An Incremental Time-Domain Mixed-Signal Matrix-Vector-Multiplication Technique for Low-Power Edge-AI

Kévin Hérissé, Benoit Larras, Bruno Stefanelli, Andreas Kaiser, Antoine Frappé

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

An Incremental Time-Domain Mixed-Signal Matrix-Vector-Multiplication technique for low-power edge-AI

Kévin Hérissé, *Member, IEEE*, Benoit Larras, *Member, IEEE*, Bruno Stefanelli, Andreas Kaiser, *Senior Member, IEEE*, Antoine Frappé, *Senior Member, IEEE*

***Abstract*—This paper proposes a time-domain mixed-signal computing architecture for Matrix-Vector Multiplication suited for embedded in-memory computing applications. The system leverages the low data rate of sensors' data in embedded AI applications to target an energy-efficient implementation of the matrix-vector multiplication array. The mixed-signal computing scheme relies on incremental time-domain multiply-and-accumulate operations using switched current sources. The concept is demonstrated on a 28 nm FDSOI prototype chip of a 100 x 4 compute array that shows a 15.8 TOPS/W energy efficiency for 5-bit MAC operations. Extrapolating the array to 100 x 100 computing units leads to a 99.2 TOPS/W energy efficiency.**

***Index Terms*—In-Memory Computing, Matrix-Vector Multiplication, Multiply-and-Accumulate, 28nm FDSOI, multi-bit computing**

I. Introduction

APPLICATIONS for embedded artificial intelligence (AI) are numerous and cover multiple domains, from consumer electronics, home automation, and health to industry. To run neural network accelerators on a device with limited energy budget and limited connection to a server, Tiny Machine Learning (TinyML) can be deployed on a dedicated ASIC. However, a system composed of embedded sensors that continuously send data to the embedded classification core consumes a lot of energy since the processing elements need to stay always on. It is possible to divide the main classification task into simpler tasks that produce a wake-up signal for the main processor to start its local computation. This hierarchical scheme, shown in Figure 1, relies on a dedicated pre-processing unit coupled with the main processor and offers a lower energy consumption. For example, in the case of natural language processing, the pre-processing unit can run a Voice Activity Detection (VAD) or Keyword Spotting (KWS) algorithm and Speaker Verification (SV) to wake up the main processor when a specific keyword or sound is detected [1]. By shifting the continuous classification task to a dedicated ultra-low energy pre-processing unit, the main processor only performs the remaining and more computationally hungry tasks on relevant input data. The pre-processing unit is generally composed of a feature extraction and a classification block. This work focuses on the latter.

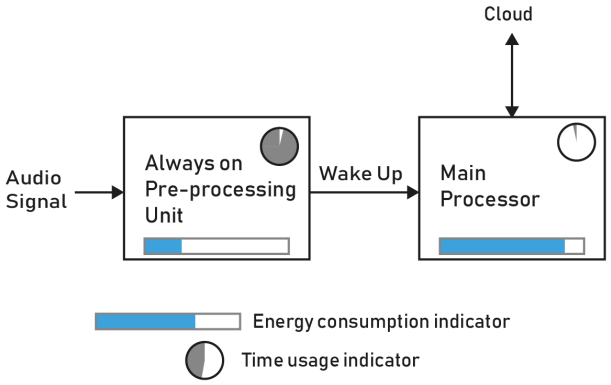


Fig. 1: Exemple of architecture using pre-processing unit for audio processing.

Two levers can be exploited to reduce the energy per inference of a neural network (NN), as shown in Equation 1:

$$\frac{\text{energy}}{\text{inference}} = \underbrace{\frac{\text{operation}}{\text{inference}}}_{\text{Software approach}} \times \underbrace{\frac{\text{energy}}{\text{operation}}}_{\text{Hardware approach}} \quad (1)$$

The software approach aims to minimize the number of operations performed during inference. This is achieved using pruning techniques [2], [3] to reduce the number of parameters, quantization to reduce the weights and activation bitwidth using Post Training Quantization (PTQ - the network is quantized after training) [4]–[6] or Quantization Aware Training (QAT - the network is quantized during training) [7]–[11]. Quantization can be exploited up to a binary representation of weights with a tradeoff regarding memory size and performance [12]. Binary neural networks are particularly suited for digital implementations [13], [14]. However, being able to implement configurable multi-bit neural networks is required to fit a large range of neural network architectures targetting a wide range of applications. Consequently, this study concentrates on the implementation of multi-bit architectures. The software approach is helpful for training networks, with fewer parameters and less complex activation functions

This work was supported in part by the French National Research Agency under Grant ANR-18-CE24-0006-01 LEOPAR and in part by the Nano 2022 - IPCEI program. The team would like to thank STMicroelectronics for providing the silicon of this chip and especially Andreia Cathelin for her support.

The authors are with Univ. Lille, CNRS, Centrale Lille, Junia, Univ. Polytechnique Hauts-de-France, UMR 8520-IEMN, France. (e-mail: kevin.herisse@junia.com; benoit.larras@junia.com; bruno.stefanelli@junia.com;andreas.kaiser@junia.com; antoine.frappe@junia.com)

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

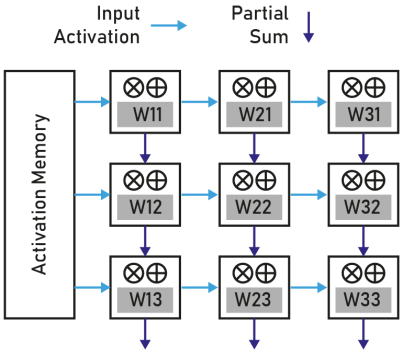


Fig. 2: Weight-stationary MVM data flow from [15].

keeping the accuracy level high. The hardware approach will allow reducing the energy consumption of the operations, theoretically without affecting the accuracy.

The hardware approach aims at optimizing the energy consumption per operation, especially minimizing the energy of the multiply and accumulate (MAC) operation that composes the Matrix-Vector Multiplication (MVM). To avoid the energy cost due to the ”memory wall”, the processing elements (PE) are typically implemented inside the memory. PEs embedding their own local memory are placed in an array to provide a weight-stationary data flow of the MVM, as shown in Figure 2.

The literature offers a wide range of In-Memory Computing approaches. Digital Compute-In-Memory (DCIM) [16], [17] offers good energy efficiency, high configurability and scalability and finds applications in large-scale accelerators. Analog Computing-In-Memory (ACIM) shows a higher potential to reach very high energy efficiency, but is limited to medium-precision weights (4 to 8 bits) [18], since this type of architecture is more susceptible to noise and mismatch.

Time-domain architectures represent a promising alternative that harnesses the strengths of both digital and analog techniques. It exhibits attributes of scalability and configurability and can be seamlessly integrated with an ACIM (Analog Computational In-Memory) framework to facilitate the incorporation of an ultra-low power pre-processing unit.

In the time-domain paradigm, as documented in prior studies [19]–[21], input signals undergo transformation into pulse modulations based on their values. A typical architectural illustration of a time-domain system is presented in Figure 3, wherein the input signals are modulated into pulses through a Digital-to-Time Converter (DTC). Subsequently, these modulated signals are transmitted to the weighting elements for multiplication, with a dedicated DTC for each input. The accumulation of results takes place in the analog domain, requiring an Analog-to-Digital Converter (ADC) to store a digital output.

To facilitate multi-bit operations, time-domain solutions employ either a parallel strategy, involving weighted cells, or an iterative approach, which entails multiple ADC conversions (N conversions for N-bit weights), or a combination of both techniques. Regardless of the chosen strategy, additional shift and add circuitry is essential for reconstructing the final output.

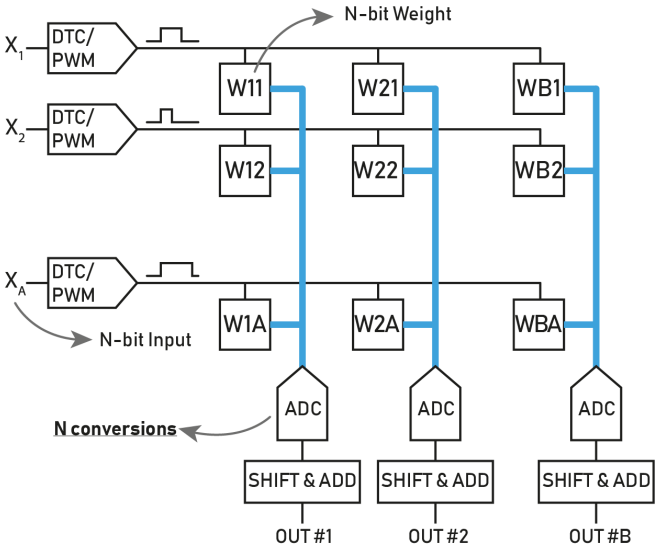


Fig. 3: Time-domain iterative conversion.

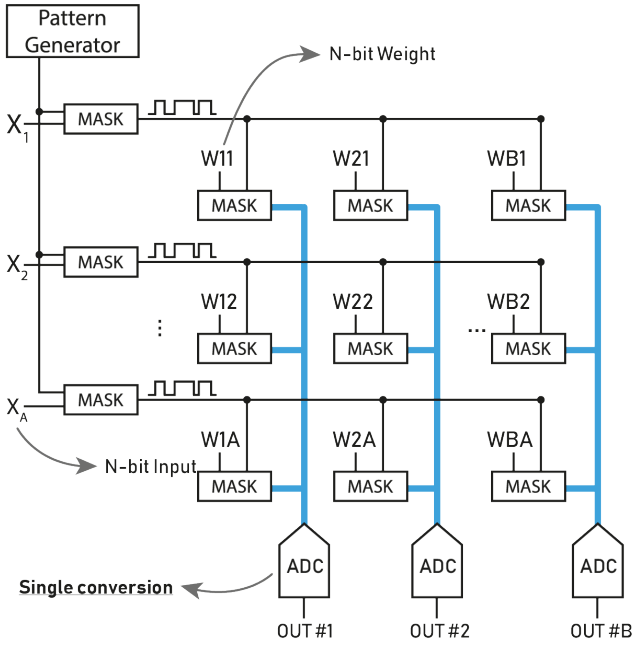


Fig. 4: Illustration of a time-domain single conversion exploiting the double bit-masking technique.

An excellent example of such techniques is found in [22]. A digital-to-time conversion is performed on each input using a global pulse generator. For the duration of the pulses, each bitcell is selectively read on a positive or negative bit line, depending on the input sign. To avoid an ADC conversion for each bit position, a weighted charge-sharing technique is employed to divide the voltage by the corresponding binary weight and a differential 6-bit SAR ADC exclusively converts the resulting output.

Figure 4 proposes a novel approach for addressing the challenge at hand. The solution uses a double bit-masking technique for the inputs and weights. A unique pattern genera-

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

tor creates a repetitive weighted pulse signal that is masked by the inputs at each row. The resulting signals are propagated to the multi-bit weights that will further locally mask the signal. It allows to perform the $N \times N$ -bit MAC operation in a single ADC conversion without requiring additional combinational circuitry.

This paper proposes a time-domain current-based macro that allows for one-shot operation for multi-bit MAC operations. The following contributions are presented:

- A new mixed-signal architecture using incremental time- and current-based macro ready for ACIM.
- A double masking technique allowing just one ADC conversion per 100 MACs
- A 28 nm FDSOI prototype circuit of the macro for a 100×4 matrix array performing 5-bit MAC operations, reaching 15.8 TOPS/W and 99.2 TOPS/W when scaled to a 100×100 matrix array.

The remainder of this article is organized as follows. Section II presents the memory storage, mixed-signal combination techniques and the time and current-based principle. Section III details the on-chip implementation of the proposed concept. Section IV shows the measurements of the 28nm FDSOI prototype. Section V presents the key contributions of this work compared to prior In-Memory Computing macro implementations and opens the discussion about the work. Finally, section VI concludes the article.

II. Time-domain current-based MAC

A. Memory storage and mixed-signal combination techniques

MVM dataflow at the edge of a Deep Neural Network (DNN) requires fixed weights in a dense array to reach high memory capacity. On-chip non-volatile memories (NVM), such as Resistive-random-access-memory (RRAM), are often used in ACIM applications for their high density and their ability to mitigate the leakage of SRAM-based solutions. RRAM-based solutions [23] use memristors to store the weights in the form of a conductance value. To increase the density, some works [24] store up to 4-bit on an RRAM, with a programming endurance of 100k cycles across the 16 levels. Although compatible with back-end-of-line (BEOL) CMOS, this solution requires additional process steps to the standard CMOS, high-current pulses to program the weights, and are limited to low precision applications.

Therefore, SRAM-based in-memory computing solutions are a great choice for robust integration in any CMOS node. In the charge-based approach, the bitcell is composed of a modified 6T SRAM and additional transistors to charge a capacitor according to an element-wise multiplication. The charges are redistributed across all capacitors on an Accumulation Line (AL), resulting in a voltage to be further converted by an ADC [25]–[27]. To perform a multi-bit operation with switched capacitors, [25] shows a topology similar to a digital multiplier using one AL for each output bit position. However, this method needs additional circuitry to combine all the line's results, which is adequate for a reduced number of lines (< 5) but dominates the power consumption for higher numbers of bits.

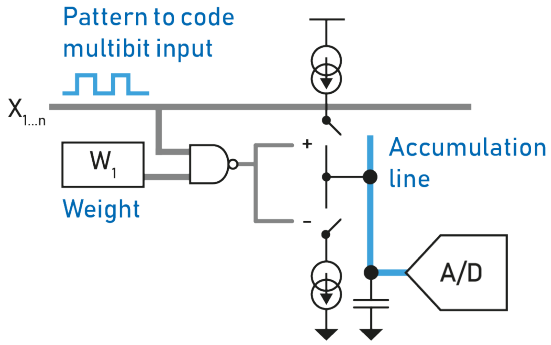


Fig. 5: Current sources used to charge/discharge a capacitive line controller by a PWM signal.

Other charge-based SRAM arrays implement MAC as a weighted average of voltages, which are proportional to the word line voltages, converted from the digital inputs. However, with this approach [28], [29], where multiple wordlines are activated with the same bit line, the system suffers from writing disturbance where bits can be flipped if the level of the line is too low. [30] proposes a 10T SRAM that decouples the memory writing and the MAC operation, therefore increasing the area overhead of the bitcell. Reference [31] exploits charge-domain architecture, performing bitwise multiplication using XNOR and AND gates to drive a capacitor causing charge redistribution across a capacitance.

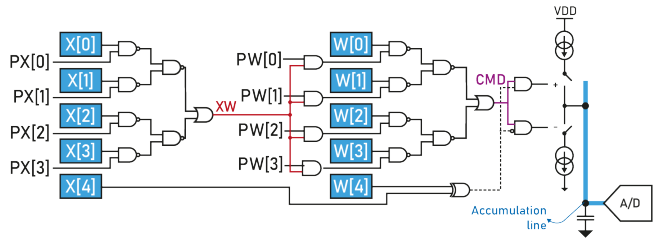
Current-based approach uses a current to charge/discharge a capacitive line proportional to the inputs and weight multiplication. The main principle of time-domain and current-based MAC, derived from [32], is to use two current sources associated with one switch each, controlling the flow of current charging/discharging a capacitive accumulation line (AL) as shown in Figure 5. Recent CMOS technologies allow for reducing the current of the current sources ($< 1nA$) without too much mismatch and envisioning high efficiency for MAC operations.

B. Time-domain computation

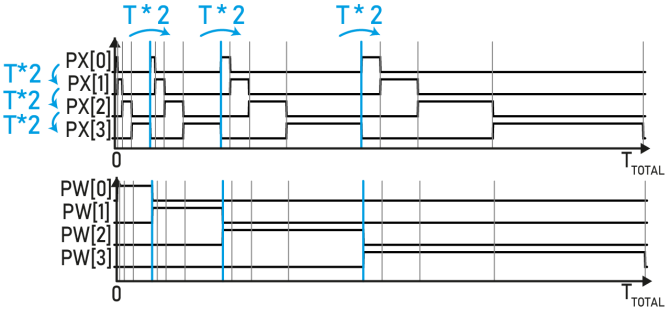
In the context of the TinyML environment, it is possible to leverage the low output rate of embedded sensors to perform computation. Although it results in a lower throughput, this does not affect the possibility of implementing a wide range of applications. Examples of applications with moderate speed requirements are Keyword spotting and parking spot detection. In Keyword spotting (KWS) a new spectrogram frame is processed every 16 ms. In parking spot detection, a new video frame every second meets the requirements. This work envisions a solution for the low throughput application of TinyML targeting embedded sensors and preprocessing of the data on battery-constrained devices.

A pulse stream gated by each input bit and weight bit is used to control a block able to perform accumulation in the analog domain. However, to avoid the need for recombination, the current contribution of each bit-wise operation needs to be scaled according to their respective bit rank. This is achieved

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154



(a) Masking architecture for a single MAC operation.



(b) Patterns used to create a command signal.

Fig. 6: Pattern and the masking logic for MAC operations.

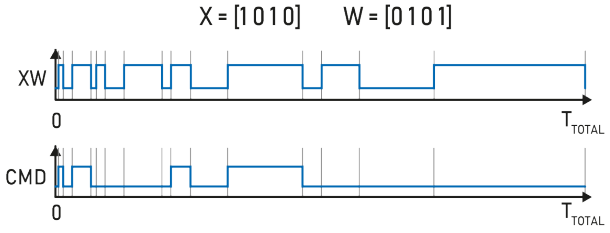


Fig. 7: Numerical example of output signal used to drive the switch connected to the accumulation line.

by modulating the pulse-width length by power of two ratios. As shown in Figure 6b, there is one pulse for each bit of X, the first pulse associated with each bit is of length $T * 2^k$, k being the bit rank. Each pulse is shifted to prevent overlapping with the other one. For each weight position, the pattern for the X bits is cycled through, with a pulse width multiplied by two to scale the current according to the weight bit rank. Figure 6a details the masking architecture used to command the analog accumulation block and perform a 5-bit MAC operation with a single ADC conversion. Figure 7 illustrates the generation of the XW and CMD signals based on the input vector $X = [1010]$ and the weight vector $W = [0101]$. The XW signal is produced by combining the PX patterns and masking them with the input vector X. This XW signal is then further masked by the weights W, utilizing the corresponding PW patterns to create the CMD signal. The final CMD signal is routed to the appropriate switch, governed by the sign bits of both the input ($X[4]$) and the weight ($W[4]$).

C. Methods for accumulation

To be able to compute matrix-vector multiplication, the X pattern is broadcast across all the input columns. The resulting masked signals are then broadcast to the complete row to be masked by the weights stored inside the processing element. Figure 8 shows a diagram of the complete architecture. The processing elements (PE) are composed of the weight storage, the pattern masking logic (WLOGIC), and the current sources with their switches. Each column of PE shares a capacitive AL. According to the sign of the MAC operation, computed with a simple XOR between the weight and input sign, the AL will be charged or discharged by the corresponding current source during a time T proportional to $X \times W$. All the contributing PE outputs result in a total current that will charge the AL to a voltage proportional to the accumulation of the consecutive sums of all PE outputs. This voltage can then be converted by an analog-to-digital converter, the digital value being equal to the result of 100 MAC operation. .Figure 9 shows an example of the AL behavior for 100 random inputs and 100 random weights operation simulated with Matlab®. The AL is initialized to a voltage reference of $\frac{V_{DD}}{2}$ (here 0.4V) to allow for negative results, as shown on the numerical axis.

III. On-chip implemented architecture

The designed IC embeds a macro of 100 inputs and 4 outputs MVM for 5-bit MAC operation. This section will first present the design of the PE and then the design of the shared elements of each AL.

A. Processing Elements

1) Weight storage and masking logic: The weights are stored on D-latch registers. The masking is performed with logic gates. We use poly-biased gates to further reduce the leakage of those elements. To further decrease the energy consumption, the storage could easily be replaced with SRAM cells.

2) Current Sources: The current sources are implemented as $800\,nm \times 800\,nm$ cascode current sources presented in Figure 10. For this transistor dimensions, the mismatch is simulated at 18% $(\frac{\sigma}{\mu})$. The bias references of the current sources are shared for all PEs of a column. To mitigate the gate leakage effect arising when 100 gates are connected together, the transistors use thick gate oxide, reducing the gate leakage from 30 pA to a negligible current of 10aA according to simulation with $800\,nm \times 800\,nm$ unit transistors. The cascode architecture increases the output impedance on the drain connected to the switch, providing a constant current over a wide voltage range. As the AL voltage evolves, it will reach the limits of the voltage range, causing the current to drop as the current mirror transistor is no longer saturated.
The reference current is drawn from an external analog pin and divided by three successive current mirrors with a 10:1 ratio. A 100 nA reference results in a 100 pA current inside the PE. There is one current reference for each type of current sources, IREF_P and IREF_N. Switches are composed by complementary PMOS and NMOS transistors of size $100\,nm \times 100\,nm$ resulting in a 0.5 pA leakage when both switches are off. This

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

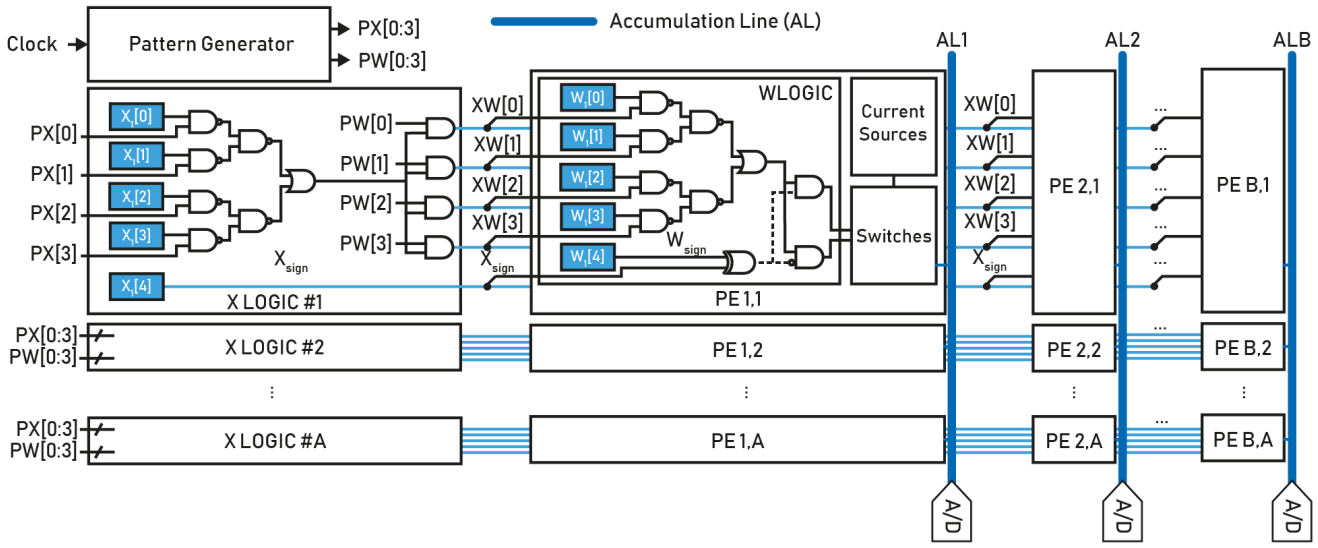


Fig. 8: Diagram of the architecture for a Matrix-Vector Multiplication using time and current-based In-Memory COmputing ready solution

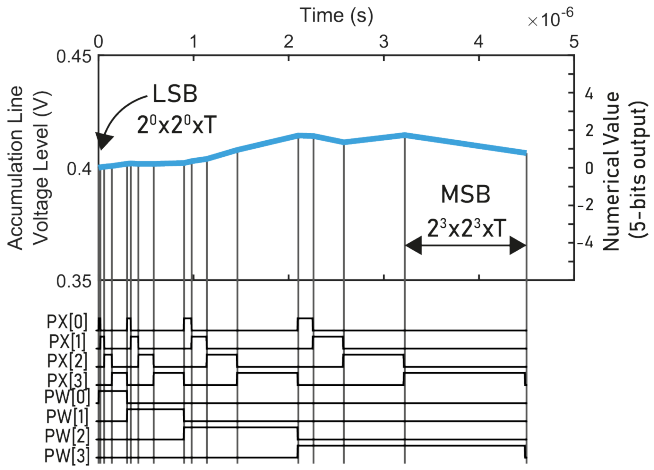


Fig. 9: Accumulation Line behavior for a 100 random inputs multiplication, with corresponding numerical value.

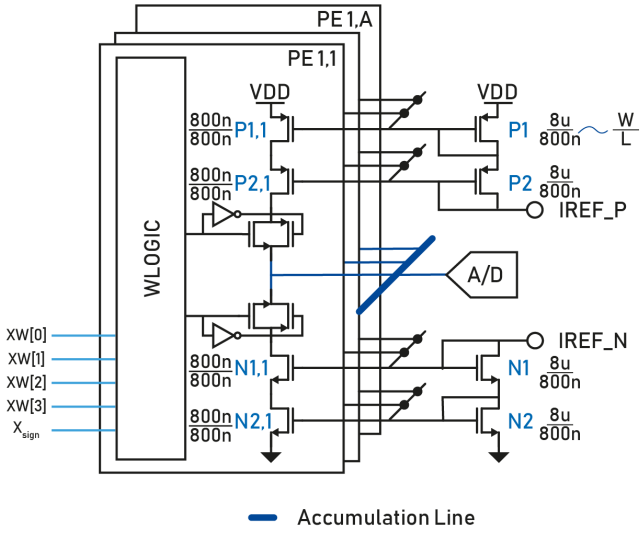


Fig. 10: Processing element schematics.

leakage does not contribute to the output, since it is mitigated by the two opposed switches drawing a similar leakage current with opposed signs.

B. Accumulation Line

The total capacitance of the accumulation line is primarily determined by the parasitic capacitances from the switches and the metal line. This capacitance value is obtained through individual post-layout simulations of the processing elements (PEs) with extracted parasitics. To estimate the overall capacitance of the accumulation line, the capacitance of a single PE was scaled by the number of PEs connected to the line. Figure 11 illustrates the estimated capacitance for a configuration of 100 PEs. It is important to note that the capacitance is not constant; it varies depending on the voltage level and the operational state of the switches. When the switches are active, additional parasitic capacitances of about 30 to 40fF are introduced. However, this estimation does not account for the capacitance contribution from the metal line or other components connected to the accumulation line, which will significantly increase the total capacitance to about 400fF and mitigate the variations.

1) Current source switches: When the switch is off, the output of the current mirror is floating. This situation makes the current establishment time not suited for a reference time T needed for medium throughput ($\approx 20\,ns$). To be able to switch fast enough, a current steering technique is used as

This article has been accepted for publication in IEEE Transactions on Circuits and Systems–I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

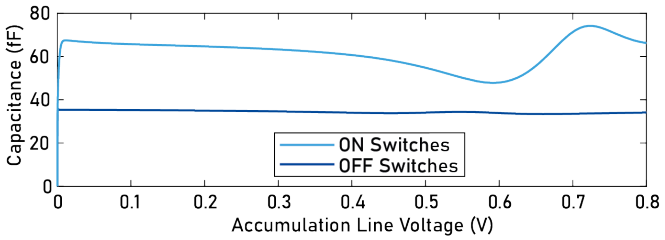


Fig. 11: Post-layout estimation of the capacitance of 100 PEs as a function of the accumulation line voltage. This does not include the approx. 400fF accumulation line capacitance.

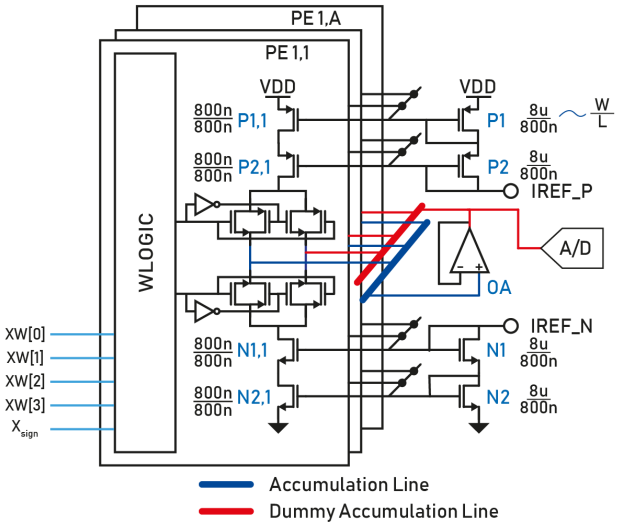


Fig. 12: Processing element schematics with current steering and operational amplifier.

shown in Figure 12. The current source output is connected to two switches that work in opposition. When the current does not flow into the accumulation line, it is redirected into a dummy accumulation line, hence, ensuring a fast reliable current establishment time. This improvement creates however a charge-sharing issue. As the two switches are not perfectly synchronized, there is a moment when they are both on and the AL and dummy AL are sharing charges and trying to balance to the same voltage level. This effect is mitigated by the addition of a voltage follower, at the column-level, between the two accumulation lines, making the operational amplifier (OA) offset the only difference between the two lines.

2) Operational Amplifier: The operational amplifier, with dual differential pair, shown in Figure 13, is able to handle common mode voltages from 0.2 V to 0.6 V which corresponds to the maximum voltage reachable by the accumulation line (at $V_{DD} = 0.8\,V$). According to the input level, one of the differential pairs will amplify the signal. However, the OA non-constant offset is now impacting the output result since part of this offset is added to the accumulation line via charge sharing when switching.

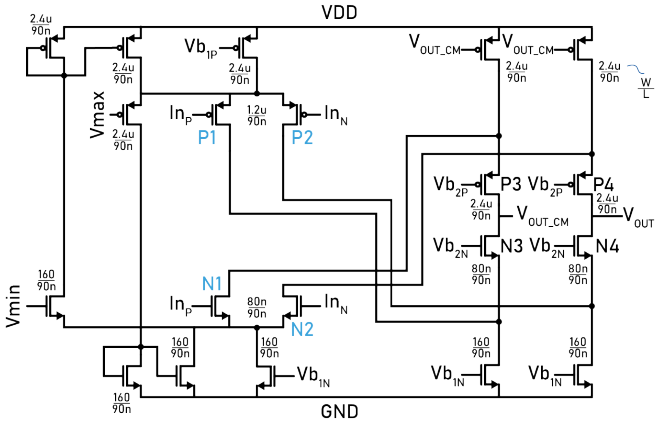


Fig. 13: Schematic of the rail-to-rail operational amplifier.

C. Pattern Generation

The pattern generator architecture is implemented as a finite-state machine, using multiple counters and an external clock with period T. The sequence is generated according to a predefined number of clock periods, depending on the number of bits of the signed inputs X and weights W, denoted N_x and N_w, respectively. The total time of the sequence T_{tot} is given in the following equation, with respect to the reference time T.

$$T_{tot} = (2^{N_x-1} - 1)(2^{N_w-1} - 1) \times T \tag{2}$$

At the end of each pulse, a counter is incremented and this value is compared to the configuration register to activate the next pulse with the correct duration. The outputs PX and PW are distributed to all the XLOGIC blocks. Like the other digital blocks of this chip, the elements composing the pattern generator are polybiased to reduce energy consumption.

D. Model of the computation

A model was developed on Matlab® to evaluate the impact of mismatch, and accumulation line capacitance variation on the output result. As errors might compensate each other, they are evaluated separately first and together in the last part of this section. Inputs and weights are converted into an array storing their binary representation (Sign + 4b Magnitude). At the beginning of each pattern pulse, the number of contributing current sources is calculated by summing the number of corresponding bits noted N_{ON} for the number of discharging current sources and P_{ON} for the number of charging current sources. P_{ON} and N_{ON} are updated at each time step corresponding to a new pulse. For higher precision, the simulation is performed discretely in time according to a time step T_s chosen by the user. For each time step, the output voltage is:

$$V_{out}(T_s) = V_{out}(T_{s-1}) - \frac{I_{ref}T_sN_{ON}}{C_{ref}} + \frac{I_{ref}T_sP_{ON}}{C_{ref}} \tag{3}$$

To increase the precision of the model, the current reference I_{ref} is interpolated from transistor-level simulated values. The different error values are computed for 10,000 random inputs

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

and weights drawn from a normal distribution. The mean error due to the non-linearity of the current sources is equal to $-2.13\mu V$ error with a standard deviation of $1.36\mu V$.

1) Impact of mismatch: The mismatch is modeled by drawing current from the following normal distribution:

$$I_{mismatch} = \mathcal{N}(I_{ref}, \sigma = I_{ref} \times \epsilon) \tag{4}$$

with ϵ the mismatch value in percentage of μ. The $I_{mismatch}$ is drawn once at the beginning of the computation and kept constant during the computation for each element of the accumulation line. With an input mismatch of 20% ($\frac{\sigma}{\mu}$), the mean impact of the mismatch is equal to $3.45\mu V$ with a large standard deviation of $2.8mV$.

2) Impact of accumulation line capacitance non-linearity: Figure 11 shows the capacitance value of an accumulation line composed of 100 PEs depending on the state of the switch. A mean capacitance value is calculated, which corresponds to one of the two switches controlling the two current sources being on and the other off. By using interpolation we can use a capacitance value at each time step according to the accumulation line voltage. The mean error added is equal to $-1.26\mu V$ with a standard deviation of $21\mu V$.

3) Summary of contributions: The final error contribution is equal to $2.18\mu V$ with a standard deviation of $2.8mV$ dominated by the mismatch value. With an available voltage dynamic ranging from 0.2 V to 0.6 V the number of quantization levels N_Q is equal to:

$$N_Q = \frac{0.6 - 0.2}{0.0028} = 142 \tag{5}$$

With 142 quantization levels, the number of available output bits is equal to 7.15 bits. This is in line with the 5-bit precision of inputs and weights.

E. Simulations

1) Mismatch Analysis: The mismatch is simulated as 18% for 800 nm by 800 nm PMOS and 6% for NMOS transistors of the same size. The values are estimated from a 200-point Montecarlo analysis with a current reference of 100 pA.

2) Corner analysis: The behavior of a single 100-input accumulation line is simulated in different process corners with random inputs and weights and is shown in Figure 14. The maximum final deviation with respect to the typical (TT) curve is $3\,mV$ (less than 1 LSB for a 5-bit output) for the SS corner. Calibration can be performed at start-up to correct for process variations. In a practical implementation, a simple calibration scheme would rely on performing MAC operations with a known target output voltage, recording the output value and adjusting either the global reference current or the global clock period to match the expected output result. This calibration scheme can also be used for voltage and temperature variations, most probably on a regular basis to account for time-dependent variations of the voltage or temperature.

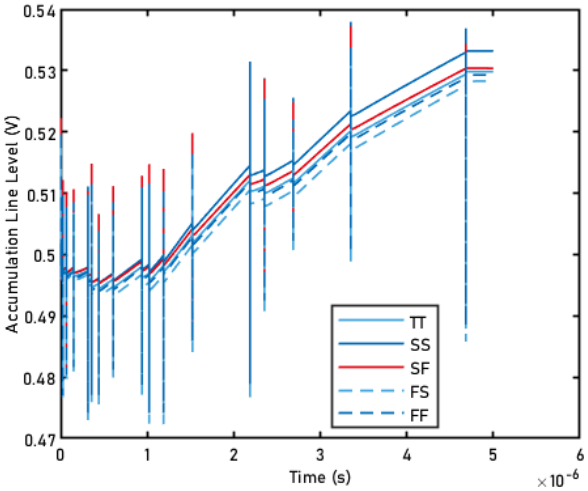


Fig. 14: Accumulation Line Behavior with Corner Analysis.

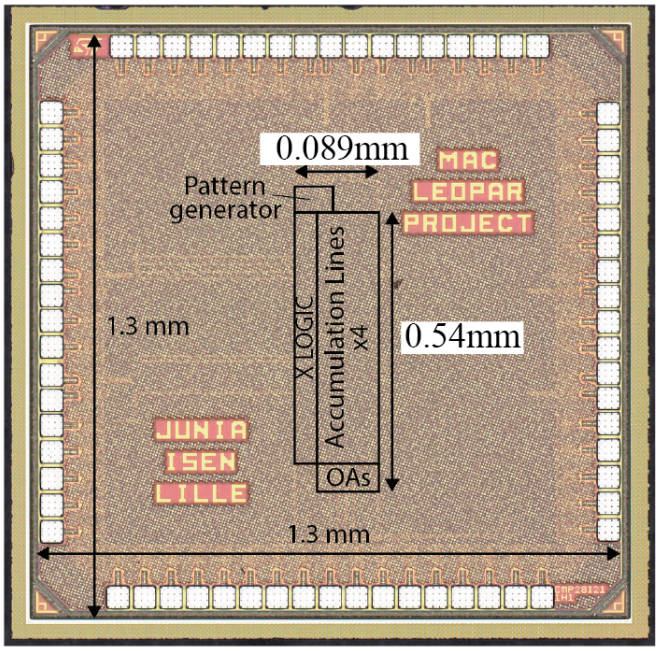


Fig. 15: Photograph of the die.

IV. Measurements Results

The prototype IC, composed of 100 inputs and 4 accumulation lines was fabricated in STMicroelectronics 28nm FDSOI CMOS process. Figure 15 shows a die photograph and Table I a summary of the features and performances of the circuit.

Figure 16 shows the detailed layout of one PE. Memory and logic gates account for 50% of the total area and current sources for 25%. The measurements were performed at room temperature under 0.8V supply. The first part of this section presents the initialization and calibration of the macro. Finally, the computation precision and energy breakdown are presented.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

TABLE I: Summary of the circuit characteristics.

Technology	28 nm
Cell Type	Flip Flop Register
Test Chip Area (w/ IO	$1.69\,mm^2$
Macro Area	$0.048\,mm^2$
MAC Computing Latency	4.5µs @ 5bIN-5bW 20ns @ Ternary IN & W
Throughput	0.177 GOPS (5b) 40 GOPS (Ternary)
Throughput (scaled for a 100x100 array)	4.44 GOPS (5b) 1,000 GOPS (Ternary)
Efficiency	15.8 TOPS/W (5b) 3,573 TOPS/W (Ternary)
Efficiency (scaled for a 100x100 array)	99.2 TOPS/W (5b) 22,295 TOPS/W (Ternary)

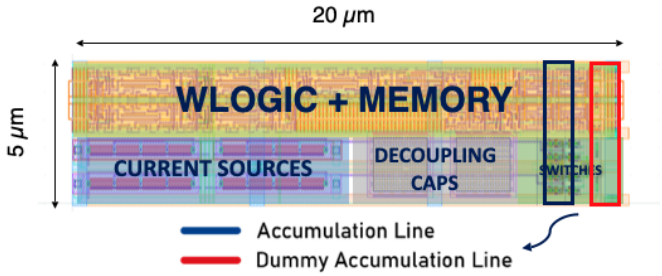


Fig. 16: Layout of the Processing Element (PE).

A. OA Characterization

In addition to the operational amplifier placed between the accumulation line and the dummy accumulation line, named OA_A on Figure 17, a second voltage follower operational amplifier is placed between the dummy accumulation line and the output pin of the circuit OA_B. To measure it, each line can be charged to an external voltage reference ($VREF\_LN$ and $VREF\_DLN$) thanks to two pass gate switches controlled by RST_LN and RST_DLN respectively. Each Accumulation line has dedicated voltage reference pins, and the reset commands are shared across all the accumulation lines. The offset is measured by applying a known voltage reference on each accumulation line. The offset of the operational amplifier OA_A is measured by first measuring the offset of OA_B by applying a known reference on the dummy accumulation line and measuring the difference with the output value, then the cumulated offset of OA_A and OA_B is measured by applying a known reference on the accumulation line, measuring the difference with the output value and subtract the OA_B offset. The measured offset of OA_A is shown in Figure 18. The offset is not linear and ranges from 5 mV to 70 mV.

B. Accumulation Line Behavior

1) Capacitance measurement: To measure the accumulation line capacitance, X and W bits are set to 1, except for the sign. Figure 19 shows the evolution of the accumulation line voltage when a 40pA current per cell is applied. An estimated

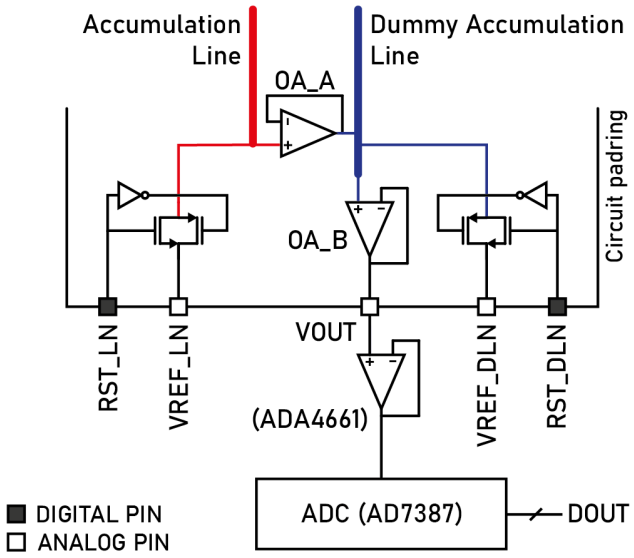


Fig. 17: Architecture for OA characterization.

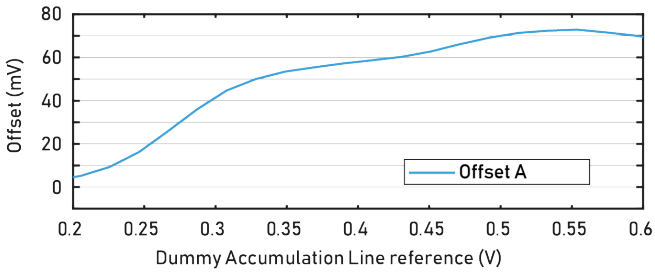


Fig. 18: Measurement of the offset of OA A.

capacitance value of 400 fF is extracted from this curve. The difference between the measured and the simulated values can come from the simulation that was performed on one processing element with extracted parasitics and then scaled to a complete accumulation line. In addition, the simulation did not take into account the OA input capacitance. To compensate for the higher accumulation line capacitance, the current is scaled up to 300 pA per cell for a 50MHz clock frequency.

2) AL behavior: Figure 20 shows an example of the measured output during the accumulation process for randomly selected multiplications. The accumulation behavior is in line

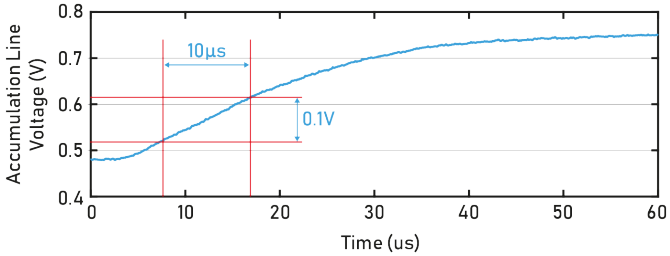


Fig. 19: Accumulation Line capacitance measurement.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

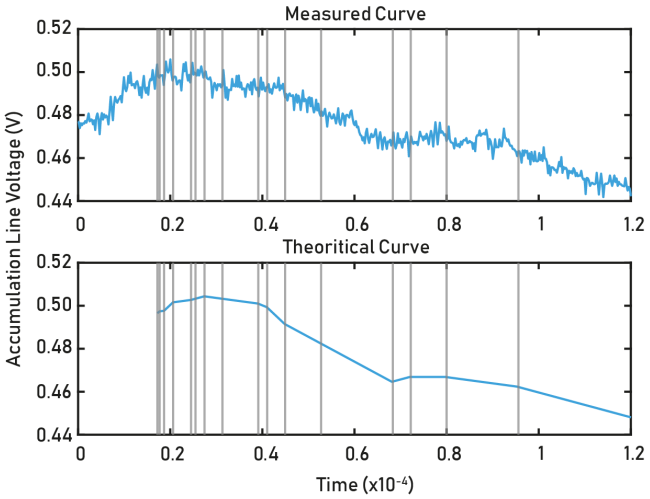


Fig. 20: Accumulation Line behavior with random input and weight.

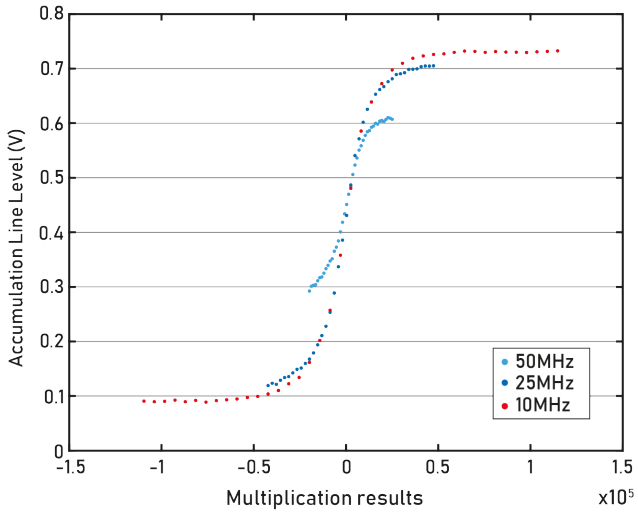


Fig. 21: Transfer function for different input clock frequencies.

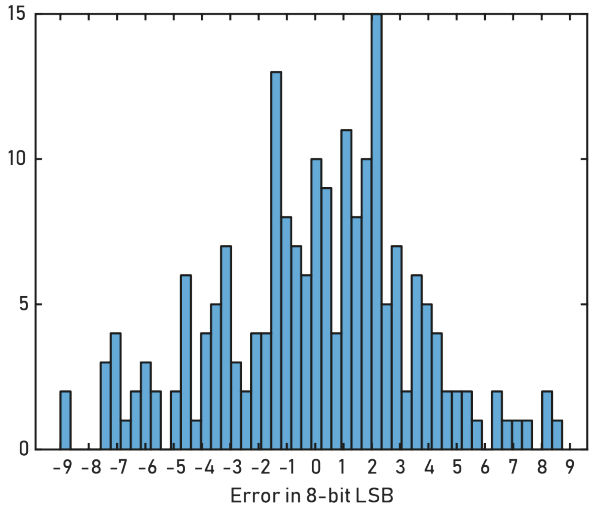


Fig. 22: Histogram of the computed errors between the expected theoretical output value and the measured output on 200 randomly selected inputs and weight. The error is normalized to an 8b output precision with a dynamic range of 330mV. This test is performed at a 50MHz clock frequency and 0.8V voltage supply.

with the expected values, simulated on Matlab®.

C. Transfer function

The input-output transfer function is given in Figure 21. The curve is computed by setting X and W values accordingly to span across the value range. The figure shows three curves measured at three different time references: 20 ns (50 MHz), 40 ns (25 MHz) and 100 ns (10 MHz), hence the gain applied on the figures x axis. Different saturation levels are observed for each time reference, the lowest saturation level corresponding to the 50 MHz curves is due to the bandwidth limitation of the operational amplifiers, whereas the saturation of the two other curves is a combination of the bandwidth limitation and the current mirrors entering their triode region. This behavior does not impact the linearity of the operation as normally distributed random inputs and weights result in a Gaussian distribution of the output that is centered on 0.4V.

D. Output precision evaluation

To evaluate the output precision, the output value of the computation on a single accumulation line is measured for 200 randomly selected inputs and weights. A Matlab© script selects the inputs and weights values and computes the theoretical result matching the gain of the current and time references applied to the circuits. The output level of the circuits is retrieved thanks to the oscilloscope controlled by the computer via serial communication. The error between the measured and expected values is then computed. Figure 22 shows the histogram of the computed error. It is normalized to the LSB of an 8b output, considering a 330 mV dynamic range. Assuming the distribution follows a Gaussian shape, the calculated standard deviation σ is equal to 7.2 mV. In [33], it is shown that small LSTM networks can tolerate σ/LSB of up to 100% without significant degradation, which translates here in 46 possible quantization levels, equivalent to an effective 5.5-bit output precision. Classic benchmarked datasets, like MNIST or CIFAR-10, show near SoA accuracy with such configurations of 5x5x5 input/weight/output bitwidths [17], [34], [35].

E. Energy Efficiency

The energy consumption of each block is measured thanks to a dedicated supply pin. For the 400 MACs array (100 inputs $\times$ 4 ALs) under 0.8 V, the current sources consume $0.55\mu W$, the logic blocks for X (100 blocks) and W masking (400 blocks) consume 0.098 µW, the 4 operational amplifiers OA_A consume $0.75\mu W$ and the pattern generator consumes $9.79\mu W$. The 4 OA_B consume $40\mu W$ but are not included in the final energy evaluation since they are only used for test. The diagram in Figure 23 shows the domination of the pattern

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

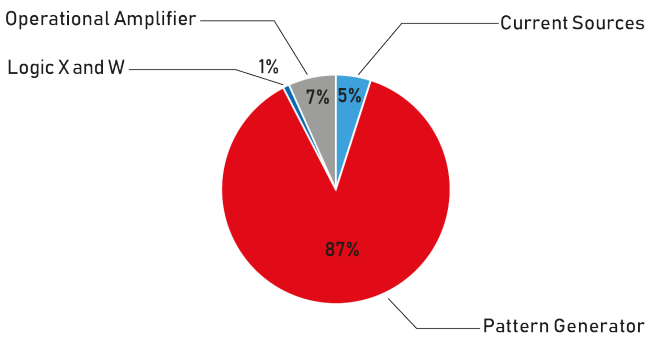


Fig. 23: Energy repartition of the 4x100 chip prototype.

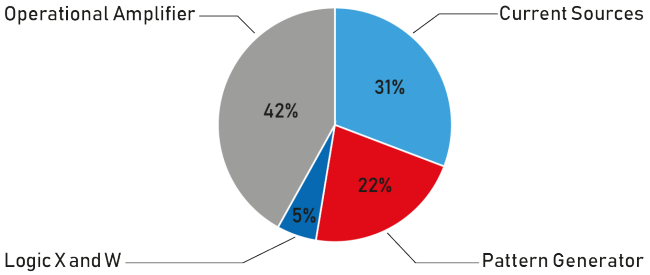


Fig. 24: Energy repartition with the pattern generator amortized across a 100x100 matrix.

generator with 87% of the total 11.188µW power consumption. Using a 4.5µs inference time, the energy efficiency of the macro reaches 15.8 TOPS/W. It is to note that the pattern generator provides the PX and PW signals to the 100 rows of the array. Thus, the large portion of energy consumed by the pattern generator can be amortized if we implement more ALs in parallel. To estimate the potential gain in energy efficiency, the consumption is extrapolated for a 25x larger array composed of 10,000 MACs (100 input × 100 ALs). The following process is performed to have a fair estimation of the potential performance. The power consumption related to the current sources and the operational amplifiers will scale linearly as we increase the number of ALs and will be respectively 13.75 µW and 18.75 µW. The number of X logic block will stay constant, while the number of W logic blocks will scale with the number of ALs. However, since the drivers inside the X logic block need to be increased to handle the larger parasitic capacitance of the longer row lines, we considered the pessimistic scenario in which the measured consumption of 1 X and 4 W logic blocks is multiplied by 25, leading to 2.45 µW. Finally, the pattern generator consumption will be equal, since it only depends on the number of rows. The total consumption is estimated to 44.74 µW. For the 10,000 MACs array, this is equivalent to 99.2 TOPS/W. Figure 24 reports the energy consumption distribution in the case of the 10,000 MACs array, highlighting that OAs and current sources are now the dominant contributors.

By modifying the length of the sequence generated by the pattern generator, configurable bitwidths for the inputs and weights can be selected. As shown in Equation (2), the total computation time T_{tot} depends on the bit precision of the inputs and weights. Table II gives the sequence length and the recalculated energy efficiency with several configurations. The efficiency can reach up to 20,000 TOPS/W when configured for ternary inputs and weights.

V. Comparison with prior work and discussion

In this section, this work is compared to state-of-the-art digital and analog implementations of MVM In-Memory Computing techniques. All the references presented in Table III show macro efficiency only. Due to different bit precisions and technology nodes it is difficult to make a meaningful comparison

TABLE II: Evolution of the energy efficiency as a function of the number of bits for the inputs and weights.

Number of bits N_x / N_w	T_{tot}	Energy efficiency
5 / 5	$4.5\mu s$	99.2 TOPS/W
4 / 4	$0.98\mu s$	455 TOPS/W
3 / 3	$0.18\mu s$	2,477 TOPS/W
1.5 / 1.5	$0.02\mu s$	22,295 TOPS/W

of these works with the TOPS/W metric. A normalized efficiency over 1b activation and 1b weights has been calculated from the available data for each implementation to allow easier comparison.

Our prototype circuit does not include the ADC needed for the readout of the accumulation lines. To have a more realistic and fair comparison with the other works we need to add a power budget for the ADC to the measured power consumption of our circuit. Power consumptions of 10-bit SAR ADCs with ENOB of 9 bits at 200kS/s reported in the most recent available state of the art [36], [37], [38] are 57nW and 85nW. The targeted ENOB in our application is 7 to 8 bits, knowing that 5 bits would actually be sufficient for feeding the result to successive layers of the neural network. Assuming 100nW power consumption for the ADC at the required conversion rate of 222kS/s leaves a reasonable error margin knowing that the target ENOB is well below 9 bits. The last column of Table III includes this additional power consumption in the efficiency calculations.

Compared to other works and not including the ADC power consumption, our solution presents the highest efficiency at 2,480 TOPS-1b/W with the consumption scaled on a 10,000 MAC array. When taking into account the estimated ADC power consumption the efficiency reaches 81.1 TOPS/W and 2,026 TOPS-1b/W which still compares well to other references and remains among the best values for the TOPS-1b/W efficiency. This prototype chip has not been optimized for area, since flip-flop registers, from standard cell library, have been instantiated to store inputs and weights. The area associated with the memory storage can be reduced with appropriate SRAM cell design.

Finally, by leveraging time, the obtained throughputs are obviously lower than those of other references. However, this throughput is sufficient for medium-scaled neural network

TABLE III: Comparison with prior work

Reference	ISSCC'22 [23]	JSSC'22 [17]	JSSC'23 [34]	ISSCC'21 [31]	JSSC'23 [39]	ISSCC'22 [35]	This Work		
Technology	40nm	65nm	28nm	16nm	22nm	28nm	28nm		
MAC Operation	Digital RRAM	Digital CIM	Charge Domain	Charge Domain	RRAM Time Domain	Time Domain	Time Domain		
Supply (V)	0.9	0.9 - 1.5	0.9	0.8	0.8	0.65 - 0.9	0.8		
# Input Channel	-	64	16	1,152	-	64	100		
# Output Channel	-	64	16	256	-	256	4	100	
Macro Area (mm^2)	-	-	0.468	25	18	-	0.048	1.09[a]	-
Input Precision (bits)	4	2/4/6/8	4	1-8	8	4/8	5		
Weight Precision (bits)	4	4/8	4	1-8	8	4/8	5		
Output Precision (bits)	32	5	12	8	8	14/22	5		
GOPS	94.75	2,000	767.5	11,800 (4b)	142.2	4,256(4b)	0.177	4.44[a]	
GOPS/mm^2	-	-	1,640	2,670	10	-	3.69	4.07[a]	-
TOPS/W	3.79	158.7 (2b I,4b W)	94.31	121 (4b)	21.6	84.45 (4b)	15.8	99.2[a]	81.1[ab]
TOPS-1b/W	60	1,269	1,508	1,936	1,382	1,351	395	2,480[a]	2,026[ab]

[a]Estimated [b] Additional ADC consumption included

applications. For example, a VGG16 model in [17] requires 0.63 GOPS for MNIST and CIFAR-10. This throughput can be further improved by parallelizing multiple macros.

VI. Conclusion

This work shows the implementation of a time- and current-based analog in-Memory computing macro able to reach 99.2 TOPS/W for 5-bit 100×100 matrix-vector multiplications. This architecture is suitable for low-to-medium resolution embedded AI applications. Targeting tiny machine learning applications, this work exploits the trade-off between efficiency and the available time for computation.

References

[1] J. S. P. Giraldo, C. O'Connor, and M. Verhelst, "Efficient Keyword Spotting through Hardware-Aware Conditional Execution of Deep Neural Networks," in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. Abu Dhabi, United Arab Emirates: IEEE, Nov. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9035275/

[2] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the State of Neural Network Pruning?" Mar. 2020, arXiv:2003.03033 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2003.03033

[3] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," Feb. 2016, arXiv:1510.00149 [cs]. [Online]. Available: http://arxiv.org/abs/1510.00149

[4] R. Banner, Y. Nahshan, E. Hoffer, and D. Soudry, "ACIQ: Analytical Clipping for Integer Quantization of neural networks," Feb. 2022. [Online]. Available: https://openreview.net/forum?id=B1x33sC9KQ

[5] I. Hubara, Y. Nahshan, Y. Hanani, R. Banner, and D. Soudry, "Accurate Post Training Quantization With Small Calibration Sets."

[6] M. Nagel and R. A. Amjad, "Up or Down? Adaptive Rounding for Post-Training Quantization."

[7] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A White Paper on Neural Network Quantization," Jun. 2021, arXiv:2106.08295 [cs]. [Online]. Available: http://arxiv.org/abs/2106.08295

[8] E. Park, S. Yoo, and P. Vajda, "Value-Aware Quantization for Training and Inference of Neural Networks," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, vol. 11208, pp. 608–624, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-030-01225-0_36

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, arXiv:1512.03385 [cs]. [Online]. Available: http://arxiv.org/abs/1512.03385

[10] M. Nagel, M. Fournarakis, Y. Bondarenko, and T. Blankevoort, "Overcoming Oscillations in Quantization-Aware Training."

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, arXiv:1704.04861 [cs]. [Online]. Available: http://arxiv.org/abs/1704.04861

[12] C. Yuan and S. S. Agaian, "A comprehensive review of Binary Neural Network," Feb. 2022, arXiv:2110.06804 [cs]. [Online]. Available: http://arxiv.org/abs/2110.06804

[13] W. Shan, M. Yang, T. Wang, Y. Lu, H. Cai, L. Zhu, J. Xu, C. Wu, L. Shi, and J. Yang, "A 510-nW Wake-Up Keyword-Spotting Chip Using Serial-FFT-Based MFCC and Binarized Depthwise Separable CNN in 28-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, Jan. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9233931/

[14] M. Yang, C.-H. Yeh, Y. Zhou, J. P. Cerqueira, A. A. Lazar, and M. Seok, "Design of an Always-On Deep Neural Network-Based 1-μ W Voice Activity Detector Aided With a Customized Software Model for Analog Feature Extraction," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 6, pp. 1764–1777, Jun. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8693834/

[15] J.-s. Seo, J. Saikia, J. Meng, W. He, H.-s. Suh, Anupreetham, Y. Liao, A. Hasssan, and I. Yeo, "Digital Versus Analog Artificial Intelligence Accelerators: Advances, trends, and emerging designs," *IEEE Solid-State Circuits Magazine*, vol. 14, no. 3, pp. 65–79, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9864008/

[16] H. Fujiwara, H. Mori, W.-C. Zhao, M.-C. Chuang, R. Naous, C.-K. Chuang, T. Hashizume, D. Sun, C.-F. Lee, K. Akarvardar, S. Adham, T.-L. Chou, M. E. Sinangil, Y. Wang, Y.-D. Chih, Y.-H. Chen, H.-J. Liao, and T.-Y. J. Chang, "A 5-nm 254-TOPS/W 221-TOPS/mm ^{2} Fully-Digital Computing-in-Memory Macro Supporting Wide-Range Dynamic-Voltage-Frequency Scaling and Simultaneous MAC and Write Operations," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2022, pp. 1–3. [Online]. Available: https://ieeexplore.ieee.org/document/9731754/

[17] J. Yue, Y. Liu, Z. Yuan, X. Feng, Y. He, W. Sun, Z. Zhang, X. Si, R. Liu, Z. Wang, M.-F. Chang, C. Dou, X. Li, M. Liu, and H. Yang, "STICKER-IM: A 65 nm Computing-in-Memory NN Processor Using Block-Wise Sparsity Optimization and Inter/Intra-Macro Data Reuse," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 8, pp. 2560–2573, Aug. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9714739/

[18] B. Murmann, "Mixed-Signal Computing for Deep Neural Network Inference," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 3–13, Jan. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9197673/

[19] Y. Toyama, K. Yoshioka, K. Ban, S. Maya, A. Sai, and K. Onizuka, "An 8 Bit 12.4 TOPS/W Phase-Domain MAC Circuit for Energy-Constrained Deep Learning Accelerators," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2730–2742, Oct. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8771205/

[20] A. Sayal, S. S. T. Nibhanupudi, S. Fathima, and J. P. Kulkarni, "A 12.08-TOPS/W All-Digital Time-Domain CNN Engine Using Bi-Directional Memory Delay Lines for Energy Efficient Edge Computing," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 60–75, Jan. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8867969/

[21] S. Gweon, S. Kang, K. Kim, and H.-J. Yoo, "FlashMAC: A Time-Frequency Hybrid MAC Architecture With Variable Latency-Aware Scheduling for TinyML Systems," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 10, pp. 2944–2956, Oct. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9813564/

[22] S. Jain, L. Lin, and M. Alioto, "±CIM SRAM for Signed In-Memory Broad-Purpose Computing From DSP to Neural Processing," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 10, pp. 2981–2992, Oct. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9481107/

[23] M. Chang, S. D. Spetalnick, B. Crafton, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40nm 60.64TOPS/W ECC-Capable Compute-in-Memory/Digital 2.25MB/768KB RRAM/SRAM System with Embedded Cortex M3 Microprocessor for Edge Recommendation Systems," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2022, pp. 1–3. [Online]. Available: https://ieeexplore.ieee.org/document/9731679/

[24] E. R. Hsieh, X. Zheng, B. Q. Le, Y. C. Shih, R. M. Radway, M. Nelson, S. Mitra, and S. Wong, "Four-Bits-Per-Memory One-Transistor-and-Eight-Resistive-Random-Access-Memory (1T8R) Array," *IEEE Electron Device Letters*, vol. 42, no. 3, pp. 335–338, Mar. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9336666/

[25] D. Bankman and B. Murmann, "An 8-bit, 16 input, 3.2 pJ/op switched-capacitor dot product circuit in 28-nm FDSOI CMOS," in *2016 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. Toyama, Japan: IEEE, Nov. 2016, pp. 21–24. [Online]. Available: http://ieeexplore.ieee.org/document/7844125/

[26] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-Tile 2.4-Mb In-Memory-Computing CNN Accelerator Employing Charge-Domain Compute," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8660469/

[27] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks," *IEEE Journal of Solid-State Circuits*, pp. 1–11, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8959407/

[28] J. Zhang, Z. Wang, and N. Verma, "In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, Apr. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7875410/

[29] M. Kang, S. K. Gonugondla, A. Patil, and N. R. Shanbhag, "A Multi-Functional In-Memory Inference Processor Using a Standard 6T SRAM Array," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 2, pp. 642–655, Feb. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8246704/

[30] A. Biswas and A. P. Chandrakasan, "CONV-SRAM: An Energy-Efficient SRAM With In-Memory Dot-Product Computation for Low-Power Convolutional Neural Networks," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 217–230, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8579538/

[31] H. Jia, M. Ozatay, Y. Tang, H. Valavi, R. Pathak, J. Lee, and N. Verma, "15.1 A Programmable Neural-Network Inference Accelerator Based on Scalable In-Memory Computing," in *2021 IEEE International Solid- State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2021, pp. 236–238. [Online]. Available: https://ieeexplore.ieee.org/document/9365788/

[32] D. Del Corso and L. Reyneri, "Mixing analog and digital techniques for silicon neural networks," in *IEEE International Symposium on Circuits and Systems*. New Orleans, LA, USA: IEEE, 1990, pp. 2446–2449. [Online]. Available: http://ieeexplore.ieee.org/document/112505/

[33] S. Cosemans, B. Verhoef, J. Doevenspeck, I. A. Papistas, F. Catthoor, P. Debacker, A. Mallik, and D. Verkest, "Towards 10000TOPS/W DNN Inference with Analog in-Memory Computing – A Circuit Blueprint, Device Options and Requirements," in *2019 IEEE International Electron Devices Meeting (IEDM)*. San Francisco, CA, USA: IEEE, Dec. 2019, pp. 22.2.1–22.2.4. [Online]. Available: https://ieeexplore.ieee.org/document/8993599/

[34] J.-W. Su, Y.-C. Chou, R. Liu, T.-W. Liu, P.-J. Lu, P.-C. Wu, Y.-L. Chung, L.-Y. Hong, J.-S. Ren, T. Pan, C.-J. Jhang, W.-H. Huang, C.-H. Chien, P.-I. Mei, S.-H. Li, S.-S. Sheu, S.-C. Chang, W.-C. Lo, C.-I. Wu, X. Si, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, and M.-F. Chang, "A 8-b-Precision 6T SRAM Computing-in-Memory Macro Using Segmented-Bitline Charge-Sharing Scheme for AI Edge Chips," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 3, pp. 877–892, Mar. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9896828/

[35] P.-C. Wu, J.-W. Su, Y.-L. Chung, L.-Y. Hong, J.-S. Ren, F.-C. Chang, Y. Wu, H.-Y. Chen, C.-H. Lin, H.-M. Hsiao, S.-H. Li, S.-S. Sheu, S.-C. Chang, W.-C. Lo, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, C.-I. Wu, and M.-F. Chang, "A 28nm 1Mb Time-Domain Computing-in-Memory 6T-SRAM Macro with a 6.6ns Latency, 1241GOPS and 37.01TOPS/W for 8b-MAC Operations for Edge-AI Devices," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 65, Feb. 2022, pp. 1–3, iSSN: 2376-8606.

[36] H.-Y. Tai, Y.-S. Hu, H.-W. Chen, and H.-S. Chen, "11.2 A 0.85fJ/conversion-step 10b 200kS/s subranging SAR ADC in 40nm CMOS," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2014, pp. 196–197. [Online]. Available: http://ieeexplore.ieee.org/document/6757397/

[37] H. S. Bindra, A.-J. Annema, S. M. Louwsma, and B. Nauta, "A 0.2 - 8 MS/s 10b flexible SAR ADC achieving 0.35 - 2.5 fJ/conv-step and using self-quenched dynamic bias comparator," in *2019 Symposium on VLSI Circuits*. Kyoto, Japan: IEEE, Jun. 2019, pp. C74–C75. [Online]. Available: https://ieeexplore.ieee.org/document/8778093/

[38] B. Murmann, "ADC Performance Survey 1997-2021." 1997. [Online]. Available: https://github.com/bmurmann/ADC-survey.

[39] J.-M. Hung, T.-H. Wen, Y.-H. Huang, S.-P. Huang, F.-C. Chang, C.-I. Su, W.-S. Khwa, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, Y.-D. Chih, T.-Y. J. Chang, and M.-F. Chang, "8-b Precision 8-Mb ReRAM Compute-in-Memory Macro Using Direct-Current-Free Time-Domain Readout Scheme for AI Edge Devices," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 1, pp. 303–315, Jan. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9874915/

Kévin Hérissé received his engineering degree in 2018 from the Institut Supérieur d'Electronique et du Numérique (ISEN), Lille, France and his Ph.D. degree from the University of Lille, Lille, France in 2022. He was the chair of the Lille IEEE Student Branch from 2020 to 2022 which won the best student branch awards in 2022 given by the IEEE France section. Currently, he was appointed chair of the Young Professional Affinity Group. He was the recipient of the Lille Catholic University thesis awards (2nd place) in 2021 and was awarded the 1st place at the regional final of the Hauts-de-France region of "Ma thèse en 180 secondes". His current research interests include In-Memory Computing, event-driven and ultra-low power circuits for embedded machine learning.

Bruno Stefanelli (S'86–M'87) received the Engineering Diploma degree from the Institut Supérieur d'Electronique du Nord (ISEN), Lille, France, in 1986, and the Ph.D. degree from the University of Lille, Lille, France, in 1992.,In 1992, he joined the Analog/RF IC Design Group, Institut d'Electronique, de Microelectronique et de Nanotechnologies, Villeneuve-d'Ascq, France, where he was involved in continuous and discrete-time analog circuits, data converters, and RF-MEMS. He is currently a Professor with ISEN. His current research interests include RF and millimeter-wave circuits for telecommunications.

Benoit Larras was born in Nancy, France, in 1988. He received the engineering and master's degrees in telecommunications, in 2012, and the Ph.D. degree in electrical engineering from IMT Atlantique, Brest, France, in 2015. He is currently an Associate Professor with Junia, Lille, France, leading the Electronic Team. His research interests include analog/mixed-signal IC design and circuit implementation of neural networks and associative memories, in the context of near-sensor computing and edge computing. He is the co-recipient of the Best Paper at the IEEE AICAS2020 Conference.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems--I: Regular Papers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSI.2024.3480154

Andreas Kaiser (Senior Member, IEEE) received the Engineering Diploma degree from the Institut Supérieur d'Electronique du Nord (ISEN), Lille, France, in 1984, and the Ph.D. and HDR degrees from the University of Lille, Lille, in 1990 and 1998, respectively. In 1990, he joined the Centre National de la Recherche Scientifique (CNRS), Paris, France, where he was responsible for the Analog/RFIC Design Group at the Institut d'Electronique, de Microelectronique et de Nanotechnologies (IEMN), Lille. He has been the Co-Director of the IEMN-STMicroelectronics Common Laboratory from 2002 until 2012. From 2012 until 2017, he was the Dean of the JUNIA-ISEN Engineering faculty and from 2017 to 2023 Scientific Director of the Junia Graduate School of Engineering. He has been the advisor of 38 completed Ph.D. students. He has published more than 160 publications in peer-reviewed journals and conferences. He holds 17 patents. His research interests are continuous- and discrete-time analog circuits, data converters, analog design automation, RF-MEMS, and RF circuits.,Dr. Kaiser served as the TPC Chair for the European Solid-State Circuits Conference in 1995 and 2005. He has been a Guest Editor and an Associate Editor of the IEEE Journal of Solid- State Circuits.

Antoine Frappé (Senior Member, IEEE) graduated from the Institut Supérieur d'Electronique du Nord (ISEN), Lille, France, in 2004. He received the M.Sc., Ph.D., and HDR (French highest academic degree) degrees in electrical engineering from the University of Lille, Lille, France, in 2004, 2007, and 2019, respectively.,Since 2004, he has been a member of the Silicon Microelectronics Group, Institute of Electronics, Microelectronics, and Nanotechnologies (IEMN), Villeneuve-d'Ascq, France. He obtained a Fulbright grant in 2008 to pursue research in communication systems at the Berkeley Wireless Research Center (BWRC), University of California at Berkeley, Berkeley, CA, USA. He is currently an Associate Professor at Junia ISEN, Lille, leading the Electronics Team. His research interests concern digital RF transmitters, high-speed converters, mixed-signal design for RF and millimeter-wave (mmW) communication systems, energy-efficient integrated systems, and event-driven and neuro-inspired circuits for embedded machine learning.,Dr. Frappé was a co-recipient of the Best Student Paper Award at the 2011 Symposium on VLSI Circuits, the Best Paper Award at the 2020 IEEE AICAS Conference, and the Industrial Best Paper Award at the 2021 IEEE RFIC Symposium. He plays an active role as a Board Member of the France Section of the IEEE Circuits and Systems Society and a Counselor of the IEEE Lille Student Branch.