ (Output) | 0 V | 0 V | 0 V | 0 V |
| Symbol |  | |  | |



Fig. 5.2. The bias condition and the circuit symbols when *n*-/*p*-type GSD are operating and cutoff. The NMOS of the *n*-type GSD and the P-N diode of the *p*-type GSD are intrinsically formed by the device structure.

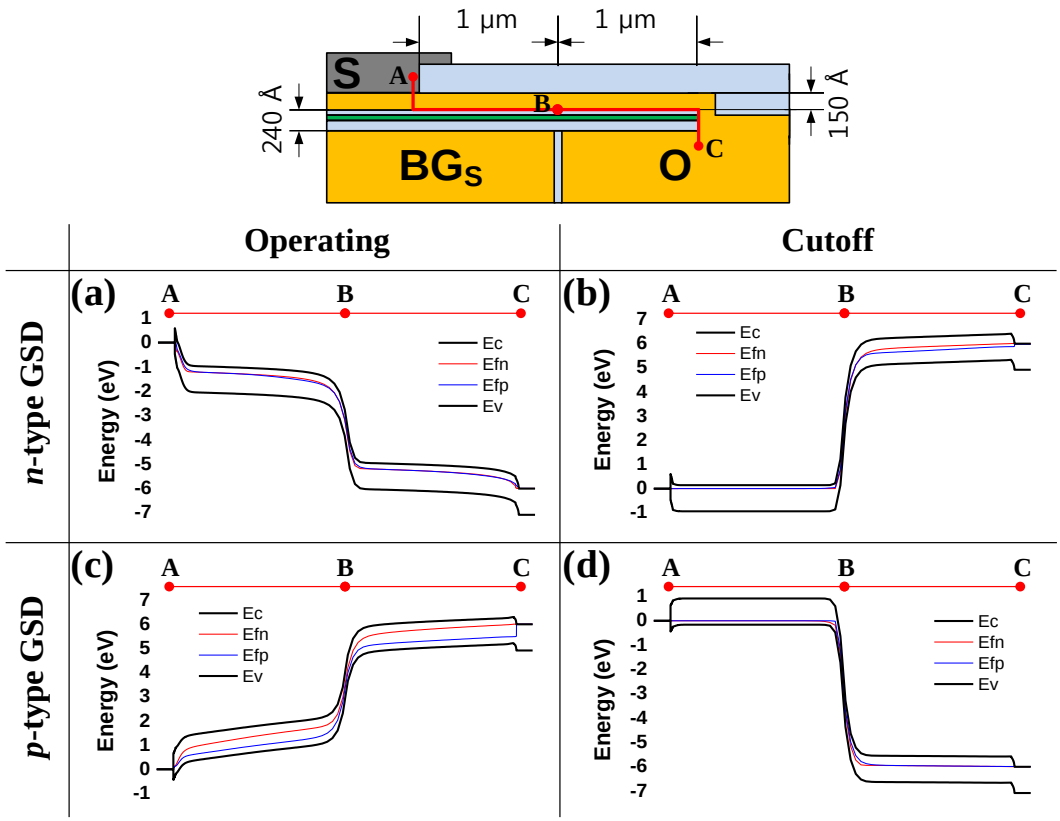



Fig. 5.3. The simulated band diagrams cut along the red line A-B-C when *n*-type GSD is (a) operating and (b) cutoff, and when *p*-type GSD is (c) operating and (d) cutoff.

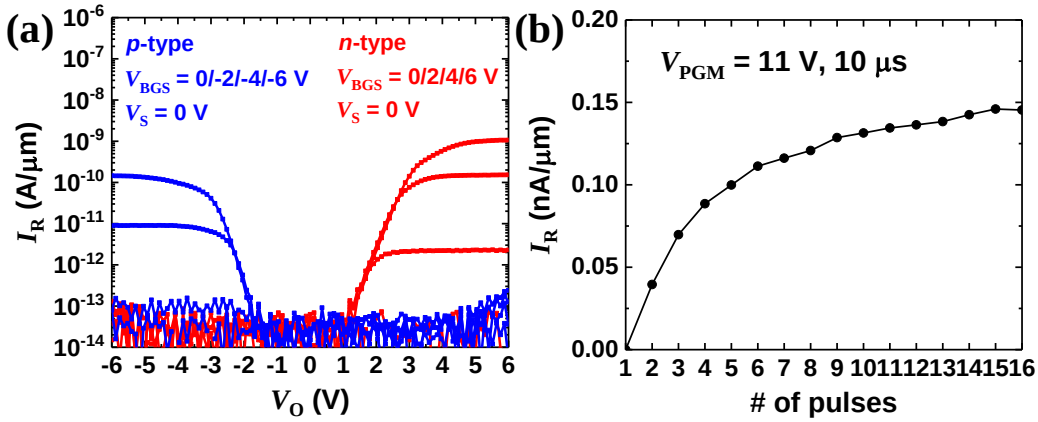



Fig. 5.4. (a) Input/output characteristics ($I_R/V_O$) when $V_{BGS}$ changes to modulate the Schottky barrier height. (b) In case of the *p*-type GSD, the conductance behavior with respect to the number of applied program pulses to the $BG_S$.

## 5.2 VMM using GSD array

When synapse devices are configured as a crossbar array, the sneak path and the IR drop problems are main challenges. The unintended current by the sneak path problem can result in inaccurate VMM computation. In addition, the IR drop along metal wires in crossbar array can distort the voltage across the synapse device [39], [47]. The synapse device at the far-end of the array is most affected by the IR drop, so size of the array can be limited. We assume for simplicity that the resistance of synapse devices are the same, and calculate the voltage across the synapse device at the far-end of the $N \times N$ crossbar array ($V_{NN}$) (Fig. 5.5 (a)). Even when the input voltage ($V_{input}$) is applied to the synapse device, the $V_{NN}$ can be changed from $V_{input}$ to a reduced voltage due to the IR drop along metal wires in crossbar array. The resistance of metal wire between adjacent synapse devices is assumed to be 2.5 Ω [47]. As shown in Fig. 5.5 (b), if the resistance of the synapse device is 5 kΩ (low resistance state in [47]) and the array size is 64×64, $V_{NN}$ becomes 31% of $V_{input}$. Therefore, the resistance of the synapse device even in the low resistance state should be sufficiently large considering the array size. However, the synapse array

consists of the GSD is free from these problems. As mentioned before, the synaptic current of the GSD flows only in one direction, so the sneak path current is cutoff. In addition, since the synaptic current is saturated with respect to the input voltage, the distortion of input voltage caused by the IR drop in metal wires does not affect the synaptic current of the GSD. These characteristics are very important in synapse array, and we fabricated a synapse array based on GSDs. The SEM view and part of the layout of the array are shown in Fig. 5.6. The view shows a synapse array consisting of 10 inputs, 20 outputs, and 10 control units. The input voltage is applied to the O node, and the output current is measured from the grounded S node. To control the synaptic weight, program or erase pulses can be applied to the $BG_S$ node. The total number of devices that make up the array is 200 (10×20). Except for the control nodes ($BG_S$), it is similar to a conventional crossbar array configured with RRAMs. Fig. 5.7 shows the distribution of the synaptic current in a GSD array. Input voltage ($V_O$) is −4 V, the synaptic current ($I_R$) is measured when the $V_{BGS}$ is −4/−5/−6 V. When $V_{BGS}$ is −4/−5/−6 V, the mean values of the synaptic current in 200 devices are 0.040/0.12/0.25 nA, respectively, and the standard deviations are

0.014/0.027/0.035 nA, respectively. The larger the synaptic current, the larger the standard deviation. The variation, represented by the standard deviation divided the mean (σ/μ) of the synaptic currents, at each weight state are obtained as 0.34, 0.22, and 0.14, respectively. Fig. 5.8 shows the vector-by-matrix multiplication operation performed by a GSD array. The vector-by-matrix multiplication is major burden in software calculation, but in the synapse device array, it can be easily performed by measuring the synaptic current. Ten inputs (−4 V) are applied to the GSD array, and one output current is measured when the $V_{BGS}$ is −6 V. The sum of current of ten individual GSDs ($I_1 + I_2 + \cdots I_{10}$) is almost the same as the $I_{tot}$ measured by applying the input voltage simultaneously to ten GSDs. In summary, GSD performs well as a synapse device with the help of rectification and saturation characteristics. In addition, the power consumption can be low due to the reverse diode operation and the device reliability is also good because it is a Si-based device.

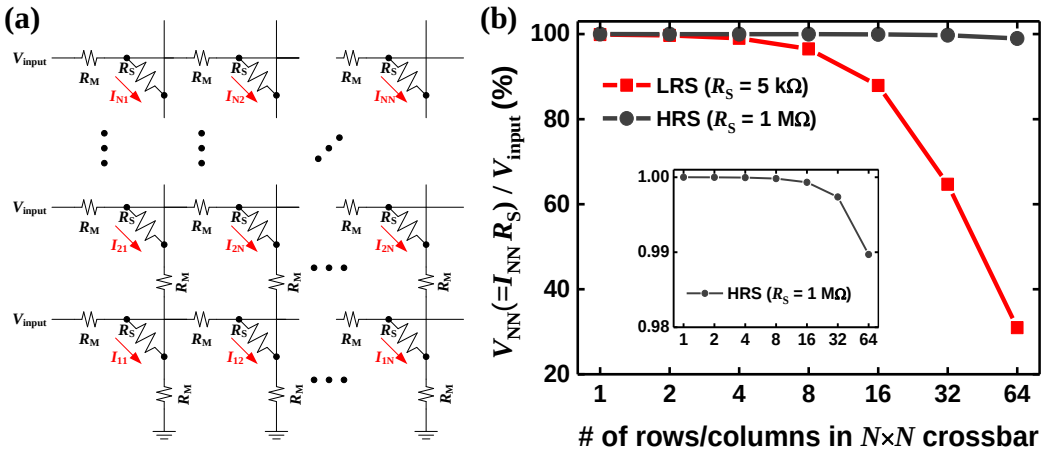



Fig. 5.5. (a) A crossbar array showing parasitic resistance along metal wires. (b) Voltage across the synapse device at the far-end of the array ($V_{NN}$)

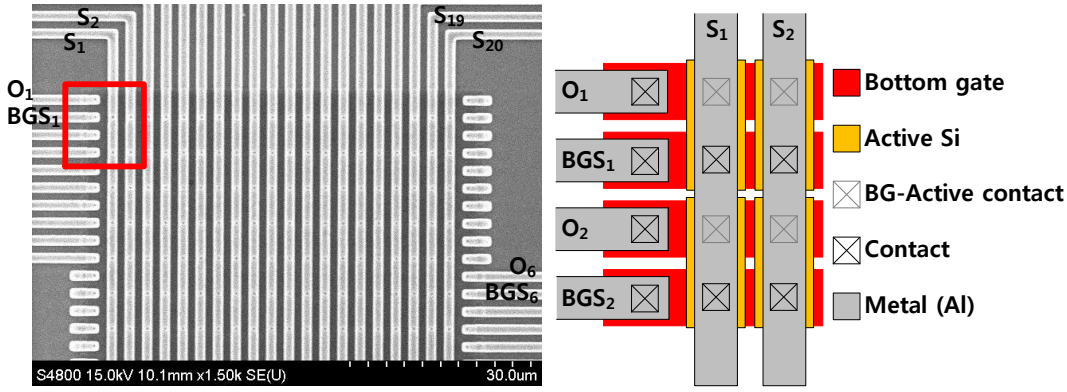



Fig. 5.6. The synapse array based on the GSDs.

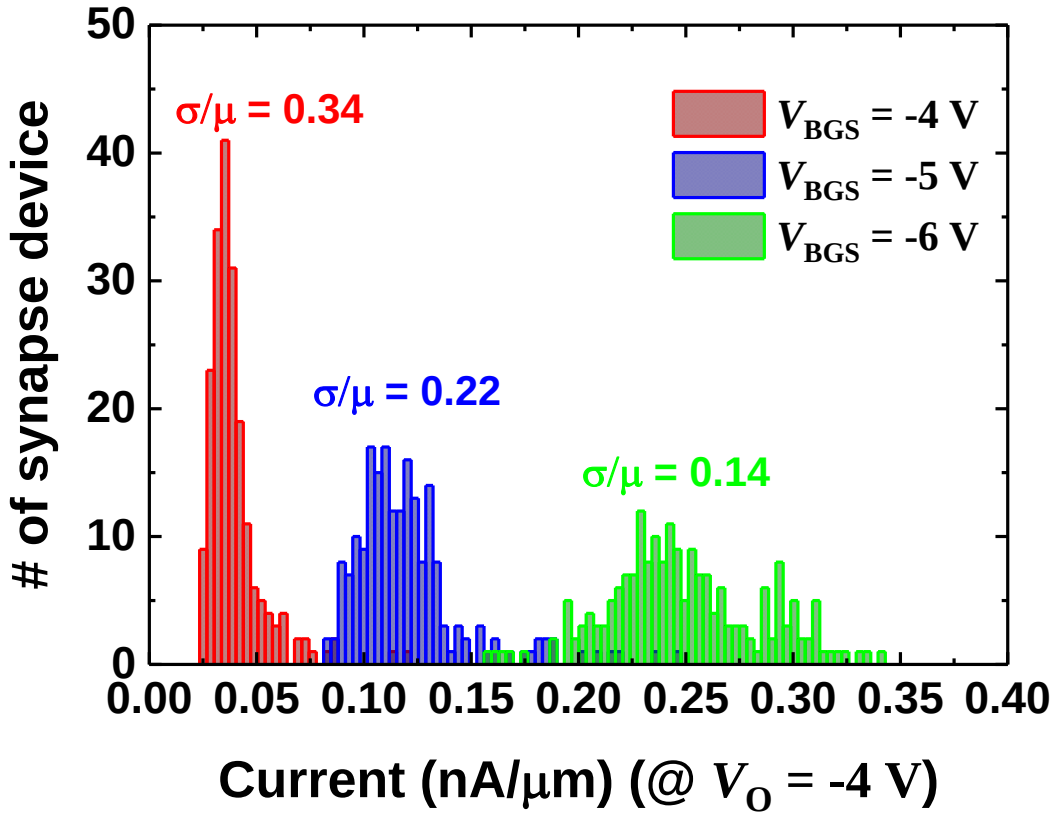



Fig. 5.7. Distribution of the synaptic current in a GSD array. Red, blue, and green boxes show three different weight levels represented by $I_R$ when $V_{BGS}$ is −4 V, −5 V, and −6 V, respectively.

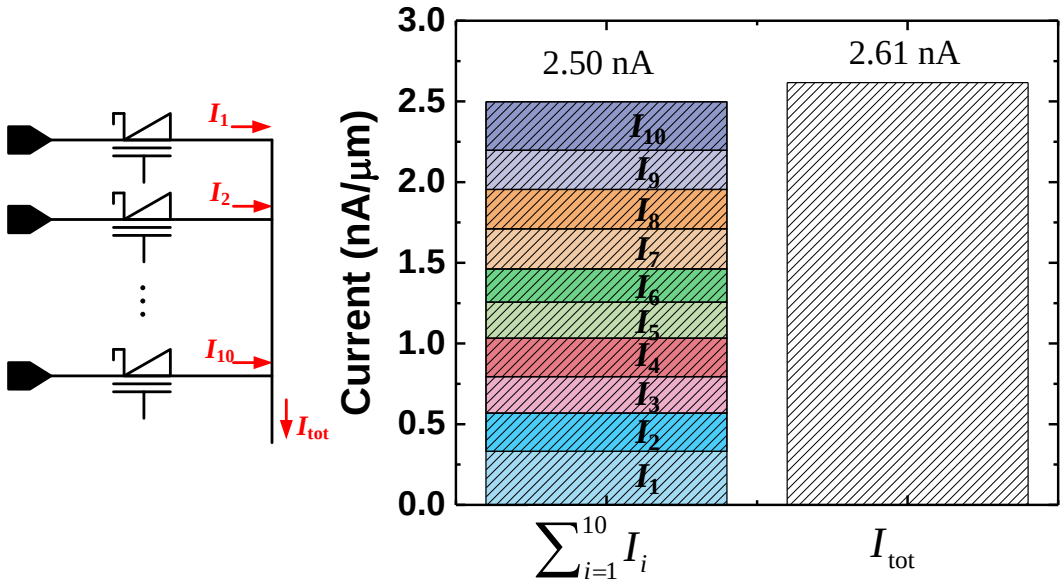



Fig. 5.8. Vector-by-matrix multiplication by using the GSD array. The output current is measured by applying the input voltage to the ten GSDs.

# Chapter 6
# Conclusion

In this work, we have investigated a neural network system based on the proposed gated Schottky diode (GSD). The GSD operates reverse Schottky diode, so the synaptic current is very low. In addition, it exhibits a saturated current characteristic with respect to the input voltage change, and thus has a high immunity to the noise voltage and IR drop along metal wires. This saturated synaptic current is modulated by the program pulses applied to the bottom gate under the Schottky junction between the aluminum electrode and silicon active layer. As the Schottky barrier height decreases with the applied program pulses, the reverse current of Schottky diode linearly increases with respect to the number of pulses. Considering these device characteristics, we have designed the neuron circuits. A pulse width modulation (PWM) scheme compatible with the saturation characteristics was introduced, and circuits supporting this were designed. In addition, the GSD shows linear conductance response, it is advantageous to copy the optimized weights in

software to synaptic devices. Using the MNIST test set in SPICE simulation, we obtained a 94% classification accuracy rate, which is very similar to the accuracy obtained by software-based learning. Furthermore, we studied the on-chip learning rule by considering various non-ideal characteristics of synapse devices. Using Manhattan update rule and three different weight-updating methods, we evaluated the learning performance of neural networks. If the conductance response is as linear as possible and the dynamic range is as high as possible, better learning performance is obtained. We also confirmed that the proposed unidirectional weight-updating method can mitigate the decline in the accuracy due to non-ideal characteristics of electronic synapse devices. Because most electronic synapse devices have nonlinear, asymmetric, and finite dynamic range conductance responses, the proposed unidirectional weight-updating method is appropriate to obtain the best learning performance. Furthermore, on-chip learning can mitigate the inevitable device-to-device variation effect of analog-based HW-DNNs. Based on the GSD as a synapse device, neuron circuits compatible with GSDs and on-chip learning rule, we designed a hardware-based neural network system. As a part of

that process, we fabricated a synapse device array. The total number of devices consisting of the GSD array is 200 (10 inputs, 20 outputs). In this array, the variation of the GSD array shows 0.34, 0.22, and 0.14 at three different weight states. By using the GSD array, we efficiently performed the vector-by-matrix multiplication.

The described synapse device, neuron circuits, and on-chip learning rule can be helpful to implement power efficient and low power HW-DNNs.

# Bibliography

[1] C.S. Poon, K. Zhou, "Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities," *Frontiers in neuroscience*, 5:108, 2011.

[2] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: application to complex visual pattern extraction," *IEEE Electron Devices Meeting*, 2011.

[3] C.D. Wright, Y. Liu, K.I. Kohary, M.M. Aziz, R.J. Hicken, "Arithmetic and biologically-inspired computing using phase-change materials," *Advanced Materials*, vol. 23, pp. 3408-3413, 2011.

[4] D. Kuzum, R.G. Jeyasingh, B. Lee, H.S. Wong, "Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing," *Nano letters*, vol. 12, pp. 2179-2186, 2012

[5] S.H. Jo, T. Chang, I. Ebong, B.B. Bhadviya, P. Mazumder, W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, pp. 1297-1301, 2010

[6] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J.K. Gimzewski, M. Aono, "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature Materials*, vol. 10, pp. 591-595, 2011.

[7] Y. Wu, S. Yu, H.S. Wong, "AlOx-based resistive switching device with gradual resistance modulation for neuromorphic device application," *IEEE Memory Workshop*, 2012.

[8] S. Yu, B. Gao, Z. Fang, "A neuromorphic visual system using RRAM synapse

devices with Sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling," *IEEE Electron Devices Meeting*, 2012.

[9] A. Chanthbouala, V. Garcia, R.O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N.D. Mathur, M. Bibes, A. Barthelemy, J. Grollier, "A ferroelectric memristor," *Nature Materials*, vol. 11 pp. 860-864, 2012.

[10] C. Diorio, P. Hasler, A. Minch, C.A. Mead, "A single-transistor silicon synapse," *IEEE Transactions on Electron Devices*, vol. 43, pp. 1972-1980, 1996.

[11] M. Ziegler, M. Oberländer, D. Schroeder,W.H. Krautschneider, H. Kohlstedt, "Memristive operation mode of floating gate transistors: A two-terminal MemFlash-cell," *Applied Physics Letters*, vol. 101, 263504, 2012.

[12] H. Kim, J. Park, M.-W. Kwon, J.-H. Lee, B.-G. Park, "Silicon-based floating-body synaptic transistor with frequency-dependent short- and long-term memories," *IEEE Electron Device Letters*, vol. 37, pp. 249-252, 2016.

[13] G.-q. Bi, M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of Neuroscience*, vol. 18, 10464, 1998.

[14] D. Kuzum, S. Yu, H.S. Wong, "Synaptic electronics: materials, devices and applications," *Nanotechnology*, vol. 24, 382001, 2013.

[15] G.W. Burr, R.M. Shelby, C. di Nolfo, J.-W. Jang, R.S. Shenoy, P. Narayanan, K. Virwani, E.U. Giacometti, B. Kurdi, H. Hwang, "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," *IEEE Electron Devices Meeting*. 2014.

[16] M.V. Nair, P. Dudek, "Gradient-descent-based learning in memristive crossbar arrays," *IEEE International Joint Conference on Neural Networks*, 2015.

[17] M. Prezioso, F. Merrikh-Bayat, B.D. Hoskins, G.C. Adam, K.K. Likharev, D.B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, pp. 61-64, 2015.

[18] F. Merrikh-Bayat, X. Guo, H.A. Om'mani, N. Do, K.K. Likharev, D.B. Strukov, "Redesigning commercial floating-gate memory for analog computing applications," *IEEE International Symposium on Circuits and Systems*, 2015.

[19] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning internal representations by error propagation," *in Parallel Distributed Processing: Explorations in Macrostructure of Cognition, Vol. I Cambridge, MA: Badford*, 1986.

[20] C. Merkel, R. Hasan, N. Soures, D. Kudithipudi, T. Taha, S. Agarwal, M. Marinella, "Neuromemristive systems: Boosting efficiency through brain-inspired computing," *Computer*, vol. 49, pp. 56-64, 2016.

[21] S.K. Gonugondla, M. Kang, N. Shanbhag, "A 42pJ/Decision 3.12TOPS/W Robust In-Memory Machine Learning Classifier with On-Chip Training," *IEEE International Solid-State Circuits Conference*, 2018.

[22] E.O. Neftci, C. Augustine, S. Paul, G. Detorakis, "Event-driven random back-propagation: enabling neuromorphic deep learning machines," *Frontiers in Neuroscience*, 11:324, 2017.

[23] F. Merrikh-Bayat, S. Bagheri Shouraki, I. Esmaili Paeen Afrakoti, "Bottleneck of using a single memristive device as a synapse," *Neurocomputing*, vol. 115, pp. 166-168, 2013.

[24] J.-W. Jang, S. Park, G.W. Burr, H. Hwang, Y.-H. Jeong, "Optimization of conductance change in PrCaMnO based synapse devices for neuromorphic systems," *IEEE Electron Device Letters*, vol. 36, pp. 457-459, 2015.

[25] J. Woo, K. Moon, J. Song, M. Kwak, J. Park, H. Hwang, "Optimized

programming scheme enabling linear potentiation in filamentary HfO2 RRAM synapse for neuromorphic systems," *IEEE Transactions on Electron Devices*, vol. 63, pp. 5064-5067, 2016.

[26] E.J. Fuller, F.E. Gabaly, F. Leonard, S. Agarwal, S.J. Plimpton, R.B. Jacobs-Gedrim, C.D. James, M.J. Marinella, A.A. Talin, "Li-ion synaptic transistor for low power analog computing," *Advanced Materials*, 29, 2017.

[27] J.-H. Bae, S. Lim, B.-G. Park, J.-H. Lee, "High-density and near-linear synapse device based on a reconfigurable gated Schottky diode," *IEEE Electron Device Letters*, vol. 38, pp. 1153-1156, 2017.

[28] T. Gokmen, Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: design considerations," *Frontiers in Neuroscience*, 10:333, 2016.

[29] A. Fumarola, P. Narayanan, L.L, Sanches, S. Sidler, J. Jang, K. Moon, R.M. Shelby, H. Hwang, G.W. Burr, "Accelerating machine learning with Non-Volatile Memory: Exploring device and circuit tradeoffs," *IEEE International Conference on Rebooting Computing*, 2016.

[30] F. Su, W.-H. Chen, L. Xia, C.-P. Lo, T. Tang, Z. Wang, K.-H. Hsu, M. Cheng, J.-Y Li, Y. Xie, Y. Wang, M.-F. Chang, H. Yang, and Y. Liu, "A 462GOPs/J RRAM-Based Nonvolatile Intelligent Processor for Energy Harvesting IoE System Featuring Nonvolatile Logics and Processing-In-Memory," *Symposium on VLSI circuits*, 2017.

[31] J.-H. Bae, S. Lim, J.-H. Lee, "Investigation of current saturation and short channel effect in gated Schottky diode-type synapse device under reverse bias condition," *Silicon Nanoelectronics Workshop*, 2018.

[32] F. R. Libsch and M. H. White, "Charge transport and storage of low programming voltage SONOS MONOS memory devices," *Solid-State Electronics*,

vol. 33, pp. 105-126, 1990.

[33] T. Kim, H. Kim, J. Kim, and J.-J. Kim, "Input voltage mapping optimized for resistive memory-based deep neural network hardware," *IEEE Electron Device Letters*, vol. 38, no. 9, pp. 1228-1231, 2017.

[34] F.Merrikh-Bayat, X. Guo, and D. Strukov, "Exponential-weight multilayer perceptron," *in Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017.

[35] S. Lim, C. Sung, H. Kim, T. Kim, J. Song, J.-J. Kim, and H. Hwang, "Improved synapse device with MLC and conductance linearity using quantized conduction for neuromorphic systems," *IEEE Electron Device Letters*, vol. 39, no. 2, pp. 312-315, 2018.

[36] S. Lim, J.-H Bae, J.-H Eum, S. Lee, C.-H. Kim, D. Kwon, and J.-H. Lee, "Hardware-based neural networks using a gated Schottky diode as a synapse device," *in Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018.

[37] P. Wijesinghe, A. Ankit, A. Sengupta, and K. Roy, "An all-memristor deep spiking neural computing system: A step toward realizing the low-power stochastic brain," *IEEE Transactions on Emerging Topics on Computational Intelligence*, vol. 2, no. 5, pp. 345–358, 2018.

[38] R. Hasan, T. M. Taha, and C. Yakopcic, "On-chip training of memristor based deep neural networks," *in Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017.

[39] B. Liu, H. Li, Y. Chen, X. Li, T. Huang, Q. Wu, and M. Barnell, "Reduction and IR-drop compensations techniques for reliable neuromorphic computing systems," *in Proc. IEEE/ACM International Confernece on Computer-Aided Design (ICCAD)*, 2014.

[40] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, pp. 0705201, 2012.

[41] 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 synapse devices," *IEEE Transactions on Electron Devices*, vol. 66, no. 1, pp. 395-401, 2018.

[42] W. Schiffmann, M. Joost, R. Werner, "Optimization of the backpropagation algorithm for training multilayer perceptrons," *Technical report, University of Koblenz, Institute of Physics*, Rheinau, 1994.

[43] O. Bichler, M. Suri, D. Querlioz, D. Vuillaume, B. DeSalvo, C. Gamrat, "Visual pattern extraction using energy-efficient "2-PCM Synapse" neuromorphic architecture," *IEEE Transactions on Electron Devices*, vol. 59, pp. 2206-2214, 2012.

[44] D. Querlioz, O. Bichler, P. Dollfus, C. Gamrat, "Immunity to device variations in a spiking neural network with memristive nanodevices," *IEEE Transactions on Nanotechnology*, vol. 12, pp. 288-295, 2013.

[45] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems*, pp. 153-160, 2006.

[46] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.

[47] J. Liang, H.S. Philip Wong, "Cross-Point Memory Array Without Cell Selectors-Device Characteristics and Data Storage Pattern Dependencies," *IEEE Transactions on Electron Devices*, vol. 57, no. 10, pp. 2531-2537, 2010.

# 초 록

신경망 기술을 기반으로 한 AI (인공 지능)는 인간의 인지 능력을 능가하는 모습을 보여줄 수 있기 때문에 다양한 산업 분야에서 널리 연구되고 있다. 그러나, 기존의 컴퓨팅 아키텍처인 폰 노이만 (von Neumann) 아키텍처의 메모리와 컴퓨팅 유닛 사이의 병목 문제 및 매우 큰 전력 소모 문제 때문에, 뉴로모픽 컴퓨팅의 필요성이 대두되고 있다. 특히, 시냅스로 동작하는 전자 시냅스 소자는 저전력 및 고속으로 벡터 행렬 곱셈 (VMM)을 수행 할 수 있기 때문에 전력 소비의 상당 부분을 줄일 수 있다. 이 논문에서는 시냅스 소자로서 게이티드 쇼트키 다이오드 (GSD)를 제안한다. 역 쇼트키 다이오드로 동작하므로 시냅스 전류가 낮고, 입력 전압에 대해 포화 상태로 동작한다. 또한, 시냅스 소자에 적용하는 펄스 수에 대한 컨덕턴스 변화가 선형적인 특성을 보여준다. 이러한 특성들을 고려하여 전류 합산, 펄스 폭 변조 및 활성화 기능을 수행하는 뉴런 회로를 설계한다. 설계한 뉴런 회로를 검증하기 위해 SPICE 시뮬레이션을 통해 2 계층 신경망의 추론 정확도를 평가한다. 검증 결과로서 MNIST 테스트 세트의 100 개 이미지의 분류 정확도는 94%이며, 이는 소프트웨어를 통해 얻은 기준 정확도와 비슷하다. 또한, 우리는 전자 시냅스 소자를 기반으로 하는 신경망의 온칩 학습 방법을 제안한다. 시냅스 소자의 비이상성 때문에, 맨해튼 업데이트 규칙을 기반으로 하는 가중치 갱신 방법을 제안하고, 시냅스 장치의 다양한 비이상성에 대해 온칩 학습 방법을 평가한다. 시냅스 디바이스가 선형 컨덕턴스 응답

과 높은 동적 범위를 가지면 소프트웨어로 얻은 기준 정확도와 비슷한 정확도를 얻을 수 있다. 오프칩 트레이닝 체계는 디바이스 변형에 취약한 반면, 앞서 설명한 온칩 트레이닝 방법은 이러한 변동 효과를 완화할 수 있기 때문에 하드웨어 기반의 신경망 시스템에 필수적이라고 할 수 있다. 더 나아가, 신경망 시스템을 구현하기 위해 GSD를 기반으로 시냅스 장치 어레이를 제작한다. 시냅스 소자의 포화된 전류 특성으로 인해, 금속 와이어에서의 전압 강하 문제없이 VMM 연산을 잘 수행한다. 또한, 제작된 시냅스 소자 어레이는 세 가지 시냅스 가중치 상태에 대해 0.34, 0.22 및 0.14의 변화를 보여준다.

우리가 제안한 게이티드 쇼트키 다이오드, 이와 호환 가능한 뉴런 회로 및 온칩 학습 방법은 하드웨어 기반 신경망을 구현하는 데 도움이 될 것이다.

주요어 : 인공지능, 시냅스 소자, 게이티드 쇼트키 다이오드, 전류 포화, 선형적인 전도도 응답 특성, 뉴런 회로, 온칩 학습 방법.

학번 : 2015-30202

# List of Publications

## Journals

1. *Jong-Ho Bae, ***Suhwan Lim**, Byung-Gook Park, and Jong-Ho Lee, "High-Density and Near-Linear Synapse device Based on a Reconfigurable Gated Schottky Diode," *IEEE Electron Device Letters*, 2017.
2. Chul-Heung Kim, Soochang Lee, Sung Yun Woo, Won-Mook Kang, **Suhwan Lim**, Jong-Ho Bae, Jaeha Kim, and Jong-Ho Lee, "Demonstration of Unsupervised Learning With Spike-Timing-Dependent Plasticity Using a TFT-Type NOR Flash Memory Array," *IEEE Transactions on Electron Devices*, 2018.
3. **Suhwan Lim**, Jong-Ho Bae, Jai-Ho Eum, Sungtae Lee, Chul-Heung Kim, Dongseok Kwon, Byung-Gook Park, and Jong-Ho Lee, "Adaptive learning rule for hardware-based deep neural networks using electronic synapse devices," *Neural Computing and Applications*, 2018
4. Kyu-Bong Choi, Sung Yun Woo, Won-Mook Kang, Soochang Lee, Chul-Heung Kim, Jong-Ho Bae, **Suhwan Lim**, and Jong-Ho Lee, "A Split-Gate Positive Feedback Device With an Integrate-and-Fire Capability for a High-Density Low-Power Neuron Circuit," *Frontiers in Neuroscience*, 2018.
5. *Chul-Heung Kim, ***Suhwan Lim**, Sung Yun Woo, Won-Mook Kang, Young-Tak Seo, Sung Tae Lee, Soochang Lee, Dongseok Kwon, Seongbin Oh, Yoohyun Noh, Hyeongsu Kim, Jangsaeng Kim, Jong-Ho Bae, and Jong-Ho Lee, "Emerging memory technologies for neuromorphic computing," *Nanotechnology*, 2018
6. Dongseok Kwon, **Suhwan Lim**, Jong-Ho Bae, Sung-Tae Lee, Hyeongsu Kim, Chul-Heung Kim, Byung-Gook Park, and Jong-Ho Lee, "Adaptive weight quantization method for nonlinear synapse devices," *IEEE*

*Transactions on Electron Devices*, 2018.

7 Jong-Ho Bae, **Suhwan Lim**, Dongseok Kwon, Jai-Ho Eum, Sung-Tae Lee, Hyeongsu Kim, Byung-Gook Park, and Jong-Ho Lee, "Near-Linear Potentiation Mechanism of Gated Schottky Diode as a Synapse device," *IEEE Journal of the Electron Devices Society*, 2019.

8 Jong-Ho Bae, Hyeongsu Kim, Dongseok Kwon, **Suhwan Lim**, Byung-Gook Park, and Jong-Ho Lee, "Reconfigurable Field-Effect Transistor as a Synapse device for XNOR Binary Neural Network," *IEEE Electron Devices Letters*, 2019.

9 Sung Yun Woo, Kyu-Bong Choi, **Suhwan Lim**, Sung-Tae Lee, Chul-Heung Kim, Won-Mook Kang, Dongseok Kwon, Jong-Ho Bae, Byung-Gook Park, and Jong-Ho Lee, "Synapse device Using a Floating Fin-Body MOSFET With Memory Functionality for Neural Network," *Solid-State Electronics*, 2019.

10 Jong-Ho Bae, **Suhwan Lim**, Dongseok Kwon, Sung-Tae Lee, Hyeongsu Kim, and Jong-Ho Lee, "Gated Schottky Diode-Type Synapse device with a Field-Plate Structure to Reduce the Forward Current," *Journal of Nanoscience and Nanotechnology*, 2019.

11 ***Suhwan Lim,** *Dongseok Kwon, Jai-Ho Eum, Sungtae Lee, Jong-Ho Bae, Hyeongsu Kim, Chul-Heung Kim, Byung-Gook Park, Jong-Ho Lee, "Highly Reliable Inference System of Neural Networks Using Gated Schottky Diodes," *IEEE Journal of the Electron Devices Society*, 2019

## Conferences

1. **Suhwan Lim**, Jong-Ho Bae, Jai-Ho Eum, Sungtae Lee, Chul-Heung Kim, Dongseok Kwon, and Jong-Ho Lee, "Hardware-based Neural Networks using a Gated Schottky Diode as a Synapse Device," *IEEE International Symposium on Circuits and Systems (ISCAS)*, May. 2018.

2. Jong-Ho Bae, **Suhwan Lim**, Dongseok Kwon, Sungtae Lee, Byung-Gook Park, and Jong-Ho Lee, “Investigation of Current Saturation and Short Channel Effect in Gated Schottky Diode-type Synapse device under Reverse Bias Condition,” *IEEE Silicon Nanoelectronics Workshop (SNW)*, June. 2018.

3. Sung-Tae Lee, **Suhwan Lim**, Nagyong Choi, Jong-Ho Bae, Chul-Heung Kim, Soochang Lee, Dong Hwan Lee, Tackhwi Lee, Sungyong Chung, Byung-Gook Park, and Jong-Ho Lee, “Neuromorphic Technology Based on Charge Storage Memory Devices,” *Symposia on VLSI Technology and Circuits*, June. 2018.

4. **Suhwan Lim**, Jong-Ho Bae, and Jong-Ho Lee, “Input voltage modulation for hardware-based neural networks using NOR-type flash memory cells,” *Asia-Pacific Workshop on Fundamentals and Applications of Advanced Semiconductor Devices (AWAD)*, July. 2018.

5. *Suhwan Lim, *Dongseok Kwon, Sung-Tae Lee, Hyeongsu Kim, Jong-Ho Bae, and Jong-Ho Lee, “Investigation of Neural Networks Using Synapse Arrays Based on Gated Schottky Diode,” *International Joint Conference on Neural Networks (IJCNN)*, July. 2019.

## Honors

1. Silver Prize, The 23rd Humantech Thesis contest, Samsung Electronics, Feb. 2017.