Modeling and Demonstration of Hardware-based Deep Neural Network (DNN) Inference using Memristor Crossbar Array considering Signal Integrity

Taein Shin[1], Shinyoung Park[1], Seongguk Kim[1], Hyunwook Park[1], Daehwan Lho[1], Subin Kim[1], Kyungjune Son[1], Gapyeol Park[1], and Joungho Kim[1]
Terabyte Interconnection and package Laboratory
Korea Advanced Institute of Science and Technology (KAIST)[1]
Daejeon, Republic of Korea
taeinshin@kaist.ac.kr

***Abstract*— A hardware-based artificial intelligence (AI) operation using memristor crossbar array is a promising AI computing architecture due to its energy-efficiency. It mimics the computational form of matrix-vector multiplication, which is the main AI operation and is implemented in an analog way. However, the reliability problem is serious because of the hardware-based operation. In this paper, we propose a hybrid circuit model of a hardware-based deep neural network (DNN) for a large-scale memristor crossbar array in terms of signal integrity (SI) problems. After DNN classification training for the optimized weight matrix in memristors, we demonstrated and analyzed the effect of SI on DNN accuracy using the proposed model. It is possible to analyze the effect of the SI problems due to interconnection at the crossbar on the reliability of computational accuracy through this hybrid circuit model. Simulated accuracy of DNN inference is degraded up to 36.4% in the worst case due to IR drop and ringing depending on the physical dimension of array interconnection and operating frequency in a memristor crossbar array.**

***Keywords*— Deep neural network, inference, interconnection, memristor crossbar array, signal integrity**

I. Introduction

The main operation of an artificial intelligence (AI) algorithm is the matrix-vector multiplication, which is characterized by massively parallel matrix operations. In terms of hardware, it consumes lots of energy because it requires frequent off-chip memory access than conventional serial operation [1]. Therefore, the minimization of off-chip memory access is a more important issue on AI application especially.

Hardware-based operation using a memristor crossbar array is to integrate computation into the memory using a resistive non-volatile memory [2]. This is to combine arithmetic functions with the memory to eliminate the off-chip memory access. This mimics the computational form of matrix-vector multiplication and is implemented in an analog way. In other words, a matrix operation is done directly by detecting the current at each column of crossbar array from the multiplication of voltage and conductance of memristor-based on Kirchhoff's Current Law (KCL). The equations of conventional software-based matrix operation and memristor crossbar array-based matrix operation are as follows:

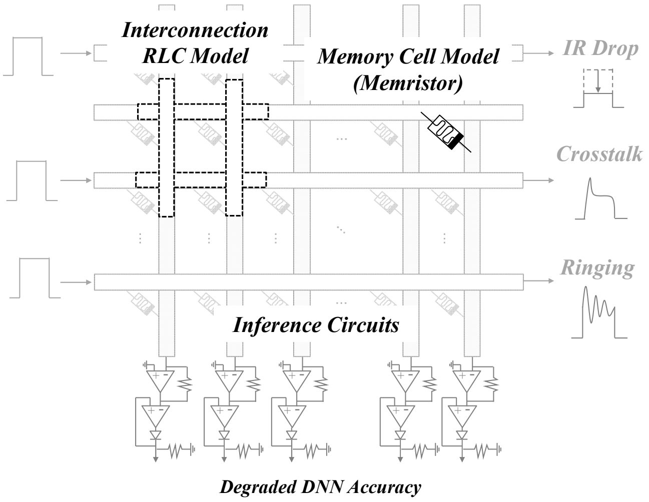


Fig. 1. A conceptual figure of hybrid circuit model including interconnection RLC model, memory cell model (Verilog-a) and inference circuit for demonstration of signal integrity effects on DNN, such as IR drop, crosstalk and Ringing from interconnection parasitic.

$$Y = \sum_{i=1}^{n} W_{ik} \cdot X_i \tag{1}$$

$$I_k = \sum_{i=1}^{n} \frac{1}{R_{ik}} \cdot V_i \tag{2}$$

where input X_i is same as the input voltage, V_i, and weight W_{ik} is same as the conductance of memristor, $\frac{1}{R_{ik}}$. In other words, since this memristor operates simultaneously with storage, it has a revolutionary energy-efficiency in AI operation.

However, hardware-based operation using memristor crossbar array has a serious reliability problem from circuit issues on analog-domain. Circuit issues such as interconnection parasitic, conductance and process variation of memristors and external noises, etc all directly affect the computation. These circuit issues in hardware-based operation greatly degrade the accuracy than the software-based operation. Especially, interconnection parasitic of crossbar array can be the main problem such as IR drop, crosstalk, and ringing depending on the array size and operating frequency. There have been several studies on the interconnection effects of crossbar array [3], [4]. However, conventional studies have not considered the crossbar on a real-like large scale, also do not use the complete interconnection model and memristor model, so it is insufficient to analyze the SI effects on the results of DNN.

In this paper, we propose a hybrid-circuit model and demonstrate the hardware-based DNN inference using large-scale memristor crossbar array, in terms of signal integrity problems that cause reliability problems from the interconnection parasitic of an array, as shown in the Fig. 1. We obtained a 2-layers DNN simple classification model that is achieved 99.4% of accuracy in software. Then, the optimized weight matrix and input are transferred to the memristor crossbar array. It is a hybrid circuit-modeled of interconnections, memristors, and inference circuits. The accuracy of DNN inference is degraded largely due to IR drop and ringing depending on the physical dimension of array interconnection and operating frequency in a memristor crossbar array. As a result, the accuracy dropped to 68.9% by IR drop from the resistance of interconnections and dropped to 36.4% by ringing from inductance and capacitance, in a worst-case.

II. Modeling of Large-scale Memristor Crossbar Array for DNN

A. The overall process of DNN Traning and Inference for classification

Fig. 2 describes the detailed overall process of training and inference. In training, first, define the target network size and parameters. After the data generation, training data is used for the general DNN training algorithm of feed-forward propagation, backpropagation, and gradient descent. Then, an initial weight matrix is obtained, it is used in test inference with random inference data for checking the accuracy of the network. If the desired accuracy is not achieved, retrain the weight matrix by tuning the hyper-parameters. After several iterations, the training process ends when a high enough accuracy is achieved.

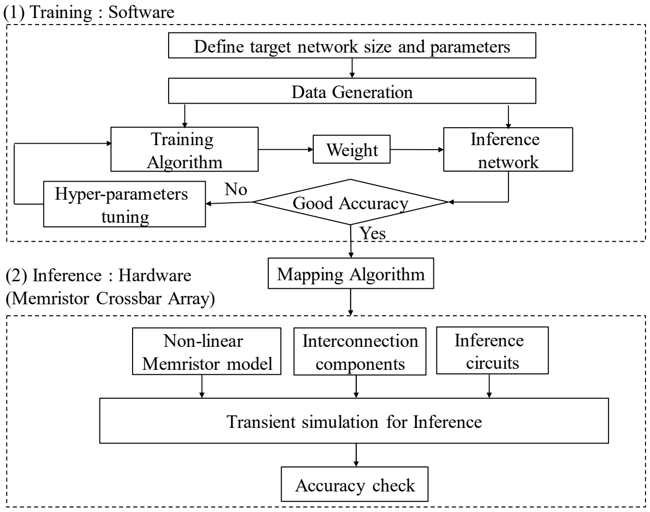


Fig. 2. Flow chart of overall training and inference process. Training is completed on the software and the weight matrix is converted to the conductance matrix of the memristor through a mapping algorithm. Finally, inference is performed on the hardware through the hybrid circuit model.

TABLE I. Hyper-paramemter Set-up of Deep Neural Network

Parameter			Value
Deep Neural Network	Classification of 3 Characters		
	Input Layer	Number of Nodes	3
	Hidden Layer	Number of Nodes	128
		Activation Function	ReLU
	Output Layer	Number of Nodes	3
		Activation Function	ReLU
Training	Training Sets		660
	Test Sets		60
	Accuracy		99.4%

Then, the optimized weight matrix is transferred to the memristor crossbar array with a proper mapping algorithm for inference. Since the array operates in an analog way, non-linear memristors and interconnection parasitic, which can directly affect the operation values, should be considered. Therefore, non-linear current-voltage memristor model, interconnection RLC components, and additional inference circuits are co-modeled in SPICE. Finally, transient simulation is repeated for inference, then the accuracy of DNN inference is verified with signal integrity effects.

Table I shows the hyper-parameters of our designed DNN classification model. The network has three layers. Each node of the input layer and output layer is three because this model is for a simple classification of three characters, '1', '2', and '3'. Matrix values corresponding to '1', '2', '3' are input, and three numbers are classified through output. The nodes of the hidden layer are 128 for targeting large-scale memristor crossbar array. The rectified linear unit (ReLU) activation function is used for the hidden and output layers. As a result, this network achieves 99.4% accuracy on software and an optimized weight matrix can be obtained.

B. Hybrid-Circuits Modeling of Memristor Crossbar Array for DNN Inference

In this section, we model the memristor crossbar array with inference circuits for the implementation of DNN inference in a memristor crossbar array. A memristor crossbar array is modeled into two parts with array interconnections and memristors. Fig. 3 shows that array interconnections are modeled as unit-cell RLC lumped elements. The RLC values of interconnections are extracted from Ansys Q3D extractor, and those are composed with unit resistance of signal line (R_w), the unit inductance of signal line (L_w), unit mutual inductance (L_m), the unit capacitance between adjacent signal lines (C_w) and unit capacitance between the signal to ground (C_g). The interconnection width (W) and space (S) between lines are designed equivalent, ranging from 50 nm to 500 nm, that physical dimensions of conventional memristor crossbar array interconnections fall within this range. The aspect ratio is fixed as two, so the thickness is always twice of width.

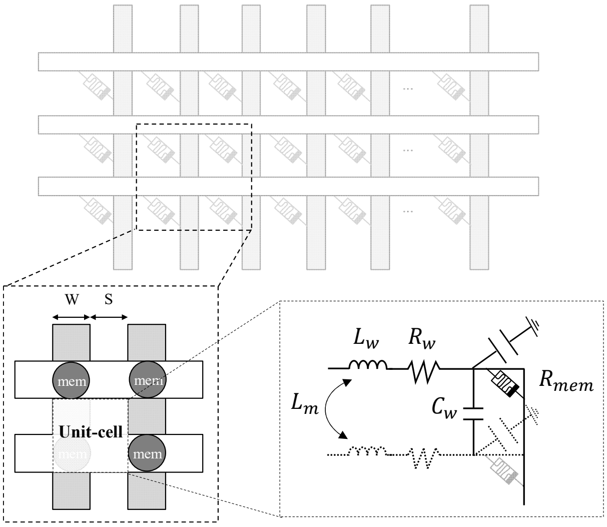


Fig. 3. A memristor crossbar array consisting of unit-cell RLC circuit modeled array interconnections and resistance of memristor. The unit-cell is modeled from 5 nm to 500 nm based on the interconnection width (W).

The next part is the model of a memristor. During the operation, the resistance of a memristor changes due to its non-linear characteristics to the applied voltage, unlike passive resistance. The HfOx RRAM Verilog-a model is used for more accurate implementation [5]. The current flowing through the memristor depends on the applied voltage the following equation:

$$I = I_0 \exp\left(-\frac{d}{d_0}\right) \sinh\left(\frac{V}{V_0}\right) \tag{3}$$

Since the current is proportional to the hyperbolic sine of the applied voltage, the change in the resistance decreases as the voltage approaches zero. Therefore, the input voltage as small as possible below 0.15V is assumed. The resistances of the Verilog-a modeled memristors respectively are entered at the intersections of the crossbar arrays as shown in Fig. 3.

The last part is the inference circuits. An inference is just simple multiplication and addition of input voltage and conductance of a memristor. However, the conductance of a memristor cannot have a negative value, and the weight matrix from training has lots of negative value or complex value. Therefore, the weight mapping algorithm is used for transferring the weight matrix to the proper conductance of the memristor [6]. Denote that components of weight matrix as w_{ij} and conductance of memristor as G_{ij}, where $w_{min} \leq w_{ij} \leq w_{max}$ and $G_{min} \leq G_{ij} \leq G_{max}$. Using linear transformation and matrix simultaneous equations, then the following output is obtained:

$$y = \frac{w_{max}-w_{min}}{G_{on}-G_{off}} \mathrm{Gx} + \frac{G_{max}w_{min}-G_{off}w_{max}}{G_{on}-G_{off}} \sum_{i=1}^{N} x_i \tag{4}$$

where Gx is matrix multiplication of conductance and input, from the output of only memristor crossbar array. Fig. 4 describes designed inference circuits using an operational amplifier (op-amp) and passive resistance with a memristor crossbar array for weight mapping and inference [7]. The output voltage $V_{inference}$ in Fig. 4 is as follows:

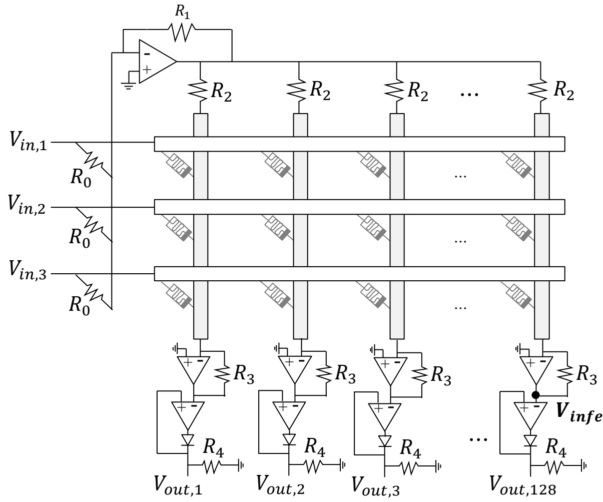


Fig. 4. The inference circuits for weight mapping and inference with memristor crossbar array. The mapping algorithm for converting the weight matrix on the software to the conductance matrix of the memristor is implemented as an ideal circuit.

$$V_{inference} = -R_3 \sum_{i=1}^{3} G_{ij} V_{in,i} + \frac{R_3 R_1}{R_2 R_0} \sum_{i=1}^{3} V_{in,i} \tag{5}$$

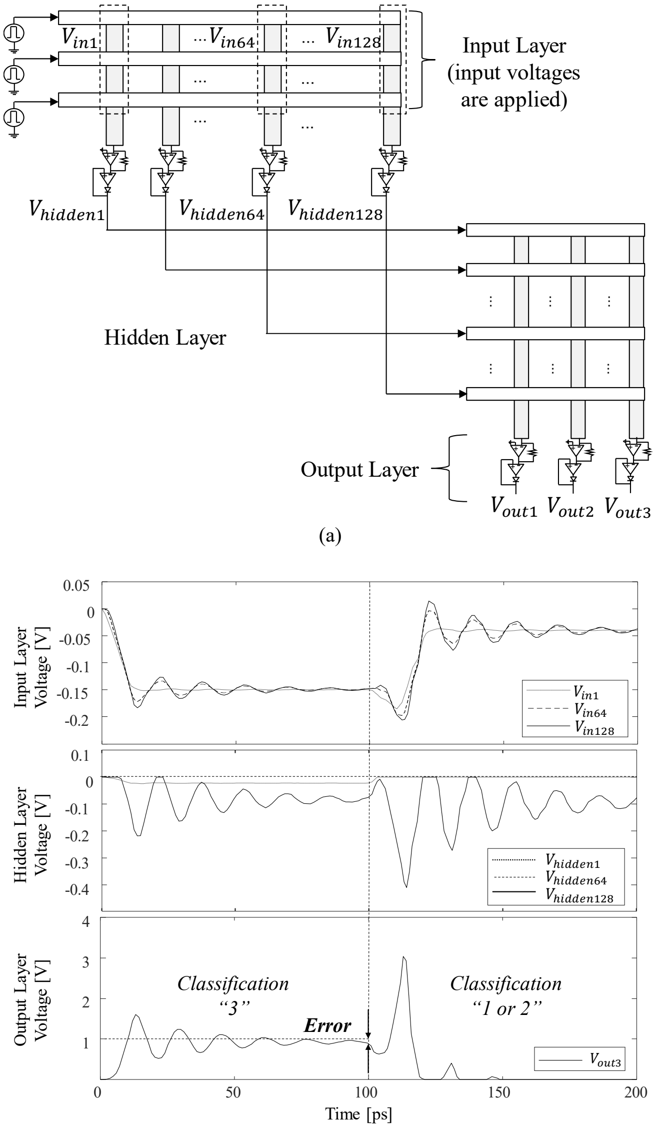
Therefore, if x and V_{in} of (4) are made the same, the coefficient of Equation (4) can be replaced by setting the resistance values of Equation (5). Finally, one layer can be fully implemented by adding circuits that represent the 'ReLU' activation function in front of V_{out} of Fig. 4.

III. Demonstration of DNN Inference in A Memristor Crossbar Array

In this chapter, we demonstrate DNN inference in memristor crossbar array according to array interconnection dimension and operating frequency. The results over the entire range are obtained to identify the high-frequency effect from DC to 10 GHz for the operating frequency and from 5nm to 500nm based on the width of interconnection.

Fig. 5(a) shows the schematic diagram of the designed memristor crossbar array in terms of input, hidden, and output layer. The input voltages are transferred from column 1 ($V_{in,1}$) to column 128 ($V_{in,128}$), and each voltage passes through the input layer, resulting in a hidden layer ($V_{hidden,k}$). The voltages of the hidden layer are transferred to the next layer, and finally, the same operation is repeated so that the output voltages appear from column 1 ($V_{out,1}$) to column 3 ($V_{out,3}$) in the output layer.

Fig. 5(b) shows the results of voltage waveforms of the input, hidden, and output layer in turn. This is the result at 2.5 GHz and 400 nm of interconnection width, and the rise/fall time of the driver is 5% of the signal period. For input and hidden layers, the voltage waveforms in the 1st, 64th, and 128th columns are shown respectively. As shown in Fig.5(b), the further the distance from the input, the greater the ringing voltage waveform occurs due to the inductance and capacitance of interconnections at the high-frequency range. These input



(a)

(b)

Fig. 5. (a) Schematic diagram of designed memristor crossbar array in terms of input, hidden and output layer. (b) Voltage waveform of input, hidden and output layer @2.5GHz, 400nm width of interconnection. The IR drop and ringing generated on the input layer voltage are multiplied and added according to the algorithm of DNN inference, and appear as output layer voltage.

voltages are added in each column, then the voltages of the hidden layer have a larger ringing according to operation values. The hidden layer voltage of the 128th column has a very large ringing, and the values above zero are eliminated by the ReLU activation function. After the sum of all row voltages at each column, output voltage indicates the answer of inference. As shown in Fig. 5(b) output voltage waveform, if the classification answer is '3', then the output of the third column indicates value '1' and others indicate '0'. The accuracy of operation is defined in this research as the following equations:

$$Error\ (\%) = \frac{1-(output\ voltage)}{1} \times 100 \tag{6}$$

$$Accuracy\ (\%) = 100 - Error \tag{7}$$

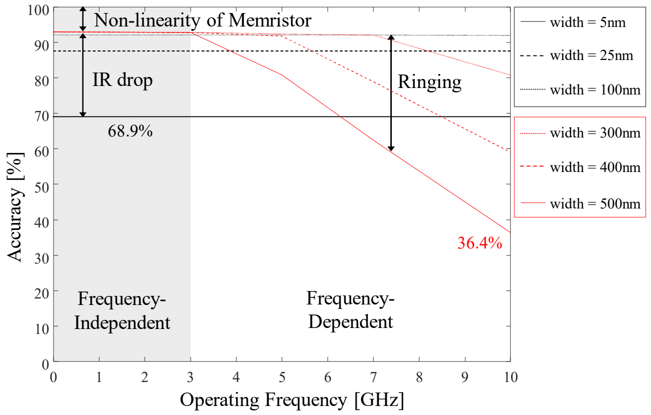


Fig. 6. The results of degraded hardware-based DNN inference accuracy on the memristor crossbar array depending on interconnection dimension and operating frequency.

Therefore, the error means the indicator of deviation from the answer, and the accuracy means the probability of classification.

Fig. 6 shows the results of DNN inference accuracy in our design memristor crossbar array depending on the interconnection dimension (width 5nm-500nm) and operating frequency (DC-10GHz). There are three factors to degrade accuracy. First is the default error that appears in all ranges by non-linearity of memristor about 5.75%. As mentioned earlier, the resistance of the memristor is changed gradually, because it has a non-linear relationship between current and applied voltage. Second is IR drop, as the interconnection dimension is smaller, its effect becomes serious because of the larger resistance of interconnection. It appears in all range of operating frequency as a DC error, and the accuracy becomes up to 68.9% at the smallest dimension of 5nm.

The last is the ringing, this depends on both interconnection dimension and operating frequency, so the accuracy is degraded up to 36.4%. As the larger interconnection dimension and the higher frequency, the error becomes serious. This ringing results from the reflections that occur with impedance mismatching in the last memristor load, due to the large resistance of the memristor. Since the inductance and capacitance of interconnection are small 5 nm, 25 nm, and 100 nm case, the ringing effect does not appear until 10 GHz, and only the IR drop effect is the dominant factor in reducing accuracy. On the other hand, the resonance effect due to the reflection starts to appear from about 300 nm due to the sufficiently large inductance and capacitance, and this effect becomes the dominant factor in reducing accuracy. In particular, the decreasing tendency of the accuracy due to ringing is almost linear with frequency. Because the DNN inference algorithm itself consists only of simple multiplication and addition, the ringing occurring at the input voltage is increased and combined at the same rate. As a result, the effect of ringing is hardly seen (frequency-independent domain) up to 3 GHz in all range of interconnection dimensions, however after 3 GHz, the accuracy drops sharply from the 300 nm of interconnection dimension due to larger inductance and capacitance of interconnections.

IV. Conclusion

In this paper, we modeled and demonstrated the hardware-based DNN inference using memristor crossbar array, in terms of signal integrity problems by interconnection parasitic of large-scale memristor crossbar array. After training of simple classification that is achieved 99.4% of accuracy in software, we modeled memristor crossbar array that has the same software-based network size with hybrid-circuits models of interconnections, memristor, and inference circuits. The main cause of accuracy degradation is IR drop in a relatively small dimension of interconnection below 100nm due to large resistance, and the accuracy is degraded up to 68.9% in a worst-case. On the other hand, ringing degrades accuracy in high operating frequency above 3GHz and the large dimension of interconnection over 300nm and the accuracy is degraded by up to 36.4% in the worst case. Therefore, the interconnection design is important in a large-scale memristor crossbar array because it has a crucial impact on the accuracy of DNN inference. This means the scaling of memristor crossbar array can be limited by an interconnection problem though there is a dense memristor.

Acknowledgment

We would like to acknowledge the technical support from ANSYS Korea. This work was supported in part by the National Research Foundation of Korea Grant funded by the Korean Government (2019M3F3A1A03079612).

References

[1] V. Sze, Y. Chen, T. Yang and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," in Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, Dec. 2017.

[2] M. Hu et al., "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2016, pp. 1-6.

[3] P. Chen *et al*., "Technology-design co-optimization of resistive cross-point array for accelerating learning algorithms on chip," *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, 2015.

[4] W. Lee and J. Kim, "Accuracy Investigation of a Neuromorphic Machine Learning System Due to Electromagnetic Noises Using PEEC Model", *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 10, pp. 2066-2078, Oct. 2019.

[5] X. Guan, S. Yu and H. -. P. Wong, "A SPICE Compact Model of Metal Oxide Resistive Switching Memory With Variations," in IEEE Electron Device Letters, vol. 33, no. 10, pp. 1405-1407, Oct.

[6] M. S. Tarkov, "Mapping Weight Matrix of a Neural Network's Layer onto Memristor Crossbar", *Optical Memory and Neural Networks*, 2015.

[7] SN. Troung, SM. Kim, "New Memristor-Based Crossbar Array Architecture with 50-% Area Reduction and 48-% Power Saving for Matrix-Vector Multiplication of Analog Neuromorphic Computing", *JSTS*, 2014.