Energy-Efficient Time-Domain Vector-by-Matrix Multiplier for Neurocomputing and Beyond

Mohammad Bavandpour, Mohammad Reza Mahmoodi, and Dmitri B. Strukov

***Abstract*—We propose an extremely energy-efficient mixed-signal $N \times N$ vector-by-matrix multiplication (VMM) in a time domain. Multi-bit inputs/outputs are represented with time-encoded digital signals, while multi-bit matrix weights are realized with adjustable current sources, e.g., transistors biased in subthreshold regime. The major advantage of the proposed approach over other types of mixed-signal implementations is very compact peripheral circuits, which would be essential for achieving high energy efficiency and speed at the system level. As a case study, we have designed a multilayer perceptron, based on two layers of 10 × 10 four-quadrant multipliers, in 55-nm process with embedded NOR flash memory technology, which allows for compact implementation of adjustable current sources. Our analysis, based on memory cell measurements, shows that >6 bit operation can be ensured for larger ($N > 50$) VMMs. Post-layout estimates for 55-nm 6-bit VMM, which take into account the impact of PVT variations, noise, and overhead of I/O circuitry for converting between conventional digital and time domain representations, show ∼7 fJ/Op for $N > 500$. The energy efficiency can be further improved to POp/J regime for more optimal and aggressive designs.**

***Index Terms*—Time-domain computing, floating gate memory, vector matrix multiplication, neuromorphic computing.**

I. Introduction

VECTOR-BY-MATRIX multiplication (VMM) is one of the most common operations in many computing applications and, therefore, the development of its efficient hardware is of the utmost importance. The most promising implementations of low to medium precision VMMs are arguably based on analog and mixed-signal circuits [1]–[6]. In this brief, we propose to perform vector-by-matrix multiplication in a time-domain, combining configurability and high density of the current-mode implementations [2]–[4] with energy-efficiency of the switch-capacitor approach [5], however, avoiding costly I/O conversion of the latter. Our approach draws inspiration from prior work on time-domain computing [6]–[10], but different in several important aspects. The main difference with respect to [8]–[10] is that our approach allows for precise four-quadrant VMM using analog input and weights. Unlike the work presented in [7], there is no weight-dependent scaling factor in the time-encoded outputs, which allows chaining multipliers to implement large-scale circuits completely in a time domain. Finally, post-layout performance results are presented for a representative circuit, designed in 55 nm process with embedded NOR flash memory technology.

Manuscript received October 27, 2018; revised December 8, 2018; accepted January 5, 2019. Date of publication January 9, 2019; date of current version August 27, 2019. This work was supported by NSF under Award CCF-1528502. This brief was recommended by Associate Editor H. Li. *(Corresponding author: Dmitri B. Strukov.)*

The authors are with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: mbavandpour@ece.ucsb.edu; mrmahmoodi@ece.ucsb.edu; strukov@ece.ucsb.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2019.2891688

II. Time-Domain Vector Matrix Multiplier

In our approach, inputs are encoded with durations of the digital pulses Δ, while weights with currents I from adjustable current sources (Fig. 1a). Assuming that the i-th digital input pulse turns on i-th current source with current I_i for a total duration Δ_i, the N-element dot product is calculated by simply integrating the charge flowing into an output capacitor C, i.e.,

$$Q_\Sigma = \sum_{i=1}^{N} I_i \Delta_i.$$

Fig. 1b explains how charge Q_Σ is converted back to the time-domain representation. Let us first assume that $\Delta_i \in [0, T]$, i.e., inputs are always applied during the first T-long interval (phase I), and $I_i \in [0, I_{\max}]$. As a result of integration, voltage at the capacitor would range from is 0 to $V_{\text{TH}} \equiv I_{\max} NT$ by the end of phase I. (Here, the smallest voltage corresponds to the case when all current sources are set to 0, i.e., zero weights, or when all duration of pulses are 0, i.e., zero input, or both. The largest value corresponds to the case when all N current sources contribute $I_{\max}$ during period of time T, i.e., the whole duration of phase I.) Capacitor voltage is converted to the pulse-duration-encoded output $\Delta_\Sigma \in [0, T]$ in the second T-long interval (phase II). This is done by continuing charging the capacitor with a constant rate during phase II (Fig. 1b), until it reaches the threshold (maximum) voltage V_{TH}. The threshold crossing triggers an output pulse which always ends at time $2T$.

Specifically, during phase II, all current sources are turned on. Additionally, an extra 'bias' current source with current

$$I_0 = I_{\max} N - \sum_{i=1}^{N} I_i$$

is introduced to ensure a total constant current $NI_{\max}$ during Phase II and to linearly map the end of Phase I capacitor voltage to the corresponding pulse duration. Indeed, with such scheme, the relative time t_Σ within phase II at which the threshold voltage is crossed is determined by equation $CV_{\text{TH}} = Q_\Sigma + NI_{\max} t_\Sigma$, so that

$$\Delta_\Sigma = T - t_\Sigma = \frac{1}{I_{\max} N} \sum_{i=1}^{N} I_i \Delta_i. \tag{1}$$

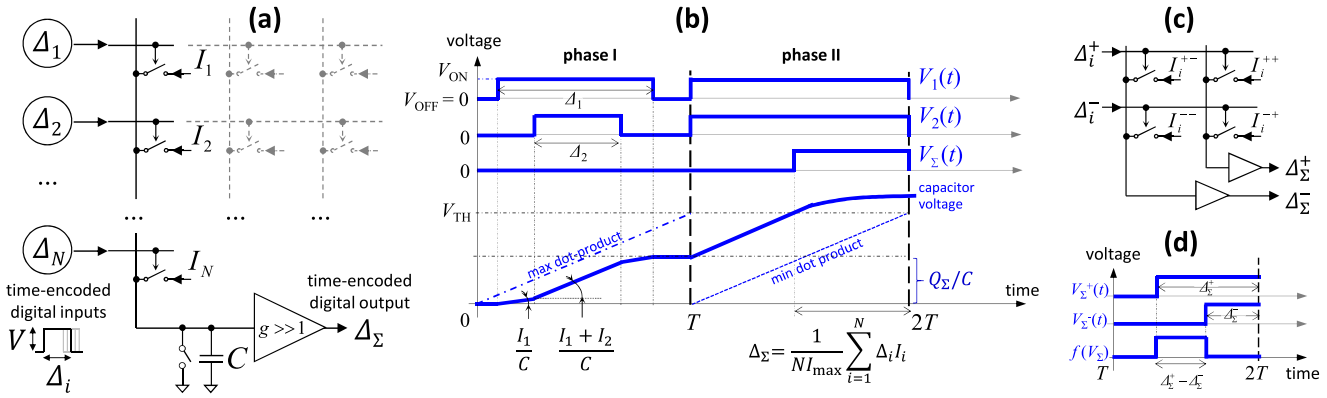

Fig. 1. The main idea of time-domain vector-by-matrix multiplier: (a) Circuit and (b) timing diagrams explaining the operation for single-quadrant multiplier. Note that panel a does not show bias input I_0. (c) Four-quadrant multiplier circuit diagram, showing for clarity only one matrix weight, implemented with four current sources I_i^{++}, I_i^{--}, I_i^{-+}, and I_i^{+-}. (d) The timing diagram showing example of the positive output of the four-quadrant multiplier. The bottom diagram for $f(V_\Sigma)$ corresponds to the case-study circuit implementation (Fig. 2).

Equation (1) defines N-element time-domain one-quadrant (1Q) dot-product with non-negative inputs and weights. The maximum weight is mapped to maximum current $I_{\max}$, while the largest input would correspond to T. The dot-product output is normalized such that its range is similar to input values, irrespective of the utilized weights. This makes our approach different from prior proposals [6], [7] and also simplifies chaining of multiple VMM circuits and connecting it to other time-domain circuits [11].

The extension to 4Q VMM is shown in Fig. 1c. In this case, each weight is represented by four current sources. To multiply input by the positive weight, I_i^{++} and I_i^{--} are set to the required positive values, with the other current source pair set to $I_i^{+-} = I_i^{-+} = 0$, while it is the opposite for the multiplication by the negative weights. The magnitude of the applied inputs / computed outputs are still encoded with time, similar to the 1Q dot-product operation, while their signs are explicitly implied from the specific row / column of a pair. Because the output pulses are always aligned with the end of phase II, a simple logical AND operation between a pair of differential outputs allows converting from differential representation into single-ended one (Fig. 1d), which, in turn, leads to simpler peripheral circuits.

It is worth noting that a similar approach with bias current turned on during both phase I and II, which requires appropriate adjustment of all currents I_i, was introduced in earlier version of this brief [12].

III. Case Study: Perceptron Network

We next apply proposed approach to implement two-layer perceptron (Fig. 2a,b). Figure 2c shows gate-level implementation of VMM and ReLU circuits, suitable for pipelined operation. The thresholding (digital buffer) is implemented with S-R latch. The ReLU functionality is realized with one AND gate which takes input from two latches that are serving a differential pair. The AND gate generates a voltage pulse with $\Delta_\Sigma^+ - \Delta_\Sigma^-$ duration for positive VMM outputs (see Fig. 1d) and zero output voltage for negative ones.

Additional pass gates, one per each output line, are controlled by RESET signals (Fig. 2c) and are used to pre-charge $C = C_O + 2NC_{cell} \approx C_O$, where $C_{cell} = 0.2$ fF is drain line capacitance of a single cell, while C_O is external capacitor.

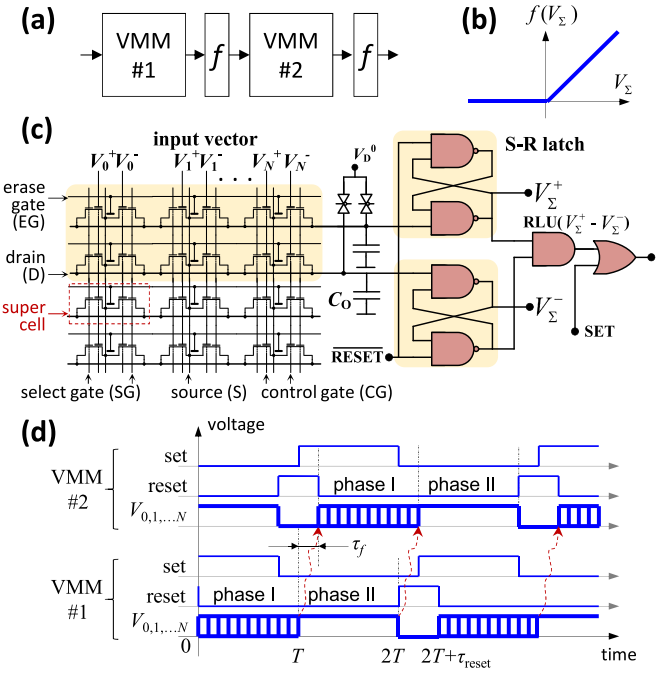

Fig. 2. Two-layer perceptron: (a) Diagram of the circuit with (b) ReLU function, and (c) their implementation details. Panel (c) shows only peripheral circuitry required for 4Q dot-product operation, which involves two rows of adjustable current sources (FG transistors) and two S-R latches, all highlighted with yellow background. (d) Timing diagram for pipelined operation (schematically). τ_f is a combined propagation delay of S-R latch and rectify-linear circuit. The dashed red arrows show the flow of information between two VMMs. Note that the inputs to VMMs are always high during the phase II.

Controlled by SET signal, the output OR gate is used to decouple computations in two adjacent VMMs. Specifically, the OR gate and SET signal generate phase II's T-long pulses applied to the second VMM and, at the same time, pre-charge and start new phase I computation in the first VMM. Using appropriate periodic synchronous SET and RESET signals, pipelined operation with period $2T + \tau_{reset}$ is established, where τ_{reset} is a time needed to pre-charge output capacitor (Fig. 2d).

We have designed two-layer perceptron network, based on two 10×10 4Q multipliers, in 55 nm CMOS process with

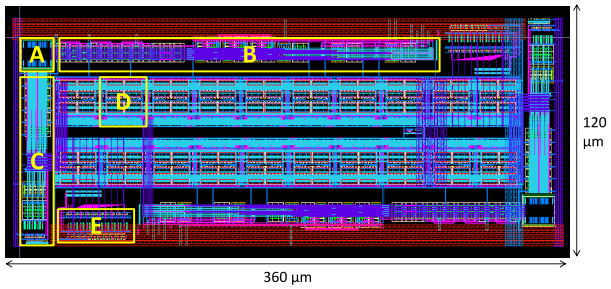

Fig. 3. Two-layer perceptron layout. Label A denotes 10×20 supercell array, B/C shows column/row program and erase circuitry, D is one "neuron" block, which includes $C_O = 0.4$ pF MOSCAP output capacitor, S-R latch based on $W = 120$ nm, $L = 900$ nm transistors, pass gates, and ReLU/pipelining circuit, and E is output multiplexer. Clock/control signals are generated externally.

modified embedded ESF3 NOR flash memory technology [3]. In such technology, erase gate lines in the memory cell matrix were rerouted (Fig. 2c) to enable precise individual tuning of the FG cells' conductances. (The details on the redesigned structure, static and dynamic I-V characteristics, analog retention, and noise of FG transistors, as well as results of high precision tuning experiments can be found in [3].) The network is implemented with two identical 10×20 arrays of supercells (with each supercell hosting two FG transistors), CMOS circuits for the pipelined VMM operation and ReLU transfer function, as well as CMOS circuitry for programming/erasure of the FG cells (Fig. 3). During operation, all FG transistors are biased in subthreshold regime. The input voltages are applied to control gate lines, while the output currents are supplied by the drain lines. Because FG transistors are N-type, VMM is performed by sinking currents via memory cells. In this case, the output lines are pre-charged to $V_D^0 = V_{TH} + \Delta V_D$ with RESET signal (i.e., ΔV_D above the threshold voltage V_{TH} of S-R latch), and then discharged to the ground during computation.

IV. Performance and Tradeoffs Analysis

In this section we discuss important tradeoffs and optimization process for the design in 55 nm technology, focusing on the computing precision. To simply analysis, we always assume (very) conservative $C_O = 100 \times 2NC_{cell}$.

A. Precision

The weight precision is affected by the tuning accuracy and drift of analog memory state. We have shown earlier that at least in small-scale current-mode VMM circuits based on 55-nm NOR flash technology, even without any optimization, these factors combined allow up to 6 bit effective precision for the majority of the weights [3] under wide range of ambient temperatures. We expect that similar to current-mode VMM circuits, the temperature sensitivity will be improved due to differential design and utilization of higher drain currents [3], [4], which is desired for optimal design.

Compute (output) precision p_O can be defined separately from weight precision as

$$p_O = -\log_2(\text{Error}) - 1, \quad \text{Error} = \frac{1}{T} \max_{\Delta_\Sigma} |\Delta_\Sigma^{\text{ideal}} - \Delta_\Sigma|, \quad (2)$$

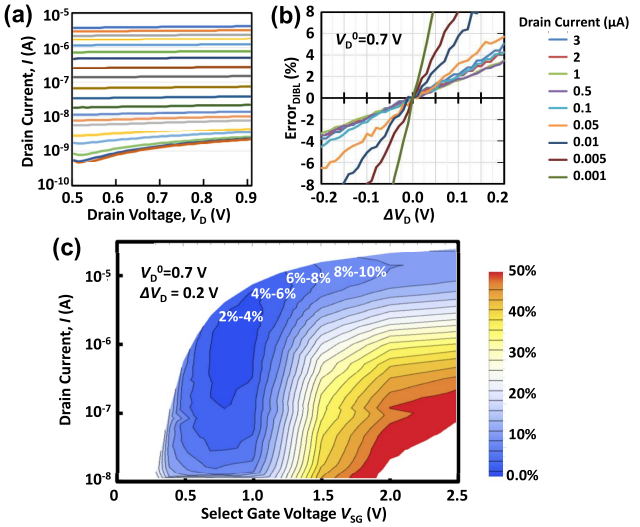

Fig. 4. (a) Measured drain current for different memory states for $V_{SG} = 0.8$ V and (b) corresponding Error_{DIBL}. (c) DIBL error as a function of memory state and V_{SG} obtained from experimental data. In all cases, $V_{CG} = 1.2$ V and $V_S = V_{EG} = 0$ V.

where Error is a maximum absolute difference between the ideal ($\Delta_\Sigma^{\text{ideal}}$) and actual ($\Delta_\Sigma$) output pulse durations, normalized by its maximum value. We found that among many potential factors affecting p_O, the main one is non-negligible dependence of FG transistor subthreshold currents on the drain voltage, due to the drain-induced barrier lowering (DIBL). It is convenient to characterize DIBL error of a single FG transistor as

$$\text{Error}_{DIBL} \approx 1 - I(V_D^0 - \Delta V_D)/I(V_D^0). \quad (3)$$

To minimize DIBL-related error, we have first selected $V_D^0 = 0.7$ V and $\Delta V_D = 0.2$ V, and used $V_{CG} = 1.2$ V, which is a standard CMOS logic voltage in 55 nm process. These choices correspond to quasi optimal operation condition. For example, V_D^0 should be much larger than thermal voltage V_T, due to $I \propto 1 - \exp(-V_D/V_T)$ in subthreshold regime. Also, DIBL error is linearly decreased by reducing ΔV_D, and the latter was chosen to be large enough to have negligible static and short-circuit leakages in CMOS gates (see below).

The experimental measurements have shown that the cell's current is especially sensitive to select gate voltages with the distinct optimum at $V_{SG} \sim 0.8$ V (Fig. 4). The optimum is apparently due to shorter effective channel length for higher V_{SG} values, and hence more severe DIBL, while due to voltage divider effect at lower V_{SG}. Furthermore, the drain dependency is the smallest at higher currents $I_{max} \sim 1$ μA, though further increase in I_{max} is naturally bounded by the upper limit of the subthreshold conduction (Fig. 4a,b). At such optimal conditions, Error_{DIBL} could be less than 2% (Fig. 4c). Note that DIBL error in Fig. 4c is always positive, so that the corresponding (two-sided) Error in Eq. (2) will be twice smaller after appropriately adjusted ideal values.

Similar to previous NVM-based analog computing studies [3], [4], majority of the process variations, a typical concern for any analog circuits, are compensated by adjusting currents of FG cells. Let us first note that sub-threshold slope variations are not important because of digital inputs. Current tuning

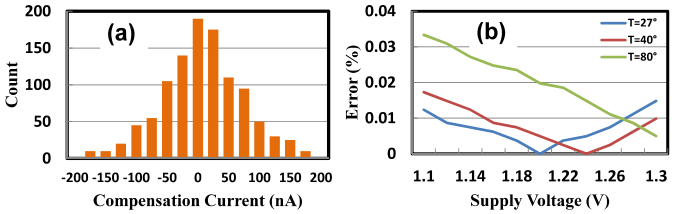

Fig. 5. PVT variation in peripheral CMOS circuitry: (a) A typical distribution, based on 1000 runs, for an additional bias current that should be injected to a drain line to compensate process variations in a 10 × 10 VMM; (b) The relative output error caused by supply voltage variations under various ambient temperatures.

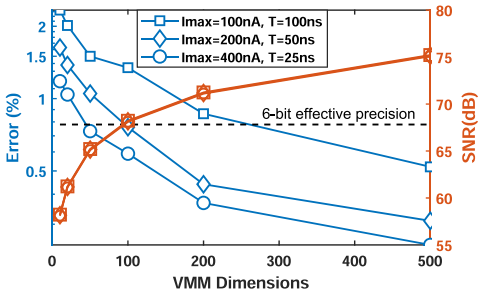

Fig. 6. Output precision simulation results. The left axis corresponds to 99.9 percentile of Error values for 1000 runs of VMM operation with randomly chosen parameters. Specifically, Δ_Σ in Eq. (2) was simulated in SPICE by taking into account DIBL, voltage drop across drain, CG line parasitics, and process variations in cell's drain coupling. For each run, weights and inputs were randomly selected from $[0, T]$ and $[0, I_{\max}]$, respectively, while CG to D line capacitive coupling was randomly varied within ±10% of its mean. Signal to noise ratio (SNR) is calculated from experimental data [4] assuming the worst case scenario, i.e., when all cells supply $I_{\max}$. The corresponding effective precision is roughly $\mathrm{SNR}/6.021 - \log_2 a - 1$, where a is a maximum swing of the noise relative to its rms value. Typical values for a are between 10 and 20 and are determined from the overall system complexity (i.e., number of VMMs), its operation speed, and the required mean-time-to-failure.

would naturally resolve the problem of FG cells' *I-V* variations. Variations in V_{TH}, which can be up to 20 mV rms for the implemented S-R latch according to our Monte Carlo simulations, are also fully compensated by adjusting bias currents (Fig. 5a). Moreover, V_{TH} is always crossed at the same voltage slew rate (in phase II), which reduce variations in S-R latch delay. The simulation results also show very high resilience against variations in the supply voltage and ambient temperature (Fig. 5b) of the peripheral CMOS circuitry, mainly due to its digital design.

Output precision can be also impacted by the FG transistor noise, as well as the factors similar to those of switch-capacitor approach, including leakages via OFF-state FG devices, channel charge injection from pass transistor at the RESET phase, and capacitive coupling of the drain lines. Fortunately, the dominating coupling between D and CG lines is input-independent and can be again compensated by adjusting the weights, and only its variations must be addressed.

Figure 6 shows SNR due to cells' intrinsic noise and output Error from detailed simulations which account for all important non-ideality sources. Specifically, for the studied range of N, Error is decreased by increasing $I_{\max}$ from 100 nA to 400 nA due to smaller DIBL effect for larger currents (Fig. 4). Also, due to averaging out of DIBL, capacitive coupling variations, and noise both -log(Error) and 1/SNR scale roughly as $1/\sqrt{N}$, resulting in higher precision for larger N. The particular scaling for SNR is due to dominant Nyquist noise, whose standard deviation is proportional to $\sqrt{N}$. It should be also noted that for $N > 1000$ non-negligible voltage drop across drain line would decrease output precision.

B. Latency, Energy, and Area

Simulation results show that $T = 25$ ns VMM operation at 6 bit compute precision can be achieved for $N > 50$ when using $I_{\max} = 400$ nA (Fig. 6). The latency can be further decreased by using higher $I_{\max}$ and/or reducing C. As discussed in previous section, the limitation to both approaches are degradation in precision for higher than optimal $I_{\max}$ and also intrinsic parasitics of the array, most importantly CG to drain line capacitive coupling.

The energy per operation is contributed by the dynamic component of charging/discharging of control gate and drain lines as well as external output capacitors, and the static component, including vdd-to-ground and short-circuit leakages in the digital logic. CMOS leakage currents are suppressed exponentially by increasing drain voltage swing, and are further reduced by lowering CMOS transistor currents (i.e., increasing length to width ratio), while still keeping propagation delay τ_{f} negligible as compared to T. The increase in the drain swing, however, have negative impact on VMM precision (Fig. 4b) and dynamic energy. Determining optimal value of ΔV_{D} and by how much CMOS transistor currents can be reduced without negatively impacting precision is important future research. The estimates for the implemented circuit and using parameters discussed in the previous section show that the total energy is about 10 fJ for 10 × 10 VMM, or equivalently 100 TOps/J, with the static energy contributing roughly 65% of the total budget (Fig. 7. The energy-efficiency improves for larger VMMs, e.g., reaching ∼ 150 TOps/J for $N = 1000$, at which point it is completely dominated by dynamic energy related to charging/discharging external capacitor.

The area breakdown by the circuit components was evaluated from the layout in Fig. 3. Because of rather small implemented VMMs, the peripheral circuitry dominates, with one neuron block occupying ∼ 1.5× larger area than the whole memory array. With larger and more practical array sizes ($N = 1000$), the area is completely dominated by the external capacitors and memory array, i.e., ∼ 85% and ∼ 15%, respectively, of the total area (Fig. 7).

V. Discussion and Summary

In some cases, e.g., convolutional layers in deep neural networks, the same matrix of weights is utilized repeatedly to perform large number of multiplications. To increase density, VMM operations are performed using time-division-multiplexing scheme which necessitates storing temporal results and, for our approach, performing conversion between digital and time-domain representations. Fortunately, the conversion circuitry for the proposed VMM is very efficient due to digital time-encoded input/output signals. We have designed such circuitry in which the input conversion is performed with a shared counter and a simple comparator-latch to create time-modulated pulse, while the pulse-encoded outputs are converted to digital signals by using shared counter and a multi-bit register (Fig. 7a). Figure 7b,c summarizes energy

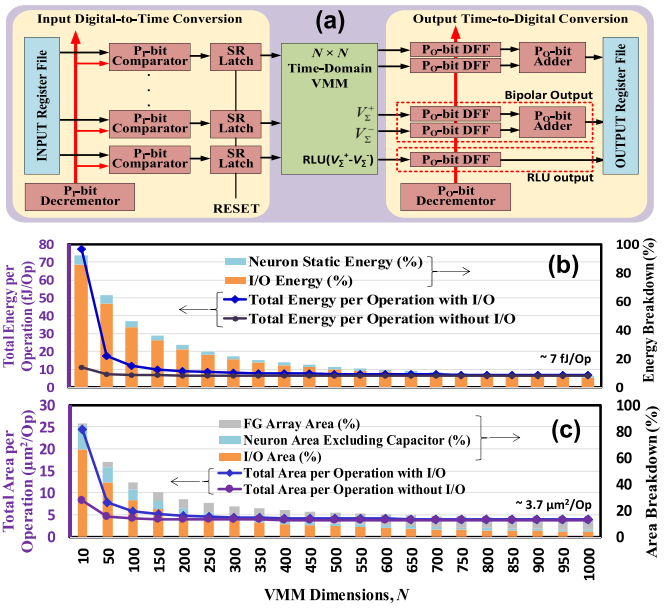

Fig. 7. (a) Time-domain VMM with I/O interface to conventional digital signals. Note that for the considered implementation the falling edge of pulse duration encoded inputs is always aligned with the end of phase I. (b) Energy and (c) area per operation and their breakdowns assuming fixed 6-bit I/O circuitry, and internal time-domain VMM computing precision from Fig. 6. N is incremented by 50, except for the first bar. In panels b / c, the remaining energy / area on the stacked bar chart (right axis) us due to dynamic energy / external capacitor. Circuitry for cells' erasure/programming and testing/characterization of periphery was not included in the area estimates. Based on our prior work, the overhead of such circuitry is rather low, because it can be shared among multiple VMMs [13].

and area for a time-domain multiplier based on the conservative design, in particular showing that the overhead of the I/O conversion circuitry drops quickly and becomes negligible as VMM size increases.

In a more advanced design, the capacitive coupling can be suppressed by adding dummy input lines and using differential input signaling. In this case, the external capacitor can be significantly scaled down or eliminated completely. This, in turn, would lead to almost five-fold increase in density (Fig. 7c), while the latency and energy, limited only by intrinsic parasitics of the memory cell array, can be below 2 ns and 1 fJ per operation, respectively, for 6-bit 1000×1000 VMM. (For such high-precision, high-performance VMMs, generation of the clock, and its distribution for digital I/O circuits, would become a challenging issue, and an important future work.)

In summary, we have proposed novel time-domain approach for performing vector-by-matrix computation and then showed how to chain multiple VMMs completely in a time domain. As a case study, we have designed a simple multilayer perceptron network, which involves two layers of 10×10 four-quadrant vector-by-matrix multipliers, in 55-nm process with embedded NOR flash memory technology. The post-layout estimates for the conservative design, including (excluding) I/O overhead with conventional digital circuits, show up to $> 80(120)$ TOps/J energy efficiency and 25 ns delay at > 6 bit computing precision for 100×100 time-domain VMM, which can be further improved to $> 145(150)$ TOps/J for larger ($N > 500$) arrays. These numbers compare very favourably with previously reported work (Table I). Moreover, there are many reserves in the original work, so that implementation

TABLE I
COMPARISON WITH PREVIOUS WORK. $^*N = 100/500$

Reference	[1]	[2]	[4]	[5]	[6]	This work
Technology	CMOS	ReRAM	NOR flash	CMOS	ReRAM	NOR flash
Approach	current-mode	current-mode	current-mode	switch-cap	time-based	time-based
Process (nm)	180	22	180	40	14	55
Precision (bit)	3	~4	~5	3	<8	6
EE (POps/J)	0.0064	0.06	0.0057	0.008	0.018	0.085/0.135*
I/O included	yes	no	yes	yes	no	yes
Results	sim	sim	exp	exp	sim	sim

of time-domain VMM circuit using more advanced design and more aggressive technology could lead to POps/J-scale energy efficiency. Checking this opportunity, as well as refining trade-off analysis using more optimal operating conditions are important future research directions. Finally, we expect that our approach is also suitable for 3D NAND flash memory technology [14], that could further improve integration density of the proposed time-domain VMM circuits.

VI. ACKNOWLEDGMENT

The authors would like to thank X. Guo and A. Madhavan for useful discussions.

REFERENCES

[1] J. Binas, D. Neil, G. Indiveri, S.-C. Liu, and M. Pfeiffer, "Precise deep neural network computation on imprecise low-power analog hardware," *arXiv:1606.07786*, 2016. [Online]. Available: https://arxiv.org/abs/1606.07786

[2] M. Hu *et al.*, "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Proc. DAC*, Austin, TX, USA, Jun. 2016, pp. 1–6.

[3] X. Guo *et al.*, "Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm NOR flash memory cells," in *Proc. CICC*, Austin, TX, USA, Apr./May 2017, pp. 1–4.

[4] X. Guo *et al.*, "Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology," in *Proc. IEDM*, San Francisco, CA, USA, Dec. 2017, pp. 6.5.1–6.5.4.

[5] E. H. Lee and S. S. Wong, "24.2 A 2.5GHz 7.7TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40nm," in *Proc. ISSCC*, San Francisco, CA, USA, Jan./Feb. 2016, pp. 418–419.

[6] M. J. Marinella *et al.*, "Multiscale co-design analysis of energy, latency, area, and accuracy of a ReRAM analog neural training accelerator," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 86–101, Mar. 2018.

[7] V. Ravinuthula, V. Garg, J. G. Harris, and J. A. B. Fortes, "Time-mode circuits for analog computation," *Int. J. Circuit Theory Appl.*, vol. 37, no. 5, pp. 631–659, Jun. 2009.

[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," *Appl. Phys. Exp.*, vol. 9, no. 3, 2016, Art. no. 034201.

[9] T. Morie *et al.*, "Spike-based time-domain weighted-sum calculation using nanodevices for low power operation," in *Proc. IEEE NANO*, Sendai, Japan, Aug. 2016, pp. 390–392.

[10] D. Miyashita, S. Kousai, T. Suzuki, and J. Deguchi, "A neuromorphic chip optimized for deep learning and CMOS technology with time-domain analog and digital mixed-signal processing," *IEEE J. Solid-State Circuits*, vol. 52, no. 10, pp. 2679–2689, Oct. 2017.

[11] A. Madhavan, T. Sherwood, and D. B. Strukov, "A 4-mm^2 180-nm-CMOS 15-giga-cell-updates-per-second DNA sequence alignment engine based on asynchronous race conditions," in *Proc. CICC*, Austin, TX, USA, Apr./May 2017, pp. 1–4.

[12] M. Bavandpour, M. R. Mahmoodi, and D. B. Strukov, "Energy-efficient time-domain vector-by-matrix multiplier for neurocomputing and beyond," *ArXiv:1711.10673*, Sep. 2017. [Online]. Available: https://arxiv.org/abs/1711.10673

[13] M. Bavandpour *et al.*, "Mixed-signal neuromorphic inference accelerators: Recent results and future prospects," in *Proc. IEDM*, San Francisco, CA, USA, Dec. 2018.

[14] C. M. Compagnoni *et al.*, "Reviewing the evolution of the NAND flash technology," *Proc. IEEE*, vol. 105, no. 9, pp. 1609–1633, Sep. 2017.