CAPC: A Configurable Analog Pop-Count Circuit for Near-Memory Binary Neural Networks

F. Jebali, A. Majumdar, A. Laborieux, T. Hirtzlin, E. Vianello, J.P. Walder, Marc Bocquet, D. Querlioz, Jean-Michel Portal

CAPC: A Configurable Analog Pop-Count Circuit for Near-Memory Binary Neural Networks

F. Jebali*, A. Majumdar†, A. Laborieux†, T. Hirtzlin‡, E. Vianello‡,
J.P. Walder*, M. Bocquet*, D. Querlioz†, J.M. Portal*
*Institut Matériaux Microélectronique Nanosciences de Provence, IM2NP, Univ. Aix-Marseille et Toulon, CNRS, France.
†Université Paris-Saclay, CNRS, C2N, 91120 Palaiseau, France.
‡CEA, LETI, Grenoble, France.

***Abstract*—Currently, a major trend in artificial intelligence is to implement neural networks at the edge, within circuits with limited memory capacity. To reach this goal, the in-memory or near-memory implementation of low precision neural networks such as Binarized Neural Networks (BNNs) constitutes an appealing solution. However, the configurability of these approaches is a major challenge: in neural networks, the number of neurons per layer vary tremendously depending on the application, limiting the column-wise or row-wise mapping of neurons in memory arrays. To tackle this issue, we propose, for the first time, a Configurable Analog auto-compensate Pop-Count (CAPC) circuit compatible with column-wise neuron mapping. Our circuit has the advantage of featuring a very natural configurability through analog switch connections. We demonstrate that our solution saves 18% of area compared to non configurable conventional digital solution. Moreover, through extensive Monte-Carlo simulations, we show that the overall error probability remains low, and we highlight, at network level, the resilience of our configurable solution, with very limited accuracy degradation of 0.15% on the MNIST task, and 2.84% on the CIFAR-10 task.**

***Index Terms*—BNN, Analog Pop-Count, Near-Memory**

I. Introduction

Current applications, such as sensor fusion coupled with data analysis [1], emphasize the need for Artificial Intelligence (AI) treatment at the edge. However, the deployment of neural networks on microcontroller units is still limited, on one hand by the high power consumption of computation and data movement from and to embedded memory, and on the other hand by the embedded memory capacity [2]. For this reason, considerable research investigates dedicated AI architectures where logic and memory are closely integrated, following the principles of in-memory or near-memory computation. In this context, Binarized Neural Network (BNN), or the closely related XNOR-NETs, are particularly attractive as, in these networks, both synaptic weights and neural activations are coded during inference with a single binary value [3], [4].

BNNs function with an arithmetic considerably simplified with regards to conventional neural networks. The main equation in conventional neural networks is the computation of the neuronal activation $A_j = f\left(\sum_i W_{ji} X_i\right)$, where A_j, the synaptic weights W_{ji}, and input neuronal activations X_i assume real values, and f is a non-linear activation function.

This work was supported by the European Research Council Grant NANOINFER (715872) and ANR grant NEURONIC (ANR-18-CE24-0009).

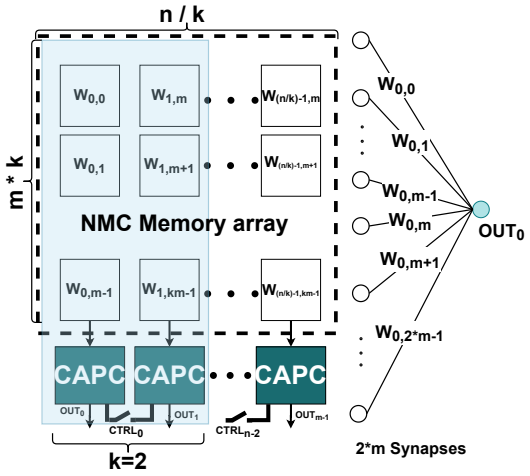


Fig. 1. Schematic overview of the CAPC circuits connection at the bottom of a column-wise NMC memory array, with equivalent neuron model for k=2

This equation becomes, in the case of BNNs, since A_j, and W_{ji} are binary values meaning $+1$ and -1

$$A_j = \text{sign}\left(\text{POPCOUNT}_i\left(\text{XNOR}\left(W_{ji}, X_i\right)\right) - T_j\right), \quad (1)$$

where sign is the sign function, T_j is an integer threshold associated with the neuron, and the POPCOUNT operation counts the number of ones in a series. The low memory requirements and simple arithmetic of BNNs make them particularly adapted to near-memory computation (NMC) using static RAM [5], [6], or emerging non-volatile memory technologies such as resistive RAMs, memristors, or magnetic RAM [7], [8].

In an NMC context, neurons can be mapped either column-wise or row-wise, each configuration having a similar throughput. In a column-wise configuration, the activation is performed through a sequential read of each memory row for all the columns (neuron) in parallel, and the pop-count operation is performed for each column at the bottom of the memory array. For simplifying logic, the XNOR operation is sometimes embedded within the memory sense amplifier [7], [8]. By contrast, in a row-wise configuration, the activation is performed sequentially one row at a time. The pop-count operation is performed through all column with each row activation, with dedicated digital or mixed-signal circuits [5], [6].

A significant challenge is that layers of neural networks often feature more neurons than the number of rows or columns of memory arrays, and the number of neurons per layer varies tremendously depending on applications. Therefore, a strict column-wise or row-wise system has limited applicability, and some reconfigurability is needed. In this context, several studies have been proposed to optimize memory mapping and data-flow at the system level [9]–[11] in dedicated neural network accelerator. However, the configurability has a high circuit overhead cost, and is often ignored in NMC works at the circuit level.

In this work, we propose, for the first time, a Configurable Analog Pop-Count circuit (CAPC), which has the advantage of featuring a very natural configurability, based on capacitor discharge and sum. Moreover, the circuit auto-compensates discharge non-linearity in applying the same counting process to the pop-count value and the threshold value. Each CAPC circuit is located at the bottom of a column of the memory array. The configurability is ensured by a simple connection through analog switches between adjacent column. When this connection is activated, the different columns act as a single one, without needing any other change to the system.

An alternative to our approach is constituted by in-memory computing (IMC) solutions, which exploits Ohm's and Kirchhoff's laws to perform neural network arithmetics, and may present excellent performances when targeting non-volatile memory technologies such as resistive RAMs. However these approaches, require significant overhead circuitry, in particular analog-to-digital converters [10]–[15]. Beyond its configurability, our approach avoids the use of such circuit entirely.

The contributions of this paper are as follows:

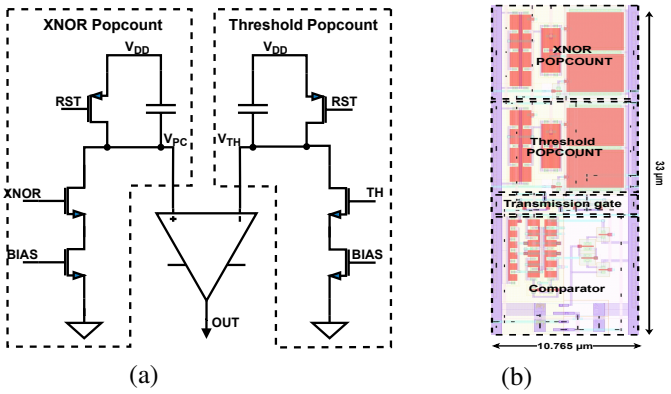
- We introduce, for the first time, a CAPC circuit based on auto-compensate capacitor discharge, compliant with column-wise NMC solutions. (sec. II.A).
- We validate our CAPC solution through extensive Monte-Carlo simulations to extract pop-count/comparison errors for various configurations (sec. II.B).
- We carry simulations at the neural network level to show the impact of configuration choice on the MNIST and CIFAR-10 tasks, and evidence the error resilience of our approach (sec. III).

II. Configurable Analog Pop-Count (CAPC) circuit

Fig. 1 presents our memory architecture with m rows and n columns. The configurability factor k (in the Figure, $k = 2$), allows configuring the number of neurons to n/k and the number of synapses in the memory array to $m \times k$ per neuron. Each column of the memory array features a CAPC circuit, following the column sense amplifier and the XNOR gate (both potentially co-integrated [7], [8]).

A. CAPC description

The CAPC architecture, presented schematically in Fig. 2(a), is mainly composed of five functional blocks:



(a) (b)

Fig. 2. (a) Schematic of the analog pop-count circuit applied to $XNOR$ and Threshold pop-count. (b) Layout view of the CAPC with an area footprint of 355 µm^2

- two identical analog counting blocks, one for the XNOR pop-count and one for the threshold pop-count,
- two identical analog switches to configure the connection to the next right CAPC block, here also one for the XNOR pop-count and one for the threshold pop-count,
- a strongARM comparator [16].

Moreover, a constant current source I_{REF} is shared between all CAPC blocks (see Fig. 3).

The analog counting block, at the core of the CAPC, relies on the discharge of a MOS capacitor with a constant current I_{REF}. The discharge step only occurs when the input signal of the CAPC (XNOR respectively TH) exhibits a 1. Since the capacitance discharge is similar for the XNOR popcount value and for the threshold popcount value, the CAPC auto-compensate the discharge non-linearity. To keep the current mirror transistors in the saturation region and to maximize the voltage capacitor swing (1.8V down to 0.6V), we set the supply voltage V_{DD} to 1.8V, which remains close to safe operation regime (the nominal V_{DD} is 1.2V for the 130 nm technology used to benchmark the proposed solution).

The fully laid out CAPC block (Fig. 2(b)) exhibits an area of $355\mu m^2$. As a benchmark, we synthesized a non-configurable digital solution with the same technology node, using the Synopsys Design Vision tool. The benchmark digital solution is based on a down-counter and a comparator to 0. The popcount process starts with a pre-charge of the down-counter to the threshold value. During the pop-count process, the down-counter value decreases with each XNOR value equal to 1. At the end of the pop-count process, the down-counter value is compared to 0 to activate the neuron. We found the area of the non-configurable digital benchmark to be 18% higher than our CAPC circuit.

In addition to its area efficiency, the major advantage of the CAPC circuit is its reconfigurability. The configuration principle is based on connecting k CAPCs together by activating analog switches between neighbor CAPCs, as illustrated in Fig. 3 for $k = 2$. Doing so, the capacitors are connected in parallel, thus adding the capacitance value shared between k columns, for the XNOR pop-count part on one side and for the threshold pop-count part on the other side. After doing so,

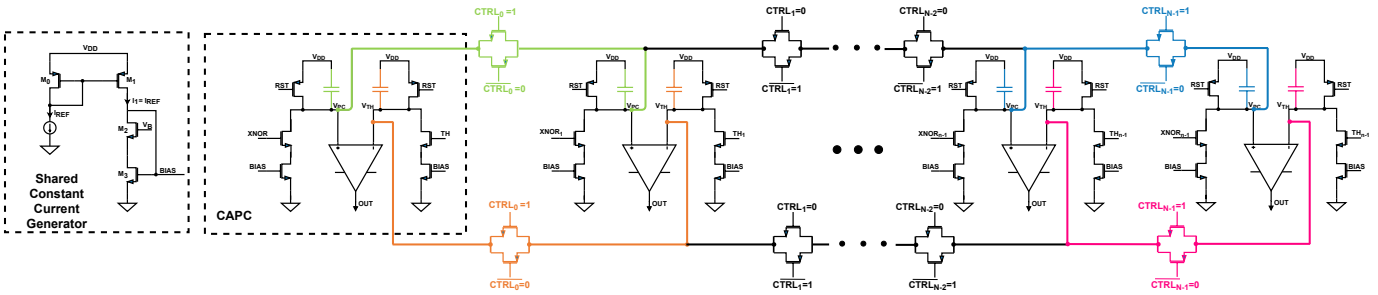


Fig. 3. Schematic of the connection of two by two CAPC blocks (k=2), to illustrate the parallel connection of the adjacent capacitances

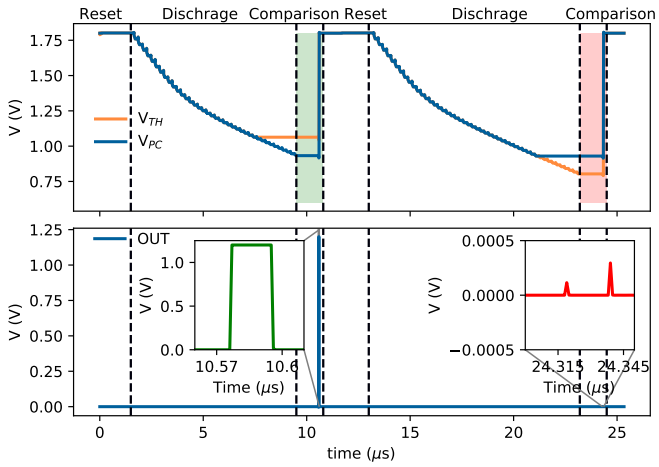
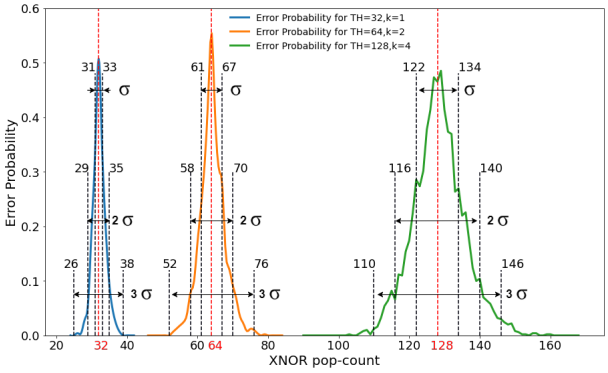


Fig. 4. Results of functional simulation of the CAPC circuit, with m=64, k=1, with $V_{TH} > V_{PC}$ meaning that the threshold value is lower than the pop-count value (green region), thus activation value (OUT) is set to one, followed by $V_{TH} < V_{PC}$ meaning that the threshold value is larger than the pop-count value (red region), thus activation value (OUT) is set to zero



(a)

k (# value)	1	2	4
Synapses per neuron (# value)	64	128	256
Full range mean activation error (%)	2.87	3.04	2.82
σ of XNOR pop-count (# value)	1.5	3	6

(b)

Fig. 5. Estimated error range and error probability extracted from Monte-Carlo simulation of the proposed CAPC circuit for k ranging from 1 to 4.

a single comparator is used to deliver the activation output.

With this technique, the counting capability of the solution is multiplied by k, keeping the same voltage discharge full range (1.8V down to 0.6V) with a voltage step divided by k and the same reference current I_{REF} for each block. The advantage of the solution is to keep the auto-compensation capability whatever the number k of CAPCs connected together.

The working scheme of our solution, is divided into three successive phases, illustrated in Fig. 4 for two different cases (XNOR pop-count value above and below threshold pop-count value). First, a reset phase occurs grounding the RST signal, and thus charging all capacitors to V_{DD}. Then, the RST signal is inactivated (RST signal tied to V_{DD}) and the discharge process starts for both pop-count and threshold block, sequenced by the system clock in $m \times k$ steps. The threshold bit stream has to be divided between the different CAPC blocks to fit the $m \times k$ steps. After $m \times k$ steps, the comparison occurs by activating the comparator of one block from the k blocks in parallel, generating the binary activation output value.

B. CAPC - Validation

Fig. 4 shows the simulation of the CAPC circuit in typical situations and validates its functionality. Additionally, we performed extensive Monte-Carlo simulations (500 runs, with global and local variations at three sigmas, including mismatch) for different scenarios. The simulated scenarios cover the full range of the XNOR popcount values, compared to the full range of the threshold popcount values, for reconfigurability factors $k = 1$, 2, and 4, with $m = 64$. Fig. 5(a) presents the error extraction process in different cases where the threshold value is set to the middle of the popcount value range, for $k = 1$, 2, and 4. A similar extraction procedure was performed in all the considered scenarios, and Fig. 5(b) summarizes the obtained results. As expected, the standard deviation of the XNOR popcount values leading to error is doubled when k is doubled. The voltage step corresponding to one popcount, is indeed divided by k.

The maximum error is reached when the XNOR popcount value and threshold popcount values are close: the error probability as a function of XNOR popcount value follows a Gaussian function centered around the threshold popcount value. The standard deviation of this Gaussian function is very low: 1.5, 3 and 6 popcount values for respectively $k = 1$, 2 and 4 (and thus 64, 128 and 256 synapses per neuron). Out of the three-sigma range of the Gaussian function, the CAPC is error free, whatever was the configuration. This result also

shows that the corresponding voltage range remains constant, whatever was the configuration, since the ratio between capacitance value and discharge current remains constant.

III. Network level estimations

We now use the errors distributions extracted for all scenarios as input for neural network simulation. More precisely, to assess the impact of the errors on BNN accuracy during inference, we perform simulations of a fully-connected architecture for the handwritten digit recognition (MNIST) task and of a convolutional architecture for an object classification (CIFAR-10) task. We train the networks in an ideal setting and only introduce the errors during inference. The probability of error in the comparison of XNOR popcount and threshold is modeled as to follow a Gaussian function with a zero mean and a standard deviation proportional to the number of neurons, extracted from the results of the Monte Carlo simulations shown in Fig. 5(a).

The fully-connected network employed for the MNIST task had a single hidden layer with 3,000 neurons, and it showed, due to the errors, a very small mean accuracy degradation of 0.15% on the test dataset. For the more difficult CIFAR-10 task, six convolutional layers followed by three fully-connected layers are used, as is common for convolutional neural networks. The erroneous thresholding was not used for the first layer, as the input to a BNN is not typically binarized, and thus we cannot use such circuits. We tested the impact of CAPC errors on the fully-connected layers, as they are the most memory intensive and the ones for which CAPC is the most adapted. This task showed a small mean accuracy degradation of 2.84%. Without errors, the network is trained to have an accuracy of 90.09%. When errors are included, this accuracy degrades to 87.25%, on average, with a standard deviation of 0.13%.

Fig. 6 shows the detailed impact of the errors on the CIFAR-10 task: to see the impact of the errors on individual fully-connected layers, we remove the erroneous comparison for each of those layers successively. The results suggest that the first fully-connected layer of the network is the most sensitive to errors.

IV. Conclusion

In this work, we propose, for the first time, a Configurable Analog Pop-Count circuit, suitable for Near-Memory Computing solutions where neuron mapping is performed in a column-wise fashion. The proposed circuit has been laid out, showing an area reduction of 18% when compared to non configurable classical digital pop-count implementation. The CAPC circuit has been simulated to assess the auto-compensation of capacitance discharge non-linearity between XNOR pop-count value and threshold pop-count value. Simulation results, accounting for global as well as local variability (500 MC runs), show low overall activation error probability. From the error probabilities extracted at the circuit level, neural network simulations has been carried out. The simulations results, with less than 3 percentage points reduction in inference accuracy, confirm the resilience of the approach. These results open the way to improve neural network mapping on various memory array sizes, through configurability at the circuit level.

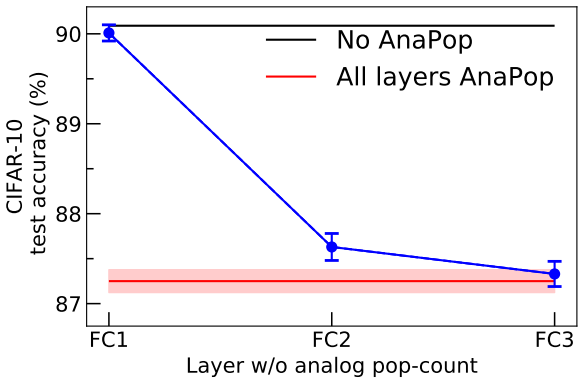


Fig. 6. Influence of the error introduced because of the analog pop-count on the CIFAR-10 image classification task. The inference accuracy for analog pop-count in no layer (black), all fully-connected (FC) layers (red), and when excluded in various FC layers (blue).

References

[1] B. Reese, "Ai at the edge: A gigaom research byte," *GigaOm*, 2019.

[2] X. Xu *et al.*, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, no. 4, p. 216, 2018.

[3] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[4] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proc. ECCV*. Springer, 2016, pp. 525–542.

[5] T. Wang and W. Shan, "An energy-efficient in-memory bnn architecture with time-domain analog and digital mixed-signal processing," 2019.

[6] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 μj/86% cifar-10 mixed-signal binary cnn processor with all memory on chip in 28-nm cmos," *IEEE JSSC*, vol. 54, p. 158, 2019.

[7] E. Giacomin, T. Greenberg-Toledo, S. Kvatinsky, and P.-E. Gaillardon, "A robust digital rram-based convolutional block for low-power image processing and learning applications," *IEEE Trans Circuits Syst I Regul Pap*, vol. 66, no. 2, pp. 643–654, 2019.

[8] T. Hirtzlin *et al.*, "Digital biologically plausible implementation of binarized neural networks with differential hafnium oxide resistive memory arrays," *Frontiers in neuroscience*, vol. 13, p. 1383, 2020.

[9] Y. Kim, H. Kim, D. Ahn, and J. J. Kim, "Input-splitting of large neural networks for power-efficient accelerator with resistive crossbar memory array." Institute of Electrical and Electronics Engineers Inc., 7 2018.

[10] H. Jia, M. Ozatay, Y. Tang, H. Valavi, R. Pathak, J. Lee, and N. Verma, "15.1 a programmable neural-network inference accelerator based on scalable in-memory computing," in *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 64, 2021, pp. 236–238.

[11] T. Tang, L. Xia, B. Li, Y. Wang, and H. Yang, "Binary convolutional neural network on rram," in *Proc. ASP-DAC*, 2017, pp. 782–787.

[12] M. Prezioso *et al.*, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, p. 61, 2015.

[13] A. Shafiee *et al.*, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proc. ISCA*, 2016, p. 14.

[14] L. Song, Y. Wu, X. Qian, H. Li, and Y. Chen, "R e bnn : in - situ acceleration of binarized neural networks in reram using complementary resistive cell," *CCF THPC*, 2019.

[15] S. Ambrogio *et al.*, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, p. 60, 2018.

[16] B. Razavi, "The strongarm latch : A circuit for all seasons," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, 2015.