An Energy-Efficient and High Throughput in-Memory Computing Bit-Cell With Excellent Robustness Under Process Variations for Binary Neural Network

GOBINDA SAHA\textsuperscript{1,3}, ZHEWEI JIANG\textsuperscript{1,4}, SANJAY PARIHAR\textsuperscript{2}, (Member, IEEE), XI CAO\textsuperscript{1}, JACK HIGMAN\textsuperscript{2}, AND MUHAMMED AHOSAN UL KARIM\textsuperscript{1}

\textsuperscript{1}GLOBALFOUNDRIES, Santa Clara, CA 95054, USA
\textsuperscript{2}GLOBALFOUNDRIES, Austin, TX 78735, USA
\textsuperscript{3}Purdue University, West Lafayette, IN 47907, USA
\textsuperscript{4}Columbia University, New York, NY 10027, USA

Corresponding author: Muhammed Ahosan Ul Karim (rumi.ahosanulkarim@globalfoundries.com)

ABSTRACT In-memory computing (IMC) is a promising approach for energy cost reduction due to data movement between memory and processor for running data-intensive deep learning applications on the computing systems. Together with Binary Neural Network (BNN), IMC provides a viable solution for running deep neural networks at the edge devices with stringent memory and energy constraints. In this paper, we propose a novel 1T bit-cell with a back-end-of-line (BEOL) metal-oxide-metal (MOM) capacitor laid on pitch for in-memory computing. Our IMC bit-cell, when arranged in a memory array, performs binary convolution (XNOR followed by Bit-count operations) and binary activation generation operations. We show, when binary layers of BNN are mapped into our IMC arrays for MNIST digit classification, 98.75% accuracy with energy efficiency of 2193 TOPS/W and throughput of 22857 GOPS can be obtained.

We determine the memory array size considering the word-line and bit-line nonidealities and show how these impact classification accuracy. We analyze the impact of process variations on classification accuracy and show how word-line pulse tunability provided by our design can be used to improve the robustness of classification under process variations.

INDEX TERMS In-memory computing, SRAM, binary neural network, nonideality, process variation.

I. INTRODUCTION

Recent success of deep learning [1] in various domain specific tasks involving object recognition, classification and decision making [2] is largely driven by the architectural innovations of neural network commonly known as Deep Convolutional Neural Networks (DCNNs) [3], [4]. However, high performance of DCNNs comes with huge memory requirements for storing the network parameters for computation which constrains the deployment of such networks at edge devices for mobile applications. Moreover, data-intensive deep learning applications, when implemented in conventional von-Neumann hardware architecture, require frequent memory access and data movement between memory and computation units which can overweigh the energy cost of computation [5] and add large energy overhead.

Binary Neural Networks (BNNs) [6] provide a hardware-friendly solution to these problems without sacrificing the state-of-the-art DCNN accuracy. BNN and its variants (e.g. XNOR-Net [7]) reduce the precision of the network weights to single bit. Such extreme quantization opens up the prospect of storing BNN weights fully in the on-chip memory and reduces the energy consumption for off-chip data access. Moreover, energy intensive multiply-and-accumulate (MAC) operations of DCNNs are reduced to simple bit-wise XNOR and subsequent pop-count operations in BNNs which offers large computational energy saving. Nevertheless, even BNNs, the memory and energy efficient variant of DCNN, consume orders of magnitude more energy than the human
brain for similar recognition or classification tasks [8]. One of the prime reasons for this gap is that in von-Neumann architecture, unlike in the human brain, memory and computation units are physically separated and data movement between these two units incurs large energy cost.

To cut down the energy cost of data movement, the concept of In-Memory Computing (IMC) [9]–[27] has been proposed and applied to accelerate the computations in neural networks including BNN. Although most of these prior works showed potential for high energy efficiency, design concerns such as computational reliability degradation with increasing simultaneous memory row access [11] exists. Modern day high-performing neural networks usually have large filter sizes in different layers. Therefore, convolution operations (XNORs and pop-counts in BNNs) involving those filters would require large data row access in the memory concurrently. Hence, the issue of computational reliability due to signal-to-noise-ratio (SNR) degradation needs to be solved in order to achieve high computational throughput. Additionally, although IMC designs such as local bit-line grouping [14] and word-line underdrive [11] is proposed in the literature to address data corruption issue, in general, 6T based IMC designs are prone to memory disturb [11]–[15].

To increase the computational scalability of in-memory computing, Valavi et al. [18] proposed charge-domain computation based on a linear and stable metal-oxide-metal (MOM) finger capacitor. They augmented a standard 6T SRAM bit-cell with two PMOS transistors to conditionally charge the MOM capacitor that has been laid over each bit-cell. Although accessing multiple memory rows of such an IMC bit-cell configuration improves computational SNR, [18] reports a relatively long time of 250 ns for one convolution operation involving charge, discharge and reset phase of the MOM capacitor.

In an effort to reduce the time per convolution operation, increase energy efficiency and at the same time ensure the scalability of in-memory computation, we propose a novel IMC bit-cell configuration with a 10T SRAM bit-cell augmented with a BEOL MOM capacitor on pitch. Additionally, use of the NFET read stack pairs, which decouple the read path from 6T storage unit, enable large voltage swing on the read bit-line to increase the signal margin without triggering read-disturb issues commonly observed for a 6T bit-cell based design [13], [15]. Unlike Valavi et al. [18], we precharge all of the MOM capacitors of a memory column at the beginning of each operation and then conditionally discharge them based on the XNOR conditions. We show that our in-memory computation scheme, involving discharge, compare and precharge phase, takes ~70x less time per convolution operation. Moreover, in our scheme, we have eliminated the need for area and energy hungry ADCs [19] and Multi-level Sense Amplifiers (MLSAs) [20] for pop-counting operation. Rather, in the compare phase, we use a comparator at the end of each memory column to compare the bit-line voltage after the discharge phase (XNORs in IMC bit-cells) against a predefined threshold voltage and generate binary activation outputs for next layer. Furthermore, we show that the read word-line voltage amplitude and pulsewidth tunability provided by our design enables us to control the average energy consumption per convolution operations thus providing a knob to increase energy efficiency. We also show the usefulness of this on-Silicon tunability feature in improving network classification accuracy in the presence of process variation. Overall, the main contributions of our work are as follows:

- We propose a novel IMC bit-cell, with 10T SRAM bit-cell and a BEOL MOM capacitor laid on-pitch, that can perform binary convolution operations when organized in a memory array with a comparator per column.
- We show that 98.75% classification accuracy for MNIST dataset with high energy efficiency and throughput can be obtained when a trained BNN is mapped onto our IMC hardware for data processing.
- We analyze the impact of word-line and bit-line non-idealities on classification accuracy and determine the optimum memory sub-array size.
- We perform an analysis of read bit-line discharge voltage variability in the presence of process variations by Monte Carlo simulations using GLOBALFOUNDRIES 22FDX® PDK. We show tight classification accuracy distribution under process variations. We also show how on-silicon tunability can be used to reduce the impact of process variations further.

The rest of the paper is organized as follows: Section II describes the operation of our IMC bit-cell and how we perform binary convolution and activation generation when these cells are organized in a memory array. Section III shows the impact of nonidealities in determining the memory array size and reports the BNN classification accuracy on MNIST dataset when implemented on our IMC bit-cell arrays. In section IV, we analyze the impact of process variations on the classification accuracy and show how on-silicon tunability can be useful in mitigating the impacts. We discuss and compare our results with state-of-the-art implementation in section V and in section VI we conclude the paper.

II. IMC BIT-CELL AND BINARY CONVOLUTION

A. IMC BIT-CELL CIRCUIT AND XNOR OPERATION

A schematic of our proposed IMC bit-cell is shown in Fig. 1. In our design, we use a conventional 6T SRAM bit-cell for data storage. Complementary storage of a bit-cell is denoted by W and Wb. We augment the design with two readports (read-port 1 with $M_1 - M_2$ NMOS, read-port 2 with $M_3 - M_4$ NMOS) to form the 10T bit-cell structure which provides read stability. However, unlike conventional 10T SRAM bit-cell, where read-ports are differentially connected, in our design we tie them to the same read bit-line (RBL). Additionally, without area penalty, a BEOL MOM capacitor is laid above each bit-cell on-pitch and tied to the RBL.
The MOM capacitor is shared among all cells on column and its capacitance scales with the column height. The use of large MOM cap in RBL is an intentional design choice to limit the capacitance of RBL to an area of 1 μm² and has ∼ 1.2 fF of MOM capacitance. The cell layout is shown in Fig. 2(a) for the active and poly layers where the additional read-ports needed for IMC operation double the cell height compared to a 6T cell. Fig. 2(b) shows how the MOM backend capacitor, which uses metal 6 to 8 and is placed above the SRAM WL metal layers.

The MOM capacitor is shared among all cells on column and its capacitance scales with the column height. The use of large MOM cap in RBL is an intentional design choice to limit the capacitance of RBL to an area of 1 μm² and has ∼ 1.2 fF of MOM capacitance. The cell layout is shown in Fig. 2(a) for the active and poly layers where the additional read-ports needed for IMC operation double the cell height compared to a 6T cell. Fig. 2(b) shows how the MOM backend capacitor, which uses metal 6 to 8 and is placed above the SRAM WL metal layers.

The write operation into our IMC cell is similar to the conventional 6T write operation through write ports (WWL, BL, BLB). For reading, at first, RBL is precharged to ‘VDD’ then the complementary inputs of the read word-lines (RWL, RWLB) are activated. Depending on the stored value on the storage nodes (W,Wb) and read word-line inputs, bit-line voltage will either discharge through any of the read-port to ground or no discharge will happen. The table in Fig. 1 shows discharge conditions for various combinations of RWL inputs and stored weight values (W) where a RBL discharge is defined as logic ‘1’ and no discharge is defined as logic ‘0’. For RWL of ‘0’ (RWLB of ‘1’) and W of ‘0’ (Wb of ‘1’), M1 and M4 NFETs are ‘ON’ and RBL discharges through the M3 − M4 stack. Similarly, for RWL of ‘1’ (RWLB of ‘0’) and W of ‘1’ (Wb of ‘0’), M1 and M2 NFETs are ‘ON’ and RBL discharges through the M1 − M2 stack. For the rest of the two cases RBL does not discharge as one of the NFETs in either M1 − M2 or M3 − M4 stack is turned ‘OFF’.

B. BINARY CONVOLUTION IN IMC ARRAY

In a neural network, convolution is defined by the dot product between input patches and filters. In BNN, as operands of the dot products are binary (+1/−1), the convolution operation can be reduced to binary convolution which is XNOR followed by bit-counting of the XNOR’ed output [6], [7]. As our IMC bit-cell can perform XNOR operation, it can be readily used to perform binary convolution operations. To do so, IMC cells are arranged in a memory array and weights of each filter of the BNN are mapped (written) into each column. One such memory column is shown in Fig. 3(a), where input patches are mapped as read word-line (RWL) voltage pulses. A collection of these columns produces the output feature map of a BNN layer where each column performs convolution and binarization operation. Notably, binary levels of +1 and −1 are mapped as logical ‘High’ and ‘Low’ in our design.

In our method, binary convolution operation has three phases (shown in Fig. 3(b)): Discharge, Compare and Precharge phase. At the start of the computation, RBL is charged to VDD by turning on precharge PMOS, Mpch. During this phase, all the read word-lines (both RWL and RWLB) are kept at logic ‘Low’ thus cutting off all the discharge paths. Next, in the discharge phase, Mpch is turned off and complementary read word-line pulses are set according to the binary input activations. Read word-line pulses trigger discharge conditions in different IMC cells in the column (as shown by the green arrow in Fig. 3(a)) and the RBL keeps discharging (shown in Fig. 3(b)) until the RWL pulses are turned off marking the end of the discharge period. Final RBL voltage (VRBL) after this period reflects the dot product value of the input (I) and weight (W) vector. In the compare phase, we use a comparator circuit to compare VRBL with a reference voltage
will be lower (higher) than the $V_{REF}$ and comparator produce Logic ‘High’ (‘Low’). Therefore, we can leverage our method of in-memory computing to eliminate the need for ADCs or MLSAs for efficiently computing binary activations from the XNOR’ed output from the IMC cells.

Word-line pulse amplitude, $V_{RWL}$ and pulselength, $T_{PW}$ are chosen to meet certain design specifications. One such case is the selection of comparator reference voltage level, $V_{REF}$. Fig. 3(b) shows 50% RBL discharge condition for different $V_{RWL}$ amplitudes. The solid line indicates the $V_{RWL}$ and $T_{PW}$ conditions found for 50% RBL voltage level, hence $V_{REF}$, to be at $V_{DD}/2$. If $V_{RWL}$ is increased (decreased) keeping $T_{PW}$ fixed, 50% RBL discharge level will go down (up) due to faster (slower) RBL discharge as indicated by dashed blue (red) line. Thus, tunability of $V_{RWL}$ and $T_{PW}$ provides us flexibility in the $V_{REF}$ level selection.

Finally, after the binary convolution and activation generation is finished, a precharge phase is initiated for the next convolution cycle by turning on the $M_{pch}$ while turning off all the RWLs to restore the RBL voltage level to $V_{DD}$. The amount of energy required for the restoration is the dominant dynamic energy consumption during a binary convolution operation. Moreover, with the way we perform binary convolution, we expect 50% RBL voltage discharge on average. From the energy efficiency point-of-view, therefore, the choice of $V_{REF}$ level is important as we are expected to precharge the $V_{DD} - V_{REF}$ voltage in the RBL for a typical convolution cycle. This, as we will see in section IV, will provide a knob for energy-efficiency/accuracy tuning.

III. MNISt CLASSIFICATION UNDER IMC ARRAY NONIDEALITIES

We built an in-house simulation framework that trains the BNN on MNIST [29] dataset with Python and Pytorch [30] libraries. Within the framework, we then map the binary network weights and activations in the IMC array with the comparator and simulate the corresponding circuit in HSPICE using the GLOBALFOUNDRIES 22FDX® technology models. In this work, we used post-layout extracted netlist which gives the most accurate results obtainable by simulation. Computations of full precision layers (first & last layer) are done in software (Python). Ultimately, we extract the accuracy, energy and timing results using combined Python-HSPICE simulation framework.

A. BNN TRAINING ON MNIST

Table 1 shows the BNN network structure that we have used for our simulation. This is a simplified LeNet architecture [31] which consists of two convolution (Conv.) layers and two fully connected (FC) layers. The first and last layers are full precision layers [7] and we process these layers in software (Python). Conv. 2 and FC 1 layers are binary layers and notably, they have almost 99% of the full network parameters. Therefore, performing in-memory computation in such layers provides large energy efficiency and saves computation time. Full software implementation of full precision version of our
TABLE 1. BNN used for MNIST classification. Activation size and filter size are represented in (input channels, height, width) and (output channels, input channels, kernel height, kernel width) formats respectively. First layer has input activation of size (1,28,28). No bias units for Conv. 2 and FC 1.

| Layer | Output Activation Size | Filter Size | # Parameters |
|-------|------------------------|-------------|--------------|
| Conv 1 | (20,34,24)            | (20, 1, 5, 5) | 520          |
| Maxpool 1 | (20,12,12)          | (2, 2)     | 0            |
| Conv 2 | (30,6,6)             | (50, 20, 5, 5) | 25,000       |
| Maxpool 2 | (30,4,4)         | (3, 3)     | 0            |
| FC 1   | (500)                 | (800, 500) | 400,000      |
| FC 2   | (10)                  | (500, 10)  | 510          |

BNN architecture provides 99.21% classification accuracy. Whereas its binarized version (with binarized Conv. 2 and FC 1 Layers), with some minor accuracy degradation, provides 99.03% accuracy in software implementation. We train and test the binary network for MNIST classification without any batch normalization [32] layers and without any bias units in the binary layers. This enabled us to use uniform voltage threshold (e.g. 50% RBL discharge) for all the neurons (comparators) for binarization. For MNIST dataset, such design choice did not lead to any accuracy degradation while it reduced the complexity of the peripheral circuits.

B. IMC ARRAY SIZING AND IMPACT OF NONIDEALITIES

Mapping of binary Conv. 2 layer weights requires a memory array size of 500 × 50 and 800 × 500 for binary FC1 layers. Since memory array size impacts the throughput of the design, determining the size of the memory sub-array is critical. In our design, we consider two main non-ideal factors that limit the size of the memory sub-array: word-line non-ideality and bit-line nonideality. In modeling such nonidealities, we consider word-line and bit-line parasitic resistance and capacitance which is calibrated with GLOBALFOUNDRIES 22FDX® technology.

In our implementation, we use $V_{DD} = 0.8$ V. We tune and set the word-line pulses at $V_{RWL} = 0.37$ V and duration of the discharge phase at $T_{PW} = 1.2$ ns to set the 50% discharge condition, hence $V_{REF}$, to $V_{DD}/2$ (0.4 V). Fig. 3 (a) implies that the same word-line pulses will be shared by the IMC cells in the same row in the memory array. When parasitic word-line resistance and capacitance are considered, there will be a RC-delay of the RWL pulses fed from the near-end IMC cells to far-end IMC cells along the memory row. This effectively means that the far-end IMC cells would get less discharge time than the near-end cells, since RWL pulse will be cut-off for all the IMC cells in a row after a constant time, $T_{PW}$. As a result, for the same discharge condition, RBL voltage at the farthest column of the same row would be higher than the near-end column (ideal discharge level). The RBL voltage difference between the near-end and far-end columns for 50% discharge condition are plotted in Fig. 4(a). For example, if 100 columns are used, for 50% discharge condition, $RBL_{100}$ will be 400 mV while $RBL_{100}$ is found to be 412.2 mV which results in a RBL difference of 12.2 mV. From the plot, we also observe that as the number of columns is increased this difference keeps increasing. Since $V_{REF}$ is the same for all the end-of-the-column comparators, this result indicates that as the number of columns increases, positive bit-count values will be more likely interpreted as negative by the comparators at the far-end columns.

Fig. 4 (b) shows the impact on such nonidealities on MNIST classification accuracy. From the plot we observe that if number of columns is increased beyond a certain number (e.g. 150), classification accuracy degrades drastically. Although a high number of columns are desired for high throughput, we select a memory sub-array with 100 columns to attain high classification accuracy (98.75%). For bit-line nonidealities, we simulated 500 and 800 IMC cells in a single column with 50% discharge condition and observe that only ~1 mV RBL discharge voltage difference between these two cases. We observed such negligible impact of RBL nonidealities on discharge voltage as RC discharge time constant depends mainly on the on-state resistance of the read-stack transistors and MOM capacitors ($C_{MOM}$) as they overshadow the parasitic resistance and capacitance of the RBL. Therefore, for the given BNN mapping we use one 500 × 50 sub-array for Conv.2 layer and five 800 × 100 sub-arrays for FC1 layer.

IV. IMPACT OF PROCESS VARIATION

Since process variation plays a critical role in the operation of circuits composed of nano-scale devices, we perform an analysis to determine the impact of device level variations on the discharge voltage and overall classification accuracy of BNN. Since RBL voltage primarily depends on the discharge currents through the read-ports, we consider variations of the NMOS devices in these paths. Fig. 5 shows histogram of RBL voltage variation for 50% discharge condition (with $V_{RWL} = 0.37$ V, $T_{PW} = 1.2$ ns for 800 IMC cells) from
FIGURE 5. Histogram of RBL voltage variation for 50% discharge condition from Monte Carlo simulation for case 1: $V_{RWL} = 0.37$ V, $T_{PW} = 1.2$ ns.

FIGURE 6. Quantification of disturb margin from $3\sigma$ variation of RBL voltage.

10,000 Monte Carlo simulations. Red line (distribution fit) shows that RBL voltage variation follows a normal distribution. Since the on-silicon tunability feature of our design can trim the impact of global variations in the transistors and MOM capacitors, we only considered local variations in our simulations.

A. DISTURB MARGIN

To quantify the impact of RBL variation on the classification accuracy, we define the disturb margin as the range of discharge levels that are prone to misclassification due to variation. Fig. 6 shows how we calculate disturb margin for $3\sigma$ variations. From the figure it is evident that, left-tails of RBL voltage distributions which are less than 50% discharge levels and right-tails of RBL voltage distributions which are greater than 50% discharge levels are at risk of misclassification due to variations. We define a discharge level as misclassified if its left or right distribution tail crosses the reference level (mean of 50% discharge level). Table 2 shows disturb margin for $3\sigma$ variation cases for different memory row sizes. For example, disturb margin of 50(±3)% for 800 rows means that, discharge levels from 47% to 53% will be misclassified in $3\sigma$ cases by the comparators.

B. IMPACT OF PROCESS VARIATION ON ACCURACY

From RBL discharge voltage distributions and disturb margin calculations, we determine the classification accuracy distributions (shown in Fig. 7(a)-(c)) of our network from Monte Carlo simulations. From Fig. 7(a) we observe that, for $V_{RWL} = 0.37$ V, $T_{PW} = 1.2$ ns word-line voltage configuration (case 1), we obtain a mean classification accuracy of 98.23% with a standard deviation of 0.39%, which is a slight degradation from the ideal case (98.75%). To improve mean classification accuracy and reduce the classification variance we can use the word-line voltage tunability feature of our design. Fig. 7(b) shows that for $V_{RWL} = 0.47$ V, $T_{PW} = 0.5$ ns word-line voltage configuration (case 2), we get an improved mean classification accuracy of 98.72% with reduced standard deviation of 0.12%. For this case, however, we need to set the $V_{REF}$ to 30 mV (near zero reference). Since, for case 2, $V_{DD} - V_{REF}$ difference is almost twice than the case 1 configuration (with $V_{REF} = 400$ mV), we need to spend twice the energy per unit computation on average than case 1. Thus, our on-silicon tunability feature enables us to perform accuracy/energy-efficiency tuning based on design requirements.

We also analyzed how the coefficient of variation ($CV$ : ratio of standard deviation and mean of a distribution) and disturb margin vary as we increase the number of IMC cells in a column. Since in modern deep networks filter dimension is usually large, we expect to have large numbers of memory rows to map such filters. From Table 2 we observe that as the number of memory rows is increased, both CV and disturb margin decrease. To explain the reduction in CV with increasing IMC cells per bit-line let’s consider the read bit-line discharge rate, $D = I/CV$, where $C$ refers to the read bit-line capacitance which is dominated by the MOM capacitor of the designed IMC-cell, $\Delta V$ refers to the RBL voltage change (i.e., the difference between the precharged voltage and the voltage at the end of the RWL pulse), and $I$ refers to the total discharge current through the readstacks. As the number of cells per bit-line increases average discharge rate remains the same as the number of cells discharging the bit-line capacitance increases proportionally. On the other hand, as we increase the number of cells per bit-line, while the variability in $C$ assumes insignificant change, assuming normal distribution, the standard deviation of $I$ follows the square root of the sum of square formula. For example, if we add $n$ number of discharging IMC-cell in a RBL column, the mean of $I$ increases by $n$ while the standard deviation of $I$ increases by $\sqrt{n}$ which in turn reduces the coefficient of variation (CV) by $\sqrt{n}$. As CV defines the variability of any

---

**TABLE 2. Impact on variation with the increase of memory row size.**

| No. of Rows | 500   | 800   | 1500  | 2500  | 5000  |
|-------------|-------|-------|-------|-------|-------|
| 1-sigma (mV)| 9.43  | 7.91  | 6.89  | 6.70  | 6.56  |
| CV (%)      | 2.50  | 1.98  | 1.81  | 1.74  | 1.73  |
| Disturb Margin (%) | 50(±4) | 50(±3) | 50(±2.5) | 50(±2) | 50(±2) |
system, with increase in cells per bit-line, the RBL discharge rate variability improves. This simple estimation of reduction of CV with increasing cells per bit-line matches very well with the simulated values in Table 2 for up to ∼1000 cells per bit-line. With very large cells per bit-line the reduction in CV saturates. Therefore, we expect to have less classification error due to process variability as larger BNN filters are mapped to our IMC array for binary convolutions.

V. RESULTS AND DISCUSSION
Performance of IMC array is commonly measured in terms of energy efficiency (TOPS/W) and throughput (GOPS). Here, we present the TOPS/W and GOPS for our MNIST network. Since, in our design, we have a maximum sub-array size of \(800 \times 100\), each column can perform 800 operations (XNOR). Our simulation shows that, during inference, 800 operations on average consume 346.5 fJ energy which corresponds to energy efficiency of 2308 TOPS/W. For each column, an additional 18.2 fJ energy is required for binary activation generation which corresponds to energy efficiency of 2193 TOPS/W. Here we report energy efficiency and throughput numbers for the product-sum-binarization operation during inference which includes the RWL driver, RBL driver and comparator. In Table 3 we compare the energy efficiency and throughput of our design for product-sum operation with recently published binary neural network accelerators that uses in-memory computing. This approach is aligned with the reporting of the energy efficiency and throughput numbers in the literature (e.g. see Valavi et al. [18]) and ensures a fair comparison. Since, inference operation does not require to operate 6T WL/BL driver, as the trained weights are already written in the memory, energy contributions from those elements are kept out of the calculation.

We get better energy efficiency primarily from two sources. First, from voltage scaling enabled by 22FDX® technology node, we get ∼1.4x energy improvement compared to state-of-the-art accelerator proposed by Valavi et al. [18]. Second, on-silicon tunability feature of the design enables us to operate at \(V_{\text{REF}} = V_{\text{DD}}/2\). Thus, on average (for a typical convolution cycle) we need to precharge \(V_{DD}/2\) voltage in RBL instead of \(V_{DD}\) which provides the rest of the energy efficiency improvement.

As far as the throughput is concerned, we perform one binary convolution and activation generation (that includes discharge, compare and precharge phase) within 3.5 ns. Since we use a maximum array size of \(800 \times 100\), we can achieve computational throughput of 22857 GOPS from our design.

TABLE 3. Summary of recently published BNN accelerators based on IMC architecture. An innovative cell design along with the use of a scaled technology node enables excellent energy efficiency and throughput in our IMC design.

|                  | Biswas [9] | Si [33] | Jiang [21] | Okumura [27] | Valavi [18] | This work |
|------------------|------------|---------|------------|--------------|-------------|-----------|
| **Target Dataset** | MNIST      | MNIST   | MNIST      | MNIST        | MNIST       | MNIST     |
| **Accuracy**     | 96.00%     | 98.74%  | 98.30%     | 99.00%       | 98.60%      | 98.75%    |
| **Technology**   | 65nm       | 65nm    | 65nm       | 12nm         | 65nm        | 22nm      |
| **Fabricated**   | Yes        | Yes     | Yes        | Yes          | Yes         | No        |
| **Operating \(V_{DD}\) (V)** | 1.2/0.9  | 0.8-1.2 | 0.6-1.0    | 0.72         | 0.94/0.88/1.2 | 0.37/0.8 |
| **Throughput (GOPS)** | 10.7        | 278.2   | 60         | 172          | 18876       | 22857     |
| **TOPS/W**       | 28.1       | 30.49-55.8 | 139-403 | 79.3         | 866         | 2308      |
TABLE 4. BNN used for CIFAR-10 Classification. First layer has input activation of size (3,32,32). No bias units for Conv. 2 - Conv. 8.

| Layer | Output Activation | Filter Size | # Parameters |
|-------|-------------------|-------------|--------------|
| Conv. 1 | (192,32,32) | (192, 3, 5) | 14,592 |
| Conv. 2 | (160,32,32) | (160, 192, 1) | 30,720 |
| Conv. 3 | (96,32,32) | (96, 160, 1) | 15,360 |
| Maxpool 1 | (96,16,16) | (3, 3) (Stride=2) | 0 |
| Conv. 4 | (192,16,16) | (192, 96, 5) | 460,800 |
| Conv. 5 | (192,16,16) | (192, 192, 1) | 36,864 |
| Conv. 6 | (192,16,16) | (192, 192, 1) | 36,864 |
| Avgpool 1 | (192,8,8) | (3, 3) (Stride=2) | 0 |
| Conv. 7 | (192,8,8) | (192, 192, 3) | 331,776 |
| Conv. 8 | (192,8,8) | (192, 192, 1) | 36,864 |
| Conv. 9 | (192,8,8) | (10, 192, 1) | 1,930 |
| Avgpool 2 | (10,1,1) | (8, 8) (Stride=1) | 0 |

However, we did not use IMC arrays with Neuron-tile architecture like Valavi et al. [18]. Therefore, with our IMC arrays even larger throughput can be obtained by adopting architecture-level innovations.

In order to assess the scalability of our design for larger networks with complex tasks, we train a binarized version of convolutional neural network of Network-In-Network architecture (NIN) [34] and mapped it into our IMC hardware for CIFAR-10 [35] dataset. We use a 9-layer convolutional network, as shown in Table 4, where we binarize all the layers except the first (Conv. 1) and last (Conv. 9) layer. We do not use any bias units in the binarized layers during training and inference. Since, the network has a maximum of 192 filters, the IMC array design requires us to consider maximum column size of 192. For this task, we partition the memory array along the columns and design the memory sub-arrays having column size of 50 and process the data in these sub-arrays independently to achieve good classification performance under circuit nonideality due to the RC delay of RWL pulses at far-end cells (as discussed in Section III.B). Moreover, the network has the largest filters in Conv. 4 layer each having 2400 binary weight values. We were able to map these filters in the IMC array having 2400 rows without needing array partitioning as our IMC design is not limited by the number of cells per bit-line. In a traditional 6T SRAM operation, larger bit-line capacitance related to the higher number of cells per bit-line provokes a read disturb. On the other hand, in our IMC design, we use a separate read-stack where we intentionally increase the capacitance of the RBL by adding the MOM capacitor. As discussed in section III.B, such design choice nullifies the impact of RBL nonidealities on classification accuracy and enables the design of large number of rows per bit-line. In our simulations for CIFAR-10 dataset, we used the maximum IMC sub-array size of 2400 × 50. For example, for Conv. 4 layer, 4 such sub-arrays are used to map the entire layer (2400 × 192) in the IMC hardware for data processing.

In order to achieve better classification accuracy for CIFAR-10, we trained the binary NIN with batch normalization layers [32]. In this case, since each neuron learns its own binary activation threshold value, we can not use a uniform threshold for all neurons. After training, we computed these threshold values from the batch normalization parameters as demonstrated by Valavi et al. [18] and used them as reference values for the comparators. With these design considerations, for CIFAR-10, we obtain classification accuracy of 85.83% from our hardware simulation which is comparable to the accuracy (86.61%) from software implementation. Moreover, for the IMC sub-array size considered for CIFAR-10, compared to MNIST, throughput will increase while energy efficiency will remain the same. Furthermore, as discussed in section IV, for larger rows or cells per bit-line, variation in RBL discharge voltage (quantified by %CV and disturb margin) reduces. Thus, for larger network, like the NIN considered here, the impact of process variability is expected to be less pronounced.

Overall, with simulation based study, we show a methodology to integrate our proposed IMC bit cell in an array considering relevant circuit nonidealities and process variations while achieving high energy efficiency and throughput. Such analysis can be useful for optimizing and enhancing the performance and robustness of existing or future IMC bit-cell array design for deep learning applications.

VI. CONCLUSION

In order to meet the growing demands for running deep learning applications at the memory and resource constrained edge devices, data processing with binary neural networks running on in-memory computing hardware platforms emerge as the most promising approach. Motivated by the quest for energy-efficient and scalable in-memory computation, in this work, we propose a novel IMC bit-cell, with 10T SRAM and a MOM capacitor, that can perform binary convolution operations when organized in a memory array. We show that, when binary layers of BNN are processed on our IMC array, classification accuracy of 98.75% with 2193 TOPS/W and 22857 GOPS can be achieved (i.e. ~2.5x TOPS/W and ~1.2x GOPS improvement compared to high performing state-of-the-art IMC solution [18]). We show that our design is robust against process variations. Moreover, our design provides a knob for improving classification accuracy under process variations trading-off energy-efficiency. Future work will be directed towards performance evaluation of our IMC arrays on larger datasets, overall system level evaluations with necessary architectural innovations and exploration of on-silicon tunability with back gate bias [36].

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Proces. Syst., 2012, pp. 1097–1105.
[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[5] M. Horowitz, “1.1 Computing’s energy problem (and what we can do about it),” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2014, pp. 10–14.

[6] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binary-}rized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1,” 2016, arXiv:1602.02830. [Online]. Available: http://arxiv.org/abs/1602.02830

[7] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “XNOR-Net: Imagenet classification using binary convolutional neural networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Oct. 2016, pp. 525–542.

[8] M. L. Schneider, C. A. Donnelly, S. E. Russek, B. Baek, M. R. Pufall, P. F. Hopkins, P. D. Dresselhaus, S. P. Benz, and W. H. Rippard, “Ultralow power artificial synapses using nanotextured magnetic Josephson junctions,” Sci. Adv., vol. 4, no. 1, Jan. 2018, Art. no. e1701329.

[9] A. Biswas and A. P. Chandrakasan, “Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2018, pp. 488–490.

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

[11] J. Zhang, Z. Wang, and N. Verma, “In-memory computation of a machine-learning classifier in a standard 6T SRAM array,” IEEE J. Solid-State Circuits, vol. 52, no. 4, pp. 915–924, Apr. 2017.

[12] W.-S. Khwa, J.-J. Chen, J.-F. Li, X. Si, E.-Y. Yang, X. Sun, R. Liu, P.-Y. Chen, Q. Li, S. Yu, and M.-F. Chang, “A 65 nm 4Kb algorithm-dependent computing-in-memory SRAM unit-macro with 2.3ns and 55-STOPSW fully parallel product-sum operation for binary DNN edge processors,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2018, pp. 496–498.

[13] S. K. Gonugondla, M. Kang, and N. Shanbhag, “A 420J/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2018, pp. 490–492.

[14] W. A. Simon, Y. M. Qureshi, M. Rios, A. Levisse, M. Zapater, and D. Atienza, “An in-cache computing architecture for edge devices,” IEEE Trans. Comput., early access, Feb. 10, 2020, doi: 10.1109/TC.2020.2972528.

[15] C. Eckert, X. Wang, J. Wang, A. Subramaniyan, R. Iyer, D. Sylvester, D. Blaauw, and R. Das, “Neural Cache: Bit-serial in-cache acceleration of deep neural networks,” in Proc. ACM/IEEE 45th Annu. Int. Symp. Comput. Archit. (ISCA), Jun. 2018, pp. 383–394.

[16] M. Kang, M.-S. Kim, N. R. Shanbhag, S. Eilert, and C. Kurewitz, “An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2014, pp. 8326–8330.

[17] R. Liu, X. Peng, X. Sun, W.-S. Khwa, X. Si, J.-J. Chen, J.-F. Li, and M.-F. Chang, “Parallelizing SRAM arrays with customized bit-cell for binary neural networks,” in Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC), Jun. 2018, pp. 1–6.

[18] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, “A 64-tile 2.4-mb in-memory-computing CNN accelerator employing charge-domain compute,” IEEE J. Solid-State Circuits, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[19] A. Agrawal, A. Jaiswal, D. Roy, B. Han, G. Srinivasan, A. Antik, and K. Roy, “Xcel-RAM: Accelerating binary neural networks in high-throughput SRAM compute arrays,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 8, pp. 3064–3076, Aug. 2019.

[20] X. Sun, S. Yin, X. Peng, R. Liu, J.-S. Seo, and S. Yu, “XNOR-RAM: A scalable and parallel resistive synaptic architecture for binary neural networks,” in Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE), Mar. 2018, pp. 1423–1428.

[21] Z. Jiang, S. Yin, M. Seok, and J.-S. Seo, “XNOR-RAM: In-memory computing SRAM macro for binary/ternary deep neural networks,” in Proc. IEEE Symp. VLSI Technol., Jun. 2018, pp. 173–174.

[22] X. Si, J.-J. Chen, Y.-N. Tu, W.-H. Huang, J.-H. Wang, Y.-C. Chiu, W.-C. Wei, S.-Y. Wu, X. Sun, R. Liu, S. Yu, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, Q. Li, and M.-F. Chang, “A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2019, pp. 396–399.

[23] J. Su et al., “A 28 nm 64 kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, 2020, pp. 240–242.

[24] Q. Dong, M. E. Sinanigil, B. Erbagci, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, “A 351TOPS/W and 372.4GOPS compute-in-memory SRAM macro in 7 nm FinFET CMOS for machine-learning applications,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2020, pp. 242–244.

[25] X. Si et al., “A 28 nm 64 kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips,” in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, 2020, pp. 246–248.

[26] K. C. Akyel, H.-P. Charles, J. Mottin, B. Giraud, G. Suraci, S. Thuries, and I.-P. Noel, “DRC2: Dynamically reconfigurable computing circuit based on memory architecture,” in Proc. IEEE Int. Rebooting Comput. (ICRC), Oct. 2016, pp. 1–8.

[27] S. Okumura, M. Yabuuchi, K. Hijioka, and K. Nose, “A ternary based bit-cell for binary neural networks,” IEEE Trans. Circuits Syst. 2, vol. 66, no. 11, pp. 4172–4185, Nov. 2019.

[28] M. Lin, Q. Chen, and S. Yan, “Network in network,” in Proc. 2nd Int. Conf. Learn. Represent. (ICLR), 2014. [Online]. Available: http://arxiv.org/abs/1312.4400

[29] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[30] R. Carter et al., “22 nm FDSOI technology for emerging mobile, Internet-of-Things, and RF applications,” in IEEE Trans. Edg. Technol., Dec. 2016, pp. 2.2.1–2.2.4.
SANJAY PARIHAR (Member, IEEE) received the B.Eng. degree in electrical engineering from McGill University, in 1987, and the Ph.D. degree in electrical engineering from Princeton University, in 1994. He joined the Advanced Products Research and Development Laboratory, Motorola, in 1995, where he carried out process integration of BiCMOS and CMOS nodes from 0.40 um to 90 nm. In 2002, he joined the Crolles 2 Alliance in France to work on 90 nm and 45 nm bulk CMOS Technologies. He returned to Freescale in 2006 and focused on sram memory specific challenges in 45 nm low power bulk technology and 45 nm high performance SOI Technology. In 2013, he joined GLOBALFOUNDRIES, Austin where he is currently a Member of the Technical Staff engaged in 12 nm node sram bitcell development. His research interests are sram bitcell design and margining.

XI CAO received the M.S. and Ph.D. degree in electrical engineering from the University of Florida, Gainesville, USA, in 2013 and 2016, respectively. During his Ph.D. career, he specialized in computational nanodevice, emerging two-dimensional semiconductor materials parameterization, and low-power transistor/memory design. He is currently working for GLOBALFOUNDRIES, focusing on SRAM targeting, cell definition, technology qualification, and IP development support across several nodes (28 nm, 22 nm, and 12 nm), which would be the mainstream for high performance or low-power/IoT applications.

JACK HIGMAN received the B.S. degree in electrical engineering from Purdue University, in 1983, and the M.S. and Ph.D. degrees in electrical engineering and computer engineering from the University of Illinois at Urbana-Champaign, in 1985 and 1988, respectively. From 1989 to 1992, he was a Visiting Assistant Research Professor with the Beckman Institute for Advanced Science and Technology, University of Illinois. From 1992 to 2012, he was at Motorola and Freescale developing embedded non-volatile and SRAM technologies. Since 2012, he has been the Senior Manager with GLOBALFOUNDRIES, Austin, Texas focused on SRAM development.

MUHAMMED AHOSAN UL KARIM received the Ph.D. degree in electrical engineering from the University of California at Berkeley, in 2016, and the B.S. and M.S. degrees in electrical engineering from the Bangladesh University of Engineering and Technology, in 2007 and 2010, respectively. He is currently working for GLOBALFOUNDRIES in the memory solution team. His research interests include emerging memories and devices, design-technology co-optimization, memory reliability, neuromorphic computing, and in-memory computing for neural networks.

* * *