ENASA: Towards Edge Neural Architecture Search based on CIM acceleration

Shixin Zhao[1,2], Songyun Qu[1,2], Ying Wang*[,1], Yinhe Han[1]
Institute of Computing Technology, Chinese Academy of Sciences[1]
University of Chinese Academy of Sciences[2]
Beijing, China
Email: {zhaoshixin22s, qusongyun18z, wangying2009, yinhes}@ict.ac.cn[1]

***Abstract*—This work proposes a ReRAM-based Computing-in-Memory (CIM) architecture for Neural Architecture Search (NAS) acceleration, ENASA, so that the compute-intensive NAS technology can be applied to various edge devices to customize the most suitable individual solution for their cases. In the popular one-shot NAS process, the system must repetitively evaluate the sampled sub-network within a large-scale supernet before converging to the best sub-network architecture. Thereby, how to map these iterative network inference tasks onto the CIM arrays makes a big difference in system performance. To realize efficient in-memory supernet sampling and evaluation, we design a novel mapping method that tactically executes a group of sub-nets in the CIM arrays, not only to boost the sub-net concurrency but also to eliminate the repetitive operations shared by these subnets. Meanwhile, to further enhance the subnet-level operation concurrency and sharing in the CIM arrays, we also tailor a novel CIM-friendly one-shot NAS algorithm that purposely samples those operation-sharing subnets in each iteration while still maintaining the convergence performance of NAS. According to the experimental results, our CIM NAS accelerator achieves an improvement of 196.6× and 1200× in performance speedup and energy saving respectively compared to the CPU+GPU baseline.**

***Index Terms*—CIM, DNNs, NAS, ReRAM**

I. Introduction

One-shot NAS [1] is a popular technology aiming at the automatic search for high-performance neural networks. In one-shot NAS, a supernet containing all candidate architectures is trained only once, so the architecture sampled from it can inherit the pre-trained supernet parameters [2], [3] before being evaluated on tasks. Nevertheless, finding the optimal network for tasks still takes hundreds of GPU hours [1]. The computation overhead prevents NAS from being applied to the edge and terminal devices, which are supposed to customize the best network architecture for their application scenarios. One of the major sources that cause tedious one-shot NAS time is the frequent dynamic network switches and memory access.

CIM is an emerging technology to overcome the well-known memory wall problem. [4] It carries out in-situ multiply-accumulate(MAC) operation efficiently [5], [6], which is ideal for NN acceleration. For NAS acceleration on edge, we can preload the supernet weights on the CIM arrays to reduce weight re-access. However, most of the current CIM NN accelerators are dedicated to single or batched neural network acceleration and they have problems supporting an efficient NAS search process. First, most CIM mapping strategies do not optimize the online mapping process for the networks generated dynamically. Second, prior scheduling methods for CIM do not explore the intermediate computing result-sharing opportunity between multiple sub-networks. Lastly, the current NAS search strategy is not aware of the CIM architecture, and it will generate sub-networks with poor locality and result in low utility in the CIM arrays.

Based on the above observation, we propose ENASA, an efficient NAS accelerator based on CIM architecture and evaluate the proposed architecture on the ReRAM-based CIM design.

Specifically, this work makes the following contributions:

- We develop ENASA, a CIM-based accelerator to accelerate the one-shot NAS search process and enable secure and lightweight network customization on edge devices.
- We design a novel CIM architecture for NAS support and also the mapping algorithm for efficient NAS deployment on CIM crossbars.
- We propose a CIM-friendly stage-wise network search strategy.

II. CIM-based Efficient NAS Accelerator

A. workflow

Fig. 1 shows the workflow of ENASA. We divide the acceleration process into three parts that are respectively CIM-oriented network generation, Crossbar mapping, and CIM evaluation.

CIM-oriented Network generation gears the evolutionary algorithm towards the network architectures that produce the most network weight replication chances on the ReRAM arrays for higher CIM computation throughput while still keeping the convergence performance of the NAS algorithm. As shown in Fig.1 part I, for the CIM-friendly networks generated by our improved stage-wise search strategy that alternately fixes the top and bottom layers of sub-nets during network generation, the CIM accelerators have a bigger chance of higher-level computation parallelism.

After network generation, we introduce a novel mapping method to further promote the sub-network level concurrency to enhance the CIM crossbar array utility. Specifically, we employ the operator-parallel scheduler to achieve fast online subnetwork mapping on crossbars and boost the execution concurrency of subnetworks generated dynamically. As Fig. 1 part II shows, we will schedule all schedulable operators if

*Corresponding author

978-3-9819263-7-8/DATE23/© 2023 EDAA

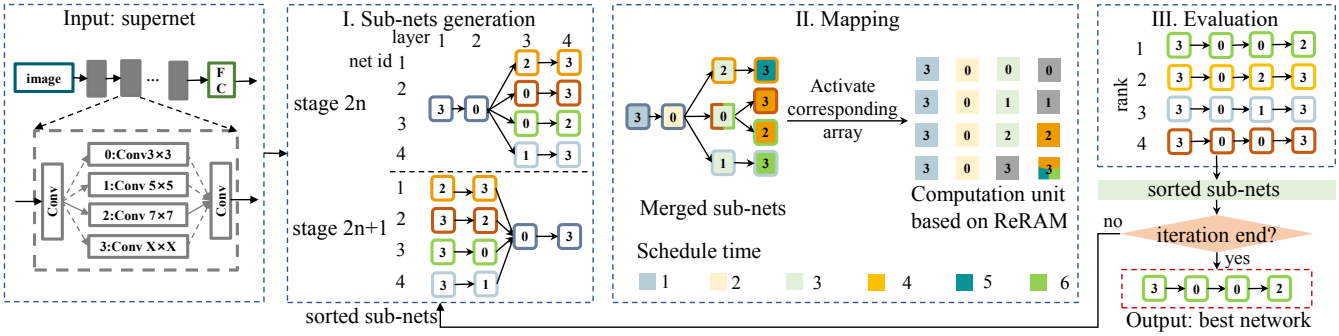

Fig. 1: Accelerator workflow overview. Numbers in the computation unit mean those PEs preload the weight of the corresponding operations.

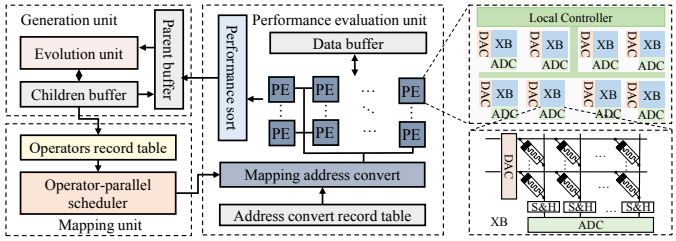

Fig. 2: Overview of ENASA

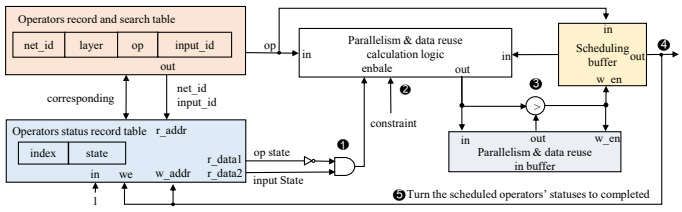

Fig. 3: Operator-parallel scheduler

sufficient crossbars are unoccupied. However, when the chosen operators conflict, we will rank the combinations of operators. Different operators from one layer and the parallel operators that share the same input will be prioritized together to boost sub-net concurrency while reducing the memory access time.

Finally, we obtain the performance of all sampled networks on the given tasks through CIM evaluation.

B. Hardware Implementation

As Fig.2 shows, ENASA includes a CIM-oriented network generation unit, a mapping unit, and a CIM evaluation unit according to the workflow.

The network generation unit efficiently performs the stage-wise based evolutionary algorithm to produce new networks. The generation controller randomly selects two networks in the parent buffer, performs the genetic operation on them, and replaces half the layers with the corresponding ones of the current best-performing sub-net.

Then, the on-chip mapping unit optimizes the operators' mapping order and accelerates the mapping process. As Fig. 3 shows, the operator-parallel scheduler (OPS) completes the hardware implementation of the parallel scheduling algorithm, which mainly includes an operator status register, a scheduling operators buffer, the arithmetic logic of operator parallel and data reuse score and the judgment logic.

For higher operator-level parallelism and fast address conversion, the CIM evaluation unit uses a hierarchical ReRAM-based architecture similar to [5]

III. Evaluation

We evaluate ENASA on two one-shot NAS frameworks, SPOS [1] and FairNAS [2]. We extract 10000 images from ImageNet [7] as our test set. We use the DNN_NeuroSim_v1.3 simulator [8] to measure the ReRAM arrays and the system-level performance of the ENASA. For the other logic components, we implement them in RTL and synthesize it by the Synopsys design compiler with the TSMC 22nm technology. The result shows that compared with GPU-CPU systems, our accelerator achieves 196.6× in performance speedup and 1200× in energy consumption while 6.5× and 9× compared with state-of-the-art specialized NAS accelerator design [9]. This improvement is attributed to OPS, which optimizes the dynamic network mapping process and fully explores the sub-net level parallel. As CIM can preload the weights on the independent crossbars, stage-wise search and parallel scheduling allow maximum utilization of the crossbars, actually more than 90%.

References

[1] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *Proc. of ECCV*, vol. 12361, 2020, pp. 544–560.

[2] X. Chu, B. Zhang, and R. Xu, "Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search," in *Proc. of ICCV*, 2021, pp. 12 219–12 228.

[3] Y. Xu, L. Cheng, X. Cai, X. Ma, W. Chen, L. Zhang, and Y. Wang, "Efficient supernet training using path parallelism," in *Proc. of HPCA*, 2023.

[4] S. A. McKee, "Reflections on the memory wall," in *Proceedings of the 1st conference on Computing frontiers*, 2004, p. 162.

[5] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu *et al.*, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proc. of ISCA*, 2016, pp. 14–26.

[6] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu *et al.*, "PRIME: A novel processing-in-memory architecture for neural network computation in reram-based main memory," in *Proc. of ISCA*, 2016, pp. 27–39.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[8] X. Peng, S. Huang, Y. Luo, X. Sun, and S. Yu, "Dnn+ neurosim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies," in *2019 IEEE international electron devices meeting (IEDM)*. IEEE, 2019, pp. 32–5.

[9] X. Ma, C. Si, Y. Wang, C. Liu, and L. Zhang, "NASA: accelerating neural network design with a NAS processor," in *Proc. of ISCA*, 2021, pp. 790–803.