Mixed-Signal Vector-by-Matrix Multiplier Circuits Based on 3D-NAND Memories for Neurocomputing

Mohammad Bavandpour, Shubham Sahay, Mohammad Reza Mahmoodi, Dmitri Strukov
ECE Department, UC Santa Barbara, Santa Barbara, USA
{mbavandpour, shubhamsahay, mrmahmoodi, strukov}@ece.ucb.edu

***Abstract*— We propose an extremely dense, energy-efficient mixed-signal vector-by-matrix-multiplication (VMM) circuits based on the existing 3D-NAND flash memory blocks, without any need for their modification. Such compatibility is achieved using time-domain-encoded VMM design. We have performed rigorous simulations of such a circuit, taking into account non-idealities such as drain-induced barrier lowering, capacitive coupling, charge injection, parasitics, process variations, and noise. Our results, for example, show that the 4-bit VMM of 200-element vectors, using the commercially available 64-layer gate-all-around macaroni-type 3D-NAND memory blocks designed in the 55-nm technology node, may provide an unprecedented area efficiency of 0.14 μm^2/byte and energy efficiency of ~11 fJ/Op, including the input/output and other peripheral circuitry overheads.**

Keywords—Mixed-signal VMM, 3D-NAND flash memory, Time domain encoding scheme.

I. Introduction

The vector-by-matrix multiplication (VMM) is the most common operation in deep neural networks and many other tasks. This fact is the motivation for the current intensive development of efficient VMM circuits and optimal architectures for their deployment in neuromorphic processors. Most VMM implementations are digital, with many commercial and experimental processor architectures developed recently, see, e.g. review in [1]. The performance of such processors on VMM-heavy benchmarks is much higher compared to the standard CPUs, in part due to using low-precision operations, suitable for the most frequent inference function. Digital approaches, however, lead to relatively sparse design, which necessitates storing most of the synaptic weights off-chip, hence paying large performance penalty for memory access. As demonstrated by prior work, these inefficiencies could be overcome by utilizing mixed-signal (MS) circuits based on advanced analog-grade non-volatile memory devices [2, 3]. On the other hand, MS approaches to the VMM tasks have their own challenges. The developed technologies for fabrication of highly scalable emerging memristive devices are not yet mature, still requiring a substantial improvements in device-to-device uniformity, and in device current reduction. The floating-gate memory cells, whose optimal design mitigates these problems, have relatively large cells, even if implemented by re-design of highly optimized commercial flash memories [3]. The resulting relatively low circuit density may lead, just like in the case of the digital implementations, to significant inter- and intra-chip data transfer overheads [3]. Additional concern is substantial area/energy overhead of conversion between analog and digital domains in MS inference accelerator architectures.

This work was supported by Samsung award SB190104.

These challenges have provided the main motivation for our work - the development of VMM circuits and architectures based on 3D-NAND memories [4]. Indeed, even the already developed commercial 3D-NAND memory technology enables record-breaking effective bit density, ultra-low fabrication cost per bit, and multi-level cell programming capability [4], while still rapidly advancing. Fig. 1a shows a typical 3D-NAND memory architecture. In it, many layers of memory cells are stacked on top of each other, with the cells connected in the z-direction (normal to the chip surface) to form a "string". On the top of each string, there is a bit-select-line (BSL) transistor that connects it to the bit line (BL). The memory block consists of a 2D (x-y-plane) mesh of such strings, with all memory cells of the same level (i.e., at the same z-position) sharing the common word-line (WL) metal plate. In addition, the strings share BSLs in the x-direction, and BLs in the y-direction.

While showing a possible dramatic increase of the stored weight density (scaling as the number of the cell layers), Fig. 1 also points to a major problem for the VMM implementation. Namely, sharing of each word line by all cells of that layer does not allow to use the "current-mode" approach that was successfully employed for the adaptation of a commercial 2D flash memory for MS-VMM [3]. In future, an appropriate redesign of the 3D wiring (perhaps, as in the 2D work, not touching the highly optimized memory cells) may be the best option. However, such modification (assumed in the recent work [5]) would require a major technological effort. (The approach in [5] also requires using high-resistance and high-capacitance WL on the critical path).

The main contribution of our work is to show that the time-domain approach to the VMM function [6-9] may enable using commercial 3D-NAND memories without any modification. After describing this approach in the beginning of section 2, we then use the balance of the paper to present quantitative analysis of the possible performance of the resulting 3D-VMM blocks, taking into account various non-idealities impacting their performance.

II. 3D-VMM Design

A. Time-domain VMM

The target analog VMM operation may be represented as

$$y_j = \frac{1}{M}\sum_{i=1}^{M} w_{ij} x_i \,, \qquad (1)$$

where x_i, w_{ij}, and y_j are real numbers, which may take any values within range [0, 1]. In the time-domain approach [9], the components x_i and y_i of the input and output vectors are encoded

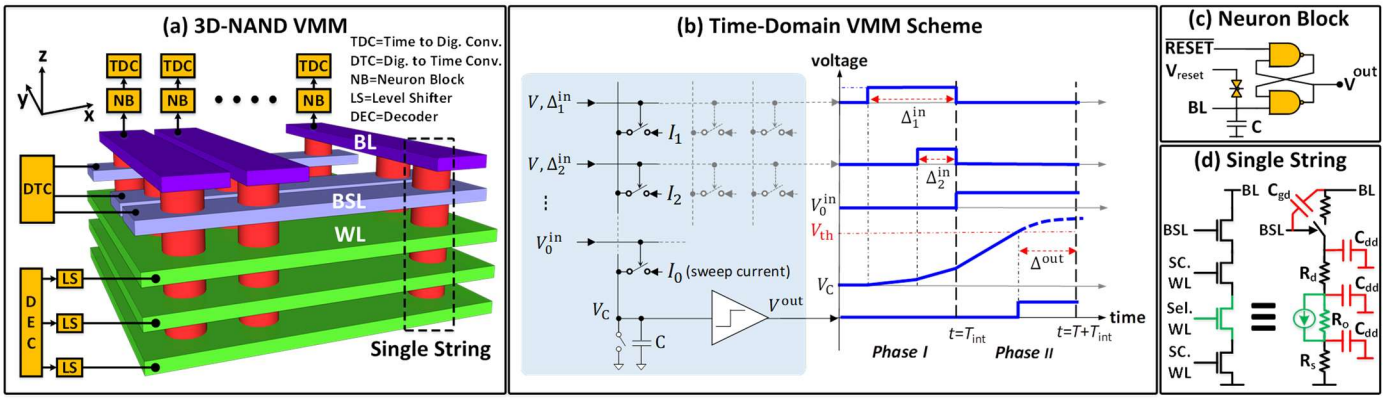

Fig.1. The main idea of the 3D-VMM circuit. (a) Cartoon of 3D-NAND flash memory block and its use in the proposed circuit. For simplicity, a layer of transistors at the bottom of the block, which connects the cell strings to the common source (ground) is not shown. (b) Basic structure and example of operation in the utilized time-domain approach [9]. (c) Circuit diagram of the peripheral neuron, which consists of a load capacitor C, connected to the bit line (BL), and an SR latch, implementing a unit step function of its input. (d) Equivalent circuit of a single string for the operation mode.

with the durations Δ of fixed-amplitude pulses: $\Delta_i^{in} = x_i T$, $\Delta_j^{out} = y_j T$, where T is a certain fixed time window, while the matrix elements ("weights") w_{ij} are represented by adjustable current sources I_{ij} within a fixed range $[0, I_{max}]$: $w_{ij} = I_{ij}/I_{max}$. (In floating-gate memory cells, the weights are kept in the form of stored floating gate charges, which define the source-to-drain currents I_{ij} at a fixed drain voltage.)

The computation is performed in two phases (Fig. 1b). During the first T_{int}-long (integration) phase, the input pulse Δ_i^{in} turns on fixed drain voltages, and hence the current sources I_{ij} of the i^{th} row, leading to the injection of electric charges equal to $I_{ij}\Delta_i^{in} \propto w_{ij}x_i$ into the j^{th} column through the corresponding memory cells. The charges from multiple rows of the j^{th} column are summed up on its load capacitor C. As a result, by the end of phase I, the capacitor voltages V_C (which are reset before the operation) become proportional to the component of the desired VMM output vector:

$$V_{C,j} = \frac{1}{C}\sum_{i=1}^{M} I_{ij}\Delta_i^{in}. \qquad (2)$$

During the second T-long phase, these voltages are converted into the durations Δ_j^{out} of the output pulses (Fig. 1b). This is done by additional charging of each load capacitor with a constant "sweep" current equal to MI_{max}, inducing a linear ramp-up of its voltage in time, starting from the value (2). At the moment when the total voltage reaches the fixed threshold V_{th}, an output fixed-amplitude pulse is initiated, with its falling edge aligned with the end of this phase II. As a result, the duration of the output pulse generated in phase II is

$$\Delta_j^{out} = \frac{1}{MI_{max}}\sum_{i=1}^{M} I_{ij}\Delta_i^{in} \quad . \qquad (3)$$

where, just for convenience, all load capacitances are assumed to be equal to $C = MI_{max}/V_{th}$. Also, note that $T \geq T_{int}$, because of the extra voltage margins reserved for coupling (see below).

The described approach can be easily extended to four-quadrant time-domain VMM, by using differential rows/columns, and a set of four cells for each weights, to represent positive and negative inputs/outputs [9].

B. 3D-VMM structure and operation

In 3D-VMM block, each elementary ("single-shot") VMM operation uses the weights recorded in the floating-gate cells of one x-y layer of the 3D-NAND memory circuit (see Fig. 1a). This layer is selected by setting its WL voltage to 2 V, while setting the cells of all other layers to the highly conductive "pass" state by applying 5 V to those WLs. The cell currents are collected and integrated at the BL. However, irrespective of the selected layer of cells, the inputs are always applied to bit-select lines. The "sweep" currents, necessary for phase II of the operation, are injected through the top layer of cells of all strings, enabled by a positive voltage applied to all BSLs.

Such elementary VMM operations, based on different layers, are used as steps of the time-division-multiplexing operation. Clearly, such VMM operation mode does not require changes in the usual NAND flash memory array, and only needs to complement it with custom-designed peripheral decoder and level-shifter circuits.

Note that because of significant WL parasitics in 3D-NAND memory, the total delay for performing one VMM elementary operation is $2T_{LS} + T_{int} + T$, where T_{LS} is the time required to select a certain layer.

C. Non-idealities

For our detailed analysis, we have specifically considered the 3D-NAND memory based on polysilicon gate-all-around macaroni-body charge-trap cells. Besides its widespread use, another reason for this choice is availability of a behavioral compact model for such memory, which may be used for quantitative simulation. In such model, individual cells are approximated as cylindrical gate-all-around nanowire FETs with a voltage-controlled-current-source [10]. The model takes into account various parasitic capacitance coupling effects, and accurately reproduces the experimental string current characteristics.

We next discuss the most important factors affecting computing precision:

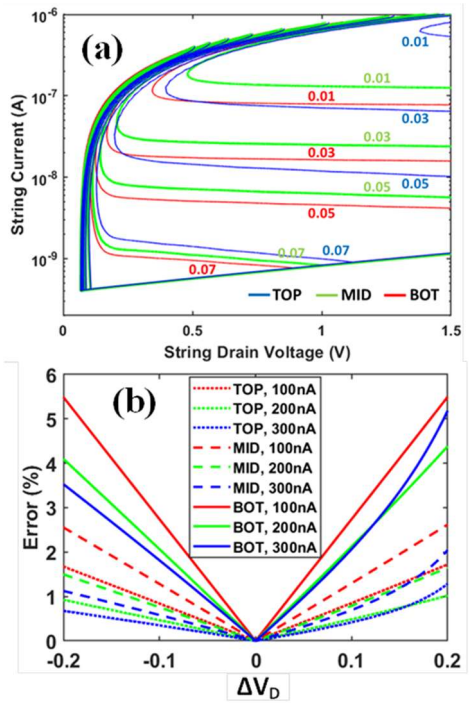

Fig. 2: (a) Small-signal DIBL error contours (shown in %) in I_D-V_D space for top, middle, and bottom layer memory cells, programmed in various states in a 64-layer 3D-NAND memory. Small-signal error is defined as $100\times(1 - I(V_D) / I(V_D+1\text{ mV}))$, i.e. relative change in string current for a 1 mV change in the BL voltage. (b) Total DIBL error (%) for ±0.2 V swing on the drain voltage around $V_D = 0.7$ V for various memory states.

Drain-induced barrier lowering (DIBL): Let us first note that since the current is sunk through the cells to the source line, we consider the scheme in which BL voltage is charged to a voltage $\Delta V_D + V_{th}$ at the start of phase I, where ΔV_D is the total voltage swing on BL during computation, and then discharged to V_{th} in the phase II.

DIBL error is defined as a relative difference of currents via string of cells at two extreme BL voltages, i.e.

$$E_{DIBL} \approx 1 - I(V_{th})/I(V_{th} + \Delta V_D). \quad (4)$$

Without considering additional headroom to deal with capacitive coupling, the typical values are $V_{th} = 0.6$ V and $\Delta V_D = 0.2$ V, which correspond to the quasi optimal operation conditions for the CMOS-based neuron implementation [9].

According to Eq. 4, the DIBL error is proportional to the small signal transconductance gain $\delta I_D/\delta V_D$ of a string over the target operating regime. Given the small signal model shown in Fig. 1d, the transconductance gain can be formulated as:

$$\frac{\partial I_D}{\partial V_D} = \frac{1}{R_D + R_0 + (1 + g_m R_0) R_S}, \quad (5)$$

where g_m and R_0 are the small signal parameters of a single memory cell, and R_D and R_S are the lumped string resistances on the drain and source side, respectively, of the selected memory cell. According to Eq. 5, larger R_D and R_S help reducing the DIBL error, but at the cost of limiting the current range. Moreover, because of stronger effect of R_S, DIBL error is less for top memory cells (which was the reason for using top layer for sweep currents). Also, DIBL error is less for larger string currents due to intrinsically larger R_0, when the selected cell operates closer to strong inversion mode. These observations are confirmed by modeling (Fig. 2). In line with Eq. 4, DIBL error increases almost linearly with the total swing in the target operation region (Fig. 2b).

Capacitive coupling: Due to the switched-capacitor nature of the proposed approach, capacitive coupling is a significant source of compute error. We break down the sources of coupling into two components. The first component, gate-drain (GD) coupling, is caused by their overlap in BSL transistor and coupling between BSL and BL wires. The second one (DD) is caused by the parasitic capacitors between the string and the rest of the memory block. These two lumped capacitors are denoted as C_{gd} and C_{dd}, respectively (Fig. 1d).

Note that C_{dd} is distributed over the total length of the string. When a 2.5 V rising edge is applied to BSL line, GD coupling results in an immediate positive disturbance charge on the BL voltage with the amount of $C_{gd}\times$ (2.5 V). Moreover, when the string is selected via BSL, DD coupling causes a negative disturbance charge on BL to charge the string parasitic capacitors C_{dd} from their initial voltage (ground) to their final DC voltage at which the string sinks the target current. When a 2.5 V falling edge is applied to BSL, the capacitive coupling is dominated by the GD coupling which causes an immediate negative disturbance charge on BL by $-C_{gd}\times$(2.5 V).

GD coupling disturbance is almost independent of the selected cell location and programming state, while the DD coupling disturbance during rising edge is highly dependent on both (Fig. 3). The amplitude and time constant of the DD charge disturbance are both larger for the cells closer to the bottom of the string due to higher voltage variation on the parasitic capacitors (C_{dd}), especially the ones closer to the bottom but higher than the selected cell where the path to both ground and BL are highly resistive.

Taking into account the coupling, we can formulate the amount of voltage disturbance on the BL for each input as $\Delta V_{cp} = Q_D/C_0$ where C_0 is the amount of load capacitance per input, and Q_D is the total disturbance charge caused by one input in both phase I when the target weight layer is selected and a rising edge followed by a falling edge is applied to BSL, and also phase II when the sweeping layer, i.e. top layer, is selected and one rising edge is applied to BSL. A major portion of Q_D, and consequently ΔV_{cp} is dependent on the location of target weight layer (Fig. 3b). Hence the maximum disturbance charge $(Q_D)_{max}$, which causes the largest disturbance voltage swing on BL $(\Delta V_{cp})_{max} = (Q_D)_{max}/C_0$, occurs when the target weight layer is at the bottom of the string.

In order to support VMM operation on all the layers, reset voltage $\Delta V_D + V_{th}$ should be selected to reserve a portion of total voltage swing on BL for the worst case voltage variation due to coupling. Hence, we select $\Delta V_D = \Delta V_{cmp} + (\Delta V_{cp})_{max}$, where ΔV_{cmp} is the voltage swing without considering the capacitance coupling for the weight and sweep current sources. Though the utilized differential scheme is robust to coupling, the output time window in which the output pulse is generated should be scaled by a coupling coefficient $\alpha_{cp} = 1 + (\Delta V_{cp})_{max}/\Delta V_{cmp}$. Note

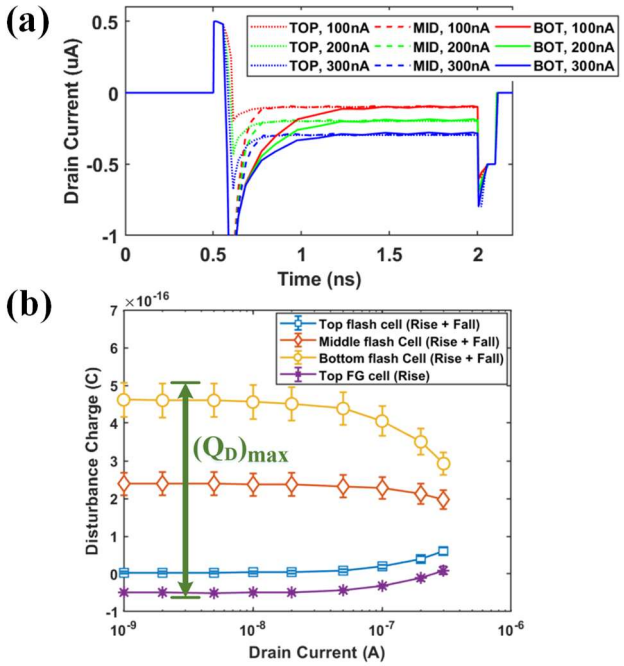

Fig. 3: Charge disturbance on BL due to capacitive coupling. (a) Time domain representation of drain (BL) current and its disturbances caused by coupling when a 2.5V rising edge (at t = 0.5 ns) followed by a same-amplitude falling edge (at t = 2 ns) is applied to the BSL for various programming states where the selected cell is located at top, middle, and bottom layer of the string. (b) Total string disturbance charge on a drain caused by capacitive coupling when a 2.5 V rising + falling edge is applied to BSL and target cell is located in top, middle, and bottom layer and programmed in various states (corresponding to phase I of computation), as well as when a single 2.5 V rising edge applied to BSL and target cell is located in top layer and programmed in various states (corresponding to phase II of computation). Error bar represents 3σ distribution of the disturbance charge due to process variations.

that a small portion of $(\Delta V_{cp})_{max}$ still affects the output precision because of difference in disturbance charge caused by positive and negative sub-weights due to process variation, and dependence of disturbance charge on the programmed state of the flash cells. Also note that a larger $(\Delta V_{cp})_{max}$ leads to a higher BL voltage swing and consequently a larger DIBL error.

Noise: White (shot/thermal) noise will dominate at the considered high-bandwidth operation. (We assume that the cells with extremely high flicker noise will be set to high conductive states and avoided during mapping.). The noise power for a single string operating in subthreshold can be approximated as ~ $2qI_{max}/T$, while SNR for a single device as $\text{SNR}^{\text{cell}} \approx 2q/I_{max}$, where q is an electron charge. Accordingly, for an M×1 VMM unit (a dot product), noise and signal power are $P_{\text{noise}}^{M\times1} = \frac{2qMI_{\max}}{T}$ and $P_{\text{signal}}^{M\times1} = (MI_{\max})^2$, respectively. Hence,

$$\text{SNR}^{M\times1} = \frac{P_{\text{signal}}^{M\times1}}{P_{\text{noise}}^{M\times1}} \approx \frac{MI_{\max}T}{2q} = M \times \text{SNR}^{\text{cell}}. \quad (6)$$

The equivalent 3σ error due to noise is derived as

$$E_{3\sigma}^{M\times1} \approx \frac{2\times3\times\sqrt{\frac{2qMI_{\max}}{T}}}{MI_{\max}} = 6 \times \sqrt{\frac{2q}{MI_{\max}T}} = \frac{E_{3\sigma}^{\text{cell}}}{\sqrt{M}}. \quad (7)$$

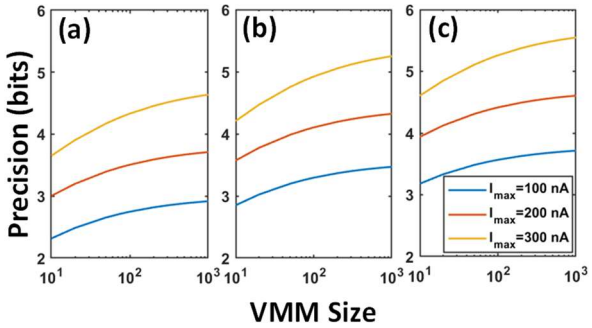

Fig. 4: 3D-NAND based VMM bit precision with respect to VMM size for I_{max} = 100 nA, 200 nA, and 300 nA for T_{int} = (a) 8 ns, (b) 16 ns, and (c) 32 ns.

In the above equation, the distribution is multiplied by two due to the differential scheme. According to the derived equation, compute error is inversely proportional to the square root of maximum current, compute time window, and the VMM size.

D. Compute precision

The compute (output) precision p_O can be defined separately from the weight precision [9] as

$$p_O = -\log_2(E_c) - 1, E_c = \frac{1}{T}\max_{\Delta^{out}}\left|\Delta^{ideal} - \Delta^{out}\right| \quad , (8)$$

where E_C is a maximum absolute difference between the ideal (Δ^{ideal}) and actual (Δ^{out}) output pulse durations, normalized by its maximum value.

The 3D-VMM circuit can be designed following various optimization targets such as precision, energy, speed and area. Here, we focus on the precision which generally limits the design space in application-specific hardware design. The main tunable circuit parameters impacting precision are I_{max} and T_{int}.

In Table I, various combinations of (T_{int}, I_{max}) are targeted to investigate the impact of these parameters on 3D-VMM's compute precision. Assuming ΔV_{cmp} = 0.2 V and $(Q_D)_{max}$ = 6×10^{-16}, we first calculate dependent parameters such as load capacitor, coupling voltage disturbance, and output time window for every combination of I_{max} and T_{inp}. Then, full circuit-level SPICE simulations are performed on 10 different VMM sizes from 10×10 to 1000×1000 with 1000-times randomized inputs/weights considering detailed parasitic models for the interconnect wires and devices, and also process variations considering the 55-nm technology node. The results for different simulated scenarios show that the compute error for the noise-free circuit remains relatively constant over the target VMM size range.

Table I also reports the SNR and 3σ noise error parameters, calculated according to Eqs. 6 and 7, and total error targeting three representative VMM sizes. Fig. 4 shows that bit-precision, corresponding to the calculated error, increases with respect to I_{max}, T_{int}, and VMM size.

E. Weight precision

Similar to 2D flash memory circuits [3], the weight precision in 3D-VMM is also expected to be affected by the tuning accuracy and drift of the analog memory state. The additional challenge for cell current tuning will be relatively large resistance R_D and R_S (Fig. 1d). The voltage drops across

TABLE I. DESIGN SPACE EXPLORATION. CIRCUIT SPECIFICATIONS AND COMPUTE ERROR (DUE TO NOISE AND CIRCUIT NONIDEALITIES) FOR VARIOUS CHOICES OF T_{INT} AND I_{MAX}. FINAL VMM ERROR IS REPORTED FOR THREE DIFFERENT VMM SIZES (M = 10, 100, AND 1000), AND THE ACHIEVABLE OUTPUT BIT-PRECISION IS SHOWN BY A COLOR CODING SCHEME IN WHICH ORANGE = 2 BITS, BLUE = 3 BITS, GREEN = 4 BITS, AND YELLOW = 5 BITS.

Input time window T_{int}	**8 ns**			**16 ns**			**32 ns**		
Maximum cell current I_{max}	**100nA**	**200nA**	**300nA**	**100nA**	**200nA**	**300nA**	**100nA**	**200nA**	**300nA**
Load capacitor per input C_0 (fF)	4	8	12	8	16	24	16	32	48
Coupling vol. swing ΔV_{cp}^{max} (mV)	150	75	50	75	32.5	25	32.5	16.25	12.5
Coupling coefficient, α_{cp}	1.75	1.375	1.25	1.375	1.1875	1.125	1.1875	1.094	1.062
Output time window T_{out} (ns)	14	11	10	22	19	18	38	35	34
Single device SNR^{cell} (dB)	33.97	36.98	38.75	36.98	40	41.76	40	43.01	44.77
Single device noise 3σ error (%)	12	8.48	6.92	8.48	6	4.89	6	4.24	3.46
Noise-free VMM comp. error (%)	6.24	3.55	1.79	4.25	2.31	1.16	3.62	1.92	0.96
Final compute error M =10 (%)	10.03	6.23	3.98	6.93	4.20	2.71	5.51	3.26	2.05
Final compute error M =100 (%)	7.44	4.40	2.48	5.10	2.91	1.65	4.22	2.34	1.30
Final compute error M =1000 (%)	6.62	3.81	2.01	4.52	2.50	1.31	3.81	2.05	1.07

these resistors (especially R_S) must be taken into account while optimizing the programming scheme for a target output current.

Quantitative analysis of such factors is challenging, mostly due to the lack of published relevant data. It should be noted, however, that the utilization of barrier-engineered materials and the gate all-around architecture in the 3D-NAND memory results in a narrower threshold voltage distribution and a lower threshold voltage shift due to cell-cell coupling as compared to the planar counterparts. In fact, multi-level state capabilities (> 3 bits) have been routinely demonstrated in 3D-NAND memories, and is expected to further improve as its technology continues to advance [4].

III. CASE STUDY: 4-BIT VMM WITH DIGITAL I/O

The 3D-VMM parameters can be chosen to operate with any precision from 2 bits to 5 bits. Here we describe the results obtained for the 4-bit precision, which is sufficient for many neuromorphic inference tasks [3]. A 4-bit 3D-VMM block consists of the following main components (Fig. 1a):

- **DTC** converts the digital input to the time-domain pulse of fixed amplitude and controllable duration. As was described earlier [9], this unit includes one shared 4-bit counter and one 4-bit comparator connected to a 1-bit latch per input.
- **3D-FM** is the 3D-NAND memory block for the $M\times N$ (per layer) VMM, which consists of $M\times 2N$ cells with the dimensions reported in [10], as well as an extra marginal space for routing the word and bit-select lines. Note that the parasitics of the word-line plate extensions by routing and vias/wires are taken into account in the simulations.
- **CAP** stands for the load capacitor. We assume MOSCAP implementation in the 55-nm technology, and also account for an extra marginal space around each capacitor. The use of MOM/MIM capacitors would further improve density.
- **NB** represents the neuron circuit, consisting of a pair of NAND latches and a couple of AND and NOT logic gates for implementing the differential scheme.
- **TDC** converts the time-encoded digital output to the corresponding digital output number. This unit consists of a 4-bit adder and a 4-bit DFF per output. The adder and the DFFs form an accumulator, which counts the duration of the output pulse, using clock pulses (shared by all accumulators). This unit along with DTC constitutes the "I/O".
- **$\overline{\text{WL}}$** represents the word-line level shifters, which apply the read/pass voltages (2 V / 5 V) to the word-line plates (Fig. 1a). Note that the width of each driver transistors is made proportional to the area ($M\times N$) of the plate it serves, in order to keep the layer selection time (T_{LS}) within a limited range comparable to the computation time.
- **$\overline{\text{BSL}}$** is an array of level-shifters driving the bit-select lines and converting the 1.2 V time-encoded, fixed-amplitude input pulses to 2.5 V digital pulses.

As Table I shows, the optimal design point, which guarantees the 4-bit precision across VMMs of various size is I_{max} = 300 nA, and T_{int} =16 ns. Fig. 5 shows the energy, area, and throughput calculation results for various sizes of our 3D-VMM, as well as the energy and area breakdowns for this design point. The energy consumption is dominated by the

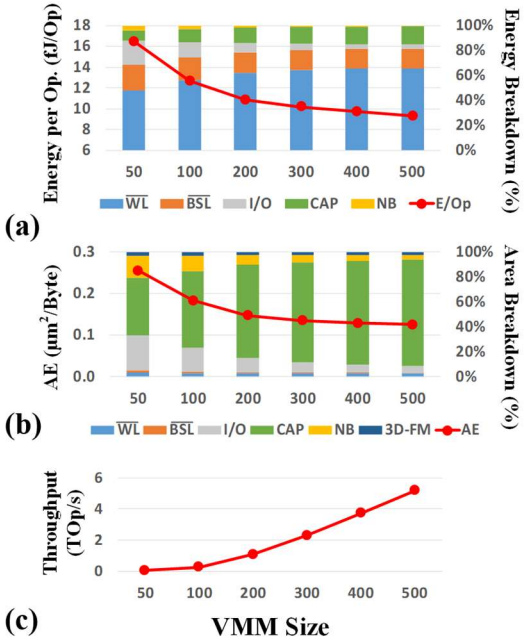

Fig. 5: 3D-NAND based VMM performance metrics. (a) Energy per operation breakdown. (b) Area efficiency breakdown. (c) Throughput as a function of VMM size.

word line selection and by feeding the inputs into the bit-select lines. (The per cell capacitance of the bit-select lines is lower than that of the load capacitance C_0, but their voltage swing is higher). The contribution of I/O and neuron circuits to the energy consumption decreases as the VMM size increases, due to their higher sharing factor. As a result, the energy per operation is only ~9 fF for $M = N = 500$. The area is dominated by the CAP, though its contribution is minor in energy consumption (Fig. 5b). Similar to the energy trend, the relative areas of I/O and neuron get smaller for larger VMM sizes. Finally, Fig. 5c shows the VMM's throughput for its various sizes considering scaling to maintain T_{LS} within the range of [20 ns, 30 ns].

IV. VMM Design for Lower Currents

In the presented performance analysis, the largest current flowing into a neuron is $M \times I_{max}$, which corresponds to the largest possible values of weights and inputs for the dot product operation. For digital circuits, this is analogous to rounding full precision (i.e., $2p + \log_2 M$ bit long) dot product result to p most significant bits. In some applications, neuron input currents might be always well below their maximum possible value. In this case, it is natural to tailor VMM design for the specific largest expected dot-product output currents to minimize the impact of rounding and quantization.

The straightforward modification of the proposed design to accommodate lower currents is to shrink the load capacitor. Such approach, however, may result in large voltage disturbance caused by the capacitive coupling of 3D-NAND array, and, in turn, in a significant drop of VMM output precision. A better approach in this case is to use resistive successive integration and re-scaling (RSIR) VMM design, which is adapted from SIR concept recently introduced in [11] (Fig. 6a,b). In such VMM, input bits are presented in a sequential manner, and an iterative integration and re-scaling (division by 2) operation is performed to calculate the final results, similar to a digital serial multiplier. A load resistor (R_I) is added to the conventional SIR-VMM to minimize the effect of capacitive coupling so that a coupling-free weighted-sum is calculated in each iteration as $V_{c,j}^k = R_I \sum_{i=1}^{M} I_{ij} x_i^k + V_{c,j}^{k-1}$, where k is a step number (bit position). Fig. 6c shows the preliminary estimates for the error in 3D-NAND RSIR VMM as a function of its size for two scenarios in which the maximum VMM current output is equal to $M^{1/sq} \times I_{max}$, with considered sq = 2 and sq = 3. (For example, sq=2 is representative of extracting p bits from the middle of the full precision result.) These results show that 3D-VMM precision can be maintained in >4 bits range even for a very low output current range of [0, $M^{1/3} \times I_{max}$]. These results are preliminary, and a thorough design-space optimization is an important future work.

V. Summary

We have proposed and performed detailed simulations of VMM circuits based on the native 3D NAND memories, not requiring any redesign. As a case study, we have considered 4-bit 3D-VMM with digital input/output interface and showed that such design achieves a ~100× better area efficiency than

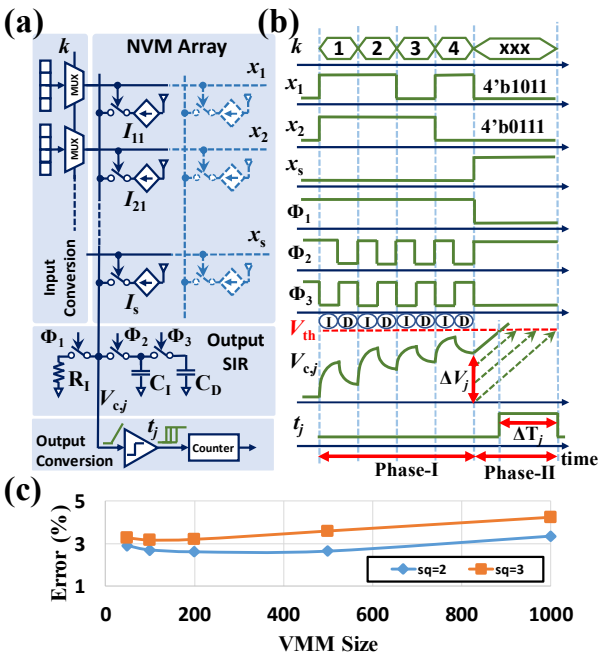

Fig. 6: (a) VMM structure and (b) signaling of the proposed resistive successive integration and re-scaling (RSIR) VMM. (c) Compute error (%) for 3D-NAND RSIR VMM as a function of its size.

that of its 2D-NOR memory-based counterpart [3], while maintaining a comparable energy efficiency and throughput. Such mixed-signal 3D-NAND VMM circuits are especially appealing for accelerating inference function of large complexity neural network models, whose weights cannot be stored locally on a chip using conventional approaches.

References

[1] V. Sze, Y.H. Chen, T.J. Yang, J. Emer, "Efficient processing of deep neural networks: A tutorial and survey", *Proc. IEEE*, vol. 105, no. 12, pp. 2295-2329, 2017.

[2] G.W. Burr et al., "Neuromorphic computing using non-volatile memory", *Advances in Physics: X*, vol. 2, no. 1, pp. 89-124, 2017.

[3] M. Bavandpour et al., "Mixed-signal neuromorphic inference accelerators: Recent results and future prospects," in: *Proc. IEDM'18*, San Francisco, CA, Dec. 2018, pp. 20.4.1-20.4.4.

[4] C.M. Compagnoni, A. Goda, A.S. Spinelli, P. Feeley, A.L. Lacaita, and A. Visconti, "Reviewing the evolution of the NAND flash technology," *Proc. IEEE*, vol. 105, no. 9, pp. 1609-1633, 2017.

[5] P. Wang et al., "Three-dimensional NAND flash for vector-matrix multiplication," *IEEE TVLSI,* vol. 27, no. 4, pp. 988-991, 2019.

[6] V. Ravinuthula, V. Garg, J.G. Harris, and J.A. Fortes, "Time-mode circuits for analog computation," *Int. J. Circ. Theor. App.,* vol. 37, pp. 631-659, 2009.

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

[8] T. Tohara et al., "Silicon nanodisk array with a fin field-effect transistor for time-domain weighted sum calculation toward massively parallel spiking neural networks," *APEX*, vol. 9, art. 034201, 2016.

[9] M. Bavandpour, M.R. Mahmoodi, and D.B. Strukov, "Energy-efficient time-domain vector-by-matrix multiplier for neurocomputing and beyond", *IEEE TCAS-II: Express Briefs*, vol. 66, pp. 1512-1516, 2019.

[10] S. Sahay and D. B. Strukov, "A behavioral compact model for static characteristics of 3D NAND flash memory," *IEEE EDL*, vol. 40, no 4, pp. 558-561, 2019.

[11] M. Bavandpour, S. Sahay, M.R. Mahmoodi, and D.B. Strukov, "Efficient mixed-signal neurocomputing via successive integration and re-scaling," *IEEE TVLSI,* 2019 (early view).