XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks

Shihui Yin, *Student Member, IEEE*, Zhewei Jiang, *Student Member, IEEE*, Jae-Sun Seo, *Senior Member, IEEE*, and Mingoo Seok, *Senior Member, IEEE*

***Abstract*— We present XNOR-SRAM, a mixed-signal in-memory computing (IMC) SRAM macro that computes ternary-XNOR-and-accumulate (XAC) operations in binary/ternary deep neural networks (DNNs) without row-by-row data access. The XNOR-SRAM bitcell embeds circuits for ternary XNOR operations, which are accumulated on the read bitline (RBL) by simultaneously turning on all 256 rows, essentially forming a resistive voltage divider. The analog RBL voltage is digitized with a column-multiplexed 11-level flash analog-to-digital converter (ADC) at the XNOR-SRAM periphery. XNOR-SRAM is prototyped in a 65-nm CMOS and achieves the energy efficiency of 403 TOPS/W for ternary-XAC operations with 88.8% test accuracy for the CIFAR-10 data set at 0.6-V supply. This marks 33× better energy efficiency and 300× better energy–delay product than conventional digital hardware and also represents among the best tradeoff in energy efficiency and DNN accuracy.**

***Index Terms*— Binary weights, deep neural networks (DNNs), ensemble learning, in-memory computing (IMC), SRAM, ternary activations.**

I. Introduction

DEEP neural networks (DNNs) and convolutional neural networks (CNNs) have unprecedentedly improved the accuracies in large-scale recognition tasks [1]–[6]. However, the arithmetic complexity and memory access have limited the energy efficiency and acceleration of DNN hardware [7]–[11].

To address this, in recent algorithms, weights and neuron activations are binarized to $+1$ or -1 [12], [13] such that the multiplication between an weight and an activation becomes an XNOR operation and the accumulation of the XNOR operations becomes bitcount of those XNOR results. Although the initial XNOR-Net [12] showed a relatively large test accuracy degradation (∼10%–20%) for the ImageNet data set, recent works that employ 2-bit precision [14] have shown 1%–3% accuracy degradation for ImageNet, and this is an active research area in the machine learning community. Taking advantage of the reduced computation complexity, dedicated hardware accelerators [15], [16] for the CIFAR-10 data set [17] have been proposed with digital or mixed-signal neuron array, achieving ∼86% test accuracy with all weights stored on a chip. It should also be noted that ternary precision has demonstrated better performance to binary precision [18], especially for large-scale data sets. To that end, implementing DNNs with ternary activation precision and binary weight precision is of a particular interest.

The arithmetic complexity reduction from the binary and ternary algorithms, however, makes row-by-row memory access dominating the speed and energy efficiency of DNN hardware [7]. Conventional on-chip static random-access memory, SRAM, requires row-by-row accesses, and fetching a very large number of weights in this manner consumes substantial energy and delay.

To reduce the delay and energy associated with on-chip SRAM accesses, recent works have proposed an SRAM-based in-memory computing (IMC) scheme, which performs computation on the bitline without reading out each row of bitcells [19]–[25], demonstrating large improvement in energy efficiency and throughput.

For example, CONV-SRAM [20] integrates digital-to-analog converters (DACs) for analog wordlines, binary weights stored in SRAM, and analog-to-digital converters (ADCs) to convert the in-memory computation results back to digital values. In-memory computation in [20] targets the convolution operation, which is accomplished by row-wise charge sharing of the SRAM bitcells in the same row. This design integrates the local analog multiply-and-average circuits every 16 rows (out of the 256-row bitcell array). It reported 98.3% accuracy for the MNIST data set [26].

IMC with on-chip training capability was presented in [21], where the weights were fine-tuned based on chip's variability. This article reported 96% accuracy on the MIT-CBCL data set [27]. This hardware reads out 8-bit weights across four rows in the analog domain and performs the analog voltage signed multiplication and accumulation in the peripheral analog processor.

A binarized CNN accelerator was presented in [23]. It performs both multiply-and-accumulate and modified batch normalization in the analog domain. It reported 83.27% test accuracy for the CIFAR-10 data set. In this design, each column end is binarized with a single sense amplifier. Therefore, it cannot directly support the operations that require more than binary precision, such as max pooling.

Manuscript received July 29, 2019; revised November 11, 2019 and November 12, 2019; accepted December 19, 2019. Date of publication January 14, 2020; date of current version May 27, 2020. This work was supported in part by NSF under Grant 1652866, in part by the Wei Family Private Foundation, in part by the Catalyst Foundation, and in part by the Center for Brain-Inspired Computing (C-BRIC), one of six centers in JUMP, a Semiconductor Research Corporation (SRC) Program sponsored by the Defense Advanced Research Projects Agency (DARPA). *(Corresponding author: Mingoo Seok.)*

Shihui Yin and Jae-Sun Seo are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: syin11@asu.edu; jaesun.seo@asu.edu).

Zhewei Jiang and Mingoo Seok are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: zj2139@columbia.edu; ms4415@columbia.edu).

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

Digital Object Identifier 10.1109/JSSC.2019.2963616

Twin-8T [25] employed two of the conventional 8T SRAM structures [28]. Supporting multi-bit CNNs, it achieves the accuracy of up to 90.42% for CIFAR-10. This design can simultaneously turn on 9/18 rows in single-/dual-channel mode, achieving the energy efficiency of 37.5/72.1 TOPS/W.

Compute SRAM [29] employs transposable SRAM [30] and implements bit-serial digital operations near the peripherals. The proposed "digital" computing scheme avoids less robust analog computation but allows to turn on only two rows of bitcells in one clock cycle, resulting in limited throughput and energy efficiency.

While prior in-SRAM computing works have made different design decisions, we focus on robust and scalable IMC with the goal to further advance the tradeoff between the DNN accuracy and energy efficiency. Unlike some of the prior works [22], [23] that connect the drain/source of additional transistors directly to the SRAM storage nodes, we only connect the gate of additional transistors. This is critical to eliminate any write disturb when all rows are asserted. Also, unlike [22] and [23] that prematurely binarize the analog bitline voltage with a single sense amplifier, we employ a multi-bit ADC. This enables scalability to arbitrary-sized DNNs.

In this article, we propose an in-memory mixed-signal SRAM macro titled "XNOR-SRAM" that not only energy efficiently computes ternary-XNOR-and-accumulate (XAC) in binary/ternary DNNs but also supports the DNNs/CNNs of arbitrary size with high accuracy. Our XNOR-SRAM performs a 256-input XAC without explicit memory readout, via analog accumulation of bitwise ternary-XNOR results on the read bitline (RBL) voltage of the SRAM array, and digitizes the RBL voltage (V_{RBL}) using a flash ADC embedded in the periphery. XNOR-SRAM supports binary weights ($+1$, -1) and binary inputs ($+1$, -1) as well as ternary inputs ($+1$, 0, -1).

Our 65-nm prototype chip achieves 300× better energy–delay product (EDP) than a digital baseline in computing XAC. DNN classification using our XNOR-SRAM achieves 98.3%/98.8% accuracy for MNIST and 87.3%/88.8% accuracy for CIFAR-10 using binary/ternary precision. This article is an extended version of [24], providing the detailed design space exploration, optimizations to compensate variability, and additional measurement results with five chips.

II. XNOR-SRAM Macro Design and Optimization

A. XNOR-SRAM Bitcell Design

Fig. 1 shows the proposed XNOR-SRAM architecture, which can map convolutional and fully connected (FC) layers of CNNs and multi-layer perceptrons (MLPs). It consists of a 256-by-64 custom bitcell array, a row decoder, an XNOR-mode WL driver, and a column periphery, including a 3.46-bit flash ADC. The XNOR-SRAM operates in either of two modes: memory mode and XNOR mode. In the memory mode, it performs row-by-row digital read/write as regular SRAM. In the XNOR mode, it performs in-memory XAC computation with all rows asserted simultaneously.

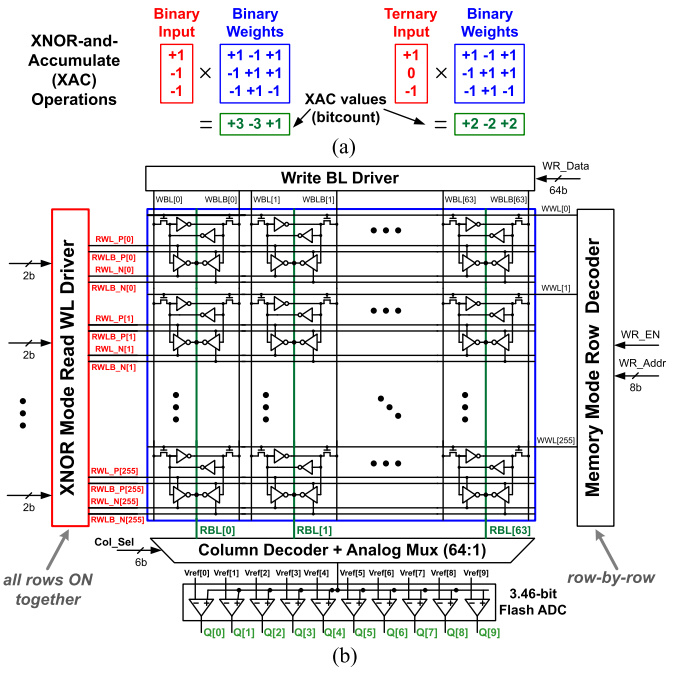


Fig. 1. (a) XAC operation illustration. (b) Proposed XNOR-SRAM macro architecture.

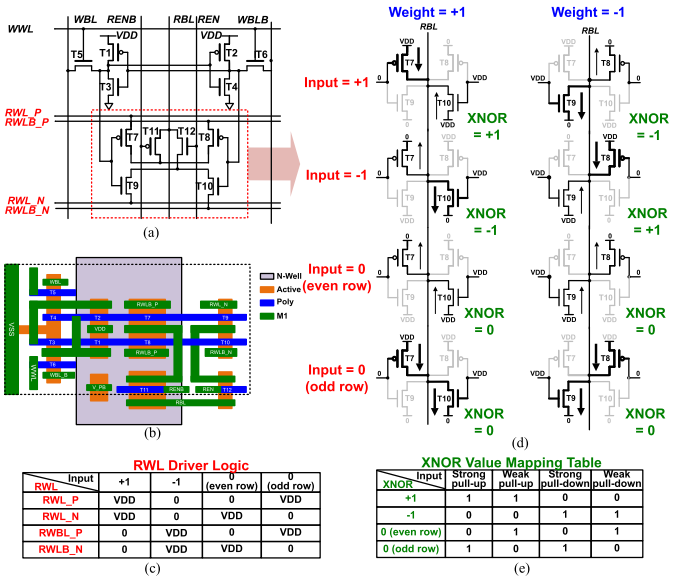


Fig. 2. XNOR-SRAM bitcell design and XNOR-ACC operation with ternary inputs/activations and binary weights. Bitwise ternary-XNOR output from each bitcell forms pull-up (PU)/pull-down (PD) paths on the RBL voltage, which represents the XNOR-ACC value.

Fig. 2(a) shows the proposed 12T bitcell for XNOR-SRAM. T1–T6 form a 6T cell, T7–T10 form complimentary PU and PD circuits for XNOR mode (and memory mode read), and T11 and T12 power-gate the PU/PD circuits when the corresponding column is disabled. Fig. 2(b) shows the layout of the bitcell drawn in logic ground rules. The area is 3.915 μm^2 (2.7 $times$- 1.45 μm). Except T7, T8, and T11, all transistors in the bitcell use the minimum size. We slightly sized up the PMOS transistors T7, T8, and T11 to match its strength to their NMOS counterparts.

In the XNOR mode, the read wordline (RWL) driver translates each ternary/binary input activation to four RWLs according to Fig. 2(c). In the second half of a clock cycle, T11 and T12 in a selected column are turned on, and T7–T10 perform the ternary-XNOR operation between the RWLs (activations of +1, 0, or −1) and the binary weight (+1 or −1) stored in the bitcell. The RBL voltage finally settles and is read by the flash ADC.

B. Proposed IMC Operation and Analysis

1) Binary Activations and Binary Weights: For binary activations, the bitcell produces the XNOR output of "+1" with one strong PU by PMOS and one weak PU by NMOS. It produces the XNOR output of "−1" with one strong PD by NMOS and one weak PU by PMOS. This operation is summarized in the first two rows of Fig. 2(d). The 256 bitcells in a column contribute such XNOR-output-controlled PU and PD circuits and essentially form a resistive voltage divider from the supply voltage to the ground, where RBL is the output. If PU and PD resistances are identical, the RBL voltage (V_{RBL}) will be a symmetric and monotonic function of the XAC value. In practice, they are different due to process variations. Our design is capable of correcting this non-ideality by tuning the PMOS body bias of the bitcell array, which is made as a separate pin in our prototype chip (see more details in Fig. 14).

The first-order analysis on the relationship between XAC value and RBL voltage is as follows. If the number of rows is N, the range of XAC is from $-N$ to $+N$. Suppose that u is the number of PU cells among N cells in a column and d is the number of PD cells. As shown in (1), we can represent N as the sum of u and d. Given that each PU and PD cell represents the bitwise XNOR output of "+1" and "−1," respectively, the XAC result that accumulates all cells' XNOR outputs is formulated as (2). As shown in Fig. 2(c), the bitwise XNOR output of "+1" and "−1" results in two PU and two PD paths, respectively. This is shown in Fig. 3, and V_{RBL} can be represented as (3) with the resistive divider. Using (1) and (2), V_{RBL} can be formulated as (4), showing a linear relationship with XAC value. Note that V_{RBL} is not affected by the activation/weight patterns as long as they result in the same the XAC bitcount

$$N = u + d \tag{1}$$

$$XAC = u - d \tag{2}$$

$$V_{\mathrm{RBL}} = \frac{2u}{2u + 2d} \tag{3}$$

$$V_{\mathrm{RBL}} = \frac{XAC + N}{2N} \tag{4}$$

2) Ternary Activations and Binary Weights: To support ternary activations, we have additionally considered the activation value of "0" and have derived the equations that are similar to (4). Suppose that the number of cells that exhibit the bitwise ternary-XNOR output of "0" as z, N would be the sum of u, d, and z ((5)). Since those z bitcells do not contribute to the XAC output, the equations for the XAC value are identical for the binary and ternary activation case

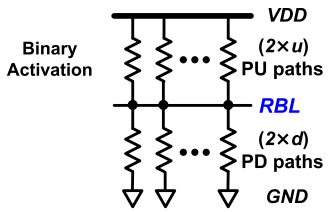


Fig. 3. PU/PD paths for V_{RBL} with binary activations.

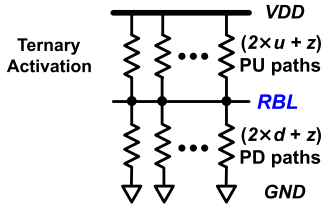


Fig. 4. PU/PD paths for V_{RBL} with ternary activations.

[see (2) and (6)]. To maintain the same linear relationship between V_{RBL} and N as in (4), the z bitcells should contribute z PU and z PD circuits. This is shown in Fig. 4

$$N = u + d + z \tag{5}$$

$$XAC = u - d \tag{6}$$

$$V_{\mathrm{RBL}} = \frac{2u + z}{2u + 2d + 2z} \tag{7}$$

$$V_{\mathrm{RBL}} = \frac{XAC + N}{2N}. \tag{8}$$

Since each u and d cell leads to $2u$ PU paths and $2d$ PD paths (one strong plus one weak), each z cell should ideally yield the average strength of the u and d cells or 0.5 strong PU + 0.5 strong PD + 0.5 weak PU + 0.5 weak PD. However, since T9 and T10 (T7 and T8) use (close to) minimum size, splitting T7–T10 transistors to support such half/full strengths will complicate and enlarge the XNOR-SRAM bitcell design by about 50% and double the current consumption. The bitcell-embedded ternary XNOR computation and operation are summarized in Fig. 2(c). Note that having z cells to exhibit no PU and PD paths (i.e., turning off "0" activation rows) will make (8) deviate from (4) and introduce further difference in V_{RBL} depending on the number of "0" activation rows. Without changing the bitcell design that implements binary activations and weights, we propose to drive even "0" rows with weak PU/PD and odd "0" rows with strong PU/PD [see Fig. 2(e)], to effectively support ternary activations. This design is based on the assumption that "0" activations are evenly distributed on even and odd rows. Deviation from this assumption, i.e., the number of even-row zeros and odd-row zeros are not equal, would cause V_{RBL} deviation. According to our post-layout simulation with parasitics annotated, the V_{RBL} variance caused by the mismatch in these two numbers in the ternary VGG-like and ResNet CNNs (for CIFAR-10) and MLPs (for MNIST) that we benchmarked is negligible compared to other variability sources, such as transistor mismatch in the XNOR-SRAM cells.

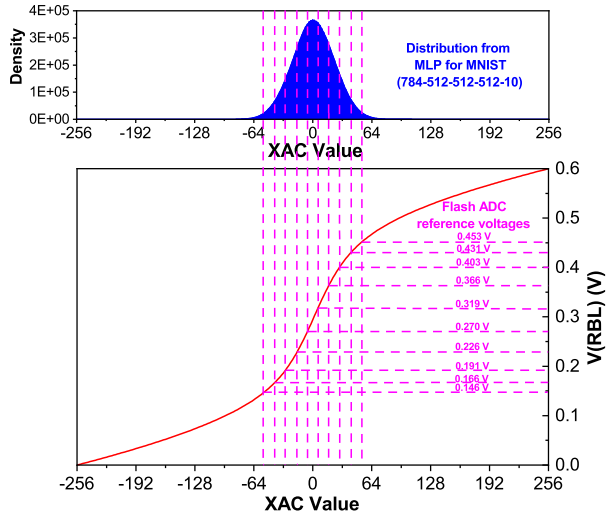
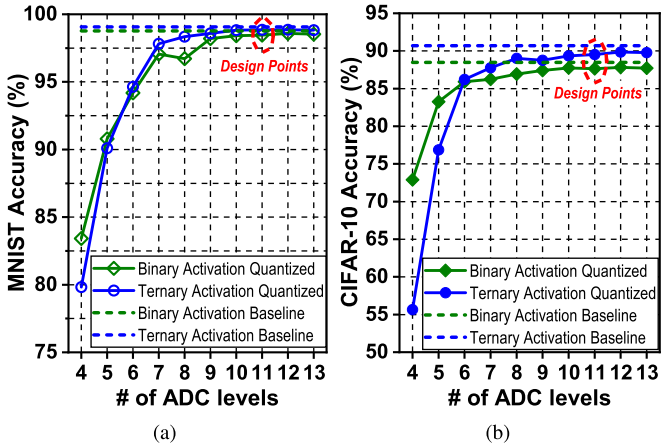


Fig. 5. XAC is mapped to V_{RBL}. The confined linear quantization scheme is shown, along with the corresponding ten reference voltages (V_{ref}) for the 11-level flash ADC.



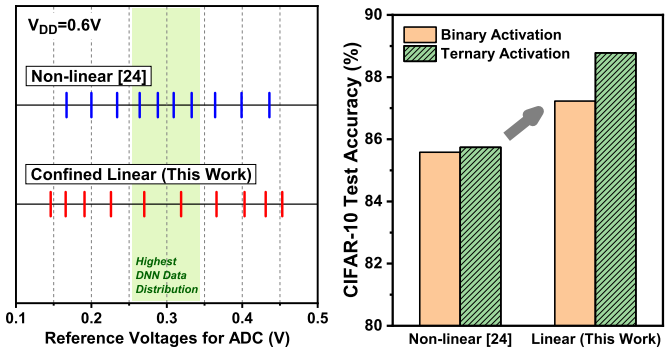
(a) (b)

Fig. 6. (a) Accuracy of the MLP trained for MNIST and (b) accuracy of the CNN for CIFAR-10 as a function of ADC levels.

C. Transfer Function and ADC Optimization

The ADC plays an important role in the computing throughput and accuracy. It digitizes the analog RBL voltage to the digital output. We chose to use the flash ADC using strong-arm comparators for the high-speed advantages. We shared the ADC among 64 columns via a 64-to-1 analog multiplexer for two reasons. First, the RWL drivers could be considerably large to support the column-parallel operation as the drivers need to supply the current flowing in the resistor dividers. Second, the 64 ADCs would incur a large overhead for the ADC area. As pointed out in [31], the column multiplexing scheme would not degrade the energy efficiency in the first order, since the amount of voltage switching on all the wordlines and bitlines is roughly the same for both column multiplexing and column-parallel schemes. Nonetheless, the column multiplexing in our design hurts the throughput by roughly 64 times compared to a fully column-parallel design.

We investigate the distribution of XAC values. As the data distribution from MLP for MNIST shows (see Fig. 5), the XAC value is highly concentrated around zero. Exploiting such statistics, we confined the quantization range to the region that covers most data (-60 to $+60$), within which we linearly divided the quantization levels with reference values. Each quantization reference in the XAC value maps to a particular reference voltage (V_{ref}) for the flash ADC (see Fig. 5). Note that the non-linearity of the PD and PU resistance makes the V_{RBL} transfer function non-linear, placing V_{ref}s non-uniformly, as shown in Fig. 5.



(a) (b)

Fig. 7. Comparison of non-linear quantization [24] and confined linear quantization (this work). (a) Comparison of V_{ref}s for ADC. (b) CIFAR-10 accuracy improvement.

We investigated the required ADC precision based on the MLP for MNIST (784-512-512-512-10) and the VGG-like CNN for CIFAR-10 (128C3-128C3-MP2-256C3-256C3-MP2-512C3-512C3-MP2-1024FC-1024FC-10FC). Fig. 6 shows the simulation results across the different numbers of ADC levels. This simulation ignores analog non-ideality, such as offset voltage and transistor variability. We have found that employing 11 quantization levels (i.e., 3.46 bits) results in satisfactory accuracy. In addition, the accuracy saturates for the ADC levels beyond 11. Based on these results, we have designed the 11-level flash ADC, which consists of ten strong-arm comparators.

Note that in [24], we employed a non-linear quantization scheme based on the Llyod–Max algorithm [32]. This scheme produces finer grain reference levels where more data exist (i.e., XAC $\simeq$ 0). However, we have found that this made the difference between two adjacent V_{ref}s to be very small, increasing the error associated with ADC. In addition, the non-linear quantization scheme does not consider the difference of RBL voltage variance for different XAC values. We have found that near-zero XAC values have larger RBL voltage variance and ideally require wider ADC quantization intervals. As shown in Fig. 7, the difference of two adjacent V_{ref}s near the XAC value of zero with the confined linear quantization scheme increases from 21 mV [24] to 49 mV in this article (at 0.6-V supply). As shown later, this helps to improve the CIFAR-10 accuracy from 85.7% in [24] to 88.8%.

III. Measurement Results

We prototyped the proposed XNOR-SRAM macro in a 65-nm CMOS [see Fig. 8(a)]. The area and power breakdowns

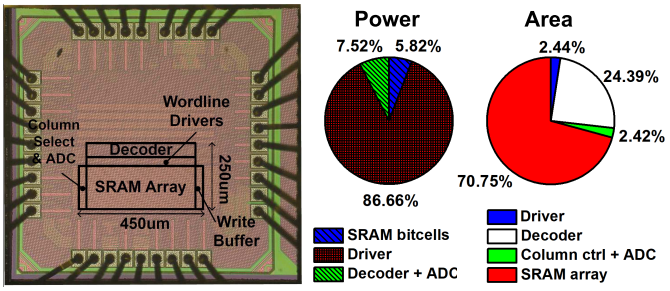


Fig. 8. (a) 65-nm XNOR-SRAM prototype chip micrograph. (b) Power and area breakdown.

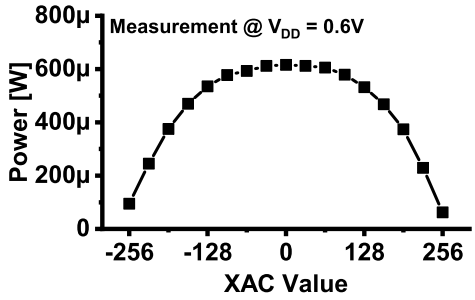


Fig. 9. Data-dependent XNOR-SRAM power.

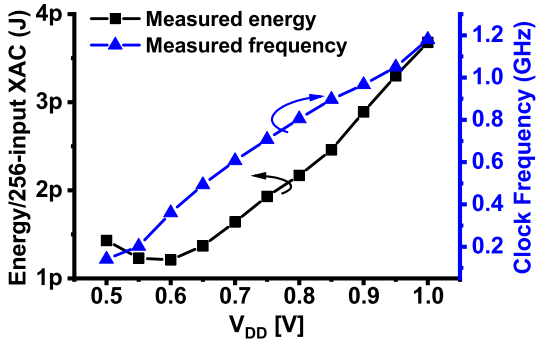


Fig. 10. Energy and frequency scaling with supply voltage.

are presented in Fig. 8(b). The area of XNOR-SRAM is majorly consumed by the bitcell array, where the array efficiency is 70.75%. On the other hand, the XNOR-mode driver dominates the total power as it needs to supplies the current of the restive voltage divider formed for XAC evaluation.

A. Energy Consumption and Performance Measurements

We have measured the power and energy dissipation of the XNOR-SRAM macro under a range of conditions. First of all, the power consumption depends on the XAC result (see Fig. 9). This is because most of the power is consumed in the form of the crowbar current in the resistive voltage divider. The input data that correspond to XAC value near 0 pose the worst case for energy efficiency. The post-layout simulation shows that the worst case crowbar current is 1 mA and lasts for 1.26 ns in the second half of a clock cycle at 0.6 V. Under this worst case, we measured that XNOR-SRAM consumes 235.5 pJ and takes 54.21 ns for 64 operations of 256-input XAC at 1.0 V. Fig. 10 shows the energy and the maximum

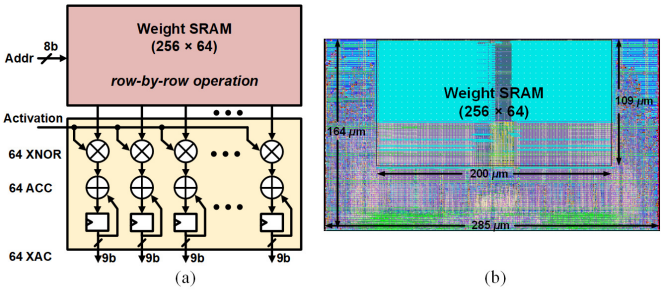


Fig. 11. Digital baseline design for XAC accelerator. (a) Block diagram. (b) Layout in 65-nm CMOS.

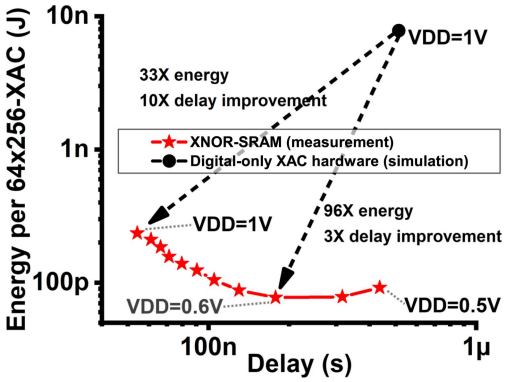


Fig. 12. Energy and delay comparison with digital baseline.

frequency with the voltage scaling from 1.0 to 0. 5 V. At 0.6 V, XNOR-SRAM achieves 2.48 fJ per operation. Considering that one operation is either ternary multiplication or accumulation, this marks the energy efficiency of 403 TOPS/W.

B. Comparison to a Digital Baseline Design

As a comparison, we designed a well-crafted digital baseline in the same technology. It performs the exact same function of the XNOR-SRAM but uses digital XNOR gates, digital adders, and conventional SRAM (see Fig. 11). This baseline hardware reads 64 binary weights row by row, receives 256 binary/ternary inputs also one by one, and finally accumulates 64 XAC values over 256 cycles. The SRAM for weight memory was generated by a commercial memory compiler. The digital design was synthesized and automatically placed/routed using commercial EDA tools [see Fig. 11(b)]. The digital baseline achieves 500-MHz clock frequency. The post-layout simulation based on parasitic-annotated netlists at 1.0-V supply (typical corner, 25°C) shows that the digital baseline consumes 7.81 nJ and 514 ns for the same 64 256-input XAC operations, which represents 33× worse energy and ∼300× worse EDP than the XNOR-SRAM macro also operating at 1.0 V (see Fig. 12). A similar EDP gain of ∼300× is achieved for XNOR-SRAM down to 0.6-V supply. Note that the XNOR-SRAM prototype chips were functional down to 0.5-V supply, and however, additional energy/EDP savings were not achieved below 0.6 V, because the circuit delay increases rapidly as the supply voltage gets closer to the near-threshold voltage. During this increased cycle time,

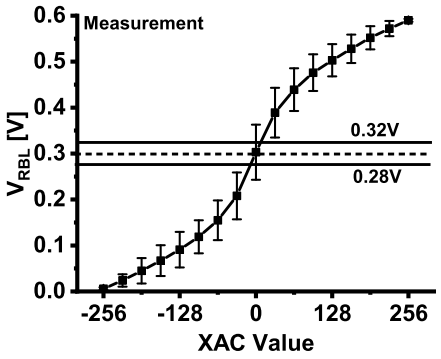


Fig. 13. Measured transfer function and variability.

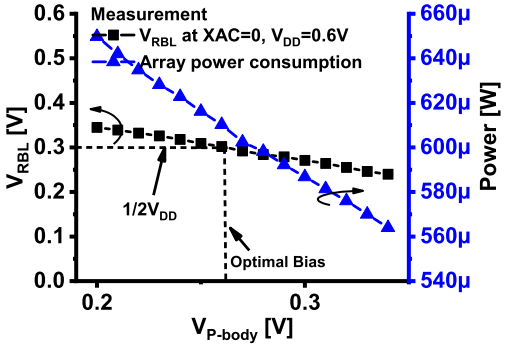


Fig. 14. Body bias tuning for PMOS/NMOS mismatch.

the XNOR-SRAM consumes more energy from leakage and crowbar current.

C. Variability and Compensation

Fig. 13 shows the measured V_{RBL} variability resulting from process variation and parasitics across different columns and data patterns, where the top and bottom bars represent $+3\sigma$ and -3σ points, respectively. The highest variation (20-mV standard deviation) occurred at the lowest XAC value of 0. The systematic strength imbalance between NMOS and PMOS can skew the transfer function. We addressed this by providing a knob to tune the body bias for PMOS transistors in the XNOR-SRAM array at marginal area/power penalty (see Fig. 14).

To compensate for the local variability or offset of the ADC, ten external V_{ref}s for the 11-level (3.46-bit) flash ADC were first calibrated for the corresponding ten reference XAC values. For each reference XAC value X_f, 2000 combinations of random input vectors, columns, and weight matrix that yield XAC values that fall in the range of $[X_f - 5, X_f + 5]$ were used to optimize each comparator's V_{ref}. V_{ref} was initialized at $0.5 \times V_{\mathrm{DD}}$. After each combination, the actual comparator output was compared to the ideal output. If it was correct, no change was made to the reference voltage; otherwise, a small and exponentially decayed correction amount will be added to or subtracted from the current reference voltage, depending on whether the ideal output is low or high. Each chip undergoes this type of reference voltage calibration process before performing the IMC operation.

TABLE I

V_{RBL} VARIANCE OF A SINGLE COLUMN AT 1.0- AND 0.6-V SUPPLY EXTRACTED FROM POST-LAYOUT MONTE CARLO SIMULATIONS

VDD [V]	ΔV_{RBL} [mV] from mismatch	ΔV_{RBL} [mV] from IR drop	ΔV_{RBL} [mV] from both
1.0	2.71	36.4	36.8
0.6	7.09	6.06	9.33

After compensation of the ADC offset, there are two remaining major variability sources: 1) the transistor mismatch in the XNOR-SRAM cells and 2) the IR drop on RBL wires. The transistor mismatch causes the bitcells in the different rows but in the same column have different PU/PD strength (after the array-wide body bias calibration), making RBL voltage depend on the input/weight pattern even for the same XAC value. On the other hand, the RBL IR drop is a function of input/weight patterns.

We performed the Monte Carlo simulations for the parasitic-annotated netlist of a single column of bitcells to characterize these two variability sources. We used 1000 random combinations of input/weight vectors that result in the XAC value of 0. To isolate the impact of each variability source, in our extracted simulations, we included only the mismatch for the cell transistors, only the extracted resistance of the RBL, and both two variability sources. Table I summarizes the results of the post-layout simulations. We can see that at 1 V, as the current is very large, the IR drop along the RBL contributes most to the overall variation of V_{RBL}. At 0.6 V, variation due to IR drop significantly decreases as the current decreases, reducing the overall amount of variation to just a quarter of that at 1 V.

D. Statistical Model of XNOR-SRAM and Voltage Scaling

We developed the statistical model of XNOR-SRAM as a function of the XAC value. To do so, we measured the ADC output for 1600 times for each XAC value, 25 times per column. Each time a random test vector that will result in the target XAC value for a given column is generated. Based on 1600 measured ADC outputs for each XAC value, we estimated the probability distribution of the ADC output as function of the XAC value (see Fig. 15). We iterated this experiment at four different supply voltages of 1.0, 0.8, 0.6, and 0.5 V.

Counter-intuitively, Fig. 15 shows that, as supply voltage lowers, the ADC output distribution becomes tighter. To see it clearly, we can model the RBL voltage V_{RBL} as a function of XAC value X and V_{DD} as

$$V_{\mathrm{RBL}}(X, V_{\mathrm{DD}}) = \bar{V}_{\mathrm{RBL}}(X, V_{\mathrm{DD}}) \pm \Delta V_{\mathrm{RBL}}(X, V_{\mathrm{DD}}) \quad (9)$$

where $\bar{V}_{\mathrm{RBL}}(X, V_{\mathrm{DD}})$ represents the average V_{RBL} for given XAC value X under supply voltage V_{DD} over all possible combinations of input and weight vector and $\Delta V_{\mathrm{RBL}}(X, V_{\mathrm{DD}})$ represents the standard deviation of the actual V_{RBL} over all possible combinations of input and weight vector for given XAC value X. ADC's V_{ref}s are calibrated against $\bar{V}_{\mathrm{RBL}}(X, V_{\mathrm{DD}})$ as aforementioned. ADC quantization error is

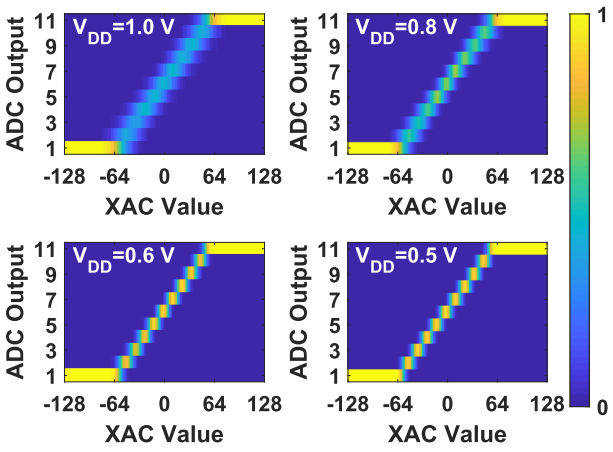


Fig. 15. Measured ADC output probability distribution as a function of XAC value at V_{DD} of 1.0, 0.8, 0.6, and 0.5 V.

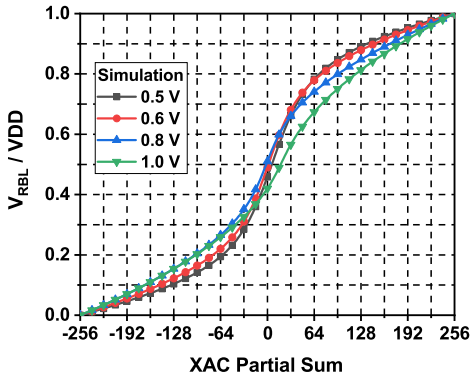


Fig. 16. Normalized transfer function at different values of V_{DD}.

then governed by the distribution of $\Delta V_{RBL}(X, V_{DD})$ and quantization scheme. The reduced ADC quantization error at lower V_{DD} can be explained from two aspects: reduced $\Delta V_{RBL}(X, V_{DD})/V_{DD}$ and enhanced normalized slope of transfer function.

1) Voltage Scaling of RBL Voltage Variance: As shown in Table I, according to our post-layout simulation on a single column of XNOR-SRAM array, RBL voltage variance at $X = 0$ reduces from 36.8 to 9.33 mV when we scale V_{DD} from 1.0 to 0.6 V. This reduction in RBL voltage variance majorly comes from the reduction of variance contribution from IR drop along the RBL, which is a result of reduction in current.

2) Normalized Slope of Transfer Function: As V_{DD} decreases, the transfer function slope in the near-zero region increases when normalized to V_{DD}, as shown in Fig. 16. As a result, two adjacent XAC values will be more separated in terms of V_{RBL}/V_{DD}, tolerating larger variance in V_{RBL}/V_{DD}.

Combining the above-mentioned two aspects, as V_{DD} decreases, the enhanced normalized slope of V_{RBL} transfer function and the reduced variance of V_{RBL} lead to the reduced ADC quantization error, as shown in Fig. 15.

E. Strategy for Mapping DNNs onto XNOR-SRAM Arrays

Mapping convolution layers of deep CNNs onto XNOR-SRAM arrays is shown in Fig. 17(a). We propose

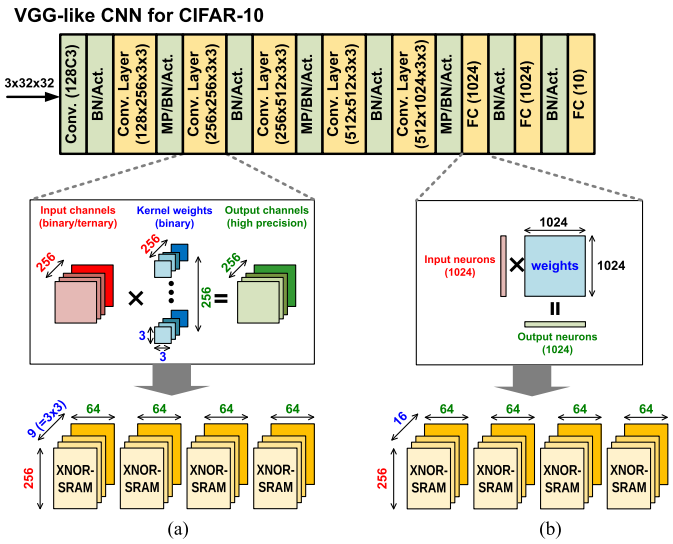


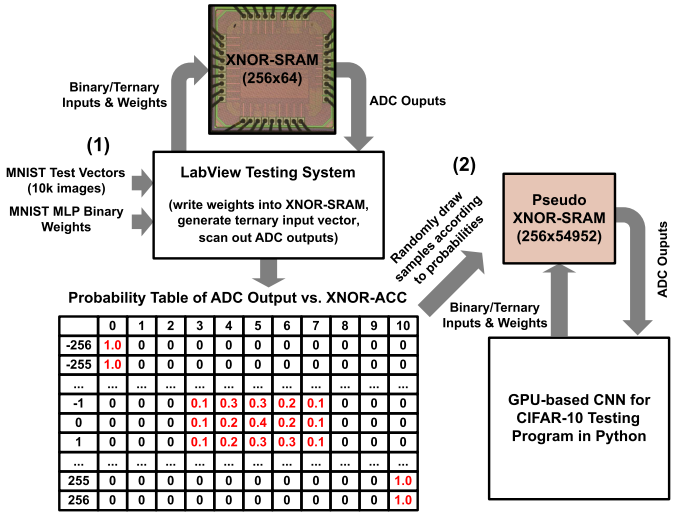
Fig. 17. Mapping (a) convolution layers and (b) FC layers of deep CNNs onto XNOR-SRAM arrays.

to use a mapping strategy where kernels for separate input channels are stored on separate rows, kernels for separate output channels are stored on separate columns, and weights within each kernel (e.g., nine weights for 3×3 kernel) are stored in separate XNOR-SRAM macros. The XAC results (partial sums) obtained from separate macros are accumulated digitally, to generate the final sum results for each neuron. This scheme extensively re-uses the input activations, with stationary weights in the XNOR-SRAM arrays.

As shown in Fig. 17(b), it is straightforward to map FC layers of DNNs, where activations are in vectors and weights are in matrices. This nicely maps to the row drivers for activations and weights stored in the XNOR-SRAM. For the FC layers whose size is larger than 256×64, we break the large weight matrix into a number of small sub-matrices (that fit XNOR-SRAM macros) and accumulate the matrix-vector multiplication results accordingly.

F. DNN Accuracy Characterization

Using the XNOR-SRAM macro, we evaluated the accuracy of DNNs for MNIST and CIFAR-10 data sets. For MNIST, an MLP with three hidden layers, each with 512 neurons, is used (784-512-512-512-10). For CIFAR-10, we evaluated two deep CNNs: VGG-like CNN and ResNet-14. VGG-like CNN [13] has six convolutional layers and three FC layers: 128C3-128C3-MP2-256C3-256C3-MP2-512C3-512C3-MP2-1024FC-1024FC-10FC, where nC3 represents a convolutional layer with n 3×3 filters, mFC is an FC layer with m neurons, and MP2 is a max-pooling layer with 2×2 pooling size. ResNet-14 [13], [33] consists of three basic residual blocks (block widths of 80, 160, and 320), with a total of 13 3×3 convolution layers, two 1×1 convolution layers in short-cut paths (not counted for the number of layers), and one FC layer. Starting from the first hidden layer of the MLP/CNN, XNOR-SRAM computes 256-input XACs for MAC/convolution operations in all convolution/FC layers



	0	1	2	3	4	5	6	7	8	9	10
-256	1.0	0	0	0	0	0	0	0	0	0	0
-255	1.0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
-1	0	0	0	0.1	0.3	0.3	0.2	0.1	0	0	0
0	0	0	0	0.1	0.2	0.4	0.2	0.1	0	0	0
1	0	0	0	0.1	0.2	0.3	0.3	0.1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
255	0	0	0	0	0	0	0	0	0	0	1.0
256	0	0	0	0	0	0	0	0	0	0	1.0

Fig. 18. Measurement-based simulation framework for CIFAR-10 accuracy evaluation using XNOR-SRAM macros.

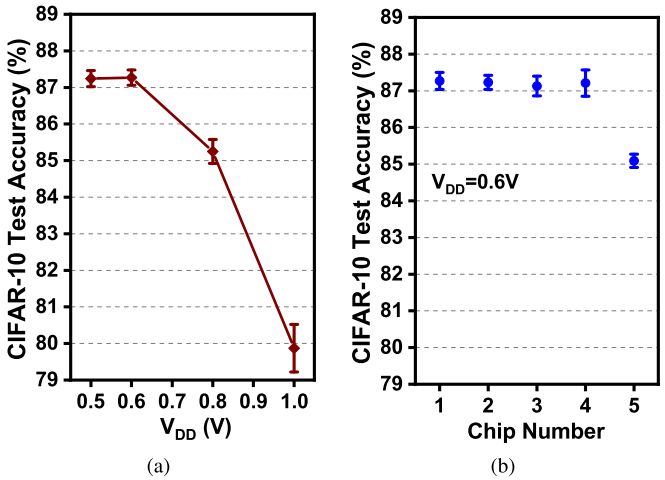


Fig. 19. Accuracy characterization of binary CNN for CIFAR-10. (a) Accuracy for chip #1 across different supply voltages. (b) Accuracy distribution for five different chips at 0.6 V.

TABLE II
MEASURED MLP (FOR MNIST) AND CNN (FOR CIFAR-10) ACCURACY SUMMARY AT 0.6-V SUPPLY

Dataset	MNIST		CIFAR-10			
DNN Model	MLP		VGG-like CNN		ResNet-14 CNN	
Activation Precision	Binary	Ternary	Binary	Ternary	Binary	Ternary
SW Baseline Accuracy	98.77%	99.07%	88.60%	90.70%	89.61%	90.80%
HW Measured Accuracy	98.65% (-0.12%)	98.84% (-0.23%)	87.23% (-1.37%)	88.78% (-1.92%)	85.72% (-3.89%)	87.05% (-3.75%)

(e.g., all yellow-colored layers in Fig. 17). Accumulation of XAC outputs, pooling, and batch normalization is performed in digital simulation with the bit precisions of 12, 12, and 10, respectively. Note that the digital hardware that executes these functions would degrade the overall energy efficiency to some extent. Recently, several works have tried to shed a light on this matter [34].

The MNIST accuracy results were obtained entirely from measurements, while the CIFAR-10 accuracy results were obtained from our measurement-based simulation framework (shown in Fig. 18) due to the limited scan chain throughput of the prototype chip. Employing the same methodology used to generate the probability distribution in Fig. 15, ADC output distributions for each possible XAC value were estimated from the measured samples from 10k MNIST test images MLP inference. Each column was sampled from the probability table, obtaining an ADC output for each bitcount, and then, this mapping was kept for all the 10k CIFAR-10 test images. The measured distributions were then used to draw random samples in a GPU-accelerated Python program that simulates XNOR-SRAM XAC and quantization operations for inference of 10k CIFAR-10 test images using trained binary CNN [13]. Our Python simulation program repeated 20 runs with different random seeds, and the average accuracy values are reported.

Table II summarizes the measured accuracy results of MLP for MNIST and VGG/ResNet-14 CNNs for CIFAR-10 data sets with binary/ternary activations at 0.6-V chip #2. It is hypothesized that ResNet-14 CNN has more accuracy degradation than VGG CNN compared to the software baselines, because ResNet-14 CNN is deeper and requires higher partial sum precision to maintain high accuracy. It can be seen that DNNs with ternary activations demonstrate a relatively higher accuracy than those with binary activations for both MNIST and CIFAR-10 data sets, and our XNOR-SRAM can execute both ternary and binary activations in a single cycle with the same design change (see Section II-B).

Fig. 19 shows the accurate characterization of CNN for CIFAR-10 with binary activations/weights across different supply voltages and different chips, where top and bottom bars represent $+\sigma$ and $-\sigma$ points, respectively. With tighter ADC output distribution for XAC values at lower V_{DD} (see Fig. 15), the CNN accuracy improves as we lower V_{DD}, down to 0.6 V [see Fig. 19(a)]. Fig. 19(b) shows the accuracy distribution of five different chips at 0.6-V supply, where most chips exhibit a relatively constant mean accuracy of $>87\%$. Note that chip 5 achieves lower accuracy compared to the other four chips we measured. While we are in lack of sufficient access to the internal signals of the prototype chip, we still found that chip 5 behaves similarly to other chips except that the ADC outputs exhibit larger errors compared to those of other chips.

G. Ensemble Networks for Accuracy Improvement

In the deep learning literature, ensemble neural networks have been commonly used to improve classification accuracy [1], [2], [35]–[37], where a collection of networks (typically less than 10) are trained separately with different initial random weights or training data, and the final prediction is calculated as the average of the predictions of all networks models. The reason that ensemble methods improve accuracy is that different models will usually not make the same errors on the test set [38].

Inspired by such ensemble algorithms, we investigated combining multiple XNOR-SRAM arrays from the same

TABLE III
COMPARISON WITH THE RECENT IN-SRAM COMPUTING WORKS

	Biswas et al. [20]	Valavi et al. [23]	Si et al. [25]	Digital XAC Baseline	This work
Technology	65nm	65nm	55nm	65nm	65nm
SRAM bitcell	10T	9T1C	8T_MSB 8T_LSB	Off-the-shelf 6T	12T
SRAM Array size	256x64	64x64, 64tiles	64x60	256x64	256x64
# of rows turned on simultaneously	16	64	9/18	1	256
Supply voltage	0.8-1.0V	1.0V	1.0V	1.0V	0.6-1.0V
Column sensing	7b Integrating ADC	SA + DAC	5b SAR ADC	Digital adder	3.46b Flash ADC
Energy-efficiency (operation)	51.3 TOPS/W (A: 6b, W: 1b, mult./ avg.)	658 TOPS/W (binary XNOR-Acc.)	18.4-71.9 TOPS/W (A: 1/2/4b, W: 2/5/5b)	4.2 TOPS/W (ternary XNOR-Acc.)	403 TOPS/W (ternary XNOR-Acc.)
GOPS/mm^2	57	1498	Not reported	1369	5461
MNIST accuracy	98.0%	98.58%	99.02%/99.18%/99.52% (A: 1/2/4b, W:2/5/5b)	99.07%	98.84%
CIFAR-10 accuracy	N/A	84.09%	85.56%/90.2%/90.42% (A: 1/2/4b, W: 2/5/5b)	90.70%	88.78%

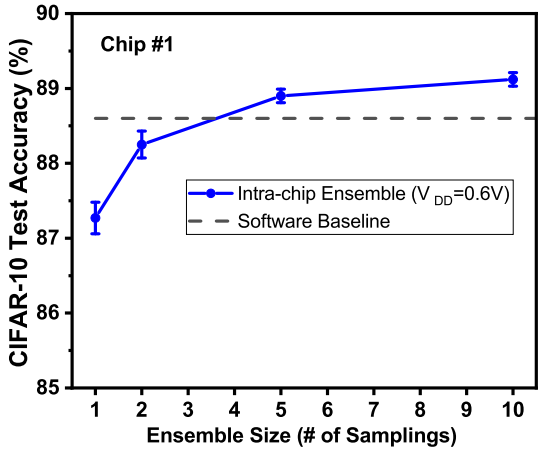


Fig. 20. Binary DNN accuracy improves with ensemble hardware exploiting intra-die variation.

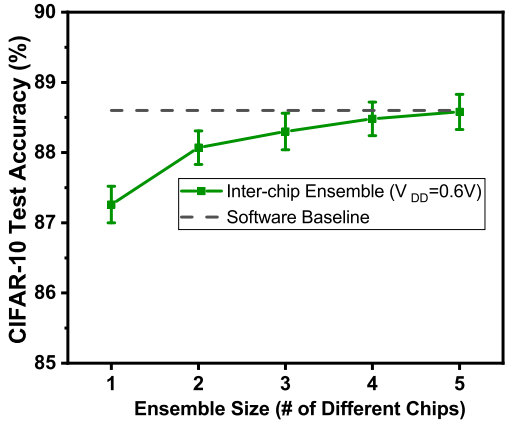


Fig. 21. Binary DNN accuracy improves with ensemble hardware exploiting intra-die variation from five different chips.

chip or different chips, in order to possibly improve the XNOR-SRAM hardware accuracy. Compared to the software baseline, the mixed-signal XNOR-SRAM-based DNN accuracy was degraded due to intra-/inter-column transistor mismatch, RBL resistance difference, and ADC offsets. Compared to the ensemble algorithms, the main difference in our proposed ensemble hardware for XNOR-SRAM is that we do not train different DNNs; instead, we use identical DNNs but exploit the hardware variability as the source of the difference in prediction errors.

First, we used the measurement data from a single chip (chip #1) and randomly sampled the probability table in Fig. 18 and evaluated the CNN accuracy for CIFAR-10. To test the ensemble hardware exploiting intra-die variation, we iterated this procedure for 2, 5, and 10 times (ensemble size) with different random sampling and averaged the values of ten neurons of the last output layer from ensemble hardware for classification. This intra-chip ensemble hardware accuracy results are shown in Fig. 20, where considerable accuracy

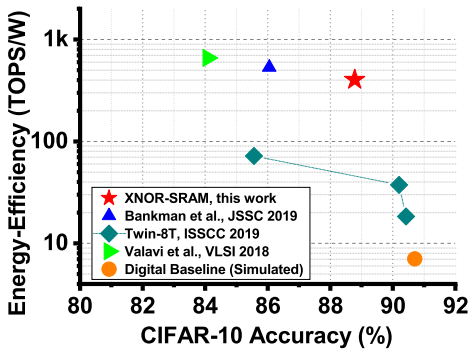


Fig. 22. Energy efficiency (TOPS/W) and CIFAR-10 accuracy comparison against in-/near-memory computing literature.

improvement is achieved while trading off area and energy. For ensemble sizes of 5 or larger, the CNN hardware accuracy values are even higher than the software baseline accuracy.

Next, we used the measurement data from five different chips and experimented to ensemble a different number of

chips to exploit inter-die variation. Even though chip #5 exhibits somewhat lower CNN accuracy than other chips [see Fig. 19(b)], Fig. 21 shows that the CNN accuracy for CIFAR-10 continuously improved when we increasingly used more chips. In Figs. 20 and 21, top/bottom bars represent $+\sigma/-\sigma$ points.

H. Comparison

Fig. 22 shows the comparison of energy efficiency (TOPS/W) and CIFAR-10 accuracy against prior IMC works. The neural network architecture of ours (XNOR-SRAM), digital baseline, and [23] is the same. Reference [16] uses a simplified version (2×2 convolution filters and so on) of the same network. Reference [25] uses a different residual CNN (ResNet) for the same data set of CIFAR-10. Compared to [23], energy efficiency is $\sim$38% lower, but CIFAR-10 accuracy is significantly higher by 4.7%. Compared to [25], the CIFAR-10 accuracy is 1.4% lower, but the energy efficiency is $>10\times$ higher. Table III shows a more detailed comparison to recent in-SRAM computing works [20], [23], [25] and the digital baseline. Our XNOR-SRAM macro improves energy/EDP by $\sim 100\times/300\times$ over digital baseline performing the same XAC operations. Compared to other in-SRAM computing works in the literature, XNOR-SRAM also demonstrates the best tradeoff in high energy efficiency and high CIFAR-10 classification accuracy.

IV. Conclusion

In this article, we present an IMC SRAM macro titled "XNOR-SRAM" that computes ternary-XAC operations in binary/ternary MLP and CNNs with high energy efficiency and high accuracy. Our 256×64 XNOR-SRAM asserts all 256 rows simultaneously, performing a 256-input ternary XAC in a single cycle, via analog accumulation of bitwise XNOR results on the RBL voltage, which is digitized using an optimized 11-level flash ADC embedded in the periphery. Our 65-nm XNOR-SRAM prototype achieves energy efficiency of 403 TOPS/W for XAC operations and 88.8% test accuracy for CIFAR-10 data set, achieving $33\times$ better energy and $300\times$ better EDP than digital ASIC baseline with off-the-shelf SRAM. Compared to non-linear quantization, the proposed confined linear quantization scheme improves the CIFAR-10 accuracy from 85.7% to 88.8% due to wider reference voltage intervals. Further accuracy improvement has been shown by employing an ensemble hardware of XNOR-SRAM arrays/chips implementing the same DNNs, exploiting intra-/inter-die variation.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016.

[3] K. J. Geras *et al.*, "High-resolution breast cancer screening with multi-view deep convolutional neural networks," 2017, *arXiv:1703.07047*. [Online]. Available: https://arxiv.org/abs/1703.07047

[4] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5934–5938.

[5] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019.

[6] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[7] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy–efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[8] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "ENVISION: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28 nm FDSOI," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 246–247.

[9] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H.-J. Yoo, "UNPU: A 50.6TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 218–220.

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

[11] J. Song *et al.*, "An 11.5TOPS/W 1024-MAC butterfly structure dual-core sparsity-aware neural processing unit in 8 nm flagship mobile SoC," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 130–132.

[12] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.

[13] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4107–4115.

[14] J. Choi, S. Venkataramani, V. Srinivasan, K. Gopalakrishnan, Z. Wang, and P. Chuang, "Accurate and efficient 2-bit quantized neural networks," in *Proc. Conf. Syst. Mach. Learn. (SysML)*, 2019.

[15] B. Moons, D. Bankman, L. Yang, B. Murmann, and M. Verhelst, "BinarEye: An always-on energy-accuracy-scalable binary CNN processor with all memory on chip in 28 nm CMOS," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2018, pp. 1–4.

[16] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8 μJ/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 158–172, Jan. 2019.

[17] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, USA, Tech. Rep., 2009, vol. 1, no. 4, p. 7.

[18] D. Wan *et al.*, "TBN: Convolutional neural network with ternary inputs and binary weights," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018.

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

[20] A. Biswas and A. P. Chandrakasan, "CONV–SRAM: An energy–efficient SRAM with in–memory dot–product computation for low–power convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 217–230, Jan. 2019.

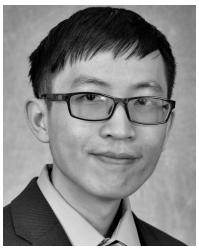
[21] S. K. Gonugondla, M. Kang, and N. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018.

[22] W.-S. Khwa *et al.*, "A 65 nm 4 Kb algorithm-dependent computing-in-memory SRAM unit-macro with 2.3 ns and 55.8TOPS/W fully parallel product-sum operation for binary DNN edge processors," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018.

[23] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A mixed–signal binarized convolutional-neural-network accelerator integrating dense weight storage and multiplication for reduced data movement," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 141–142.

[24] Z. Jiang, S. Yin, M. Seok, and J. Seo, "XNOR-SRAM: In-memory mixed-signal accelerator for binary/ternary-input and binary-weight deep neural networks," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 173–174.

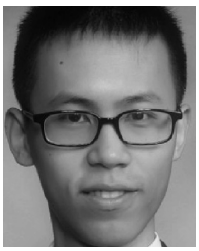
[25] X. Si *et al.*, "A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 396–398.
[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[27] B. Heisele, T. Poggio, and M. Pontil, "Face detection in still gray images," Center Biol. Comput. Learn., MIT, Cambridge, MA, USA, Tech. Rep. AIM-1687, 2000.
[28] L. Chang *et al.*, "An 8T-SRAM for variability tolerance and low–voltage operation in high–performance caches," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 956–963, Apr. 2008.
[29] J. Wang *et al.*, "A compute SRAM with bit-serial integer/floating-point operations for programmable in-memory vector acceleration," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 224–226.
[30] J.-S. Seo *et al.*, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.
[31] N. Verma *et al.*, "In-memory computing: Advances and prospects," *IEEE Solid State Circuits Mag.*, vol. 11, no. 3, pp. 43–55, Aug. 2019.
[32] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inf. Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960.
[33] I. Hubara. *BinaryNet.pytorch*. Accessed: Jan. 7, 2020. [Online]. Available: https://github.com/itayhubara/BinaryNet.pytorch
[34] H. Jia, Y. Tang, H. Valavi, J. Zhang, and N. Verma, "A microprocessor implemented in 65 nm CMOS with configurable and bit-scalable accelerator for programmable in-memory computing," 2018, *arXiv:1811.04047*. [Online]. Available: https://arxiv.org/abs/1811.04047?
[35] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Jul. 2018.
[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.
[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 4171–4186.
[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.



Shihui Yin (Student Member, IEEE) received the B.S. degree in microelectronics from Peking University, Beijing, China, in 2013, and the M.S. degree in electrical engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2015. He is currently pursuing the Ph.D. degree in electrical engineering with Arizona State University, Tempe, AZ, USA.

His research interests include low-power biomedical circuit and system design, and energy-efficient hardware design for machine learning and neuromorphic computing.

Mr. Yin was a recipient of the University Graduate Fellowship from Arizona State University in 2015 and the IEEE Phoenix Section Student Scholarship for the year 2016.



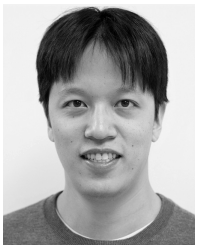
Zhewei Jiang (Student Member, IEEE) received the dual B.S. degree in physics from Adelphi University, Garden City, NY, USA, and in electrical engineering from Columbia University, New York, NY, USA, in 2013 and the M.S. degree in electrical engineering from Columbia University in 2015, where he is currently pursuing the Ph.D. degree in electrical engineering.

He has been a Research Assistant with the VLSI Laboratory, Columbia University, since 2015. His research interests include neuromorphic computing architecture, neural signal processing, in-memory computation for machine learning, and other algorithm implementations.

Jae-Sun Seo (Senior Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2006 and 2010, respectively.

From 2010 to 2013, he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, where he worked on cognitive computing chips under the DARPA SyNAPSE Project and energy-efficient integrated circuits for high-performance processors. In 2014, he joined the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA, as an Assistant Professor. In 2015, he was a Visiting Faculty with the Intel Circuits Research Lab, Hillsboro, OR, USA. His current research interests include efficient hardware design of machine learning and neuromorphic algorithms and integrated power management.

Dr. Seo was a recipient of the Samsung Scholarship from 2004 to 2009, the IBM Outstanding Technical Achievement Award in 2012, and the NSF CAREER Award in 2017.



Mingoo Seok (Senior Member, IEEE) received the B.S. degree *(summa cum laude)* in electrical engineering from Seoul National University, Seoul, South Korea, in 2005, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2007 and 2011, respectively, all in electrical engineering.

He was a member of the Technical Staff with Texas Instruments Inc., Dallas, TX, USA, in 2011. Since 2012, he has been with Columbia University, New York, NY, USA, where he is currently an Associate Professor of electrical engineering. His current research interests include ultra-low-power SoC design for emerging embedded systems, machine-learning VLSI architecture and circuits, variation, voltage, aging, thermal-adaptive circuits and architecture, on-chip integrated power circuits, and nonconventional hardware design.

Dr. Seok received the 1999 Distinguished Undergraduate Scholarship from the Korea Foundation for Advanced Studies, the 2005 Doctoral Fellowship from the Korea Foundation for Advanced Studies, and the 2008 Rackham Pre-Doctoral Fellowship from the University of Michigan. He also received the 2009 AMD/CICC Scholarship Award for picowatt voltage reference work and the 2009 DAC/ISSCC Design Contest for the 35-pW sensor platform design. He also received the 2015 NSF CAREER Award and the 2019 Qualcomm Faculty Award. He is also a Technical Program Committee Member for several conferences, including the IEEE International Solid-State Circuits Conference (ISSCC) and the IEEE Custom Integrated Circuits Conference (CICC). He has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS from 2013 to 2015. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS since 2015 and the IEEE SOLID-STATE CIRCUITS LETTER since 2017. He also serves as a Guest Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS for the 2019 ISSCC Special Issue.