DIMCA: An Area-Efficient Digital In-Memory Computing Macro Featuring Approximate Arithmetic Hardware in 28 nm

Chuan-Tung Lin, *Graduate Student Member, IEEE*, Dewei Wang, Bo Zhang, *Graduate Student Member, IEEE*, Gregory K. Chen, *Member, IEEE*, Phil C. Knag, *Member, IEEE*, Ram Kumar Krishnamurthy, *Fellow, IEEE*, and Mingoo Seok, *Senior Member, IEEE*

***Abstract*— Recent SRAM-based in-memory computing (IMC) hardware demonstrates high energy efficiency and throughput for matrix–vector multiplication (MVM), the dominant kernel for deep neural networks (DNNs). Earlier IMC macros have employed analog-mixed-signal (AMS) arithmetic hardware. However, those so-called AIMCs suffer from process, voltage, and temperature (PVT) variations. Digital IMC (DIMC) macros, on the other hand, exhibit better robustness against PVT variations, but they tend to require more silicon area. This article proposes novel DIMC hardware featuring approximate arithmetic (DIMCA) to improve area efficiency without hurting compute density (CD). We also propose an approximation-aware training model and a customized number format to compensate for the accuracy degradation caused by the approximation hardware. We prototyped the test chip in 28-nm CMOS. It contains two versions: the DIMCA with single-approximate hardware (DIMCA1) and DIMCA with double-approximate hardware (DIMCA2). The measurement results show that DIMCA1 supports a 4 b-activation and 1 b-weight (4 b/1 b) CNN model, achieving 327 kb/mm^2, 458–990 TOPS/W (normalized to 1 b/1 b), 8.27–392 TOPS/mm^2 (normalized to 1 b/1 b), and 90.41% accuracy for CIFAR-10. DIMCA2 supports a 1 b/1 b CNN model, achieving 485 kb/mm^2, 932–2219 TOPS/W, 14.4–607 TOPS/mm^2, and 86.96% accuracy for CIFAR-10.**

***Index Terms*— Approximate computing, approximation-aware training, deep learning, in-memory computing (IMC), neural network accelerators.**

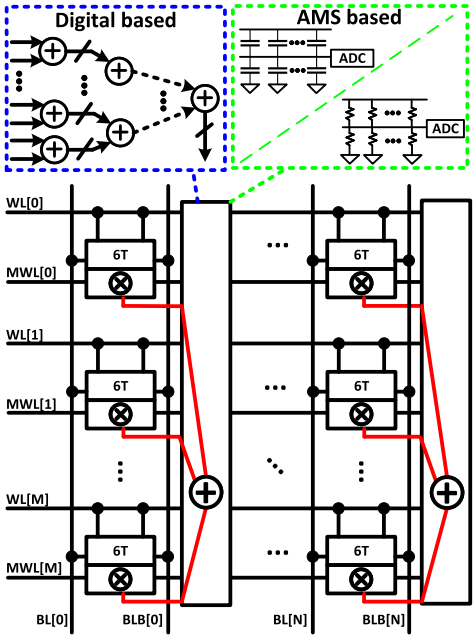


Fig. 1. Typical SRAM-based IMC circuits employing digital or AMS arithmetic hardware.

I. Introduction

DEEP neural network (DNN)-based inference has gained a large amount of research and development attention as it achieves unprecedented accuracy in a range of cognitive tasks such as image classification, object detection, speech recognition, and language processing [1], [2], [3], [4], [5].

However, DNN-based inference incurs much computational complexity. To mitigate the complexity, recent works have proposed specialized hardware architectures [6], [7], [8], [9], [10]. Especially, the matrix–vector multiplication (MVM), the dominant computational kernel of DNN-based inference workloads, involves many on-chip SRAM accesses because we can access SRAM in a row-by-row fashion. This poses a bottleneck to further improve performance and energy efficiency, even in the specialized processors.

To overcome the memory-access bottleneck, the concept and the circuit implementation for SRAM-based in-memory computing (IMC) hardware have been proposed. The IMC hardware aims to merge computing elements and memory elements at the array and bitcell levels. Some recent test chips demonstrate accessing all rows and performing multiply-and-accumulate (MAC) in one cycle, avoiding the slow row-by-row data access and demonstrating orders of magnitude improvement in energy efficiency and computing throughput [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

Manuscript received 17 January 2023; revised 25 April 2023 and 15 August 2023; accepted 1 September 2023. Date of publication 28 September 2023; date of current version 27 February 2024. This article was approved by Associate Editor Meng-Fan Chang. This work was supported in part by the Semiconductor Research Corporation (SRC) under Grant Task 2810.034 and in part by NSF under Grant 1919147. *(Corresponding author: Mingoo Seok.)*

Chuan-Tung Lin, Dewei Wang, Bo Zhang, and Mingoo Seok are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: ms4415@columbia.edu).

Gregory K. Chen, Phil C. Knag, and Ram Kumar Krishnamurthy are with the Circuit Research Laboratory, Intel Corporation, Hillsboro, OR 97229 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2023.3313519.

Digital Object Identifier 10.1109/JSSC.2023.3313519

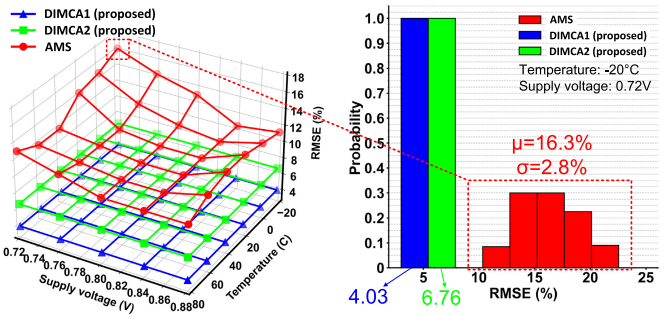


Fig. 2. PVT variations negatively affect AMS computing hardware's accuracy. The results are based on the simulation of a capacitor-based IMC SRAM macro. Jiang et al. [12] computed a 256-D binary dot product.

Fig. 1 shows the typical architecture of an SRAM-based IMC macro. Most earlier macros employ analog-mixed-signal (AMS) circuits to reduce area overhead and power consumption [12], [18], [20], [21], [22]. However, these so-called AIMC macros exhibit significant variability across process, voltage, and temperature (PVT) variations, degrading computation accuracy. As shown in Fig. 2, we also confirm the large root-mean-square error (RMSE) of 22.5% at the worst-case PVT corner via simulations.

On the other hand, digitally implemented IMC circuits can achieve better robustness over PVT variations [23], [24], [25], [26], [27], [28], [29], [30], [31]. This so-called digital IMC (DIMC) can also achieve better technology and voltage scalability. However, due to the bulky digital arithmetic gates, DIMC consumes more area than its AMS counterparts. Note that a CMOS full adder (FA) cell requires 28 transistors. The large area overhead degrades the weight density (kb/mm^2). To reduce the area overhead, recent works propose to time-share arithmetic gates among multiple bitcells [29]. While the time-sharing approach can reduce the area overhead, it inevitably reduces the compute density (CD) (TOPS/mm^2).

To improve area efficiency while maintaining CD, we propose employing digital *approximate* arithmetic circuits in this work. The approximate circuits incur the error in the computation result, but this error is deterministic and some of them can be compensated in the DNN training process. Based on this idea, we prototype the DIMC hardware featuring approximate arithmetic (DIMCA) in a 28-nm CMOS. Specifically, we create two versions: DIMCA with single-approximate hardware (DIMCA1) that employs single-approximate compressors and supports 1–4 b activations and 1 b weights and DIMCA with double-approximate hardware (DIMCA2) that employs double-approximate compressors and supports 1 b activations and 1 b weights. We also develop and adopt a customized number representation, called multibit XNOR (MB-XNOR), for both versions.

The measurement results show that DIMCA1 achieves 327 kb/mm^2, 458–990 TOPS/W (normalized to 1 b/1 b activation/weight), 8.27–392 TOPS/mm^2 (normalized to 1 b/1 b), and 90.41% accuracy for CIFAR-10 with a 4 b-activation and 1 b-weight (4 b/1 b) CNN model. DIMCA2 achieves 485 kb/mm^2, 932–2219 TOPS/W, 14.4–607 TOPS/mm^2, and 86.96% accuracy for CIFAR-10 with a 1 b/1 b CNN model.

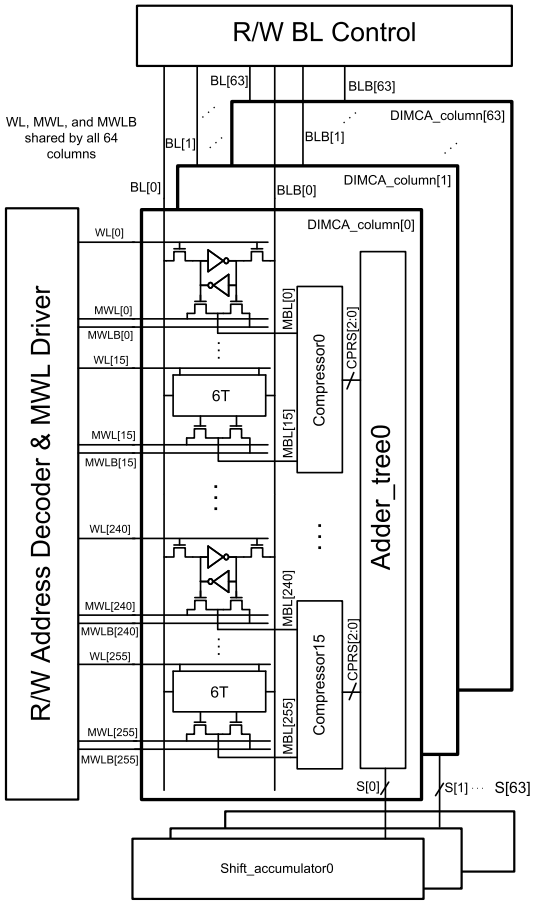


Fig. 3. Architecture of the proposed DIMCA macro.

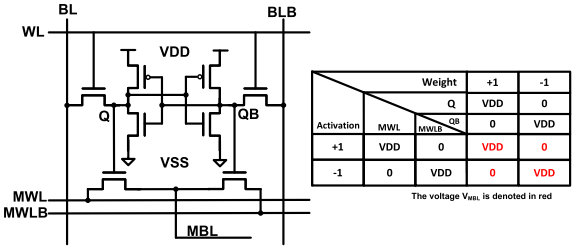


Fig. 4. DIMCA bitcell design and the XNOR operand table.

The remainder of this article is as follows. In Section II, we will present the proposed DIMCA architecture and circuits, the associated training model, and the custom number representation. In Section III, we will show the chip prototype and measurement results. Section IV presents the analytical models of DIMC hardware and the design space exploration using the models. Finally, Section V concludes this article.

II. DIMC With Approximate Arithmetic

A. Macro Architecture

Fig. 3 illustrates the architecture of DIMCA consisting of 256 × 64 bitcells. Each column contains 256 bitcells, 16 double-approximate compressors, one 16-input adder tree, and one 11-b shift accumulator. DIMCA1 uses single-approximate compressors, while DIMCA2 double-approximate compressors. The single-approximate compressor

exhibits a smaller error, while the double-approximate compressor requires a smaller number of transistors. We will compare them in detail in Section II-B.

Fig. 4 shows the bitcell consists of: 1) the standard 6T SRAM bitcell to store a binary weight and 2) two other pass transistors to perform XNOR (multiplication). It can support an input activation of up to 4 bits. We feed the input activation in a bit serial fashion through a pair of MAC wordline (MWL) and MWLB. The multiplication result on the MAC bitline (MBL) goes to the compressor.

A compressor counts the number of 1's in the Input. The compressor's input is in the unsigned integer format. For example, the standard 15-4 compressor generates a 2^0–2^3 unsigned integer from 15 2^0 weighted input bits.

The adder tree is responsible for adding 16 unsigned integers that 16 compressors produce. The result is sent to the shift accumulator, which accumulates the adder tree outputs, while the multibit input activations are fed in a bit serial fashion. Therefore, it takes a total of four clock cycles for DIMCA to multiply a 4 b 256-D input activation vector with a 1 b 256 × 64-D weight matrix.

Note that from the shift-and-accumulator output, we subtract the number of −1s, which is equal to 256-(the number of +1s), and produce the final output in the 2's complement format.

B. Compressor and Adder Tree Optimization

If implemented based on the exact (nonapproximate) arithmetic, each column of DIMC employing binary multipliers, compressors, and an adder tree, requires a total of 247 FA cells, marking the device efficiency of 19.5 transistors per bit (T/b). Therefore, we aim to improve the device efficiency by leveraging approximate compressors [32], [33], [34] and other circuit techniques.

As shown in Fig. 5, we have designed three compressor circuits based on [32]. They support three different levels of approximation: exact, single-approximate, and double-approximate. The approximate compressors replace FAs with much smaller AND and OR gates. Interleaved AND and OR gates remove the bias of the error. For example, an AND gate can potentially cause a "−1" error, while an OR can cause a "+1" error. Therefore, interleaving AND and OR gates helps minimize the mean value of errors.

In Fig. 5, the double-approximate compressor results in 55% fewer transistors than an exact compressor, and the single-approximate compressor consumes 40% fewer transistors. However, the simulation shows that they exhibit nonzero RMSE values ranging from 4.03% to 6.76%. Yet, since DIMCA uses digital circuits, the error that it produces is deterministic.

To further reduce the area, we propose a pass-gate-based FA (see Fig. 6) and design a ripple-carry-adder (RCA) using those FAs [see Fig. 7(a)]. However, the pass-gate logic incurs V_t drop, and if accumulated across more than three pass gates, it can largely hurt the noise margin of the circuits. Hence, we have identified all the nodes in an FA that do not have full-swing signals [marked in red in Fig. 6(a)] and have inserted inverters such that the number of series-connected pass-gates is less than two.

However, naïve inverter insertion often requires inserting *two* inverters to remove data inversion, increasing area overhead. As shown in Fig. 8, it could increase the silicon area by 73%. Therefore, we designed a custom 12T FA cell [see Fig. 6(a)] employing both pass- and transmission-gates. The layout [see Fig. 6(c)] takes the area of 1.764 μm^2. Also, we designed the second version of the 12T FA cell which has the inverted inputs [see Fig. 6(b)]. These techniques reduce the number of inserted inverters, enabling 17% silicon area savings (see Fig. 8). Note that the pass-gate logic can increase the static power because the V_t drop cannot fully turn off the PMOS of the inserted inverter. The simulation of the adder tree shows the static power consumption of 28.9 μW and the dynamic power consumption of 111.9 μW at 0.9 V, confirming that the static power still takes a smaller portion than the dynamic power.

Fig. 7(b) shows the block diagram of one DIMCA column, which employs the approximate compressor (CPRS) and RCAs [see Fig. 7(a)]. As shown in Fig. 9, the FA cell level optimization reduces the transistor count by 39%. The approximate compressors led to the total transistor count reduction of 46.4%–56.4%. The DIMCA2 column exhibits the device efficiency of 13.67 T/b, a 30% reduction compared to the exact arithmetic counterpart.

C. Approximate-Aware Training Model

The approximate compressors produce positive and negative errors. As a result, DIMCA1 (DIMCA2) exhibits the worst-case RMSE error of 4.03% (6.76%). Although these worst-case RMSEs are smaller than that of AIMC, they degrade inference accuracy. We have DIMCA to perform a VGG-like 1 b/1 b activation/weight CNN model for CIFAR-10.[1] Trained by the conventional training model, DIMCA1 (DIMCA2) achieves a poor accuracy of 50.9% (25.2%). As a comparison, the exact arithmetic hardware achieves 89.6%.

To improve the inference accuracy, we propose an approximation-aware training algorithm based on [35]. We employ approximate arithmetic such as bitwise AND, OR, and FA to emulate the behavior of single-approximate and double-approximate compressors in the forward path computation. The backpropagation does not use approximate arithmetic. As shown in Fig. 10, DIMCA1 (DIMCA2) now can achieve a much higher accuracy of 89.0% (86.9%).

D. Custom Number Representation

In addition to the 1 b/1 b CNN model, we have DIMCA to perform a 4 b/1 b CNN model for CIFAR-10. However, as shown in Fig. 11, the 4 b/1 b CNN model achieves lower accuracy than the 1 b/1 b counterpart. This is because the multibit activation model generally requires more precise arithmetic. Also, conventionally, multibit activations are in the 2's complement format, which also requires weights to be in the same 2's complement format for simple arithmetic

[1]The CNN model has the following topology: 128C3-128C3-P2-256C3-256C3-P2-512C3-512C3-P2-FC1024-FC1024-FC10, where 128C3 represents 128 features 3 × 3 convolution, P2 represents 2 × 2 pooling, and FC1024 represents a 1024 fully-connected layer.

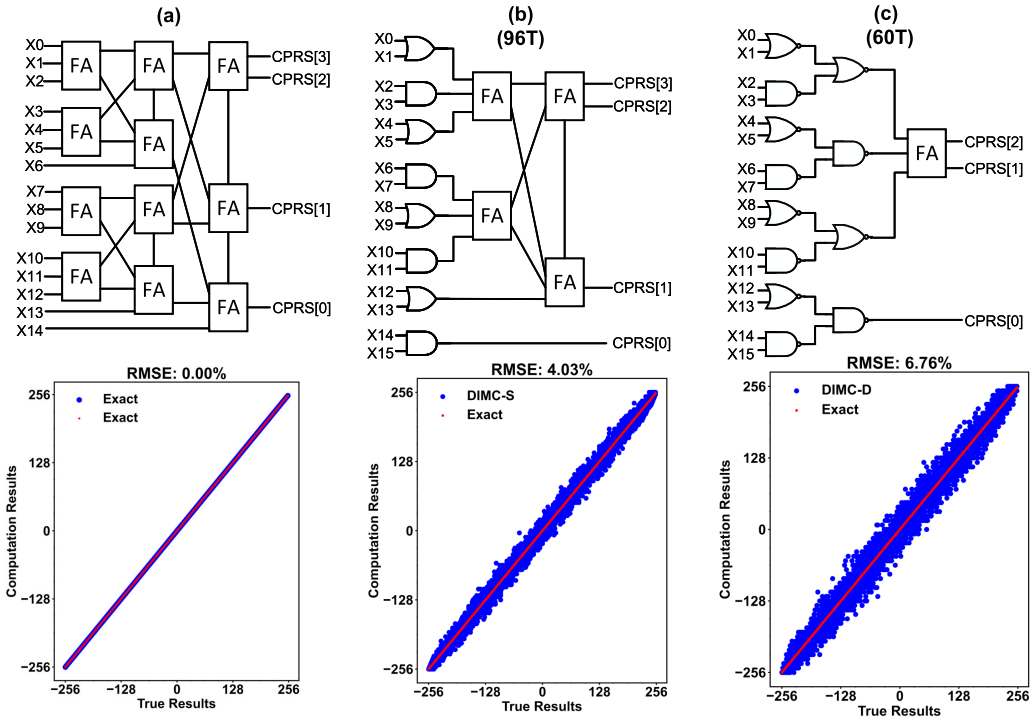


Fig. 5. (a) Exact, (b) single-approximate, and (c) double-approximate compressor circuits and the corresponding RMSE errors in computing 256-D binary dot products.

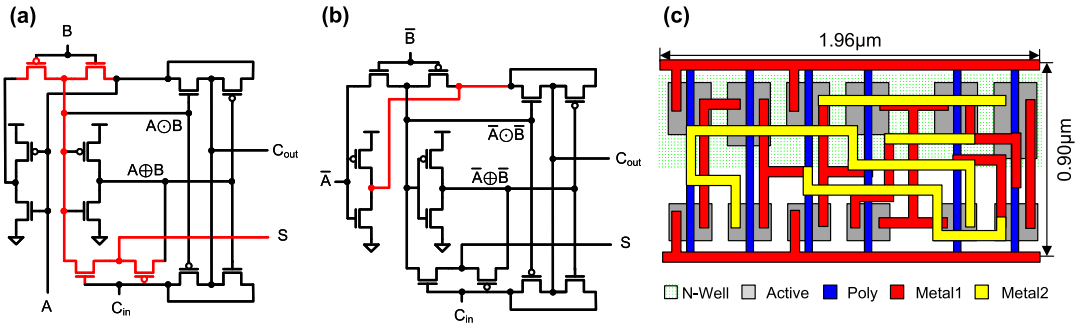


Fig. 6. Schematics of the 12T FA with (a) regular inputs and (b) complementary inputs. (c) Layout of the 12T FA circuits.

TABLE I
MB-XNOR TO 2'S COMPLEMENT MAPPING FOR 4 B ACTIVATIONS

MB-XNOR	2's complement	MB-XNOR	2's complement
1111	01111	0111	11111
1110	01101	0110	11101
1101	01011	0101	11011
1100	01001	0100	11001
1011	00111	0011	10111
1010	00101	0010	10101
1001	00011	0001	10011
1000	00001	0000	10001

hardware implementation. However, in the 2's complement format, a 1 b weight represents -1 or 0, while our 1 b weight represents -1 or $+1$. This discrepancy further degrades inference.

To better support a multibit activation CNN model, we propose a new custom number format titled MB-XNOR based on the bipolar encoding scheme [36]. We make each 1 b weight represent $+1$ or -1 (instead of -1 or 0) and use a similar format for an N-bit activation

$$b_{N-1}b_{N-2},\ldots,b_0 = \sum_i b_i \cdot 2^i \tag{1}$$

where b_i is $+1$ or -1. If the inputs of DIMCA are in the 2's complement, we need to convert them from the 2's complement to our MB-XNOR format, which can be done with a small lookup table. Table I shows one to support 4 b activations.

Our DIMCA hardware shows a higher signal-to-noise ratio (SNR) that achieves a higher inference accuracy with the proposed MB-XNOR format than the 2's complement format. We formulate the SNR as follows:

$$\text{SNR} = \sum y_{\text{true}}^2 \Big/ \sum (y_{\text{true}} - y_{\text{approx}})^2 \tag{2}$$

where y_{true} is the ground truth of the dot product between a 256-dimension 1–4 b Gaussian-distributed input vector and a 256-dimension binomial-distributed weight vector and y_{approx} is the same dot product results yet using DIMCA . Fig. 12(a)

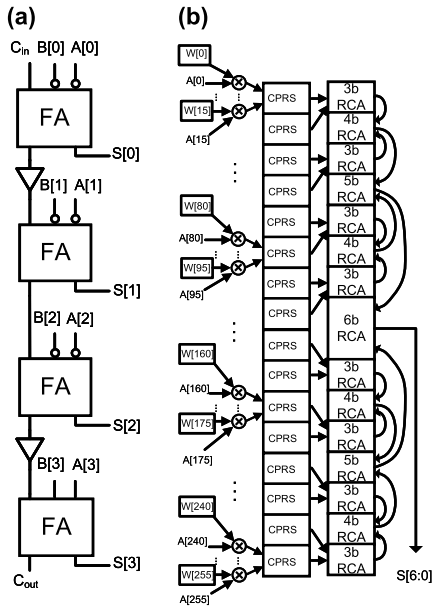


Fig. 7. (a) 4 b RCA circuits. (b) Block diagram of one DIMCA column.

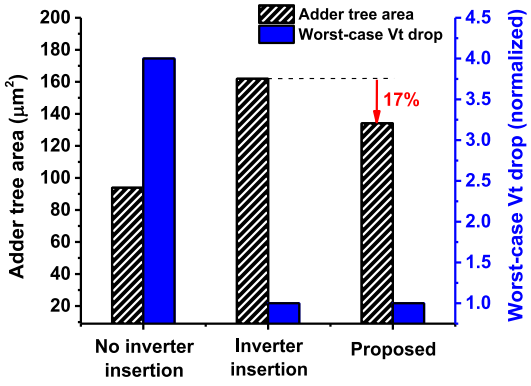


Fig. 8. Proposed RCA minimizes V_t drop at 17% less area than the RCA with a naïve inverter insertion.

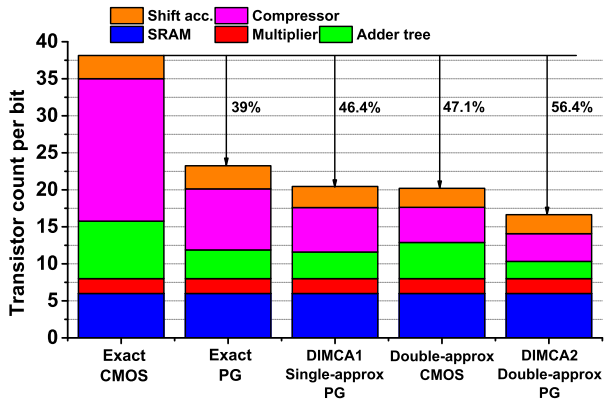


Fig. 9. Approximate arithmetic hardware reduces the transistor count per bit by 46.4%–56.4%.

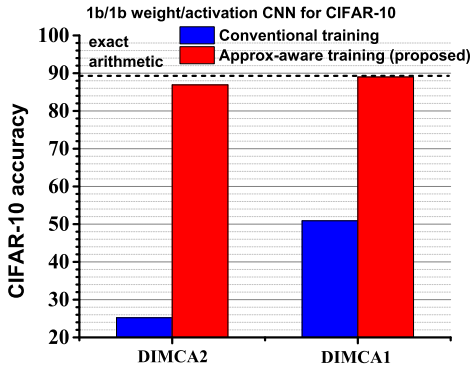


Fig. 10. CIFAR-10 accuracy: the conventional versus the proposed approximation-aware training model.

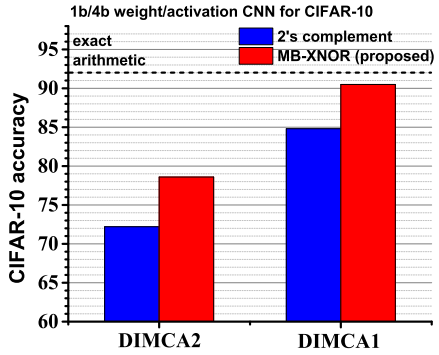


Fig. 11. MB-XNOR format offers better CIFAR-10 accuracy than the 2's complement format for the DIMCA.

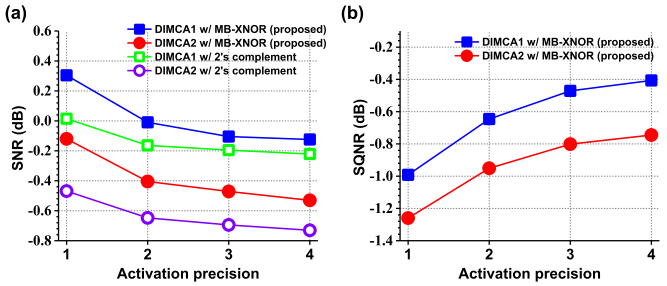


Fig. 12. (a) MB-XNOR format provides a greater SNR than the 2's complement format for the DIMCA. (b) SQNR of the proposed MB-XNOR format across 1–4 b activation.

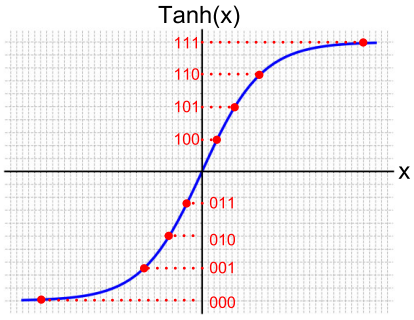


Fig. 13. Tanh activation quantized to 3 b in the MB-XNOR format.

shows the SNR simulation results. The DIMCA2 employing the proposed MB-XNOR format achieves a 0.15 higher SNR than 2's complement. On the other hand, as shown in Fig. 12(b), we also analyze the signal-to-quantization noise ratio (SQNR). The SQNR increases with higher activation precision because of smaller quantization noise. The lower SNR than [15] because we include the approximate circuits, such as

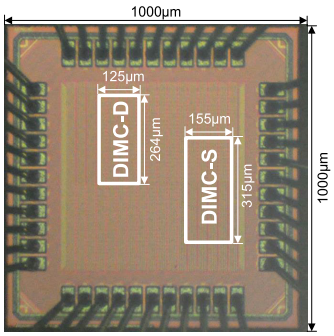


Fig. 14. Die micrograph.

single-approximate compressors and double-approximate compressors, in our simulation, whereas [15] does not employ any approximation scheme. Although the approximation degrades the SNR, the proposed DIMCA causes a deterministic error only, which the approximation-aware training can restore significantly.

We also evaluate the CNN inference accuracy. DIMCA1 using the MB-XNOR format achieves a 5.4% higher inference accuracy (see Fig. 11). DIMCA2 using the MB-XNOR format also improves the accuracy by 6% compared to the 2's complement, but the accuracy is low (78.6%) for the 1 b/4 b weight/activation model. Therefore, we consider DIMCA2 to use only 1 b/1 b weight/activation models.

One caveat is that this number format disallows the use of some activation functions, such as ReLU, because it cannot represent 0. We can still use other popular activation functions such as hyperbolic tangent (tanh) (see Fig. 13) and leaky ReLU. It would be possible to support high-precision (8 b or 16 b) arithmetic only with a better approximation scheme. Currently, the proposed approximate scheme relies on bit serial input, and the error increases for the MSB-related computation.

III. Chip Prototype and Measurement

We prototyped the DIMCA test chip in a 28-nm CMOS technology. Fig. 14 shows the chip photograph. The 16-kb DIMCA1 consumes 0.049 mm^2 and the 16-kb DIMCA2 takes 0.033 mm^2.

We measure the two DIMCA macros across 0.5–1.1 V at 25 °C. As shown in Fig. 15(a), DIMCA2 achieves 932–2219 TOPS/W and 475–20 032 GOPS; DIMCA1 attains 458–990 TOPS/W and 405–192 15 GOPS. The energy efficiency and throughput metrics are all normalized to 1 b/1 b. As shown in Fig. 15(b), we also measured the energy efficiency and throughput across five chips at the nominal voltage of 0.9 V. Both energy efficiency and throughput measurements show only a small distribution. Fig. 15(c) shows the energy efficiency measurements across VDDs at 25% and 50% input toggle rates (TRs). In the SRAM mode, both macros take 340 ns (256 cycles at 752 MHz) to update all 16-kb weights at 0.9 V and 25 °C.

We also measure the energy efficiency and throughput across different temperatures. As shown in Fig. 16(a), the energy efficiency degrades with higher temperatures. The reason is that the leakage power dissipation increases steeper than the throughput does. At 0.9 V from −15 °C to 25 °C, the energy efficiency still increases because the leakage only accounts for 10%–25% of the total power consumption. At 25 °C and above, the leakage starts to consume a more significant portion of the total power consumption (32%–58%). We also measure the leakage power across the supply voltages and temperatures. As shown in Fig. 16(b), the leakage power increases exponentially with temperature.

Fig. 16(c) shows the energy consumption breakdown. For DIMCA1, the adder trees account for 43.03% of the total energy consumption, and the compressors take 38.05%. The shift accumulators and MWL drivers consume 9.11% and 8.62%, respectively. The bitcell array accounts for only 1.2%. Fig. 16(d) shows the area breakdown.

Table II compares the proposed hardware to the recent works. Compared with [25], DIMCA2 attains 53% higher weight density, 2.4× higher CD, and 10% higher energy efficiency. On the other hand, Yan et al. [29] adopt a time-sharing architecture and trade the throughput and CD for the weight density. As a result, Yan et al. [29] achieve a high weight density of 1067 kb/mm^2 but achieve a lower CD of 178 TOPS/mm^2, which is 49% lower than DIMCA2. The proposed DIMCA macros achieve high CD and weight density at the same time while maintaining state-of-the-art energy efficiency and CNN inference accuracy.

IV. Modeling and Design Space Exploration

This section aims to develop the parameterized analytical models of SRAM-based DIMC macros for quick design space exploration. Fig. 17 shows the high-level block diagram of the DIMC array, which we assume in the modeling process. The array contains macros and additional adders to sum the outputs of the macros. The model development considers the following design parameters: array dimensions, activation and weight precisions, approximation schemes, and the degree of arithmetic hardware multiplexing (aka reuse and time-sharing). Then, we calibrate the developed models based on the two recent DIMC test chips' measurements (see [25] and [37]), which we call the base DIMC hardware hereafter. Then, we will use these models to estimate/predict the energy efficiency, throughput, area, and RMSE of various SRAM-based DIMC macros.

A. Silicon Area

In this section, we develop the area model of the DIMC array. First of all, we formulate the size (memory capacity) of a DIMC array in kb as follows:

$$S_{\text{DIMC}} = \frac{N_{\text{row}} \cdot N_{\text{col}}}{1024} \tag{3}$$

where N_{row} is the total number of rows of the DIMC array and N_{col} is the total number of columns of the DIMC array. Refer to Fig. 17 for the definitions.

On the other hand, the total silicon area of the IMC array is roughly proportional to S_{array}. However, it is impractical

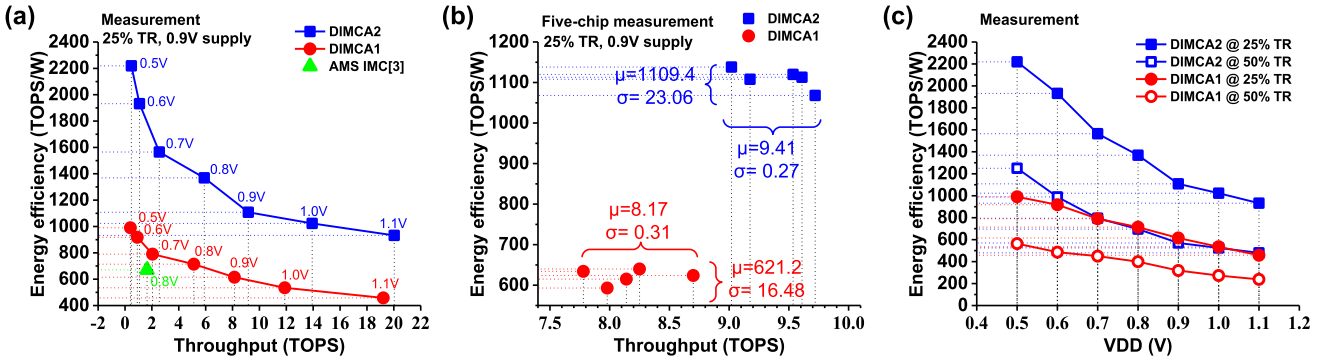


Fig. 15. (a) Energy efficiency and throughput across different supply voltages. (b) Multichip measurements of energy efficiency and throughput at 0.9 V supply. (c) Energy efficiency at 25% and 50% TRs.

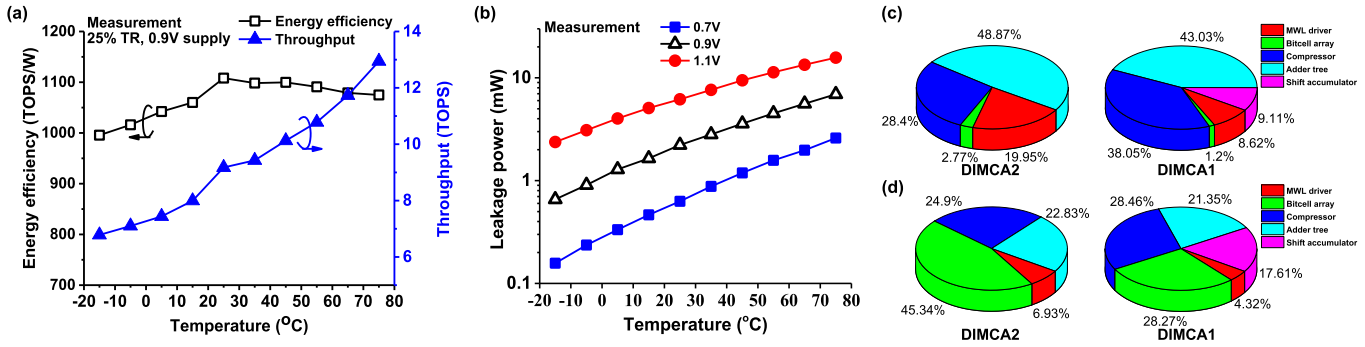


Fig. 16. (a) Energy efficiency and throughput measurements of DIMCA2 across temperatures at 0.9 V. (b) Leakage power consumption of DIMCA2 across supply voltages and temperatures. (c) Energy breakdown of the two proposed DIMCA macros. (d) Area breakdown of the two proposed DIMCA macros.

TABLE II

PERFORMANCE COMPARISON WITH STATE-OF-THE-ART AIMC AND DIMC MACROS

	This work		Jia, ISSCC'21 [15]	Jiang, JSSC'20 [12]	Yan, ISSCC'22 [29]	Tu, ISSCC'22 [28]	Fujiwara, ISSCC'22 [26]	Chih, ISSCC'21 [25]	Kim, JSSC'21 [24]	He, ISSCC'23 [30]	Yue, ISSCC'23 [31]
	DIMCA2	DIMCA1									
Technology[nm]	28	28	16	65	28	28	5	22	65	28	28
MAC operation	Digital	Digital	AMS	AMS	Digital	Digital	Digital	Digital	Digital	Digital	Analog + digital
Array size	16Kb	16Kb	4.5Mb	16Kb	32Kb	96Kb	64Kb	64Kb	16Kb	16Kb	36Kb
Macro area [mm²]	0.033	0.049	11	0.081	0.030	0.941	0.0133	0.202	0.227	0.028	0.0524
Weight density [Kb/mm²]	485	327	419	198	1067	102	4812	317	70.5	571	687
Supply voltage [V]	0.45-1.10	0.45-1.10	0.8	0.8	0.8	0.6-1	0.5-0.9	0.72	0.6-0.8	0.54-0.9	0.6-0.9
Activation precision [bit]	1	1-4	1-8	1	1-8	8/16	4	1-8	1-16	2-8	1-8
Weight precision [bit]	1	1	1-8	1	1/4/8	8/16	4	4/8/12/16	4/8/12/16	2-8	1-8
Operating frequency [MHz]	280	250	20[1]	50	333	50-220	360-1140	500	138	20-230	50-286
Input toggle rate	25%	25%	NA	NA	NA	NA	NA	18%	NA	18%	NA
Energy efficiency [TOPS/W]	1,108 @ 0.9V 2,219 @ 0.5V	154 @ 0.9V (4b1b) 248 @ 0.5V (4b1b)	121 @ 0.8V (4b4b)	671 @ 0.8V	27.38 @ 0.8V (8b8b)	57.8 @ 0.65V (8b8b)	254 @ 0.5V (4b4b)	89 @ 0.72V (4b4b)	117 @ 0.6V (1b1b)	102 @ 0.54V (8b8b)	1158 @ 0.6V (1b1b)
Throughput [GOPS][2]	9,175 @ 0.9V 20,032 @ 1.1V	2,035 @ 0.9V (4b1b) 4,804 @ 1.1V (4b1b)	41 @ 0.8V (4b4b)	1,638 @ 0.8V	2.67 @ 0.8V (8b8b)	225 @ 1V (8b8b)	737.5 @ 0.9V (4b4b)	825 @ 0.72 (4b4b)	567 @ 0.8V (1b1b)	118 @ 0.9V (8b8b)	560 @ 0.9V (1b1b)
Compute density [TOPS/mm²]	607 (1b1b)	98 (4b1b)	2.67 (4b4b)	20.22 (1b1b)	0.178 (8b8b)	1.43 (8b8b)	221 (4b4b)	16 (4b4b)	2.5 (1b1b)	4.2 (8b8b)	27.36 (1b1b)
CIFAR-10 accuracy	86.96%	90.41%	91.51%	85.50%	NA	NA	NA	NA	NA	NA	NA

[1] Computed from throughput and array size; [2] Normalized array size to 16kb.

to create one macro with very large N_{row} and N_{col} since the long bitlines and wordlines slow down the read and write operations. Therefore, we assume that each macro has 256 rows and 64 columns while having multiple macros to create a large DIMC array. We also assume to have additional adders to sum up the results from the macros in an array. As a result, we can formulate the area of a DIMC array A_{DIMC} as follows:

$$A_{\text{DIMC}} = A_{\text{macros}} + A_{\text{adders}} \tag{4}$$

where A_{macros} is the area of all IMC macros and A_{adder} is the area of the additional adders.

We formulate A_{macros} while considering the degree of multiplexing (aka hardware reuse or time-sharing) arithmetic hardware. The fully digital design of DIMC hardware makes it straightforward for columns to reuse the arithmetic hardware. If we time-share the arithmetic hardware across $D_{\text{multiplex}}$ columns, only one column of weights, out of every $D_{\text{multiplex}}$ columns, performs MVMs with the MWLs/MWLBs. The higher degree of time-sharing can save area at a throughput

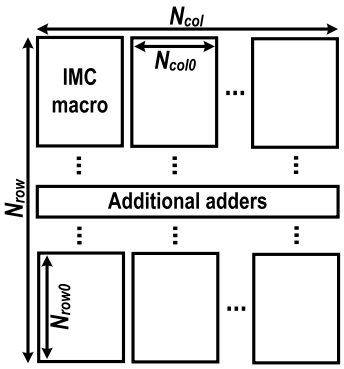


Fig. 17. High-level block diagram of the DIMC array we assume in the model development. The array contains macros and additional adders to sum the outputs of the macros.

penalty. Based on this idea, we formulate A_{macros} as follows:

$$A_{\text{macros}} = A_0 \cdot \frac{N_{\text{row}}}{N_{\text{row0}}} \cdot \frac{N_{\text{col}}}{N_{\text{col0}}} \cdot \left\{1 - r_{\text{arith}} + \frac{r_{\text{arith}}}{D_{\text{multiplex}}}\right\} \quad (5)$$

where A_0 is the area of the base DIMC macro and N_{row0} and N_{col0} represent N_{row} and N_{col} of the base macro. r_{arith} is the ratio of the area of the arithmetic hardware, that is, compressors, adder trees, and shift accumulators to the total macro area. $D_{\text{multiplex}}$ is the number of columns that share arithmetic hardware.

On the other hand, approximate-arithmetic compressors, like those used in DIMCA, can effectively reduce the area of the base DIMC macro (A_0). Our DIMCA employs approximate compressors. To develop a more detailed model for A_0, we formulate the area of the exact compressors as follows:

$$A_{\text{exact}} = \{(N+1) - \log_2(N+1) - 1\} \cdot A_{\text{FA}} \cdot N_{\text{exact}} \quad (6)$$

where N is the number of inputs, A_{FA} is the area of one FA cell, and N_{exact} is the number of exact compressors in a DIMC macro.

Then, we formulate the area of the single-approximate compressors as follows:

$$A_{\text{approx1}} = \left\{\left\{\frac{N}{2} - \log_2 N\right\} \cdot A_{\text{FA}} + \frac{N}{2} \cdot A_{\text{AND}}\right\} \cdot N_{\text{single-approx}} \quad (7)$$

where A_{AND} is the area of an AND gate and N_{approx1} is the number of single-approximate compressors in one macro. Similarly, the area of the double-approximate compressors can be formulated as follows:

$$A_{\text{approx2}} = \left\{A_{\text{FA}} + \left(\frac{N}{2} + \frac{N}{4}\right) \cdot A_{\text{AND}}\right\} \cdot N_{\text{approx2}} \quad (8)$$

where N_{approx2} is the number of double-approximate compressors in one macro.

Finally, we create a model for A_{adders} by assuming we need to accumulate only across the rows of DIMC. Under this assumption, a DIMC array having N_{row} and N_{col} needs to accumulate $N_{\text{row}}/N_{\text{row0}}$ partial sums for all N_{col} columns. Therefore, we can formulate A_{adders} as follows:

$$A_{\text{adders}} = A_{\text{FA}} \cdot N_{\text{adders}}$$

$$N_{\text{adders}} = \frac{N_{\text{col}}}{N_{\text{col0}}} \cdot \Sigma_{i=0}^{\log_2 \frac{N_{\text{row}}}{N_{\text{row0}}} - 1} \times \left\{2^i \cdot \left(\text{BW}_{\text{out}} + \log_2 \frac{N_{\text{row}}}{N_{\text{row0}}} - 1 - i\right)\right\} \quad (9)$$

where A_{FA} is the area of the one FA cell, N_{adders} is the total number of additional FA cells, and BW_{out} is the bit width of the output of each macro.

B. Throughput and CD

In this section, we create the models for computing throughput in TOPS and CD in TOPS/mm^2. First of all, we formulate compute throughput (CT) as follows:

$$\text{CT} = \text{CT}_0 \cdot \frac{1}{D_{\text{multiplex}}} \cdot \frac{N_{\text{row}}}{N_{\text{row0}}} \cdot \frac{N_{\text{col}}}{N_{\text{col0}}} \cdot \frac{\text{BW}_{\text{in0}}}{\text{BW}_{\text{in}}} \cdot \frac{\text{BW}_{\text{wt0}}}{\text{BW}_{\text{wt}}} \quad (10)$$

where CT_0 is the throughput of the base DIMC, BW_{in0} is the bit width of input activation (BW_{in}) of the base DIMC, and BW_{wt0} is the bit width of weights (BW_{wt}) of the base DIMC.

Based on (4) and (10), we can formulate CD as follows:

$$\text{CD} = \frac{\text{CT}}{A_{\text{DIMC}}}. \quad (11)$$

The second row of (10) shows that CD increases proportionally with S_{DIMC}. The third row of (10) normalizes CD to the activation and weight bit widths (precision).

C. Energy Efficiency

In this section, we create the energy efficiency model in TOPS/W. We can formulate the energy efficiency of DIMC hardware (EE) as follows:

$$\text{EE} = \text{EE}_0 \cdot \frac{\text{TN}}{\text{TN}_0} \cdot \frac{\text{BW}_{\text{in0}}}{\text{BW}_{\text{in}}} \cdot \frac{\text{BW}_{\text{wt0}}}{\text{BW}_{\text{wt}}} \quad (12)$$

where EE_0 is the energy efficiency of the base DIMC hardware and TN_0 is the technology node (TN) of the base DIMC hardware uses. The second and third rows of (12) are added to normalize EE to the TN and activation and weight bit widths of the base DIMC hardware. (12) does not contain N_{row} and N_{col} since the amount of arithmetic hardware in DIMC scale with N_{row} and N_{col}. In other words, EE remains almost unchanged across different N_{row}s and N_{col}s. Note that we ignore the energy consumption of the additional adders (those adders to sum the outputs of the macros) since it is later found to be insignificant compared to total power consumption.

D. Normalized Root Mean Squared Error (NRMSE)

In this section, we introduce the NRMSE model of DIMCA hardware. We denote the MAC result of the ith column to be $M^{(i)}$ and its error to be $e^{(i)}$. For simplicity, we denote the max value of $M^{(i)}$s to be M_{max} and the minimum value of $M^{(i)}$s to be M_{min}. Without the loss of generality, we assume that the error from each column follows the Gaussian distribution. We also assume that the error is independent of the errors from other columns, that is, $e^{(i)} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma_0^2)$. Based on those assumptions, we can create the NRMSE model of a single column as follows:

$$\text{NRMSE}_{\text{single-col}} = \frac{\sigma_0}{M_{\text{max}} - M_{\text{min}}}. \quad (13)$$

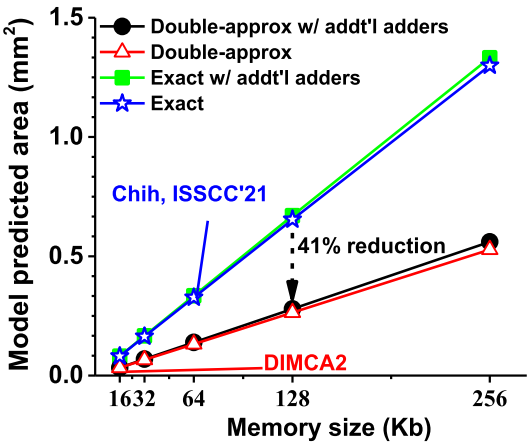


Fig. 18. Area for double-approximate and exact IMC across different IMC array sizes.

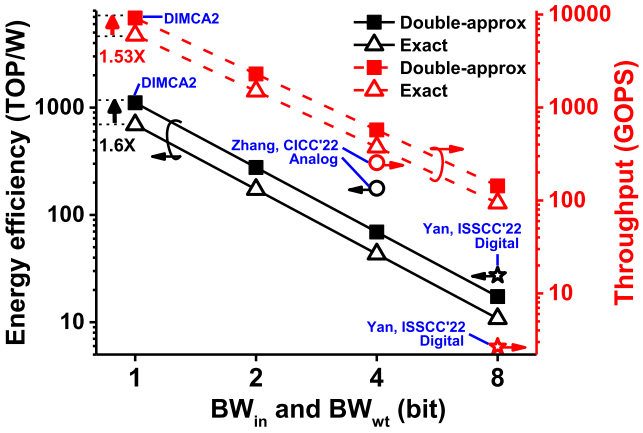


Fig. 19. Model-predicted TOPS/W and GOPS values across different activation and weight bit widths for DIMCA2 and exact DIMC. One operation refers to either 1-b addition or 1-b multiplication.

The binary-weighted sum across $\mathrm{BW_{wt}}$ columns is $M = \sum_{i=1}^{\mathrm{BW_{wt}}} 2^{i-1} M^{(i)}$. The variance of the binary-weighted sum error (e) is

$$\begin{aligned}\mathrm{Var}[e] &= \sum_{i=1}^{\mathrm{BW_{wt}}} \mathrm{Var}[2^{i-1} e^{(i)}] \\ &= \sum_{i=1}^{\mathrm{BW_{wt}}} 4^{i-1}\sigma_0^2 = \frac{4^{\mathrm{BW_{wt}}}-1}{3}\sigma_0^2. \end{aligned} \tag{14}$$

As a result, the binary-weighted sum error (e) across $\mathrm{BW_{wt}}$ columns is

$$e = \sum_{i=1}^{\mathrm{BW_{wt}}} 2^{i-1} e^{(i)} \sim \mathcal{N}\left(0, \frac{4^{\mathrm{BW_{wt}}}-1}{3}\sigma_0^2\right). \tag{15}$$

Since the full range of M is $(2^{\mathrm{BW_{wt}}}-1)(M_{\max}-M_{\min})$, we can formulate the NRMSE of M for $\mathrm{BW_{wt}}$ columns as follows:

$$\mathrm{NRMSE}_{\mathrm{BW_{wt}-col}} = \frac{1}{2^{\mathrm{BW_{wt}}}-1} \cdot \sqrt{\frac{4^{\mathrm{BW_{wt}}}-1}{3}} \cdot \frac{\sigma_0}{M_{\max}-M_{\min}}. \tag{16}$$

We can find from (16) that the NRMSE of the multi-column sum is the product of the single-column NRMSE ($\sigma_0/M_{\max}-M_{\min}$) and a factor which depends on $\mathrm{BW_{wt}}$.

Finally, since the activation and weights are interchangeable, we can treat the impact of multibit input activation on NRMSE as similar to the multibit weights. Therefore, for $\mathrm{BW_{in}}$-bit input activations and $\mathrm{BW_{wt}}$-bit weights, we can formulate the NRMSE as follows:

$$\begin{aligned}\mathrm{NRMSE} = &\frac{1}{2^{\mathrm{BW_{in}}}-1} \cdot \sqrt{\frac{4^{\mathrm{BW_{in}}}-1}{3}} \\ &\cdot \frac{1}{2^{\mathrm{BW_{wt}}}-1} \cdot \sqrt{\frac{4^{\mathrm{BW_{wt}}}-1}{3}} \cdot \frac{\sigma_0}{\mu_{\max}-\mu_{\min}}. \end{aligned} \tag{17}$$

E. Model Calibration and Model-Based Design Exploration

We calibrate the developed models against the measurement results. We use [25] for the exact arithmetic and [37] for approximate arithmetic as our base DIMC macros. Fig. 18 shows the measurement results lie on the model-predicted area curves. On the other hand, Fig. 19 shows the measurement results lie on the model-predicted energy-efficiency curves. It also shows the measurement results lie on the model-predicted throughput curves. Similarly, Fig. 20 shows that we calibrate the NRMSE model based on the measurements from [25] and this work.

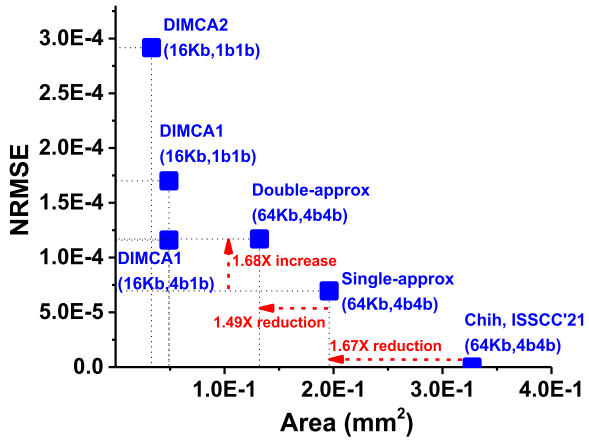


Fig. 20. NRMSE and area across different approximation schemes.

We use the models to predict the area, energy efficiency, throughput, and NRMSE across several design points. As shown in Fig. 18, the area of the DIMC is roughly proportional to the DIMC's total size. The additional adders exhibit a proportionally larger area with respect to the DIMC size but consume only <3% of the total area. On the other hand, for the same DIMC size, double-approximate compressors can reduce DIMC area by ~2.46×.

Fig. 19 shows the predicted energy efficiency and throughput. The double-approximate compressors improve energy efficiency by 1.6× and throughput by 1.53× compared to the exact counterparts. Also, the figure shows that both input and weight bitwidth ($\mathrm{BW_{in}}$ and $\mathrm{BW_{wt}}$) linearly decrease throughput and energy efficiency. Yan et al. [29] time-share the

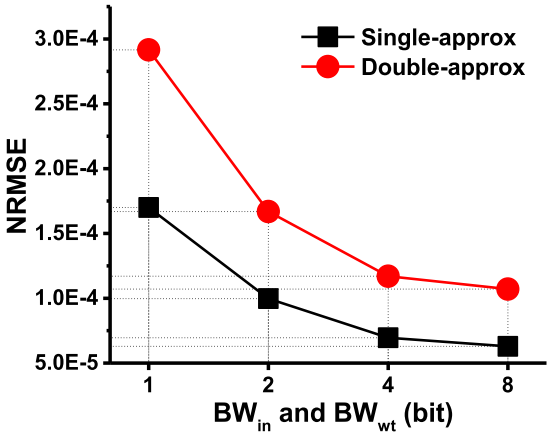


Fig. 21. NRMSE across different activation and weight bit widths.

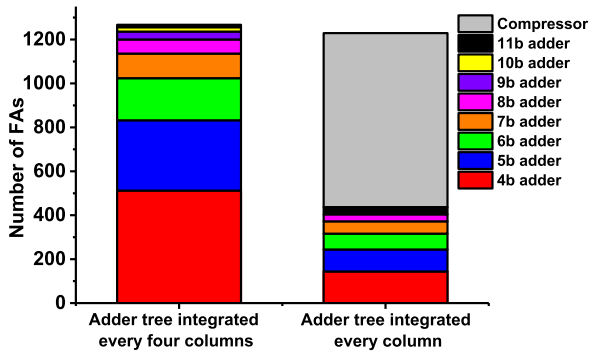


Fig. 22. Number of FAs required for adding four columns (4 b-weight) with 256 bitcells in each column.

arithmetic gate across multiple bitcells. This reuse architecture improves area efficiency (μm^2 per bit) by 2.2× and energy efficiency by 6.3% at 5.2× throughput degradation.

Fig. 20 shows the model-predicted tradeoff between the area and NRMSE. DIMCA2 has the smallest area but the largest NRMSE. On the other hand, the exact-arithmetic DIMC consumes more area. The 64 kb and 4 b/4 b, single-approximate DIMC consumes 1.67× less area than the counterpart [25]. The model predicts a 1 b/1 b single-approximate DIMC exhibit 1.47× worse NRMSE than 4 b/1 b. Note that the double-approximate DIMC can gain an additional 1.49× area reduction, but it exhibits 1.68× worse NRMSE, which matches the measurement results (see Fig. 5). We also show the model-predicted NRMSE across different bit widths in Fig. 21. NRMSE decreases as the input and weight bitwidth increase, and double-approximate DIMC generates 1.7× larger NRMSE than single-approximate DIMC across 1–8 b.

We also compare the area of macros employing different adder tree architectures. DIMCA integrates an adder tree in each column and sums the partial products of multibit weights and activations. In contrast, Chih et al. [25] employ one adder tree every four columns and process the multibit partial sums in a single adder tree. Fig. 22 shows the comparison results of the number of FA cells for those two architectures, which shows the total area is roughly the same. In both cases, we consider the nonapproximate arithmetic.

V. Conclusion

This article proposes two DIMCA prototypes, DIMC macros which employ approximate arithmetic hardware for improving area and power efficiency. The 28-nm test chip measurements show that DIMCA2 (DIMCA1) achieves the weight density of 485 kb/mm^2 (327 kb/mm^2), the energy efficiency of 2219 TOPS/W (990 TOPS/W), the CD of 607 TOPS/mm^2 (392 TOPS/mm^2), and 86.96% CIFAR-10 accuracy for a 1 b/1 b CNN model (90.41%). We also develop parameterized analytical models for the area, energy efficiency, throughput, and NRMSE of a DIMC array for design space exploration. We calibrate the models to recent test chip results.

References

[1] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.

[2] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision—ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 213–229.

[4] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 558–567.

[5] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[6] 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.

[7] C.-H. Lin et al., "A 3.4-to-13.3 TOPS/W 3.6 TOPS dual-core deep-learning accelerator for versatile AI applications in 7 nm 5G smartphone SoC," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 134–136.

[8] J.-S. Park et al., "A 6K-MAC feature-map-sparsity-aware neural processing unit in 5 nm flagship mobile SoC," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 152–154.

[9] A. Agrawal et al., "A 7 nm 4-core AI chip with 25.6TFLOPS hybrid FP8 training, 102.4TOPS INT4 inference and workload-aware throttling," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 144–146.

[10] Y. Ju and J. Gu, "A 65 nm systolic neural CPU processor for combined deep learning and general-purpose computing with 95% PE utilization, high data locality and enhanced end-to-end performance," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[11] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks," *IEEE J. Solid-State Circuits*, vol. 55, no. 6, pp. 1733–1743, Jun. 2020.

[12] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: An in-memory-computing SRAM macro based on robust capacitive coupling computing mechanism," *IEEE J. Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, Jul. 2020.

[13] 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.

[14] N. Verma et al., "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Mag.*, vol. 11, no. 3, pp. 43–55, Summer 2019.

[15] H. Jia et al., "Scalable and programmable neural network inference accelerator based on in-memory computing," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 198–211, Jan. 2022.

[16] A. Sebastian, M. L. Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnol.*, vol. 15, no. 7, pp. 529–544, Jul. 2020.

[17] H. Dbouk, S. K. Gonugondla, C. Sakr, and N. R. Shanbhag, "A 0.44-μJ/dec, 39.9-μs/dec, recurrent attention in-memory processor for keyword spotting," *IEEE J. Solid-State Circuits*, vol. 56, no. 7, pp. 2234–2244, Jul. 2021.

[18] J.-W. Su et al., "Two-way transpose multibit 6T SRAM computing-in-memory macro for inference-training AI edge chips," *IEEE J. Solid-State Circuits*, vol. 57, no. 2, pp. 609–624, Feb. 2022.

[19] J.-S. Seo et al., "Digital versus analog artificial intelligence accelerators: Advances, trends, and emerging designs," *IEEE Solid-State Circuits Mag.*, vol. 14, no. 3, pp. 65–79, Summer 2022.

[20] B. Zhang et al., "A 177 TOPS/W, capacitor-based in-memory computing SRAM macro with stepwise-charging/discharging DACs and sparsity-optimized bitcells for 4-Bit deep convolutional neural networks," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2022, pp. 1–2.

[21] X. Si et al., "A twin-8T SRAM computation-in-memory unit-macro for multibit CNN-based AI edge processors," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 189–202, Jan. 2020.

[22] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[23] J. Wang et al., "A 28-nm compute SRAM with bit-serial logic/arithmetic operations for programmable in-memory vector computing," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 76–86, Jan. 2020.

[24] H. Kim, T. Yoo, T. T. Kim, and B. Kim, "Colonnade: A reconfigurable SRAM-based digital bit-serial compute-in-memory macro for processing neural networks," *IEEE J. Solid-State Circuits*, vol. 56, no. 7, pp. 2221–2233, Jul. 2021.

[25] Y.-D. Chih et al., "An 89 TOPS/W and 16.3 TOPS/mm^2 all-digital SRAM-based full-precision compute-in memory macro in 22 nm for machine-learning edge applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 252–254.

[26] H. Fujiwara et al., "A 5-nm 254-TOPS/W 221-TOPS/mm^2 fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous MAC and write operations," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[27] C.-F. Lee et al., "A 12 nm 121-TOPS/W 41.6-TOPS/mm^2 all digital full precision SRAM-based compute-in-memory with configurable bit-width for AI edge applications," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2022, pp. 24–25.

[28] F. Tu et al., "A 28 nm 29.2 TFLOPS/W BF16 and 36.5 TOPS/W INT8 reconfigurable digital CIM processor with unified FP/INT pipeline and bitwise in-memory booth multiplication for cloud deep learning acceleration," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[29] B. Yan et al., "A 1.041-Mb/mm^2 27.38-TOPS/W signed-INT8 dynamic-logic-based ADC-less SRAM compute-in-memory macro in 28 nm with reconfigurable bitwise operation for AI and embedded applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 188–190.

[30] Y. He et al., "A 28 nm 38-to-102-TOPS/W 8b multiply-less approximate digital SRAM compute-in-memory macro for neural-network inference," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 130–132.

[31] Z. Yue et al., "7.7 CV-CIM: A 28 nm XOR-derived similarity-aware computation-in-memory for cost-volume construction," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 138–139.

[32] K. Kim, J. Lee, and K. Choi, "Approximate de-randomizer for stochastic circuits," in *Proc. Int. SoC Design Conf. (ISOCC)*, Nov. 2015, pp. 123–124.

[33] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 124–137, Jan. 2013.

[34] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in *Proc. 25th, Ed., Great Lakes Symp. VLSI*. New York, NY, USA: Association for Computing Machinery, May 2015, pp. 343–348, doi: 10.1145/2742060.2743760.

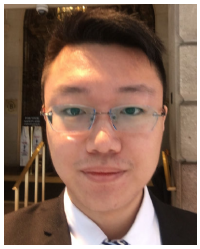
[35] S. K. Gonugondla, M. Kang, and N. R. Shanbhag, "A variation-tolerant in-memory machine learning classifier via on-chip training," *IEEE J. Solid-State Circuits*, vol. 53, no. 11, pp. 3163–3173, Nov. 2018.

[36] T. Dennis and J. F. Young, "Optical implementation of bipolar codes," *IEEE J. Quantum Electron.*, vol. 35, no. 3, pp. 287–291, Mar. 1999.

[37] D. Wang, C.-T. Lin, G. K. Chen, P. Knag, R. K. Krishnamurthy, and M. Seok, "DIMC: 2219TOPS/W 2569F2/b digital in-memory computing macro in 28 nm based on approximate arithmetic hardware," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 266–268.

Chuan-Tung Lin (Graduate Student Member, IEEE) received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2019, and the M.S. degree in electrical engineering from Columbia University, New York, NY, USA, in 2021, where he is currently pursuing the Ph.D. degree.

His research interests include energy-efficient architecture for machine-learning algorithms and VLSI circuit and system design.

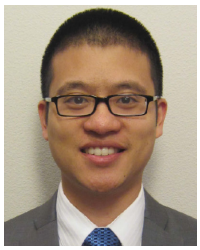


Dewei Wang received the B.S. degree in biomedical engineering (BME) from Xi'an Jiaotong University, Xi'an, China, in June 2017, and the M.S. degree in BME and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in December 2018 and May 2023, respectively.

His research interests include energy-efficient neuromorphic hardware and algorithm design.

Bo Zhang (Graduate Student Member, IEEE) received the B.S. degree in microelectronics (major) and mathematics (minor) from Shanghai Jiao Tong University, Shanghai, China, in 2018, and the M.S. degree in electrical engineering from Columbia University, New York, NY, USA, in 2020, where he is currently pursuing the Ph.D. degree in electrical engineering.

His research interests include computer architecture for machine-learning algorithms, in-memory computing architectures, and ultra-low-power VLSI circuit and system design.



Gregory K. Chen (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2006, 2009, and 2011, respectively.

He is currently a Researcher with the Circuit Research Laboratory, Intel Corporation, Hillsboro, OR, USA. His research interests include deep learning, RISC-V, neuromorphic computing, networks on chip (NoCs), and ultra-low-power computing.

Phil C. Knag (Member, IEEE) received the B.S. degree in computer engineering and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2010, 2012, and 2015, respectively.

He is currently a Researcher with the Circuit Research Laboratory, Intel Corporation, Hillsboro, OR, USA. He has authored 23 conference papers and journal articles and two book chapters. He holds more than 40 pending and issued patents. His research interests include machine learning, RISC-V, compute in/near memory, neuromorphic computing, and energy-efficient circuits.

Ram Kumar Krishnamurthy (Fellow, IEEE) received the B.E. degree in electrical engineering from the National Institute of Technology at Trichy, Tiruchirappalli, India, in 1993, the M.S. degree in electrical and computer engineering from The State University of New York, Buffalo, NY, USA, in 1994, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 1997.

He has been with Intel Corporation, Hillsboro, OR, USA, since 1997, where he is currently a Senior Principal Engineer with the Circuits Research Laboratory and heads the High-Performance and Low-Voltage Circuits Research Group. In this role, he leads research in high-performance, energy-efficient, and low-voltage circuits for next-generation microprocessors, accelerators, and systems-on-chip (SoCs). He is also an Adjunct Faculty with the Electrical and Computer Engineering Department, Oregon State University, Corvallis, OR, USA, where he taught advanced VLSI design. He has led circuit technology research directions in high-speed arithmetic units, on-chip interconnects, reconfigurable computing, energy-efficient clocking, ultralow voltage design, hardware security, compute-in-memory, neuromorphic computing, and machine-learning accelerators. He has made circuit technology contributions to multiple generations of Intel's data center, client, FPGA, the IoT, and AI products spanning across 180–7 nm process technology nodes. He has filed 350 patents and holds 215 issued patents. He has published 200 articles and four book chapters on high-performance and energy-efficient circuits.

Dr. Krishnamurthy has received two Intel Achievement Awards for pioneering the first 64-bit sparse-tree ALU technology and the first advanced encryption standard hardware security accelerator on Intel products. He has received the IEEE International Solid State Circuits Conference Distinguished Technical Paper Award, the IEEE European Solid State Circuits Conference Best Paper Award, the Outstanding Industry Mentor Award from SRC, Intel Awards for most patents filed and most patents issued, the Intel Labs Gordon Moore Award, the Alumni Recognition Award from Carnegie Mellon University, the Distinguished Alumni Award from the State University of New York, the MIT Technology Review's TR35 Innovator Award, and was recognized as a top ISSCC Paper Contributor. He serves as the Chair of the Semiconductor Research Corporation (SRC) Technical Advisory Board for the circuit design thrust. He has served as the Technical Program Chair and the General Chair for the IEEE International Systems-on-Chip Conference and currently serves on the Conference's Steering Committee. He has served as a Distinguished Lecturer for the IEEE Solid-State Circuits Society, a Guest Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS, and an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He has served on the Technical Program Committee for ISSCC, CICC, and SOCC conferences. He is a Board Member of the Industry Advisory Board for The State University of New York.

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 intelligent systems, machine-learning VLSI architecture and circuits, variation, voltage, aging, thermal-adaptive circuits and architecture, on-chip integrated power circuits, and nonconventional hardware design, including in-memory computing and analog-mixed-signal computing hardware.

Prof. Seok is/was the Technical Program Committee Member of several conferences, including the IEEE International Solid-State Circuits Conference (ISSCC) and ACM/IEEE Design Automation Conference (DAC). He has 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 has also received the 2009 AMD/CICC Scholarship Award for picowatt voltage reference work, the 2009 DAC/ISSCC Design Contest for the 35-pW sensor platform design, the 2015 NSF CAREER Award, the 2019 Qualcomm Faculty Award, and the Best Paper Award from IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS in 2022. He serves/served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS from 2013 to 2015, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS since 2015, and IEEE SOLID-STATE CIRCUITS LETTERS from 2017 to 2022. He has also served as the Guest Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS (JSSC). He was selected as the Solid-State Circuits Society (SSCS) Distinguished Lecturer for the period 2023–2025.