Live Demonstration: 5-bit signed SRAM-based DNN CIM for Image Recognition

Ó. Pereira-Rial, D. García-Lesta, L. Vaquero, P. López, V.M. Brea, D. Cabello
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)
Universidade de Santiago de Compostela
Santiago de Compostela, Spain
email: victor.brea@usc.es

***Abstract*—This live demonstration shows a mixed-signal Computer In Memory (CIM) macro deep neural network (DNN) integrated circuit in 180 nm CMOS technology for image recognition. Images are coded as pulse width modulation (PWM) signals. DNN weights are stored as voltages in 6T-SRAM memories which drive current sources inside every multiplier. Multipliers are arranged within processing elements laid down in a 2D mesh suitable for image processing. The power consumption per multiplier of the CIM macro is of 0.22 μW, below state-of-the-art competitors following the same multiply and accumulate (MAC) principle.**

I. INTRODUCTION

In memory computing is nowadays a commonly accepted design methodology to break the memory wall from the needs of deep learning models in order to provide custom circuits with high throughput and low power consumption in a small footprint [1]. The 5-bit mixed signal DNN CIM macro for image recognition shown in this demo is our first step towards a stand-alone smart vision sensor system, started in [2].

II. CIM MACRO DESIGN

Our CIM macro comprises 16 $\times$ 16 processing elements (PE) made up of 3 $\times$ 3 multipliers arranged in a 2D array. PE's along a column run 3D filters in one clock cycle. Larger 3D filters are possible in several clock cycles by means of an accumulator provided with memory to store partial sums, and laid down outside the PE array. The MAC principle is that of the input image or intermediate results from hidden layers coded as 5-bit PWM signals multiplied by currents that code the weights. Said weights are 5-bit signed signals stored in 6T-SRAM cells in every multiplier, resulting in compute in memory. The output charge from the MAC operation along every column is gathered on a programmable capacitor bank on the bottom of every column of the PE array. The array yields up to 16 output channels in one clock cycle. The CIM macro implements a ReLU function at the bottom of the 16 $\times$ 16 PE array with a two to one multiplexer that chooses between the truncation and the unity function buffered by a voltage follower amplifier. Intermediate results from up to 16 output channels and the DNN outputs are digitized with a 5-bit flash ADC implemented with a resistive ladder and a bank of comparators. Intermediate digital values from hidden layers are converted to PWM through a digital to time converter before their processing by the PE array.

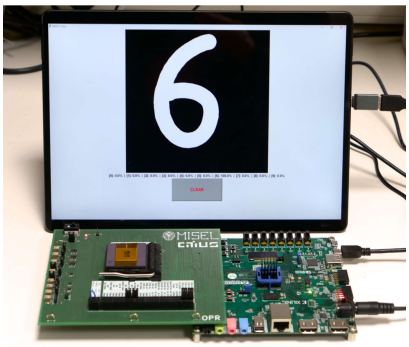

Fig. 1: DNN CIM set-up.

III. LIVE DEMONSTRATION SET-UP AND VISITOR EXPERIENCE

The tool chain of the system comprises a tablet with Python scripts to interface with an FPGA that provides control and I/O signals to our mixed-mode CIM DNN integrated circuit in 180 nm CMOS technology. Fig. 1 shows our DNN set-up. Our design comprises a mother board with the CIM macro chip next to a side board with a Xilinx Nexys Video FPGA. For this demo, we have designed a custom design of a lightweight CNN with 5 layers and 6k parameters for digit recognition trained on the MNIST data set. The DNN layers have been on-chip tuned to reach 91.89% accuracy, slightly above that of the numerical model, of 90.57%. Visitors will have the experience to draw digits on the tablet and see the probability of their try.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101016734; and the European Union (European Regional Development Fund): from the Xunta de Galicia-Consellería de Cultura, Educación e Ordenación Universitaria Accreditation 2019–2022 ED431G-2019/04 and Reference Competitive Group Accreditation 2021–2024, GRC2021/48, and from the Spanish Ministry of Science, Innovation and Universities under grant PID2021-128009OB-C32.

REFERENCES

[1] N. R. Shanbhag and S. K. Roy, "Benchmarking In-Memory Computing Architectures," *IEEE Open Journal of the Solid-State Circuits Society*, vol. 2, pp. 288–300, 2022.

[2] O. Pereira-Rial *et al.*, "Design of a 5-bit Signed SRAM-based In-Memory Computing Cell for Deep Learning Models," in *ISCAS*, 2022.