3D-aCortex: an ultra-compact energy-efficient neurocomputing platform based on commercial 3D-NAND flash memories

NEUROMORPHIC
Computing and Engineering

PAPER

3D-aCortex: an ultra-compact energy-efficient neurocomputing platform based on commercial 3D-NAND flash memories

Mohammad Bavandpour, Shubham Sahay, Mohammad Reza Mahmoodi and Dmitri B Strukov*

University of California, Santa Barbara, ECE Department, United States of America
* Email: strukov@ece.ucsb.edu

Keywords: neuromorphic architecture, 3D NAND memory, energy efficiency, in memory computing, inference accelerator, analog computing

Supplementary material for this article is available online

OPEN ACCESS

RECEIVED
15 March 2021

REVISED
21 May 2021

ACCEPTED FOR PUBLICATION
2 June 2021

PUBLISHED
15 July 2021

Abstract

We first propose an ultra-compact energy-efficient time-domain vector-by-matrix multiplier (VMM) based on commercial 3D-NAND flash memory structure. The proposed 3D-VMM uses a novel resistive successive integrate and re-scaling (RSIR) scheme to eliminate the stringent requirement of a bulky load capacitor which otherwise dominates the area- and energy-landscape of the conventional time-domain VMMs. Our rigorous analysis, performed at the 55 nm technology node, shows that RSIR-3D-VMM achieves a record-breaking area efficiency of $\sim$0.02 μm^2/Byte and the energy efficiency of $\sim$6 f J/Op for a 500 $\times$ 500 4-bit VMM, representing 5$\times$ and 1.3$\times$ improvements over the previously reported 3D-VMM approach. Moreover, unlike the previous approach, the proposed VMM can be efficiently tailored to work in a smaller current output range. Our second major contribution is the development of 3D-aCortex, a multi-purpose neuromorphic inference processor that utilizes the proposed 3D-VMM block as its core processing unit. Rigorous performance modeling of the 3D-aCortex targeting several state-of-the-art neural network benchmarks has shown that it may provide a record-breaking 30.7 MB mm^{-2} storage efficiency, 113.3 TOp/J peak energy efficiency, and 10.66 TOp/s computational throughput. The system-level analysis indicates that the gain in the area-efficiency of RSIR leads to a smaller data transfer delay, which compensates for the reduction in the VMM throughput due to an increased input time window.

1. Introduction

The recent boost in the availability of labeled data and processing power has led to the development of various high-performance neural networks and their deployment in a wide range of tasks, including IoT/mobile devices [1–6]. The urgent need to realize efficient hardware neural processing platforms to accelerate these applications has resulted in the development of vector-by-matrix multiplier (VMM) circuits, which form the core operator/kernel in neural processors, and the exploration of optimal architectures for their system-level deployment. The majority of today's commercial and experimental VMM circuits and their architectural implementations in the neuroprocessors are inherently digital [7–14]. The performance of such processors on VMM-heavy benchmarks is much higher than that of the standard CPUs, in part due to the use of low-precision operations, sufficient for most neuromorphic inference tasks [15–17]. However, such digital approaches to the VMM lead to a relatively sparse design, which necessitates storing most of the synaptic weights off-chip, consequently paying a substantial performance penalty for memory access [18].

Mixed-signal VMMs in which the advanced analog-grade non-volatile memory (NVM) devices are employed as both storage elements for weight kernels and multiply-and-accumulate operators have emerged as a promising solution to outperform the digital platforms in terms of area, energy, and speed [19–32]. Indeed, prior works on NVM-based mixed-signal VMM engines have demonstrated the possibility of a rather

dramatic, orders-of-magnitude advantages in energy, speed, throughput, and circuit density, over their digital counterparts [18, 19, 26, 28]. However, the device-to-device variations and the large cell current of the ReRAM devices [19–22], or the relatively large footprint of the floating-gate memory cells [25, 33] hinder the realization of a scalable system-level design.

The 3D-NAND flash memory technology, while still advancing rapidly, exhibits a record-breaking effective bit density, an ultra-low fabrication cost per bit, and multi-level cell programming capability [34–39]. The time-domain VMM approach (which outperforms the current-mode VMM approach [27, 40–42] in terms of area- and energy-efficiency) is inherently compatible with the complex 3D-NAND flash memory architecture without their modification. We recently proposed a compact time-domain VMM accelerator based on 3D-NAND flash memory [43]. Unlike the previously proposed current-mode approach, this approach does not require a major technological effort in redesigning the 3D wiring of the highly optimized 3D-NAND memory matrix.

One common issue in 3D-NAND memory is large parasitic capacitances, which requires a large load capacitor to minimize the coupling error in the charge-based time-domain VMM implementation. Large load capacitor significantly degrades their area/energy efficiency [43]. Furthermore, the VMM operations with relatively small output range are desired for many DNN/RNN models to minimize output quantization error. Designing a VMM for such output ranges while maintaining an appropriate computational precision becomes extremely challenging in the presence of large parasitic capacitances since the load capacitor cannot be arbitrarily shrunk due to the significant capacitive coupling.

In this paper, we address these challenges by proposing novel VMM circuits. We further use such circuits to develop inference processor based on 3D-NAND memory technology. The specific major contributions are:

- A novel compact resistive successive integrate and re-scaling (RSIR) approach that allows for efficient implementation of time-domain VMM, including VMMs with sub-maximal output ranges.
- 3D-VMM design employing RSIR approach, and its detailed modeling, considering shot/thermal noise, DIBL, capacitive coupling, and process variations in CMOS circuitry. The proposed VMM design is suitable for a native 3D-NAND memory in that it does not require modification of its highly optimized memory matrix.
- The design of 3D-aCortex, a multi-purpose mixed-signal neuromorphic inference accelerator based on 3D-NAND memory, and detailed performance modeling for the common neural network benchmarks. The modeling results shows superior storage efficiency at slightly worse energy-efficiency and throughput compared to the mixed-signal 2D counterparts (table 1).

2. RSIR time-domain 3D-VMM

Optimizing the output quantization range is a crucial step in low-precision mixed-signal VMM design for neural applications. Such optimization maximizes the information content in the output of the analog-to-digital converters and reduces the impact of quantization on the network's functional performance. The output distribution in a particular layer of the neural network is often limited to a sub-maximal range, well below $y_{max} = K \times w_{max} \times x_{max}$ where K, w_{max}, and x_{max} are the number of inputs, the maximum weight, and the maximum input, respectively [46]. Tailoring the charge-based VMM with 3D-NAND flash [43] (supplementary section I, available online at https://stacks.iop.org/NCE/1/014001/mmedia) for such sub-maximal output range while maintaining the original computing precision requires increasing the input/output time window, T. However, this leads to a reduction in the throughput of the VMM. Other probable solutions for designing the VMM with sub-maximal output range is to decrease drain voltage swing (ΔV_D) or shrink the load capacitor. Both result in a degradation of the computational precision due to variation/noise and capacitive coupling, respectively [43]. Therefore, an alternative approach that facilitates the VMM operation with sub-maximal output cell current range without significant degradation in the throughput is necessary. Also, the load capacitor typically dominates the footprint of the conventional charge-based time-domain VMM approach and hence needs to be minimized.

To this end, inspired by our recent work [55], we propose a novel resistive successive integrate and re-scaling (RSIR) based time-domain VMM scheme with resistive load. It is based on an alternate representation of the dot-product in the digital domain. In the digital domain, the dot-product operation can be represented as:

$$y_j = \sum_{p=0}^{P-1} 2^p \sum_{i=1}^{K} x_i(p) w_{ij}, \tag{1}$$

where $x_i(p)$ is the pth bit (with $p = 0$ corresponding to the LSB) of the ith digital input, and P is the input bit-precision. A scaled version of this dot-product, namely $2^{-p} y_j$, can be obtained in an iterative manner as:

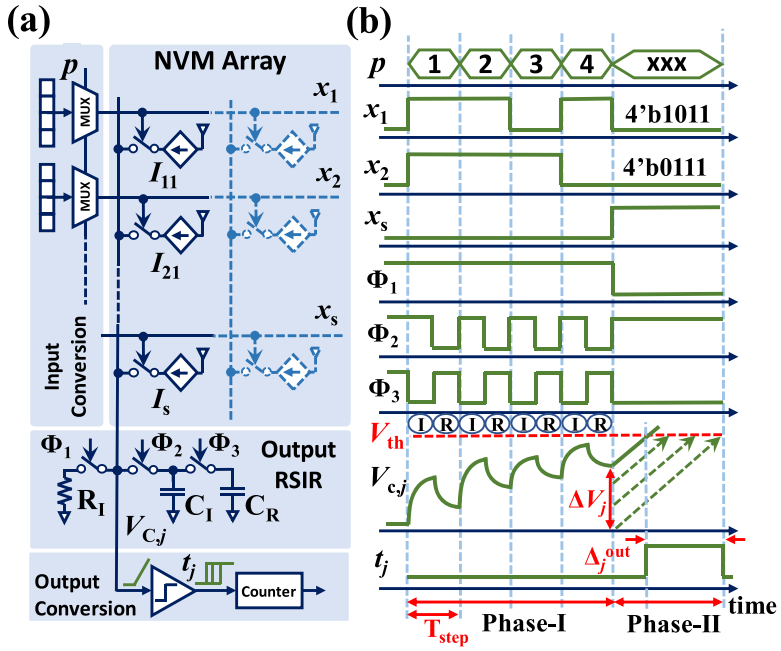

Figure 1. (a) Basic structure of the resistive successive integrate and re-scaling (RSIR) time-domain approach, and (b) illustration of the timing diagram of its operation, assuming 4-bit input conversion precision. For clarity, the figure shows operation when memory cells are sourcing current. In 3D-NAND memories, the memory cells would typically sink the current. In this case, the output circuitry is connected to the bit-line side of the memory cell string. The output would be pre-charged to a certain voltage, similar to the scheme discussed in reference [27], and then discharged during RSIR operation.

$$y_j^{(p)} = \sum_{i=1}^{K} x_i(p) w_{ij} + \frac{1}{2} y_j^{(p-1)} \tag{2}$$

where $y_j^{(-1)}$ is initialized to zero, and subsequent intermediate results ($y_j^{(p)}$) are successively computed by adding the weighted sum of the pth bits of input vector elements $x_i(p)$, starting from the least significant bit, to half of the previous intermediate result $y_j^{(p-1)}$. After $P-1$ such successive dot-product calculation (integration') and re-scaling (division') operations, we get $y_j^{(p-1)} = 2^{-p} y_j$, which forms the basis for the proposed RSIR scheme.

The circuit schematic and the timing diagram of the RSIR VMM, which implements equation (2), are shown in figure 1. The input bits are applied to the VMM one at a time, so phase I includes P steps for P-bit input precision. Each step is completed by performing two operations: (a) integrate followed by (b) re-scaling, which are clocked by two non-overlapping control signals Φ_2 and Φ_3. In the p'th step, the p'th bit of all the inputs, starting from LSB, are selected using multiplexers and applied to the input lines. The application of the inputs activates the adjustable current sources with current values proportional to the weights. At the end of integrate operation, the voltage over the resistor R_I, and on the capacitor C_I, is equal to $R_I \sum_{i=1}^{K} x_i(p) I_{ij}$, i.e. proportional to the weighted sum of the pth bits of input vector elements. In the re-scaling operation, C_I is disconnected from R_I and weight current sources, and then connected to an identical re-scaling capacitor C_R C_I. Charge sharing between C_I and C_R results in a re-scaling of the intermediate results. Finally, the voltage on both C_I and C_R at the end of p'th step is:

$$V_{c,j}^{(p)} = \frac{1}{2} R_I \sum_{i=1}^{K} x_i(p) I_{ij} + \frac{1}{2} V_{c,j}^{(p-1)}, \tag{3}$$

which is equivalent to equation (2) when using $w_{ij} = \frac{1}{2} R_I I_{ij}$ and $y_j^{(p)} = V_{c,j}^{(p)}$. After P integrate and re-scaling operations during phase I, the voltage on C_I and C_R is proportional to the re-scaled version of the final dot-product value. This voltage is then converted to a digital pulse in phase II of the RSIR scheme, similar to the charge-based VMM scheme [43] (see supplementary section I for more details). For this conversion, the integrate resistor is disconnected, and the capacitors are further charged using a sweeping current source. Once the voltage at the capacitor exceeds the neuron's threshold voltage (V_{th}), a digital pulse with a duration proportional to the weighted summation of the inputs is generated at the output of the neuron circuitry. Accordingly, the total VMM operation time can be formulated as $T_{VMM} = T_{WL} + P \times T_{step} + T_{out}$, where T_{WL} is the time

needed to bias the WL of the selected layer to 'read' voltage, T_{step} is the time spent in I and R steps, and T_{out} is the time spent in the final output sweep.

The RSIR approach can be implemented on the commercial 3D-NAND memory blocks without the need for any modification in the memory structure and wiring. This approach utilizes similar peripheral circuit connectivity as compared to the charge-based time-domain scheme [43] see figure S2(a) (supplementary section I). The weights are realized by tuning the threshold voltage (which dictates the string current) of the cells in different strings of a layer in the 3D-NAND flash memory. Also, since the 3D-NAND flash strings sink currents (rather than sourcing), the reference voltage, i.e. initial voltage of the resistor/capacitors in figure 1(a), is set to $V_{reset} = V_{th} + \Delta V_D$.

A major advantage of this resistive load based scheme is that the charges injected to the integrate capacitor due to the parasitic capacitive coupling from input strings are discharged through the load resistor over the transient period. Therefore, the disturbance charges during the switching transients are eliminated and do not impact the voltage on the integrate capacitor. This alleviates the stringent need for large output capacitors to maintain computing accuracy in time-domain VMM designs based on 3D-NAND flash memory. As a result, in the proposed RSIR scheme, the value of the output capacitors (C_I and C_R) are independent of the parasitic capacitance components of the 3D-NAND string and determined only by the coupling capacitors of the pass-transistor based switches, which are significantly smaller.

Also, a smaller load capacitor demands a lower sweeping current during phase II of VMM operation. This reduced sweeping current can be provided by encoding cells of some rows within the same WL layer to the maximum string current value and enabling them by applying high input to their BSL only in phase II of computation. Such an option is not available in the charge-based approach in which the top layer memory cells need to be selected to provide the relatively large sweeping currents. Therefore, the energy/delay overhead of switching the WL layers for phase II of the VMM operation to provide the sweeping current is also eliminated when utilizing the RSIR approach.

The other advantage of RSIR approach is its suitability for VMM operation with sub-maximal output range that is essential for reducing quantization error. Indeed, the value of R_I is determined by the range of the string current and the target voltage swing on the drain (ΔV_D). Hence, VMM with smaller output ranges can be implemented by simply increasing R_I value.

To cover a broad spectrum of practical VMM output ranges, we consider three scenarios. In the first, full-range (FR) scenario, the VMM output is the top P most significant bits of the maximal output range. In other words, assuming that the hypothetical, full precision binary representation of VMM's maximal output range is $Y_P\, Y_{P-1} \ldots Y_0$ with P P, the output in FP scenario are Y_P to Y_{P-P} bits. In the other two studied scenarios, more relevant for state-of-the-art deep-learning classifiers, sub-maximal output range are considered in which the P output bits are extracted from lower half bits ($Y_{P/2}$ to $Y_{P/2-P}$) and lower third bits ($Y_{P/3}$ to $Y_{P/3-P}$) of the virtual high-precision full range. For a K-input single-quadrant VMM operation with weights/inputs scaled to [0, 1], the sub-maximal scenarios are roughly equivalent to the analog ranges of $^2\overline{K}$ and $^3\overline{K}$, which are called 'sq2' and 'sq3' scenarios, respectively, in the rest of the manuscript.

We performed a detailed VMM-level analysis considering the behavioral compact model for 3D-NAND memory based on polysilicon gate-all-around macaroni-body charge-trap cells [38] see supplementary section II for more details on the cell modeling and parameter tuning. Moreover, the line resistance and parasitic capacitances of the WL metal plates and BL/BSL lines were also considered in this work. (Level shifters are resized to keep this T_{WL} in the range of [20 ns, 30 ns].) The end-to-end VMM-level circuit simulations are performed using 55 nm GlobalFoundries PDK considering device/line parasitics, variations, and non-idealities. Figure 2 shows the computational error, calculated as the maximum difference between the theoretically computed output times utilizing ideal current sinks and the simulated output times, as a function of the linear size of square-shaped $K \times K$ VMM. The results were obtained from the VMM-level simulations over multiple VMM-level runs, performed for different inputs and weights in each run to span the entire set of weights and inputs, for sq2 and sq3 submaximal output ranges. To achieve a semi-optimal design point targeting a particular VMM precision, each scenario is explored for two different output voltage swings of $\Delta V_D = 0.2$ V, and 0.3 V and four different durations of the time-step for phase I of the RSIR VMM operation: $T_{step} = 10$ ns, 20 ns, 40 ns, and 80 ns. Shrinking VMM output range from sq2 to sq3, R_I has to be increased to capture the output current range. This results in a larger RC-transient, and consequently, leads to a reduction in the computational precision, which can be compensated by increasing T_{step}. However, the reduction in VMM throughput due to an increase in the T_{step} for sub-maximal VMM output ranges is significantly smaller for RSIR-3D-VMM compared to its charge-based counterpart. For instance, assuming fixed load capacitor and voltage swing, an input time window of 16×100 ns $= 1.6$ μs is required for 1000×1000 VMM with an output range of $^3\overline{1000} = 10$ (sq3 scenario) to maintain 4-bit computing precision [43], while the RSIR-3D-VMM can achieve such precision with the input time window of 4×80 ns $= 320$ ns (figure 1(b)). Furthermore, 4-bit computational

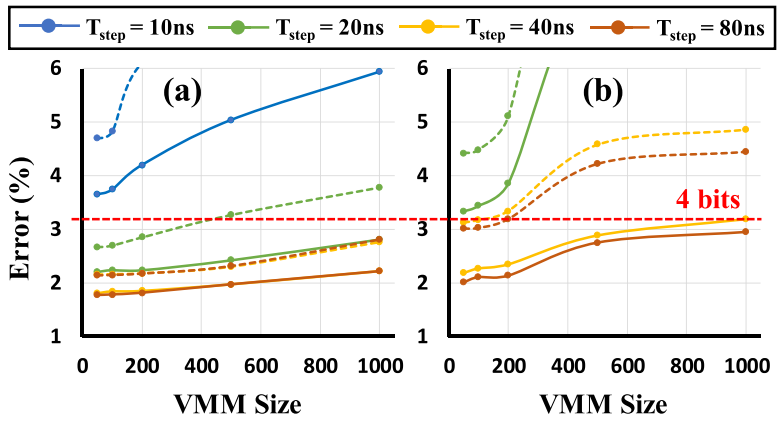

Figure 2. RSIR-3D-VMM computing error as a function of VMM size (K) for (a) sq2, and (b) sq3 sub-maximal output range scenarios targeting 55 nm technology node ($C_I = C_R = 20$ fF). Each graph shows the error for two different drain voltage swings of $V_D = 0.2$ V, and 0.3 V (solid and dotted lines, respectively), and also four different step duration of $T_{step} = 10$ ns, 20 ns, 40 ns, and 80 ns (blue, green, yellow, and brown, respectively). Each data point represents 3 distribution of VMM error, defined as $100 \times |T_{VMM\ ideal} - T_{VMM}|/(T_{out})_{max}$, where $(T_{out})_{max} = 2^P \times T_{step}$, considering device/interconnect variations and randomized inputs/weights.

precision can be achieved in the RSIR VMM for $V_D = 0.2$ V and $T_{step} = 40$ and 80 ns for sq2 to sq3 cases, respectively.

The energy, area, and throughput of the RSIR-3D-VMM for sq2 and sq3 ranges are compared against the charge-based 3D-VMM with FR output [43] in figures 3(a)–(c), respectively. A computational precision of 4-bits is assumed for these estimates, which is sufficient for most of the tasks involving neuromorphic computations [15–17]. As can be observed from figure 3, the RSIR-3D-VMM achieves a 30% improvement in energy efficiency (∼6 fJ/Op) as compared to the charge-based time-domain 3D-VMM [43]. This gain in energy efficiency is attributed to the elimination of the energy consumed while using level-shifters to select the top layer for sweeping currents during phase-II of the VMM operation. More importantly, the RSIR-3D-VMM achieves a record-breaking area efficiency of ∼0.02 μm^2/Byte, which is 5× improvement over the charge-based 3D-VMM, since the size of the integrate capacitor can be minimized without increasing the disturbance charge error, i.e. degrading the computational precision, in the RSIR-3D-VMM, unlike the charge-based VMM. Such high area efficiency of the RSIR-3D-VMM enables efficient system-level deployment due to a significant reduction in the data transfer delay/energy overhead see the next section.

3. 3D-aCortex

3.1. Top-level architecture

The architecture of the proposed 3D-aCortex is derived from that of the 2D-aCortex [18] (supplementary section III), using the general transformation scheme shown in figure 4(a). As shown in figure S2, the 2D-aCortex is equivalent to a very large VMM operator in which the digital inputs are read into the buffer blocks (shown black), which can be configured as shift registers to minimize the need in the main memory (MM) access at convolution tasks. The inputs are converted into analog/time-domain signals and propagated to vertical input lines of the 2D NVM array, while analog output signals, aggregated on the shared output lines of the array, are converted back to digital values and stored in local buffers (shown green), waiting to be written into the MM.

The 3D equivalent of such a 2D-VMM operator is shown on the right panel of figure 4(a), which assumes a multi-step (here 4-step) VMM operation, at which each weight sub-matrix is selected in one step. To avoid an increase in the number of MM accesses per VMM operation during the 2D-to-3D transformation, the input/output blocks are redesigned. At the input, the shift registers are folded, and an extra selector is added to discriminate between vertically aligned buffer blocks at various VMM steps. Moreover, at the output, a digital accumulator with extra precision is added to temporally aggregate the partial results. Finally, the most significant part of the result is selected for the final output, using a barrel shifter to match the target precision.

Following such a transformation scheme, we have proposed the 3D-NAND-based DNN/RNN processor architecture shown in figure 4(b). Its main components are:

PE: in this architecture, PEs are placed as an $M \times 2N$ 2D structure where they share time-domain inputs in the vertical direction (I-bus), and analog BL output in the horizontal direction (O-bus). Each PE includes a 64-layer $K \times 2K$ matrix of 3D-NAND memory cells and peripheral circuits, which is required to implement a differential-weight $64 \times K \times K$ 3D-NAND VMM circuit (figure 5(a)). The peripheral circuitry for each

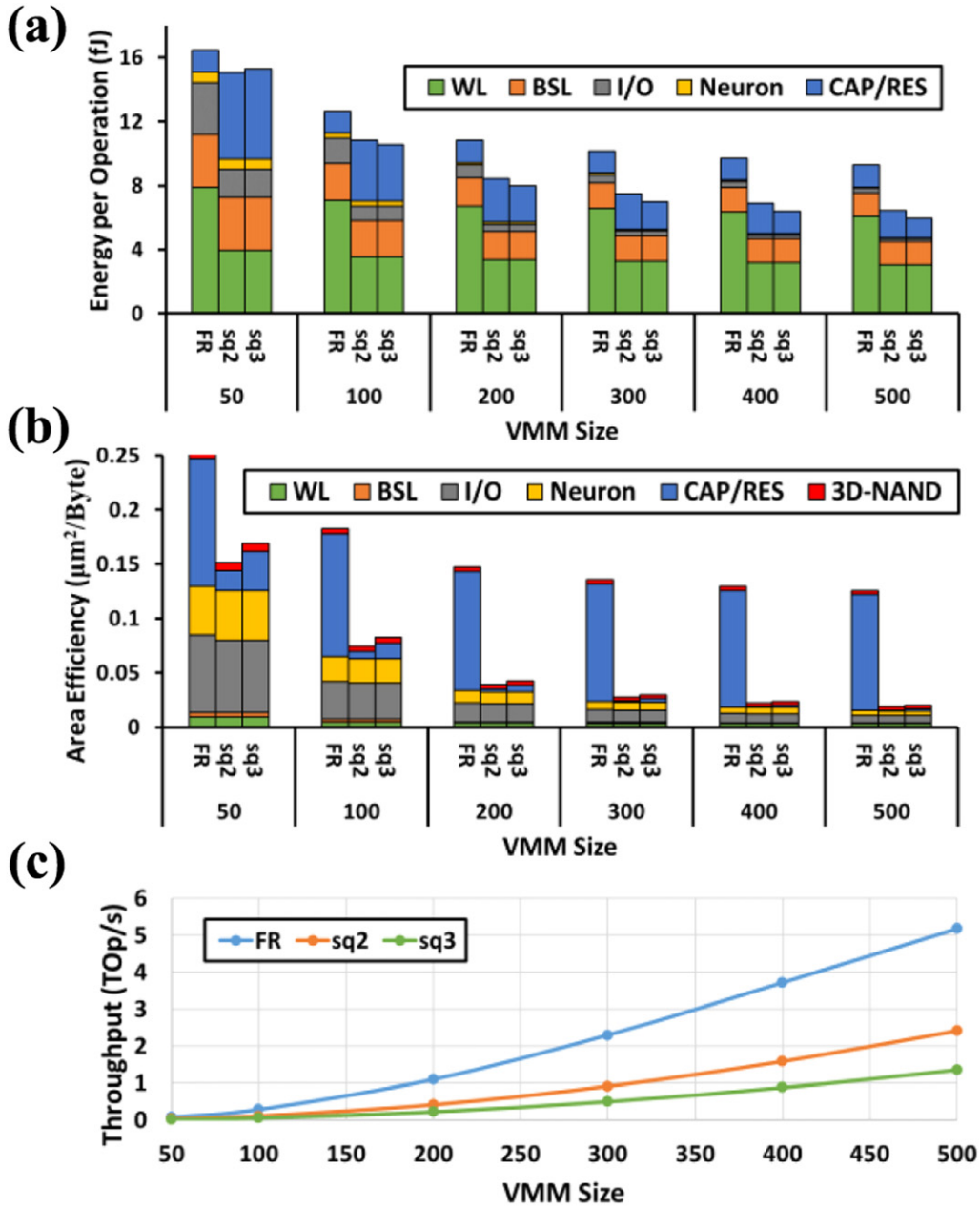

Figure 3. Performance metrics and breakdown of 3D-NAND based RSIR VMM approach as a function of VMM size (K) and their comparison to charge-based VMM approach in terms of (a) energy efficiency, (b) area efficiency and (c) throughput. FR indicates full output range for which the charge-based VMM approach is used while RSIR approach is used for sq2/sq3 output range scenarios. I/O includes both digital-to-time and time-to-digital converters. Neuron includes the threshold circuitry and the output switch-cap circuitry for RSIR scheme. CAP/RES component indicates either the load capacitors in the charge-based scheme or the load resistor in RSIR.

PE includes: (1) 64 word level level-shifters ($\overline{\mathbf{WL}}$) and drivers for selecting the target layer; (2) K bit select line level-shifters ($\overline{\mathbf{BSL}}$) and drivers for changing the voltage level of the shared time-domain input, and also activating the inputs during phase II of computation; (3) control logic gates for enabling/disabling the unit components. In the case of the charge-based approach, each PE also includes K local load capacitors (**CAP**) connected to the shared BLs and also V_{reset} through pass gates. Since the VMM area is largely dominated by load capacitors in the charge-based approach (figure 3(b)), we have also investigated a variation of the baseline charge-based 3D-aCortex architecture in which a CAP unit is shared among 16 3D-NAND blocks, each with its own BSL and WL level-shifter logic circuits (figure 5(b)). The column select (CS) and row select (RS) lines are propagated respectively in vertical and horizontal directions to select and enable the target PEs. Moreover, the **CAP** pass-gates in the enabled PEs are set to VMM operation mode at the appropriate time through a control signal called VMM_OP.

Integrate-digitalize unit (IDU): each IDU block includes three subblocks: (1) neuron latches receiving input from O-bus; (2) time-to-digital converters (TDCs) which are digital accumulators with higher precision (here 6-bit where two extra bits enables accumulating results for VMM operation on four-layers, i.e.,

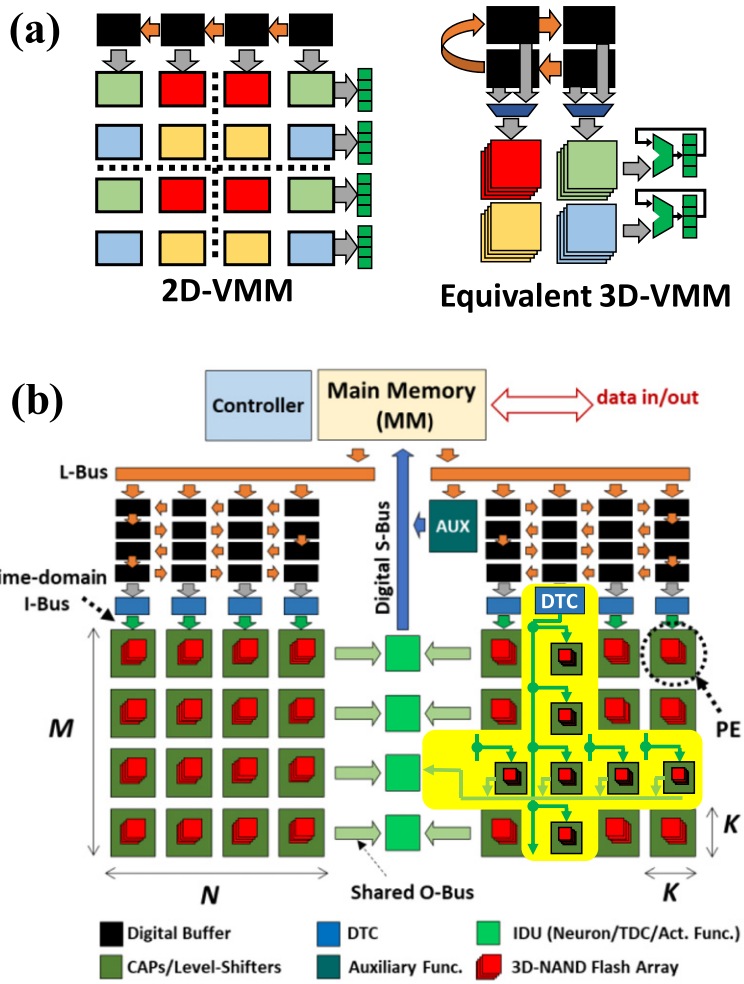

Figure 4. (a) 2D to 3D architecture transformation. (b) 3D-aCortex architecture layout with its main parameters and blocks.

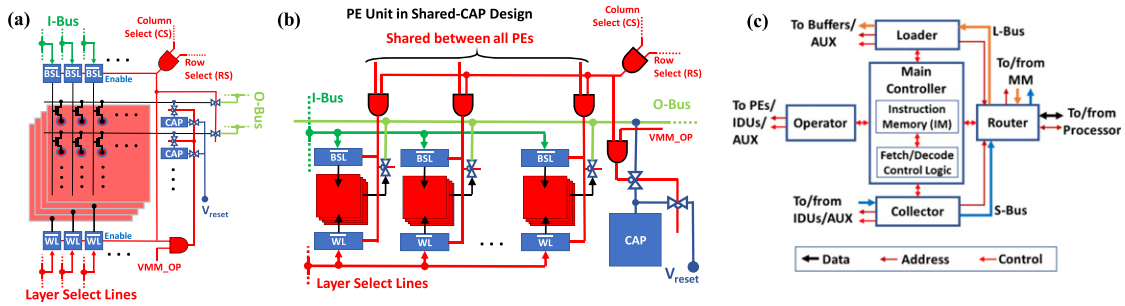

Figure 5. (a) PE's main circuit components and control circuitry, and (b) main changes in PE design with respect to the baseline architecture in order to share the area-hungry capacitors in charge-based approach among multiple 3D-NAND memory blocks. (c) Controller sub-blocks and signaling between them.

$4 \times 2N \times K$ inputs without overflow); (3) barrel shifters to select the target output bit locations; and (4) activation function circuitry which applies a target nonlinear function (here linear, ReLU, tanh, or sigmoid) to the TDC's output. In the RSIR approach, this unit also includes the load resistor, integrate and re-scaling capacitors, and control switches.

Controller: due to the flexibility of 3D-aCortex, any VMM operation up to $MK \times NK$ can be performed in one VMM step. The following actions are performed for VMM computation on one layer of the 3D-NAND memory: (1) target PEs and their corresponding digital-to-time converter (DTC) and IDU units are enabled; (2) input data are loaded into the buffers and, simultaneously, target 3D-NAND memory layer is selected; (3) enabled PEs are set to VMM operation mode and DTCs convert and apply time-domain inputs during phase I

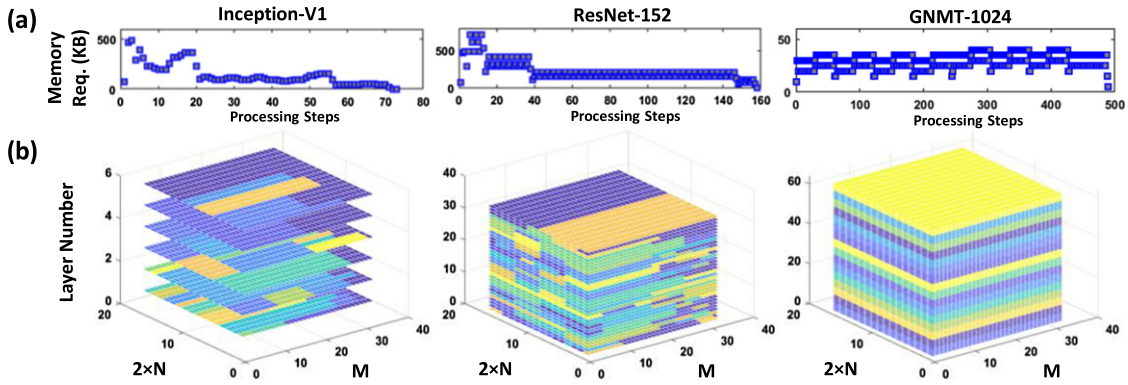

Figure 6. (a) Memory requirement for various processing steps of a single inference task, and (b) mapping of the network weight kernels into the 3D-NAND memory block (different colors represent different kernels, and dark blue represents not utilized locations) for Inception-v1 [4], ResNet-152 [5], and GNMT-1024 [6].

of operation; (4) sweeping (first) layer in 3D-NAND memory is selected, (5) outputs are calculated, converted back to digital; and finally (6) the results pass through the target activation function and stored in the MM.

To control and synchronize various units while taking advantage of the eligible time-overlap between consecutive VMM operations as well as steps of one VMM operation, a multi sub-unit controller is proposed as shown in figure 5(c). In this controller, various duties are delegated to different sub-units. Router handles the data transfer between MM and collector/loader/processor port. Loader performs single/burst read from MM into auxiliary block (AUX) or buffers with various buffer/MM address strides. Similarly, collector performs single/burst write from IDUs/AUX into MM with various buffer/MM address strides. The control signals for synchronizing various steps of VMM operation such as selecting 3D-NAND memory layers, applying inputs, performing VMM operation, and calculating outputs are issued by operator. Finally, the main controller configures the rest of the sub-units and synchronize their operations. Such controller design enables reading the data for the next VMM operation, while writing back the output results from the previous one. It also eliminates heavily nested loops (e.g. convolution tasks) in the machine code through implementing the most frequently used loops in hardware.

3.2. Network mapping

The goal of network mapping is to break down inference computation into a sequence of steps (instructions) and to determine optimal locations for storing kernel weights in VMM arrays and temporary results in the main memory. (The mapping process was also crucial for fine-tuning architectural parameters, e.g., understanding minimal requirements for main memory capacity).

The network mapping algorithm is based on the approach proposed in the context of 2D-aCortex architecture [18]. Specifically, the neural network is first converted into a computational graph in which each node represents one (convolution, fully-connected, max-pooling, etc) network layer, while each edge represents the amount of data that has to be transferred from one node (layer) to others. The layers are processed sequentially as a sequence of processing steps', and we assume that all input and output data of the currently processed layer must be stored in memory. With such a scheme, the total amount of main memory that will be occupied after each processing step can be calculated by counting the edges in the computational graph, which are cut by a line separating all already processed nodes from yet-to-be processed ones. Figure 6(a) shows the memory requirement graph extracted from such assessment performed for the studied networks.

The weight matrices are mapped into the 3D structure of memory blocks using a weight placement scheme, including three steps namely quantization, reshaping, and 3D packing. According to this scheme, first, the weight kernel dimensions, i.e., number of inputs and outputs, are quantized by K. In convolution operation, the quantization, reshaping, and packing are performed in such a way that the shift operation in hardware is equivalent to the shift in convolution.

In the second step, the quantized weight matrix dimensions are compared to the maximum dimensions of one-step VMM in the hardware, i.e., $2N \times M$. If the kernel dimensions exceed the maximum allowable 2D-VMM in hardware in any dimension, the weight matrix is broken in that dimension and reshaped to a 3D matrix in such a way that the third dimension, which is equivalent to the memory layer in hardware, indicates different weight sub-matrices (either in a row-first or column-first manner).

In a third step, weight kernels are mapped into specific locations in a 3D memory array using a heuristic algorithm whose goal is to minimize the number of utilized memory cell layers. Specifically, one iteration of the algorithm involves the generation of a randomly ordered list of kernels and then the sequential mapping

Table 1. Performance comparison of the various versions of 3D-aCortex to the state-of-the-art digital and mixed-signal neuromorphic processor architectures. Except for TPU and UNPU, all performance results are based on simulations. *Estimated, highly optimistic performance for 4-bit computing precision and 55-nm technology node implementation. Note that 4-bit weight/activation quantization results in negligible decrease in functional performance (and actually better performance for ResNet) [51,52]. # The performance numbers do not include overhead of external memory access (weights/intermediate data). & Baseline charge-based / 16x cap sharing charge-based / RSIR-sq2 / RSIR-sq3 architectures.

Platform	DaDianNao [8]	TPU [7]#	[14]#	ISAAC [44]	PUMA [45]	2D-aCortex [18]	3D-aCortex&
Technology node	28 nm	28 nm	65 nm	32nm	32 nm	55 nm	55 nm
Approach	digital	digital	digital	ReRAM	ReRAM	2D-NOR	3D-NAND
Clock (MHz)	606	700	200	1200	1000	700	1000
Precision (bits)	16 fixed point	8 fixed point	1–16 (4 here)	16 fixed point	16 fixed point	4 fixed point	4 fixed point
Area (mm^2)	88	330	16	85.4	90.6	292.9	18.43/41.7/0.079/0.056
Power (W)	20.1	40	297	65.8	62.5	0.039	0.151/0.126/0.079/0.056
Throughput (TOp/s)	5.54	92	1.38	39.9	52.31	14.97	10.66/8.2/8.63/6.34
CE (TOp/s-mm^2)	0.063	0.28	0.086	0.46 (0.62*)	0.58 (0.78*)	0.051	0.58/0.2/0.96/0.7
SE (MB/mm^2)	0.2	off-chip	off-ship	0.74 (0.25*)	0.76 (0.257*)	0.273	4.34/30.7/8.92/8.9
EE (TOp/J)	0.286	0.43	11.6	0.35 (5.14*)	0.84 (12.09*)	380.25	70.43/65/108.7/113.3

of kernels from the list by greedily searching for the locations within already occupied memory layers and only allocating new layers if no such location is found. The best solution is then chosen among several iterations of the algorithm. The output results of such an algorithm are shown in figure 6(b) for the three studied networks.

4. System-level performance

To evaluate the system-level performance for target DNN/RNN networks running on the 3D-aCortex, we have developed a software framework that utilizes the post-layout energy/speed/area metrics of all its blocks (buffers, buses, DTCs, TDCs, neurons, and digital circuits) in the 55 nm technology node. (The energy/throughput/area numbers for the SRAM-based instruction memory and the eDRAM-based main memory are obtained using the CACTI memory estimator [49]). This framework extracts the list of processing tasks for a given network, maps the VMM kernels on the 3D array of memory devices, and provides estimates for the energy/throughput of the inference operation along with the area of the processor for the given set of architecture specifications.

Two DNN networks, Inception-v1 [4] and ResNet [5], with different computational graphs and network sizes, and also Google's natural machine translation (GNMT), a very common RNN network [6], have been selected as the benchmarks for the evaluation of the proposed general-purpose architecture. The evaluation was performed for 3D-aCortex with 4-bit computing (activation) precision, which has been shown to be adequate for the studied networks. For example, [15–17] reported a negligible drop in functional performance compared to the full precision one for precisely the same version of ResNet, which was studied in our work, a larger version of Inception, and, similar to GNMT, LSTM-based recurrent networks.

Furthermore, we have performed a preliminary exploration of architectural parameters to optimize the processor's performance. As a result, the value $K = 64$ was chosen, which is an optimal point balancing the trade-off between the computational block utilization and the data transfer parallelism. Moreover, partitioning the memory blocks into such a small array size as compared to the large block size of the commercial 3D-NAND flash results in a significantly low layer selection (access) time owing to the considerable reduction in the parasitic capacitance of the WL/BSL plates/wires and dedicated level-shifter driving each individual partition which further reduces the time for charging/discharging these plates/wires. However, the significant improvement in the computation throughput and energy (due to higher utilization) is achieved at the cost of an increased area overhead of the peripheral circuitry. Note that in the commercial 3D-NAND flash blocks, memory density is the primary focus of the design, so a single peripheral circuitry drives a vast, $\sim$1000$\times$ larger 3D-NAND block resulting in a considerable access time of $\sim$7 μs [56, 57].

Moreover, the parameters M and N were selected to balance the read and write time/energy. Note that the parasitic of the shared bit lines (O-BUS) bounds the horizontal dimension of the processor and consequently affects the PE's aspect ratio as well as the number ($2N$) of these elements sharing one bit line. A detailed study of the benchmark networks has shown that a 1 MB MM is sufficient to store all intermediate data, while the flow control program requires at most 4 KB IM. Finally, our detailed analysis indicates that $M = 32$ and $N = 8$ satisfy the aforementioned conditions while being sufficient to perform even the largest, 128M-weight GNMT benchmark.

The architecture specifications, performance measures, and their breakdown are summarized in table 2 for the baseline charge-based 3D-aCortex, area-optimized charge-based approach with 16$\times$ CAP sharing, and RSIR-based 3D-aCortex with sq2 and sq3 output range scenarios. As shown, RSIR approaches achieve a peak

Table 2. System-level performance results for four different versions of 3D-aCortex designed in 55 nm process: (1) baseline charge-based 3D-aCortex, (2) area-optimized charge-based approach with 16× CAP sharing, and (3) and (4) RSIR-based 3D-aCortex with sq2 and sq3 output range scenarios, respectively. The benchmarks include inception-v1 [4], ResNet-152 [5], as well as Google's neural machine translation recurrent network, GNMT-1024 [6] with the number of parameters/operations equal to 7.2 M/5.2 B, 55 M/20 B, and 0.13 B/2.6 B, respectively. The semi-optimal architectural parameters for the reported results are $K = 64$, $M = 32$ and $N = 8$.

Approach	Baseline charge-based			16× CS charge-based			RSIR sq2			RSIR sq3		
Benchmark	GNMT	INC-V1	ResNet	GNMT	INC-V1	ResNet	GNMT	INC-V1	ResNet	GNMT	INC-V1	ResNet
					Area breakdown (%)							
3D-NAND	2.95			20.86			6.06			6.04		
MM	34.83			15.39			71.59			71.32		
CAP/RES	52.70			23.29			0.49			0.87		
DTC/TDC/N	1.72			0.76			3.48			3.47		
Level shifters	4.71			33.3			9.88			9.65		
Others	3.09			6.4			8.5			8.65		
					Energy breakdown (%)							
WL	58.6	38.6	41.7	54.1	27.8	31.8	45.2	27.2	28.2	47.1	27	28.1
MM	2.3	10.3	9.7	2.1	7.4	7.4	3.5	14.5	13.2	3.7	14.4	13.1
CAP/RES	10.6	7	7.5	9.8	5	5.7	10.1	7.8	10.9	5.9	4.8	7.6
DTC/TDC/N	1.6	3	4.5	1.4	2.3	3.5	1.6	2.1	3.6	1.5	2	3.5
BSL	22.4	14.7	16	20.67	10.6	12.2	35	20.8	21.6	36	20.6	21.5
Buses	3.9	22.2	17	8.7	37	28.5	3.7	18.8	15.2	3.9	18.7	15.1
Leakage	0.6	2.6	2.5	3.13	8.7	10	1.1	6.3	5.9	1.5	10.1	9.4
Others	1<	1.6	1.1	1<	1.2	1<	1<	2.5	1.4	1<	2.4	1.7
					Performance summary							
Area (mm^2)	18.43			41.7			8.96			9		
# Occupied layers	64	6	33	64	6	33	64	6	33	64	6	33
SE (MB mm^{-2})	4.34			30.7			8.92			8.9		
Power (mW)	151.35	33.27	33.64	126.1	45.64	39.46	79.34	13.22	14.32	55.9	8.34	8.97
Latency (ms)	0.23	5.211	12.61	0.29	5.27	14.1	0.28	9.28	21.9	0.38	14.84	35.05
EE (TOp/J)	70.43	27.42	44.52	65	19.76	33.94	108.72	38.7	60.19	113.33	38.38	60.02
Throughput (TOp/s)	10.66	0.91	1.49	8.2	0.9	1.34	8.63	0.51	0.86	6.34	0.32	0.54
Energy/frame (J)		1.7×10^{-4}	4.2×10^{-4}		2.4×10^{-4}	5.6×10^{-4}		1.2×10^{-4}	3.1×10^{-4}		1.2×10^{-4}	3.1×10^{-4}

area and energy efficiency of 8.88 MB mm^{-2} and ∼110 TOp/J, which is 2× and 1.57× higher than their baseline charge-based counterpart. Moreover, the natural drop in throughput with respect to the output range has been effectively compensated at the system level with the throughput only dropping from 10.6 TOp/s to 6.3 TOp/s despite the drastic reduction in the output range in sq3 scenarios, e.g., a factor of 0.01× for a 1000-input neural layer.

5. Comparison with prior work

On the circuit level, to the best of our knowledge, 3D-NAND/AND-based VMMs have been studied in only three works [39, 43, 58]. In [39], inputs are applied to WL terminals which require partitioning them to several independent WLs per layer along x-direction, or using cells which do not share the same WL by applying input to WLs in different blocks and then performing a sparse current-mode VMM operation and also adding extra switching circuitry to enable individual access to multiple WLs in one layer. This approach also suffers from the challenging problem of managing a large number of word lines, which would likely result in a very large peripheral circuitry overhead. Moreover, such a sparse VMM operation would be highly inefficient as it does not exploit the inherent ultrahigh-density feature of the 3D NAND flash memory. In addition, in this scheme based on the current-mode VMM, analog input signals are applied to highly resistive and capacitive word lines, leading to higher energy consumption and larger delays.

In contrast, our approach is fully compatible with the commercial 3D-NAND flash memory. Encoding inputs via application of digital pulses on the bit-select lines results in better energy-efficiency and speed. Moreover, our RSIR 3D-VMM eliminates the stringent requirement for large area hungry integrate capacitors used in the time-domain 3D-VMM [43] which results in a record-breaking area efficiency of ∼0.02 μm2/Byte, which is 5× better compared to the design based on the original approach. A smaller integrate capacitor also reduces the sweeping current requirement and alleviates the need for using level-shifters and switching the WL layer for sweeping currents. This results in a high energy efficiency of ∼6 fJ/Op, which is 1.3× improvement over its charge-based counterpart. Our results also show that the proposed RSIR 3D-VMM achieves a ∼500× improvement in the area efficiency compared to the 2D-NOR flash-based time domain VMM [27] while maintaining a comparable energy efficiency and throughput. Moreover, the digital nature of the circuit peripheries (level-shifters, neuron, DTC, and TDC) in our proposed time-domain VMM significantly relaxes the limitation for technology-node scaling as opposed to the analog nature of the peripheral circuits in the amplifier/current-mirror (voltage/current-mode) based approaches. Also, in the proposed design, the precision is mainly constrained by the inherent 3D-NAND flash cell characteristics such as DIBL and capacitive coupling, and not by the peripheral circuitry characteristics such as gain, noise, and their sensitivity to process variation. Considering the extremely high density of 3D-NAND cells due to 3D integration, the proposed approach can significantly benefit from technology-node scaling even when the size of the individual flash cells does not scale proportionately.

On the system level, quite a few efforts were recently made to exploit the efficiency of mixed-signal (MS) operators to develop better DNN/RNN processor architectures [44, 45, 50–54]. For example, the ISAAC [44] and PUMA [45] architectures are 2D mesh structures of tiles where each tile contains several small (typically 128×128) ReRAM-based VMM units with their I/O peripheries. In these architectures, one shared memory is implemented in each tile for storing intermediate data and communication between the VMMs, while communications between the tiles are performed through a shared 2D bus structure. Such a heavily-granular, multi-core design approach aims at increasing the VMM unit utilization, minimizing the data transfer overhead, and maximizing the system throughput via pipelining and parallel processing. However, the data conversion / communication overhead due to the partial VMM operation, static power consumption and large area overhead of the neurons / DACs / ADCs, and a large control and communication overhead between tiles /VMMs likely limits the performance of such architectures, especially when running relatively complex computational graphs such as those of the Inception [4] and ResNet [5] tasks.

In contrast to this prior system-level work, our 3D-aCortex processor architecture is harmonically matched with the proposed 3D-NAND VMM as the core processing unit. It includes a flexible/programmable granular single-bank 3D analog operator and a reconfigurable folded chain of buffers, which allows the contingent implementation of various size VMMs and convolution kernels fully in the time/analog domain. Such design results in maximizing the data reuse while minimizing the area overhead of peripheral and control circuitry, as well as the energy overhead of the VMM operation (integration and I/O conversion) and control/data movement associated with heavily multi-core designs performing partial VMM operations [44, 45]. The main advantages of the proposed architecture are:

A flexible single-bank design, which results in a very large sharing factor of costly peripheral circuitry such as buffers, DTCs, neurons, TDCs, and programming circuitry, while maintaining the capability of

performing various size VMM operations. The large sharing factor of the peripheral circuitry and a high density of 3D-NAND memory result in remarkable storage efficiency.

Such a design provides a flexible large VMM operator fully in the time/analog domain, and consequently allows the contingent implementation of VMMs of various sizes, fully exploiting the energy efficiency and speed of computation in the time/analog domain, i.e., avoiding overheads of partial VMM operations.

The layer-by-layer processing scheme, combined with the single-bank deployment of analog operators, results in relatively simple control circuitry, with low energy/area overhead while still supporting even complex computational graphs.

The data reuse in convolution layers is fully preserved via a configurable folded buffer chain design.

Due to the time-domain approach, zero static power of the computational blocks improves energy efficiency.

The detailed simulation results for 3D-aCortex, benchmarked on representative RNN/DNN models, have shown a performance significantly higher than all published prior results, including the fully digital [7, 8, 14] and MS [18, 44, 45] systems - especially for mobile/IoT applications, for which the storage and energy efficiencies are the most important metrics (table 1). To make a fair comparison between 3D-aCortex and other MS approaches, we have performed a highly optimistic rescaling of the published performance metrics to the 55-nm, 4-bit design point. Even with this highly optimistic projection, different versions of 3D-aCortex (i.e., baseline charge-based, charge-based with 16× CAP-sharing, RSIR-sq2, and RSIR-sq3) provide a ∼(17, 119, 34, 34)× improvement of the storage efficiency, and a ∼(14, 13, 22, 23)× improvement of the energy efficiency overthe ISAAC [44], while maintaining a comparable computational efficiency of (0.58, 0.2, 0.96, 0.7) TOp/(s-mm^2). In comparison with PUMA [45], these numbers are, respectively ∼(17, 119, 34, 34)× and ∼(6, 5.5, 9.5, 10)×. These results also show that in comparison with the 2D-aCortex based on 55-nm NOR flash memory [18], the chip footprint of the 3D-aCortex is ∼(16, 7, 32, 32) times smaller, while its energy efficiency is lower only by a factor of ∼(5.4, 5, 3. 2.85)×.

Moreover, the proposed 3D-aCortex architecture is based on the 3D-NAND flash technology and digital time-domain peripheral circuitry, allowing for its further scaling beyond 20-nm technology node without performance/precision degradation. This fact promises even more compact and energy-efficient neuromorphic processors based on future, more advanced technology nodes.

6. Discussion and summary

It is worth stressing that the focus of this work is on neuromorphic inference, which typically does not require frequent weight updates, and hence a more advanced, longer write-verify schemes could be utilized for writing memory cells. The use of such write schemes would be essential for high precision tuning of the memory cell currents to the desired values in the presence of significant variations in memory cell characteristics see an example of such tuning for NOR flash circuits in reference [59]. Moreover, advanced write-verify algorithms would be naturally required for dealing with the NAND array back pattern effect [60], i.e., the dependence of the programmed currents of some specific cell on the programmed states of other memory cells in the string. For example, back pattern effect can be significantly reduced by exploiting the programming sequence starting from the drain side, by increasing the V_{WL} for the pass mode, or even strategically choosing V_{WL} based on the information about all the weights in the memory string, and/or by performing fine tuning of the memory cells in several passes. (Note, however, that the utilized behavioral compact model developed for 3D-NAND flash memory in [38] effectively captures the impact of the DIBL and series resistance of the unselected cells on the string current characteristics and the shift in the threshold voltage in the read mode).

An essential future work includes quantifying the impact of retention-induced charge losses in 3D-NAND memory cells and stuck-at-0/1 cell type of defects on the performance and functionality of the proposed circuits and the development of mitigation schemes against these imperfections. Our results for NOR flash memory neuromorphic circuits (e.g., measuring drift in cell currents after seven months of shelf time [25]) are very encouraging, indicating that the retention-induced charge losses for most cells could be addressed by infrequent cell re-tuning. Furthermore, it should be possible to avoid cells with more significant drift and stuck-at cells by re-mapping the weight kernels or mapping zero' weight to these cells. It should be noted, however, that most 3D-NAND flash memories utilize charge trap mechanisms instead of floating gate structures, and, therefore, the drift issue requires further investigation. Additionally, our modeling study is based on the technology reported in references [47, 48], while the device parameters might be somewhat different for more common commercially-used 3D-NAND memories, e.g., in that they utilize denser hexagonally string packing instead of the cubic one assumed in this paper and larger effective (tunneling and blocking) silicon oxide thicknesses. Larger effective oxide thicknesses would likely result in worse DIBL and hence somewhat

worse computing precision, while tighter string spacing would lead to higher parasitic capacitance, i.e., somewhat lower performance and energy efficiency. Unfortunately, more accurate modeling is challenging now since many details of such advanced 3D-NAND memories are still missing.

To summarize, in this work, we propose an alternative time-domain scheme with a resistive load called the RSIR approach, which effectively eliminates the requirement of a bulky load capacitor leading to a significant improvement in the area-efficiency. Moreover, this scheme facilitates the efficient implementation of VMMs with sub-maximal output cell current ranges, which is a key feature in the mixed-signal neural processors. VMM-level simulations considering various non-idealities including line/device parasitics, variations, and noise show that the RSIR-VMM targeting sub-maximal output ranges achieves 4 bits of computational precision while exhibiting an improvement of 5×/1.3× in area/energy-efficiency compared to its charge-based counterpart with maximal, full output current range.

We also propose a novel neural accelerator architecture called 3D-aCortex embedding time-domain 3D-VMMs as its core processing element. This architecture aims to minimize the input/output peripheral circuitry overhead, a major limiting factor for area/energy efficiency in mixed-signal architectures [44, 45], via sharing these elements among a large 2D array of time-domain VMMs. Flexible activation of processing elements and multilayer 3D-NAND memory structure allows efficient 3D packing of neural layer weight matrices onto the array of processing elements. Moreover, the data transfer cost is further minimized using a flexible folded chain of digital input buffers that are taking advantage of the data reuse for convolution operation and a multi-agent controller enabling the time overlapping between input and output data transfer for sequential VMM operations on the same weight kernel.

The system-level results are estimated using an in-house 3D-aCortex estimator, which imports detailed block-level simulation results and the computational graph for the target neural network and generates a comprehensive system-level performance report. Using such tool, we performed rigorous system-level estimations for charge-based and RSIR approaches designed for full output current range, and multiple sub-maximal output ranges targeting a neural benchmark set which includes various network types/sizes, consisting of Inception-v1 [4] and ResNet-152 [5] deep neural networks, and GNMT network [6]. The results indicate that RSIR-3D-aCortex with sub-maximal output range achieves ∼9 MB/mm^2 storage efficiency, 113 ÷ 118 TOp/J peak energy efficiency, and 6.3 ÷ 8.6 TOp/s computational throughput, which is a 2× and 1.5× ÷ 1.6× improvements, respectively, in storage and energy efficiency at the cost of 20%–40% degradation in throughput compared to its full-range charge-based counterpart.

7. Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

This work was supported by a Semiconductor Research Corporation (SRC) funded JUMP CRISP center, by NSF/SRC E2CDA grant 1740352, and by Samsung grant SB180048

ORCID iDs

Dmitri B Strukov https://orcid.org/0000-0002-4526-4347

References

[1] Mohammadi M, Al-Fuqaha A, Sorour S and Guizani M 2018 Deep learning for IoT big data and streaming analytics: a survey *IEEE Commun. Surv. Tutorials* **20** 2923–60
[2] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
[3] Krizhevsky ASutskever IHinton G E 2017 ImageNet classification with deep convolutional neural networks *Communications of the ACM* **60** 84–90
[4] Szegedy C *et al* 2015 Going deeper with convolutions *Proc. IEEE CVPR* (Boston, MA) pp 1–9
[5] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE CVPR* pp 770–8
[6] Wu Y *et al* 2016 Google's neural machine translation system: bridging the gap between human and machine translation (arXiv:1609.08144)
[7] Jouppi N P *et al* 2017 In-data center performance analysis of a tensor processing unit *Proc. ACM/IEEE Int. Symp. on Computer Architecture* (Toronto, ON) (ISCA) pp 1–12
[8] Chen Y *et al* 2014 DaDianNao: a machine-learning supercomputer *Proc. IEEE/ACM Int. Symp. on Microarchitecture* (Cambridge) 609–22

[9] Chen Y-H, Krishna T, Emer J S and Sze V 2017 Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks *IEEE J. Solid-State Circuits* **52** 127–38

[10] ARM ML processor (https://arm.com/products/processors/machine-learning)
Intel Mobileye (https://mobileye.com/en-us/)
Google edge TPU (https://cloud.google.com/edge-tpu/)

[11] Moons B, Roel U, Wim D and Marian V 2017 14.5 envision: a 0.26-to-10 TOps/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm FDSOI *Proc. IEEE Solid-State Circuits Conf. (ISSCC)* pp 246–7

[12] Davies M *et al* 2018 Loihi: a neuromorphic manycore processor with on-chip learning *IEEE Micro* **38** 82–99

[13] Merolla P A *et al* 2014 A million spiking-neuron integrated circuit with a scalable communication network and interface *Science* **345** 668–73

[14] Lee J *et al* 2018 UNPU: an energy-efficient deep neural network accelerator with fully variable weight bit precision *IEEE J. Solid-State Circuits* **54** 173–85

[15] Hubara I *et al* 2016 Quantized neural networks: training neural networks with low precision weights and activations (arXiv:1609.07061)

[16] McKinstry J L *et al* 2018 Discovering low-precision networks close to full-precision networks for efficient embedded inference (arXiv:1809.04191)

[17] Xu C, Yao J, Lin Z, Ou W, Cao Y, Wang Z and Zha H 2018 Alternating multi-bit quantization for recurrent neural networks (arXiv:1802.00150)

[18] Bavandpour M *et al* 2018 Mixed-signal neuromorphic inference accelerators: recent results and future prospects *Proc. IEDM 18* (San Francisco, CA)

[19] Hu M *et al* 2016 Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication *Proc. DAC 16* (Austin, TX) pp 1–6

[20] Bayat F M, Prezioso M, Chakrabarti B, Nili H, Kataeva I and Strukov D 2018 Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits *Nat. Commun.* **9** 2331

[21] Yao P *et al* 2017 Face classification using electronic synapses *Nat. Commun.* **8** 15199

[22] Kim K-H, Gaba S, Wheeler D, Cruz-Albrecht J M, Hussain T, Srinivasa N and Lu W 2011 A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications *Nano Lett.* **12** 389–95

[23] Burr G W *et al* 2014 Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element *Proc. IEEE Int. Electron Devices Meeting* (San Francisco, CA) 29.5. 1–4

[24] Boybat I *et al* 2018 Neuromorphic computing with multi-memristive synapses *Nat. Commun.* **9** 2514

[25] Guo X *et al* 2017 Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology *Proc. IEDM 17* (San Francisco, CA) 6.5. 1–4

[26] Mahmoodi M R and Strukov D B 2018 An ultra low energy internally analog, externally digital vector-matrix multiplier circuit based on NOR flash memory technology *Proc. Design Automation Conf.* (New York: ACM)

[27] Bavandpour M, Mahmoodi M R and Strukov D B 2019 Energy-efficient time-domain vector-by-matrix multiplier for neurocomputing and beyond *IEEE Trans. Circuits Syst.* **66** 1512

[28] Hasler J and Marr B 2013 Finding a roadmap to achieve large neuromorphic hardware systems *Front. Neurosci.* **7** 118

[29] Schlottmann C R and Hasler P E 2011 A highly dense, low power, programmable analog vector-matrix multiplier: the FPAA implementation *IEEE J. Emerg. Sel. Topics Circuits Syst.* **1** 403–11

[30] Chakrabartty S and Cauwenberghs G 2007 Sub-microwatt analog VLSI trainable pattern classifier *IEEE J. Solid-State Circuits* **42** 1169–79

[31] Fick L, Manar E C, Skrzyniarz S and Fick D 2019 Mythic Inc. system and methods for mixed-signal vomputing *US Patent Application* 10/255205

[32] Busch K F, Vorenkamp P and Bailey S W 2019 Syntiant Corp. systems and methods for customizing neural networks *US Patent Application* 16/164671

[33] Superflash technology overview, SST, Inc. available online at https://sst.com/technology/sst-superflash-technology

[34] (Compagnoni C.C.M.) Goda A, Spinelli A S, Feeley P, Lacaita A L and Visconti A2017 Reviewing the evolution of the NAND flash technology *Proc. IEEE* **105** 1609–33

[35] Park K-T *et al* 2015 Three-dimensional 128 Gb MLC vertical NAND flash memory with 24-WL stacked layers and 50 MB/s high-speed programming *IEEE J. Solid-State Circuits* **50** 204–13

[36] Kim C *et al* 2017 A 512 Gb 3b/cell 64-stacked WL 3D V-NAND flash memory *Proc. IEEE Int. Solid-State Circuits Conf.* (San Francisco, CA) (ISSCC) pp 202–3

[37] Shibata N *et al* 2019 A 1.33 Tb 4-bit/Cell 3D-flash memory on a 96-word-line-layer technology *IEEE Int. Solid-State Circuits Conf.* (San Francisco, CA) (ISSCC) pp 210–2

[38] Sahay S and Strukov D B 2019 A behavioral compact model for static characteristics of 3D NAND flash memory *IEEE Electron Device Lett.* **40** 558

[39] Wang P *et al* 2018 Three-dimensional NAND flash for vector-matrix multiplication *IEEE Trans. VLSI Syst.* **27** 988

[40] Ravinuthula V, Garg V, Harris J G and Fortes J A B 2009 Time-mode circuits for analog computation *Int. J. Circ. Theor. Appl.* **37** 631–59

[41] Wang Q, Tamukoh H and Morie T 2018 A time-domain analog weighted-sum calculation model for extremely low power VLSI implementation of multi-layer neural networks (arXiv:1810.06819)

[42] Tohara T, Liang H, Tanaka H, Igarashi M, Samukawa S, Endo K, Takahashi Y and Morie T 2016 Silicon nanodisk array with a fin field-effect transistor for time-domain weighted sum calculation toward massively parallel spiking neural networks *Appl. Phys. Express* **9** 034201

[43] Bavandpour M, Sahay S, Mahmoodi M R and Strukov D B 2020 Mixed-signal vector-by-matrix multiplier circuits based on 3D-NAND memories for neurocomputing *Proc. Design, Automation, and Test in Europe (DATE)* (Grenoble: France)

[44] Shafiee A, Nag A, Muralimanohar N, Balasubramonian R, Strachan J P, Hu M, Williams R S and Srikumar V 2016 ISAAC: a convolutional neural network accelerator with *in situ* analog arithmetic in crossbars *Proc. ACM/IEEE Int. Symp. Comp. Arch. (ISCA)* vol 44 (Seoul) pp 14–26

[45] Ankit A *et al* 2019 PUMA: a programmable ultra-efficient memristor-based accelerator for machine learning inference (arXiv:1901.10351)

[46] Klachko M, Mahmoodi M R and Strukov D 2019 Improving noise tolerance of mixed-signal neural networks *Proc. Int. Joint Conf. on Neural Networks (IJCNN)* (Piscataway, NJ: IEEE) pp 1–8
[47] Resnati D, Mannara A, Nicosia G, Paolucci G M, Tessariol P, Spinelli A S and Lacaita A L (Monzio Compagnoni C) 2018 Characterization and modelling of temperature effects in 3D NAND flash arrays-part I: polysilicon-induced variability *IEEE Trans. Electron Devices* **65** 3199–206
[48] Malavena G, Lacaita A L and Spinelli A S (Monzio Compagnoni C) 2018 Investigation and compact modeling of the time dynamics of the GIDL-assisted increase of the string potential in 3D NAND flash arrays *IEEE Trans. Electron Devices* **65** 2804–11
[49] Muralimanohar N, Balasubramonian R and Jouppi N P 2009 CACTI 6.0: a tool to understand large caches *Technical Report* HP Labs HPL-2009-85
[50] Liu X *et al* 2016 Harmonica: a framework of heterogeneous computing systems with memristor-based neuromorphic computing accelerators *IEEE Trans. Circuits Syst.* **63** 617–28
[51] Song L *et al* 2017 PipeLayer: a pipelined ReRAM-based accelerator for deep learning *Proc. IEEE Int. Symp. on High Perf. Comp. Arch.* (Austin, TX) (HPCA) pp 541–52
[52] Chi P, Li S, Xu C, Zhang T, Zhao J, Liu Y, Wang Y and Xie Y 2016 PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory *Proc. ISCA* (Seoul, Korea) pp 27–39
[53] Imani M *et al* 2018 RAPIDNN: in-memory deep neural network acceleration framework (arXiv:1806.05794)
[54] Srivastava P *et al* 2018 PROMISE: an end-to-end design of a programmable mixed-signal accelerator for machine-learning algorithms *Proc. ACM/IEEE Int. Symp. Comp. Arch.* (Los Angeles, CA) (ISCA) pp 43–56
[55] Bavandpour M, Sahay S, Mahmoodi M R and Strukov D 2019 Efficient mixed-signal neurocomputing via successive integration and rescaling *IEEE Trans. VLSI Syst.* **28** 823
[56] Micheloni R 2016 *3D Flash Memories* (Berlin: Springer)
[57] Samsung V-NAND Technology Available: https://samsung.com/semiconductor/global.semi.static/2bit_V-NAND_technology_White_Paper-1.pdf[online]
[58] Lue H T *et al* 2018 A novel 3D AND-type NVM architecture capable of high-density, low-power in-memory sum-of-product computation for artificial intelligence application *Proc. IEEE Symp. on VLSI Tech.* (Piscataway, NJ: IEEE) pp 177–8
[59] Merrikh Bayat F *et al* 2016 Model-based high-precision tuning of NOR flash memory cells for analog computing applications *Proc. DRC 16* (Newark, DE) pp 1–2
[60] Chen C *et al* 2014 Study of the programming sequence induced back-pattern effect in split-page 3D vertical-gate (VG) NAND flash *Proc. IEEE Int. Symp. VLSI Tech., Sys. and App. (VLSI-TSA)* pp 1–2