A 105 MS/s 4.18 nJ/inference 0.765 mm^2 Mixed-Signal Streaming ML Engine for Wireless Baseband in 12LP+ FinFET

Oscar Castañeda, Lavinia Recchioni, Sueda Taner, Thomas Burger, Jérémy Guichemerre, and Christoph Studer

Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland

***Abstract*—We present SIXPPAC, the first streaming machine-learning engine for real-time integrated sensing and communication (ISAC) in wireless systems. SIXPPAC integrates two SAR ADCs and a digital binary-weight six-layer neural network (NN). The 0.765 mm^2 12 nm FinFET engine stores all NN parameters on chip, enabling sustained 105 MS/s processing of 8b I/Q baseband samples at 0.8 V. By deploying bipolar inner products using AND instead of XNOR gates, SIXPPAC consumes 4.18 nJ/inference (0.42 POPS/W) at 0.8 V. We evaluate SIXPPAC on three example applications to demonstrate its versatility.**

I. Introduction

Machine learning (ML) is widely used in wireless communication systems, improving high-level physical-layer tasks such as channel estimation, data detection, forward error correction, and even positioning [1], [2]. In contrast, ML remains largely unexplored for low-level baseband tasks that operate directly on received in-phase and quadrature (I/Q) samples. Many such tasks (e.g., blind SNR estimation, channel busy rate detection, and time synchronization in frequency-selective channels) are in principle ill-posed, with no unique, optimal solution. Practical approaches therefore exploit specific signal properties: For instance, the M2M4 method [3] for blind SNR estimation relies on the noise being Gaussian while the signal is not, whereas OFDM systems rely on a cyclic prefix to tolerate timing deviations within its guard interval [4]. These examples illustrate that classical methods rely on structural assumptions to yield practical solutions. Thus, ML offers a promising alternative, as it can learn signal properties directly from data.

However, deploying ML in this setting is challenging: Supporting low-level baseband tasks requires processing hundreds of millions of I/Q samples per second under strict latency constraints. These tight requirements rule out conventional large neural networks (NNs) due to their reliance on off-chip data movement. Instead, these demands call for lightweight, domain-specific NNs that can be mapped to high-throughput spatial architectures with compact on-chip storage, while remaining expressive enough to support various baseband tasks.

This work has received funding from the Swiss State Secretariat for Education, Research, and Innovation (SERI) under the SwissChips initiative, and has also been supported in part through the Rigoletto project that has received funding from Chips Joint Undertaking (CHIPS-JU) under Grant Agreement 101194371. The authors would like to thank GlobalFoundries for providing silicon fabrication through the 12LP+ University Partnership Program. Contact author: O. Castañeda (e-mail: caoscar@ethz.ch)

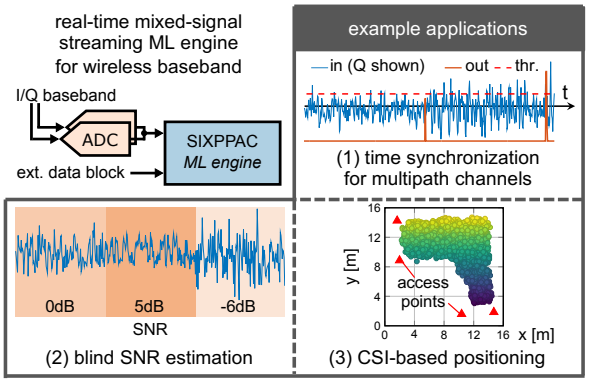


Fig. 1. Overview of SIXPPAC ML engine with three example applications: I/Q samples from integrated ADCs are used for (1) time synchronization and (2) SNR estimation, while external CSI data is used for (3) positioning.

Contributions: We present SIXPPAC (cf. Fig. 1), the first (to the best of our knowledge) streaming ML engine for real-time wireless baseband. SIXPPAC implements a six-layer feedforward NN with binary weights, storing all weights, biases, and activations on chip to attain high throughput and low latency. Two integrated 8b SAR ADCs digitize incoming I/Q baseband signals, eliminating digital input bottlenecks. Besides continuous I/Q streaming operation, SIXPPAC supports block processing of multidimensional data. We showcase three applications to demonstrate that SIXPPAC outperforms classical approaches (cf. Sec. III-A) and enables ASIC efficiency for I/Q-sample-based integrated sensing and communication (ISAC).

II. VLSI Architecture

Fig. 2 shows the top-level architecture of SIXPPAC. The core engine is a six-layer perceptron with fully programmable weights and biases. Each fully-connected layer is implemented with dedicated hardware, enabling concurrent processing of consecutive input vectors. SIXPPAC has two operation modes: In the streaming mode, two layers process a window of 128 I/Q samples provided by a pair of integrated ADCs. In the block-processing mode, all six layers process a 256-dimensional vector read from a test memory. SIXPPAC operates bit-serially, processing one bit from all entries of the input vector per clock cycle. Since its inputs are in bit-parallel format, parallel-to-serial converters are used. SIXPPAC writes its results in bit-parallel format to an output test memory.

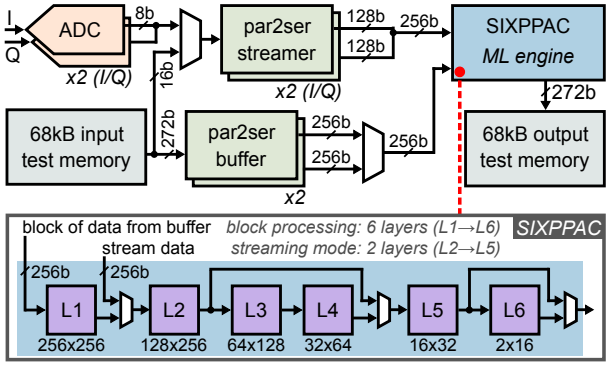


Fig. 2. Top-level architecture of SIXPPAC. Each layer has dedicated hardware and all NN weights, biases, and activations are stored on chip.

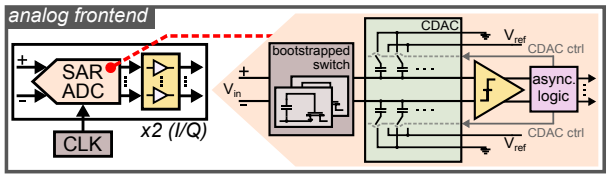


Fig. 3. Details of the SAR ADC analog frontend integrated within SIXPPAC.

A. Analog Frontend

Fig. 3 details the two integrated 8b SAR ADCs, which reside in an independent power and clock domain, and buffer their outputs before delivery to the digital core. The ADCs achieve 7.30 ENOB at 105 MS/s (cf. Sec. III), enabling SIXPPAC to process 8b I/Q samples over a 105 MHz RF bandwidth. Thanks to SIXPPAC's bit-serial operation, higher sampling rates are possible when processing fewer input bits. For example, processing 1b per I/Q sample increases the sampling rate 8× to 840 MS/s, where the ADCs provide a sufficient 5.71 ENOB.

B. ML Engine

Fig. 4 shows the dedicated hardware for each NN layer. Each layer has an $M \times N$ matrix-vector multiplier (MVM) built upon the digital in-memory compute (IMC) "PPAC" architecture from [5]. The MVM multiplies a binary weight matrix $\mathbf{W}$ by a binary input vector $\mathbf{x}$. The 1b MVM outputs are further processed by row peripherals that (i) accumulate the MVM results to support multi-bit inputs, (ii) apply per-row 14b scale factors (to mitigate effects from coarse weight quantization) and 13b biases learned during off-chip training, and (iii) perform linear or ReLU activation. Since the MVM requires bit-serial inputs but produces bit-parallel outputs, the outputs of each layer are serialized before being fed to the next layer.

C. AND-Based Bipolar Inner Product

Prior NN accelerators (e.g., [5]–[8]) implement bipolar (HI/LO logic levels correspond to +1/-1, respectively) multiplication using XNOR gates, as an XNOR gate outputs the same result when its inputs have the same logic level (or sign). Then, the inner product $\mathbf{w}^T\mathbf{x}$ between two bipolar binary vectors $\mathbf{w}$ and $\mathbf{x}$ of dimension N can be computed as follows:

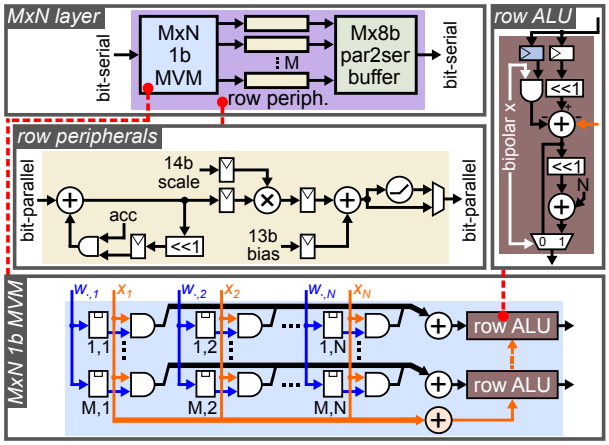


Fig. 4. Hardware architecture for each SIXPPAC layer, consisting of a digital in-memory 1b MVM, row peripherals, and a parallel-to-serial buffer.

$$\mathbf{w}^T\mathbf{x} = 2\sum_{i=1}^{N} w_i \odot x_i - N, \tag{1}$$

where $\odot$ is the XNOR operator, and w_i and x_i take the value of their logic level (HI/LO). The sum operator $\sum$ treats HI as 1 and LO as 0. Therefore, the factor of 2 and bias of $-N$ correct for the mismatch arising from mapping a LO x_i (or w_i) to 0 instead of its intended bipolar value of -1 [5].

With $\overline{x}$ being the logical negation of x, the XNOR operator is

$$w_i \odot x_i = w_i \cdot x_i + \overline{w_i} \cdot \overline{x_i}, \tag{2}$$

where the $\cdot$ and $+$ operators correspond to logical AND and OR, respectively. However, since w_i and x_i take their logical values HI/LO, the AND $\cdot$ operator is equivalent to multiplication with HI/LO≡1/0. Likewise, since $w_i \cdot x_i$ and $\overline{w_i} \cdot \overline{x_i}$ cannot be simultaneously HI, the OR $+$ operator can be treated as addition with HI/LO≡1/0. Thus, we can re-express logical negation as $\overline{x} = 1 - x$ and the XNOR operator (2) as

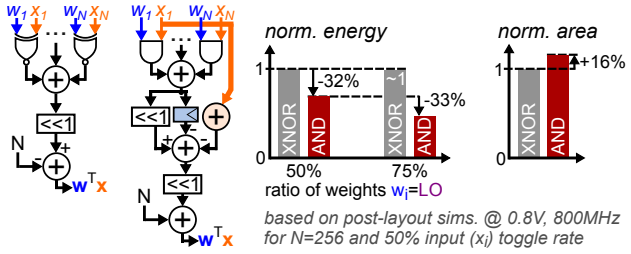
$$w_i \odot x_i = w_i \cdot x_i + (1-w_i)\cdot(1-x_i) = 2w_i \cdot x_i - w_i - x_i + 1. \tag{3}$$

Applying (3) to (1), we arrive at

$$\mathbf{w}^T\mathbf{x} = 4\sum_{i=1}^{N} w_i \cdot x_i - 2\sum_{i=1}^{N} w_i - 2\sum_{i=1}^{N} x_i + N, \tag{4}$$

which, unlike (1), does not use XNOR $w_i \odot x_i$, but AND $w_i \cdot x_i$.

Fig. 5 shows both XNOR- and AND-based inner-product hardware implementations, and compares their energy and area. Our AND-based inner product reduces energy by 32% while increasing area by only 16% compared to the traditional XNOR approach: Lower energy is achieved as a LO weight prevents any toggling activity at its AND's output. The area increase is due to the additional hardware (colored in Figs. 5 and 4) required to compute $\sum_i x_i$ and $\sum_i w_i$ in (4). This additional hardware has a negligible impact on the overall energy, as the input vector $\mathbf{x}$ is the same for a given layer, and as weights $\mathbf{w}$ remain static over several inferences. Hence, the dynamic energy overheads of computing $\sum_i x_i$ and $\sum_i w_i$ are diluted across the IMC array and multiple inferences, respectively.



(a) XNOR (b) AND (c) Energy and area comparison

Fig. 5. Block diagrams of 1b MVMs that perform bipolar (±1) inner products using (a) XNOR vs. (b) AND-based logic. (c) Compared to previous works [5]–[8] using XNOR gates, our novel AND-based approach saves energy at a modest area increase, while supporting unipolar (0,1) inputs as well.

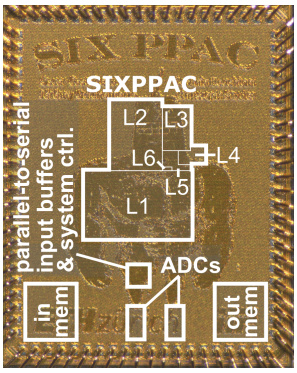
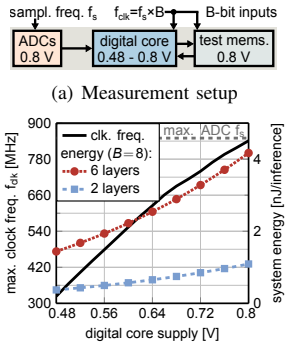


Fig. 6. Micrograph of 2 mm×2.5 mm 12LP+ FinFET SIXPPAC die.



(a) Measurement setup

(b) Measurement results

Fig. 7. Measurement (a) setup and (b) results at 300 K room temperature.

In addition, the energy of AND-based MVMs decreases as more weights are LO: When 75% of the weights are LO, energy reduces by 33% compared to when 50% of the weights are LO, as more AND outputs do not toggle. The worst-case energy occurs at 50% LO weights, since, if a row has more HI than LO weights, all weights can be flipped and the row's scale factor sign inverted without affecting the result. In stark contrast, the energy of the conventional XNOR approach remains practically constant regardless of the number of LO weights. Finally, our AND approach naturally extends to inner products with unipolar (HI/LO ≡ 1/0) inputs, enabling SIXPPAC to support both multi-bit bipolar and two's complement inputs.

III. Implementation Results

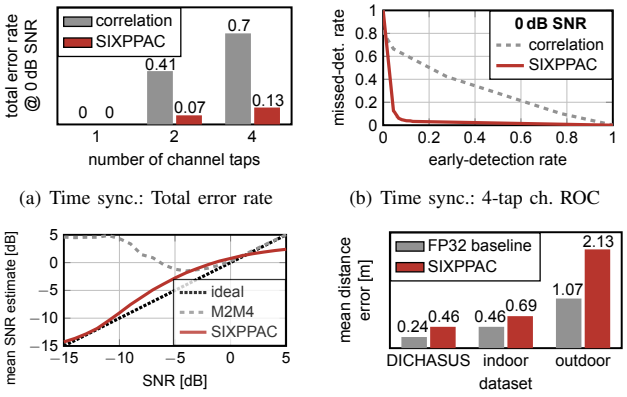
Fig. 6 shows the SIXPPAC die fabricated in 12LP+ FinFET, integrating the ML engine and two SAR ADCs in 0.765 mm^2. Table I and Fig. 7 detail our measurements for the ADCs and digital core, respectively. At 0.8 V, the digital core operates at 840 MHz, processing 105 MS/s for 8b inputs, a rate supported by the ADCs. The digital core achieves its minimum energy at 0.48 V, while the ADCs are always supplied with 0.8 V.

A. Performance for Example Applications

Fig. 8 demonstrates SIXPPAC's versatility through three applications. For time synchronization, SIXPPAC processes blocks of 128 I/Q samples from the ADCs to detect WiFi

TABLE I
Measurements of Implemented SAR ADCs

Technology [nm]	12	Sampling rate f_s [MS/s]	105	600	850
Resolution [bit]	8	ENOB @ $f_{in} \cong f_s/2$ [bit]	7.30	7.26	5.71
Supply [V]	0.8	SNDR @ $f_{in} \cong f_s/2$ [dB]	45.7	45.5	36.1
Active area [mm^2]	0.144	Total power [mW]	0.618	2.29	2.63



(a) Time sync.: Total error rate

(b) Time sync.: 4-tap ch. ROC

(c) Blind SNR estimation

(d) CSI-based positioning

Fig. 8. SIXPPAC performance for three applications: (a-b) time synchronization at 0 dB SNR, (c) blind SNR estimation, and (d) CSI-based positioning (the DICHASUS dataset [9] covers a 12×12 m^2 area, the indoor dataset a 9.2×26 m^2 area, and the outdoor dataset an 81×119 m^2 area).

preambles without channel knowledge. This task is traditionally tackled by computing the correlation between the received samples and the known preamble signal to detect when this correlation exceeds a certain threshold. As mentioned in Sec. I, small deviations in time synchronization can be tolerated in OFDM systems [4]. However, to highlight the capabilities of SIXPPAC, our results in Figs. 8(a) and 8(b) treat any deviation as a synchronization error. Under this strict metric, correlation remains error-free for frequency-flat channels (i.e., channels with 1 tap), but is significantly outperformed by our ML-based SIXPPAC for frequency-selective channels.

For blind SNR estimation, SIXPPAC estimates SNR directly from blocks of 128 I/Q samples captured by the ADCs. Traditionally, this task is performed using the moment-based M2M4 method [3], which fails under some conditions. Fig. 8(c) shows that low SNR is one of such conditions, where SIXPPAC delivers robust performance.

For positioning, SIXPPAC applies block processing to downlink channel-state information (CSI) from distributed multi-antenna access points. This task is typically tackled using ML and floating-point (FP) arithmetic [2]. Fig. 8(d) shows that, although SIXPPAC attains a higher positioning error than a 32b FP baseline, it still achieves sub-2.5 m accuracy with binary weights, which enable a much more efficient hardware implementation. These three examples demonstrate that SIXPPAC's binary-weight NN provides sufficient expressivity, while its programmable weights enable additional ISAC applications.

Fig. 9 shows how precision can be traded for higher processing rate and lower energy thanks to SIXPPAC's bit-serial design: With the DICHASUS CSI positioning dataset [9], using 6b

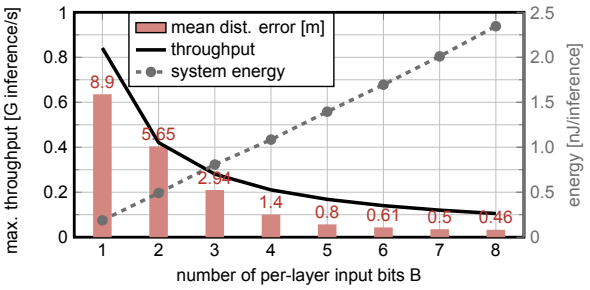


Fig. 9. Impact of per-layer input resolution B on throughput, energy, and positioning performance on the DICHASUS dataset [9] at 0.8 V supply.

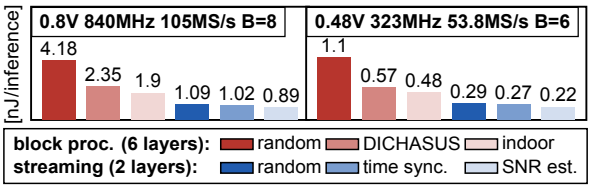


Fig. 10. Energy (including ADC energy) per inference for different applications; "random" corresponds to 50% LO weights and 50% input toggle rate.

instead of 8b inputs increases throughput by 33% and reduces energy by 28%, at the cost of a 15 cm increase in average positioning error. Fig. 10 shows energy (including ADCs) for different tasks: Random weights and inputs consume the most energy, while real-world applications benefit from natural weight and input sparsity. At 0.8 V and 105 MS/s with 8b inputs, mean energy is 1.9 nJ/inference (max. 4.18 nJ/inference); at 0.48 V and 53.8 MS/s with 6b inputs, mean energy drops 3.9×.

B. Comparison with State-of-the-Art

SIXPPAC is, to the best of our knowledge, the first design integrating the input ADCs required to enable real-time baseband ML processing. The closest design to SIXPPAC is [2], an unfabricated ASIP with a CNN accelerator for communications and positioning that delivers 390 positioning/s. In contrast, SIXPPAC achieves 105 M positioning/s with 87 ns latency in a comparable area. Table II compares SIXPPAC to state-of-the-art IMC inference accelerators that support bipolar inner products. SIXPPAC delivers the highest throughput (TOPS) among sub-1 mm^2 designs, with competitive compute density even when including the ADC area. Its AND-based compute enables best-in-class energy efficiency (POPS/W) for exact bipolar inner products. Alternatives [7], [8] improve energy efficiency through approximate or analog compute, which preclude bit-accurate results. In particular, analog compute enables [8] to achieve higher energy efficiency at low compute density; the all-digital SIXPPAC offers a more balanced point in this trade-off. Complete NN inference systems like [10] suffer from low compute density and energy efficiency due to data-movement overheads. In contrast, thanks to AND-based IMC and a fixed NN architecture with fully programmable parameters, SIXPPAC approaches the performance of inference accelerators [6]–[8] while implementing complete I/Q-sample-based ISAC tasks with tight real-time constraints.

TABLE II
COMPARISON WITH STATE-OF-THE-ART IMC INFERENCE ACCELERATORS THAT SUPPORT BIPOLAR INNER PRODUCTS

	This work SIXPPAC		Knag [6]	Lin [7]	Jia [8]	Desoli [10]
Input ADCs	**yes**		no	no	no[a]	no
Computation paradigm	digital, **exact**		digital, **exact**	digital, approx.	analog+ digital	digital, **exact**
Bipolar inner product	**AND-based**		XNOR	XNOR	XNOR	no[b]
IMC array size [kb]	107		128	16	4 608	2 048
Technology [nm]	12		10	28	16	18
Area [mm^2]	0.765		0.39	0.033	25	4.2
Weight prec. [bit]	1	1	1	1	4	4
In/act. prec. [bit]	8	6	1	1	4	4
Core supply [V]	0.8	0.48	0.75	0.9	0.8	1.0
Clock freq. [MHz]	840	323	622	280	200	-
Power [mW]	439	59	607	-	-	-
ADC rate [MS/s]	**105**	53.8	-	-	20[a]	-
Latency [ns]	**87**	189	-	-	-	-
TOPS[c]	183	70.5	163	9.18	189	**229**[d]
TOPS/mm^2	240	92.1	**418**	278	7.55	54.5
POPS/W[c,e]	0.42	1.2	0.27	0.57[f]	**1.9**	0.2[g]

[a]Includes ADCs for internal analog compute; these ADCs cannot be used for external analog inputs. [b]Only supports unipolar/two's complement inner products. [c]Numbers normalized to 1b×1b compute. [d]Throughput for 1b×1b compute extracted from [10, Fig. 16.7.7]. [e]Energy efficiency at 50% input toggle rate, except for [6], [8], which use CIFAR10 and a CNN toggle activity, respectively. [f]Energy efficiency at 50% input toggle rate extracted from [7, Fig. 15.c]. [g]Energy efficiency at 1.0 V supply not provided in [10]; instead, we show energy efficiency at 0.525 V, extracted from [10, Fig. 16.7.5].

IV. CONCLUSIONS

We have presented SIXPPAC, the first mixed-signal ML engine for wireless baseband that directly operates on a stream of I/Q samples. By integrating SAR ADCs with an AND-based NN engine with fully on-chip storage, SIXPPAC unlocks ASIC hardware efficiency for various I/Q-sample-based ISAC tasks. SIXPPAC outperforms traditional approaches in time synchronization and blind SNR estimation, and enables high-rate positioning, demonstrating that real-time ML processing of I/Q baseband samples is not only feasible, but advantageous.

REFERENCES

[1] C. Zhang *et al.*, "Artificial intelligence for 5G and beyond 5G: Implementations, algorithms, and optimizations," *IEEE JETCAS*, Jun. 2020.
[2] M. Attari *et al.*, "Accelerator-assisted floating-point ASIP for communication and positioning in massive MIMO systems," *arXiv preprint arXiv:2502.09785*, Feb. 2025.
[3] D. R. Pauluzzi *et al.*, "A comparison of SNR estimation techniques for the AWGN channel," *IEEE Trans. Commun.*, vol. 48, no. 10, Oct. 2000.
[4] T. M. Schmidl *et al.*, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, Dec. 1997.
[5] O. Castañeda *et al.*, "PPAC: A versatile in-memory accelerator for matrix-vector-product-like operations," in *IEEE ASAP*, Jul. 2019.
[6] P. C. Knag *et al.*, "A 617-TOPS/W all-digital binary neural network accelerator for 10-nm FinFET CMOS," *IEEE JSSC*, vol. 56, Apr. 2021.
[7] C.-T. Lin *et al.*, "DIMCA: an area-efficient digital in-memory computing macro featuring approximate arithmetic hardware in 28nm," *IEEE JSSC*, vol. 59, Mar. 2024.
[8] H. Jia *et al.*, "Scalable and programmable neural network inference accelerator based on in-memory computing," *IEEE JSSC*, Jan. 2022.
[9] F. Euchner *et al.*, "A distributed massive MIMO channel sounder for "big CSI data"-driven machine learning," in *Int. ITG Workshop Smart Antennas (WSA)*, Nov. 2021.
[10] G. Desoli *et al.*, "A 40-310 TOPS/W SRAM-based all-digital up to 4b in-memory computing multi-tiled NN accelerator in FD-SOI 18nm for deep-learning edge applications," in *IEEE ISSCC*, Feb. 2023.