Hardware-Software Co-Design for Analog Compute-in-Memory Accelerators using 1-bit Sense Amplifiers

Jihwan Cho[1] and Wanyeong Jung[a]
Department of Electrical Engineering, Korea Advanced Institute of Science and Technology
E-mail: [1]jihwancho@kaist.ac.kr

***Abstract* - In this paper, we present a hardware-software co-design methodology for analog compute-in-memory (CIM) accelerators with 1-bit sense amplifiers (SAs). While CIM macro using 1-bit SAs achieves high energy efficiency by eliminating multi-bit analog-to-digital converters (ADCs), it faces two key challenges: limitations in BNN layer size and the impact of SA random noise. Through neural network splitting, which divides layers into sub-blocks matching the size of the CIM macro rows, the BNN model fits the CIM macro while preserving accuracy. Additionally, the SA output probability model is obtained through measurements and replaces the binarization function of BNN, incorporating SA random noise into the BNN training process. Using these two approaches, we develop a framework to retrain BNNs tailored to the CIM accelerator using 1-bit SAs. The prototype 128x128 CIM accelerator fabricated in 28 nm technology achieves 97.81% MNIST inference accuracy and 12.6x reduction in readout energy compared to the prior work using 5-bit ADCs.**

***Keywords*— Compute-in-memory (CIM), hardware-software co-design, analog computing, binary neural network (BNN)**

I. INTRODUCTION

The development of deep neural networks (DNNs) has revolutionized various workloads with the high inference accuracies of the models. The size and computational complexity of models have increased significantly with the demand of complicated tasks. The digital accelerators utilizing parallelism have been developed for the DNN workloads which is dominated by massive amounts of matrix-vector multiplications (MVMs). The data transfer between computing unit and memory, however, significantly limits the throughput and energy efficiency of the accelerators [1]. To reduce the memory access, prior works optimized dataflow to maximize data reuse and compressed the model using quantization [2], [3].

To this end, the analog compute-in-memory (CIM) which integrates computation and memory has been proposed [4]. As shown in Fig. 1, the weights are stored in the bit cells of the CIM macro, and multiplications with the input activations fed to the word lines (WLs) are performed. The multiplication outputs of all the cells in the same column are accumulated in analog domain on the bit line (BL), which enables parallel access of all rows. CIM achieved high energy efficiency with this fully parallel multiply-accumulate (MAC) using analog computing. Meanwhile, the multi-bit analog-to-digital converter (ADC) which converts the accumulated analog voltage/current to digital output consumes significant energy, and this limits the gain obtained from the energy efficient analog computing. To reduce the overhead of output readout, low precision ADCs (e.g. 5-bit ADC) are used by quantizing the partial sum results. To further increase the energy efficiency of CIM, the multi-bit ADC can be replaced with the 1-bit sense amplifier (SA) by running the binary neural network (BNN). The accumulated results can be directly quantized to the 1-bit output activations of BNN with the 1-bit SA.

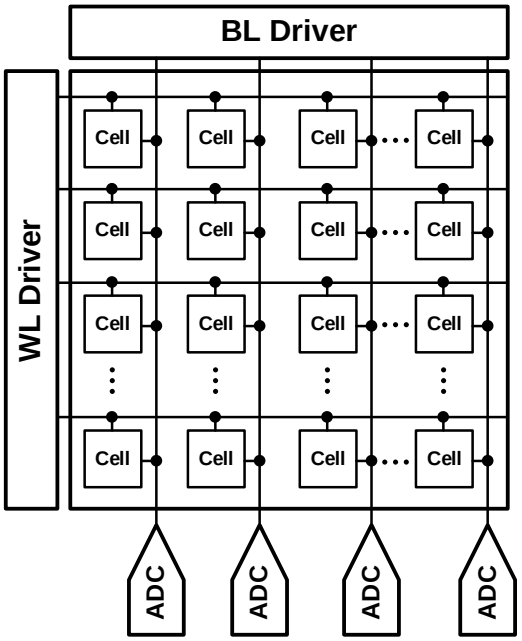


Fig. 1. The architecture of CIM macro with multi-bit ADC.

However, there are two challenges in mapping the BNN to the CIM macro using 1-bit SA as shown in Fig. 2. 1) The entire MAC operation required for computing a single output activation has to be performed in each column. In other words, the number of rows should be larger than the number of weights for a single output. Otherwise, the multi-bit

a. Corresponding author; wanyeong@kaist.ac.kr

Manuscript Received Nov. 20, 2024, Revised Dec. 31, 2024, Accepted Dec. 31, 2024

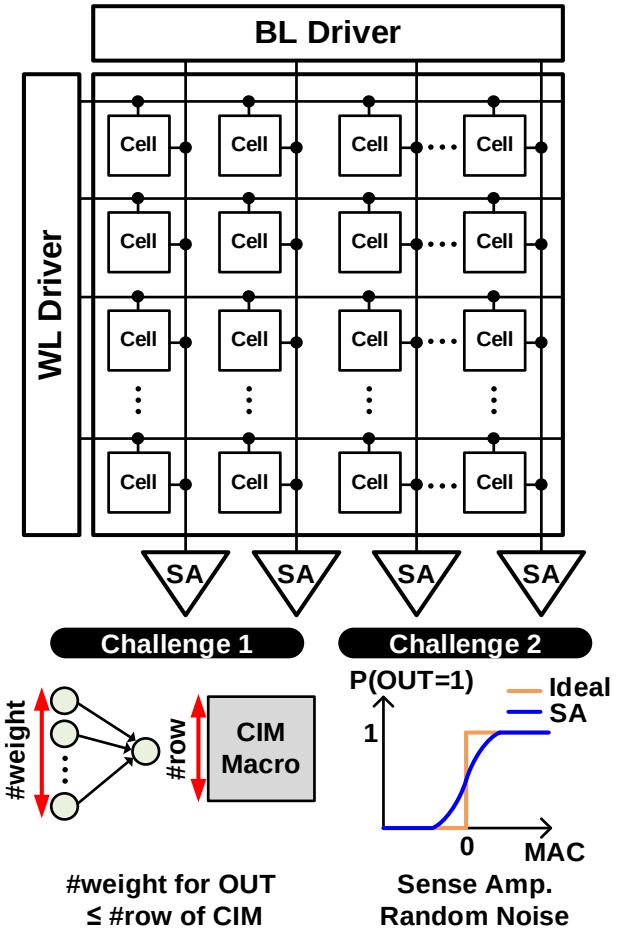


Fig. 2. The architecture of CIM macro with 1-bit SA and challenges in mapping the BNN to the CIM macro.

partial sums should be computed multiple times, and it requires multi-bit ADC. The limited number of rows in CIM macro due to the signal-to-noise ratio and readout delay makes this challenging. 2) The random noise of SA induces error during the 1-bit quantization of the output activation, resulting in inference accuracy drop. The modification of the BNN, therefore, is required to address these challenges.

This work presents a hardware-software co-design for analog CIM accelerators using 1-bit SAs. The BNN is retrained using neural network splitting and SA output probability model. The prototype chip achieves measured test accuracy of 97.81% for MNIST dataset.

The remainder of this paper is organized as follows. Section II presents two methodologies of the hardware-software co-design for the analog CIM accelerator using 1-bit SAs. Section III shows the measurement results of the prototype chip. Section IV concludes this work.

II. PROPOSED HARDWARE-SOFTWARE CO-DESIGN FOR ANALOG CIM ACCELERATOR USING 1-BIT SENSE AMPLIFIERS

A. Neural network splitting

BNNs use 1-bit representations for both input activations and weights. Output activations are computed by applying 1-bit quantization to the MAC results of inputs and weights, as described by the following equation:

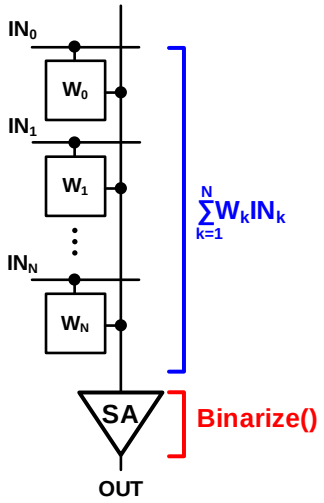


Fig. 3. The MAC operation of CIM Macro with 1-bit SA.

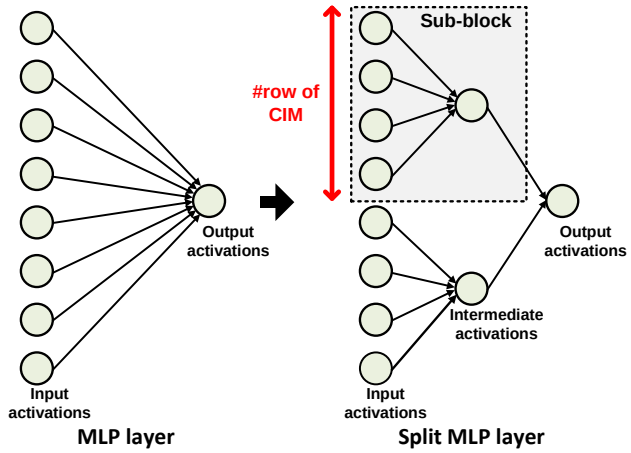


Fig. 4. The reconstruction of BNN using input splitting.

$$\text{OUT} = \text{Binarize}\left(\sum_{k=1}^{N} W_k IN_k\right) \quad (1)$$

Here, N denotes the number of inputs and weights required to compute the output. The conventional CIM macro can compute outputs even when N exceeds the number of rows in the macro by using multi-bit ADCs to iteratively compute partial sums. In contrast, the CIM macro with 1-bit SAs directly execute the Binarize() function in equation (1), as shown in Fig. 3. In this design, the number of rows should be larger than N. In other words, it is required that all MAC operations for the output is completed in a single column of the CIM macro. However, due to challenges such as SNR degradation and readout delay, analog CIM macros are typically designed with a limited row count, often in the hundreds [4]. Meanwhile, the weight vector size, N, of the simplest BNN model for MNIST is 2048, which exceeds the typical number of rows in CIM macro. This mismatch limits the direct mapping of conventional BNNs to CIM macro with 1-bit SAs.

To overcome this limitation, we adopt the input splitting technique proposed in [5] to rebuild the BNN. As shown in Fig. 4, the BNN layer is divided into multiple sub-blocks, each with a weight size matching the number of rows in the CIM macro. Intermediate activations are added between input activations and output activations. This enables each

TABLE I. The baseline BNN MLP for MNIST dataset and the reconstructed BNN applying input splitting.

Sub-block size	Layer
Baseline	748-2048-2048-2048
64	748-(32x64)-(32x64)-2048
128	748-(16x128)-(16x128)-2048
256	748-(8x256)-(8x256)-2048
512	748-(4x512)-(4x512)-2048

TABLE II. Test inference accuracy of the baseline BNN and the reconstructed BNNs for MNIST dataset.

Sub-block size	Test Accuracy (%)
Baseline	98.25
64	97.92
128	98.02
256	98.12
512	98.21

sub-block to be mapped onto a CIM macro. To maintain the network complexity, weights between intermediate and output activations are fixed to 1, as described in [5]. The reconstructed BNN model is then retrained.

The neural network splitting achieves high accuracy for a wide range of row counts which can vary depending on the design constraints. For the baseline model, we used the BNN MLP for MNIST from [6]. The model is reconstructed for four typical numbers of CIM rows (64, 128, 256, 512) as shown in Table 1. The reconstructed models are trained and Table 2 shows their inference accuracy. For all cases, the maximum accuracy drop was 0.33%, demonstrating the feasibility of applying input splitting to CIM macros.

B. Sense amplifier output probability modeling

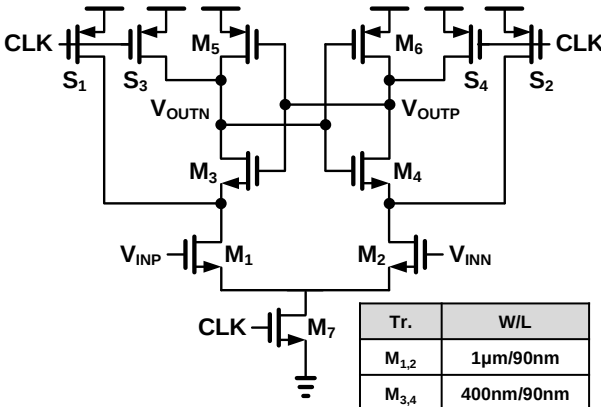
As shown in Fig. 3, the SA performs 1-bit quantization of the MAC operation results. Conventional BNNs utilize deterministic binarization defined by the Sign() function:

$$\mathrm{Sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (2)$$

However, due to the influence of random noise, the SA does not perfectly replicate the behavior of Sign(). Therefore, running the conventional BNN directly on the CIM accelerator results in substantial accuracy drop, depending on the magnitude of the random noise.

The 1-bit SA used in the CIM macro is implemented with a StrongARM latch, as shown in Fig. 5. This design uses a clocked differential pair, offering high sensitivity and low static power consumption [7]. The SA takes the analog BL voltage of the MAC result as its input and outputs a 1-bit digital value. However, during this binarization process, random noise sources, such as thermal noise, cause the SA output to exhibit a probabilistic distribution near the threshold (MAC value of 0).

The top plot of Fig. 6 shows the measured output probabilities of the prototype chip's 128 column SAs as a function of the MAC value. By averaging them, the SA output of the CIM accelerator can be modeled as a



Tr.	W/L
$M_{1,2}$	1μm/90nm
$M_{3,4}$	400nm/90nm
$M_{5,6}$	200nm/30nm
M_7	200nm/180nm
S_{1-4}	100nm/30nm

Fig. 5. The schematic of the 1-bit sense amplifier.

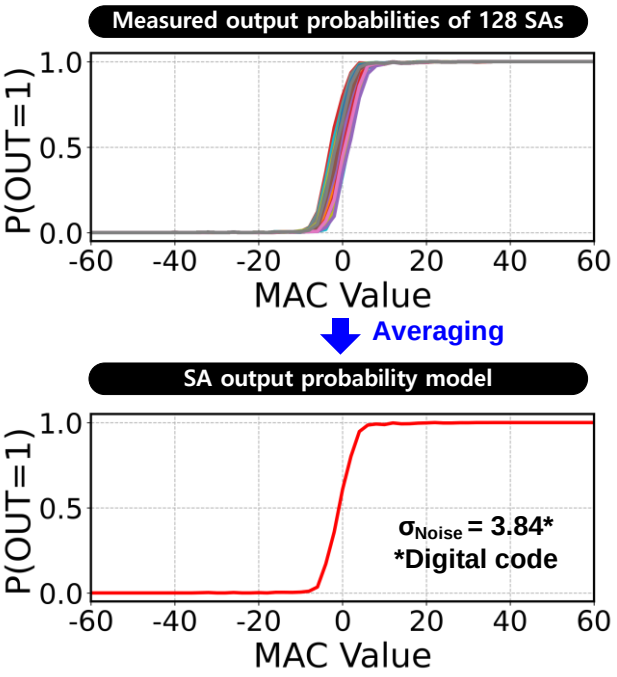


Fig. 6. The measured output probabilities of 128 1-bit sense amplifiers and 1-bit sense amplifier output probability model.

probabilistic distribution with a variation of σ_{Noise} = 3.84 digital codes, as shown in Fig. 6 (bottom). Based on this probability model, a new binarization function can be defined:

$$\mathrm{Binarize}(x) = \begin{cases} +1, & with\ probability\ p(x) \\ -1, & with\ probability\ 1-p(x) \end{cases} \quad (3)$$

Here, *p(x)* represents the SA output probability model. By replacing the Sign() in the BNN with this modeled binarization function, Binarize(), the effect of the SA random noise can be incorporated into the training process. During training, the straight-through estimator with hard tanh was applied to the backpropagation of the binarization layer, as in conventional BNNs. This method is performed on a per-chip basis, since the SA output probability model *p(x)* varies from chip to chip due to process variations.

The measurement results of the prototype analog CIM

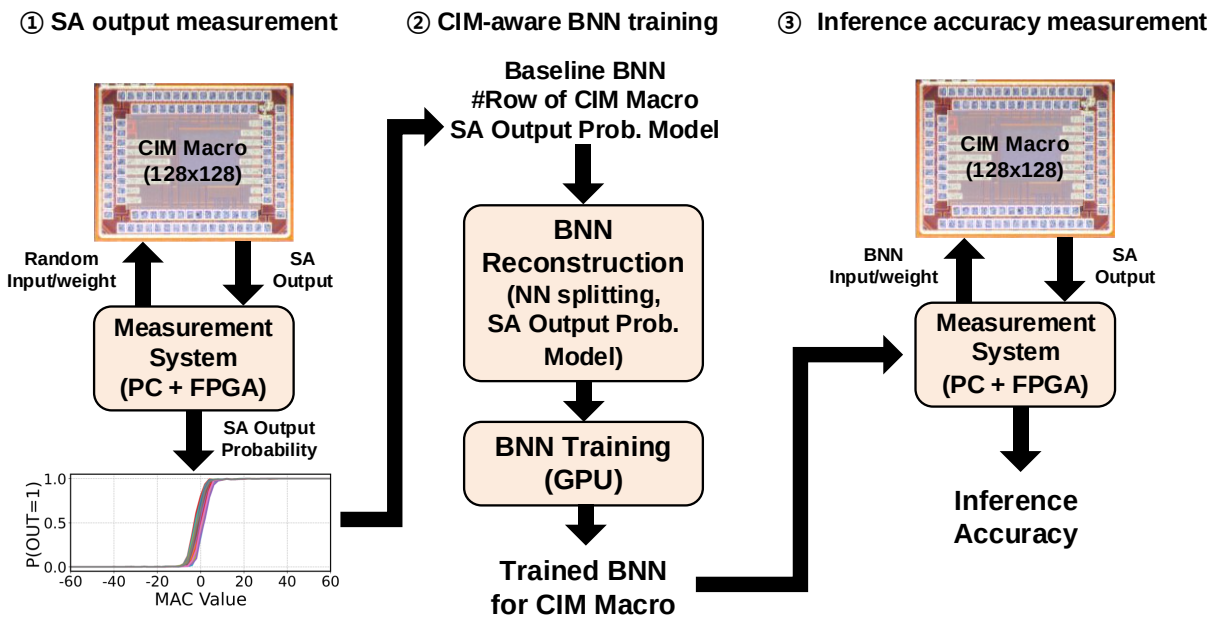


Fig. 7. The measurement framework for BNN inference accuracy evaluation using hardware-software co-design.

accelerator using a BNN trained with the two methodologies – neural network splitting and SA output probability modeling – will be discussed in the next section.

III. RESULTS AND DISCUSSION

The prototype chip of the analog CIM accelerator is fabricated in a 28 nm CMOS technology. The CIM accelerator consists of 128x128 CIM macro, a digital controller, interface logics for I/O.

Fig. 7 shows the measurement framework for evaluating the BNN inference accuracy using the proposed hardware-software co-design. The framework consists of the following steps. First, the SA output is measured using the measurement system consisting of a host PC and FPGA. Random weights and inputs are fed to the CIM accelerator, and the outputs of 128 SAs are read as the results of MAC operations. The SA output probability model is derived from the read output data. Second, CIM-aware BNN training is performed. The baseline BNN is reconstructed through neural network splitting and the SA output probability model, and the reconstructed model is trained on GPUs. Finally, the inference accuracy of the trained BNN model is measured on the CIM accelerator using the test dataset.

We trained the baseline BNN MLP for MNIST from Section II using this framework and measured the inference accuracy on 10k test images. Fig. 8 shows the comparison between the inference accuracy of the baseline BNN model and the CIM accelerator. By reconstructing and retraining the BNN considering the characteristics of the CIM hardware, a high accuracy of 97.81% on the CIM accelerator is achieved, only 0.21% lower than the baseline.

Additionally, replacing multi-bit ADCs with 1-bit SAs significantly reduces the energy consumption of the output readout. As shown in Fig. 9, our prototype chip consumes 12.6 times less energy for A/D conversion compared to the CIM macro using 5-bit ADCs [8].

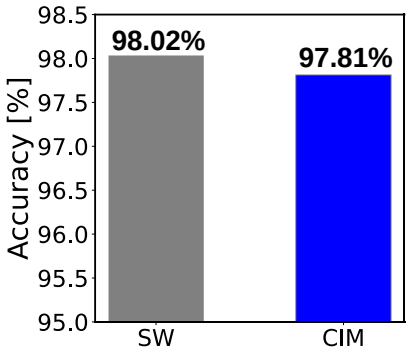


Fig. 8. The inference accuracies of baseline model (SW) and prototype chip (CIM) for 10k MNIST test images.

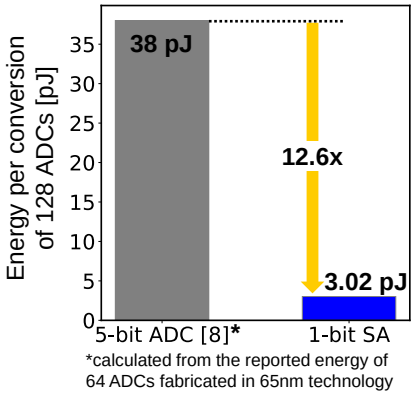


Fig. 9. The energy per conversion of 128 5-bit ADCs [8] and 128 1-bit SAs (this work).

IV. CONCLUSION

This paper presents a hardware-software co-design for analog CIM accelerators using 1-bit SAs. To address the significant power overhead of multi-bit ADCs in conventional CIM accelerators, 1-bit SA-based BNN CIM accelerators have been proposed. However, two major challenges, limitations in BNN layer sizes and the SA

random noise, hinder the mapping of BNNs onto the CIM accelerators. To overcome these issues, we utilize two key techniques: neural network splitting and SA output probability modeling. The neural network splitting restructures BNN layers into sub-blocks matching the size of the CIM macro, enabling compatibility with hardware constraints while maintaining baseline-level accuracy through retraining. Additionally, SA output probabilities are measured and modeled, and this model is incorporated into the BNN to recover the accuracy drop caused by the SA random noise. A prototype CIM accelerator fabricated in 28nm CMOS technology demonstrates our framework, achieving 97.81% inference accuracy on the MNIST test dataset and 12.6x reduction in readout energy compared to the prior work.

ACKNOWLEDGMENT

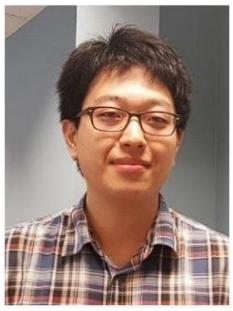
This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (Ministry of Science and ICT, MSIT) (No.2020-0-01297, Development of Ultra-Low Power Deep Learning Processor Technology using Advanced Data Reuse for Edge Applications). The chip fabrication was supported by the IC Design Education Center (IDEC), Korea.

REFERENCES

[1] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pp. 10-14, 2014.
[2] Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149, 2015.
[3] Han, Song, et al. "EIE: Efficient inference engine on compressed deep neural network." in ACM SIGARCH Computer Architecture News, 44.3: 243-254, 2016
[4] Verma, Naveen, et al. "In-memory computing: Advances and prospects." in IEEE Solid-State Circuits Magazine, 11.3: 43-55, 2019.
[5] Kim, Yulhwa, et al. "Input-splitting of large neural networks for power-efficient accelerator with resistive crossbar memory array." in Proceedings of the International Symposium on Low Power Electronics and Design, pp. 1-6, 2018.
[6] Hubara, Itay, et al. "Binarized neural networks." in Advances in neural information processing systems, 29, 2016.
[7] Razavi, Behzad. "The StrongARM latch [a circuit for all seasons]." in IEEE Solid-State Circuits Magazine, 7.2: 12-17, 2015.
[8] Song, Jiahao, et al. "A 4-bit Calibration-Free Computing-In-Memory Macro With 3T1C Current-Programed Dynamic-Cascode Multi-Level-Cell eDRAM." in IEEE Journal of Solid-State Circuits, vol. 59, no. 3, pp. 842-854, 2024.

Jihwan Cho received the B.S. degree in electrical and electronics engineering from Chung-Ang University, Seoul, South Korea, in 2021, and the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2023. He is currently pursuing the Ph. D. degree at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea.

His research interests include low-power digital circuits and energy-efficient deep neural network accelerators.



Wanyeong Jung received the B.S. degree from Seoul National University, Seoul, South Korea, in 2012, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2014 and 2017, respectively. He was a Research Intern with NVIDIA Research, Austin, TX, USA, in 2016. From 2017 to 2019, he was a Post -Doctoral Associate with Microsystems Technology Laboratories, Massachusetts Institute of Technology, Cambridge, MA, USA. Since 2019, he has been an Assistant Professor with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea.

His research interests include low -power circuits and systems, energy -efficient edge computing, and Internet of Things (IoT).