Ternary Compute-Enabled Memory using Ferroelectric Transistors for Accelerating Deep Neural Networks

Sandeep Krishna Thirumala, Shubham Jain, Sumeet Kumar Gupta and Anand Raghunathan
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
Email: {sthirum, jain130, guptask, raghunathan}@purdue.edu

Abstract— **Ternary Deep Neural Networks (DNNs), which employ ternary precision for weights and activations, have recently been shown to attain accuracies close to full-precision DNNs, raising interest in their efficient hardware realization. In this work we propose a Non-Volatile Ternary Compute-Enabled memory cell (TeC-Cell) based on ferroelectric transistors (FEFETs) for in-memory computing in the signed ternary regime. In particular, the proposed cell enables storage of ternary weights and employs multi-word-line assertion to perform massively parallel *signed* dot-product computations between ternary weights and ternary inputs. We evaluate the proposed design at the array level and show 72% and 74% higher energy efficiency for multiply-and-accumulate (MAC) operations compared to standard near-memory computing designs based on SRAM and FEFET, respectively. Furthermore, we evaluate the proposed TeC-Cell in an existing ternary in-memory DNN accelerator. Our results show 3.3X-3.4X reduction in system energy and 4.3X-7X improvement in system performance over SRAM and FEFET based near-memory accelerators, across a wide range of DNN benchmarks including both deep convolutional and recurrent neural networks.**

Keywords—Deep Neural Networks, Dot-Product, Ferroelectric Transistors, In-Memory Computing, Low-Precision, Multiply-and-Accumulate, Ternary DNN.

I. Introduction

Deep Neural Networks (DNNs) have gained immense popularity in recent years due to their ability to achieve remarkable accuracies in a wide range of cognitive tasks [1]. However, the high computation and storage demands pose key challenges to their ubiquitous adoption. An important scenario that exemplifies this challenge is low-power inference, wherein DNN models are executed on deeply embedded IoT devices and wearables that have severe energy and area constraints [2].

To deploy DNNs on cost-constrained systems, low-precision is of great interest as it lowers all aspects of energy usage, viz., compute, interconnect, and memory. Recent studies suggest that ternary precision networks are especially promising as they offer accuracy close to full-precision networks and significantly higher than binary networks [3, 4]. Ternary networks drastically reduce the complexity of matrix multiplication which constitutes >90% of DNN computations, thereby facilitating reductions in computation time and energy. In this work, we explore the design of efficient hardware for ternary DNNs.

Traditional CPUs, GPUs and specialized DNN accelerators suffer from frequent memory accesses, limiting their energy efficiency and performance [5]. To address this issue, various works have proposed in-memory computing, wherein computations are performed within the memory array, eliminating the memory access overheads associated with traditional von-Neumann architectures [6-15]. Most existing designs perform in-memory multiplication of binary operands [6, 7, 12], binary activations with ternary weights [13], or target higher-than-ternary precisions for analog vector-matrix multiplication [10, 11]. Recently, a CMOS based ternary in-memory DNN (TiM-DNN) architecture was proposed for *pure* signed ternary computation (ternary inputs *and* weights: '-1', '0', '+1') [9]. Such an approach enables massively parallel *signed ternary vector-matrix multiplications* in a single array access, for efficient realization of ternary DNNs.

Although CMOS-based in-memory computing designs are promising for achieving energy and performance improvements compared to traditional CPU/GPU architectures, they face some major drawbacks. For instance, in 6T SRAMs, coupling of read-write paths may lead to cell disturbances during computations with multi-word-line assertion [14]. Moreover, static leakage due to technology scaling offsets the efficiency gain achieved during in-memory compute operations [16]. Lastly, large bit-cell area limits their on-chip capacity and in-memory computation bandwidth. Emerging non-volatile memories (NVMs) such as spin-transfer-torque magnetic RAM (STT-MRAM), Resistive RAMs (RRAMs) and FEFETs have showcased great potential to replace or complement CMOS based memories by overcoming their drawbacks. FEFETs, in particular, are extremely promising due to their electric-field-driven low-power write operation compared to current-driven write in STT-MRAMs and RRAMs [17]. These desirable properties have driven recent interest towards in-memory computing with NVM [10-12]. However, to the best of our knowledge, ternary in-memory computation using any emerging NVM has not been previously explored.

In this work, we propose a non-volatile ternary compute-enabled memory cell (TeC-Cell) that can perform massively parallel in-memory matrix multiplication in the *signed ternary regime*. The proposed TeC-Cell is designed by utilizing CMOS compatible FEFETs coupled with a judicious selection of input, weight and output encodings, which enable a compact cell design compared previous SRAM-based in-memory computing designs. The key contributions of this work are:

- We propose a ternary compute-enabled NVM cell (TeC-Cell), which can perform scalar multiplication of the stored value (weight) and an external input, where *both the weight and the inputs are signed ternary numbers*.
- Utilizing the TeC-Cell, we design an array (TeC-Array) that performs massively parallel signed ternary dot-products in-memory. We demonstrate that the TeC-Array achieves significant energy-delay benefits compared to near-memory designs with FEFET-based NVM and SRAM.
- Finally, we incorporate the proposed TeC-Array in a ternary DNN accelerator to evaluate its performance and energy benefits across a wide range of state-of-the-art DNN benchmarks including both deep convolutional and recurrent neural networks. We achieve 3.3X-3.4X energy efficiency and 4.3X-7X performance boost compared to SRAM and FEFET-based near-memory DNN accelerator.

978-3-9819263-4-7/DATE20/©2020 EDAA

II. Background

A. Ternary precision Networks

Ternary networks have emerged as an attractive option in the quest for low-precision DNNs. However, the performance and energy efficiency of near-memory accelerators for ternary networks are bottlenecked by the on-chip memory due to the sequential row-by-row access. The closest prior efforts on in-memory computing involve dot product computation of either ternary inputs with binary weights [6] or vice-versa [13]. Although these are attractive design choices to achieve improved energy efficiency, a *pure* ternary network with signed ternary weights and inputs ('-1', '0', '+1') can achieve substantially better accuracy compared to the binary networks [3-4]. Furthermore, techniques presented in [6, 13] can only enable simultaneous activation of a limited numbers of rows due to sensing constraints. This limits the parallelism achieved in vector-matrix multiplication. A recent CMOS-based design, TiM-DNN [9] overcomes such limitations by performing massively parallel in-memory dot product computations in the signed ternary regime.

This work proposes a novel compute-enabled ternary cell (TeC-Cell) using emerging NVM based on FEFETs, featuring non-volatility, higher integration density and near-zero stand-by leakage power compared to SRAMs. Notably, the input, weight and output encodings that we propose here enable in-memory dot product computation of weight and input vectors with the addition of just two more transistors to a pair of FEFET NVM cells [18]. The compact TeC-Cell design enables a higher degree of parallelism for in-memory computing compared to other memories at iso-area (as discussed later in Section V). The built-in non-volatility of TeC-Cell, along with its low power operation can potentially enable energy-efficient realization of DNNs for edge computing devices such as IoT sensors. Note that our design technique is not limited to FEFETs but can also be applied to other memories with separate read-write paths (such as Spin Orbit Torque MRAMs (SOT-MRAMs) [19], eDRAMs [13], *etc*.), to enable ternary in-memory computation.

B. Ferroelectric Transistors (FEFETs) and NVM Designs

FEFETs are promising emerging memory devices which exhibit non-volatility, zero stand-by leakage and excellent CMOS compatibility [17] (Fig. 1(a)). The unique capacitance interactions of the ferroelectric (FE) and the underlying FET channel results in non-volatility built into the transistor. When the polarization stored is positive/negative (+P/-P), then the n-type FEFET is in the low/high resistance state (LRS/HRS; Fig. 1(b)). Due to the non-volatility of polarization in the FE, the resistance state is retained even in the absence of any externally applied voltage bias [17]. Such unique features along with CMOS compatibility of FE materials such as Hafnium Zirconium Oxide (HZO) have made FEFETs promising for non-volatile data storage [18] as well as for neuromorphic computing applications [10-11].

Several variants of FEFET-based NVMs have been proposed earlier. 1T-FEFET NVM in [20] achieves high density but at the cost of reduced write disturb margins. A 2T-FEFET NVM in [18] uses a write access transistor to eliminate write disturbs present in 1T-FEFET NVM. However, such a design requires an unconventional read biasing scheme leading to design overheads. A 3T-FEFET NVM consists of read and write access transistors connected to the FEFET. The 3T design enables isolation of the bit-cell during its access, thereby overcoming the drawbacks of the 1T-FEFET and 2T-FEFET NVM designs [18]. In this work, we utilize 3T FEFET NVM (given their benefits) and add minimal number of transistors (discussed in Section III) in order to achieve ternary storage and computation in the proposed NVM (TeC-Cell). It may be noted, however, that the proposed technique can also be utilized with other FEFET NVMs.

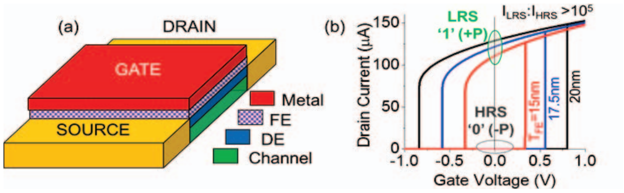


Fig. 1. (a) FEFET device structure (b) Simulated transfer characteristics of FEFET with varying thickness of FE, showing non-volatility at V_{GS}=0V.

Recent studies on FEFETs such as [10-11] have utilized the multi-domain (MD) effects to achieve multi-level analog weights (beyond ternary) for DNN applications as well as compact memory cells. Although they are very attractive, the scalability of MD effects needs further exploration. Therefore, we do not explore this direction and instead focus on realizing ternary in-memory computation for DNNs with two-level FEFETs. Although beyond the scope of this work, we believe the proposed TeC-Cell can utilize the multi-domain effects of FE and potentially achieve richer set of functionalities.

Modeling and Simulation Methodology: We employ a SPICE based circuit-compatible model for FEFETs, where the FE is modeled using time-dependent Landau Khalatnikov (LK) equations, self-consistently coupled with the underlying FET based on a predictive technology model [21]. The LK parameters used in this work are α=-0.7x10^{9}m/F; β=6x10^8 m^5/F/C^2; γ=3x10^{11} m^9/F/C^4; and viscosity coefficient (ρ) =0.025Ω-m [22]. We consider FE thickness in FEFETs (T_{FE}) = 15nm (unless stated otherwise). We design our proposed cell with minimum-sized FEFETs and CMOS FETs for high density. A hysteresis window of ~1V used in this work (Fig. 1(b)) has also been experimentally demonstrated in previous works [10].

III. FEFET Based Ternary Compute-Enabled Memory

A. Ternary Compute-Enabled Memory Cell (TeC-Cell)

To enable ternary in-memory computation, we propose a non-volatile ternary cell (TeC-Cell) which consists of 2 FEFETs and 6 standard FETs. The schematic and layout are shown in Fig. 2. The core of the TeC-Cell involves two 3T-FEFET based memories [18] (for ternary storage), which are cross-coupled with each other using just 2 additional transistors per cell (M_5 and M_6). Transistors M_1 and M_2 are the write access transistors, which enable selective writing of the data in the array as the polarization of the two FEFETs (P_A in M_A and P_B in M_B). M_3 and M_4 are the read access transistors, used to sense the data without disturbances from the unaccessed cells. Cross-coupled transistors M_5 and M_6 (along with M_3 and M_4) enable in-memory ternary scalar multiplication as discussed later. The proposed technique of designing a TeC-Cell can also be employed in other memories with separate read-write paths (such as SOT-MRAMs [19], eDRAMs [13] etc.), using just 2 additional cross-coupled transistors. In contrast, TiM-DNN requires 7 additional transistors for SRAMs [9]. In this paper, we focus our discussion on FEFETs since they demonstrate appealing properties such as non-volatility, near-zero leakage energy and low-power write.

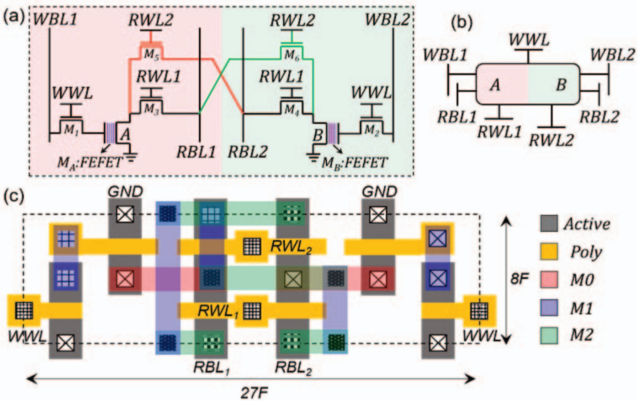


Fig. 2. (a) Circuit design, (b) schematic and (c) layout of ternary compute-enabled non-volatile memory cell (TeC-Cell)

Moreover, their high distinguishability (Fig. 1(b)) is particularly useful for robust in-memory computation, as explained later.

B. Standard Read-Write Operations

For storing ternary data (storage/weight encoding in Fig. 3(b)), we assert the write word-line (WWL=V_{DD}=1V) and drive the write bit-lines (WBL1 and WBL2) to appropriate values. To write '+1' ('-1'), WBL1= V_{DD} (-V_{DD}) and WBL2= -V_{DD} (V_{DD}) is applied. This brings the polarization of the FEFETs to P_A= +P (-P) and P_B= -P (+P). To write '0', WBL1 and WBL2 are driven to -V_{DD}, resulting in P_A=P_B=-P. After write, WWL is de-asserted with WBL1=WBL2=0V, resulting in storage of the bit-information in M_A and M_B as P_A and P_B in a non-volatile fashion. Note that, as mentioned in Section II, polarization stored in the FEFETs corresponds to its resistance states (+P: LRS; -P: HRS), which is used for the read operation as discussed next.

For sensing the bit stored, the read word-line (RWL1) is asserted with the read bit-lines (RBL1 and RBL2) pre-charged to V_{DD}. Now, based on the polarization stored, RBL1 and RBL2 will either discharge or remain at V_{DD}, due to high LRS and low HRS currents, respectively (Fig. 1). For the case when the bit stored is '+1' (P_A=+P; P_B=-P), RBL1 discharges (M_A in LRS), while RBL2 remains at V_{DD} (M_B in HRS). The opposite occurs when the bit stored is '-1' (P_A=-P; P_B=+P). When the TeC-Cell stores a '0' (P_A=-P; P_B=-P), both RBL1 and RBL2 remain at V_{DD}. We use a voltage sense amplifier to compare the RBL1 and RBL2 voltages with a reference voltage (0.95V in our analysis). Note that, during read and write, RWL2 is always de-asserted. Additionally, the proposed TeC-Cell can also be used as a 2-bit binary memory where P_A and P_B correspond to independent bits, without any circuit modifications.

C. In-Memory Ternary Multiplication using TeC-Cell

In this section, we propose in-memory scalar multiplication of ternary weight (stored in the TeC-Cell) with ternary input to obtain a ternary output. Initially, the read bit-lines (RBL1 and RBL2) are pre-charged to V_{DD}. The ternary inputs are encoded as read word-line (RWL1 and RWL2) voltages as shown in Fig. 3(a). Depending on the ternary weight (encoded as P_A and P_B; see Fig. 3(b)), the final RBL1 and RBL2 voltages represent the multiplication output (output encoding in Fig. 3(c)). We explain this further with the following examples:

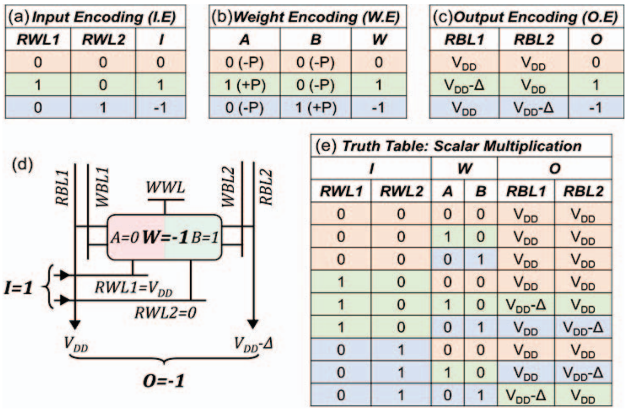
- When input I= +1 (RWL1=V_{DD}; RWL2=0) and weight W= -1 (A=0; B=1), transistors M_3, M_4, M_B are ON and M_5, M_6 and M_A OFF. This condition results in a discharge path for RBL2, resulting in a voltage drop of Δ=100mV (which is sensed with the sense amplifier), while RBL1 remains pre-charged at V_{DD}. This corresponds to output (O=I*W) = -1. Note that the output is inferred with single-ended sensing of RBL1 and RBL2 (see Fig. 3(d)). The same voltage conditions of RBL1 and RBL2 hold true for the case when I= -1 (RWL1=0; RWL2=1) and W= +1 (A=1; B=0) as shown in Fig. 3(d).
- When I= +1 (RWL1 =V_{DD}; RWL2 =0V) and W= +1 (A=1; B=0), transistors M_3, M_4, M_A are ON while M_5, M_6 and M_B remain OFF. This corresponds to a discharge path for RBL1 (resulting in Δ drop) with RBL2 remaining at its pre-charged voltage, V_{DD}. This voltage condition corresponds to O= I*W= +1. This condition also holds true when I= -1 and W= -1.
- When W or I=0, RBL1 and RBL2 remain pre-charged at V_{DD}, corresponding to O= I*W= 0.

The truth table for all permutations is shown in Fig. 3(e). Note that the proposed TeC-Cell exhibits isolation of read-write paths and therefore, in-memory scalar multiplication has no effect on the information stored as polarization in the FEFETs.

D. Ternary Dot Product Computation

We next discuss how TeC-Cells enable in-memory ternary dot product computation for vector-matrix multiplication. This is achieved by simultaneously asserting the read word-lines of TeC-Cells present in a single column as illustrated in Fig. 4(a) [9]. The weight vector with ternary elements W_i is stored in the TeC-Cells, while the input vector with elements I_i is encoded using the voltages of $RWL1_i$ and $RWL2_i$ (Fig. 3(a). With RBL1 and RBL2 (which are pre-charged to V_{DD}) connected to cells in the same column, the scalar products from each TeC-Cell (as discussed in previous sub-section) add up through cumulatively discharge of RBL1 and RBL2, resulting in a multiply-and-accumulate (MAC) operation. The final RBL1 (RBL2) voltages correspond to the number of TeC-Cells producing +1 (-1) as the scalar product. For example, if '*a*' scalar multiplication produced an output of '+1' and '*b*' scalar multiplications produced an output of '-1', then the final RBL1 and RBL2 voltages are V_{DD} - *a*Δ and V_{DD} - *b*Δ respectively. Flash analog-to-digital converters (ADCs) are employed to yield the digital value corresponding to '*a*' and '*b*'. The final dot product given by$\sum_{i=1}^{n} I_i * W_i = a - b$, is achieved by subtracting *'b'* from *'a'* using a digital CMOS subtractor. Fig. 4(b) illustrates the sensing circuit required to realize the final dot product computation.

It is important to mention that the sense margin reduces as *'a'* or *'b'* increase, due to the exponential nature of the bit-line



(a) Input Encoding (I.E)		
RWL1	RWL2	I
0	0	0
1	0	1
0	1	-1

(b) Weight Encoding (W.E)		
A	B	W
0 (-P)	0 (-P)	0
1 (+P)	0 (-P)	1
0 (-P)	1 (+P)	-1

(c) Output Encoding (O.E)		
RBL1	RBL2	O
V_{DD}	V_{DD}	0
V_{DD}-Δ	V_{DD}	1
V_{DD}	V_{DD}-Δ	-1

(d)
RBL1
WBL1
WWL
WBL2
RBL2
A=0 W=-1 B=1
I=1
RWL1=V_{DD}
RWL2=0
V_{DD}
V_{DD}-Δ
O=-1

(e) Truth Table: Scalar Multiplication					
I		W		O	
RWL1	RWL2	A	B	RBL1	RBL2
0	0	0	0	V_{DD}	V_{DD}
0	0	1	0	V_{DD}	V_{DD}
0	0	0	1	V_{DD}	V_{DD}
1	0	0	0	V_{DD}	V_{DD}
1	0	1	0	V_{DD}-Δ	V_{DD}
1	0	0	1	V_{DD}	V_{DD}-Δ
0	1	0	0	V_{DD}	V_{DD}
0	1	1	0	V_{DD}	V_{DD}-Δ
0	1	0	1	V_{DD}-Δ	V_{DD}

Fig. 3. (a) Input, (b) Weight and (c) Output encoding for ternary compute operations. (d) Example of scalar multiplication in TeC-Cell. (e) Truth table for all permutation of I.E and W.E.

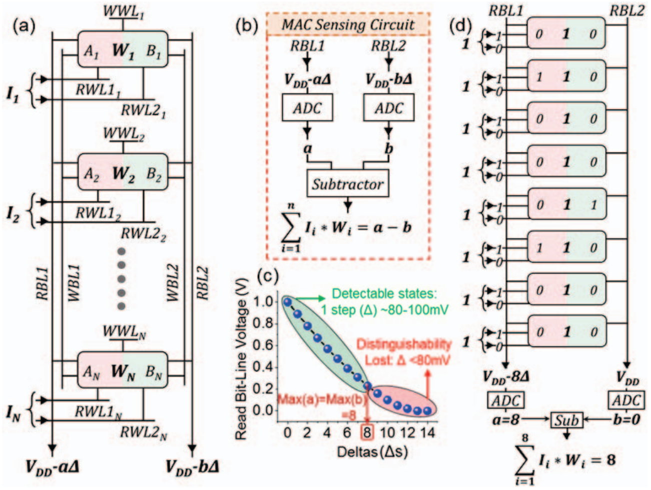


Fig. 4. (a) Ternary dot-product computation of input vector I and weight vector W (b) MAC sensing unit consisting of ADCs and subtractor. (c) RBL voltage vs number of Δ drops. (d) An example of worst-case input-weight scenario for sensing, resulting in *a*=8.

capacitance discharging. For example, if *'a'* or *'b'* increase from 1 to 8, Δ reduces from 100mV to 80mV as shown in Fig. 4(c). This limits the number of cells that can be simultaneously activated during dot-product computation. Fig. 4(d) illustrates an example of a worst-case input-weight vector scenario for a stack of eight TeC-Cells. However, the statistics of the data, specifically the prevalence of xero values in weights and activations, also plays a role in determining the design choice, as discussed in Section IV. Before undertaking this discussion, we first analyze the implications of process variations on the degradation of sense margins in the next sub-section.

E. Variation Analysis

We study the impact of transistor threshold voltage (V_{TH}) variation on the in-memory dot-product operation. We consider 6σ = 120mV [23] for V_{TH} of all the transistors (where σ is the standard deviation). We perform Monte-Carlo SPICE simulations considering 1000 samples each, for cases ranging from 1Δ discharge to 8Δ discharge (states >8Δ are not considered since they are not sufficiently distinguishable). As the amount of discharge increases, the probability of sensing error also increases as shown in Fig. 5(a) (higher overlapping of RBL voltages between adjacent Δ states). However, it is also important to note that the probability of occurrence of the states decrease with increasing discharge values [9] (due to data statistics, as discussed in Section IV). The probability of an error in the dot-product is equal to the product of sensing error probability and the occurrence probability of a particular discharge state (number of Δs; #Δ). Fig. 5(b) illustrates the dot product error probability as a function of #Δ, exhibiting a non-monotonic behavior. Moreover, the total probability of error (P_T) during the dot-product operation is the sum of errors observed for each #Δ (Fig. 5(b)), is 3.10e-3. In other words, for every 1000 MAC operations we have ~3 errors with magnitude ±1 [since only adjacent Δ states overlap, as seen in Fig. 5(a)]. Our system-level evaluations reveal that P_T of 3.10e-3 has negligible impact on accuracy of DNNs, attributed to the low magnitude of errors and resiliency of DNNs to computational errors [24]. Note that FEFETs may encounter variability due to variation in FE parameters such as domain size/distribution [11], whose implications on the proposed ternary computation requires additional study.

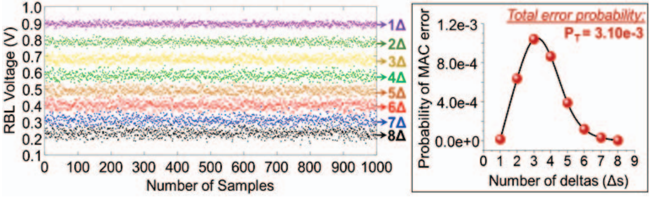


Fig. 5. (a) Variation analysis with 1000 Monte Carlo sample for each state varying from 1Δ to 8Δ. (b) Probability of MAC error with varying Δs.

IV. TeC-Array Design

In this section, we present an array architecture using the proposed TeC-Cells for accelerating ternary DNNs. The TeC-Array can perform massively parallel vector-matrix multiplication (or in-memory dot product computation) between ternary inputs and weights. The maximum number of simultaneously accessed cells in a column is determined by two factors: *(a) Sensing failure:* As discussed in the previous section, increase in *'a'* or *'b'* results in reduced sense margins and higher errors. *(b) Sparsity:* At the same time, the occurrence probability of large *'a'* or *'b'* is also low [9]. This is due to >40% of vector elements being zeros as discussed in [3, 4, 9] (known as sparsity in DNNs). Therefore, considering the above mentioned factors, the optimal number of TeC-Cells which can be accessed simultaneously is *N*=16. It is important to note that, although we can only detect a maximum of 8 states reliably [Fig. 4(c)], we are able to use *N*=16 by harnessing the advantages of sparsity in DNNs [9]. However, having only *N*=16 TeC-Cells in each column of an array may not be practical. Therefore, we designed a blocked 2D array with TeC-Cells, grouped into *M*=16 blocks, with each block containing *K*=256 columns, and each column having *N*=16 rows of TeC-Cells. Thus, the proposed array consists of *N*M*K* TeC-Cells (Fig. 6). We use a block decoder to access the N rows of a block simultaneously. WWLs, RWL1s and RWL2s of TeC-Cells in a row are connected together, while WBL1s, WBL2s, RBL1s and RBL2s of TeC-Cells in the same column are shared. *K* TeC-Cells in a row are accessed together for the read/write operations. On the other hand, in-memory ternary dot product computation is achieved at the block granularity where *K* dot-product operations of vector length *N* are performed in parallel. 3-bit Flash ADCs connected to RBL1 and RBL2 along with a 3-bit subtractor are employed for determining the dot-product (since maximum number of

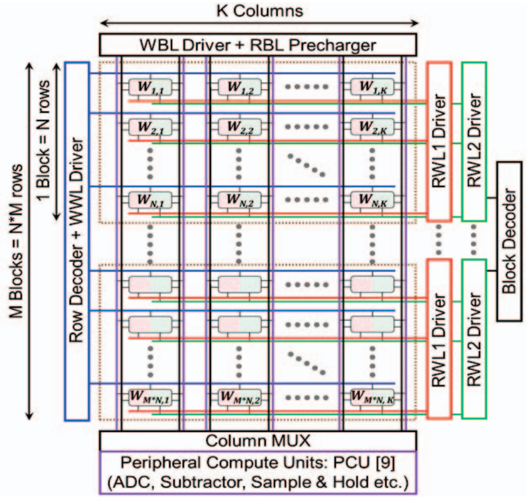


Fig. 6. TeC-Cell array design with N-rows and K-columns in a block and M-blocks in a column.

detectable Δs =8Δ; see Fig. 4(c)). Therefore, in one block access, the array can perform ternary multiplication of input vector I (with *N* elements) and weight matrix W (of size *N*K*).

In order to perform ternary dot products on vector lengths *N*=16, we utilize the technique proposed in [9] of storing partial sums in a peripheral compute unit (PCU) using a sample and hold circuitry. After multiple block accesses (in the same column), we accumulate all the partial sums to determine the final dot products. The dot products are then quantized, and passed through an activation function to derive inputs to the next DNN layer. Moreover, as discussed in [9] we utilize *L*=32 PCUs for the entire array (where *L*<*K*=256) in order to amortize area energy overheads of the peripheral circuits.

V. Results

A. Array-Level Analysis

In this sub-section, we compare the write, read and MAC performance and energy of the proposed TeC-Array with respect to two baselines: 6T-SRAM and 3T-FEFET NVM. We design near-memory ternary accelerators for the baselines, where the accelerators access scratchpad memories row-by-row before performing vector-matrix multiplication. We note that the gains shown for our design are pessimistic as we do not include the energy and latency of the processing elements in the near-memory compute baselines. All the memory arrays are designed with the same capacity (=128Kb).

(i) Layout Area (Fig. 7(a)): The proposed TeC-Cell exhibits 33% lower area compared to two 6T-SRAM cells (which can store a ternary bit) due to 4 less transistors. With respect to two 3T-FEFET-NVM cells, the proposed TeC-Cell exhibits 34% higher area attributed to the additional M_5 and M_6 transistors (Fig. 2) that are added to enable ternary in-memory computation. Note that, although two 6T-SRAM or 3T-FEFET cells can store a ternary weight, they do not support in-memory ternary compute offered by the proposed TeC-Cells.

(ii) MAC Operation (Fig. 7(b)): The major advantage of the proposed TeC-Array is massively parallel in-memory computation of ternary dot-products. This results in 91% and 89% higher performance for the TeC-Array in comparison with the SRAM and 3T-FEFET NVM array baselines. At the same time, the MAC operation using TeC-Arrays exhibits 72% and 74% improved energy efficiency compared to SRAM and 3T-FEFET NVM, respectively. This is attributed to the simultaneous assertion of multiple-word-lines unlike the near-memory compute baselines which require row-by-row access. For DNNs, the predominant contributor to energy/delay is the MAC operation. Hence, the energy savings achieved at array-level are expected to translate to system-level energy efficiency, as discussed subsequently.

(iii) Read/Write Operations (Fig. 7(c, d)): The enablement of ternary in-memory computation in the proposed TeC-Cells comes at the cost of some overhead for the read/write operations. Compared to 3T FEFET-NVM, we observe 19%, 12%, 19% and almost similar read delay, write delay, read energy and write energy, respectively, for the proposed TeC-Cells. This is mainly attributed to the larger cell area and additional BL capacitances due to the drain capacitances of M_5 and M_6 (Fig. 2). When compared to SRAM, we observe similar trends with one exception in read delay which is 7% lower. This is due to lower WWL capacitance in TeC-Cell (due to smaller area). Note that write energy of FEFET memories is ~2X compared to SRAM, mainly due to the overheads associated with negative voltages needed for polarization switching (Fig. 1).

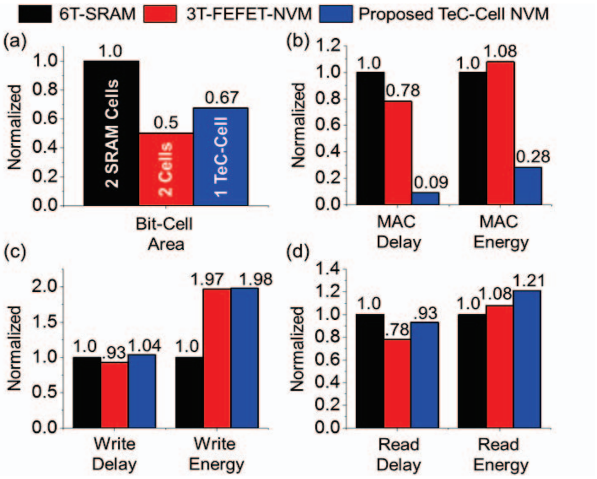


Fig. 7. (a) Cell layout area and normalized energy-delay metrics for (b) MAC, (c) Write and (d) Read operations for TeC-Cell with in-memory computation, FEFET-NVM and SRAM with near memory computation.

It is important to note that in DNN applications, more than 90% of operations are MACs. Therefore, even in the presence overheads in standard read and write operations, the total system performance and energy is drastically improved for ternary DNNs implemented using TeC-Arrays, as discussed next.

B. System Evaluation

(i) Simulation Framework: In this sub-section, we evaluate the system-level performance/energy efficiency of TeC-Cells. To that end, we utilize the TiM DNN accelerator architecture proposed in [9] and design an TeC-Cell based system (TeC-System) with 32 TeC-Arrays (256x256). We compare the TeC-System with near-memory DNN accelerators to quantify the system-level benefits due to in-memory operations enabled by the proposed TeC-Cell. The baseline accelerators are designed using memories with near-memory computation units to execute ternary dot-products. (Note, baseline memories considered here cannot perform in-memory computation). We use two memory technologies SRAM and FEFET, and design two types of baseline systems: (i) iso-area and (ii) iso-weight storage capacity (2 Mega ternary words) as the TeC-System. TeC-Arrays (256x256) are 0.89X smaller than 6T SRAM arrays (256x512) and 1.5X larger than FEFET arrays (256x512) (including the overheads of peripherals). Therefore, the SRAM based iso-area design uses 28 arrays and the FEFET based iso-area baseline utilizes 48 arrays. We use an in-house architectural simulator to obtain the energy/performance of the TeC-System compared to the baselines using a suite of DNN benchmarks [9].

(ii) Performance benefits: Fig. 8(a) shows the normalized execution time for various DNN benchmarks executed on the baseline and the proposed designs. We also show the breakdown of the execution time into two components – TMAC-Ops (Ternary vector-matrix multiplication operations) and Non-TMAC-Ops (other DNN operations). On average, we achieve 7X and 6.3X speedup over SRAM based iso-capacity and iso-area baselines, respectively, and 6.1X and 4.3X speedup over FEFET-based iso-capacity and iso-area baselines, respectively. Across our baselines, the FEFET-based iso-area design achieves the best performance due to the higher-level of parallelism available from the extra 16 arrays. For the proposed design, the

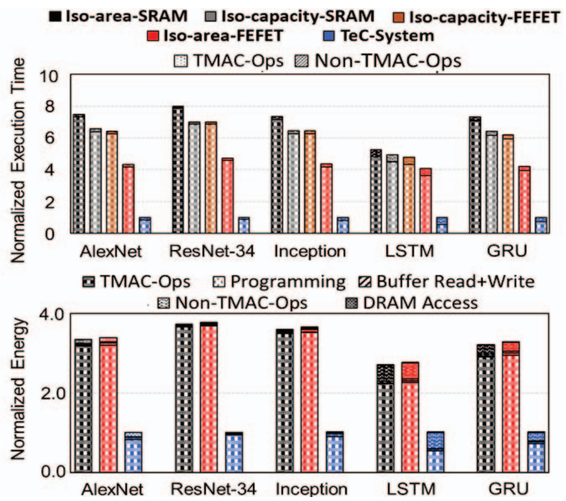


Fig. 8. (a) Normalized execution time and (b) Normalized energy consumption of the proposed TeC-System with respect to iso-capacity and iso-area baselines using SRAM and FEFET NVM based near-memory compute architectures, for a suite of DNN benchmarks.

performance benefits arise due to ternary in-memory operations in TeC-Arrays, wherein we activate and compute on 16 memory rows simultaneously. The application-level speedup depends on the fraction of the execution time spent on TMAC-ops, and therefore, benchmark applications with higher TMAC-Ops/Non-TMAC-Ops ratio achieve higher speedups.

(iii) Energy benefits: Next, we present the system-level energy benefits of the TeC-System over the iso-area SRAM and FEFET baselines. Note that, the iso-capacity baselines will exhibit similar energy consumption as iso-area baselines because, the total system energy consumption depends on the number of TMAC-Ops and Non-TMAC-Ops, which remains constant for an iso-capacity or iso-area baseline. Fig. 8(b) shows that the major components of energy consumption are TMAC-Ops, programming (writing weights into arrays), DRAM accesses, buffer reads and writes, and Non-TMAC-Ops. On an average, we achieve 3.3X and 3.4X reduction in the application-level energy over the SRAM and FEFET baselines, respectively. Across our benchmark applications, the factors indicating higher speedup are also predictive of higher energy savings, i.e., larger fraction of TMAC-Ops leads to superior energy benefits. This is because the proposed TeC-Array utilizes massively parallel in-memory TMAC-Ops which are more energy efficient than near-memory computing baselines (Fig. 7). We also observe that the FEFET-iso-area baseline consumes slightly more energy than the SRAM-iso-area design, due to high write energy of FEFETs.

Table. 1 shows the comparison of the proposed architecture with other state-of-the-art approaches. With respect to TiM-DNN [9], we achieve ~2X improvement in TOPS/W and TOPS/mm^2 due to TeC-Cell's compact layout footprint. With respect to experimental findings in XNORBIN [25] and Tesla V100 [5], which are traditional computing architectures (not in-memory), we observe 2.7X-607X and 35X-813X improvements in TOPS/W and TOPS/mm^2, respectively.

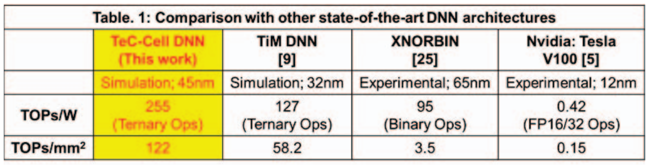


Table. 1: Comparison with other state-of-the-art DNN architectures

	TeC-Cell DNN (This work)	TiM DNN [9]	XNORBIN [25]	Nvidia: Tesla V100 [5]
	Simulation; 45nm	Simulation; 32nm	Experimental; 65nm	Experimental; 12nm
TOPs/W	255 (Ternary Ops)	127 (Ternary Ops)	95 (Binary Ops)	0.42 (FP16/32 Ops)
TOPs/mm^2	122	58.2	3.5	0.15

VI. Conclusion

In this work, we propose a compact, non-volatile, ternary compute-enable memory cell (TeC-Cell) based on ferroelectric transistors. Our design enables dot-product computation in the *signed* ternary regime with the addition of just two transistors to a pair of FEFET based NVM cells. We design a memory array using TeC-Cells to enable massively parallel in-memory ternary computations. The influence of process variations on the dot-product operations are also analyzed. Our array and system-level evaluations showed that the proposed designs achieve up to 7X and 3.4X improvement in performance and energy efficiency compared to near-memory computing architectures.

Acknowledgement

This research was supported by C-BRIC, one of the six centers in JUMP, an SRC program (#40001859).

References

[1] https://wired.com/2016/11/google-facebookmicrosoft-remaking-around-ai/ . *Online*. Accessed Sept. 17, 2017.

[2] S. Venkataramani, et al., "Efficient embedded learning for IoT devices" *In Proc. ASP-DAC*, pages 308–311, Jan 2016.

[3] A. K. Mishra et al., "WRPN: Wide reduced-precision networks" *CoRR, abs/1709.01134*, 2017.

[4] P. Wang et al., "Hitnet: Hybrid ternary recurrent neural network", *In Advances in Neural Information Processing Systems*, 31, 2018.

[5] NVIDIA Tesla V100 Tensor Core GPU. https://www.nvidia.com/enus/data-center/tesla-v100/. *Online*. Accessed March 15, 2019

[6] Z. Jiang, et al., "XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks", *Symp. on VLSI Tech.*, 2018.

[7] M. Rastegari, et al., "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks", *CoRR, abs/1603.05279*, 2016.

[8] X. Sun et al., "XNOR-RRAM: A scalable and parallel resistive synaptic architecture for binary neural networks", *DATE,* 2018.

[9] S. Jain et al., "TiM-DNN: Ternary in Memory accelerator for Deep Neural Networks", *arXiv:1909.06892*, 2019.

[10] M. Jerry et al., "Ferroelectric FET analog synapse for acceleration of deep neural network training," *in IEDM Tech. Dig*., 2017.

[11] H. Mulaosmanovic et al., "Novel ferroelectric FET based synapse for neuromorphic systems", *Symp. on VLSI Technology*, 2017.

[12] T. Tang, et al., "Binary convolutional neural network on RRAM," *IEEE/ACM Asia South Pacific Design Autom. Conf. (ASPDAC)*, 2017.

[13] T. Yoo et al., "A Logic Compatible 4T Dual Embedded DRAM Array for In-Memory Computation of Deep Neural Networks" *ISLPED*, 2019.

[14] S. Jeloka, et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) using Push-Rule 6T Bit Cell Enabling Logic-in-Memory" *IEEE Journal of Solid-State Circuits. (JSSC)*, 2016.

[15] J. Zhang, et al., "In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array", *IEEE JSSC*, 2017.

[16] A. Agarwal, et al., "DRG-cache: a data retention gated- ground cache for low power," *in Proc. of 39th Design Automation Conf.,* 2002.

[17] A. Aziz, et al., "Computing with ferroelectric FETs: Devices, models, systems, and applications," *Dsgn. Auto. And Test. In Europe Conf.,* 2018.

[18] S. K. Gupta, D. Wang, S. George, A. Aziz, X. Li, S. Dutta and V. Narayanan, "Harnessing ferroelectrics for non-volatile memories and logic", *Int. Symp. On Qual. Elec. Desgn.*, 2017.

[19] Y. Seo et al., "Spin-Hall magnetic random-access memory with dual read/write ports for on-chip caches," *IEEE Magn. Lett*., vol. 6, 2015.

[20] A. Sharma, et al., "1T Non-Volatile Memory Design Using Sub-10nm Ferroelectric FETs" *IEEE Elec. Device Lett*., 39 (3), 2018.

[21] A. Aziz, et al., "Physics-Based Circuit- Compatible SPICE Model for Ferroelectric Transistors", *IEEE Elec. Dev. Let.,* 37(6), 2016.

[22] S. Thirumala, et al., "Reconfigurable Ferroelectric Transistor – Part I: Device Design and Operation", *Trans. on Elec. Dev.,* 66(6), 2019.

[23] K. Agarwal, et al., "Characterizing Process Variation in Nanometer CMOS", *ACM/IEEE Design Automation Conference*, 2007.

[24] G. Li et al., "Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications", *Int. Conf. for High Performance Comp. , Networking, Storage and Analysis*, 2017.

[25] A. Bahou, et al., "XNORBIN: A 95 TOP/s/W Hardware Accelerator for Binary Convolutional Neural Networks" *arXiv:1803.058.*, 2018.