IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 4 August 2022, accepted 22 August 2022, date of publication 1 September 2022, date of current version 14 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203394

RESEARCH ARTICLE

All-Analog Silicon Integration of Image Sensor and Neural Computing Engine for Image Classification

BENJAMIN ZAMBRANO[1], **(Graduate Student Member, IEEE), SEBASTIANO STRANGIO**[2], **(Member, IEEE), TOMMASO RIZZO**[2,3], **ESTEBAN GARZÓN**[1], **(Member, IEEE), MARCO LANUZZA**[1], **(Senior Member, IEEE), AND GIUSEPPE IANNACCONE**[2,3], **(Fellow, IEEE)**

[1]Department of Computer Engineering, Modeling, Electronics and Systems, University of Calabria, 87036 Rende, Italy
[2]Dipartimento di Ingegneria dell'Informazione, University of Pisa, 56122 Pisa, Italy
[3]Quantavis s.r.l., 56126 Pisa, Italy

Corresponding author: Sebastiano Strangio (sebastiano.strangio@unipi.it)

This work was supported in part by the European Commission through the Quantum Engineering for Machine Learning (QUEFORMAL) Horizon 2020 (H2020) Project under Contract 829035, and in part by the European Commission and the Italian Ministry of Industry and Economic Development [Italian Ministry of Economic Development (MISE)] through the Electronics Components and Systems for European Leadership (ECSEL) Challenging environments tolerant Smart systems for IoT and AI (CHARM) Project under Contract 876362.

ABSTRACT We have designed a fully-integrated analog CMOS cognitive image sensor based on a two-layer artificial neural network and targeted to low-resolution image classification. We have used a single poly 180 nm CMOS process technology, which includes process modules for realizing the building blocks of the CMOS image sensor. Our design includes all the analog sub-circuits required to perform the cognitive sensing task, from image sensing to output classification decision. The weights of the network are stored in single-poly floating-gate memory cells, using a single transistor per analog weight. This enables the classifier to be intrinsically reconfigurable, and to be trained for various classification problems, based on low-resolution images. As a case study, the classifier capability is tested using a low-resolution version of the MNIST dataset of handwritten digits. The circuit exhibits a classification accuracy of 87.8%, that is comparable to an equivalent software implementation operating in the digital domain with floating point data precision, with an average energy consumption of 6 nJ per inference, a latency of 22.5 μs and a throughput of up to 133.3 thousand inferences per second.

INDEX TERMS Analog neural network, cognitive image sensor, neuromorphic engineering.

I. INTRODUCTION

End devices or edge devices in the Internet of Things (IoT) paradigm, with embedded intelligent visual sensor systems, are key components of recent visions of cyberphysical systems [1], where latency, scalability, and privacy represent important challenges. In a conventional machine vision system, information is captured by image sensors to be then converted into a digital format, before being stored in a local memory or being transmitted to an external computing unit for required processing tasks. With the main goal of reducing the impact of energy- and time-inefficient operations, such as analog-to-digital conversion and data transmission, the concept of a cognitive image sensor, with embedded classification capabilities becomes an attractive solution for future applications [2], [3] such as wearable and mobile healthcare electronics [4], or battery-powered systems, such as autonomous robots and drones [5].

Intelligent vision sensors provide cognitive capabilities to image sensors through the implementation of artificial neural networks (ANNs), which represent powerful modeling methods to perform human-like tasks, such as object classification and detection [1]. As a recent milestone for the CMOS image sensor (CIS) market, the world's first commercial image

The associate editor coordinating the review of this manuscript and approving it for publication was Alireza Sadeghian.

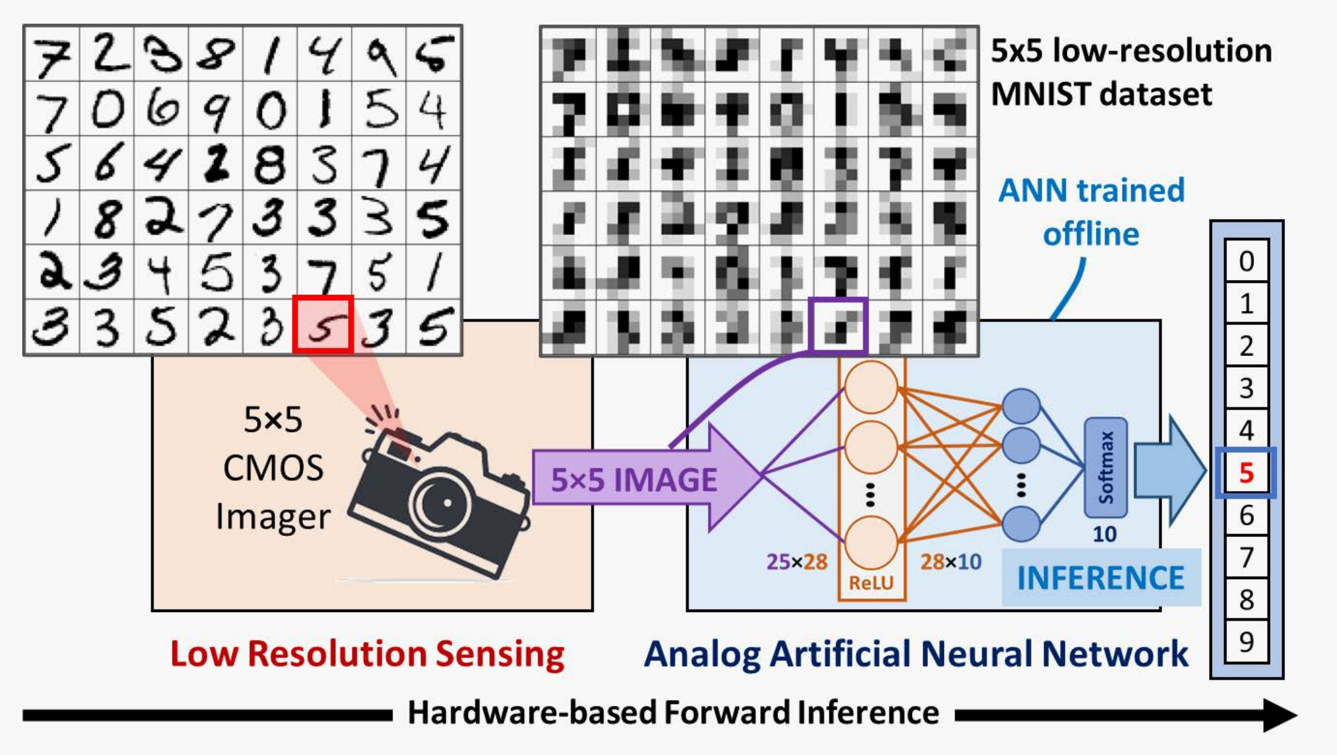


FIGURE 1. Concept of the proposed low resolution all-analog CMOS image sensor classifier, composed of a 5 × 5 image sensor and a 2-layer analog neural network.

sensor with artificial intelligence (AI) processing capability has been launched in May 2020 [6]: the intelligent CIS has been designed by equipping a conventional image sensor with a digital signal processor (DSP) dedicated to AI processing tasks and the memory for the AI model.

ANNs are composed of layers of artificial neurons. The basic computation in the ANNs is the multiply-accumulate (MAC) operation [7], [8], [9], that is the elementary operation of a vector-matrix multiplication, where an input data vector is multiplied by a matrix of fixed weights. As the size of the ANN increases, the increased number of MAC operations, as well as storage requirements and weight access operations, result in a huge energy consumption in conventional digital systems [10], [11], [12]. In order to reduce power consumption per inference, thus enabling battery-powered systems to be equipped with ANNs, a lot of research effort is today being devoted to the design of analog ANN integrated circuits [2], [13], [14], [15], [16], [17], [18], which exploit basic properties of CMOS devices and circuits to allow a very high degree of parallelism in MAC operations along with in-memory computation.

Reduced precision of both input data and of processing tasks, typical of the analog domain, has been demonstrated to be well tolerated by neural networks [14]. For instance, a fully integrated, on-chip ANN classifier architecture based on analog circuits for low-resolution image classification has been presented in [17]. There are many applications of low-resolution image classification [19], for instance when the object to be detected (i.e., the region-of-interest) is confined in just a portion of the image, or when the image has a deliberately low resolution for privacy reasons (e.g., remote health monitoring of patients).

Inspired by the above mentioned research papers, and in particular by [16] and [17], we propose the design of an analog CMOS image sensor classifier based on a two-layer ANN operating in a low-resolution context, as depicted in Fig. 1. As a main difference with respect to previous works, which typically show only partial on-chip implementations, this is the first demonstration of a fully analog design, which includes all the building blocks required to perform the whole processing task, from the image sensing to the image classification.

Our system is designed using a single-poly 180 nm commercial CMOS process, with an additional process option including specific modules for the implementation of photo diodes and supplementary building blocks of modern CMOS image sensors. Matrices of single-poly, single-transistor floating-gate memory cells store the weights of the ANN, and are part of time-domain based vector-matrix multipliers (VMMs), both proposed in [15]. All the other building blocks, such as the pixel array, sample and hold (S&H) arrays, voltage-to-time converters and activation functions [20] are carefully designed in the analog domain. In addition, as opposed to [17], where the weights are encapsulated in the design, here the weights are programmable, so that the proposed classifier is intrinsically re-configurable, and can be trained to operate on a range of classification tasks based on low resolution images.

As a case study, the classifier inference capability is validated using a low-resolution dataset derived from the MNIST database of handwritten digits [21]. The resulting inference accuracy is 87.8% at room temperature (27 °C), which is comparable to a software implementation of the same ANN architecture operating in the digital domain with floating point data precision. The accuracy is maintained over the 80% in a broad temperature range, from −10 °C to 70 °C. Our design consumes only 6 nJ per inference (where roughly half of the energy is consumed by the pixel sensing matrix), while assuring a throughput of 133k images per second with a latency of 22.5 μs. It also exhibits a very small footprint of only 4000 μm^2 for a pixel pitch of 10 μm.

The rest of the paper is organized as follows. The design methodology is presented in Section II, with discussion of the entire design flow based on the ANN architectural synthesis and training performed in a software environment (MATLAB), along with the architectural integration of the network in the corresponding CMOS implementation. Design and operation of each building block implementing the full CMOS engine are discussed in Section III. Results and discussions are reported in Section IV, also presenting a comparison with similar networks proposed in the literature. Finally, conclusions are drawn in section V.

II. ANALOG CMOS NEURAL NETWORK CLASSIFIER: OFF-CHIP TRAINING AND HARDWARE ARCHITECTURE

Fig. 1 sketches the basic block diagram of the proposed image sensor classifier. The analog CMOS chip is based on two main blocks, a low resolution image sensor and a two-layer ANN trained to perform inferences on the acquired images. In order to evaluate the functionality of our design and to benchmark it against other similar proposals, we have trained a software implementation of the ANN with a down-sampled set of images extracted from the MNIST database. MNIST handwritten digit images from both training and test databases (with original 28 × 28 resolution) have been reduced to 5 × 5 by using a Bilinear interpolation [22], as the input of the ANN is represented by a 5 × 5 pixel image, which is assumed to be captured by the low-resolution image sensor. Some examples of the images are shown in Fig. 1. In the following sub-sections we provide the details of the software off-chip training of the neural network (II-A), as well as the block diagram of the full hardware implementation of the chip (II-B). Circuit details of the single blocks will be provided in Section III.

A. NEURAL NETWORK OFF-CHIP TRAINING

The proposed ANN is composed by a 28-node hidden layer, exploiting Rectified Linear Unit (ReLU) as activation function, and a 10-node output layer followed by a Softmax activation function to yield the final inference result (similar architecture as in [17], but with different activation function at the hidden layer).

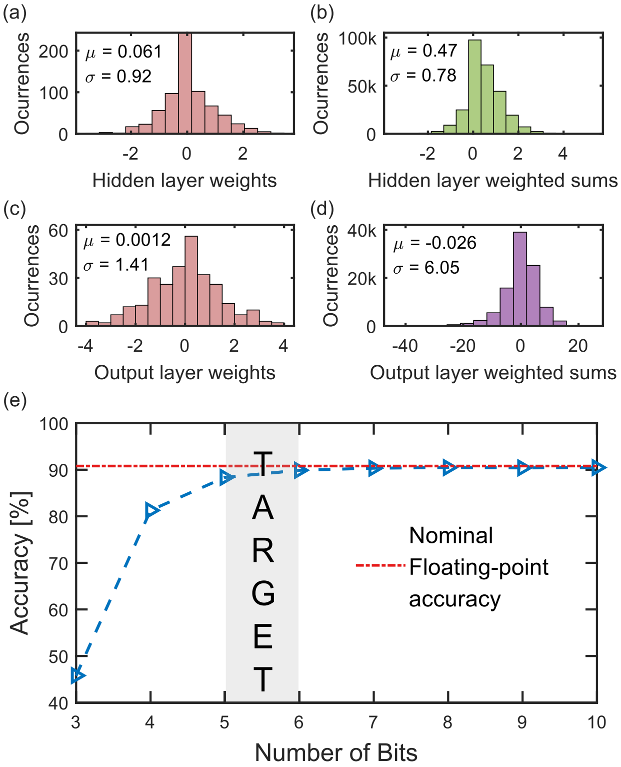


FIGURE 2. Distributions of weights and weighted sums corresponding to hidden (a-b) and output (c-d) layers used for proper selection of full-scale values; (e) impact of resolution truncation on network accuracy.

The ANN was trained in MATLAB applying the mini batch method with momentum for weight updates. After 20 epochs, the network reached an accuracy of 90.6% for the whole set of 10k test images. After the training stage, the weight matrices, as well as the values of the input and output variables of each layer, have been obtained for all testing images. All weights and signals are dimensionless variables with floating point precision of 16 decimal digits. However, for each variable, the data precision can be deliberately reduced to a given number of bits to reach the minimum precision actually required to obtain the target classifier inference accuracy. On the other hand, when considering the corresponding analog network to be designed, all signals are analog quantities (i.e, charges, currents or voltages). Thus, distributions of each quantity must be accurately analyzed at a software level to perform an adequate normalization, with definition of full-scale values. In addition, a target accuracy (in terms of equivalent number of bits) has to be decided for the weights and for the outputs of the analog operations representing critical quantities. The distributions for weights and weighted sums of the two ANN layers are shown in Fig. 2(a-d), evaluated on the whole set of tested images. They allow to predict the magnitude that each variable can assume during the inference operation. In fact, while in a software environment we can rely on perfectly

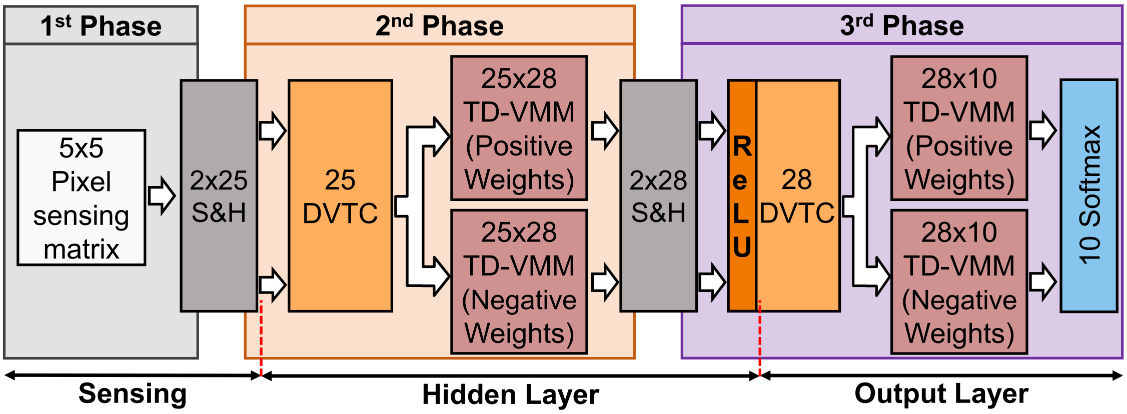


FIGURE 3. Hardware neural network block diagram.

linear operations as the only limitation is the digital data precision, in an analog hardware environment we need to cope with non-linear and noisy functions. However, hardware blocks can be optimized to feature an adequate signal-to-noise and distortion ratio (SINAD) in a limited operating range, therefore it is crucial to have the expected values of all quantities fall in their corresponding range. When considering the conversion factors from software variables to electrical signals, the full-scale ranges have been chosen to comprise about 99% of the data of the distributions: [-4,4], [-3,3], [-4,4] and [-20,20] were selected for the hidden layer weights, hidden layer weighted sums, output layer weights and output layer weighted sums, respectively.

Starting from the selected full-scale range, we have repeatedly tested the ANN via software by truncating the precision of each variable (inputs, weights and summations), by dividing the full range by 2^N, where N is the considered number of bits. The inference accuracy has been extracted for each instance of the reduced-precision ANN. In Fig. 2(e) the inference accuracy as a function of the number of bits N is shown: already with 4 bits, a precision higher than 80% is achieved, while with 5 and 6 bits we have comparable inference accuracy to full floating-point precision.

B. HARDWARE ARCHITECTURE OF THE PROPOSED ANALOG CMOS NEURAL NETWORK CLASSIFIER

The complete hardware implementation of the proposed CMOS image sensor classifier is shown in Fig. 3. The architecture has been split in three sections, one related to the sensing and the other two implementing the two layers of the ANN. Two $S\&H$ arrays are interposed between two adjacent sections in order to store intermediate data, to obtain a systolic architecture that makes it possible to execute all the three phases in parallel.

The sensing stage is represented by a 5 × 5 pixel image sensors. Each pixel provides a differential voltage signal (ΔV_{pix}) proportional to the incident light. These differential voltage signals are converted into voltage pulses, with their pulse width (PW) proportional to the captured light, by a Differential Voltage to Time Converter (DVTC) array. Two identical 25 × 28 VMMs, denoted as VMM^+ and VMM^-, receive the PW signals as inputs and implement a differential VMM stage to deal with signed weights (see Fig. 2(a,c)). The outputs of VMM^+ and VMM^- are sampled in a 2 × 28 $S\&H$ array, and each ΔV_{OUT} is converted to a PW signal by a second DTVC block, to be transferred to the output layer. The array of 28 DVTCs also plays the role of the ReLU activation function. Then, the output layer is realized with two identical 28 × 10 VMMs (again VMM^+ and VMM^-), and the ΔV_{OUT} results are translated into probabilities by a softmax block, providing the final inference.

The whole system, sketched in Fig. 3, with each block independently optimized as discussed in the following section, has been integrated in a single circuit schematic in Virtuoso Schematic Editor (within Cadence IC 6.1.8 environment) and the full operation flow has been electrically simulated in ADE XL, from image capture to final inference.

Thanks to the two S&H arrays, which break the chain at two intermediate points, the proposed network has a systolic architecture that can realize a pipelined implementation. For instance, when data related to the first image is in the second phase and is being processed in the first hidden layer of the network, the sensing block can start the acquisition of a new image in parallel. With this assumption, the network can produce a new inference every 7.5 μs, for a throughput of 133.3 thousand inferences per second, after the initial latency of 22.5 μs, and by consuming on average a total of 6 nJ per inference (with 3 nJ consumed by the pixel sensing matrix, and 3 nJ by the ANN). The obtained inference accuracy is 87.8%, which is comparable with the one we have obtained by means of the idealized software network, with floating point data precision (90.6%).

III. DESIGN OF THE BUILDING BLOCKS OF THE ANALOG CMOS CHIP

In this section we discuss the design and operation of each building block implementing the classifier. The analog design exploits the time-domain VMMs (TD-VMMs) with two-

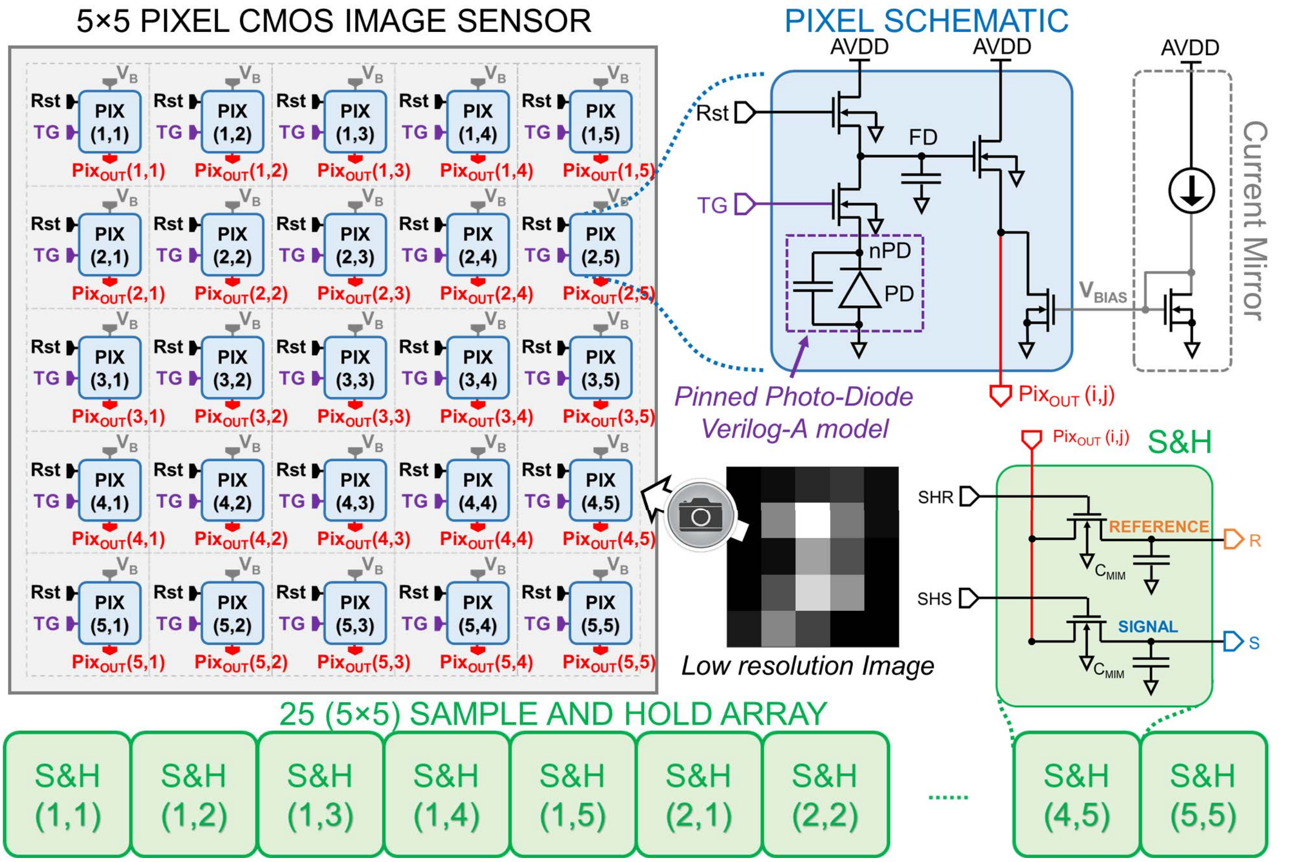

5×5 PIXEL CMOS IMAGE SENSOR
Rst
TG
V_B
PIX (1,1) PIX (1,2) PIX (1,3) PIX (1,4) PIX (1,5)
Pix_OUT(1,1) Pix_OUT(1,2) Pix_OUT(1,3) Pix_OUT(1,4) Pix_OUT(1,5)
PIX (2,1) PIX (2,2) PIX (2,3) PIX (2,4) PIX (2,5)
Pix_OUT(2,1) Pix_OUT(2,2) Pix_OUT(2,3) Pix_OUT(2,4) Pix_OUT(2,5)
PIX (3,1) PIX (3,2) PIX (3,3) PIX (3,4) PIX (3,5)
Pix_OUT(3,1) Pix_OUT(3,2) Pix_OUT(3,3) Pix_OUT(3,4) Pix_OUT(3,5)
PIX (4,1) PIX (4,2) PIX (4,3) PIX (4,4) PIX (4,5)
Pix_OUT(4,1) Pix_OUT(4,2) Pix_OUT(4,3) Pix_OUT(4,4) Pix_OUT(4,5)
PIX (5,1) PIX (5,2) PIX (5,3) PIX (5,4) PIX (5,5)
Pix_OUT(5,1) Pix_OUT(5,2) Pix_OUT(5,3) Pix_OUT(5,4) Pix_OUT(5,5)
PIXEL SCHEMATIC
AVDD
AVDD
AVDD
Rst
FD
TG
nPD
PD
V_BIAS
Current Mirror
Pix_OUT (i,j)
Pinned Photo-Diode
Verilog-A model
Low resolution Image
Pix_OUT (i,j)
S&H
SHR
REFERENCE
R
C_MIM
SHS
SIGNAL
S
C_MIM
25 (5×5) SAMPLE AND HOLD ARRAY
S&H (1,1)
S&H (1,2)
S&H (1,3)
S&H (1,4)
S&H (1,5)
S&H (2,1)
S&H (2,2)
......
S&H (4,5)
S&H (5,5)

FIGURE 4. 5 × 5 pixel CMOS image sensor architecture, with details of the 4T pixel and S&H schematics.

terminal non-volatile memory [15] and the softmax activation function [20]. Purposely designed differential voltage to pulse-width converters are used to provide proper input data to the TD-VMMs. The design of the analog circuit implementation of each individual block has been optimized to get a SINAD corresponding to an effective number of bits, $ENOB = (SINAD - 1.76)/6.02$, between 5 and 6 for the analog signals and the analog operations, as a trade-off between design complexity and power consumption on one side, and achievable inference accuracy on the other side (see Fig. 2(e)). The conversion factor from each software variable to the corresponding physical quantity (i.e, voltage, electric charge and current) will be presented in the corresponding block subsection.

A. 4T PIXEL SENSING MATRIX

A standard pixel architecture is used as a building block for the CMOS image sensor with a resolution of 5×5 pixels. The pixel schematic and its sensing scheme are shown in Fig. 4. The pixel is based on a light-sensitive pinned photo-diode (PD), complemented with four transistors. The pixel read-out is performed with a correlated double sampling approach, in which the light captured by each pixel is proportional to ΔV_{pix}, defined as the difference between a reference voltage (i.e., R, the pixel output voltage after reset) and the signal voltage (i.e., S, the pixel output voltage at the end of integration), in order to cancel the thermal noise associated to the voltage on the PD capacitance (i.e., the so-called KTC noise). However, as opposed to the standard 4T pixel architecture [23], only a transfer gate (the one driven by the TG signal in Fig. 4), reset (driven by the Rst signal) and a source-follower amplifier are used (driven by the floating diffusion FD node). The row-select transistor is not needed here, since all the 25 pixels are directly connected to the corresponding S&H blocks, 25 in total, each realized with two selectors and two MIM capacitors, to store the ΔV_{pix} in terms of R and S values, with $R > S$. This translates into the fact that all the 25 pixels can be exposed at the same time, in a global shutter mode, which is indeed feasible due to the extremely low resolution. In our case, the fourth transistor of each pixel is a biasing nMOS, operated at a constant gate voltage, used to set the working point of the source-follower amplifier. This gate voltage is provided by a diode-connected transistor, in common to the whole pixel array, realizing a current mirror.

In order to perform circuit simulations and extract the SINAD of the sensing chain, we have relied on an idealized Verilog-A model for the pinned PD, calibrated against photo-diode test-structures realized in a 0.18 μm CMOS IS technology [24]. It is important to highlight that the actual pixel response is heavily dependent on the real layout implementation, but semiconductor companies normally do not provide parametric cells for the PD within the standard process design kit. In addition, in order to simulate the light-to-charge conversion (i.e., the photoelectric effect) taking place in the n-region of each PD (i.e., the nPD), we have used a 5 × 5 matrix of idealized current pulse sources injecting a certain amount of charge in each pixel, by assuming a quantum efficiency of 0.722. The simulated pixel has a conversion gain of 70 μV/electron, with a full well of 9000 electrons (at $\Delta V_{pix} = 630\ mV$). This charge will be collected in an integration time of 6.25 μs under a 500 W/m^2 maximum light intensity exposure.

The 25 S&H are implemented with minimum size nMOS transistors as selectors, and 2.4 × 2.4 μm^2 MIM capacitors, for a total capacitance of 6.48 fF. This value has been calculated to guarantee a KTC noise on the S&H capacitors lower than 0.1 % of the pixel output voltage dynamic range. With these values as starting points, the MIM capacitors have been enlarged in order to guarantee a 6 bit ENOB on the sensing chain.

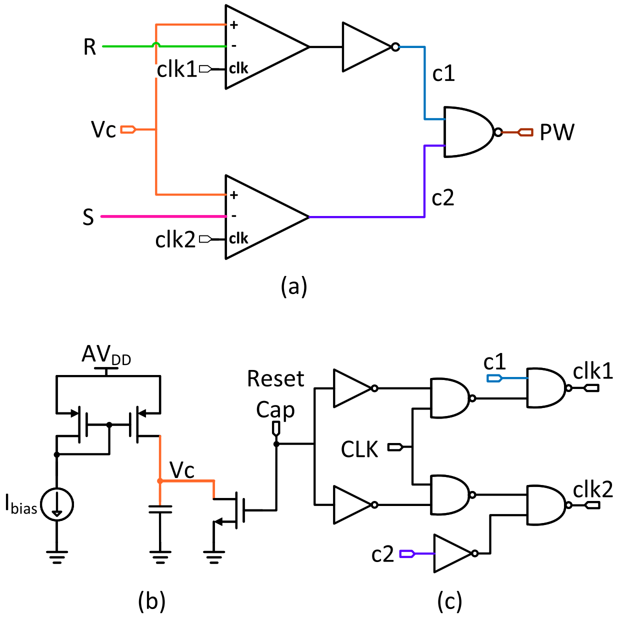


FIGURE 5. (a) Voltage to pulse width converter schematic, (b) ramp generator, (c) control circuit.

B. DIFFERENTIAL VOLTAGE TO PULSE WIDTH CONVERTER

The light intensity collected by each pixel during its integration time, stored as ΔV_{pix}, needs to be converted to a time PW to be used as input for the TD-VMM. Similarly, also the input data to be provided to the second VMM is encoded as a voltage difference, as it will be clear in the following sections. Thus, for both stages, a DVTC array is required for the conversion. In order to perform such task, we have relied on the circuits reported in Figure 5. The goal is to create a time pulse whose duration is proportional to ΔV_{pix}. The basic approach relies on the comparison of a voltage signal V_c which, starting from 0V, is increased over time with a constant slope; as reported in Fig. 5(a), two dynamic comparators (proposed in [25]) and two logic gates are used to recognize the time frame where Vc is comprised between S and R. Note that the signal $c2$ is high when $V_c > S$, while the signal $c1$ is high when $V_c < R$, thus the PW is active (n.b. active low) when both $c1$ and $c2$ are high, that is when $S < Vc < R$. The voltage ramp signal V_c is generated by charging a MIM capacitor with constant current, as shown in Fig. 5(b).

In the same figure, a low-complexity logic network is reported, exploited to activate and deactivate the dynamic comparators, while minimizing their power consumption through clock gating.

The operation of the DVTC can be easily observed by the time diagram in Fig. 6, which highlights how the ΔV_{pix} signal is translated into the time duration of the PW signal.

The same building block is employed to adapt the result of the first VMM layer, which is provided in the form of a voltage difference, to a time signal to be provided as input for the VMM of the second layer. In this regard, it is important to highlight that it also includes the ReLU activation function. In fact, if $\Delta V_{pix} < 0$, the converter does not provide any output pulse, since the condition $AND(c1, c2)$ is never satisfied during the V_c ramp up.

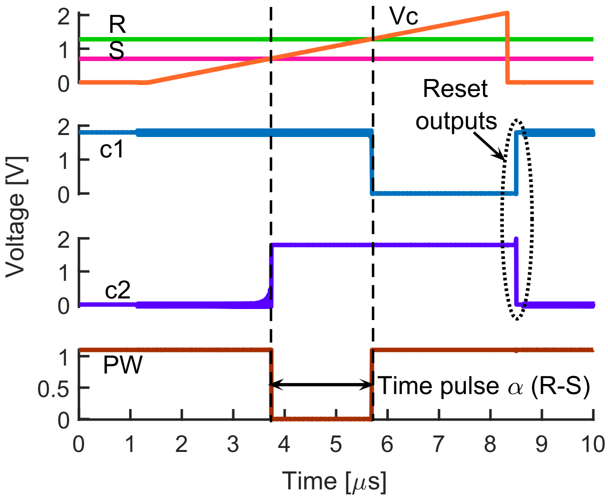


FIGURE 6. Voltage to pulse width converter timing diagram example.

There are several design trade-offs, which can limit the linearity of the DVTC, both at V_c ramp generation as well as at the comparison stages. It would be desirable to have a very fast voltage ramp in order to minimize the inference latency and to maximize the throughput of the classifier. However, this would require high charging current and/or small (and consequently noisy) capacitance. Current overshoots could degrade the linearity of the V_c ramp, which can be achieved only with actually constant charging current and capacitance. Furthermore, the dynamic comparators perform

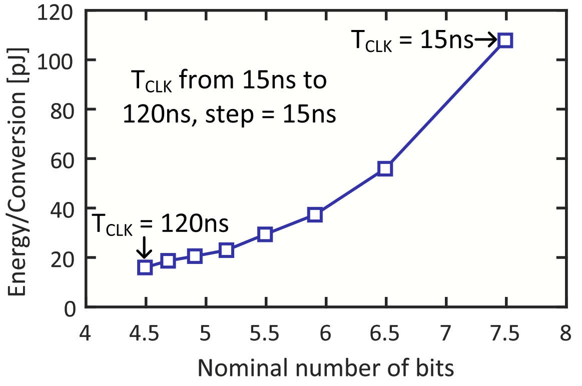


FIGURE 7. DVTC energy/conversion as a function of the nominal number of bits.

one comparison at each clock period T_{clk}, and the ratio of the V_c ramp duration to T_{clk} set a theoretical limit to the achievable resolution.

We have first optimized the V_c ramp generation circuit as follows: a current of 30 nA charging a capacitor of 100 fF results in a slope of 300 mV/ μs. This slope corresponds to a conversion factor of 3.33 μs/ V, thus for a maximum ΔV_{pix} of 0.63 V we get a maximum pulse duration T_{MAX} of 2.1 μs: such value falls within the range in which the TD-VMM that we have proposed in [15] can operate with 6-bit precision. In addition, an optimization of the comparator stage was also needed in order to perform a conversion with at least 5-bit precision, which is required for our test case. The desired clock speed and negligible input offsets can only be obtained at the cost of high power consumption of the comparators. Due to its dynamic nature, the comparator implementing the DVTC block performs a comparison between the V_c input ramp and the R (and S) signal every T_{clk}. Thus, resolution can be improved by decreasing the T_{clk} duration, as the T_{clk}/T_{ramp} ratio corresponds to the number of conversion levels. On the other hand, the increased clock frequency would adversely impact the dynamic power consumption due to increased circuit activity. This energy/precision tradeoff can be observed in Fig. 7, where the energy consumed per each conversion is plotted against the nominal number of bits calculated as $log_2(T_{\text{MAX}}/T_{\text{clk}})$. For each ΔV-to-PW conversion, the converter circuitry consumes an average energy of about 40 pJ at a clock period of 45 ns for 6 nominal bits.

C. TIME DOMAIN VECTOR-MATRIX MULTIPLIER

The VMM is the most recurring building block of a neural network. It enables the multiplication of a vector of features (i.e., input signals of a layer) with a matrix of learning weights, which are normally stored in a non volatile memory since they are fixed quantities during the inference. In a digital environment, this task is performed by recursively multiplying each element of the input vector by the corresponding weights in the row, and then summing all the results in the same column, according to the block diagram in Fig. 8(a). However, the recurring nature of these arithmetic operations can be exploited by taking advantage of a parallel processing in-memory approach, possibly by using analog nonvolatile memory cells and basic device and circuit properties to implement such operations in an energy and time efficient way.

In our classifier, we exploit the TD-VMM, proposed by some of us in [15]. The architecture of a generic $M \times N$ VMM is shown in Fig. 8(a): the inputs x_i ($i = 1, \ldots, M$) are connected to the rows of a $M \times N$ matrix of memory cells, which stores the programmable weights $w_{i,j}$ where the vector-matrix multiplication operation takes place. Then, the results of such operations are summed together by connecting each cell to its corresponding column y_j ($j = 1, \ldots, N$), giving N outputs as result.

In the time-domain approach, each input is encoded in the duration $t_i \in [0, \ldots, T]$ of an active-low voltage pulse ($V_{OFF} = 1.1\ V$, $V_{ON} = 0\,\text{V}$), applied to the i-th row, i.e., wordline (WL), of the array. The $M \times N$ matrix is realized with 1T-FG cells [15], which are minimum-area, single-poly and single-transistor analog memory cells that can be programmed by means of charge injection in their floating-gate (FG): Fig. 8(b) presents the measured I-V characteristics of the cell, showing the possibility of spanning more than 3 decades of current levels for a read voltage on the drain of 1.1 V. The programmable weights $w_{i,j}$ are therefore encoded in the currents $I_{i,j}$ of 1T-FG cells. By connecting the drains of all cells to the same column node, i.e., bitline (BL), at a reference voltage V_{OFF}, corresponding with the off-state level of the input pulse, each cell is activated by their corresponding WL input pulse, applied to their source terminal, for its time duration. With the assumed protocol, the current $I_{i,j}$ is positive when it flows from the BL towards the WL (when the WL is activated during the pulse on-state) and cannot be negative. Therefore, each cell (i, j) injects a net charge $Q_{i,j} = -I_{i,j} \cdot t_i$ into the BL.

Finally, the total negative charge injected in the BL j is converted into a positive voltage by means of an inverting charge amplifier, namely an integrator realized with an amplifier followed by a feedback capacitor, which is reset at V_{OFF} at the beginning of each operation. The output result of the operation is therefore the voltage:

$$V_{out,j} - V_{OFF} = -\frac{1}{C}\sum_{i}^{M} Q_{i,j} = \frac{1}{C}\sum_{i}^{M} I_{i,j} \cdot t_i. \qquad (1)$$

One should note that the proposed VMM block cannot accept negative inputs or weights. Although in our ANN the inputs of the VMM stages are always positive, weights can be also negative as illustrated in Fig. 2(a) and (c), thus leading to possible negative outputs as reported in Fig. 2(b) and (d). In order to enable signed weights, we have implemented each VMM of the ANN with a differential architecture realized with a VMM$^+$ and VMM$^-$, one for positive and one for negative weights, respectively. In particular, for $w_{i,j} > 0$ we have:

$$w^+_{i,j} = w_{i,j} \text{ and } w^-_{i,j} = 0, \qquad (2)$$

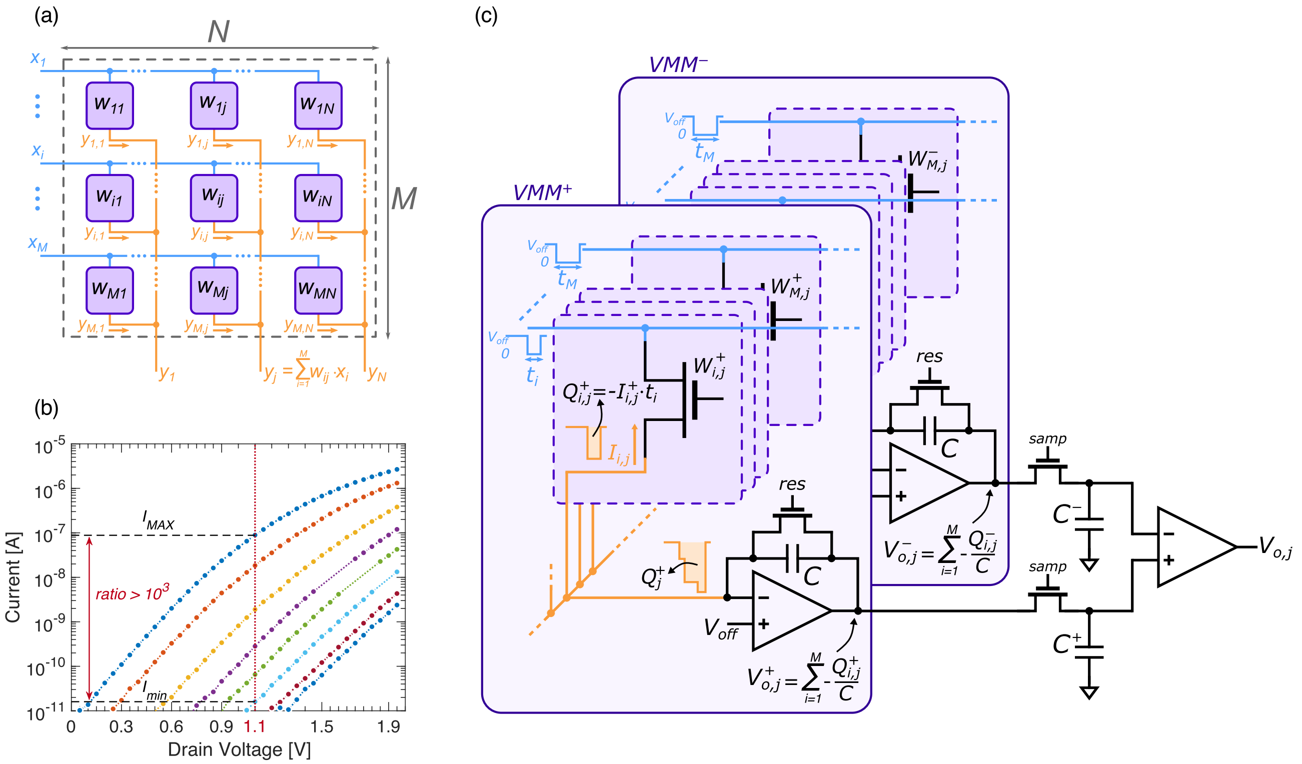


FIGURE 8. Time-domain VMM: (a) system level block diagram of a $M \times N$ VMM, with detail of the interconnection between the inputs (blue), weight matrix (purple), and outputs (yellow); (b) 1T-FG NVM cell I-V characteristics; (c) implementation of the TD-VMM differential architecture for the proposed ANN.

where $w^+_{i,j}$ and $w^-_{i,j}$ are the weights stored in VMM$^+$ and VMM$^-$, respectively, and $w_{i,j}$ is the weight in the resulting VMM. On the other side, for $w_{i,j} < 0$ we have:

$$w^+_{i,j} = 0 \text{ and } w^-_{i,j} = -w_{i,j}. \tag{3}$$

The differential architecture leads to the same result of a VMM with signed weights, if the result is taken as the voltage difference of VMM$^+$ and VMM$^-$ outputs, according to:

$$\Delta V_{out,j} = V^+_{out,j} - V^-_{out,j} = \frac{1}{C}\left(\sum_i^M I^+_{i,j}\cdot t_i - \sum_i^M I^-_{i,j}\cdot t_i\right). \tag{4}$$

In order to make the conversion from the variables of the software VMMs to the corresponding hardware implementations, we have extracted in Fig. 9 the independent histograms for the VMM$^+$ and VMM$^-$ output results of both layers.

By identifying the ranges [0,5] and [0,25] for both VMM$^+$ and VMM$^-$ at hidden and output layer, respectively, and by assuming a voltage range $V_{out,j} - V_{OFF}$ of up to 700 mV for each VMM circuit implementation, we get a conversion factor of 140 mV and of 28 mV, respectively. Starting from these values and considering that the pixel inputs have been encoded into a PW according to a conversion factor of 3 μs (considering the full input chain, from image pixel to the DVTC output), we have selected a capacitance of 600 fF, resulting in a conversion factor for the hidden layer weights of

$$\frac{I_{i,j}}{w_{i,j}} = 28\,\text{nA}. \tag{5}$$

When considering the output layer, the same DVTC and the same charge amplifier have been used. Thus, starting from the output of the first layer obtained with a conversion factor of 160 mV, after the conversion in a time pulse it is encoded with a conversion factor of 5.333 μs. The computed conversion

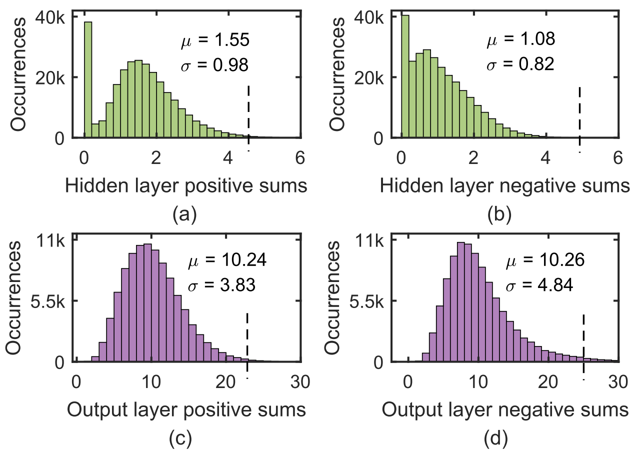


FIGURE 9. Hidden layer (a) positive, (b) negative sum results and Output layer (c) positive, (d) negative sum results.

factor for the output layer weights is

$$\frac{I_{i,j}}{w_{i,j}} = 3.15\,\mathrm{nA}. \tag{6}$$

D. SOFTMAX ACTIVATION FUNCTION

For a given input of the ANN, the results of the output VMM layer include the information about the inference, but can generally be negative and the algebraic sum over all the outputs is not necessarily 1. For this reason, the output stage is normally followed by a softmax function, which is exploited in order to put the output in the form of a probability. There is one output signal for each class and, for a given input, the k-th output of the softmax represents the probability that this input is part of the corresponding class. The softmax is basically an activation function which normalizes the outputs of a network to a probability distribution over the predicted output classes, according to:

$$Softmax_k = \frac{e^{x_k}}{\sum_{j=1}^{N=10} e^{x_j}}. \tag{7}$$

To realize the hardware implementation of softmax function, we have exploited the circuit scheme proposed in [20] and depicted in Fig. 10(a). However, since the softmax inputs (i.e., the results of the output layer VMM) are available as ΔV_{out} signals stored in differential S&Hs, we have only used the inherent softmax circuit without the need to perform a preliminary current to differential voltage conversion as suggested in [20]. By selecting an I_{SCALE} current of 100 nA, and by considering the conversion factor of the output VMM (28 mV), the probability of being part of the k-th class is $I_{OUT,k}/I_{SCALE}$, which can be comprised between 0 and 1. As an example, Fig. 10(b) shows the probability for the generic class k as a function of the $\Delta V_{OUT,k} = V_{pk} - V_{nk}$ of the output layer VMM, assuming that the outputs related to the other 9 classes are all fixed at $\Delta V_{OUT,i} = 0$. When the swept $V_{pk} - V_{nk}$ is 0, all the 10 outputs have the same probability of 1/10. However, the probability quickly falls to 0 for a $\Delta V_{OUT,k} < -100\,\mathrm{mV}$, or to a value very close to 1 for $\Delta V_{OUT,k} > 200\,\mathrm{mV}$. In the same figure, the softmax function from (7) was plotted, showing a good agreement between the simulated output characteristics and the actual function.

IV. RESULTS AND DISCUSSION

The whole system, sketched in Fig. 3, with each block independently optimized as discussed in the previous subsections, has been integrated in a single circuit schematic and the full operation flow has been simulated within the Cadence Virtuoso IC 6.1.8 design environment, from image capture to final inference.

Fig. 11 shows the timing diagram of the designed architecture. The full operation is divided in three phases, in agreement with the block diagram in Fig. 3) sensing, i.e., the image capture; 2) data processing in the hidden layer; 3) data processing in the output layer. Based on the timing diagram, we have optimized each block in order to perform its task in a time frame of 7.5 μs. This allows pipelining, where all the three stages work in parallel, while processing different images.

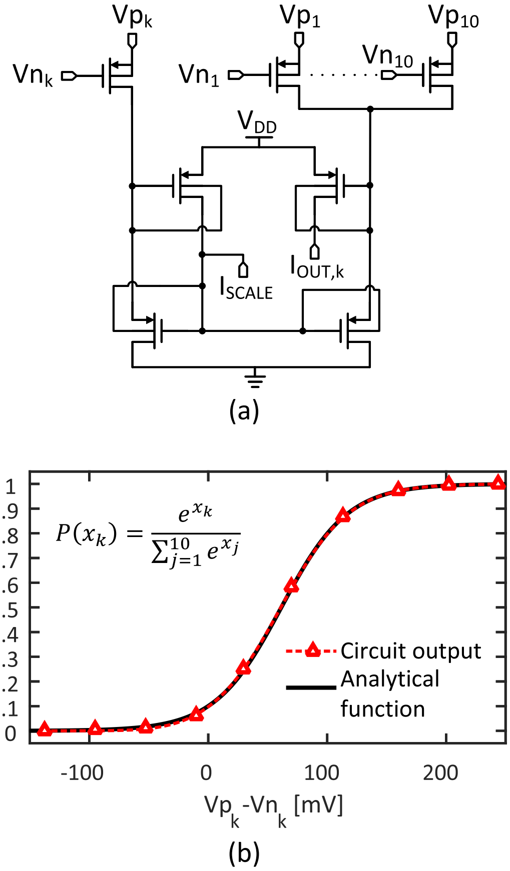


FIGURE 10. **(a) Softmax schematic, (b) circuit output and desired analytical function.**

In particular, a full inference process is performed as follows: at time 0 μs the first image capture takes place, with each pixel collecting electrons during the integration phase. The pixel integration-time window is driven by *Rst* and *TG* signals (see Fig. 4), which realize a correlated double sampling readout with the cooperation of the *SHR* and *SHS* pulses. After a reset pulse, all the pixel FD nodes are depleted and the corresponding *R* signals are sampled at the pixel outputs (triggered by *SHR* pulse). Then, all the charge collected by the PDs is transferred to the *FD* nodes when the *TG* is activated, so that the corresponding *S* signals can be sampled at the pixel outputs (at the *SHS* pulse). Sampled *R* and *S* values are stored in a *S&H* array. The second phase begins at 7.5 μs, with the 25 DVTCs starting the conversion of the ΔV_{pix} outputs into time pulses, which are then issued to the first differential TD-VMM couple, that in turn computes the first layer MAC operations, whose results are labeled as *VoP*, *NL*1.

At the end of this phase, level-one *VoP* and *VoN* signals are sampled in the second *S&H* array, with *R*2 and *S*2 voltages, respectively. At this point, the third phase can begin

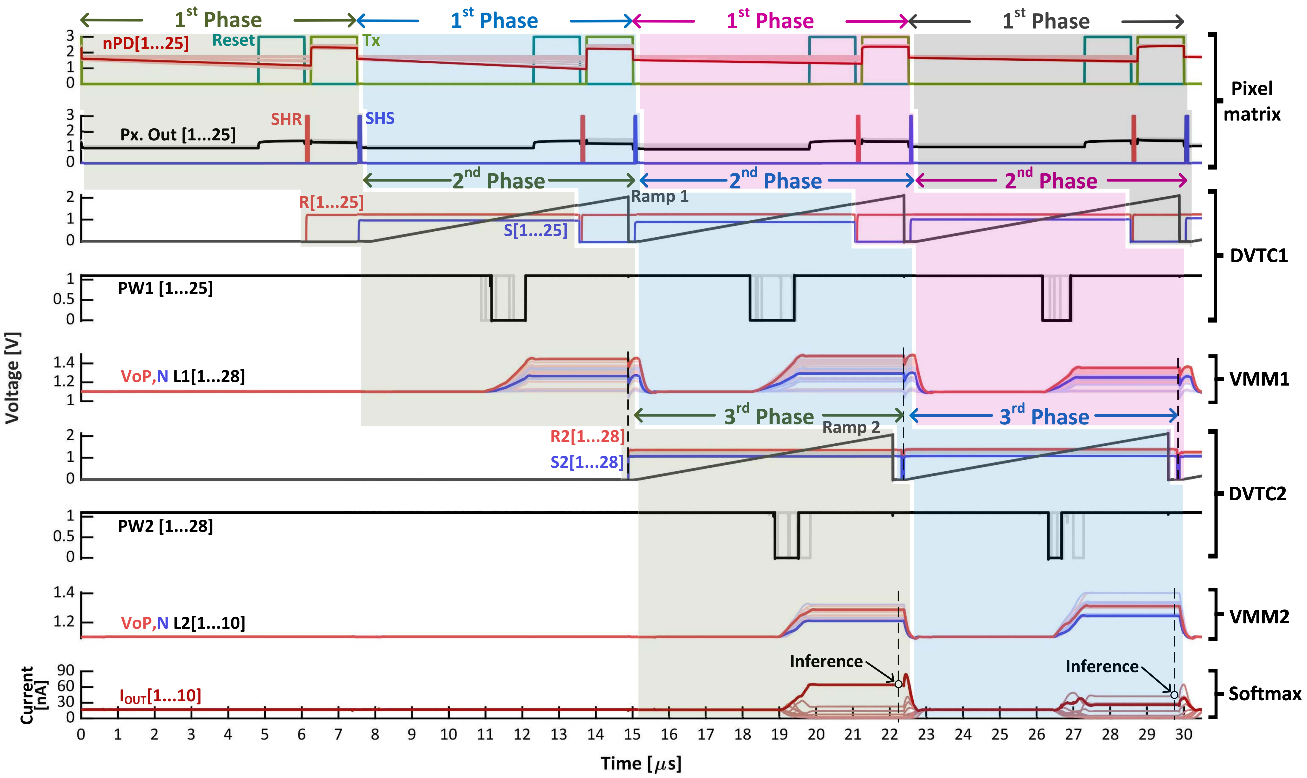


FIGURE 11. Proposed analog ANN timing diagram.

(at 15 μs), with the second DVTC array starting to provide the PWs to the output layer differential VMMs. This time, the differential outputs of the VMM are not sampled, but are directly connected to the softmax circuits performing the final inference evaluation. In conclusion, the very first result will be available after 22.5 μs, while the following inference results will be available every 7.5 μs for a throughput of 133.3 thousand inferences per second.

Due to the complexity of the circuit, the transient analog simulation of the whole network is a highly demanding task in terms of computing resources, and the inference of the whole 10k images set, as originally tested in software, would not be practical. Hence, in order to characterize our network through analog circuit simulations, a benchmark of 500 randomly selected images was used. Simulations were performed including transient noise with a maximum frequency of 100 MHz.

Details of the results obtained with the proposed testbench are reported in the following. The proposed CMOS cognitive image sensor exhibits an inference accuracy of 87.8%, which is comparable with the one we have obtained by means of the idealized software network, with floating point data precision (90.6%). This can be observed from Fig. 12, showing the confusion matrix for both the software network (a) as well as for our hardware implementation (b). For classes related to digits 0, 1 and 9, both networks make exactly the same number of correct inferences. The class of digit 3 is the one in which the hardware network had more incorrect inferences, while, for digits falling in classes 2 and 5, simulations reported more correct inferences than the software case. It is important to clarify that, considering the very low resolution, hand-written digits are very difficult to be properly recognized even by a human eye, and there are occurrences where two softmax outputs have similar values, meaning that the probability of being one of the two classes is very similar.

Beyond evaluating the inference accuracy, we have post-processed the results of all the transient simulations in order to extract the energy consumed by each block. Concerning the energy efficiency, the full circuit consumes on average a total of 6 nJ per inference, where roughly half of the energy is consumed by the pixel sensing matrix, as depicted in Fig. 13(a). These measurements were performed considering the fully loaded pipeline, with each stage executing their own task on data related to different images (see for instance the 4th cycle in Fig. 11).

The same analysis as for nominal devices, has been repeated in order to include the effects of process variability, by considering the four corners (i.e., FF, SS, SF, FS). One should consider that the effect of process variations can have a high impact on the performance of the network, potentially affecting the correct operation of any stage. The most critical stages are the ones performing MAC operations in the VMMs, where the currents of the FG-cells exhibit an exponential dependence on threshold voltage. Fortunately, any

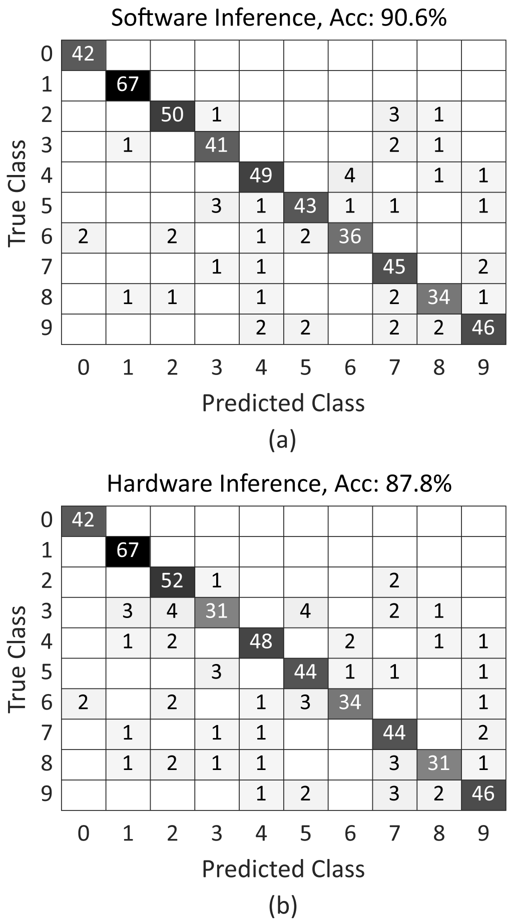


FIGURE 12. **Software neural network (a) and Hardware neural network (b) confusion matrix for a 500-image benchmark.**

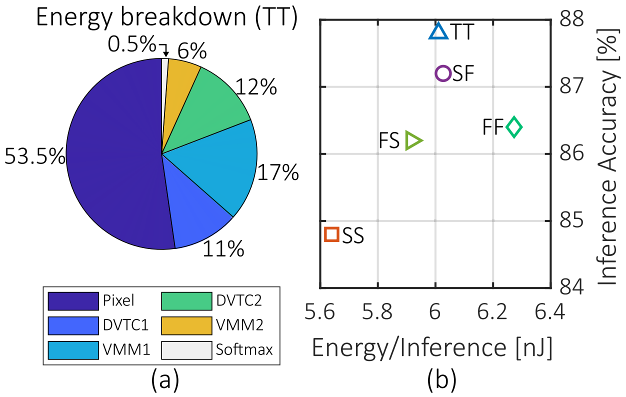


FIGURE 13. **(a) Average energy of each block composing the neural network in the TT corner; (b) inference accuracy vs. average energy/inference for different corner process.**

threshold voltage process/mismatch variation in the FG-cell array can be compensated with an appropriate program-and-verify approach when writing the desired weight. On the other hand, the inference is very sensitive to variations in the softmax circuitry, with no easy correction methods, incurring in the possibility of obtaining process corner sensitivity of inference accuracy.

Performance metrics of the network for different corners are shown in Fig. 13(b), maintaining an average energy per inference between 5.6 nJ and 6.3 nJ and an inference accuracy in the 84% to 88% range across all corners, being TT corner the best in terms of accuracy, since all blocks were optimized for the typical case.

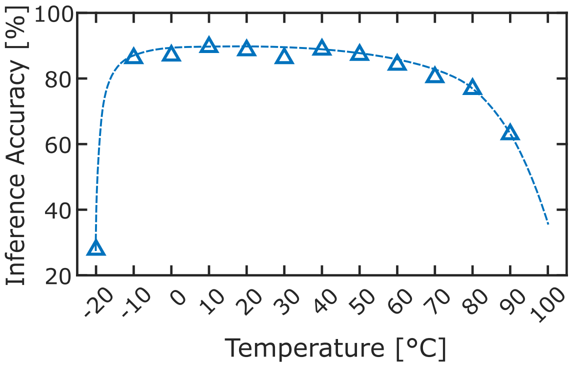


FIGURE 14. **Inference accuracy vs. temperature in the TT corner.**

The other critical variable parameter, which can have a high impact in undermining the correct operation of analog circuits, is the temperature. The assessment of circuit sensitivity to temperature is thus of primary importance. The proposed network was tested in the typical corner for different temperatures ranging from −20 °C to 90 °C (Fig. 14), using the same weight encoding of the nominal temperature (27 °C). There is a broad temperature range, from −10 °C to 70 °C, where an accuracy higher than 80% is ensured, while it steadily falls to lower values outside this range (e.g. 28% at −20 °C, or 62% at 90 °C).

Although the developed design has been investigated only at schematic-level simulation, we have performed an estimate of its potential area occupation on a silicon die. According to such a layout area estimation, based on the transistor, it is found that the intrinsic network (without considering the image sensor pixel matrix) occupies about 1500 μm^2. In Fig. 15 the estimated total area and the pixel matrix area are depicted as a function of the pixel pitch: considering a pixel pitch of 2 μm, the sensing area would be negligible, while with a pitch of 8 μm the pixel matrix would occupy 1600 μm^2, which is close to the silicon area of the ANN.

Table 1 summarizes performance of the proposed CMOS image sensor classifier along with other analog/mixed signal classifiers. The proposed circuit exploits the same ANN topology as in [17], thus a direct comparison can be only made with this work. However, differently from the work described in [17], where weights are hard-coded in the sizing of CMOS devices, our solution provides re-configurable weights thanks to the use of floating-gate memory cells, thus making our design suitable for a broad range of low-resolution image recognition applications.

TABLE 1. Comparison with state-of-the-art.

	JETCAS'2021 [26]	TCAS-I'17 [27]	TCAS-I'2021 [17]	This work
Technology [nm]	180	130	65	180
Fabricated	No	Yes	Yes	No
On-Chip implementation	Partial$^{(a)}$	Partial$^{(a)(b)}$	Partial$^{(a)}$	Full
Area [mm^2]	0.0276/classifier	0.0206/classifier	0.42	0.004 (pixel pitch = 10μm)
Reconfigurable	Yes	Yes	No	Yes
Type	mixed signal	analog	analog	analog
Classifier	Binary	Binary	ANN	ANN
Data set	MNIST	MNIST	MNIST	MNIST
Number of features (number of pixels)	48 (81)	48 (81)	25 (25)	25 (25)
Accuracy [%]	92	90	81.3	87.8
Throughput [classifications/s]	100M$^{(a)}$	1.3M$^{(a)(b)}$	5M$^{(a)}$	133.3k
Energy/classification [J]	67.3p$^{(a)}$	534p$^{(a)(b)}$	173p$^{(a)}$	6n

(a) No sensing stage.
(b) Only binary classifiers implemented, final voting performed off-chip.

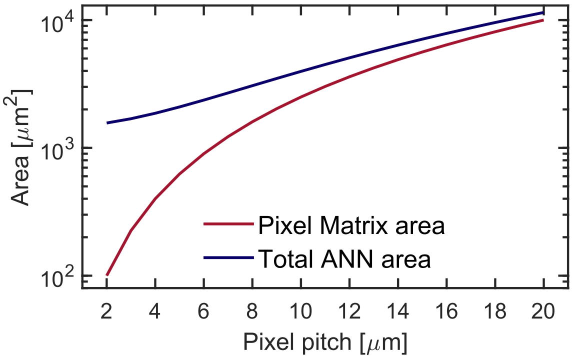


FIGURE 15. Pixel matrix area and total area as a function of pixel layout pitch.

Moreover, our proposal features higher accuracy and occupies an area that is two orders of magnitude smaller (a 1/280 of the area, excluding the pixel matrix or 1/105 of the area assuming a 10 μm pixel pitch array). When focusing the comparison on the figures of merit of energy efficiency and throughput, by considering the specific design point we have optimized to maximize the overall accuracy, the proposal in [17] seems to show better performance, with 34.6× higher energy efficiency (number of inferences per unit energy) and 37.6× higher throughput. However, data reported in [17] neglects the impact on throughput and energy consumption of the pixel sensing matrix, that is considered an external component and therefore should be taken into account separately (note that, the pixel sensing matrix is expected to consume a significant energy, as shown in Fig. 13(a) for our design). Moreover, high performance and energy hungry ADC circuits would be needed to convert within a small time frame the analog signals acquired from the pixel matrix into the digital domain. Also the ADC timing and power overhead is not included in the estimate provided in [17].

It essential to remark that these numbers should not be taken as absolute references, since they are the result of a different design optimization with a different target accuracy. Our design has been optimized for a 6-bit resolution to get an inference accuracy as close as possible to 90%, while the design in [17] has been optimized for a target weight resolution of only 4 bits, resulting in an inference accuracy close to 80%. As a general rule for a given design, the higher the accuracy the higher the energy consumption: this tradeoff can be also observed from Fig. 7 in [17], where the energy consumption increases by more than one order of magnitude moving from 4-bit to 6-bit resolution. Thus, by considering the two classifiers with the same base network architecture (i.e. both designed for the same 6-bit target) and excluding the pixel sensor contribution (not considered in [17]) in our design, we would get comparable energy consumption of a few nJ. On the other hand, the throughput of our chip is directly limited by the integration time of the image sensor pixel matrix. Without considering the image acquisition section, the throughput of our TD-VMM classifier can be boosted up to 1M classifications/s for a 6-bit resolution, and up to 5M classifications/s for a 4-bit resolution design (see details in [15]).

Other works, such as [26], [27], solve the classification problem based on an ensemble of binary classifiers and then perform the network benchmark using the MNIST database downscaled to 48 features, obtained as follows: the original images in the database are resized from 784 to 81 pixels, then Fisher's criterion is applied to the 81-pixel image to further reduce them to 48 pixels. In the case of [26], the network is fully implemented with the exception of the sensing stage, while in [27] only the binary classifiers were implemented, while data input and vote extraction stages were performed off-chip. Again, it is important to stress that a vis-à-vis comparison with works reported [26], [27], only based on throughput and energy consumption data, would be unfair, due to intrinsic differences between the implemented ANN structures and due to the fact that they do not explicitly consider the energy and timing overhead coming from the pixel sensor matrix and the analog-to-digital conversion stage.

V. CONCLUSION

We have presented the design of an all-analog cognitive image sensor, including the hardware implementation of an analog-domain artificial neural network, working as a low resolution image classifier integrated with a 5 × 5 image sensor. Weights are encoded as the injected charge in two-terminal floating-gate devices enabling the circuit to be re-programmable. The proposed CMOS cognitive image sensor was fully designed and simulated in a commercial 180 nm CMOS process, obtaining an accuracy of 87.8% which is comparable to a floating point software implementation (90.6%). It consumes only 6 nJ per inference with a latency of 22.5 μs and throughput of 133k images per second, also exhibiting a small footprint of 4000 μm^2 in the case of a pixel pitch of 10 μm.

REFERENCES

[1] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[2] A. Amaravati, S. Xu, J. Romberg, and A. Raychowdhury, "A 130 nm 165 nj/frame compressed-domain smashed-filter-based mixed-signal classifier for 'in-sensor' analytics in smart cameras," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 65, no. 3, pp. 296–300, Mar. 2018.

[3] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.

[4] B. Jiang, J. Yang, Z. Lv, and H. Song, "Wearable vision assistance system based on binocular sensors for visually impaired users," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1375–1383, Apr. 2019.

[5] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64-mW DNN-based visual navigation engine for autonomous Nano-drones," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8357–8371, Oct. 2019.

[6] S. G. Corporation. (2022) *Imx500 Image Sensor*. [Online]. Available: https://developer.sony.com/develop/imx500/

[7] N. Shavit, I. Stanger, R. Taco, M. Lanuzza, and A. Fish, "A 0.8-V, 1.54-pJ/940-MHz dual-mode logic-based 16×16-b booth multiplier in 16-nm FinFET," *IEEE Solid-State Circuits Lett.*, vol. 3, pp. 314–317, 2020.

[8] 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 J. Solid-State Circuits*, vol. 54, no. 10, pp. 2730–2742, Oct. 2019.

[9] R. Taco, I. Levi, M. Lanuzza, and A. Fish, "An 88 fJ/40 MHz [0.4 V]–0.61 pJ/1 GHz [0.9 V] dual-mode logic 8 × 8 bit multiplier accumulator with a self-adjustment mechanism in 28-nm FD-SOI," *IEEE J. Solid-State Circuits*, vol. 54, no. 2, pp. 560–568, Feb. 2019.

[10] P. C. Knag, G. K. Chen, H. E. Sumbul, R. Kumar, S. K. Hsu, A. Agarwal, M. Kar, S. Kim, M. A. Anders, H. Kaul, and R. K. Krishnamurthy, "A 617-TOPS/W all-digital binary neural network accelerator in 10-nm FinFET CMOS," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1082–1092, Apr. 2021.

[11] W. Guo, H. E. Yantir, M. E. Fouda, A. M. Eltawil, and K. N. Salama, "Toward the optimal design and FPGA implementation of spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3988–4002, Aug. 2022.

[12] J. Yue, Y. Liu, R. Liu, W. Sun, Z. Yuan, Y.-N. Tu, Y.-J. Chen, A. Ren, Y. Wang, M.-F. Chang, X. Li, and H. Yang, "STICKER-T: An energy-efficient neural network processor using block-circulant algorithm and unified frequency-domain acceleration," *IEEE J. Solid-State Circuits*, vol. 56, no. 6, pp. 1936–1948, Jun. 2021.

[13] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proc. IEEE*, vol. 102, no. 9, pp. 1367–1388, Sep. 2014.

[14] M. Paliy, S. Strangio, P. Ruiu, T. Rizzo, and G. Iannaccone, "Analog vector-matrix multiplier based on programmable current mirrors for neural network integrated circuits," *IEEE Access*, vol. 8, pp. 203525–203537, 2020.

[15] T. Rizzo, S. Strangio, and G. Iannaccone, "Time domain analog neuromorphic engine based on high-density non-volatile memory in single-poly CMOS," *IEEE Access*, vol. 10, pp. 49154–49166, 2022.

[16] H. Xu, Z. Li, N. Lin, Q. Wei, F. Qiao, X. Yin, and H. Yang, "MAC-Sen: A processing-in-sensor architecture integrating MAC operations into image sensor for ultra-low-power BNN-based intelligent visual perception," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 2, pp. 627–631, Feb. 2021.

[17] S. T. Chandrasekaran, A. Jayaraj, V. E. G. Karnam, I. Banerjee, and A. Sanyal, "Fully integrated analog machine learning classifier using custom activation function for low resolution image classification," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 3, pp. 1023–1033, Mar. 2021.

[18] S.-W. Yun, Y.-T. Ryu, and K.-W. Kwon, "High linearity vector matrix multiplier using bootstrapping and pre-emphasis charging of non-linear charge-trap synaptic devices," in *Proc. IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2021, pp. 441–444.

[19] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[20] M. Vatalaro, T. Moposita, S. Strangio, L. Trojman, A. Vladimirescu, M. Lanuzza, and F. Crupi, "A low-voltage, low-power reconfigurable current-mode softmax circuit for analog neural networks," *Electronics*, vol. 10, no. 9, p. 1004, Apr. 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/9/1004

[21] Y. LeCun, C. Cortes, and C. Burges. *MNIST Handwritten Digit Database*. Accessed: Mar. 2021. [Online]. Available: http://yann.lecun.com/exdb/mnist

[22] P. Getreuer, "Linear methods for image interpolation," *Image Process. Line*, vol. 1, pp. 238–259, Sep. 2011.

[23] A. M. Brunetti, M. Musolino, S. Strangio, and B. Choubey, "Pixel design driven performance improvement in 4T CMOS image sensors: Dark current reduction and full-well enhancement," *IEEE Trans. Electron Devices*, vol. 67, no. 1, pp. 409–412, Jan. 2020.

[24] B. B. Filgueira, "Modelling and characterization of small photosensors in advanced CMOS technologies," Ph.D. thesis, Departamento de Electrónica e Computación, Universidade De Santiago De Compostela, Santiago, Spain, 2012.

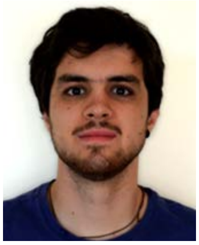
[25] S. Chevella, D. O'Hare, and I. O'Connell, "A low-power 1–V supply dynamic comparator," *IEEE Solid-State Circuits Lett.*, vol. 3, pp. 154–157, 2020.

[26] F. Kenarangi and I. Partin-Vaisband, "A single-MOSFET analog high resolution-targeted (SMART) multiplier for machine learning classification," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 11, no. 4, pp. 816–828, Dec. 2021.

[27] Z. Wang and N. Verma, "A low-energy machine-learning classifier based on clocked comparators for direct inference on analog sensors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 11, pp. 2954–2965, Nov. 2017.

BENJAMIN ZAMBRANO (Graduate Student Member, IEEE) received the B.E. degree in electronics from the Escuela Superior Politecnica del Litoral, Guayaquil, Ecuador, in 2015, and the dual M.S. degree in nanoelectronics and electronics from the University San Francisco de Quito, Ecuador, and the University of Calabria, Rende, Italy, in 2020. He is currently pursuing the Ph.D. degree in ICT with the University of Calabria. His research interests include integrated temperature sensors, ultralow-power/voltage designs, and analog-based machine learning circuits.

SEBASTIANO STRANGIO (Member, IEEE) was with the University of Udine, Udine, Italy, as a Temporary Research Associate, from 2013 to 2016, and with Forschungszentrum Jülich, Germany, as a Visiting Researcher, in 2015, researching on TCAD simulations, design, and characterization of TFET-based circuits. From 2016 to 2019, he was with LFoundry, Avezzano, Italy, where he worked as a Research and Development Process Integration and Device/TCAD Engineer, with main focus on the development of a CMOS Image Sensor Technology Platform. He is currently a Researcher in electronics at the University of Pisa, Italy. He has authored or coauthored over 35 articles, most of them published in IEEE journals and conference proceedings. His research interests include technologies for innovative devices, such as TFETs, and circuits for innovative applications, such as CMOS analog building blocks for deep neural networks (DNNs), as well as CMOS image sensors.



TOMMASO RIZZO received the B.S. and M.S. degrees *(cum laude)* in EE from the University of Pisa, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree in electronics in a joint program with the University of Pisa and Quantavis s.r.l., in the field of analog and mixed-signal IC design using standard and non-standard CMOS technologies. From 2014 to 2019, he was an ''Allievo Ordinario'' at the Sant'Anna School of Advanced Studies, Pisa. In 2017, he was with Fermilab, Batavia, IL, USA, as a Visiting Student, and in 2019, he joined imec, Eindhoven, The Netherlands, working on a wireless powering receiver system for deep implants as his master's thesis project. His research interests include the design of CMOS analog blocks for DNNs and the development of wireless power transfer solutions for IMDs.

ESTEBAN GARZÓN (Member, IEEE) received the Ph.D. degree in electronics engineering from the University of Calabria (UNICAL), Italy, in 2022. He is currently a Postdoctoral Researcher at the Department of Computer Engineering, Modeling, Electronics and Systems Engineering, UNICAL. He has authored/coauthored 30 scientific papers in international journals and conferences, and has participated in several IC tapeouts. His research interests include domain-specific hardware accelerators, and electronics/spintronics, cryogenic memories, and standard and emerging technologies for logic, memory, and low-power applications.

MARCO LANUZZA (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from the Mediterranea University of Reggio Calabria, Reggio Calabria, Italy, in 2005. Since 2006, he has been with the University of Calabria, Rende, Italy, where he is currently an Associate Professor. He has authored or coauthored more than 120 publications in international journals and conference proceedings. His research interests include the design of ultralow voltage circuits and systems, the development of efficient models and methodologies for variability-aware designs, and the design of digital and analog circuits in emerging technologies. He is an Associate Editor of *Integration, the VLSI Journal*.

GIUSEPPE IANNACCONE (Fellow, IEEE) received the M.S. and Ph.D. degrees in electronic engineering from the University of Pisa, Pisa, Italy, in 1992 and 1996, respectively. He is currently a Professor of electronics at the University of Pisa. He has authored or coauthored more than 230 articles published in peer-reviewed journals and more than 160 papers in proceedings of international conferences, gathering more than 8500 citations on the Scopus database. His interests include quantum transport and noise in nanoelectronic and mesoscopic devices, development of device modeling tools, new device concepts and circuits beyond CMOS technology for artificial intelligence, cybersecurity, implantable biomedical sensors, and the Internet of Things. He is a fellow of the American Physical Society.

Open Access funding provided by 'Università di Pisa' within the CRUI CARE Agreement