Wordline Driver for Processing-in-Memory Using the Input Bit-Slicing Method

Sojoong Kim[1], Beomjun Park[1], Jinhyeok Kim[1], Yoon Kim[1,3], and Minsuk Koo[2,3a*]
[1]School of Electrical and Computer Engineering, University of Seoul, Seoul 02504, South Korea
[2]School of Advanced Fusion Studies and AI Semiconductor, University of Seoul, Seoul 02504, South Korea
[3]IM Electronics co., Seoul 02505, South Korea
E-mail: kimsopark77@uos.ac.kr

Abstract - The von Neumann bottleneck, caused by the performance gap between memory and processing units, has become a critical issue in modern computing systems. This bottleneck fundamentally limits system throughput and energy efficiency. It becomes particularly critical as data-centric applications continue to demand higher bandwidth and faster access. To overcome this limitation, analog processing-in-memory has emerged as a promising candidate to address this challenge. In this work, we designed a wordline driver for processing-in-memory systems that use the input bit-slicing method, where fixed-length binary pulses serve as inputs. The proposed wordline driver was fabricated using the TSMC 180 nm CMOS logic process. Measurement results confirm that the wordline driver successfully generates 4-bit fixed-length binary pulses at the desired voltage levels, ensuring reliable wordline activation during input bit slicing. To provide a system-level context, we also describe a successive integration-and-rescaling neuron fabricated on the same 180 nm CMOS wafer. The integration of the proposed driver with this neuron was established in our prior work, reinforcing the claim that it effectively supports robust input bit-slicing operations in processing-in-memory architectures. Ultimately, these results confirm the feasibility of the proposed wordline driver for use in practical analog processing-in-memory systems.

***Keywords*— Processing-in-memory, Wordline driver, Input bit slicing, Level shifter, Buffer**

I. INTRODUCTION

With the rapid growth of data-centric applications such as artificial intelligence, the demand for high-efficiency and low-power computing systems has increased significantly. However, modern computing systems are still based on an architecture in which memory and processing units are physically separated. This separation leads to the von Neumann bottleneck, caused by the performance gap between the two units. To address this issue, extensive research has been devoted to processing-in-memory (PIM), which performs computation directly inside memory. Especially, analog PIM based on current-summation method for vector-matrix multiplication (VMM) has received considerable attention owing to its ability to execute large-scale VMM operations in parallel with high energy-efficiency [1]–[3].

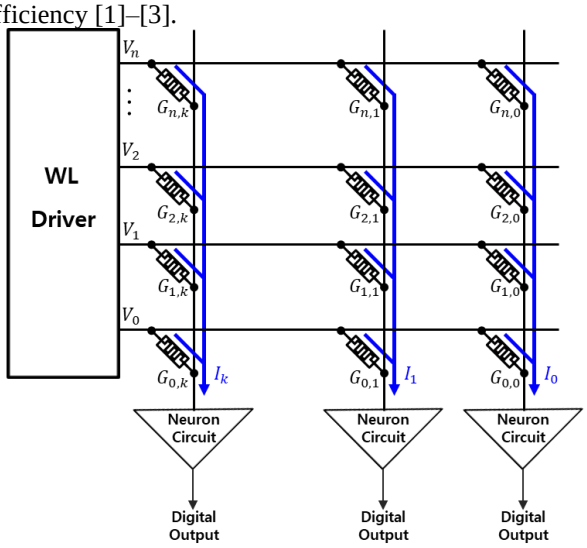


Fig. 1. Architecture for performing vector-matrix multiplication in a memory crossbar array based on the current-summation method

Fig. 1 illustrates the architecture for performing VMM in a memory crossbar array based on the current-summation method. Through the wordline (WL) driver, the input voltage V_i is applied to the i^{th} row WL of the memory crossbar array. This voltage is multiplied by the conductance $G_{i,j}$ of the memory cell located at the i^{th} row and j^{th} column, generating a current I_j on the j^{th} column bitline (BL). The corresponding relationship is expressed in Equation (1). Here, the memory array consists of n+1 WLs and k+1 BLs. The WLs are labeled from 0 to n, and the BLs are labeled from 0 to k.

$$I_j = \sum_{i=0}^{n} G_{i,j} V_i \qquad 0 \le j \le k \tag{1}$$

The generated current is then converted into a digital value through the neuron circuit. Since the VMM operation is controlled by both the WL driver and the neuron circuit, the performance of these two components plays a critical role in determining the overall efficiency of the PIM system [4].

a. Corresponding author; koo@uos.ac.kr

Manuscript Received Sep. 24, 2025, Revised Apr. 2, 2026, Accepted Apr. 6, 2026

In current-based analog PIM, the input bit-slicing scheme applies inputs as fixed-length binary pulses, enabling efficient design of both the WL driver and the neuron circuit [5].

The proposed WL driver, fabricated using the TSMC 180 nm CMOS logic process, is designed to generate precise binary pulses for input bit-sliced PIM architectures. Measurement results confirm that the driver successfully generates 4-bit pulse sequences at the target voltage levels, ensuring reliable WL activation and signal integrity. To provide a system-level context, we refer to the successive integration-and-rescaling (SIR) neuron. The detailed architecture of the SIR neuron and its integration with the driver were established in our prior work [6]. Ultimately, this system-level verification underscores the driver's versatility and its potential as a robust solution for practical, real-world PIM applications. Furthermore, in this system, the WL driver is implemented as a dedicated interface component, physically separated from the synaptic array and neuron to enable flexible system-level integration. By addressing system-level voltage, timing, and loading constraints, this work demonstrates the applicability of the proposed WL driver across diverse synaptic arrays and neuron circuits in input bit-sliced PIM systems.

This paper is organized as follows. Section II explains the input bit slicing. Section III details the proposed WL driver architecture, including the level shifter and output buffer, and presents measurement results. Section IV describes the SIR neuron used in input bit-sliced PIM systems and briefly reviews its previously reported integration with the proposed WL driver. Section V discusses the system-level integration of the proposed WL driver with external synaptic arrays and neurons in input bit-sliced PIM systems. Finally, Section VI concludes the paper.

II. INPUT BIT SLICING

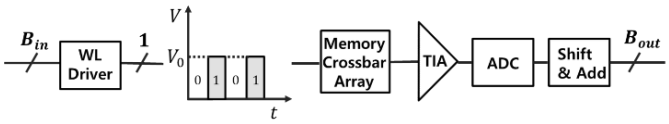


Fig. 2. Basic architecture of input bit slicing

Among the input schemes for applying signals to the memory crossbar array in an analog PIM system, the input bit-slicing method is widely used. In this scheme, fixed-length binary pulses are applied to the crossbar array one bit at a time. When employing input bit slicing, the WL driver only needs to generate two voltage levels and fixed-length pulses, which simplifies hardware implementation and reduces area overhead [5].

Fig. 2 illustrates the basic architecture of input bit slicing. An input signal B_{in} is transmitted to the WL driver, which sequentially decomposes the signal into individual bits from the MSB to the LSB. Each bit is then converted into the required voltage 0 V or V_0 and applied to the memory crossbar array. Following the same principle as Equation (1), the VMM result for each bit slice is calculated in the crossbar array. The resulting current is converted into a digital value through a transimpedance amplifier (TIA) and an analog-to-digital converter (ADC), which together act as the neuron circuit. The final output is obtained by repeatedly performing shift-and-add operations across all bits from MSB to LSB.

III. WL DRIVER

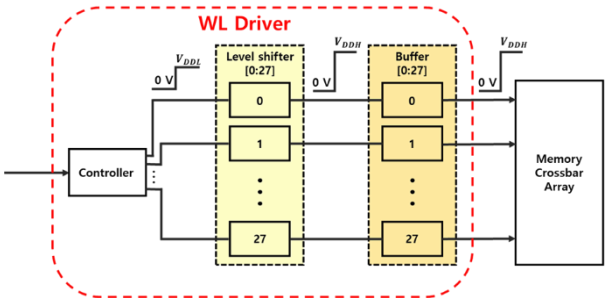


Fig. 3. Overall architecture of the WL driver

The overall architecture of the WL driver proposed in this paper is shown in Fig. 3. The controller generates 28 independent digital signals, each corresponding to a different WL, bit by bit and delivers them to the level shifter in accordance with the input bit-slicing scheme. For example, when an input of $0101_{(2)}$ is applied to the WL, the sequence low-high-low-high is transmitted to the level shifter. In this process, the digital signal has a voltage of 0 V when low and V_{DDL} when high. The level shifter then converts the received digital signals into 0 V and V_{DDH}, which are required by WL of the memory. These voltages are subsequently delivered to the memory crossbar through a buffer. The buffer minimizes the increase in rising and falling times that may occur due to the capacitive components of the WLs in the memory crossbar array. The proposed WL driver provides 28 outputs.

A. Level shifter

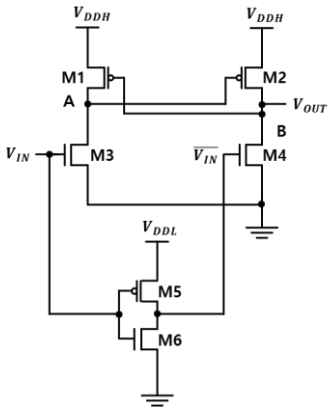


Fig. 4. Schematic of the level shifter

Fig. 4 illustrates the schematic of the level shifter. The V_{IN} enter the level shifter and is subsequently output as V_{OUT}. V_{IN} has a value of 0 V and V_{DDL}. V_{OUT} has a value of 0 V and V_{DDH}. In this context, V_{DDL} serves as the supply voltage for the inverter consisting of M5 and M6. V_{DDH} serves as the supply voltage for M1 and M2.

The operational principle is as follows. When the input V_{IN} ($\overline{V_{IN}}$) transitions from low to high, the transistor M3 (M4) is turned on and M4 (M3) is turned off. Then, node A (B) is pulled down and reaches $V_{DDH}-V_{th}$, it turns on transistor M2 (M1). Here, V_{th} denotes the threshold voltage of transistors M1 and M2. As M2 (M1) turns on, node B (A) begins to be charged, weakening M1 (M2). It allows node A (B) to be pulled down more rapidly. This action creates a positive feedback loop that accelerates the voltage level conversion. As a result, when V_{IN} is 0 V, V_{OUT} is 0 V. When V_{IN} is V_{DDL}, V_{OUT} is V_{DDH} [7].

B. Buffer

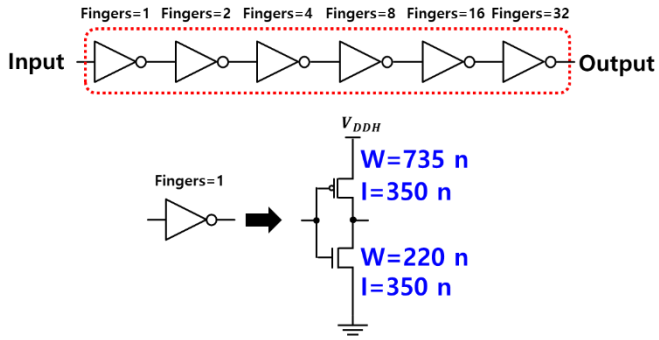
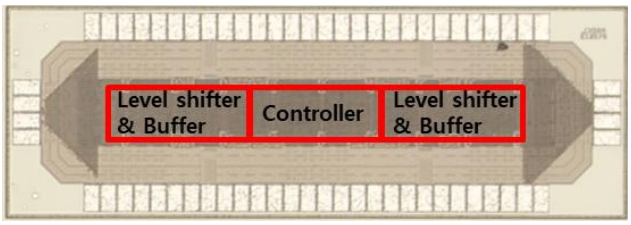


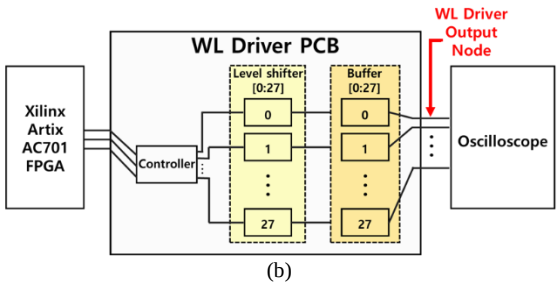
Fig. 5. Schematic of the buffer

Fig. 5 shows a buffer composed of an inverter chain. The purpose of this buffer is to drive the capacitive load of the WLs of memory crossbar array. An inverter chain is employed because a single inverter has limited drive strength. Directly driving a large capacitive load with a single inverter causes significant delays. By cascading small, medium, and large inverters, the load can be driven progressively, thereby improving the rising and falling times and minimizing the overall delay [8].

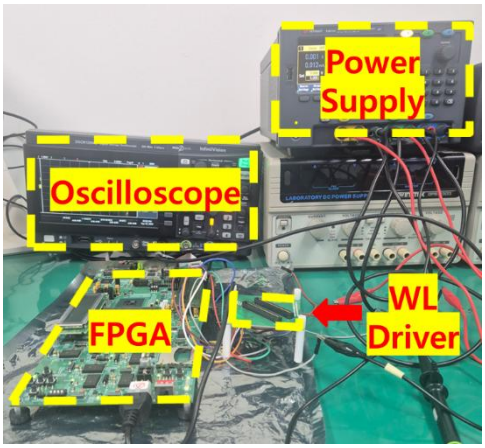
C. Measurement results



(a)



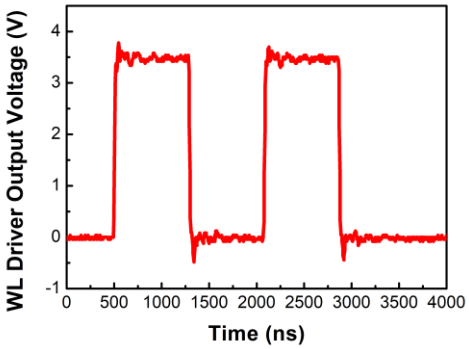
(b)



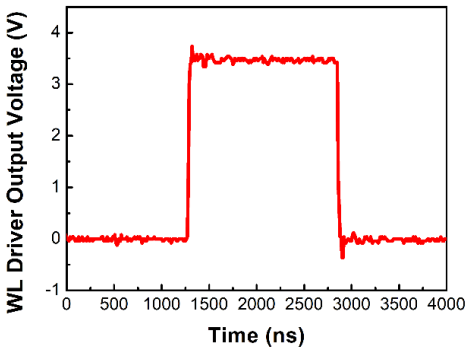
(c)

Fig. 6. (a) Die microphotograph of the WL driver chip, (b) Block diagram of the measurement system, (c) Photograph of the experimental setup

Fig. 6(a) presents the die microphotograph of the WL driver chip. The WL driver was fabricated using the TSMC 180 nm CMOS logic process. Fig. 6(b) illustrates the block diagram of the measurement system. The fabricated chip was packaged on a PCB, and the control signals for the chip were provided through a Xilinx Artix AC701 FPGA. An oscilloscope was employed to measure the waveforms at the WL driver output node. Fig. 6(c) shows a photograph of the experimental setup.



(a)



(b)

Fig. 7. Voltage waveforms at the WL driver output node: (a) $0101_{(2)}$, (b) $0110_{(2)}$

Fig. 7 shows the voltage waveform at the WL driver output node, as measured with an oscilloscope. For the

measurement, V_{DDL} was set to 1.8 V and V_{DDH} was set to 3.3 V, and the operating frequency was set to 1.25 MHz. As expected from the input bit-slicing scheme, the output waveform appeared sequentially from the MSB to the LSB. With a bit pattern of $0101_{(2)}$, the sequence of 0 V-3.3 V-0 V-3.3 V appeared in order. Similarly, with a bit pattern of $0110_{(2)}$, the sequence of 0 V-3.3 V-3.3 V-0 V was obtained.

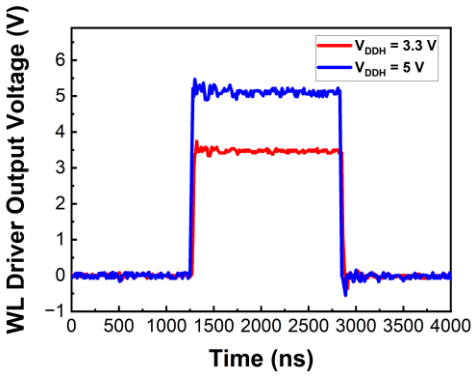


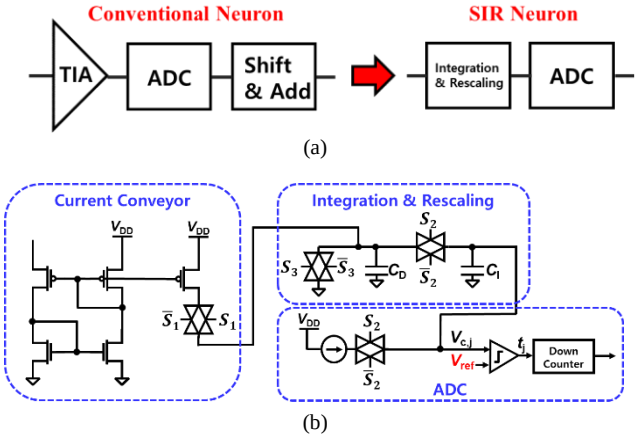
Fig. 8. Voltage waveforms at the WL driver output node for $0110_{(2)}$ when V_{DDH} was set to 5 V and 3.3 V

Fig. 8 shows the voltage waveform at the WL driver output node when V_{DDH} was set to 5 V and 3.3 V. It is confirmed that the output follows the V_{DDH} values as configured. Therefore, the WL driver can provide the voltage levels required by the memory array.

In addition, the operating voltage and frequency ranges of the WL driver were experimentally verified. When V_{DDL} was set to 1.8 V and V_{DDH} was set to 3.3 V, stable operation was confirmed up to an operating frequency of 15 MHz. Furthermore, for a fixed V_{DDL} of 1.8 V, the WL driver maintained correct functionality over a wide V_{DDH} range from 1.8 V to 7 V. Similarly, when V_{DDH} was fixed at 3.3 V, proper operation was observed for V_{DDL} values ranging from 1.5 V to 3 V.

These results demonstrate that the proposed WL driver supports flexible voltage configurations and a wide frequency range, making it suitable for various memory array requirements in input bit-sliced PIM systems.

IV. SIR NEURON



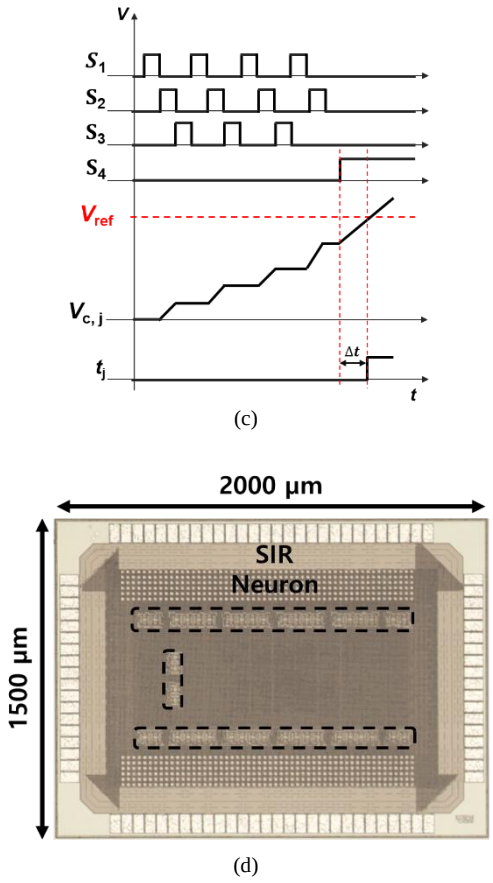


Fig. 9. (a) Conceptual comparison between a conventional neuron and an SIR neuron in input bit-sliced PIM systems, (b) Circuit schematic of the SIR neuron, (c) Timing diagram of the SIR neuron, (d) Die micrograph of the fabricated SIR neuron implemented in a 180-nm CMOS process

The SIR neuron is one of the neuron circuits specifically designed for input bit-slicing PIM systems. As illustrated in Fig. 9(a), the SIR neuron performs the shift-and-add operation in the analog domain, significantly reducing the overhead of analog-to-digital conversion. Fig. 9(b) and (c) show the circuit schematic and the timing diagram, respectively. The neuron was fabricated on the same 180-nm CMOS wafer as the driver (Fig. 9(d)). The detailed circuit implementation and measurement results of the SIR neuron, as well as its integration with the proposed WL driver in a PIM system, have been reported in our prior work [6]. This prior system-level demonstration further strengthens the claim that the proposed WL driver can be practically employed in functional PIM systems.

V. SYSTEM-LEVEL CONSIDERATIONS FOR THE WL DRIVER

The proposed WL driver is integrated at the system level within an input bit-sliced PIM architecture. As illustrated in Fig. 10, the WL driver interfaces with an external synaptic array and an SIR neuron, functioning as a critical interface block rather than a standalone pulse generator.

When connected to the synaptic array and neuron circuit, multiple design constraints must be addressed. To ensure

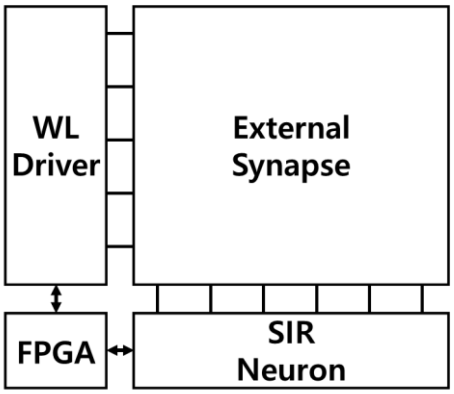


Fig. 10. System-level block diagram of the proposed WL driver integrated with an external synaptic array and SIR neuron

seamless operation within a complete PIM system, the driver is designed to accommodate the specific voltage requirements of various synaptic arrays as well as the differing current levels and operational timings of the interfaced neurons. Accordingly, the WL driver supports a wide range of tunable output voltages and operating frequencies to enable compatibility across diverse PIM configurations.

Furthermore, to maintain reliable operation under off-chip interfacing conditions, the driver incorporates a sufficiently robust output buffer capable of preserving pulse integrity in the presence of capacitive loading. By addressing these system-level constraints, the proposed WL driver architecture forms a versatile evaluation platform for input bit-sliced PIM operations across a broad spectrum of emerging memory technologies, offering the flexibility required to satisfy diverse electrical and timing constraints.

VI. CONCLUSION

In this paper, we presented a WL driver specifically designed for PIM systems employing the input bit-slicing method. Fabricated in a TSMC 180-nm CMOS logic process, the proposed driver was verified through measurement results to reliably generate 4-bit binary pulse sequences with high signal integrity. These results confirm that the driver provides the precise and stable WL activation required for robust bit-sliced operations. By addressing system-level voltage, timing, and loading constraints, the proposed WL driver functions as a scalable interface block compatible with diverse synaptic arrays and neuron architectures. Ultimately, the proposed WL driver serves as a robust and scalable foundation for evaluating emerging memory technologies within analog PIM frameworks.

ACKNOWLEDGMENT

This research was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0017011, HRD Program for Industrial Innovation), National R&D Program through the National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT(2021M3H2A1038042), and National R&D Program through the National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT(RS-2023-00217673). The chip fabrication and EDA tool were supported by the IC Design Education Center (IDEC), Korea.

REFERENCES

[1] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, “Memory devices and applications for in-memory computing,” *Nat. Nanotechnol.*, vol. 15, no. 7, pp. 529–544, Jul. 2020.

[2] M. Kim, M. Liu, L. R. Everson, and C. H. Kim, “An embedded NAND flash-based compute-in-memory array demonstrated in a standard logic process,” *IEEE J. Solid-State Circuits*, vol. 57, no. 2, pp. 625–638, Feb. 2022.

[3] E. Choi, I. Choi, V. Lukito, et al., “A 333 TOPS/W logic-compatible multi-level embedded flash compute-in-memory macro with dual-slope computation,” in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, San Antonio, TX, USA, 2023, pp. 1–2.

[4] M. J. Marinella, S. Agarwal, A. Hsia, I. Richter, R. Jacobs-Gedrim, J. Niroula, S. J. Plimpton, E. Ipek, and C. D. James, “Multiscale co-design analysis of energy, latency, area, and accuracy of a ReRAM analog neural training accelerator,” *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 8, no. 1, pp. 86–101, Mar. 2018.

[5] T. P. Xiao, C. H. Bennett, B. Feinberg, et al., “Analog architectures for neural network acceleration based on non-volatile memory,” *Appl. Phys. Rev.*, vol. 7, no. 3, 2020.

[6] S. Kim, Y. Kim, W. Y. Choi, M. Koo, and Y. Kim, “A fully integrated analog processing-in-memory system based on charge-trap flash synapse arrays and successive integration-and-rescaling neurons,” *Adv. Intell. Syst.*, early access, 2025, Art. no. e202501046.

[7] M. Lanuzza, P. Corsonello, and S. Perri, “Low-power level shifter for multi-supply voltage designs,” *IEEE Trans. Circuits Syst.* II, Exp. Briefs, vol. 59, no. 12, pp. 922–926, Dec. 2012.

[8] W. T. Overeem, M. S. Oude Alink, and B. Nauta, “Inverter chain buffer optimization for N-path filter switch drivers and validation through simulations in 22-nm FD-SOI technology,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Monterey, CA, USA, 2023, pp. 1–5.

Sojoong Kim received the B.S. degree in electrical and computer engineering from the University of Seoul, Seoul, South Korea, in 2024, where he is currently pursuing the M.S. degree

Beomjun Park is currently pursuing a B.S. degree in electrical and computer engineering from University of Seoul, Seoul, South Korea.

Jinhyeok Kim received the B.S. and M.S. degrees in electrical and computer engineering from the University of Seoul, Seoul, South Korea, in 2023 and 2025, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the same university.

Yoon Kim received the B.S. and Ph.D. degrees electrical engineering from Seoul National University, Seoul, South Korea, in 2006 and 2012, respectively. From 2012 to 2015, he worked as a Senior Engineer in Samsung Electronics Company Ltd., South Korea. Since 2019, he has been with the Department of Electrical and Computer Engineering, University of Seoul, Seoul, Korea, where he is now a Professor.

Minsuk Koo received the B.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 2007, the M.S. degree in electrical engineering from Seoul National University in 2009, and the Ph. D. degree in electrical and computer engineering from Purdue University in 2020. From 2009 to 2012, he was a Senior Engineer with Radio Pulse Inc., Seoul, South Korea, where he was involved in the development of ZigBee transceiver and SoC products. From 2020 to 2024, he was with the Department of Computer Science and Engineering, Incheon National University, Incheon, South Korea. Since August 2024, he has been with the School of Advanced Fusion Studies and AI Semiconductor, University of Seoul, Seoul, South Korea. His research interests include HW/SW co-design for neuromorphic computing systems, including neural networks.