

Microscaling–Stochastic Computing Based Systolic Arrays for Energy-Efficient Deep Neural Network Inference

Mohammad Hassani Sadi, Bilal Hammoud, and Norbert Wehn

Microelectronic Systems Design Research Group, RPTU Kaiserslautern-Landau, Kaiserslautern, Germany

Emails: {m.sadi, bilal.hammoud, norbert.wehn}@rptu.de

Abstract—Deep neural networks (DNNs) require increasingly high compute and memory resources. Microscaling (MX) data formats improve energy efficiency and preserve accuracy under aggressive bit-width reduction, but further gains from continued bit-width reduction remain challenging. This work proposes a hybrid computation scheme that integrates MX data formats with stochastic computing (SC) to improve the energy efficiency of DNN inference under constrained bit widths. Model parameters are stored in MX format, while multiplications and accumulations are performed in the SC domain. MX reduces memory footprint, while SC improves compute energy efficiency. To address the latency and accuracy challenges of SC, we employ a parallel bitstream-generation scheme and an encoding strategy that reduces random fluctuation error. Experimental results demonstrate up to a $2\times$ improvement in energy efficiency while maintaining inference accuracy within 1–2% of an FP32 baseline.

I. INTRODUCTION

Low-precision formats such as INT8, FP8, and BFP16 reduce the computational demands and memory footprints of deep neural networks (DNNs) [1]. Microscaling (MX) formats [2] further improve efficiency by applying block-wise scaling and shared exponents, enabling inference at bit widths as low as 4–6 bits. However, further improving energy efficiency by reducing the bit width of these formats becomes increasingly difficult without compromising accuracy.

To address this limitation, we introduce a hybrid computation scheme that improves energy efficiency at a fixed bit width. In this approach, MX data formats are used to store DNN parameters, while stochastic computing (SC) is employed for the core arithmetic operations. SC enables highly energy-efficient arithmetic implementations, as multiplication can be realized using a single AND gate. Despite this simplicity, SC suffers from long stochastic bitstreams, correlation errors, and random number generation overheads. As a result, prior SC accelerators typically rely on full model unrolling to avoid low throughput. We propose a systolic array architecture tailored to the proposed hybrid computation scheme. To mitigate the inherent latency of SC within the systolic array, we incorporate parallel bitstream generation schemes and

study the impact of different parallelization factors on energy efficiency. Unlike conventional monolithic SC accelerators that hardwire entire DNN models into hardware, the proposed architecture is modular and scalable. It amortizes the overhead of stochastic number generation by allowing multiple processing elements (PEs) to share stochastic number generators (SNGs). Furthermore, we introduce a novel linear feedback shift register (LFSR) architecture that improves computational accuracy.

II. PROPOSED HYBRID MX–SC METHOD

We employ an MX format consisting of a 6-bit signed mantissa, an 8-bit shared exponent, and an 8-bit linear scale factor per block. Only operations on mantissa are mapped to SC; exponent and scale computations occur once per block using binary arithmetic.

A. Stochastic Encoding

We employ unipolar stochastic bitstreams, as bipolar encoding maps zero to 0.5, which corresponds to the maximum representation error [3]. Since DNN parameters typically follow a normal distribution centered around zero, bipolar SC leads to higher representation error near the most frequently occurring values. Unipolar SC represents values in the range $[0, 1]$. Signed MX mantissas are mapped to the unipolar SC domain by handling the sign bit separately from the magnitude. This separation reduces the effective mantissa width to 5 bits, allowing all 2^5 representable magnitude values to be encoded using a bitstream length of $L = 32$. This is half the bitstream length required by bipolar SC implementations, which require $L = 64$.

B. Hybrid Computation Flow

Fig. 1 illustrates the MX–SC computation flow: mantissas are converted to SC bitstreams through shared stochastic number generators (SNGs), multiplied using AND gates, accumulated in binary, and rescaled after stochastic-to-binary conversion.

III. HARDWARE ARCHITECTURE

Fig. 2 illustrates the proposed hardware architecture. Each row and column of the systolic array shares a SNG, which converts MX mantissas into stochastic bitstreams and supplies

This paper was supported by European Union’s Horizon Europe research and innovation programme (HORIZON-CL4- 2021-HUMAN-01) under grant agreement No 101070408, project SustainML. Further, this work was also supported by Carl-Zeiss Stiftung under the Sustainable Embedded AI project (P2021-02-009) .

TABLE I
CLASSIFICATION ACCURACY (%) RESULTS

Dataset / Model	FP32	MX+Binary	MX+SC
CIFAR-10 / RN18	93.0	93.0	92.2
CIFAR-100 / RN18	71.0	70.5	68.4
SVHN / RN18	94.4	94.4	94.0
ImageNet / RN18	71.0	70.0	67.0

TABLE II

COMPARISON OF AREA, POWER, THROUGHPUT, AND ENERGY EFFICIENCY FOR 32×32 SYSTOLIC ARRAY WITH DIFFERENT PARALLELIZATION FACTORS(P)

Design	Area [mm ²]	Power [mW]	Throughput [GOPS]	Eff. [GOPS/W]
MX Binary	0.630	140	125	892
MX-SC $P = 1$	0.345	26	14.6	560
MX-SC $P = 2$	0.353	27	26.8	992
MX-SC $P = 4$	0.372	30	46.8	1560
MX-SC $P = 8$	0.413	40	71.8	1795
MX-SC $P = 16$	0.490	59	101	1711
MX-SC $P = 32$	0.565	90	125	1388

them to the PEs. These SNGs are shared among PEs within the same row or column, thereby amortizing the overhead of random number generation. We propose a novel SNG architecture, shown in Fig. 3, which employs two 16-bit LFSRs combined using XOR-based mixing to generate statistically independent 8-bit random values. The five least significant bits (LSBs) are used for stochastic bitstream generation. To overcome the latency inherent to serial stochastic bitstreams, we introduce a parallelization factor P , where P LFSRs generate P random bits per cycle, enabling the generation of P stochastic bits per clock cycle. Mantissa multiplication is then performed using P parallel AND gates followed by an adder tree.

IV. RESULTS

A. Accuracy Evaluation

We evaluate ResNet18 across CIFAR-10/100, SVHN, and ImageNet using PyTorch inference with MX-quantized parameters and SC-based mantissa multiply and accumulation. An SC length of 32 is used consistently. MX-SC accuracy remains within 1–3% of FP32, confirming that short bitstreams enabled by our LFSR design are sufficient for DNN inference.

B. Hardware Implementation Results

Synthesis in GlobalFoundries 22nm FD-SOI shows that although the proposed LFSR incurs roughly $3 \times$ area versus a conventional LFSR, the reduced bitstream length and sharing across PEs keep total overhead small. The hybrid systolic array achieves highest efficiency at $P = 8$, reaching 1795 GOPS/W— $2 \times$ higher than binary MX baselines.

V. CONCLUSION

We presented a hybrid MX-SC computation method that maps mantissa arithmetic into the stochastic domain while retaining MX storage for weights and activations. A novel LFSR design and parallel SC generation reduce bitstream

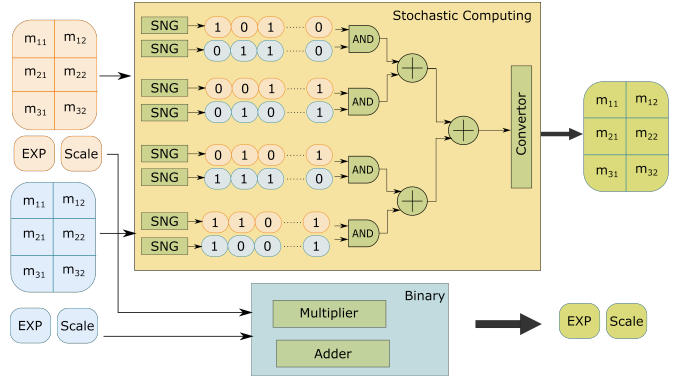


Fig. 1. Proposed Hybrid Computation Method

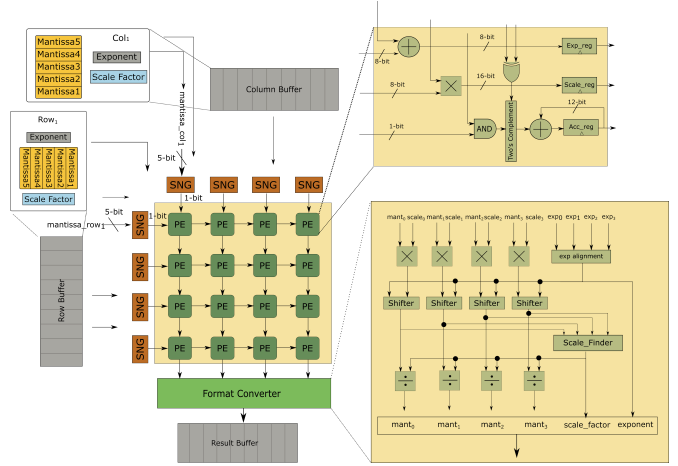


Fig. 2. Proposed hardware architecture

length and improve throughput. The proposed systolic array achieves up to $2 \times$ improvement in energy efficiency with minimal accuracy degradation, demonstrating the potential of hybrid MX-SC computation for future low-power DNN accelerators.

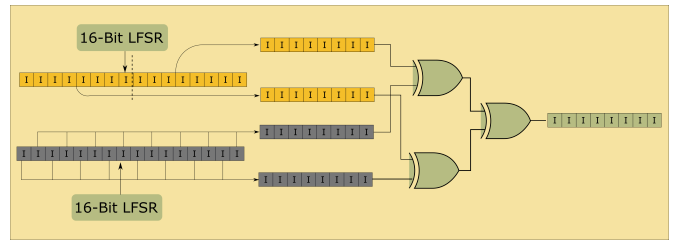


Fig. 3. Proposed LFSR architecture

REFERENCES

- [1] M. Hassani Sadi, C. Sudarshan, and N. Wehn, "Novel adaptive quantization methodology for 8-bit floating-point dnn training," *Design Automation for Embedded Systems*, vol. 28, no. 2, pp. 91–110, 2024.
- [2] B. Darvish Rouhani, R. Zhao, V. Elango, R. Shafipour, M. Hall, M. Mesmahosroshahi, A. More, L. Melnick, M. Golub, G. Varatkar *et al.*, "With shared microexponents, a little shifting goes a long way," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–13.
- [3] J. Yu, K. Kim, J. Lee, and K. Choi, "Accurate and efficient stochastic computing hardware for convolutional neural networks," in *2017 IEEE International Conference on Computer Design (ICCD)*. IEEE, 2017, pp. 105–112.