JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.18, NO.5, OCTOBER, 2018
https://doi.org/10.5573/JSTS.2018.18.5.568
ISSN(Print) 1598-1657
ISSN(Online) 2233-4866

Memristor Binarized Neural Networks

Khoa Van Pham[1], Tien Van Nguyen[1], Son Bao Tran[1], HyunKyung Nam[1], Mi Jung Lee[2], Byung Joon Choi[3], Son Ngoc Truong[4], and Kyeong-Sik Min[1,*]

Abstract—**Binarized Neural Networks (BNNs) use only binary synapses of +1 and -1, not allowing any intermediate weights between -1 and +1. Though the recognition rate of BNNs is lower than the conventional Deep Neural Networks (DNNs), BNNs have attracted many interests nowadays, because BNNs do not need the complicated multiplication such as DNNs. Binary memristor crossbars can be very suitable to realize BNN hardware. This is because, in memristor BNNs, simple binary operation can be performed in bitwise manner for all the columns in memristor crossbars, simultaneously. In this paper, single-column and double-column memristor BNNs are presented, respectively. In addition, ReLU and sigmoid activation function circuits are also proposed with CMOS circuits. The designed Memristor-CMOS hybrid circuits of BNNs have been tested for MNIST vectors. The memristor BNNs could recognize almost 90% MNIST digits when the memristance variation is as large as 25%. For variation tolerance, the memristor BNNs are compared with the multi-valued memristor neural networks such as 4-bit, 6-bit, etc, in this paper. As a result, it has been confirmed the memristor BNNs become more variation-tolerant than the multi-valued memristor NNs when the variation becomes larger than 22%. Comparing the single-column and double-column BNNs in this paper indicates that the single-column BNN can save power consumption and array area almost by half than the double-column. This is because the single-column has just half memristors than the double-column. And, we measured the single-column and double-column BNNs using the fabricated memristor array. In this measurement, both the double-column and single-column BNNs were observed to work well.**

Index Terms—Memristor binarized neural networks, memristor crossbars, Memristor-CMOS hybrid circuits

I. INTRODUCTION

Deep Neural Networks (DNNs) are very useful in many human-like applications such as image and speech recognition, etc. Usually, DNNs use high-precision numbers such as 24-bit, 32-bit, or even 64-bit, in performing the complicated multiplication to update synaptic weights [1]. The high-precision multiplication in DNNs is the main reason why DNNs demand the very complicated multiply-accumulate (MAC) circuits with consuming large amounts of energy [2]. To mitigate the burden of high-cost multiplication, low-precision DNNs such as Quantization Neural Networks (QNNs), Binarized Neural Network (BNNs), etc. have been studied recently [2, 3]. Among them, especially, BNNs that are the simplest version of neural networks (NNs) employ only binary synaptic weights not calculating any high-precision numbers. The binarized operation can be performed simply by bitwise logic, not using any

Manuscript received Mar. 6, 2018; accepted Oct. 11, 2018
[1] School of Electrical Engineering, Kookmin University, Seoul, Korea
[2] School of Advanced Materials Engineering, Kookmin University, Seoul, Korea
[3] Department of Materials Science and Engineering, Seoul National University of Science and Technology (Seoultech), Seoul, Korea
[4] School of Electrical Engineering, Kookmin University, Seoul, Korea. He is now with the Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, Vietnam
E-mail : mks@kookmin.ac.kr

complicated MAC circuits [3]. Thus, the energy and time of bitwise operation of binary synapses can be greatly reduced in BNNs, compared to the other high-precision DNNs demanding MAC operation [2-4]. In spite of these advantages due to the simple bitwise logic, the recognition rate loss of BNNs is small enough to make BNNs useful in many machine-learning applications demanding competitive performance as well as low-power consumption.

I. MEMRISTOR BINARIZED NEURAL NETWORKS

Fig. 1(a) shows a conceptual diagram of BNNs, where x_0, x_1, x_2, etc. are the input neurons that apply the input vectors such as the MNIST testing vector '5' to the BNNs. Here MNIST stands for Modified National Institute of Standards and Technology. y_0, y_1, y_2, etc. represent the hidden neurons and z_0, z_1, z_2, etc. are the output neurons. In Fig. 1(a), $w_{0,0,0}$ is the synaptic weight between the x_0 neuron and the y_0 neuron for layer #0. Similarly, $w_{0,0,1}$ represents the synaptic connection of y_0 and z_0 for layer #1. In BNNs, all the synaptic weights should be either +1 or -1 not allowing any intermediate values between -1 and +1 [3].

In realizing BNN hardware, memristor crossbars can be used to represent binary synaptic weights of BNNs, as shown in Fig. 1(b). Here memristors can be fabricated on two metal lines crossing each other. To represent binary numbers, memristors can be programmed High Resistance State (HRS) or Low Resistance State (LRS) by applying electrical voltages or current pulses. As illustrated in Fig. 1(b), x_0, x_1, x_2, etc. are the input neurons which deliver the input vector to the memristor crossbar, where the binary synaptic weights are stored on the cross-points. y_0, y_1, y_2, etc. represent the hidden neurons, as already shown in Fig. 1(a). Here y_0 is the result calculated with the bitwise operation between the input neurons and the corresponding synaptic connections. The operation can be simply expressed with $y_o = \sum_{i=0}^{n} x_i \cdot w_{i,0}$. Here x_i and $w_{i,0}$ mean the ith input neuron and the synaptic connection between x_i and y_0, respectively. Similarly, y_1, y_2, etc. can perform the same bitwise operations simultaneously, for the input neurons from x_0 to x_n. As you see in Fig. 1(b), y_0, y_1, y_2, etc. can perform the bitwise operations in parallel, at the same time. These parallel bitwise operations can make memristor crossbars very suitable for realizing BNNs in hardware.

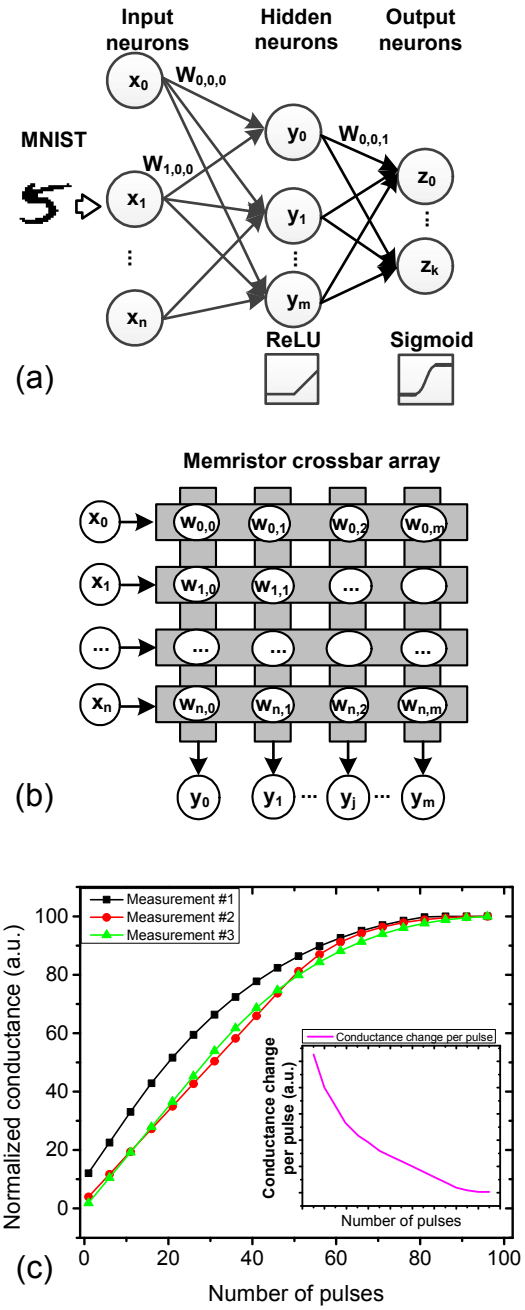

Fig. 1. (a) The conceptual diagram of BNNs with binary synaptic weights of +1 and -1, (b) The memristor crossbar for realizing BNN hardware. Here the binary memristors can represent the binary synaptic weights of BNNs, (c) The normalized memristor's conductance with increasing the number of programming pulses and the conductance change per pulse with increasing the number of pulses. Here the measurement#1, #2, and #3 are from the following references [7-9], respectively.

Memristor's programmable behaviors have been studied intensively for many years [5-15]. This is mainly due to the fact that memristors can possibly mimic biological synaptic functions well. Moreover, in terms of fabrication technology, memristor crossbars can be implemented with CMOS-compatible Back-End-Of-Line (BEOL) process [16, 17]. By doing so, memristors can be combined with CMOS circuits to realize various Memristor-CMOS hybrid systems such as memristor BNNs. Moreover, memristor crossbars can be stacked layer by layer to form a 3-dimensional architecture that seems very similar to human brain's neuronal structure [16, 17].

Finally, it should be noted that we focus on only binary memristors not multi-valued memristors in this paper, because we use memristor crossbars for realizing BNNs. In memristor BNNs, we program memristors with only LRS and HRS not allowing any intermediate levels between LRS and HRS. Of course, programming memristors with multi-values can be employed to mimic analog synapses [9]. But, most of the filamentary-switching memristors show very sharp and abrupt transition between LRS and HRS states [11, 12, 14]. This sharp and abrupt transition of memristor's conductance makes the precise tuning of multi-valued synapses very difficult [12, 14]. Compared to the multi-valued programming, the binary programming of memristors needs only LRS and HRS. As shown in Fig. 1(c), memristor's conductance becomes saturated around LRS with applying the voltage pulses [7-9, 12]. Unlike the transition region, no abrupt change of memristor's conductance is observed around LRS. The inset figure in Fig. 1(c) shows memristor's conductance change per pulse with respect to the number of pulses [12]. The transition region shows very abrupt and sharp change of memristor's conductance, as shown in the inset of Fig. 1(c). However, the conductance change becomes smaller and saturated as memristor's conductance becomes closer to LRS. From the main and inset figures of Fig. 1(c), we can know that binary programming of memristors is more controllable than the multi-valued programming.

II. MEMRISTOR CROSSBARS AND ACTIVATION FUNCTION CIRCUITS

Fig. 2(a) shows a memristor BNN with double-column crossbar [13] to calculate binary synaptic connections of +1 and -1. Here x_0, x_1, x_2, etc. represent the input neurons that deliver voltage pulses to the crossbar. The input neurons of x_0, x_1, x_2, etc. are connected with the neurons of y_0, y_1, y_2, etc. through memristive synapses. $g_{0,0+}$ and $g_{0,0-}$ in Fig. 2(a), represent the positive and negative synaptic connections, for (+) and (-) columns, respectively. Here the (+) and (-) columns for the y_0 neuron mean the positive and negative synaptic weights, respectively. The symbols denoted as f mean the activation function circuits, where the column current calculated from the positive and negative columns is converted to a voltage according to the activation

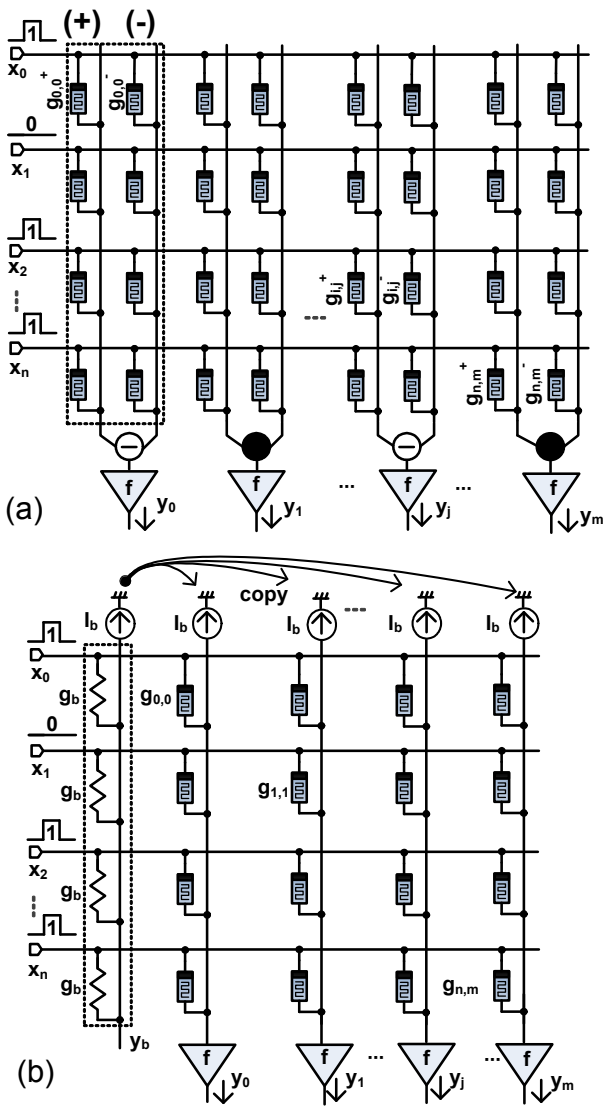

Fig. 2. (a) The double-column memristor crossbar for implementing memristor BNNs [13], (b) The single-column memristor crossbar for implementing memristor BNNs.

function f().The jth neuron, y_j, can be formulated mathematically with the following equation.

$$y_j = f\left(\sum_{i=0}^{n}\left(x_i \cdot g_{i,j+} - x_i \cdot g_{i,j-}\right)\right) = f\left(\sum_{i=0}^{n} x_i \cdot \left(g_{i,j+} - g_{i,j-}\right)\right) \quad (1)$$

Here, it should be noted that conductance variables, $g_{i,j+}$ and $g_{i,j-}$, can be either g_{LRS} or g_{HRS}. g_{LRS} and g_{HRS} are 1/LRS and 1/HRS, respectively. This is the same with that the synaptic weights of BNNs are either +1 or -1, not allowing any intermediate values between -1 and +1. Assuming $g_{LRS} >> g_{HRS}$, g_{HRS} can be ignored in the Eq. (1). If so, $\left(g_{i,j+} - g_{i,j-}\right)$ can be approximated by $+g_{LRS}$ or $-g_{LRS}$ that can be interpreted as +1 or -1 in the memristor BNNs.

Fig. 2(b) shows an alternative approach to the memristor-based BNNs, where a single-column is utilized for calculating both binary synaptic connections of +1 and -1. The single-column scheme in Fig. 2(b) is more efficient than the double-column scheme depicted in Fig. 2(a), where two columns of (+) and (-) are used to calculate positive and negative synaptic connections, respectively. Here x_0, x_1, x_2, etc. represent the input neurons which are connected to the neurons of y_0, y_1, y_2, etc. by the synaptic connections. Here $g_{0,0}$ means the binary synaptic connection between x_0 and y_0. To calculate both +1 and -1 values using the single-memristor column, in Fig. 2(b), we exploited g_b to generate the current I_b [10].

Let's look at I_b in Fig. 2(b), in detail. In the first column boxed by dashed-line, the g_b column applied by the input pulses generates I_b current as much as $I_b = \sum_{i=0}^{n} x_i \cdot g_b$. After that, I_b is copied to each column, where it can be subtracted from all crossbar columns, simultaneously, as shown in Fig. 2(b) [10]. The f symbol means the activation function circuit, where each column current is converted to a voltage, according to the activation function f(). Though the I_b generator and subtractor were already used in [10], we introduced the activation function circuits that were not used in [10], in this paper. Furthermore, the I_b generator and I_b subtractor were newly designed with CMOS circuits in this paper, not using any passive resistors which were used previously [10]. The detailed schematics of I_b generator, I_b subtractor, and activation function circuits are shown in Fig. 3(a)-(d).

Combining the I_b subtraction and f() activation function, we can express y_j with the following equation.

$$y_j = f\left(\sum_{i=0}^{n}\left(x_i \cdot g_{i,j} - x_i \cdot g_b\right)\right) = f\left(\sum_{i=0}^{n} x_i \cdot \left(g_{i,j} - g_b\right)\right) \quad (2)$$

From the Eq. (2), if g_b is given by $(g_{LRS}+g_{HRS})/2$, the synaptic connection of $g_{i,j}$-g_b can be $+(g_{LRS}-g_{HRS})/2$ or $-(g_{LRS}-g_{HRS})/2$. Here $g_{i,j}$ should be either g_{LRS} or g_{HRS}. These two values of $+(g_{LRS}-g_{HRS})/2$ and $-(g_{LRS}-g_{HRS})/2$ can be interpreted as the binary synapses like Fig. 2(a). The single-column scheme of memristor-based BNNs in Fig. 2(b) has two obvious advantages than the double-column scheme in Fig. 2(a). First, the memristor programming time can be reduced by half, because the number of crossbar columns in Fig. 2(b) is half of the number of columns in Fig. 2(a). And, also, the area of memristor array can be half compared to the double-column crossbar, due to the same reason for the first advantage.

Fig. 3(a) shows the detailed schematic of the single-column crossbar in Fig. 2(b). Here, g_b is implemented simply with a diode-connected NMOSFET, M_b. We can adjust the channel conductance of M_b in Fig. 3(a) to be equal to g_b in Fig. 2(b) by adjusting the NMOSFET size. The input pulses from x_0, x_1, x_2, etc. are applied to the corresponding g_b's to generate I_b. The I_b current is copied to each column and subtracted from all memristor columns, as shown in Fig. 3(a). $g_{0,0}$ is the binary synaptic connection between x_0 and y_0. $g_{1,1}$ represents the binary connection between x_1 and y_1. OP_0 is an op-amp for the I_b generator and subtractor. M_1 acts as a resistor for subtracting the I_b current from all memristor columns. V_C means a bias voltage to control the channel resistance of M_1.

The symbol denoted as f means a current-to-voltage converter, where the column current I_j is converted to the y_j voltage according to the activation function f(). As illustrated in Fig. 1(a), the activation functions used in this paper are Rectified Linear Unit (ReLU) and Sigmoid.

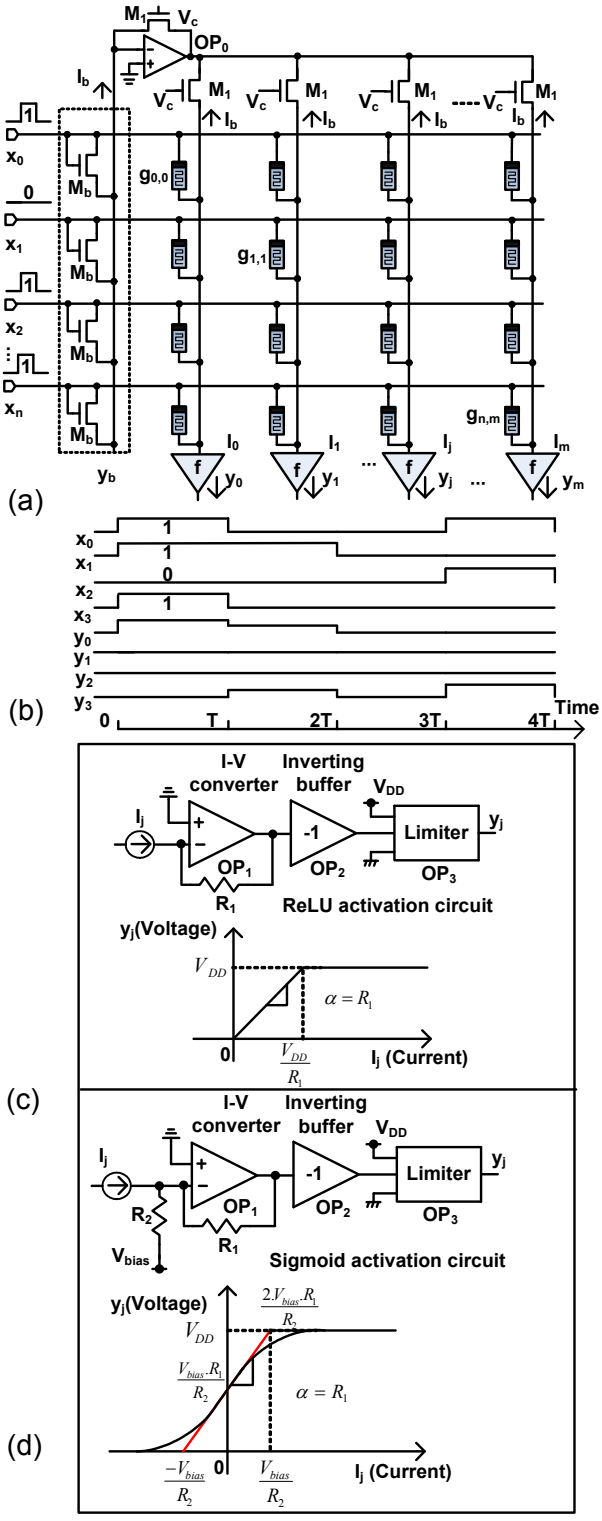

Fig. 3. (a) The detailed schematic of the single-column scheme, (b) The operational waveforms for the single-column scheme, (c) The ReLU activation circuit, (d) The Sigmoid activation circuit.

Fig. 3(b) shows the operational waveforms extracted from Fig. 3(a). For the first cycle, x_0, x_1, x_2, x_3 deliver the input pulses of 1, 1, 0, 1, respectively, to the crossbar. In this case, if the y_0 column stores LRS, LRS, HRS, and LRS, respectively, the I_0 current generated from the multiplication of the input pulses with the stored conductance values can be the largest among all the columns. This current enters the f circuit, where it is converted to the y_0 voltage according to the activation function. Assuming that the activation function is ReLU, the converted y_0 voltage is shown in Fig. 3(b). The y_0 voltage can be changed every cycle, according to the input pulses, as shown in Fig. 3(b)-(d) show the ReLU and Sigmoid activation function circuits, respectively. The ReLU is used for the hidden neurons and the Sigmoid for the output neurons. In the ReLU circuit, OP_1 converts the I_j current to the voltage of $-I_j \times R_1$. OP_2 is a simple inverting buffer, where $-I_j \times R_1$ is just inverted to $+I_j \times R_1$. OP_3 acts as a limiter. If the output y_j voltage is higher than V_{DD} or lower than GND, the output voltage is limited by V_{DD} or GND, respectively. The transfer curve of the ReLU circuit is also shown in Fig. 3(c). The Sigmoid circuit is shown in Fig. 3(d). Here R_2 and V_{bias} are used in the Sigmoid circuit. By doing so, the Sigmoid's transfer curve can be obtained by shifting the ReLU's transfer curve by $-V_{bias}/R_2$. In the Sigmoid's transfer curve in Fig. 3(d), the black line represents the mathematical Sigmoid function and the red line indicates an approximation by the Sigmoid function circuit.

III. Simulation and Experimental Results

By using the MNIST dataset, the memristor BNNs were tested to estimate the network performance. The MNIST recognition task is composed of 60,000 gray-scale handwritten images for the training and 10,000 samples for the execution. Each MNIST vector has 28x28 gray-scale pixels which belong to one of 10 digits from 0 to 9, as shown in Fig. 4(a). The multi-layer neural network shown in Fig. 1(a), was designed to have 784 input neurons corresponding to a 28x28 pixel image and 1024 hidden neurons for the MNIST simulation. After training the designed network on the original images without any data augmentation and pre-processing method, we utilized the direct-weights download method

for delivering the trained weights to the memristor crossbar. In order to investigate the impact of memristance variation on network performance, Monte-Carlo simulation was used in Matlab, in which the percentage of variation was varied from 0% to 30%. We assumed that HRS and LRS are 1 MΩ and 10 KΩ, respectively. In Fig. 4(a), first, we tested the analog neural network with full-precision numbers. The accuracy of this network, considered as the baseline reference, was estimated as high as 98.3%, as shown in Fig. 4(a). It should be noted that no memristance variation is considered for the baseline network. And, we tested 4-bit memristor and 6-bit memristor NNs, respectively, from memristance variation=0% to 30%. When the variation=0%, 4-bit and 6-bit NNs show the recognition rate as high as 97.3% and 97.7%, respectively. However, for the variation=30%, the recognition rates of 4-bit and 6-bit NNs are degraded very much as low as 42.3% and 39.7%, respectively. The binary memristor-based NNs show better performance than 4-bit and 6-bit NNs, for the variation=30%. As, expected, the binary NNs seem more robust to the variation than 4-bit and 6-bit NNs. This is because the binary values are more difficult to be flipped from 0 to 1 or vice versa than the multi-valued NNs. In other words, to flip binary numbers from 0 to 1 or vice versa, we need to apply larger variation to them than the variation to the multi-valued numbers. Table 1 compares the network accuracy for 4-bit NN, 6-bit NN, BNN with single-column, and BNN with double-column. Though 4-bit and 6-bit NNs are better in recognition rate than the memristor BNNs for the variation=0%, the BNNs become better than the multi-valued NNs, with increasing the memristance variation to 30%. Thus, the memristor BNNs can be thought more variation-tolerant than the multi-valued memristor NNs such as 4-bit, 6-bit, etc, as indicated in Table 1.

Comparing the single-column and double-column, the single-column crossbar presents a slight improvement in recognition, compared to the double-column scheme. This is because the number of memristors in the single-column scheme is just half than the double-column scheme. The smaller number of memristors can improve the recognition rate slightly better than the larger number of memristors, under the same condition of memristance variation.

The more obvious advantages of the single-column scheme can be found in power consumption and area. It is clear that the single-column scheme can reduce the power consumption by half, because it has only half memristors, compared to the double-column. As expected, Fig. 4(b) shows that the single-column scheme consumes only half power than the double-column, with varying the LRS from 10 KΩ to 50 KΩ. The power simulation was performed by CADENCE SPECTRE [18]. The CMOS model parameters were obtained from

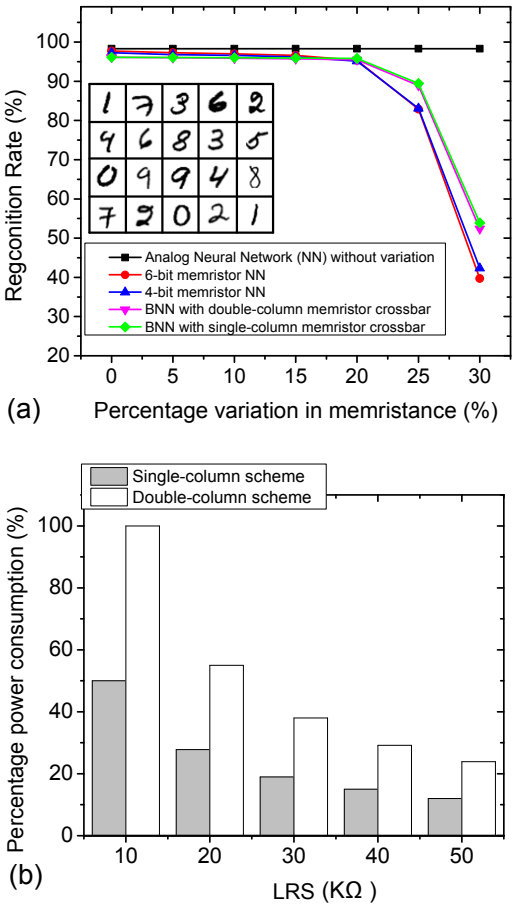

Fig. 4. (a) MNIST recognition rate with varying the percentage variation in memristance from 0% to 30%, (b) Percentage power consumption with varying LRS from 10 KΩ to 50 KΩ (HRS/LRS =100).

Table 1. MNIST recognition rate comparison of 6-bit memristor NN, 4-bit memristor NN, single-column BNN, and double-column BNN. Among 4 configurations, the single-column BNN shows the best performance for the memristance variation is as high as 30%

	6-bit NN	4-bit NN	Single-col BNN	Double-col BNN
Variation=0%	97.7%	97.3%	96.1%	96.1%
Variation=30%	39.7%	42.3%	53.9%	52.4%

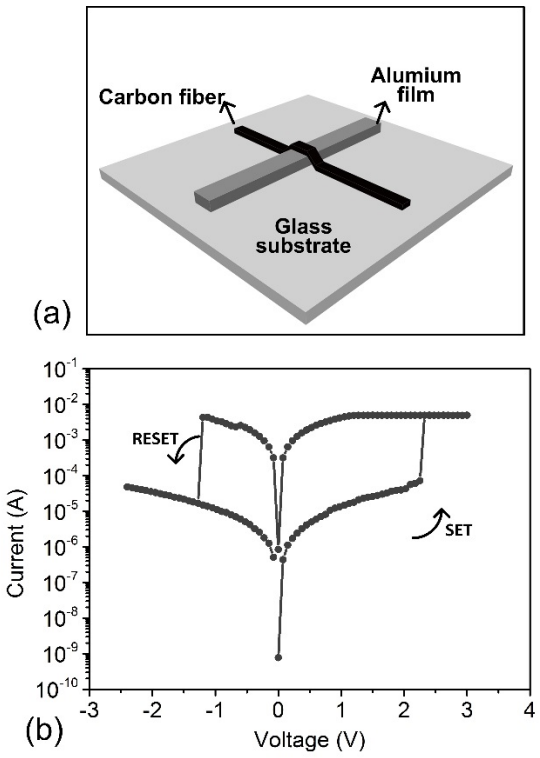

Fig. 5. (a) The structure of measured memristors are formed by carbon fiber and aluminum film [19, 20], (b) The current-voltage relationship of the measured memristors.

SAMSUNG 0.13-μm CMOS technology. The V_{DD} used in SPECTRE simulation is 1.3 V and HRS/LRS ratio of memristors is 100.

In addition to the recognition rate simulation, the measurement results of the double-column and single-column crossbars are also shown in this paper. In the experiment, the measured memristors were fabricated with the carbon fiber on the aluminum film, [19, 20]. Fig. 5(a) shows the structure of measured memristors [19, 20]. Fig. 5(b) shows the current-voltage relationship of the measured memristors. Here, the applied voltage was swept from -2.5 V to 3 V and vice versa. The measured butterfly curve in Fig. 5(b) shows that SET and RESET voltages are around 2.3 V and -1.4 V, respectively. For HRS/LRS ratio, HRS was observed at least 100 times larger than LRS of the measured memristors.

The measurement set-up for memristor array is depicted in Fig. 6(a). Here, Keithley 4200-SCS (Semiconductor Characterization System) is connected with Keithley 708B (Switching Matrix). By doing so, the Source-Measure Units (SMUs) in Keithley 4200-SCS can measure the fabricated memristor array cell by cell.

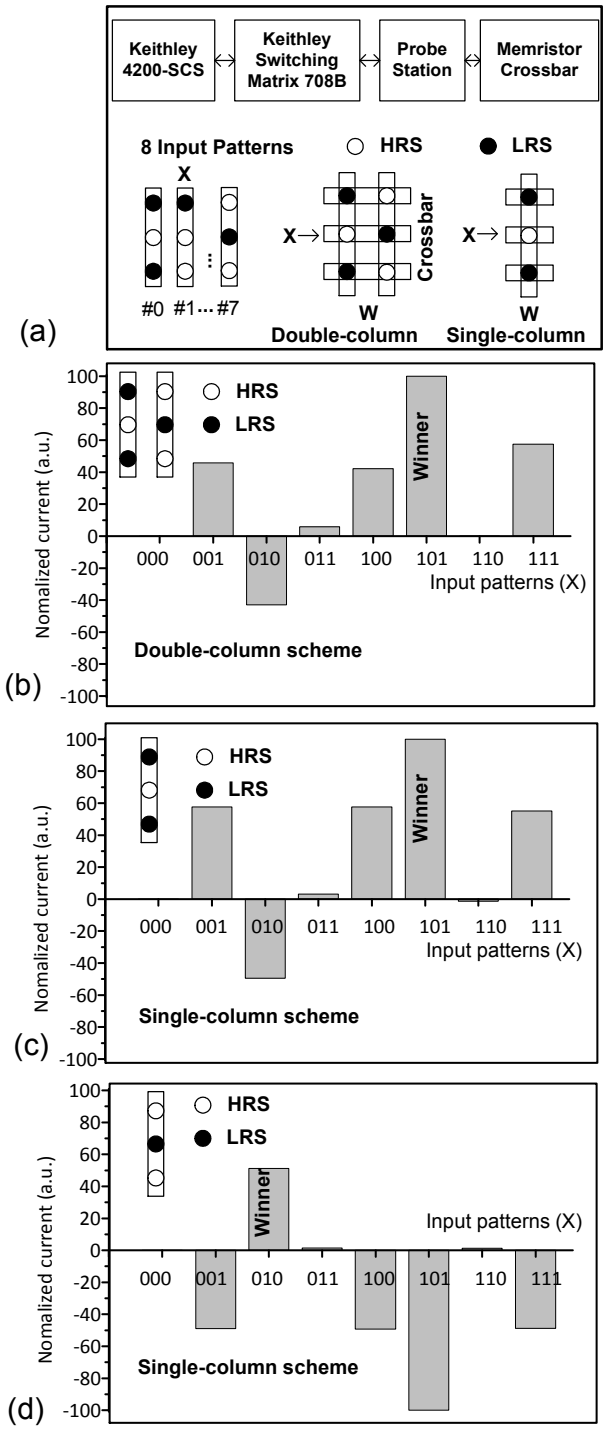

Fig. 6. (a) The measurement set-up in upper. The input patterns (X) and the stored patterns (W) for the double-column and single-column crossbars in lower. HRS and LRS are represented with the open circle and the solid circle, respectively, (b) The measured currents for the double-column scheme programmed by [LRS, HRS, LRS] and [HRS, LRS, HRS], (c) The measured currents for the single-column scheme programmed by [LRS, HRS, LRS], (d) The measured currents for the single-column scheme programmed by [HRS, LRS, HRS].

In the measurement, we tested both the double-column and single-column schemes. For the double-column scheme, we stored the complementary patterns such as [LRS-HRS-LRS] and [HRS-LRS-HRS] on two columns. After that, we applied 8 input patterns to the double-column scheme one by one, as shown in Fig. 6(b). The 8 input patterns are from [000] to [111]. Among 8 input patterns, we could find that the input pattern [101] is the winner among the 8 patterns because the input pattern [101] can maximize the column current, for the double-column with [LRS-HRS-LRS] and [HRS-LRS-HRS].

In Fig. 6(c), we measured the single-column scheme with [LRS-HRS-LRS]. For this case, the input pattern [101] shows the largest current among the others. In Fig. 6(d), we measured the single-column scheme with [HRS-LRS-HRS]. Here, we could know that the input pattern [010] matches the stored [HRS-LRS-HRS]. From the measurement, we could verify that the single-column scheme can work well like the double-column scheme. There is no different in the operation between the double-column and single-column schemes. However, the power consumption and memristor array's area can be reduced by half, if we use the single-column scheme for realizing the memristor-based BNNs.

IV. CONCLUSION

Binarized Neural Networks (BNNs) use only binary synapses of +1 and -1, not allowing any intermediate weights between -1 and +1. Though the recognition rate of BNNs is lower than the conventional Deep Neural Networks (DNNs), BNNs have attracted many interests nowadays, because BNNs do not need the complicated multiplication such as DNNs. Binary memristor crossbars can be very suitable to realize BNN hardware. This is because, in memristor BNNs, simple binary operation can be performed in bitwise manner for all the columns in memristor crossbars, simultaneously.

In this paper, we designed the single-column and double-column memristor BNNs, respectively. In addition, ReLU and sigmoid activation function circuits were also designed with CMOS circuits. The designed Memristor-CMOS hybrid circuits of BNNs were tested for MNIST test vectors. The memristor BNNs were verified to recognize almost 90% MNIST digits when the memristance variation is as large as 25%.

In terms of variation tolerance, the memristor BNNs were compared with the multi-valued memristor neural networks such as 4-bit, 6-bit, etc. As a result, we could confirm the memristor BNNs become more variation-tolerant than the multi-valued memristor neural networks if the variation becomes larger than 22%.

Comparing the single-column and double-column BNNs indicated that the single-column BNN could save the power consumption and array area almost by half than the double-column. This is because the single-column has just half memristors than the double-column. And, we measured the single-column and double-column BNNs using the fabricated memristor array. In this measurement, both the double-column and single-column BNNs were observed to work well.

ACKNOWLEDGMENT

The work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1701-07. The CAD tools were supported by IC Design Education Center (IDEC), Daejeon, Korea.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," In Advances in Neural Information Processing Systems 25, pp. 1106–1114, 2012.

[2] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations," arXiv:1609.07061, 2016.

[3] M. Courbariaux and Y. Bengio, "Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or −1," arXiv:1602.02830, 2016.

[4] M. Courbariaux, Y. Bengio, and J.-P. David, "Binary Connect: Training Deep Neural Networks with binary weights during propagations," arXiv:1511.00363, 2016.

[5] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, 1971.

[6] D. B. Strukov, G. S. Snider, D. R. Stewart, and R.

S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[7] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," Nat. Nanotechnol. 3, 429–433, 4, 2008.

[8] A. S. Oblea, A. Timilsina, D. Moore, and K. A. Campbell, "Silver chalco-genide based memristor devices," Proc. IJCNN, 1–3, 2010.

[9] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett*., vol. 10, no. 4, pp. 1297–1301, 2010.

[10] S. N. Truong and K.-S. Min, "New Memristor-Based Crossbar Array Architecture with 50-% Area Reduction and 48-% Power Saving for Matrix-Vector Multiplication of Analog Neuromorphic Computing," *Journal of Semiconductor Technology and Science*, vol. 14, pp. 356-363, 2014.

[11] Son Ngoc Truong, Seok-Jin Ham, and Kyeong-Sik Min, "Neuromorphic crossbar circuit with nanoscale filamentary-switching binary memristors for speech recognition," *Nanoscale Research Letters*, vol. 9:629, Nov. 2014.

[12] S. Ngoc Truong, K. Van Pham, Wonsun Yang, Sangho Shin, Ken Pedrotti, and Kyeong-Sik Min, "New pulse amplitude modulation for fine tuning of memristor synapses," *Microelectronics Journal*, vol. 55, pp. 162-168, 2016.

[13] L. Gao, P.-Y. Chen, and S. Yu, "Demonstration of Convolution Kernel Operation on Resistive Cross-Point Array," *IEEE Electron Device Letters*, vol. 37, pp. 870-873, 2016.

[14] Leibin. Ni et al, "An Energy-efficient Matrix Multiplication Accelerator by Distributed In-memory Computing on Binary RRAM Crossbar," *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, p. 280, 2016.

[15] C. Yakopcic, M. Z. Alom, and T. M. Taha, "Extremely parallel memristor crossbar architecture for convolutional neural network implementation," *International Joint Conference Neural Networks (IJCNN)*, USA, 2017.

[16] Sachin Bhat, "SkyNet: Memristor-based 3D IC for artificial neural networks," *Nanoscale Architectures* (NANOARCH), USA, 2017.

[17] Li, C, "Three-dimensional crossbar arrays of self-rectifying Si/SiO2/Si memristors," *Nat. Commun*. 8, 15666, 2017.

[18] Cadence, "Virtuoso Spectre Circuit Simulator User Guide," Cadence Design System, Inc., San Jose, 2011.

[19] Lee M, "Resistance-Switching device having conductive fiber, fabric type resistance-switching device array having the same, and memory and sensor having the device or the array," U.S. Patent 2016, 15,213,352, 2017.

[20] Jo A, Seo Y, Ko M, Kim C, Kim H, Nam S, Choi H, Hwang CS, Lee MJ, "Textile resistance switching memory for fabric electronics," *Advanced Functional Materials*, vol. 27, pp. 1605593:1–11, 2017.

Khoa Van Pham received the B.S. and M.S. degrees in electronic engineering from the University of Technology and Education, Hochi-minh City, Vietnam, in 2010 and 2014, respectively. He is currently working toward the Ph.D. degree at Kookmin University. His research interests include circuit design, memristor, and memristor-based crossbar for neuromorphic computing systems.

Tien Van Nguyen received the B.S. and M.S. degrees in Electronic Engineering from Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, Vietnam, in 2014 and 2017, respectively. He is currently working toward the Ph.D. degree at Kookmin University, Seoul, Korea. His research interest is memristor crossbar-based neuromorphic computing systems.

Son Bao Tran received the B.S. degree in electronic and telecommunication technology engineering from Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, Vietnam, in 2010. He is currently working toward the M.S. degree at Kookmin University, Seoul, Korea. His research interest is memristor crossbar-based neuromorphic computing systems.

HyunKyung Nam received the B.S. degree in Electrical Engineering from Kookmin University, Seoul, Korea, in 2017. She is currently working toward the M.S degree at Kookmin University, Seoul. Korea. Her research interests include memristor crossbar-based neuromorphic computing systems and wearable computing.

Mi Jung Lee received the B.A and M.S degree in materials science and engineering from Seoul National University and the Ph.D. degree in physics from the University of Cambridge, Cambridge, U.K., in 2011. She also worked for Korea Electronic Technology Institute in display research center. Her research interest involves various solution-processed electronic devices and charge transport mechanism in it. Not only is she intrigued by the investigation in fundamental principle, but also the extension to the applications of ReRAM devices composed of threads towards woven electronics. Currently, she is an associate professor with Kookmin University, Seoul, Korea.

Byung Joon Choi received Ph.D. degree from Seoul National University, Seoul, Korea, in 2009. Since then he worked at the University of Pennsylvania, Philadelphia, PA, as a Postdoctoral Researcher. In 2011, he joined Hewlett-Packard Laboratories, Palo Alto, CA. He has been an Assistant Professor at the Department of Materials Science and Engineering, Seoul National University of Science and Technology (Seoultech) since 2013. His research fields are development of thin film processes majorly including atomic layer deposition technique applying for electronic memory and Neuromorphic devices.

Son Ngoc Truong received the B.S. and M.S. degrees in Electronics Engineering from The University of Technical Education Ho Chi Minh City, Vietnam, in 2006 and 2011, respectively, and the Ph.D degree in Electronics Engineering from Kookmin University, Seoul, Korea, in 2016. He was a research assistant at Kookmin University, Seoul, Korea from 2016 to 2017. He is currently lecturer at the Ho Chi Minh City University of Technical and Education. His current research interests include the circuits and architectures with memristor device, memristor crossbar-based neuromorphic computing system, brain-inspired system, artificial intelligence, and deep learning.

Kyeong-Sik Min received the B.S. degree in Electronics and Computer Engineering from Korea University, Seoul, Korea, in 1991, and the M.S.E.E. and Ph. D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1993 and 1997, respectively. In 1997, he joined Hynix Semiconductor Inc., where he was engaged in the development of low-power and high-speed DRAM circuits. From 2001 to 2002, he was a research associate at University of Tokyo, Tokyo, Japan, where he designed low-leakage memories and low-leakage logic circuits. In September 2002, he joined the faculty of Kookmin University, Seoul, Korea, where he is currently a Professor in the School of Electrical Engineering. He was a visiting professor at University of California, Merced, from Aug. 2008 to July 2009. Prof. Min served on various technical program committees such as Asian Solid-State Circuits Conference (A-SSCC), International SoC Design Conference (ISOCC), and Korean Conference on Semiconductors (KCS). He and his students received IDEC CAD & Design Methodology Award (2011), IDEC Chip Design Contest Award (2011), and IDEC Chip Design Contest Award (2012). He is a member of Institute of Electrical and Electronics Engineers (IEEE), Institute of Electronics Engineers of Korea (IEEK), and Institute of Electronics, Information, and Communication Engineers (IEICE) in Japan. His research interests include low-power VLSI, memory design, and power IC design.