TIMELY: Pushing Data Movements and Interfaces in PIM Accelerators Towards Local and in Time Domain

Weitao Li[1,3], Pengfei Xu[1], Yang Zhao[1], Haitong Li[2], Yuan Xie[3], Yingyan Lin[1]
[1]Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA
[2]Department of Electrical Engineering, Stanford University, Stanford, CA, USA
[3] Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, USA
[1]{weitaoli,px5,zy34,yingyan.lin}@rice.edu
[2]{haitongl}@stanford.edu [3]{weitaoli,yuanxie}@ucsb.edu

Abstract*—Resistive-random-access-memory (ReRAM) based processing-in-memory (R^2PIM) accelerators show promise in bridging the gap between Internet of Thing devices' constrained resources and Convolutional/Deep Neural Networks' (CNNs/DNNs') prohibitive energy cost. Specifically, R^2PIM accelerators enhance energy efficiency by eliminating the cost of weight movements and improving the computational density through ReRAM's high density. However, the energy efficiency is still limited by the dominant energy cost of input and partial sum (Psum) movements and the cost of digital-to-analog (D/A) and analog-to-digital (A/D) interfaces. In this work, we identify three energy-saving opportunities in R^2PIM accelerators: analog data locality, time-domain interfacing, and input access reduction, and propose an innovative R^2PIM accelerator called TIMELY, with three key contributions: (1) TIMELY adopts analog local buffers (ALBs) within ReRAM crossbars to greatly enhance the data locality, minimizing the energy overheads of both input and Psum movements; (2) TIMELY largely reduces the energy of each single D/A (and A/D) conversion and the total number of conversions by using time-domain interfaces (TDIs) and the employed ALBs, respectively; (3) we develop an only-once input read (O^2IR) mapping method to further decrease the energy of input accesses and the number of D/A conversions. The evaluation with more than 10 CNN/DNN models and various chip configurations shows that, TIMELY outperforms the baseline R^2PIM accelerator, PRIME, by one order of magnitude in energy efficiency while maintaining better computational density (up to 31.2×) and throughput (up to 736.6×). Furthermore, comprehensive studies are performed to evaluate the effectiveness of the proposed ALB, TDI, and O^2IR in terms of energy savings and area reduction.

***Index Terms*—processing in memory, analog processing, resistive-random-access-memory (ReRAM), neural networks**

I. Introduction

While deep learning-powered Internet of Things (IoT) devices promise to revolutionize the way we live and work by enhancing our ability to recognize, analyze, and classify the world around us, this revolution has yet to be unleashed. IoT devices – such as smart phones, smart sensors, and drones – have limited energy and computation resources since they are battery-powered and have a small form factor. On the other hand, high-performance Convolutional/Deep Neural Networks (CNNs/DNNs) come at a cost of prohibitive energy consumption [68] and can have hundreds of layers [67] and tens of millions of parameters [50], [72]. Therefore, CNN/DNN-based applications can drain the battery of an IoT device very quickly if executed frequently [76], and requires an increase in form factor for storing and executing CNNs/DNNs [11], [58]. The situation continues to worsen due to the fact that CNNs/DNNs are becoming increasingly complex as they are designed to solve more diverse and bigger tasks [32].

To close the gap between the constrained resources of IoT devices and the growing complexity of CNNs/DNNs, many energy-efficient accelerators have been proposed [1], [7], [10], [13], [44]–[46], [73]. As the energy cost of CNN/DNN accelerators is dominated by memory accesses of inputs, weights and partial sums (Psums) (see Fig. 1 (a)) (e.g., up to 95% in DianNao [13]), processing-in-memory (PIM) accelerators have emerged as a promising solution in which the computation is moved into the memory arrays and weight movements are eliminated (see Fig. 1 (b)). Among PIM accelerators on various memory technologies [14], [42], [43], [56], [58], [62], [66], [78], resistive-random-access-memory-(ReRAM)-based-PIM (R^2PIM) accelerators have gained extensive research interest due to ReRAM's high density (e.g. 25×–50× higher over SRAM [71], [79]). However, the energy efficiency of R^2PIM accelerators (such as PRIME [14], ISAAC [58], and PipeLayer [62]) is still limited due to two bottlenecks (see Fig. 1 (b)): *(1) although the weights are kept stationary in memory, the energy cost of data movements due to inputs and Psums is still large (as high as 83% in PRIME [14]); (2) the energy of the interfacing circuits (such as analog-to-digital converters (ADCs)/digital-to-analog converters (DACs)) is another limiting factor (as high as 61% in ISAAC [58]).*

To address the aforementioned energy bottlenecks, we analyze and identify opportunities for greatly enhancing the energy efficiency of R^2PIM accelerators (see Section III-A), and develop three novel techniques that strive to push data

This work was supported in part by NIH R01HL144683 and NSF 1838873, 1816833, 1719160, 1725447, 1730309.

978-1-7281-4661-4/20/$31.00 ©2020 IEEE
DOI 10.1109/ISCA45697.2020.00073

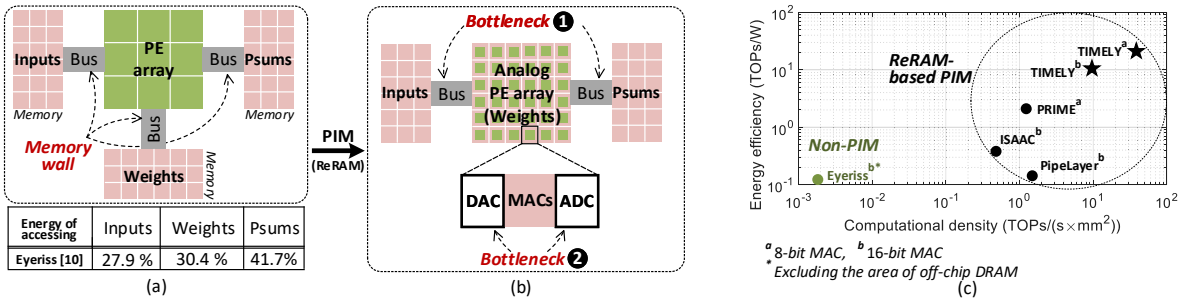


Fig. 1. An illustration of (a) the "memory wall" in CNN/DNN accelerators due to data movements of inputs, weights, and Psums, and an example of their energy breakdown [10], (b) the energy efficiency bottlenecks of PIM accelerators: (1) input and Psum movements (i.e. Bottleneck ❶) and (2) the DAC/ADC interfacing (i.e. Bottleneck ❷), and (c) bench-marking the energy efficiency and computational density of the proposed TIMELY over state-of-the-art CNN/DNN accelerators, including a non-PIM accelerator (Eyeriss [10]) and R^{2}PIM accelerators (PRIME [14], ISAAC [58], and PipeLayer [62]).

movements and interfaces in PIM accelerators towards local and in time domain (see Section III-B). While these three techniques are in general effective for enhancing the energy efficiency of PIM accelerators, we evaluate them in a R^{2}PIM accelerator, and demonstrate an improvement of energy efficiency by one order of magnitude over state-of-the-art R^{2}PIM accelerators. The contribution of this paper is as follows:

- We propose three new ideas for aggressively improving energy efficiency of R^{2}PIM accelerators: (1) adopting analog local buffers (ALBs) within memory crossbars for enhancing (analog) data locality, (2) time-domain interfaces (TDIs) to reduce energy cost of single digital-to-analog (D/A) (and analog-to-digital (A/D)) conversion, and (3) a new mapping method called only-once input read (O^{2}IR) to further save the number of input/Psum accesses and D/A conversions.
- We develop an innovative R^{2}PIM architecture (see Section IV), **TIMELY** (**T**ime-domain, **I**n-**M**emory **E**xecution, **L**ocalit**Y**), that integrates the three aforementioned ideas to (1) maximize (analog) data locality via ALBs and O^{2}IR and (2) minimize the D/A (and A/D) interfaces' energy cost by making use of the more energy-efficient TDIs, the ALBs and the O^{2}IR method. TIMELY outperforms the most competitive R^{2}PIM accelerators in both energy efficiency (over PRIME) and computational density (over PipeLayer) (see Fig. 1 (c)).
- We perform a thorough evaluation of TIMELY against 4 state-of-the-art R^{2}PIM accelerators on >10 CNN and DNN models under various chip configurations, and show that TIMELY achieves up to 18.2× improvement (over ISAAC) in energy efficiency, 31.2× improvement (over PRIME) in computational density, and 736.6× in throughput (over PRIME), demonstrating a promising architecture for accelerating CNNs and DNNs. Furthermore, we perform ablation studies to evaluate the effectiveness of each TIMELY's feature (i.e., ALB, TDI, and O^{2}IR) in reducing energy and area costs, and demonstrate that TIMELY's innovative ideas can be generalized to other R^{2}PIM accelerators.

II. Background

This section provides the background of R^{2}PIM CNN/DNN accelerators. First, we introduce CNNs and the input reuse opportunities in CNNs' convolutional (CONV) operations in Section II-A, and ReRAM basics in Section II-B. Second, we compare digital-to-time converter (DTC)/time-to-digital converter (TDC) and DAC/ADC, which are two types of digital-to-analog (D/A) and analog-to-digital (A/D) conversion, in terms of energy costs and accuracy in Section II-C.

A. CNN and Input Reuse

CNNs are composed of multiple CONV layers. Given the CNN parameters in Table I, the computation in a CONV layer can be described as:

$$O[v][u][x][y] = \sum_{k=0}^{C-1}\sum_{i=0}^{G-1}\sum_{j=0}^{Z-1} I[v][k][Sx+i][Sy+j] \times W[u][k][i][j] + B[u], \quad 0 \le v < M, 0 \le u < D, 0 \le x < F, 0 \le y < E \tag{1}$$

where O, I, W, and B denote matrices of the output feature maps, input feature maps, filters, and biases, respectively. Fully-connected (FC) layers are typically behind CONV layers. Different from CONV layers, the filters of FC layers are of the same size as the input feature maps [10]. Equation (1) can describe FC layers with additional constraints, i.e., $Z = H$, $G = W$, $S = 1$, and $E = F = 1$.

Three types of input reuses exist in CNN CONV operations yielding 3-D Psums. Consider the example in Fig. 2 where C and M are set to 1 for simplicity because input reuses are independent on them. First (see Fig. 2 (a)), one input feature map is shared by multiple (e.g. two in Fig. 2) output channels' filters. Second (see Fig. 2 (b)), as filters slide horizontally, input pixels are reused to generate outputs in the same row – e.g. b and f are used twice to generate w and x, respectively. Third (see Fig. 2 (c)), as filters slide vertically, input pixels are reused to generate outputs in the same column – e.g. e and f are used twice to generate w and y, respectively. Given a layer with D output channels, a filter size of $Z \times G$, and a stride of S, each input pixel is reused DZG/S^2 times [74]. For example, f is reused 8 times in Fig. 2 where D=2, Z=G=2, and S=1. Note that weights are private in DNN CONV operations, which is the main difference between CNNs and DNNs [82].

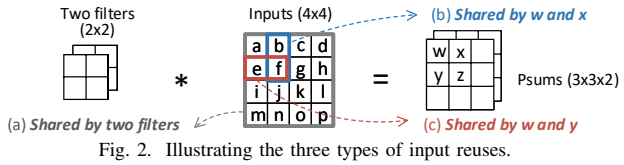


Fig. 2. Illustrating the three types of input reuses.

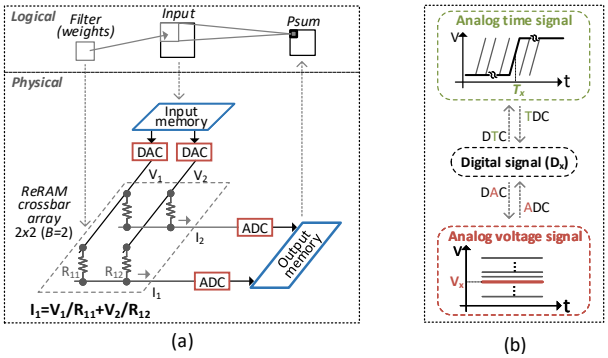


Fig. 3. (a) ReRAM operation basics and (b) two types of interfacing circuits.

B. ReRAM Basics

ReRAM is a type of nonvolatile memory storing data through resistance modulation [30], [39], [65], [69], [71]. An ReRAM cell with a metal-insulator-metal (MIM) structure consists of top/bottom electrodes and a metal-oxide layer [71]. Analog multiplication can be performed in ReRAM cells (see Fig. 3 (a)), with the biased voltages serving as inputs, ReRAM cells' conductance as weights, and resulting currents as outputs. Addition operations are realized through current summing among ReRAM cells of the same columns [28], [77] – e.g. $I_1 = V_1/R_{11} + V_2/R_{12}$ in Fig. 3 (a). At the circuit level, digital inputs are read from an input memory, converted to analog voltages by DACs, and then applied on ReRAM cells. The resulting analog Psums are converted to digital values by ADCs, and then stored back into an output memory.

C. DTCs/TDCs vs. DACs/ADCs

As shown in Fig. 3 (b), DTCs/TDCs can perform the conversion between an analog time signal and the corresponding digital signal; DACs/ADCs can do so between an analog voltage signal and the digital signal. One digital signal (e.g. D_x in Fig. 3 (b)) can be represented as a time delay with a fixed high/low voltage (corresponding to 1/0) in the time domain (e.g. T_x in Fig. 3 (b)) [3], [5], [8], [12], or as a voltage in the voltage domain (e.g. V_x in Fig. 3 (b)). Compared with a DTC/TDC which can be implemented using digital circuits [4], [16], [40], [51], [52], [80], a DAC/ADC typically relies on analog circuits that (1) are more power consuming and (2) vulnerable to noises and process, voltage and temperature (PVT) variations, and (3) benefit much less from process scaling in energy efficiency [49].

TABLE I
A SUMMARY OF PARAMETERS USED IN TIMELY

CNN Params	Description
M	batch size of 3-D feature maps
C/D	input / output channel
H/W	input feature map height / width
Z/G	filter height / width
S	stride
E/F	output feature map height / width
Arch. Params	**Description**
B	# of ReRAM bit cells in one crossbar array is B^2
N_{CB}	# of ReRAM crossbar arrays in one sub-Chip is N_{CB}^2
R_{ij}	the resistance of the ReRAM bit cell at the i^{th} row and j^{th} column of a crossbar array
T_i	the time input for the i^{th} row of an ReRAM crossbar array
$T_{o,8b/4b}$	the time Psum for 8-bit inputs and 4-bit weights
VDD	the logic high voltage of the time-domain signals
V_{th}	the threshold voltage of a comparator
C_c	the charging capacitance
T_{del}	the unit delay of a DTC/TDC
γ	one DTC/TDC is shared by γ rows/columns in one ReRAM crossbar array
ϕ	the reset phase of a sub-Chip (reset: ϕ=1)
χ	the number of sub-Chips in one TIMELY chip
ε	the potential error of one X-subBuf
Energy Params	**Description**
e_{DTC}	the energy of one conversion in DTC
e_{TDC}	the energy of one conversion in TDC
e_{DAC}	the energy of one conversion in DAC
e_{ADC}	the energy of one conversion in ADC
e_P	the unit energy of accessing P-subBuf
e_X	the unit energy of accessing X-subBuf
e_{R^2}	the unit energy of accessing ReRAM input/output buffers

III. Opportunities and Innovations

This section aims to answer the question of **"how can TIMELY outperform state-of-the-art R^{2}PIM accelerators?"** Note that all parameters used in this section are summarized in Table I.

A. Opportunities

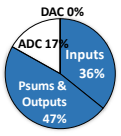
We first identify three opportunities for greatly reducing energy costs of R^{2}PIM accelerators by analyzing performance limitations in state-of-the-art designs. Specifically, Opportunity #1 is motivated by the energy bottleneck of (1) input and Psum movements (i.e., Bottleneck ❶ in Fig. 1 (b)) and (2) interfacing circuits (i.e., Bottleneck ❷ in Fig. 1 (b)); Opportunity #2 is inspired by the bottleneck of interfacing circuits; and Opportunity #3 is motivated by both types of bottlenecks.

Opportunity #1. Enhancing (analog) data locality to greatly reduce the energy/time costs of both data movements and D/A and A/D interfaces. We identify this opportunity based on the following considerations. Since in-ReRAM processing computes in the analog domain, the operands, including inputs, weights, and Psums, are all analog. If we can mostly access analog operands locally, we can expect large energy savings associated with input and Psum movements and largely remove the need to activate D/A and A/D interfaces. In the prior works, the input/Psum movements and interfaces dominate the energy cost of R^{2}PIM accelerators. First, input and Psum accesses involve energy-hungry data movements. While weights stay stationary in R^{2}PIM accelerators, input and Psum accesses are still needed. Although one input/Psum access can be shared by B ReRAM cells in the same row/column, for dot-product operations in a $B \times B$ ReRAM crossbar array, a large number of input and Psum accesses are still required. For example, more than 55 million inputs and 15 million Psums need to be accessed during the VGG-D [61] and ResNet-50 [26] inferences, respectively (see Fig. 4 (a)). While inputs require

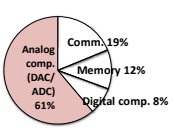
# of accessing	Input*	Psum*
VGG-D	59.9 M	59.8 M
ResNet-50	55.6 M	15.1 M

* All CONV layers

(a)



(b)



(c)

Fig. 4. (a) The number of input/Psum accesses, (b) energy breakdown of PRIME [14], and (c) energy breakdown of ISAAC [58].

only memory read, Psums involve both memory write and read, resulting in a large energy cost. As an example, 36% and 47% of the total energy in PRIME [14] are spent on input and Psum accesses, respectively (see Fig. 4 (b)). Second, voltage-domain D/A and A/D conversions involve a large energy cost. For example, in PRIME, except the data movement energy, most of the remaining energy cost is consumed by D/A and A/D conversions (see Fig. 4 (b)).

Opportunity #2. Time-domain interfacing can reduce the energy cost of a single D/A (and A/D) conversion. Since time-domain D/A and A/D conversion is more energy efficient than voltage-domain conversion (see Section II-C), we have an opportunity to use DTCs and TDCs for interfacing between the digital signals stored in memory and analog signals computated in ReRAM crossbar arrays. In prior works, DACs and ADCs limit the energy efficiency of R^{2}PIM accelerators. Although ISAAC optimizes the energy cost of its DAC/ADC interface, the interface energy is still as large as 61% in ISAAC (see Fig. 4 (c)). Specifically, ISAAC [58] decreases the number of ADCs by sharing one ADC among 128 ReRAM bitlines, and thus the ADC sampling rate increases by 128×, increasing the energy cost of each A/D conversion.

Opportunity #3. Reducing the number of input accesses can save the energy cost of both input accesses and D/A conversions. We find that the input reuse of CNNs can still be improved over the prior works for reducing the energy overhead of input accesses and corresponding interfaces. Though each input connected to one row of an ReRAM array is naturally shared by B ReRAM cells along the row, each input on average has to be accessed $DZG/S^2/B$ times. Taking ISAAC [58] as an example, one 16-bit input involves $DZG/S^2/B$ times unit eDRAM read energy (i.e. 4416× the energy of a 16-bit ReRAM MAC), input register file read energy (i.e. 264.5× the energy of a 16-bit ReRAM MAC) and D/A conversion energy (i.e. 109.7× the energy of 16-bit ReRAM MAC). For MSRA-3 [31] adopted by ISAAC, each input of CONV layers is read and activated the interfaces 47 times on average.

B. TIMELY Innovations

The three aforementioned opportunities inspire us to develop the three innovations in TIMELY for greatly improving the acceleration energy efficiency. Fig. 5 (a) and (b) show a conceptual view of the difference between existing R^{2}PIM accelerators and TIMELY. Specifically, TIMELY mostly moves data in the analog domain as compared to the fully digital data movements in the existing designs and adopts DTCs and TDCs instead of DACs and ADCs for interfacing.

Innovation #1. TIMELY adopts ALBs to aggressively enhance (analog) data locality, leading to about N_{CB}× reduction in data movement energy costs per input and per Psum compared with existing designs, assuming a total of $N_{CB} \times N_{CB}$ crossbars in each sub-Chip. Multiple sub-Chips compose one chip. One key difference between TIMELY and existing R^{2}PIM resides in their sub-Chip design (see Fig. 5 (a) vs. (b)). Specifically, each crossbar (i.e. CB in Fig. 5) in

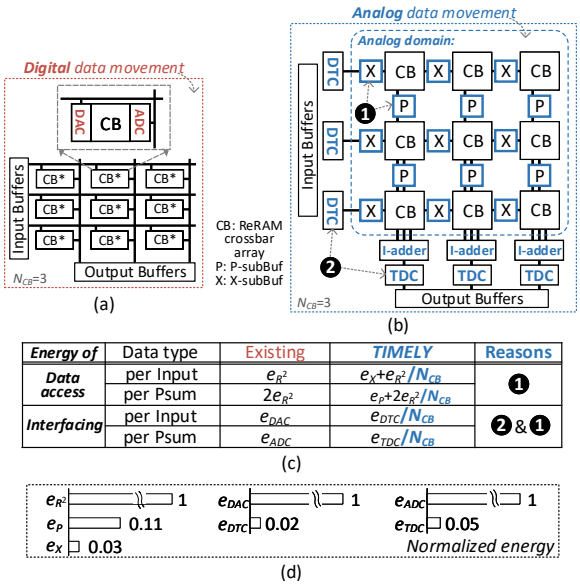


Fig. 5. A high-level view of (a) a sub-Chip within a chip of state-of-the-art R^{2}PIMs and (b) TIMELY's sub-Chip, (c) the energy cost per input and per Psum in state-of-the-art R^{2}PIMs and TIMELY, and (d) the normalized energy of different data accesses and interfaces, where e_{R^2}, e_X, and e_P are the unit energy of accessing ReRAM input/output buffers, X-subBuf, and P-subBuf, respectively, while e_{DAC}, e_{ADC}, e_{DTC}, and e_{TDC} denote the energy of one DAC, ADC, DTC, and TDC [14], [38], [41], [52], [58], [63], respectively.

existing designs fetches inputs from a high-cost memory (e.g. input buffers in Fig. 5 (a)). Therefore, for each sub-Chip, there is an energy cost of $BN_{CB}^2 e_{R^2}$ for accessing BN_{CB}^2 inputs. In TIMELY (see Fig. 5 (b)), an input fetched from the high-cost memory is shared by one row of the sub-Chip thanks to the adopted local ALB buffers (e.g. X-subBufs in Fig. 5 (b)) that are sandwiched between the crossbar arrays, resulting in an energy cost of $BN_{CB}e_{R^2} + BN_{CB}^2 e_X$ for handling the same number of inputs, leading to an energy reduction of N_{CB}× per input (see Fig. 5 (c). Similarly, each crossbar in existing R^{2}PIM accelerators directly writes and reads Psums to and from the high-cost output buffers, whereas in TIMELY the Psums in each column of the sub-Chip are accumulated before being written back to the output buffers, leading to an energy cost reduction of N_{CB}× per Psum (see Fig. 5 (c). Furthermore, accessing the high-cost memory requires about one order of magnitude higher energy cost than that of a local buffer. Specifically, the average energy of one high-cost memory access in PRIME is about 9× and 33× higher than that of P-subBufs and X-subBufs [79] in TIMELY, respectively. N_{CB} is typically >10 (e.g. $N_{CB} = 12$ in PRIME). Therefore, about N_{CB}× energy reduction for handling input/Psum accesses can be achieved in TIMELY. Additionally, the much reduced requirements of input/output buffer size in TIMELY make it possible to eliminate inter sub-Chip memory (see Fig. 6 (a) and Fig. 9 (c), leading to additional energy savings.

Innovation #2. TIMELY adopts TDIs and ALBs to minimize the energy cost of a single conversion and the total number of conversions, respectively. As a result, TIMELY reduces the interfacing energy cost per input and per Psum by $q_1 N_{CB}$ and $q_2 N_{CB}$, respectively, compared with current practices, where $q_1 = e_{DAC}/e_{DTC}$ and $q_2 = e_{ADC}/e_{TDC}$. It is

well recognized that the energy cost of ADC/DAC interfaces is another bottleneck in existing R^{2}PIM accelerators , in addition to that of data movements. For example, the energy cost of ADCs and DACs in ISAAC accounts for >61% of its total energy cost. In contrast, TIMELY adopts (1) TDCs/DTCs instead of ADCs/DACs to implement the interfacing circuits of crossbars and (2) only one TDC/DTC conversion for each row/column of **one sub-Chip**, whereas each row/column of **crossbar** needs one ADC/DAC conversion in existing designs, leading to a total of $q_1 N_{CB}\times$ and $q_2 N_{CB}\times$ reduction per input and Psum, respectively, as compared to existing designs. Specifically, q_1 and q_2 are about 50 and 20 [38], [41], [52], [58], [63], respectively.

Innovation #3. TIMELY employs O^{2}IR to further reduce the number and thus energy cost of input accesses and D/A conversions. As accessing the input and output buffers in sub-Chips costs about one order of magnitude higher energy than that of accessing local buffers between the crossbar arrays (see the left part of Fig. 5 (d)), we propose an O^{2}IR strategy to increase the input reuse opportunities for minimizing the cost of input accesses and associated D/A conversion.

IV. TIMELY Architecture

In this section, we first show an architecture overview (see Section IV-A), and then describe how the TIMELY architecture integrates the three innovations for aggressively improving the acceleration energy efficiency in Sections IV-B, IV-C, and IV-D, respectively. In addition, we introduce our pipeline design for enhancing throughput in Section IV-E and the software-hardware interface design for offering programmability in Section IV-F. Parameters are summarized in Table I.

A. Overview

Fig. 6 (a) shows the TIMELY architecture, which consists of a number of sub-Chips connected via bus [14], [58]. Specifically, each sub-Chip includes DTCs/TDCs (on the left/at the bottom), ReRAM input/output buffers (on the left/at the bottom), ReRAM crossbars (see ❹ in Fig. 6 (a)) with each having $B \times B$ bit cells, a mesh grid of local ALB buffers – i.e., X-subBufs (see ❶ in Fig. 6 (a)) and P-subBufs (see ❷ in Fig. 6 (a)) – between the ReRAM crossbar arrays, current adders (i.e. I-adders, ❸ in Fig. 6 (a)), and a block of shift-and-add, ReLU, max-pooling units.

The TIMELY architecture processes CNNs/DNNs' inference as follows. The pre-trained weights are pre-loaded into TIMELY's ReRAM arrays. Inputs of the CNN/DNN layers are fetched into the input buffers of one sub-Chip or several sub-Chips that handle the corresponding layers, starting from the first CNN/DNN layer. Within each sub-Chip, the inputs are applied to the DTCs for converting the digital inputs into analog time signals, which are then shared by ReRAM bit cells in the same row of all crossbar arrays along the horizontal direction to perform dot products with the corresponding resistive weights. The calculated Psums at the same column of all crossbars in the vertical direction are aggregated in the I-adders, and converted into a voltage signal and then an analog

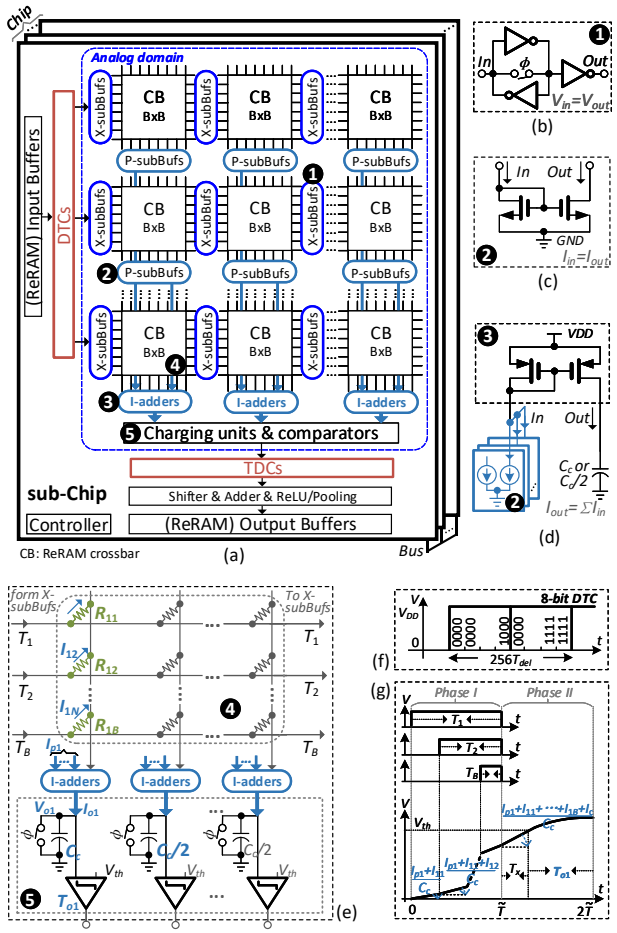


Fig. 6. An illustration of the (a) TIMELY architecture: (b) ❶ X-subBuf, (c) ❷ P-subBuf, (d) ❸ I-adder, (e) ❹ ReRAM crossbar located in the first crossbar column and last row of a sub-Chip, ❺ charging units, and comparators, (f) the input/output characteristics of an 8-bit DTC, and (g) the input/output characteristic of dot-product operations in the leftmost ReRAM column of a sub-Chip.

time signal by a charging unit and comparator block (see ❺ in Fig. 6 (a)) before being converted into a digital signal via a TDC. Note that the output of each P-subBuf is connected to the I-adder separately. Finally, the resulting digital signals are applied to the block of shift-and-add, ReLU, max-pooling units, and then written to the output buffers.

B. Enhancing (Analog) Data Locality

Within each sub-chip of TIMELY, the converted inputs and calculated Psums are moved in the analog domain with the aid of the adopted ALBs (see Fig. 6 (a)) after the digital inputs are converted into time signals by DTCs and before the Psums are converted into digital signals by TDCs. In this subsection, we first introduce the data movement mechanism and then present the operation of the local analog buffers.

Data Movement Mechanism. In the TIMELY architecture, time inputs from the DTCs move horizontally across the ReRAM crossbar arrays in the same row via X-subBufs (see ❶ in Fig. 6 (a))) for maximizing input reuses and minimizing high-cost memory accesses. Meanwhile, the resulting current Psums move vertically via P-subBufs (see ❷ in Fig. 6 (a)).

Note that only the crossbars in the leftmost column fetch inputs from DTCs while those in all the remaining columns fetch inputs from their analog local time buffers (i.e., the X-subBufs to their left). Similarly, only the outputs of the I-adders are converted into the digital signals via TDCs before they are stored back into the output buffers, while the current outputs of the crossbars are passed into the I-adders via analog current buffers (i.e., the P-subBufs right below them). In this way, TIMELY processes most data movements in the analog domain within each sub-chip, greatly enhancing data locality for improving the energy efficiency and throughput.

Local Analog Buffers. The local analog buffers make it possible to handle most (analog) data movements locally in TIMELY. Specifically, X-subBuf buffers the time signals (i.e., outputs of the DTCs) by latching it, i.e., copying the input delay time to the latch outputs (see Fig. 6 (b)); while P-subBuf buffers the current signal outputted from the ReRAM crossbar array, i.e. copying the input current to their outputs (see Fig. 6 (c)). The key is that X-subBuf and P-subBuf are more energy and area efficient than input/output buffers (see Fig. 5 (d)). Specifically, an X-subBuf buffer consists of two cross-coupled inverters that form a positive feedback to speed up the response at its output and thus reduces the delay between its inputs and outputs [70]. Since cross-coupled inverters invert the input, a third inverter is used to invert the signal back. X-subBufs are reset in each pipeline-cycle by setting ϕ to be high (see Fig. 6 (b)). The P-subBuf buffer is implemented using an NMOS-pair current mirror (see Fig. 6 (c)) [37].

C. Time-Domain Dot Products and DTC/TDC Interfacing

TIMELY performs dot products with time-domain inputs from the DTCs and converts time-domain dot product results into digital signals via TDCs. In this subsection, we first present dot product operations in TIMELY and then introduce their associated DTCs/TDCs.

Dot Products. First, let us consider Psums in one ReRAM crossbar array. Take the first column of the ReRAM crossbar array in Fig. 6 (e) as an example. A total of B time-domain inputs T_i $(i = 1, 2, ..., B)$ are applied to their corresponding ReRAM bit cells with resistance values of R_{1i} (i.e. corresponding to weights) to generate a Psum current (i.e. T_i-controlled current) based on the Kirchoff's Law. Then, let us focus on Psums in one sub-Chip. The Psum currents at the same column of all N_{CB} crossbars in the vertical direction are aggregated in the I-adder [2] (see ❸ in Fig. 6 (a)), and then are converted into a voltage V_{o1} by charging a capacitor (e.g. C_c in Fig. 6 (e)). Fig. 6 (g) shows the input/output characteristic of the dot product. We adopt a 2-phase charging scheme [5]. In phase I, the charging time is the input T_i and the charging current is VDD/R_{1i}, which corresponds to the weight. In phase II, the charging time is T_x and the charging current is a constant I_c, which is equal to $C_c BN_{CB} V_{DD}/R_{min}$. The charging in phase II ensures the voltage on C_c is larger than V_{th}, and the time output is defined by $\widetilde{T} - T_x$. R_{min} is the minimum mapped resistance of one layer. V_{th} is the threshold voltage of the comparator, which is equal to $BN_{CB}\widetilde{T}V_{DD}/R_{min}$, where V_{DD} is the logic high voltage of the time signal, and $\widetilde{T}$ is the time period of one phase. Based on Charge Conservation, we can derive the output $T_{o,8b/4b}$ (see T_{o1} in Fig. 6 (e)), where 8b/4b represents 8-bit inputs and 4-bit weights, to be:

$$T_{o,8b/4b} = \frac{R_{min}}{C_c BN_{CB}} \sum_{i=1}^{BN_{CB}} T_i/R_{1i} \tag{2}$$

To realize dot products with 8-bit weights and inputs, we employ a sub-ranging design [22], [47], [84] in which 8-bit weights are mapped into two adjacent bit-cell columns with the top-4 most significant bit (MSB) weights and the remaining 4 least significant bit (LSB) weights, respectively. The charging capacitors associated with MSB-weight column and LSB-weight column are C_c and $C_c/2$, respectively. $T_{o,8b/4b}$ of the MSB-weight column and the LSB-weight column are added to get the dot-product result for 8-bit weights.

DTCs/TDCs. We adopt 8-bit DTCs/TDCs for TIMELY based on the measurement-validated designs in [41], [52]. The input/output characteristics of a 8-bit DTC is shown in Fig. 6 (f), where digital signals of "1111111" and "00000000" correspond to the time-domain analog signals with the maximum and minimum delays, respectively, and the dynamic range of the time-domain analog signals are $256 \times T_{del}$ with T_{del} being the unit delay. Meanwhile, a TDC's input/output characteristics can also be viewed in Fig. 6 (f) by switching the V and t axes. In TIMELY, T_{del} is designed to be 50 ps, leading to a conversion time of 25 ns (including a design margin) for the 8-bit DTC/TDC. In addition, to trade off energy efficiency and computational density, one DTC/TDC is shared by γ ($\gamma \geq 1$) ReRAM crossbar rows/columns.

D. TIMELY's Only-Once Input Read Mapping Method

O^{2}IR follows three principles: (1) for reusing the inputs by different filters, we map these filters in parallel within the crossbar arrays (see Fig. 7 (a)); (2) for reusing the inputs when sliding the filter vertically within an input feature map, we duplicate the filters with a shifted offset equal to $Z \times S$ (see Fig. 7 (b)), where Z and S are the filter height and the stride, respectively; and (3) for reusing the inputs when sliding the filter horizontally within an input feature map, we transfer inputs to the adjacent X-subBufs with an step equal to S (see Fig. 7 (c)). Single-direction input transfer between adjacent X-subBufs can be implemented by introducing only one switch and one control signal to one X-subBuf.

E. Pipeline Design

To enhance throughput, we adopt pipeline designs between and within sub-Chips, i.e., inter-sub-Chip and intra-sub-Chip pipeline. Different sub-Chips work in a pipeline way. Note that a layer by layer weight mapping strategy is adopted in TIMELY, where one CNN/DNN layer is mapped into one sub-Chip if the ReRAM crossbars' size is larger than the required size; otherwise, a layer is mapped into multiple sub-Chips. In one sub-Chip, the following operations – reading inputs from input buffers, DTCs, analog-domain computation (including dot-product, charging-and-comparison operations), TDCs, and

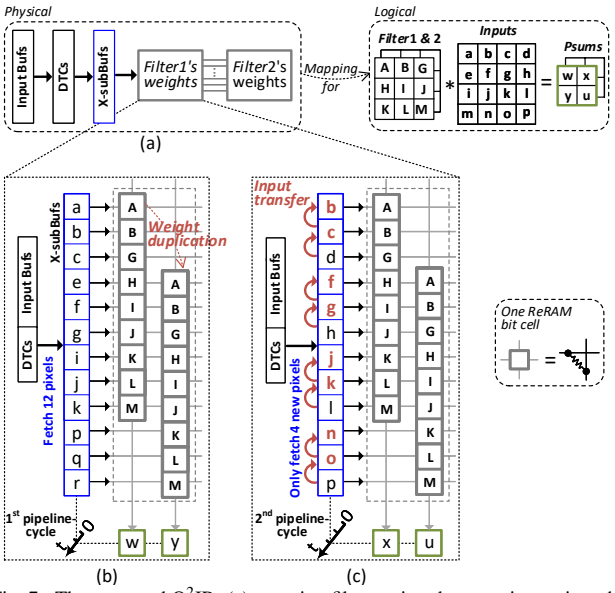


Fig. 7. The proposed O^{2}IR: (a) mapping filters using the same inputs into the same rows of crossbars; (b) duplicating filters with a vertical offset of $Z \times S$ between adjacent ReRAM columns; and (c) temporally shifting inputs by an amount equal to S.

writing back to output buffers – are pipelined. The pipeline-cycle time is determined by the slowest stage. Let us take the operations within one sub-Chip as an example to illustrate the pipeline in TIMELY. Assuming the first data is read from an input buffer at the first cycle, it spends three cycles to complete the digital-to-time conversion, analog-domain computation, and time-to-digital conversion, and is written back to an output buffer at the fifth cycle. Meanwhile, at the fifth cycle, the fifth, fourth, third, and second data is read, converted by a DTC, computed in the analog-domain, and converted by a TDC, respectively.

F. Software-Hardware Interface

A software-hardware interface is adopted to allow developers to configure TIMELY for different CNNs/DNNs, enabling programmability. Similar to the interface in PRIME, three stages are involved from software programming to hardware execution. First, the CNN/DNN is loaded into an NN parser [83] that automatically extracts model parameters. Second, with the extracted parameters, a compiler optimizes mapping strategies for increasing the utilization of ReRAM crossbar arrays and then generates execution commands (including commands for weight mapping and input data path configuration). Third, the controller (see Fig. 6 (a)) loads the commands from the compiler to (1) write pre-trained weights to the mapped addresses, and (2) configure peripheral circuits for setting up input paths of computation.

V. Discussion

Although local buffers have been adopted in digital accelerators [10], [82], it is challenging when using local buffers in R^{2}PIMs because: (1) improper design can largely compromise R^{2}PIMs' high computational density and (2) more frequent large-overhead A/D and D/A conversions may be caused. To the best of our knowledge, TIMELY is the first to implement and maximize analog data locality via ALBs, which have at least one order of magnitude lower access energy cost compared to the two level memories in PRIME [14]/ISAAC [58]/Pipelayer [62]. Additionally, TIMELY maximizes data locality without degrading R^{2}PIMs' computational density. Although a recent R^{2}PIM accelerator, CASCADE [15], has adopted analog buffers, it only uses analog ReRAM buffer to reduce the number of A/D conversions, thereby minimizing computational energy. TIMELY uses ALBs to minimize both the computational energy and data movement energy. Taking PRIME as an example, the computational energy only accounts for 17% of the chip energy. In order to minimize the computational energy, TIMELY not only reduces the number of A/D conversions by ALBs, but also decreases the energy of each A/D conversion by TDCs.

Analog computations and local buffers are efficient, but they potentially introduce accuracy loss to TIMELY. The accuracy loss is mainly attributed to the non-ideal characteristics of analog circuits. To address this challenge, TIMELY not only leverages algorithm resilience of CNNs/DNNs to counter hardware vulnerability [9], [48], [81], but also minimize potential errors introduced by hardware, thereby achieving the optimal trade-off between energy efficiency and accuracy. First, we choose time and current signals to minimize potential errors. Compared with analog voltage signals, analog current signals and digitally implemented time signals can tolerate larger errors caused by their loads, and analog time signal is less sensitive to noise and PVT variations [49]. Second, the adopted ALBs help improve the accuracy of time inputs and Psums by increasing the driving ability of loads. However, the larger the number of ALBs, the smaller the number of ReRAM crossbar arrays in a sub-Chip, compromising the computational density. Based on system-level evaluations, we adopt one X-subBuf between each pair of neighboring ReRAM crossbar arrays and one P-subBuf between each ReRAM crossbar array and its I-adder in order to achieve a good trade-off between accuracy loss and computational density reduction. Third, we limit the number of cascaded X-subBufs in the horizontal direction to reduce the accumulated errors (including noise) of time-domain inputs, which can be tolerated by the given design margin. We assign a design margin (i.e. more than 40 ps) for the unit delay (i.e. 50 ps) of the DTC conversion. We do not cascade P-subBufs to avoid introducing errors in Psum.

TIMELY adopts pipeline designs to address the speed limit of time signal operations and thus improve throughput. Adjusting the number of ReRAM rows/columns shared by one DTC/TDC allows for the trade-off between the throughput and computational density of TIMELY. TIMELY compensates for the increased area due to the special shifted weight duplication of O^{2}IR (see Fig. 7 (b) and (c)) by saving peripheral circuits' area. Besides, TIMELY also replicates weights to improve computation parallelism and thus throughput, similar to prior designs [14], [58], [62].

VI. Evaluation

In this section, we first introduce the experimental setup, and then compare TIMELY with state-of-the-art designs in terms of energy efficiency, computational density, and throughput. After that, we demonstrate the effectiveness of TIMELY's key features: ALB, TDI, and O^2IR, and show that these features are generalizable. Finally, we discuss area scaling.

A. Experiment Setup

TIMELY Configuration. For a fair comparison with PRIME/ISAAC, we adopt PRIME/ISAAC's parameters, including ReRAM and ReLU parameters from PRIME [14], and maxpool operations (scaled up to 65nm) and HyperTransport links from ISAAC [58] (see Table II). For TIMELY's specific components, we use silicon-verified results [41], [52] for DTCs and TDCs, and adopt Cadence-simulated results for X-subBuf, P-subBuf, I-adder, charging circuit, and comparator based on [35], [37], [70] – including their drives and loads during simulation. Supporting digital units (shifter and adder) consume negligibly small amounts of area and energy. All the design parameters of the peripheral circuits are based on a commercial 65nm CMOS process. The power supply is 1.2 V, and the clock rate is 40 MHz. The reset phase ϕ in Fig. 6 is 25 ns. The pipeline-cycle time is determined by the latency of 8 (setting γ to 8) DTCs/TDCs, which have a larger latency than other pipelined operations. The latency of reading corresponding inputs, analog-domain computations, and writing outputs back to output buffers are 16 ns [24], 150 ns [24], and 160 ns [24], respectively. In addition, I-adders and its inputs do not contribute to the total area because we insert I-adders and the interconnection between each P-subBuf and I-adder under the charging capacitors and ReRAM crossbars, leveraging different IC layers. We adopt 106 sub-Chips in the experiments for a fair comparison with the baselines (e.g., TIMELY vs. ISAAC: 91mm^2 vs. 88 mm^2).

Methodology. We first compare TIMELY with 4 state-of-the-art R^{2}PIM accelerators (PRIME [14], ISAAC [58], PipeLayer [62], and AtomLayer [56]) in terms of peak energy efficiency and computational density. For this set of experiments, the performance data of the baselines are the ones reported in their corresponding papers. Second, as for the evaluation regarding various benchmarks, we consider only PRIME [14] and ISAAC [58] because (1) there is lack of design detail information to obtain results for PipeLayer [62] and AtomLayer [56], and (2) more importantly, such comparison is sufficient given that PRIME [14] is the most competitive baseline in terms of energy efficiency (see Fig. 1 (c)). For this set of evaluations, we build an in-house simulator to evaluate the energy and throughput of PRIME, ISAAC, and TIMELY. Before using our simulator, we validate it against PRIME's simulator [14] and ISAAC's analytical calculations [58]. We set up our simulator to mimic PRIME and ISAAC and compare the results of our simulator with their original results. The resulting errors of energy and throughput evaluation are 8% and zero, respectively, which are acceptable by TIMELY's one order of magnitude improvement on energy efficiency (see Section VI-B). Due to the lack of ISAAC's mapping information, we only validate our simulator against PRIME's simulator to get the energy error by adopting PRIME's component parameters and weight mapping strategy [14] in our simulator. Since PRIME does not support inter-layer pipeline, we only validate our simulator against ISAAC's analytical calculations to get the throughput error by using ISAAC's component parameters and balanced inter-layer pipeline [14] in our simulator. The inter-layer pipeline corresponds to TIMELY's inter-sub-Chip pipeline.

TABLE II
TIMELY PARAMETERS.

Component	Params	Spec	Energy (fJ) /compo.	Area (μm^2) /compo.
TIMELY sub-Chip				
DTC	resolution number	8 bits 16×32	37.5	240
ReRAM crossbar	size number bits/cell	256×256 16×12 4	1792	100
Charging+ comparator	number	12×256	41.7	40
TDC	resolution number	8 bits 12×32	145	310
X-subBuf	number	12×16×256	0.62	5
P-subBuf	number	15×12×256	2.3	5
I-adder	number	12×256	36.8	40
ReLU	number	2	205	300
MaxPool	number	1	330	240
Input buffer	size/number	2KB/1	12736	50
Output buffer	size/number	2KB/1	31039	50
Total				**0.86 mm^2**
TIMELY chip (40 MHz)				
sub-Chip	number	106[a]		0.86 mm^2
Total				**91[a] mm^2**
Inter chips				
Hyper link	links/freq link bw	1/1.6GHz 6.4 GB/s	1620	5.7 mm^2

[a] Scaling TIMELY to an area of 0.86χ mm^2 by adjusting the number of sub-Chips (i.e., χ) based on applications.

TABLE III
ADOPTED BENCHMARKS AND DATASETS.

Benchmarks	Why consider these CNN/DNN models
VGG-D[a], CNN-1[b], MLP-L[b]	For a fair comparison with PRIME (i.e. benchmarks in [14])
VGG-1/-2/-3/-4[a] MSRA-1/-2/3[a]	For a fair comparison with ISAAC (i.e. benchmarks in [58])
ResNet-18/-50/-101/-152[a] SqueezeNet[a]	To show TIMELY's performance in diverse and more recent CNNs

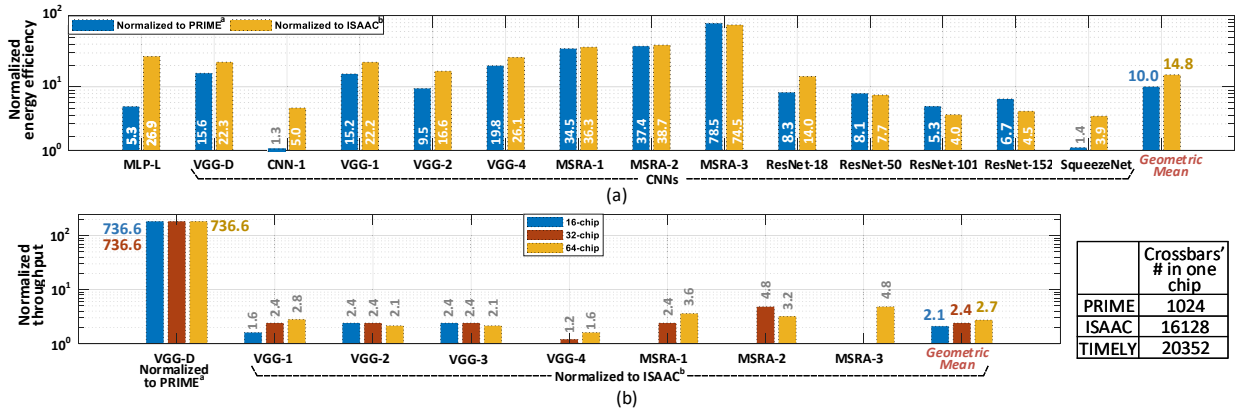
[a] ImageNet ILSVRC dataset [17]; [b] MNIST dataset [18]

Benchmarks. We evaluate TIMELY using a total of 15 benchmarks. Table III shows these benchmarks and the reasons for adopting them.

B. Evaluation Results

We evaluate TIMELY's peak energy efficiency and computational density against those reported in [14], [58], [62], and [56]. Next, we perform an evaluation of TIMELY's energy efficiency and throughput on various CNN and DNN models.

Overall Peak Performance. Compared with representative R^{2}PIM accelerators (see Table IV), TIMELY can improve energy efficiency by over 10× (over PRIME [14]) and the computational density by over 6.4× (over PipeLayer [62]). In particular, TIMELY improves energy efficiency by 10× to 49.3× and computational density by 6.4× to 31.2×. These



	Crossbars' # in one chip
PRIME	1024
ISAAC	16128
TIMELY	20352

[a] Adopting 8-bit inputs/outputs/weights when compared with PRIME which uses 6-bit inputs/outputs and 8-bit weights. Thus, the improvement over PRIME is slightly higher than results in (a) and (b)
[b] Adopting 16-bit inputs/outputs/weights when compared with ISAAC which uses the same data precision.

Fig. 8. (a) The normalized energy efficiency and (b) throughput of TIMELY over PRIME and ISAAC, respectively, considering various CNNs and DNNs.

TABLE IV
PEAK PERFORMANCE COMPARISON.

	Energy efficiency ($TOPs/W$)	Improvement of TIMELY	Computational density ($TOPs/(s \times mm^2)$)	Improvement of TIMELY
PRIME[a] [14]	2.10	+10.0×	1.23	+31.2×
ISAAC[b] [58]	0.38	+18.2×	0.48	+20.0×
PipeLayer[b] [62]	0.14	+49.3×	1.49	+6.4×
AtomLayer[b] [56]	0.68	+10.1×	0.48	+20.0×
TIMELY [a]	**21.00**	n/a	**38.33**	n/a
TIMELY [b]	**6.90**	n/a	**9.58**	n/a

[a] one operation: 8-bit MAC; [b] one operation: 16-bit MAC

large improvements result from TIMELY's innovative features of ALB, TDI, O^{2}IR and intra-sub-Chip pipelines, which can aggressively reduce energy cost of the dominant data movements and increase the number of operations given the same time and area. In Table IV, we ensure that TIMELY's precision is the same as that of the baselines for a fair comparison. Specifically, we consider a 8-bit TIMELY design when comparing with PRIME and a 16-bit TIMELY design when comparing to ISAAC, PipeLayer, and AtomLayer.

Energy Efficiency on Various CNN and DNN models. We evaluate TIMELY on various models (1 MLP and 13 CNNs) to validate that its superior performance is generalizable to different computational and data movement patterns. Fig. 8 (a) shows the normalized energy efficiency of TIMELY over PRIME and ISAAC. We can see that TIMELY outperforms both PRIME and ISAAC on all CNN and DNN models. Specifically, TIMELY is on average 10× and 14.8× more energy efficient than PRIME and ISAAC, respectively (see the Geometric Mean in the rightmost part of Fig. 8 (a)). This set of experimental results demonstrates that TIMELY's superior energy efficiency is independent of CNNs and DNNs – i.e. computational and data movement patterns. In addition, as shown in Fig. 8 (a), the energy efficiency improvement of TIMELY decreases in small or compact CNNs, such as CNN-1 [14] and SqueezeNet [29]. This is because their energy costs of data movements are relatively small. These models can be mapped into one ReRAM bank of PRIME or one ReRAM tile of ISAAC, and thus do not require high cost memory accesses and limit the energy savings achieved by TIMELY.

Throughput on Various CNNs. Fig. 8 (b) shows TIMELY's normalized throughput over PRIME and ISAAC on various CNNs (a total of 8 CNNs) considering three chip configurations (16, 32, and 64 chips). As the throughput is a function of the weight duplication ratio, we only consider CNNs for which PRIME or ISAAC provides corresponding weight duplication ratios. Compared to PRIME, TIMELY enhances the throughput by 736.6× for the 16-chip, 32-chip, and 64-chip configurations on VGG-D. TIMELY's advantageous throughput results from its intra-sub-Chip pipeline, which enables to minimize the latency between two pipelined outputs. In addition, PRIME can work in both the memory mode and computation mode (i.e. accelerating CNN), limiting the number of crossbars for CNN computations (and thus its throughput) on a chip which is over 20× smaller than that of TIMELY (i.e. 1024/20352, see the right corner of Fig. 8 (b)). Compared to ISAAC on 7 CNNs, TIMELY, on average, enhances the throughput by 2.1×, 2.4×, and 2.7× for the 16-chip, 32-chip, and 64-chip configurations, respectively. In Fig. 8 (b), we consider only 64-chip or (32-chip and 64-chip) for large CNNs, such as MSRA-1/-2/-3, to ensure that all the models can be mapped into one TIMELY or ISAAC accelerator. TIMELY's enhanced throughput is because ISAAC adopts serial operations and requires 22 pipeline-cycles (each being 100 ns) to finish one 16-bit MAC operation, for which TIMELY employs intra-sub-Chip pipelines and needs two pipeline-cycles (each being 200 ns).

Accuracy. We observe $\leq$ 0.1% inference accuracy loss under various CNN and DNN models in system-level simulations including circuit-level errors extracted from Cadence simulation. The simulation methodology is adapted from prior work [33], [34]. Specifically, we first obtain noise and PVT variations (by Monte-Carlo simulations in Cadence) of X-subBuf, P-subBuf, I-adder, DTC, and TDC. The errors follow Gaussian noise distribution. We then add equivalent noise during training and use the trained weights for inference. Note that prior work has proved that adding Gaussian noise to training can reach negligible accuracy loss [53], [54], [57]. To achieve $\leq$ 0.1% accuracy loss, we set the number of cascaded X-subBufs to 12. The accumulated error of the cascaded X-subBufs is $\sqrt{12}\varepsilon$ [20], where ε is the potential error of one X-subBuf. $\sqrt{12}\varepsilon$ is less than 20×2^8 ps, which can be tolerated

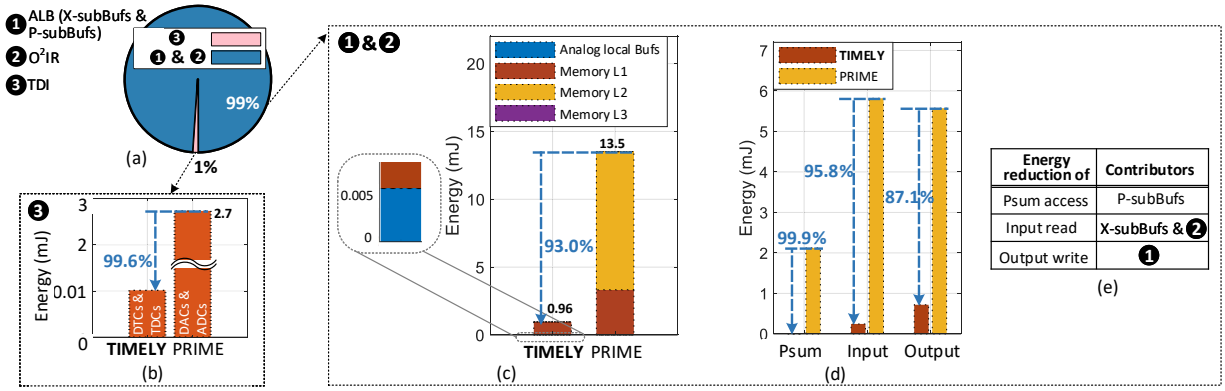


Fig. 9. The effectiveness of TIMELY's innovations: (a) a breakdown of energy savings (over PRIME on VGG-D) achieved by different features – i.e. (❶ and ❷) vs. ❸ of TIMELY; (b) comparing the energy costs of the interfacing circuits in TIMELY and PRIME; energy breakdown with regard to both (c) memory types and (d) data types in both TIMELY and PRIME; and (e) the contributing factors for the energy savings per data type in TIMELY (see (d)).

by the design margin of 40×2^8 ps and thus do not cause a loss of inference accuracy.

C. Effectiveness of TIMELY's Innovations

We first validate the effectiveness of TIMELY's innovations on energy saving and area reduction, and then demonstrate that TIMELY's innovative principles can be generalized to state-of-the-art R^{2}PIM accelerators to further improve their energy efficiency.

Effectiveness of TIMELY's Innovations on Energy Savings. We here present an energy breakdown analysis to demonstrate how TIMELY reduces the energy consumption on VGG-D as compared with PRIME, which is the most competitive R^{2}PIM accelerator in terms of energy efficiency. In Fig. 8 (a), we can see that TIMELY improves the energy efficiency by 15.6 $\times$ as compared to PRIME.

Overview. We first show the breakdown of energy savings achieved by different features of TIMELY. TIMELY's ALB and O^{2}IR contribute to up to 99% of the energy savings, and its TDI leads to the remaining 1% (see Fig. 9 (a)).

Effectiveness of TIMELY's ALB and O^{2}IR. We compare TIMELY's energy breakdown with regard to both memory types and data types with those of PRIME in Fig. 9 (c) and (d), respectively. In Fig. 9 (c), TIMELY's ALB and O^{2}IR together reduce the energy consumption of memory accesses by 93% when compared with PRIME. Specifically, the ALB and O^{2}IR features enable TIMELY to fully exploit local buffers within its sub-Chips for minimizing accesses to the L1 memory and removing the need to access an L2 memory.

In Fig. 9 (d), TIMELY reduces the energy consumption associated with the data movement of Psums, inputs and outputs by 99.9%, 95.8%, and 87.1%, respectively. The contributing factors are summarized in Fig. 9 (e). Specifically, (1) TIMELY can handle most of the Psums locally via the P-subBufs within the sub-Chips, aggressively reducing the energy cost of data movements of Psums; (2) TIMELY's O^{2}IR feature ensures all the input data are fetched only once from the L1 memory while its ALB feature (i.e. X-subBufs here) allows the fetched inputs to be stored and transferred via X-subBufs between the crossbars; and (3) thanks to employed P-subBufs and X-subBufs, TIMELY removes the need for an L2 memory, which has 146.7$\times$/6.9$\times$ higher read/write energy than that of an L1 memory, respectively, reducing the energy cost of writing outputs back to the memory (L1 memory in TIMELY vs. L2 memory in PRIME). Furthermore, as another way to see the effectiveness of TIMELY's O^{2}IR feature, we summarize both PRIME's and TIMELY's total number of input accesses to the L1 Memory in Table V (consider the first six CONV layers as examples). TIMELY requires about 88.9% less L1 memory accesses.

Effectiveness of TIMELY's TDI. Although DTCs and TDCs only contribute to 1% of TIMELY's energy savings over PRIME, the total energy of DTCs and TDCs in TIMELY is 99.6% less than that of ADCs and DACs in PRIME (see Fig. 9 (b)). It is because (1) the unit energy of one DTC/TDC is about 30%/23% of that of DAC/ADC; (2) the increased analog data locality due to ALBs largely reduces the need to activate DTCs and TDCs; and (3) TIMELY's O^{2}IR feature aggressively reduces the required DTC conversions thanks to its much reduced input accesses to the L1 memory (see Table V).

Effectiveness of TIMELY's Innovations on Area Reduction. We analyze an area breakdown to present the effectiveness of TIMELY's innovations on the area savings of peripheral circuits, which helps to improve the computational

TABLE V
THE TOTAL NUMBER OF L1 MEMORY ACCESSES FOR READING INPUTS IN TIMELY AND PRIME [14] CONSIDERING VGG-D.

	CONV1	CONV2	CONV3	CONV4	CONV5	CONV6
PRIME [14]	1.35 M	28.90 M	7.23 M	14.45 M	3.61 M	7.23 M
TIMELY	0.15 M	3.21 M	0.80 M	1.61 M	0.40 M	0.80 M
Save by	88.9%	88.9%	88.9%	88.9%	88.9%	88.9%

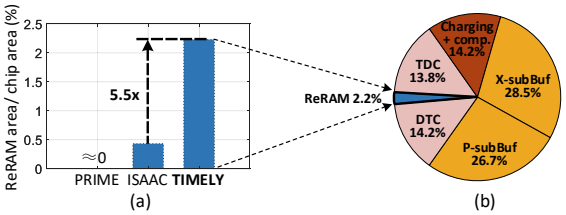


Fig. 10. (a) The percentage of ReRAM crossbar area in the area of PRIME [14], ISAAC [58], and TIMELY and (b) the area breakdown of TIMELY

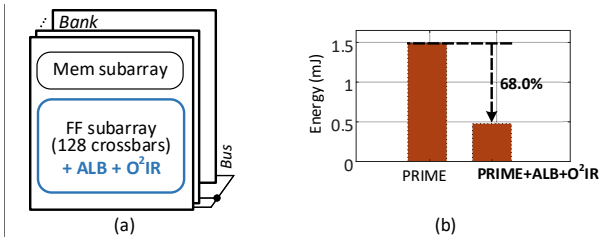


Fig. 11. Comparing the total energy cost of intra-bank data movements between PRIME and PRIME with TIMELY's ALB and O^2IR being applied: (a) applying ALB and O^2IR to PRIME architecture and (b) the resulting energy reduction.

density (see Table IV). In Fig. 10 (a), the percentage of the ReRAM array area in TIMELY (i.e. 2.2%) is 5.5× higher than that in ISAAC (i.e. 0.4%) [58]. The percentage of the ReRAM array area in PRIME is small enough and thus ignored [14]. The higher percentage of the ReRAM crossbar array area in TIMELY benefits from area-efficient circuit implementations of TIMELY's ALB, TDI and O^2IR. Specifically, in TIMELY shown in Fig. 10 (b), X-subBufs and P-subBufs occupy 55.2% of the chip area; DTCs and TDCs occupy 28% of the chip area; the area of CMOS logic introduced by O^2IR is neglectable.

Generalization of TIMELY's Innovations. TIMELY's innovative features are generalizable and can be applied to state-of-the-art R^2PIM accelerators for boosting their energy efficiency. To demonstrate, we apply ALB and O^2IR to PRIME based on the following considerations. ALB feature associated with O^2IR contributes the dominant energy savings (see Fig. 9 (a)). From the perspective of data accesses and interfaces, PRIME uses the same architecture shown in Fig 5 (a) as ISAAC [58]/PipeLayer [62]. To evaluate, we modify PRIME architecture as shown in Fig. 11 (a). We add X-subBufs and P-subBufs between 128 ReRAM crossbar arrays in FF subarray of each bank, and modify the weights mapping and input access dataflow based on O^2IR, while employing PRIME's original designs outside FF subarray. Thus, ALB and O^2IR only have an impact on the intra-bank energy. In this experiment, we adopt the same component parameters as those used in the PRIME's original design. Fig. 11 (b) shows that applying ALB and O^2IR principle to FF subarrays in PRIME reduces the intra-bank data movement energy by 68%.

D. Discussion

Area scaling of TIMELY (by adjusting the number of sub-Chips shown in Table II) does not affect throughput and slightly affects energy. This is because throughput is determined only by intra-sub-Chip pipeline (see Section IV-E); adjusting the number of sub-Chip in one chip will only change inter-chip energy (i.e., the energy of memory L3 in Fig. 9 (c)), which accounts for a negligible part of the total energy.

VII. Related Work

Non-PIM CNN/DNN Accelerators. Although memory is only used for data storage in non-PIM accelerators, computing units are being pushed closer to compact memories to reduce energy and area. For accelerators with off-chip DRAM, DRAM accesses consume two orders of magnitude more energy than on-chip memory accesses (e.g. 130× higher than a 32-KB cache at 45 nm [27]). As a result, DRAM consumes more than 95% of the total energy in DianNao [13], [19]. To break through the off-chip bottleneck, on-chip SRAM is widely used as the mainstream on-chip memory solution [25]. However, SRAM's low density has been limiting its on-die capacity even with technology scaling. For example, EIE adopts 10-MB SRAM that takes 93.2% of the total area [25]. To address the area issue, on-chip eDRAM is used in RANA [64] and DaDianNao [11], as eDRAM can save about 74% area while providing 32 KB capacity in 65 nm [55], [64]. However, the refresh energy in eDRAM can be dominant (e.g. about 10× as high as the data access' energy [64]). In terms of FPGA-based designs, the performance is also limited by the memory accesses [21], [23], [55], [59] with limited flexibility of choosing memory technologies. Different from these non-PIM accelerators, TIMELY improves energy efficiency by computing in memory and enhances computational density through adopting high-density ReRAM.

PIM CNN/DNN Accelerators. While PIM accelerators integrate computing units in memory to save the energy of accessing weights, the achievable energy efficiency and computational density remain limited. The limited energy efficiency is induced by the energy cost of input and Psum movements and the overhead of interfacing circuits. PRIME [14] takes 83% of the total energy to access inputs and Psums, and ISAAC [58] consumes 61% of the total energy to operate DACs/ADCs. The limited computational density is related to memory technologies. Processing in SRAM, for example, faces this limitation. The reasons include not only one SRAM bit-cell typically stores only 1-bit weight [6], [23], [36], [60], [75] but also SRAM's bit-cell structure – e.g. 6T [23], [36]/8T [60], [75]/10T [6] structure – decreases density. Proposed TIMELY adopts high-density ReRAM, and addresses two key energy challenges with techniques including ALBs, TDIs, and O^2IR. TIMELY achieves up to 18.2× improvement (over ISAAC) in energy efficiency, 31.2× improvement (over PRIME) in computational density, and 736.6× in throughput (over PRIME). Similar to the effect of ALBs used in TIMELY, a recent R^2PIM accelerator [15] also increases the amount of data in the analog domain for energy optimization. However, it only optimizes the computation energy (including the energy of interfacing circuits).

VIII. Conclusions

In this paper, we analyze existing designs of R^2PIM accelerators and identify three opportunities to greatly enhance their energy efficiency: analog data locality, time-domain interfacing, and input access reduction. These three opportunities inspire three key features of TIMELY: (1) ALBs, (2) interfacing with TDCs/DTCs, and (3) an O^2IR mapping method. TIMELY outperforms state-of-the-art in both energy efficiency and computational density while maintaining a better throughput.

Acknowledgment

The authors would like to thank Dr. Rajeev Balasubramonian and Dr. Anirban Nag for their discussions on ISAAC [58].

REFERENCES

[1] "NVIDIA Deep Learning Accelerator (NVDLA)." [Online]. Available: http://nvdla.org/primer.html

[2] K. Amanpreet and P. Rishikesh, "Current mode computational circuits for analog signal processing," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 4, 2014.

[3] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, 2018.

[4] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, 2018.

[5] M. Bavandpour, M. R. Mahmoodi, and D. B. Strukov, "Energy-efficient time-domain vector-by-matrix multiplier for neurocomputing and beyond," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2019.

[6] A. Biswas and A. P. Chandrakasan, "Conv-ram: An energy-efficient sram with embedded convolution computation for low-power cnn-based machine learning applications," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, 2018.

[7] I. Bratt, "Arm's First-Generation Machine Learning Processor," *Hot Chips*, 2018. [Online]. Available: https://www.hotchips.org/hc30/2conf/2.07_ARM_ML_Processor_HC30_ARM_2018_08_17.pdf

[8] N. Cao, M. Chang, and A. Raychowdhury, "14.1 a 65nm 1.1-to-9.1tops/w hybrid-digital-mixed-signal computing platform for accelerating model-based and model-free swarm robotics," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2019.

[9] L. Chen, J. Li, Y. Chen, Q. Deng, J. Shen, X. Liang, and L. Jiang, "Accelerator-friendly neural-network training: Learning variations and defects in rram crossbar," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017.

[10] Y. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016.

[11] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, and O. Temam, "Dadiannao: A machine-learning supercomputer," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 2014.

[12] Z. Chen and J. Gu, "19.7 a scalable pipelined time-domain dtw engine for time-series classification using multibit time flip-flops with 140giga-cell-updates/s throughput," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2019.

[13] S. N. e. a. Chen T, Du Z, "Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," *ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2014.

[14] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016.

[15] T. Chou, W. Tang, J. Botimer, and Z. Zhang, "Cascade: Connecting rrams to extend analog dataflow in an end-to-end in-memory processing paradigm," in *Proceedings of the 52Nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '52. New York, NY, USA: ACM, 2019, pp. 114–125. [Online]. Available: http://doi.acm.org/10.1145/3352460.3358328

[16] M. Daisuke, Y. Ryo, H. Kazunori, K. Hiroyuki, K. Shouhei, O. Yukihito, and U. Yasuo, "A 10.4 pj/b (32, 8) ldpc decoder with time-domain analog and digital mixed-signal processing," in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 420–421.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009.

[18] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012.

[19] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "Shidiannao: Shifting vision processing closer to the sensor," in *2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, 2015.

[20] P. Dudek, S. Szczepanski, and J. V. Hatfield, "A high-resolution cmos time-to-digital converter utilizing a vernier delay line," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 2, pp. 240–247, Feb 2000.

[21] J. Fowers, K. Ovtcharov, K. Strauss, E. S. Chung, and G. Stitt, "A high memory bandwidth fpga accelerator for sparse matrix-vector multiplication," in *2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines*, 2014.

[22] B. P. Ginsburg and A. P. Chandrakasan, "500-ms/s 5-bit adc in 65-nm cmos with split capacitor array dac," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 4, April 2007.

[23] S. K. Gonugondla, M. Kang, and N. Shanbhag, "A 42pj/decision 3.12tops/w robust in-memory machine learning classifier with on-chip training," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb 2018.

[24] A. Grossi, E. Vianello, M. M. Sabry, M. Barlas, L. Grenouillet, J. Coignus, E. Beigne, T. Wu, B. Q. Le, M. K. Wootters, C. Zambelli, E. Nowak, and S. Mitra, "Resistive ram endurance: Array-level characterization and correction techniques targeting deep learning applications," *IEEE Transactions on Electron Devices*, vol. PP, 2019.

[25] M. H. e. a. Han S, Liu X, "Eie: efficient inference engine on compressed deep neural network," *ACM International Symposium on Computer Architecture*, 2016.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014.

[28] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: Programming 1t1m crossbar to accelerate matrix-vector multiplication," in *Proceedings of the 53rd annual design automation conference (DAC)*, 2016.

[29] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[30] P. Jain, U. Arslan, M. Sekhar, B. C. Lin, L. Wei, T. Sahu, J. Alzate-vinasco, A. Vangapaty, M. Meterelliyoz, N. Strutt, A. B. Chen, P. Hentges, P. A. Quintero, C. Connor, O. Golonzka, K. Fischer, and F. Hamzaoglu, "13.2 a 3.6mb 10.1mb/mm2 embedded non-volatile reram macro in 22nm finfet technology with adaptive forming/set/reset schemes yielding down to 0.5v with sensing time of 5ns at 0.7v," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb 2019.

[31] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15, 2015.

[32] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit, "One model to learn them all," *arXiv preprint arXiv:1706.05137*, 2017.

[33] M. Kang, S. K. Gonugondla, M.-S. Keel, and N. R. Shanbhag, "An energy-efficient memory-based high-throughput vlsi architecture for convolutional networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1037–1041.

[34] M. Kang, M.-S. Keel, N. R. Shanbhag, S. Eilert, and K. Curewitz, "An energy-efficient vlsi architecture for pattern recognition via deep embedding of computation in sram," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 8326–8330.

[35] A. Kaur and R. Pandey, "Current mode computational circuits for analog signal processing," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2014.

[36] W. Khwa, J. Chen, J. Li, X. Si, E. Yang, X. Sun, R. Liu, P. Chen, Q. Li, S. Yu, and M. Chang, "A 65nm 4kb algorithm-dependent computing-in-memory sram unit-macro with 2.3ns and 55.8tops/w fully parallel

product-sum operation for binary dnn edge processors," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb 2018.

[37] O. Krestinskaya, I. Fedorova, and A. P. James, "Memristor load current mirror circuit," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015.

[38] L. Kull, T. Toifl, M. Schmatz, P. A. Francese, C. Menolfi, M. Braendli, M. Kossel, T. Morf, T. M. Andersen, and Y. Leblebici, "A 3.1 mw 8b 1.2 gs/s single-channel asynchronous sar adc with alternate comparators for enhanced speed in 32 nm digital soi cmos," *IEEE Journal of Solid-State Circuits*, 2013.

[39] C. Lee, H. Lin, C. Lien, Y. Chih, and J. Chang, "A 1.4mb 40-nm embedded reram macro with 0.07um2 bit cell, 2.7ma/100mhz low-power read and hybrid write verify for high endurance application," in *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov 2017.

[40] S. Levantino, G. Marzin, and C. Samori, "An adaptive pre-distortion technique to mitigate the dtc nonlinearity in digital plls," *IEEE Journal of Solid-State Circuits*, 2014.

[41] M. Li, C. Yang, and Y. Ueng, "A 5.28-gb/s ldpc decoder with time-domain signal processing for ieee 802.15.3c applications," *IEEE Journal of Solid-State Circuits*, 2017.

[42] S. Li, A. O. Glova, X. Hu, P. Gu, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "Scope: A stochastic computing engine for dram-based in-situ accelerator," in *Proceedings of the 51th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-51 '18, 2018.

[43] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "Drisa: A dram-based reconfigurable in-situ accelerator," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-50 '17, 2017.

[44] Z. Li, Y. Chen, L. Gong, L. Liu, D. Sylvester, D. Blaauw, and H. Kim, "An 879GOPS 243mw 80fps VGA fully visual cnn-slam processor for wide-range autonomous exploration," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2019, pp. 134–136.

[45] Z. Li, J. Wang, D. Sylvester, D. Blaauw, and H. S. Kim, "A 1920 $\times$ 1080 25-Frames/s 2.4-TOPS/W low-power 6-D vision processor for unified optical flow and stereo depth with semi-global matching," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1048–1058, 2019.

[46] Y. Lin, S. Zhang, and N. Shanbhag, "Variation-Tolerant Architectures for Convolutional Neural Networks in the Near Threshold Voltage Regime," in *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct 2016, pp. 17–22.

[47] C. Liu, S. Chang, G. Huang, and Y. Lin, "A 10-bit 50-ms/s sar adc with a monotonic capacitor switching procedure," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, 2010.

[48] C. Liu, M. Hu, J. P. Strachan, and H. Li, "Rescuing memristor-based neuromorphic design with high defects," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2017.

[49] M. Liu, L. R. Everson, and C. H. Kim, "A scalable time-based integrate-and-fire neuromorphic core with brain-inspired leak and local lateral inhibition capabilities," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, 2017.

[50] S. Liu, Y. Lin, Z. Zhou, K. Nan, H. Liu, and J. Du, "On-demand deep model compression for mobile devices: A usage-driven model selection framework," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2018, pp. 389–400.

[51] A. Mantyniemi, T. Rahkonen, and J. Kostamovaara, "A cmos time-to-digital converter (tdc) based on a cyclic time domain successive approximation interpolation method," *IEEE Journal of Solid-State Circuits*, 2009.

[52] D. Miyashita, R. Yamaki, K. Hashiyoshi, H. Kobayashi, S. Kousai, Y. Oowaki, and Y. Unekawa, "An ldpc decoder with time-domain analog and digital mixed-signal processing," *IEEE Journal of Solid-State Circuits*, 2014.

[53] A. S. Morcos, D. G. Barrett, N. C. Rabinowitz, and M. Botvinick, "On the importance of single directions for generalization," *arXiv preprint arXiv:1803.06959*, 2018.

[54] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," *arXiv preprint arXiv:1802.05668*, 2018.

[55] M. Poremba, S. Mittal, D. Li, J. S. Vetter, and Y. Xie, "Destiny: A tool for modeling emerging 3d nvm and edram caches," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2015.

[56] X. Qiao, X. Cao, H. Yang, L. Song, and H. Li, "Atomlayer: A universal reram-based cnn accelerator with atomic layer computation," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018.

[57] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Advances in Neural Information Processing Systems*, 2018, pp. 2483–2493.

[58] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016.

[59] H. Sharma, J. Park, D. Mahajan, E. Amaro, J. K. Kim, C. Shao, A. Mishra, and H. Esmaeilzadeh, "From high-level deep neural models to fpgas," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016.

[60] X. Si, J. Chen, Y. Tu, W. Huang, J. Wang, Y. Chiu, W. Wei, S. Wu, X. Sun, R. Liu, S. Yu, R. Liu, C. Hsieh, K. Tang, Q. Li, and M. Chang, "24.5 a twin-8t sram computation-in-memory macro for multiple-bit cnn-based machine learning," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*.

[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. ICLR*, 2015.

[62] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2017.

[63] W. Tseng, J. Wu, and Y. Chu, "A cmos 8-bit 1.6-gs/s dac with digital random return-to-zero," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 1, Jan 2011.

[64] F. Tu, W. Wu, S. Yin, L. Liu, and S. Wei, "Rana: Towards efficient neural acceleration with refresh-optimized embedded dram," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018.

[65] I. Valov, R. Waser, J. R Jameson, and M. N Kozicki, "Electrochemical metallization memories–fundamentals, applications, prospects," *Nanotechnology*, vol. 22, 2011.

[66] H. Wang, Y. Zhao, C. Li, Y. Wang, and Y. Lin, "A new MRAM-based process in-memory accelerator for efficient neural network training with floating point precision," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2020, pp. 1–4.

[67] Y. Wang, Z. Jiang, X. Chen, P. Xu, Y. Zhao, Y. Lin, and Z. Wang, "E2-Train: Training State-of-the-art CNNs with Over 80% Energy Savings," in *Advances in Neural Information Processing Systems*, 2019, pp. 5139–5151.

[68] Y. Wang, J. Shen, T.-K. Hu, P. Xu, T. Nguyen, R. Baraniuk, Z. Wang, and Y. Lin, "Dual dynamic inference: Enabling more efficient, adaptive and controllable deep inference," *IEEE Journal of Selected Topics in Signal Processing*, 2019.

[69] R. Waser, R. Dittmann, G. Staikov, and K. Szot, "Redox–based resistive switching memories–nanoionic mechanisms, prospects, and challenges," *Advanced Materials*, vol. 21, 2009.

[70] L. Weitao, L. Fule, and W. Zhihua, "High-resolution and high-speed integrated cmos ad converters for low-power applications," Springer, 2018.

[71] H.-. P. Wong, H. Lee, S. Yu, Y. Chen, Y. Wu, P. Chen, B. Lee, F. T. Chen, and M. Tsai, "Metal–oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, 2012.

[72] J. Wu, Y. Wang, Z. Wu, Z. Wang, A. Veeraraghavan, and Y. Lin, "Deep *k*-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions," *arXiv preprint arXiv:1806.09228*, 2018.

[73] P. Xu, X. Zhang, C. Hao, Y. Zhao, Y. Zhang, Y. Wang, C. Li, Z. Guan, D. Chen, and Y. Lin, "AutoDNNchip: An automated dnn chip predictor and builder for both FPGAs and ASICs," *The 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Feb 2020. [Online]. Available: http://dx.doi.org/10.1145/3373087.3375306

[74] H. Yang, Y. Zhu, and J. Liu, "End-to-End Learning of Energy-Constrained Deep Neural Networks," *arXiv e-prints*, 2018.

[75] J. Yang, Y. Kong, Z. Wang, Y. Liu, B. Wang, S. Yin, and L. Shi, "24.4 sandwich-ram: An energy-efficient in-memory bwn architecture with pulse-width modulation," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*.

[76] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," *arXiv preprint*, 2017.

[77] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong, and H. Qian, "Face classification using electronic synapses," *Nature communications*, 2017.

[78] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, 2020.

[79] S. Yu, "Neuro-inspired computing with emerging nonvolatile memory," *Proceedings of the IEEE*, 2018.

[80] C. Zhang, J. Gu, L. Gao, T. Ouyang, and B. Wang, "Time-domain computing circuits for addition and multiplication computation," in *2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, 2017.

[81] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6t sram array," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, 2017.

[82] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, and Y. Chen, "Cambricon-x: An accelerator for sparse neural networks," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*.

[83] X. Zhang, J. Wang, C. Zhu, Y. Lin, J. Xiong, W.-m. Hwu, and D. Chen, "Dnnbuilder: an automated tool for building high-performance dnn hardware accelerators for fpgas," in *Proceedings of the International Conference on Computer-Aided Design*. ACM, 2018, p. 56.

[84] Y. Zhu, C. Chan, U. Chio, S. Sin, S. U, R. P. Martins, and F. Maloberti, "A 10-bit 100-ms/s reference-free sar adc in 90 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 6, 2010.