2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS) | 979-8-3503-8363-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AICAS59952.2024.10595976

Evaluating an Analog Main Memory Architecture for All-Analog In-Memory Computing Accelerators

Kazybek Adam *, Dipesh Monga *, Omar Numan *, Gaurav Singh *, Kari Halonen *, and Martin Andraud * †
*Aalto University, Department of Electronics and Nanoengineering, Finland {firstname.lastname}@aalto.fi
†UC Louvain, ICTEAM {martin.andraud}@uclouvain.be

***Abstract*—Analog in memory Computing (IMC) has emerged as a promising method to accelerate deep neural networks (DNNs) on hardware efficiently. Yet, analog computation typically focuses on the multiply and accumulate operation, while other operations are still being computed digitally. Hence, these mixed-signal IMC cores require extensive use of data converters, which can take a third of the total energy and area consumption. Alternatively, all-analog DNN computation is possible but requires increasingly challenging analog storage solutions, due to noise and leakage of advanced technologies. To enable all-analog DNN acceleration, this work demonstrates a feasible IMC architecture using an efficient analog main memory (AMM) cell. The proposed AMM cell is 42x and 5x more power and area efficient than a baseline analog storage cell. An all-analog architecture using this cell achieves potential efficiency gains of 15x compared with a mixed-signal IMC core using data converters.**

***Index Terms*—Analog in memory Computing, Analog Memory**

I. Introduction

Dedicated hardware accelerators computing Deep Neural Networks (DNN) on edge devices are critical for future breakthroughs in wireless communication, ubiquitous sensing, health, and more. In that regard, mixed-signal In-memory computing (IMC) accelerators promise drastic improvements over classical processors [1], realizing the most power-hungry DNN operation, multiply and accumulate (MAC), in analog and in memory to limit data transfer. MAC is computed using basic Kirchoff's laws, i.e., multiplication using conductances or capacitances, and native addition in current or charge domains [2]. As DNNs tolerate low-resolution computation (3-8 bits), analog is potentially more efficient [3]. Fig. 1(a) depicts a mixed-signal IMC core. All operations except MAC are performed digitally, requiring extensive data conversions at the input (DACs) and output (ADCs), taking up to 40-50% and 30-40% of the core's power and area [4], [5].

To solve the data conversion bottleneck, Fig. 1(b) depicts all-analog IMC architecture [6]–[8]. Particularly, all-analog accelerators require storing every data point in the analog domain, which can be challenging in modern CMOS technologies where leakage and noise sensitivities are increasingly dominant. To tackle this challenge, this work presents a simple analog main memory (AMM) cell, evaluating the feasibility of a low-area and low-leakage analog memory. It employs an area and power-constrained source follower structure as a sample-and-hold buffer, with leakage compensation mechanisms to limit droop rates. We derive a design methodology to attain realistic target specifications according to the model/architecture, such as the AIMC core size, DNN layer, etc.

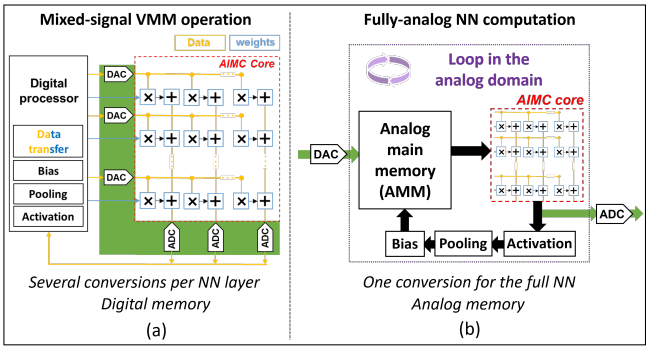


Fig. 1: (a) Illustration of a typical AIMC core; (b) fully analog NN accelerator implementation

Case study: Throughout this work, the configuration of Fig. 2 is used as an example. The dimensions are tailored to facilitate the explanation of a relatively complex convolution operation computation but could be generalized to any layer.

II. Mixed-signal versus all-analog IMC

Fig. 1(a) shows a mixed-signal IMC core [4]. DACs convert every input vector to analog, typically requiring one DAC per row. After computation, the data is transferred back to digital through ADCs. This process happens multiple times per DNN layer. Conventional ADC topologies for AIMC are either [9]: (1) power-efficient but slow Successive-Approximation Register (SAR) ADCs, which can be used per column or multiplexed on a few columns; or (2) faster but less power-efficient flash ADCs, which can be multiplexed over a larger number of columns. Data converters are a crucial bottleneck for IMC. Using data from [4], they account for 56% of the accelerators' core power (22% for DACs and 34% for ADCs) while occupying 43% of the core area. Assuming a constant operation for a core size of 288x256 (used in the case study), data conversion would require 0.12W (64μW$\times$32*9 for DACs + 400μW$\times$32*8 for ADCs). Alternatively, Fig.1(b) depicts a solution where data is stored directly in the analog domain in between computations [6]–[8]. For instance, [7] demonstrates an all-analog computation, including activation and pooling,

Authorized licensed use limited to: Univ Catholique de Louvain/UCL. Downloaded on December 20,2024 at 15:44:10 UTC from IEEE Xplore. Restrictions apply.

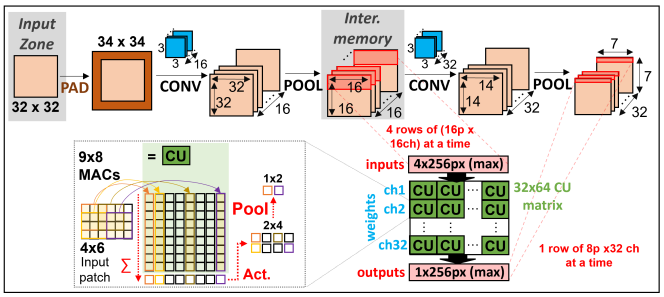


Fig. 2: Proposed computation architecture

avoiding using data converters. Intermediate results are stored in analog as well, using capacitors and buffers, yet the storage is not fully evaluated. The main objective of this work is to do such evaluation by establishing design choices and proposing an analog storage solution.

A. Computation algorithm and AMM integration

The computation strategy is crucial for the AMM as it should minimize the retention time. Fig. 2 illustrates a solution for that, based on the Im2Col algorithm [10]. Im2Col provides one full row of the output map per cycle, duplicating weights in the IMC core and performing parallel neighboring convolutions. Assume an input image of size $N_I \times N_I \times C_I$. The MAC units of the IMC core are divided in a matrix of $N_C \times N_R$ Computation Units (CU). One CU is a group of 9×8 MAC units because, with a 3x3 kernel, a 4x6 input patch contains 8 convolutions, each using 9 pixels. To complete the convolution on bigger images, pixels overlap over multiple CUs; hence, each CU uses 4 rows of 4 pixels. N_C columns of CUs compute $4 \times (4 \times N_C)$ pixels. One CU row computes one channel. It is assumed that activation and pooling are performed in analog, as in [7]. After pooling and internal channel combinations, an output feature map of size $(N_I/2)$-$1 \times (N_I/2)$-$1 \times C_O$, with C_O the number of output channels, is created, one row per clock cycle. In the case study, data enters the CU matrix in blocks of 4 rows of 256 pixels, corresponding to 16 channels of 16 pixels. The weights of each convolution channel are mapped to one row of CUs. After pooling, the output becomes 1 row of 256 pixels, corresponding to 8 pixels of 32 channels (with one blank pixel per channel as we require only 7).

a) AMM size: The AMM is divided into two zones: the input zone containing the input image and the Intermediate Memory (IM) zone storing all intermediate results after pooling. The size of the IM depends on the DNN structure. Typically, the first convolution layer leads to the largest output volume. For this case study, the input zone contains 34x34=1156 pixels or 32x32=1024 (considering padding on hardware). The IM size is 16x16x16=4096 pixels.

b) Storage time: The storage time relates to the AIMC core performance (duration of one computation cycle) and the mapping type (number of cycles). Hence, it depends on the layers to be executed. For example, assuming the architecture in Fig. 2, 4 input rows are simultaneously computed, with a stride of 2. It generates one row of the output image at a time. hence, the first convolution layer requires only 16 cycles.

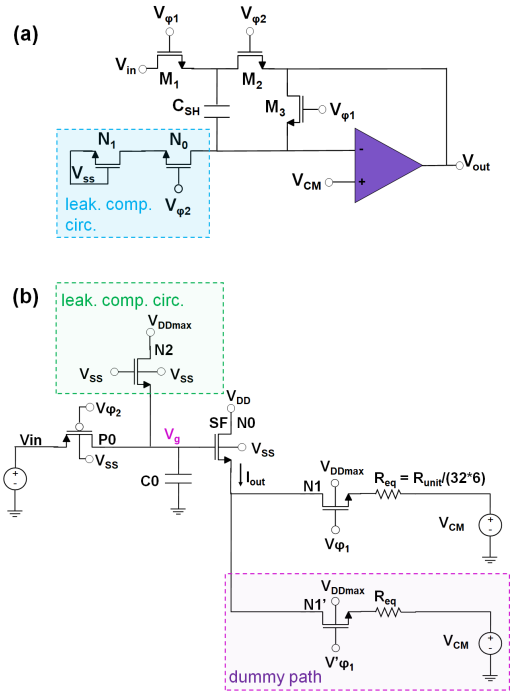


Fig. 3: (a) Traditional SH, (b) SF-based SH circuits, each with respective leakage compensation schemes

B. Baseline design for comparison

Fig. 3(a) shows a sample and hold (SH) topology used as a baseline. It comprises a storage capacitor, a two-stage Miller compensated operational amplifier (opamp) to buffer the sampled signal to the IMC core, and various switches for offset-compensation mechanisms. The common-mode voltage (V_{CM}) has been set to the middle of the supply range. This architecture achieves a droop rate of 10 μV/μs range with a power consumption of 64 μW (see Table I). For the case study, with 1024+4096=5120 AMM cells required, the AMM would consume 328 mW, more than the 120 mW of the mixed-signal AIMC. Hence, a viable all-analog AIMC approach needs a more efficient AMM.

III. Proposed Analog Main Memory

Fig. 3(b) depicts the proposed AMM. The circuit comprises a Source Follower (SF) configuration $N0$, a storage capacitor C0, and two switches $P0$ and $N1$. $N0$ is used as a buffer between the AMM storage and the AIMC core. The input voltage is sampled through $P0$ and held in the capacitor. When needed in the AIMC core, the stored value is routed with $N1$.

A. Design methodology

Load. The starting design point is one IMC column, with R_{unit} being the unit resistance of each element. The worst-case load is $R_{eq} = R_{unit}/N_r$, where N_r is the number of rows. Without loss of generality, we assume the resistive

Authorized licensed use limited to: Univ Catholique de Louvain/UCL. Downloaded on December 20,2024 at 15:44:10 UTC from IEEE Xplore. Restrictions apply.

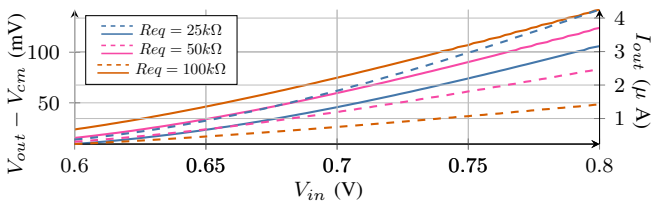


Fig. 4: Output voltage and current ranges varying V_{in} and R_{eq} (Vcm=500mV, no switch $N1$): $V_{out} - V_{cm}$ (—), I_{out} (- -)

element to be an R-2R Multiplying DAC (MDAC) [11], where $R = R_{unit}$. This type of architecture can be competitive for IMC using high-density resistors [12], [13].

Input range. The input range V_{in} is chosen so that the SF transistor operates in subthreshold (for energy efficiency) while avoiding cut-off to keep a linear mapping between inputs and outputs. If the VMM structure accumulates positive and negative MAC values on separate paths, the dynamic range of the column is larger, yet it requires subtraction at the bottom of each column. Alternatively, if both positive and negative MAC values accumulate on the same path, the mathematical zero can be mapped to the middle of the range, reducing the dynamic range but removing the need for subtraction.

Output range. Ideally, the voltage output range V_{out} should equal V_{in}. The range of the output current I_{out} per weight unit is:

$$\frac{V_{out,min} - V_{CM}}{R_{unit}} \leq I_{out} \leq \frac{V_{out,max} - V_{CM}}{R_{unit}} \quad (1)$$

Where $V_{out} = [V_{out,min}; V_{out,max}]$ and V_{cm} is the non-zero common mode voltage. Using the case study, to obtain 0 current when the input is 0, a target is $V_{cm} = V_{out,min}$, giving $0A \leq I_{out} \leq 40nA$. In the worst case, R_{unit} is replaced by R_{eq}, leading to $0A \leq I_{out} \leq 7.68\mu A$. However, these ranges are difficult to attain in practice.

Summary of the proposed design steps (S):

(S1): Choose a load and input range, starting with the largest dynamic range, avoiding cutoff for the SF transistor.

(S2): Select a minimal SF allowing to drive R_{eq} in the worst case (where R_{eq} depends on R_{unit} and the core size).

(S3): Find an optimal value for V_{CM}. The starting value can be found by fixing $V_{in} = V_{in,min}$ and choosing the value of V_{CM} that minimizes $I_{out,min}$. The final value of V_{cm} depends on acceptable leakage current levels (from $N1$) and on the next stages, e.g., summing amplifier, pooling and/or activation.

(S4): Sweep V_{in} and observe V_{out}. If the desired V_{out} range cannot be achieved, repeat steps S1-S3, increasing the SF transistor and/or reducing the output range to keep a linear relationship. Increasing R_{eq}, if possible, improves the output voltage range (see Fig. 4), but decreases the output currents.

(S5): Evaluate the droop rate. If needed, add leakage compensation to increase the hold time (see section III.B)

(S6): Perform system optimization by changing the transistor (e.g. using lower threshold voltages).

(S7): Optimize the voltages and timing of control signals.

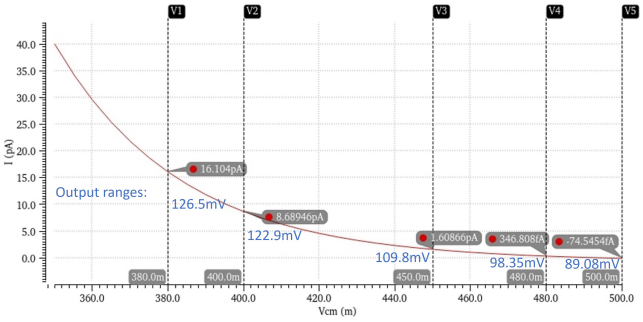


Fig. 5: Leakage currents vs V_{cm} through SF **output sw.** $N1$

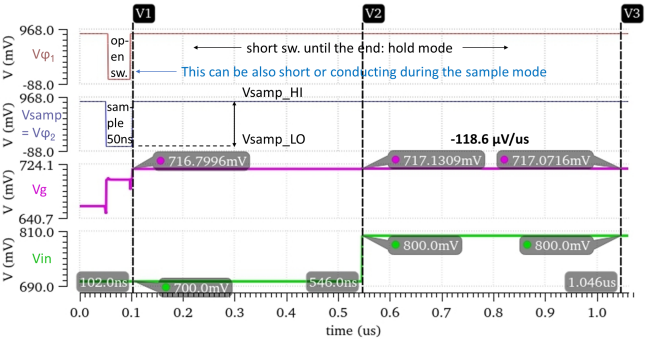


Fig. 6: Low-leak. high-thresh. **input switch** $P0$ back biased at Vss: Vin transitions from 700mV to 800mV during hold

(S8): During the layout, increase the sampling capacitor if placed on top of the cell. This will increase the performance of the cell without increasing the area.

B. Leakage and offset compensations

a) Leakage Compensation Circuit: leakage currents shorten the hold time of the AMM. Hence, both the baseline and proposed circuits (Fig. 3) include leakage compensation, based on [14]. For the baseline circuit, the leakage compensation is done through an additional path, turned on during the hold phase (V_{ϕ_2} is high), generating an opposite dummy current to the capacitor leakage. For the proposed circuit, the compensation is done by inserting a single transistor ($N2$). It generates a small current that keeps charging the sampling capacitor, compensating for leakage.

b) Offset compensation with "dummy path": the output switch $N1$ introduces leakage currents during switching (800pA to 4pA), which may disrupt the system. Also, the SF samples/holds at different operating regions if $N1$ is off during sampling, causing hold step error (-5mV). These issues can be significantly mitigated by creating an identical "dummy path" (Fig. 3(b)), which gets disconnected when driving the real load, yet causing static power consumption.

IV. Results

A. Design steps

S1 and S2: input range and load. The chosen IMC core configuration uses $R_{unit} = 5M\Omega$ to emulate the use of high-

Authorized licensed use limited to: Univ Catholique de Louvain/UCL. Downloaded on December 20,2024 at 15:44:10 UTC from IEEE Xplore. Restrictions apply.

density resistors. Assuming 32 CU rows, $R_{eq} = R_{unit}/(6 * 32) = R_{unit}/192$ where 6 is the maximum of MAC units connected to a single input (middle 2x2 pixels in a 4x6 input patch). The mathematical zero and unity map to resp. 600mV and V_{DD} (800mV), leading to V_{in}=[0.6V;0.8V].

S3 and S4: output range. Fig.5 plots the AMM leakage current against V_{cm} and reports the attainable V_{out}, showing a clear trade-off between V_{out} and leakage. As a realistic target, we chose $I_{out,min}$ 1nA, and we decreased $I_{out,max}$ to 3.84μA. Achieving a 100mV range requires $V_{cm} \leq$ 480mV but introduces at least 400fA of leakage for one cell, which is acceptable for our purpose. Also, increasing R_{eq} can reduce leakage currents but affects the output ranges (Fig.4).

S5: droop rate. Fig.6 shows a hold step and droop rate evaluation. Here, the gate capacitance of the SF transistor $N0$ is used as a sampling capacitor. The final sampling capacitor is added in step 8. The droop rate (118.6μV/μs) and hold step (16.8mV) are calculated when V_{in} just transitioned from the originally sampled V_{in} of 700mV to 800mV (aimed at evaluating the isolation capabilities of the switch).

S6: transistor types. After iterating over previous steps, Table I summarizes the SF performance of two configurations, the first with a regular SF transistor (denoted REG) and the other with a super low threshold voltage transistor (denoted LOW). The LOW configuration can work with V_{CM}=500mV to obtain a 100mV output range with 75fA leakage current through the output switch; it has 1.5x less leakage through the SF gate. Yet, the REG configuration is more power-efficient. For switch selection, $P0$ is a high threshold with back bias at V_{SS} to lower the drop rate traded off with a higher hold step. $N1$ is a low-threshold with back bias at V_{DD} (or V_{DDmax}) to maximize the output range and limit leakage.

S7: selection of control signals. The timing control of the circuit also plays a role in the AMM performance; specifically, we consider the sampling control signals $V_{samp,low}$ and $V_{samp,high}$, their rising and falling times t_{rise} and t_{fall} and the supply voltage of the SF transistor denoted V_{SF}. The optimal configuration is evaluated, varying these parameters. From these simulations, we observed that: (1) changing V_{SF} does not impact the droop rate significantly, so it can be V_{DD}; (2) droop rates can be reduced by 4-5x further by increasing the maximum sampling voltage $V_{samp,high}$ from V_{DD} to $V_{DD} + 10\%$, yet requiring an extra voltage source; (3) rise and fall times affect the droop rate and hold step.

S8: layout and final capacitance selection. Until this point, the gate capacitance of the SF transistor was used for sampling. Droop rate and hold mode offsets are optimized using an additional sampling capacitor. It can be a metal capacitor placed on top of the transistor structure, without impacting the final area. In this design example, a physical capacitor with 15x the minimum unit capacitance is added with no compromise in the cell area. Increasing the sampling capacitor proportionally decreases the average droop rate values and the absolute difference in droop rates from two neighboring cells. The same is true for the hold-mode offsets.

TABLE I: Performance comparison with target specifications

(@ TT 27*C)	**Baseline offset-comp.**	**Proposed SF-Based configuration**		
		(REG)	**(LOW)**	*Target*
Power (μW)	63.8	1.5	1.8	1.15
Area: (μm x μm)	18.6 x 6.9	2 x 12.7 (w. dum. p.)	2 x 12.7 (w. dum. p.)	2 x 5
Droop rate ($\mu V/\mu s$)	$\leq$ 3	$\leq$ 12	$\leq$ 16	$\leq$ 10
Hold step (mV)	$\leq$ 3.4	$\leq$ 3	$\leq$ 3	$\leq$ 3.4
V_{in} (mV)	[260, 540]	[600, 800]	[600, 800]	[600, 800]
V_{out} (mV)	[260, 540]	[494, 593]	[549, 649]	[500, 600]
I_{out} (μA)	[-11.7, 11.7]	[0.57, 4.51]	[1.95, 5.95]	[0.19, 3.84]
V_{cm} (mV)	400	480	500	500

B. Comparison results

Table I compares the proposed AMM and the baseline, both with leakage compensation. The baseline achieves a 280mV dynamic range with a direct input-output mapping. The droop rate is $10\mu V/\mu$s and $3\mu V/\mu$s, resp. with and without leakage compensation. It occupies an area of $\approx 19\times 7\mu m^2$ and consumes around $64\mu W$. In comparison, the proposed design consumes 42.5x less power ($1.5\mu W$) and 5x less area ($\approx 2\times 13\mu m^2$). With leakage compensation, the droop rate is similar, depending on the used voltages for the compensation ($<24\mu V/\mu$s and $[-12, +2]\mu V/\mu$s, resp. with and without compensation), showing the effectiveness of the compensation. The energy and area decrease is partly done at the expense of reduced dynamic range, resp. 200mV V_{in} and 100mV V_{out}.

a) Process and mismatch variations: We evaluated the mismatch between neighboring AMM cells by performing Monte-Carlo mismatch simulations. The results are gathered for three transistor sizes (minimal length and width of 10, 15, or 20 μm), for V_{in} varied from 600mV to 800mV. Mean and standard deviations (σ) are reported for 1000 simulations. Table II shows that for $R_{eq} = 25k\Omega$, a higher width does not significantly improve matching beyond W=10μm.

TABLE II: Vs0 - Vs1 (V) mean ± 1σ, 2σ, 3σ values after 1000 MC simulations (Req=25kΩ cases).

W/L / Vin	W=10μm / Lmin	W=15μm / Lmin	W=20μm / Lmin
Vin=600mV	28.99uV ± 893.324uV, 1.79mV, 2.68mV	33.71uV ± 1.04mV, 2.08mV, 3.12mV	37.41uV ± 1.16mV, 2.315mV, 3.47mV
Vin=700mV	77.78uV ± 2.35mV, 4.7mV, 7.05mV	76.47uV ± 2.32mV, 4.64mV, 6.96mV	76.30uV ± 2.33mV, 4.65mV, 6.98mV
Vin=800mV	102.31uV ± 3.06mV, 6.11mV, 9.165mV	96.13uV ± 2.89mV, 5.78mV, 8.67mV	93.37uV ± 2.82mV, 5.65mV, 8.47mV

b) Power consumption for a complete implementation: Using the proposed cell for an all-analog architecture (assuming the same 5120 AMM cells) the AMM would consume 7.68mW, which is a 15.7x less than the mixed-signal approach. Yet, implementing a full architecture like the one in Fig. 2 involves design specificities that must be carefully considered.

V. Conclusions

This paper presented the design exploration of an analog main memory cell toward the development of fully analog AI accelerators. It shows that by designing a simple and efficient analog storage cell, all-analog IMC architectures could be more energy-efficient than mixed signal IMC implementations.

Authorized licensed use limited to: Univ Catholique de Louvain/UCL. Downloaded on December 20,2024 at 15:44:10 UTC from IEEE Xplore. Restrictions apply.

VI. Acknowledgements

This work is supported by Academy of Finland projects EHIR (grant 13334487) and WHISTLE (grant 332218)

References

[1] N. Verma, H. Jia, H. Valavi, Y. Tang, M. Ozatay, L.-Y. Chen, B. Zhang, and P. Deaville, "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 43–55, 2019.

[2] 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.

[3] 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, 2021.

[4] H. Wang, R. Liu, R. Dorrance, D. Dasalukunte, D. Lake, and B. Carlton, "A charge domain sram compute-in-memory macro with c-2c ladder-based 8-bit mac unit in 22-nm finfet process for edge inference," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 4, pp. 1037–1050, 2023.

[5] N. R. Shanbhag and S. K. Roy, "Comprehending in-memory computing trends via proper benchmarking," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*, 2022, pp. 01–07.

[6] J. Lim, M. Choi, B. Liu, T. Kang, Z. Li, Z. Wang, Y. Zhang, K. Yang, D. Blaauw, H.-S. Kim, and D. Sylvester, "Aa-resnet: Energy efficient all-analog resnet accelerator," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2020, pp. 603–606.

[7] J.-O. Seo, M. Seok, and S. Cho, "ARCHON: A 332.7tops/w 5b variation-tolerant analog CNN processor featuring analog neuronal computation unit and analog memory," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, 2022, pp. 258–260.

[8] R. Nägele, J. Finkbeiner, V. Stadtlander, M. Grözing, and M. Berroth, "Analog multiply-accumulate cell with multi-bit resolution for all-analog ai inference accelerators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–13, 2023.

[9] H. Jiang, W. Li, S. Huang, S. Cosemans, F. Catthoor, and S. Yu, "Analog-to-digital converter design exploration for compute-in-memory accelerators," *IEEE Design & Test*, vol. 39, no. 2, pp. 48–55, 2022.

[10] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft, 2006.

[11] R. Kier, R. Harrison, and R. Beer, "An mdac synapse for analog neural networks," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 5, 2004, pp. V–V.

[12] W. Choi, M. Kwak, S. Heo, K. Lee, S. Lee, and H. Hwang, "Hardware neural network using hybrid synapses via transfer learning: Wox nano-resistors and tiox rram synapse for energy-efficient edge-ai sensor," in *2021 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2021, pp. 23–1.

[13] D. Saito, T. Kobayashi, H. Koga, N. Ronchi, K. Banerjee, Y. Shuto, J. Okuno, K. Konishi, L. Di Piazza, A. Mallik, J. Van Houdt, M. Tsukamoto, K. Ohkuri, T. Umebayashi, and T. Ezaki, "Analog in-memory computing in fefet-based 1t1r array for edge ai applications," in *2021 Symposium on VLSI Circuits*, 2021, pp. 1–2.

[14] J. Salomaa, S. Jamali-Zavareh, M. Pulkkinen, S. S. Chouhan, and K. Halonen, "Ultra-low power incremental delta-sigma adc for energy harvesting sensor applications," *Analog Integrated Circuits and Signal Processing*, vol. 92, pp. 369–382, 2017.

Authorized licensed use limited to: Univ Catholique de Louvain/UCL. Downloaded on December 20,2024 at 15:44:10 UTC from IEEE Xplore. Restrictions apply.