2024 Design, Automation & Test in Europe Conference (DATE 2024)

X-PIM: Fast Modeling and Validation Framework for Mixed-Signal Processing-in-Memory Using Compressed Equivalent Model in SystemVerilog

Ingu Jeong and Jun-Eun Park*
Department of Electrical and Computer Engineering, *Sungkyunkwan University*, Suwon, South Korea
ingu578@g.skku.edu, juneun.park@skku.edu

***Abstract*— Mixed-signal processing-in-memory (PIM) has gained prominence as a promising approach for implementing deep neural networks in energy-constrained systems. However, the co-design and optimization of mixed-signal circuits in PIM demand substantial time and effort for simulation and validation. This work presents X-PIM, a fast modeling and validation framework for mixed-signal PIMs. X-PIM encompasses not only the precise modeling of transistor-level analog computation circuits in SystemVerilog but also the rapid validation of system-level neural networks implemented using the mixed-signal PIM model. For achieving both accuracy and speed in simulation, X-PIM introduces a technique called compressed equivalent model (CEM) for the mixed-signal PIM circuits. This technique transforms a two-dimensional PIM array into an equivalent single-cell model. Furthermore, X-PIM can account for the impact of non-ideal operations in mixed-signal circuits by incorporating effects such as ADC quantization noise, parasitic components, intrinsic noise, and finite bandwidth. Based on the proposed mixed-signal PIM modeling, X-PIM can perform the system-level neural network validation with a significantly reduced simulation time at least 200 times faster than that of SPICE-based validation. X-PIM demonstrates three mixed-signal PIMs: XNOR PIM, capacitive PIM, and ReRAM-based PIM. For multi-layer perceptron (MLP) network, X-PIM can complete accuracy evaluation for MNIST-1000 dataset within only 30 minutes.**

Keywords—Deep neural network (DNN), framework, modeling, processing-in-memory (PIM), mixed-signal PIM.

I. INTRODUCTION

Revolutionary advancements in deep neural networks (DNNs) have demonstrated outstanding performance in many applications. DNNs provide powerful benefits for realizing artificial intelligence in image classification, pattern recognition, and other intelligent signal processing. However, a hardware implementation of the DNN requires considerable computing resources and energy. To harness the improvements offered by DNNs in energy-constrained devices, it is essential to implement an energy-efficient DNN system. Since the dominant operations in DNNs are multiply and accumulate (MAC), the implementation of an energy-efficient MAC operation becomes one of the crucial key factors in DNN hardware design.

One promising approach to achieving energy-efficient DNN hardware is through a mixed-signal processing-in-memory (PIM) architecture that conducts the MAC operation in the analog domain within embedded memory. The PIM can reduce data movements between processor and memory [1]. Additionally, the column-wise MAC operation in the mixed-signal PIM enables highly parallel computation, improving the throughput of the entire system. Therefore, the mixed-signal PIM architecture can enhance the energy efficiency and throughput of the DNN hardware. Prior works have demonstrated improved energy efficiency and throughput using the mixed-signal PIM [2]-[4].

Although mixed-signal PIMs offer the energy-efficient and high throughput computation, nonidealities of the mixed-signal circuits can degrade DNN performance. For instance, quantization noise introduced by the analog-to-digital converter (ADC) can degrade DNN performance due to information loss. The intrinsic noise of mixed-signal circuits reduces MAC accuracy. Parasitic capacitance can introduce computation errors in capacitive MAC operations. The finite bandwidth, attributed to the parasitic components, constrains the high-speed operation. Therefore, to design optimized DNN hardware, conducting a system-level simulation and validation of the mixed-signal PIM, while considering the nonidealities in computation circuits, is essential.

Unfortunately, conducting system-level simulation and validation of the mixed-signal PIM takes an excessive amount of time, making design optimization challenging. Co-simulation of analog computation circuits and digital logic based on HSPICE requires significantly more simulation time compared to using Verilog simulators for digital PIMs. Mixed-signal PIMs are typically built with large computation arrays to achieve high throughput, leading to an exponential increase in simulation complexity. Consequently, conducting system-level validation and optimization for mixed-signal PIMs becomes challenging using conventional design procedures. Some prior works have investigated the validation and optimization in design process [5]–[6]. A Python-based framework for the capacitive PIM [6] demonstrated DNN-level accuracy evaluation, considering ADC quantization and capacitor mismatch. Furthermore, signal integrity of the PIM cannot be validated using function-level simulation.

This work presents X-PIM, a fast modeling and validation framework for mixed-signal PIMs. Utilizing accurate modeling of transistor-level analog computation circuits in SystemVerilog, X-PIM offers rapid validation for both mixed-signal PIM model and system-level DNN implementation. Event-driven simulation of the mixed-signal PIM model overcomes the traditional trade-off between accuracy and simulation speed in SPICE simulators. Also, X-PIM introduces a compressed equivalent model (CEM) that transforms a two-dimension PIM array model into a single cell model equivalently. The CEM effectively reduces computational complexity for the mixed-signal PIM model, facilitating rapid validation and enabling iterative design optimization. X-PIM performs validation while accounting for the effects of non-ideal operations in mixed-signal circuits, including ADC quantization noise, device mismatches, parasitic components, intrinsic noise, and finite bandwidth. Leveraging the proposed mixed-signal PIM modeling, X-PIM can conduct system-level DNN validation with simulation times significantly reduced, at least 200 times faster than SPICE-based validation. X-PIM demonstrates three mixed-

* Corresponding author: Jun-Eun Park (juneun.park@skku.edu)

979-8-3503-4859-0/DATE24/© 2024 EDAA

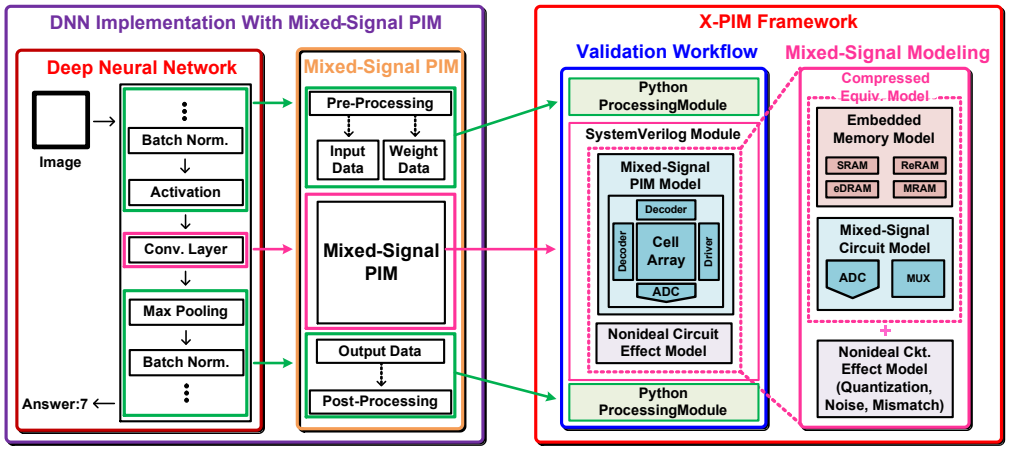


Fig. 1. Overall architecture of proposed X-PIM.

signal PIMs: XNOR PIM, Capacitive PIM, and ReRAM-based PIM. For multi-layer perceptron (MLP) network, X-PIM can complete accuracy evaluation for MNIST-1000 dataset within only 30 minutes. The main contributions of this work are as follows:

- For mixed-signal PIMs, this work presents a framework called X-PIM, which incorporates accurate modeling of mixed-signal computation circuits and enables fast validation of system-level performance using SystemVerilog.
- This work presents a compressed equivalent modeling (CEM), reducing model complexity and simulation time significantly with the assistance of an event-driven simulator. Additionally, nonideal operations of mixed-signal circuits are modeled to improve the validation accuracy.

This paper is organized as follows. Section II explains the overall architecture and workflow of X-PIM. Section III describes the CEM methodology and modeling examples. Section IV presents nonideal circuit models, and Section V explains the system-level validation results. Section VI concludes this work.

II. Modeling and Validation Framework for Mixed-Signal Processing-In-Memory

X-PIM is a framework designed with the goal of validating and optimizing mixed-signal PIMs at the system level. The framework is designed to validate popular DNN structures, including MLP, VGG-like CNN, and ResNet-18, using datasets such as MNIST-1000 and CIFAR-10. Therefore, achieving fast simulation speed and high validation accuracy is imperative for the framework. To achieve this goal, X-PIM consists of two main components: 1) an efficient modeling methodology for mixed-signal PIMs and 2) a rapid validation workflow that combines SystemVerilog and Python for joint simulation as illustrated in Fig. 1.

The modeling part comprises the mixed-signal PIM model and models for nonideal circuit effects. To provide flexible and scalable models for various mixed-signal computing circuits and memories, X-PIM supports not only function-level circuit modeling but also transistor-level circuit modeling in SystemVerilog. X-PIM employs a modeling approach based on the s-domain transfer function representation of circuit operations [7]-[9], allowing for event-driven modeling and simulation of mixed-signal circuits. Using this modeling approach, X-PIM can accommodate modeling of not only digital logic but also mixed-signal circuits in SystemVerilog. To speed up the simulation of mixed-signal PIM models, X-PIM incorporates the CEM technique. Hence, X-PIM conducts the event-driven simulation of the CEM with significantly reduced computation complexity, achieving fast simulation that enables the iterative design optimization. Modeling the nonideal operations and imperfections of mixed-signal circuits enhances simulation accuracy, making it comparable to SPICE simulations. This allows X-PIM to provide reliable validation of mixed-signal PIMs.

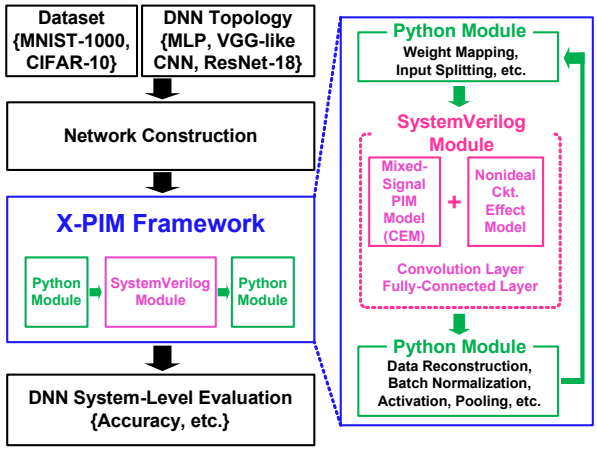


Fig. 2. Workflow diagram of X-PIM.

The validation workflow part consists of a SystemVerilog processing module and a Python processing module. The SystemVerilog module conducts the simulation of circuit models, while the Python module handles the remaining preprocessing and postprocessing in the DNN. Depending on layers in the DNN, X-PIM assigns computations for each layer to either the mixed-signal PIM in the SystemVerilog processing module or the Python processing module. Similarly to other PIM works [2]-[3], the convolution (Conv) layers and fully connected (FC) layers are mapped to the mixed-signal PIM. The remaining layers are computed by the Python processing module.

Fig. 2 illustrates the workflow of X-PIM. This work employed MLP for MNIST-1000 and utilized VGG-like CNN

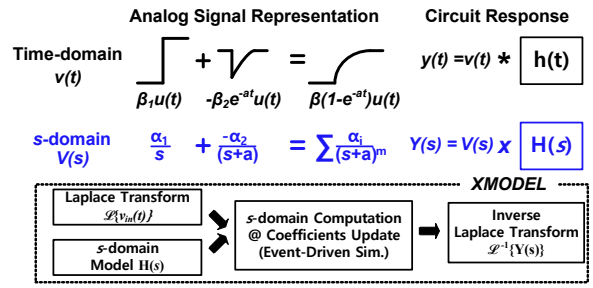


Fig. 3. Conceptual diagram of s-domain modeling and simulation methodology for mixed-signal circuits.

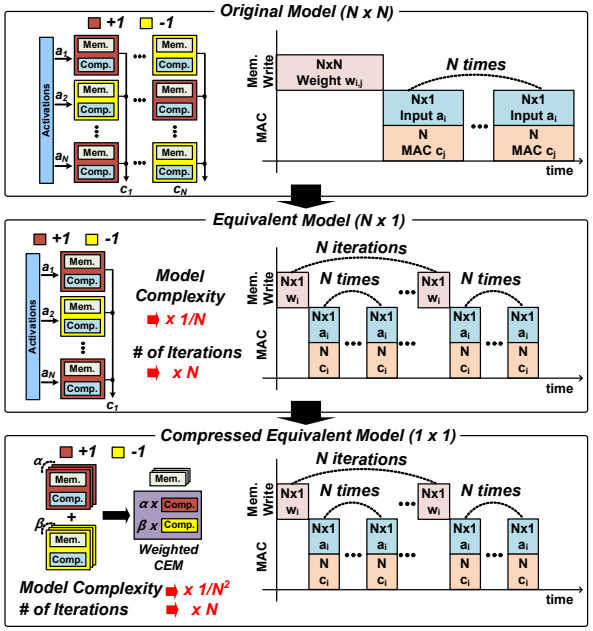


Fig. 4. Transformation of mixed-signal PIM model to compressed equivalent model.

and ResNet-18 for CIFAR-10 classification. Given that the mixed-signal PIM model in X-PIM has a dimension of 256 by 256, the preprocessing stage within the Python module assigns the trained network weights and input feature maps to the mixed-signal PIM model after transformations. Then, the SystemVerilog module conducts the event-driven simulation of the mixed-signal PIM model and the nonideal circuit effect model. The Python module reprocesses the computation results from the SystemVerilog module. After the repetitive operations, X-PIM can evaluate the system-level performance based on the mixed-signal PIM model.

III. Compressed Equivalent Model of Mixed-Signal Processing-in-Memory

A. S-domain Modeling of Mixed-Signal Circuits

X-PIM enhances simulation speed and accuracy by adopting *s*-domain transfer function representations of analog signals. Instead of using a series of time-domain representations of analog signals, X-PIM models analog circuits and signals in *s*-domain functional forms, corresponding to the Laplace transform of the time-domain representation. Fig. 3 shows the conceptual diagram of the *s*-domain modeling and simulation methodology. Circuit responses to analog input signals can be computed through simple multiplication in the s-domain, irrespective of the time step [7][8]. Furthermore, with the assistance of the commercial simulator XMODEL, the s-domain modeling

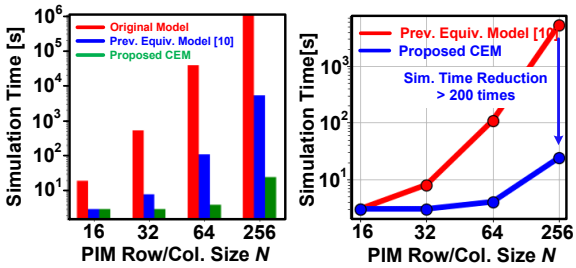


Fig. 5. Simulation time comparison with single-column equivalent model and single-cell compressed equivalent model.

approach can enhance simulation speed through event-driven simulation, computing responses only when coefficients in the transfer function are updated [9].

B. Compressed Equivalent Modeling Methodology

Many mixed-signal PIMs have dimensions of 256 by 256 or even larger, chosen to optimize throughput and energy efficiency. Each cell in the PIM has its own memory circuits and MAC computation circuits, and the overall complexity of the PIM model increases exponentially based on its dimensions. As a result, simulating the entire mixed PIM model requires a significant amount of time, hindering the iterative design optimization. To address the issue, an equivalent modeling approach for memory array was presented in [10]. By leveraging the principle that each column can be independently computed in the PIM, the equivalent modeling method reduces model complexity through iterative simulation of an *N* by 1 single-column model instead of the entire *N* by *N* PIM array model as shown in Fig. 4. The simplified PIM model leads to a reduction in simulation time, as illustrated in Fig. 5. However, for the PIM model with dimensions of 256 by 1, the equivalent modeling approach still exhibited significant simulation time, primarily due to the increased model complexity.

To further enhance the simulation speed of the mixed-signal PIM model, X-PIM incorporates the CEM, which consists of only a single-cell model. Fig. 4 illustrates the principle that transforms the equivalent single-column model into a compressed single-cell model. In the provided equivalent single-column model, the computation circuits within each cell are restructured based on their respective multiplication results. For instance, in the case of a binary neural network (BNN) model [11], the computation circuits within each cell are rearranged by grouping cells with the same multiplication result. Subsequently, the computation circuits can be grouped into two categories for +1 and -1. The computation circuits in each group are transformed into a single-cell circuit through an equivalent weight addition. Since simulation complexity in X-PIM is closely tied to model complexity, the CEM can significantly reduce simulation time for the mixed-signal PIM.

C. Modeling of XNOR PIM

The XNOR PIM described in [2] consists of a 6T SRAM and an XNOR computation circuit that carries out MAC operations. As depicted in Fig. 6, the XNOR PIM can be transformed into the CEM. Depending on the combination of weights in SRAM and input data, the XNOR computation circuits in each accumulation bitline can be transformed into a pull-up PMOS array and a pull-down NMOS array. The multiplication results in each cell are tallied for both +1 and -1, with each result assigned to the weights of the pull-up and

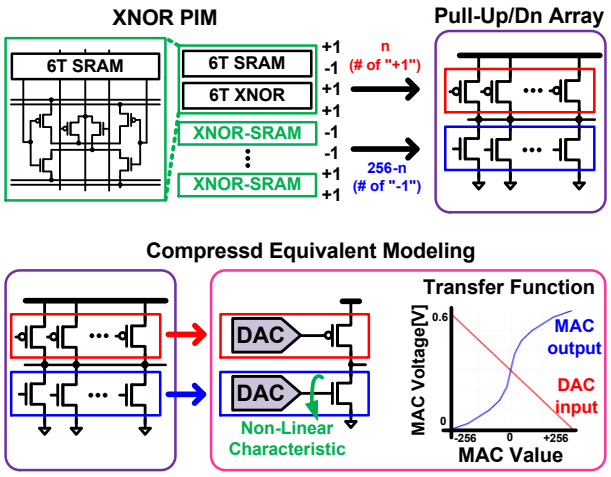


Fig. 6. Compressed equivalent model for XNOR PIM.

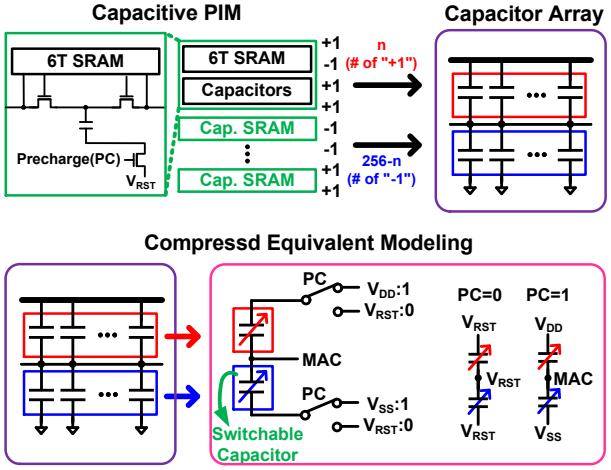


Fig. 7. Compressed equivalent model for capacitive PIM.

pull-down devices in the CEM. Additionally, the transistor models incorporate the nonlinear current-voltage (I-V) characteristics of NMOS and PMOS. Therefore, the CEM of the XNOR PIM not only offers fast simulation speeds but also provides accurate results that account for the nonlinear MAC characteristics.

D. Modeling of Capacitive PIM

The capacitive PIM in [3] achieved MAC computation through the redistribution of charge among capacitors. The capacitive PIM consists of 6T SRAM and coupling capacitors that perform the MAC operation. Fig. 7 shows the CEM of the capacitive PIM. Each column in the capacitive PIM can model the MAC bitline (MBL) using capacitor arrays that share the MBL between the reset voltage V_{RST} and V_{SS}. Similar to the XNOR PIM, the capacitor arrays can be divided into two groups based on the multiplication results in each cell. The capacitor of each cell is discharged or charged depending on the multiplication results. During MAC operation, all capacitors are connected and the voltage is divided. These two groups can be compressed into two capacitors, each with capacitance corresponding to the parallel combination of the capacitors within its group.

E. Modeling of ReRAM-Based PIM

The ReRAM-based PIM in [4] can be implemented using multiple 1-transistor and 1-resistor (1T1R) ReRAM models. Depending on the stored weight data, ReRAM can operate in a low-resistance state (LRS: 1) or a high-resistance state (HRS: 0). ReRAM exhibits hysteresis during transitions between LRS and HRS [12]. These characteristics of ReRAM can be modeled using variable resistors and a Schmitt trigger, as described in [13]. ReRAM's resistor consists of two parts: one has a different resistance value for each section depending on the voltage difference between the two plates, and the other has a different resistance value when the cycle is repeated. By the variation cell described in Verilog and variable resistor, the piece-wise variation and switching variation of resistance can be considered in ReRAM-based PIM. Based on the ReRAM model, the ReRAM-based PIM can be simplified with resistor array as shown in Fig. 8. In each column of the PIM, the numbers of LRS and HRS are counted among activated cells based on input data. Using these counts, the current sensing lines in each column can be modeled with two resistors representing the parallel combinations of LRS resistors and HRS resistors, respectively. As a result, the ReRAM-based PIM can be simplified to just two resistors, significantly reducing model complexity.

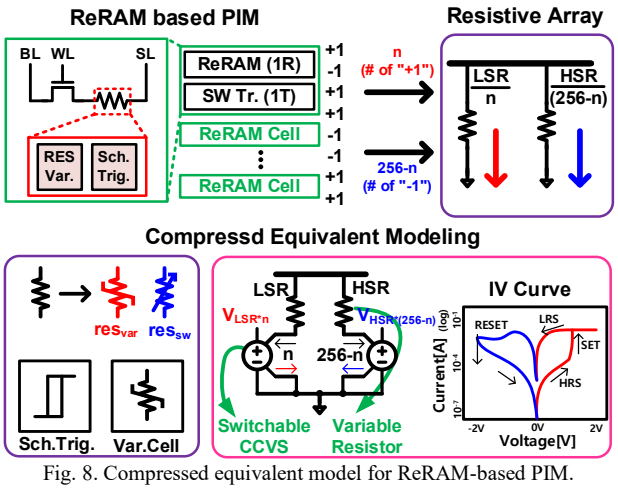


Fig. 8. Compressed equivalent model for ReRAM-based PIM.

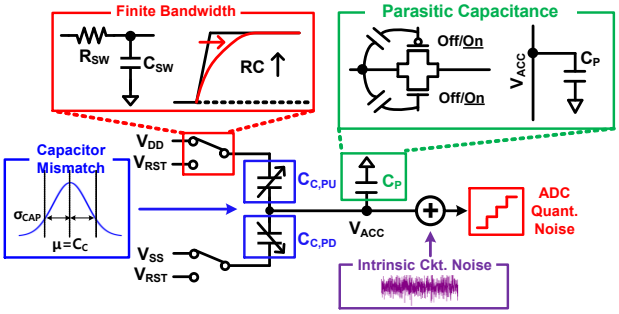


Fig. 9. Modeling example of nonideal operations and imperfections in the capacitive PIM model.

IV. Modeling of Nonidealities in Mixed-Signal PIM

To ensure the accurate validation of mixed-signal PIMs, it is essential to consider the nonideal operations and imperfections of the mixed-signal circuits. Several nonideal factors can impact the system-level performance of the PIM, including ADC quantization, parasitic capacitance and resistance, mismatches, intrinsic noise, and finite bandwidth. X-PIM can incorporate the influence of these nonideal factors into the validation process as shown in Fig. 9.

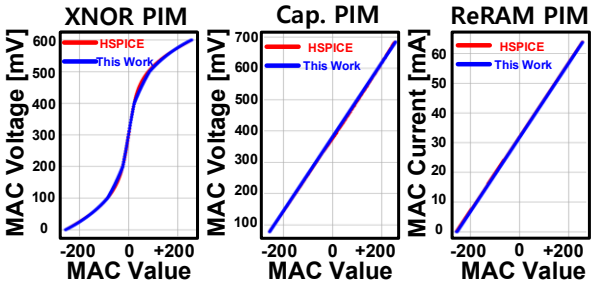
A. ADC Quantization Noise and Circuit Intrinsic Noise

X-PIM incorporates various ADC structures commonly used in PIMs, including flash ADCs and successive-approximation register (SAR) ADCs. The ADC models offer

TABLE I. NETWORK TOPOLOGY

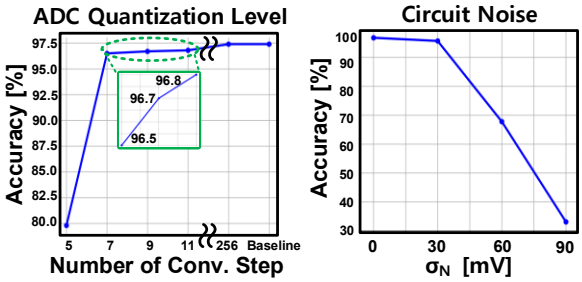
	MNIST	CIFAR-10	
Neural Network	*MLP*	*VGG-like CNN*	*ResNet-18*
Topology	784FC-512FC-512FC-512FC-10FC	128C3-128C3-MP2-256C3-256C3-MP2-512C3-512C3-MP2-1024FC-1024FC-10FC	80C3-80C3-80C3-80C3-160C3-160C3-160C3-160C3-320C3-320C3-320C3-320C3-AP8-10FC

[a] *nCk*-Convolutional layer with *k by k* kernel and *n* filters, *m*FC-Fully connected layer with *m*-neuron, MP*p* – Max-pooling layer with *p by p* pooling size, AP – Average-pooling layer with *p by p* pooling size



(a) (b) (c)

Fig. 10. Simulated MAC results of the CEMs for (a) XNOR PIM and (b) capacitive PIM (c) ReRAM-based PIM.



(a) (b)

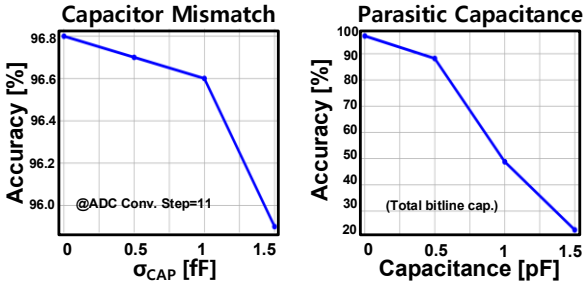
Fig. 11. MNIST-1000 classification accuracies depending on (a) ADC resolution and (b) intrinsic noise.

configurable design parameters such as the full-scale voltage V_{FS} for the conversion range and the conversion bit resolution N_{ADC}. Therefore, the impact of ADC quantization on the system-level performance of the PIM can be readily assessed by adjusting the ADC model parameters.

While intrinsic noise models of the computation circuits in the mixed-signal PIM can be individually modeled in each cell, a distributed noise model cannot fully leverage the advantages of the CEM. Since the noise signals in each cell are integrated in an accumulation bitline corresponding to the ADC input, this work combines the noise contributions from each computation circuit into a compressed model positioned in front of the ADC. The compressed noise model can inject a Gaussian random signal into the ADC input with a configurable standard deviation denoted as σ_N.

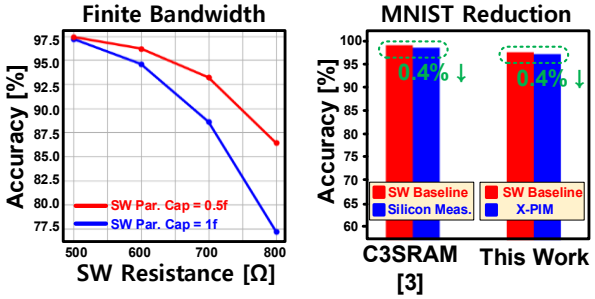
B. Parasitic Capacitance and Capacitor Mismatch

When it comes to capacitive PIMs, parasitic capacitances in each computation circuit and the accumulation bitline can impact the accuracy of the DNN. To account for the parasitic components, X-PIM can incorporate configurable parasitic capacitors into the accumulation bitline. Furthermore, mismatches between coupling capacitors can lead to a degradation in MAC accuracy. The capacitance mismatches



(a) (b)

Fig. 12. MNIST-1000 classification accuracies depending on (a) capacitor mismatch and (b) parasitic capacitance.



(a) (b)

Fig. 13. (a) MNIST-1000 classification accuracies versus finite bandwidth due to parasitic components of switches. (b) MNIST accuracy comparison with [3].

are modeled by perturbing the coupling capacitance in the CEM, following a Gaussian distribution.

C. Finite Bandwidth

Parasitic resistance and capacitance in the computation circuits can limit the bandwidth of MAC operations, which can degrade DNN accuracy. The time delay of control signals caused by finite bandwidth affects the signal integrity of the PIM. Thanks to the *s*-domain representation modeling in X-PIM, the influence of finite bandwidth is inherently included in the validation process. For example, in Fig. 9, the combination of parasitic capacitance and resistance of switch acts as a low-pass filter, thereby limiting the bandwidth of the accumulation.

V. Simulation and Validation Results

This work implemented three mixed-signal PIM models: XNOR PIM, capacitive PIM, and ReRAM-based PIM. Using the PIM models, X-PIM demonstrated MLP for the MNIST 1000 dataset, VGG-like CNN for the CIFAR-10 dataset, and ResNet-18 for the CIFAR-10 dataset. Table I shows the detailed network structures. Simulation and validation were performed using a 3.2GHz Intel Xeon processor with 128GB of main memory. Fig. 10 shows a comparison of simulated MAC results with HSPICE simulations for the CEMs of XNOR PIM, capacitive PIM and ReRAM-based PIM [2] – [4]. The CEMs exhibited accurate matching properties with the nonlinear I-V characteristic observed in the HSPICE simulation result.

To demonstrate the validation and design optimization processes using X-PIM, this work conducted system-level evaluations of the capacitive PIM model. Fig 11(a) shows the simulated classification accuracies of MNIST-1000 with MLP. The classification accuracy improves as the number of ADC quantization levels increases. At the eight-bit quantization, which is the same as the baseline quantization, the simulation

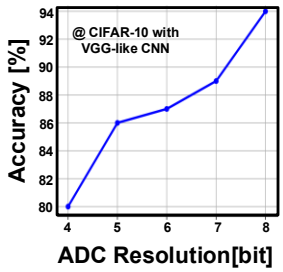
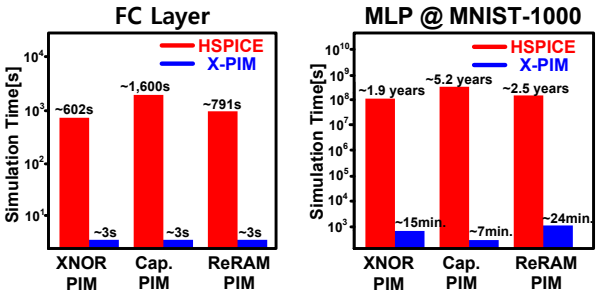


Fig. 14. CIFAR-10 with VGG-like CNN classification accuracies depending on ADC resolution.



(a) (b)

Fig. 15. Simulation time comparison between HSPICE and X-PIM for (a) FC layer computation and (b) MLP validation.

results show closely matched accuracy. Fig 11(b) shows the accuracy dependency on the intrinsic noise at the accumulation bitline or ADC input. Fig. 12 shows the accuracy degradations by coupling capacitance mismatches and by parasitic capacitance at the accumulation bitline. The finite bandwidth caused by parasitic resistance and capacitance also affects the accuracy, as depicted in Fig. 13(a). When compared with the classification accuracy results in [3], X-PIM demonstrates reliable validation accuracy, as shown in Fig. 13(b). X-PIM demonstrated not only MNIST-1000 with MLP but also CIFAR-10 with VGG-like CNN and ResNet-18. Fig. 14 shows the CIFAR-10 classification with VGG-like CNN depending on the ADC resolution. Since the VGG-like CNN was trained using 8-bit ADC resolution, lower ADC resolution degrades the classification accuracy.

In Fig. 15(a), the simulation times of three PIM models for a single fully-connected layer computation are presented. X-PIM achieves a simulation speed that is at least 200 times faster compared to HSPICE. Assuming a validation of MNIST-1000 with MLP, the estimated HSPICE simulations would take more than two years, whereas X-PIM can complete the validation within 30 minutes, as illustrated in Fig. 15(b). Table II provides a summary of the simulation time of X-PIM for different PIM structures and DNN configurations.

VI. Conclusion

This work presents a fast modeling and validation framework X-PIM for mixed-signal PIMs. This framework comprises compressed equivalent modeling of mixed-signal PIMs and the rapid validation of a system-level DNN implemented using the PIM model. Furthermore, X-PIM incorporates models for non-ideal operations in mixed-signal circuits, enhancing validation accuracy. X-PIM enables system-level DNN validation using mixed-signal PIM models with popular DNNs, significantly reducing validation time by at least 200 times compared to HSPICE. Therefore, X-PIM offers rapid design space exploration and optimization capabilities, facilitating the development of energy-efficient PIMs.

TABLE II. COMPARISON OF SIMULATION TIME

	MNIST			**CIFAR-10**	
Neural Network	***MLP***			***VGG-like CNN***	***ResNet-18***
Number of Data	***1000***			***100***	
PIM Model	***XNOR PIM***	***ReRAM PIM***	***Capacitive PIM***		
*HSPICE Sim. Time	17125h (~1.9y)	22501h (~2.5y)	45515h (~5.2y)	> 10y	> 10y
X-PIM Sim. Time	**15m**	**23m 48s**	**6m 57s**	**18h 3m**	**13h 32m**

* *Estimated simulation time based on single FC layer simulation time.*

Acknowledgment

This work was supported in part by Scientific Analog, Inc and by Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korea government(MOTIE) (No. RS-2022-00144290). The EDA tool was supported by the IC Design Education Center(IDEC), Korea.

References

[1] N. Verma et al., "In-Memory Computing: Advances and Prospects," in *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 43-55, Aug. 2019.

[2] S. Yin et al., "XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks," in *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1733-1743, June 2020.

[3] Z. Jiang et al., "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," in *IEEE Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1888-1897, July 2020.

[4] X. Sun et al., "XNOR-RRAM: A scalable and parallel resistive synaptic architecture for binary neural networks" in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1423-1428, Apr. 2018.

[5] J. Wang et al., "eF2lowSim: System-Level Simulator of eFlash-Based Compute-in-Memory Accelerators for Convolutional Neural Networks," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-6, Apr. 2023.

[6] J. Saikia et al., "Modeling and Optimization of SRAM-based In-Memory Computing Hardware Design," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 942–947, Feb. 2021.

[7] J. Jang et al., "True event-driven simulation of analog/mixed-signal behaviors in SystemVerilog: A decition-feedback equalizaing(DFE) receiver example," in *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference (CICC)*, pp. 1-4, Sep. 2012.

[8] J. Jang et al., "An event-driven simulation methodology for integratred switching power supplies in SystemVerilog," in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–7, May 2013.

[9] Scientific Analog, Inc., XMODEL Reference Manual, 2021.

[10] S. Kim et al., "An Equivalent Modeling Approach for High-Density DRAM Array System-Level Design-Space Exploration in SystemVerilog," *Design and Verification Conference and Exhibition (DVCon)*, pp. 1-11, Apr. 2021.

[11] I. Hubara et al., "Binarized neural networks," in *Neural Information Processing Systems Conference*, pp. 4107-4115, Dec. 2016.

[12] J. Reuben et al., "A Modeling Methodology for Resistive RAM Based on Stanford-PKU Model With Extended Multilevel Capability," *IEEE Transactions on Nanotechnology*, vol. 18, pp. 647–656, June 2019.

[13] I. Jeong et al., "A Modeling and Simulation of ReRAM with Nonidealities for System-Level PIM Validation in SystemVerilog," *International Conference On Consumer Electronics Asia (ICCE-Asia)*, pp. 1-4, Oct. 2023.