SCIENCE CHINA
Information Sciences

• LETTER •

December 2023, Vol. 66 229402:1–229402:2
https://doi.org/10.1007/s11432-022-3756-3

CLEAR: a full-stack chip-in-loop emulator for analog RRAM based computing-in-memory system

Ruihua YU, Wenqiang ZHANG, Bin GAO*, Yiwen GENG, Peng YAO, Yuyi LIU, Qingtian ZHANG, Jianshi TANG, Dong WU, Hu HE, Ning DENG, He QIAN & Huaqiang WU

School of Integrated Circuits, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084*, China*

Received 3 December 2022/Revised 6 February 2023/Accepted 27 April 2023/Published online 18 August 2023

Citation Yu R H, Zhang W Q, Gao B, et al. CLEAR: a full-stack chip-in-loop emulator for analog RRAM based computing-in-memory system. Sci China Inf Sci, 2023, 66(12): 229402, https://doi.org/10.1007/s11432-022-3756-3

Deep neural networks (DNNs) have undergone tremendous advancement in several practical applications, such as object detection and text generation, over the last few years, leading to a huge demand for increased computing efficiency of hardware. Resistive random-access memory (RRAM)-based computing-in-memory (CIM) chip has been investigated as a promising candidate to reduce the requirement for memory bandwidth [1]. CIM chips can conduct multiply-accumulate operations natively in memory units employing physical laws. Recent experimental demonstrations [2–4] and architectural designs [5] have revealed the considerable potential of RRAM-based CIM chips in highly energy-efficient acceleration for DNNs.

However, two problems were ignored by previously found simulation tools, which affected the development of CIM chips. First, owing to the lack of compiler optimization in the simulation tools, the deployment of DNN models on the CIM chip was difficult. Second, owing to the lack of unified representation between the DNN models and CIM hardware or simulator, the ways to obtain the outcome from the simulator and real chip were separate, considerably increasing the cost of validation between the simulator and real chip.

To overcome these restrictions, we introduce CLEAR, a full-stack chip-in-loop emulator for analog RRAM-based CIM System. It featured a proposed general intermediate representation (IR) to unify the computing flow on the real chip and simulator, two proposed compiler optimization techniques to exploit the balanced allocation of on-chip resources, and three different CIM chip emulation support levels.

CLEAR architecture. CLEAR is an end-to-end full-stack tool that consists of three hierarchical modules: a customized chip-aware training framework, compiler, and chip-in-loop emulator (Figure 1). In CLEAR, the training framework first optimizes the neural network under the constraints of hardware, including quantization and noise of the weights. After the neural network is trained, the model is sent to the compiler to be parsed and optimized depending on the operational constraints and total on-chip resources. CLEAR employs unified IR, named emulation-oriented IR, to represent the computing flow both on the real chip and the simulator. According to the IR, the compiler can generate executable codes using different backends (real chip or analog computing model) to calculate the results of DNNs. When a single operation is scheduled to run virtually, the compact computing model in the emulator is utilized to simulate the output and circuit metrics. When this operation is performed on actual hardware, the weights of this operation are programmed into the RRAM chip, and the results are calculated on-chip. The outputs of CLEAR include the simulated and chip-in-loop accuracy, circuit metrics, and throughput of the employed neural network. The analog circuit model can be calibrated based on the on-chip results under the same workflow, and it only needs to select the proper address in the IR.

Compiler design. The compiler plays a critical role in optimizing DNN model structures and model deployment. The compiler converts the trained model structure to a CIM-friendly structure using operator fusion and split techniques, and optimizes the model deployment with a critical path-reforming method. Detailed information about operator fusion and the critical path-reforming method can be seen in Appendix A. Before deploying the DNN models, the IR keeps the DNN model's information for the compiler frontend. Following deployment, the hardware address having two selections—the real chip address and virtual address, will be added into the IR for the compiler backend. Please refer to Appendix B for more information about IR.

Analog computing model. We integrated different computing models for inference and training in CLEAR, and it can simulate three different levels—array level, macro level, and system level. For the inference phase, the computing outcome is mainly influenced by read and write noises and IR drop. For the update phase, the next conductance state

* Corresponding author (email: gaob1@tsinghua.edu.cn)

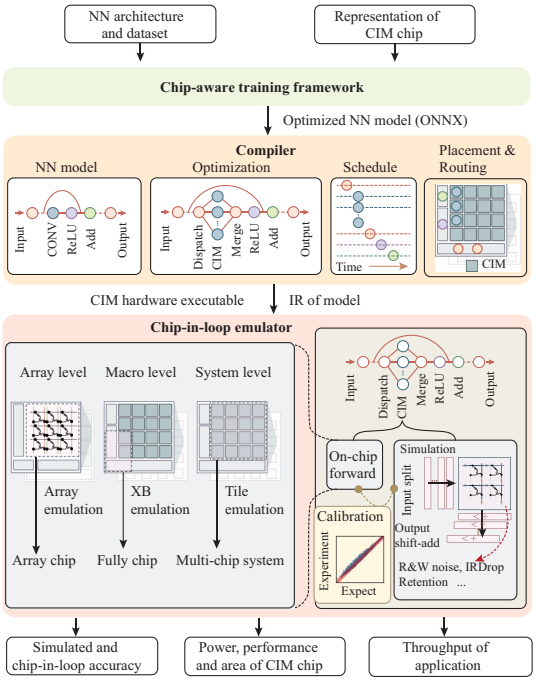

Figure 1 (Color online) Architecture of CLEAR.

following the open-loop updating without verification is related to the update pulse number and the minimum and maximum values of the analog switching window. For the close-loop update with verification, the next conductance state is modeled as the target value plus a Gaussian distribution noise. Please refer to Appendix C for more information. We used real chip results to calibrate the proposed model. The experimental outcome revealed that the suggested emulator can simulate the system considerably well, and the outcome errors between the emulator and chip can be as low as 0.17% and 0.36% in the inference and training, respectively.

Experimental results. We studied three cases—dataflow optimization, analog computing verification, and model calibration—with the proposed emulator. Owing to page limitations, more detailed experimental results and comparisons are provided in Appendixes D and E, respectively.

Conclusion. In this study, we propose and implement a full-stack chip-in-loop emulator for an RRAM-based CIM chip to simulate the realistic runtime environment of diverse DNNs. The compiler can optimize the on-chip dataflow based on the critical path-reforming method for various DNN models. The emulator employed three kinds of chips to emulate the accuracy and performance of CIM chips.

Acknowledgements This work was supported in part by STI 2030-Major Projects (Grant No. 2021ZD0201205), National Natural Science Foundation of China (Grant Nos. 92064001, 62025111), Beijing Advanced Innovation Center for Integrated Circuits, and XPLORER Prize.

Supporting information Appendixes A–E. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zhang W Q, Gao B, Tang J, et al. Neuro-inspired computing chips. Nat Electron, 2020, 3: 371–382
2 Yao P, Wu H Q, Gao B, et al. Fully hardware-implemented memristor convolutional neural network. Nature, 2020, 577: 641–646
3 Zhang W Q, Gao B, Yao P, et al. Array-level boosting method with spatial extended allocation to improve the accuracy of memristor based computing-in-memory chips. Sci China Inf Sci, 2021, 64: 160406
4 Zhou Y, Gao B, Zhang Q, et al. Application of mathematical morphology operation with memristor-based computation-in-memory architecture for detecting manufacturing defects. Fundamental Res, 2022, 2: 123–130
5 Chang L, Li C L, Zhang Z M, et al. Energy-efficient computing-in-memory architecture for AI processor: device, circuit, architecture perspective. Sci China Inf Sci, 2021, 64: 160403