16.1 DIMC: 2219TOPS/W 2569F²/b Digital In-Memory Computing Macro in 28nm Based on Approximate Arithmetic Hardware

Dewei Wang[1], Chuan-Tung Lin[1], Gregory K. Chen[2], Phil Knag[2], Ram K. Krishnamurthy[2], Mingoo Seok[1]

[1]Columbia University, New York, NY
[2]Intel, Portland, OR

In-memory-computing (IMC) SRAM architecture has gained significant attention as it achieves high energy efficiency for computing a convolutional neural network (CNN) model [1]. Recent works investigated the use of analog-mixed-signal (AMS) hardware for high area and energy efficiency [2, 3]. However, AMS hardware output is well known to be susceptible to process, voltage, and temperature (PVT) variations, limiting the computing precision and ultimately the inference accuracy of a CNN. We reconfirmed, through the simulation of a capacitor-based IMC SRAM macro that computes a 256D binary dot product, that the AMS computing hardware has a significant root-mean-square error (RMSE) of 22.5% across the worst-case voltage, temperature (Fig. 16.1.1 top left) and 3-sigma process variations (Fig. 16.1.1 top right). On the other hand, we can implement an IMC SRAM macro using robust digital logic [4], which can virtually eliminate the variability issue (Fig. 16.1.1 top). However, digital circuits require more devices than AMS counterparts (e.g., 28 transistors for a mirror full adder [FA]). As a result, a recent digital IMC SRAM shows a lower area efficiency of 6368F²/b (22nm, 4b/4b weight/activation) [5] than the AMS counterpart (1170F²/b, 65nm, 1b/1b) [3]. In light of this, we aim to adopt approximate arithmetic hardware to improve area and power efficiency and present two digital IMC macros (DIMC) with different levels of approximation (Fig. 16.1.1 bottom left). Also, we propose an approximation-aware training algorithm and a number format to minimize inference accuracy degradation induced by approximate hardware (Fig. 16.1.1 bottom right). We prototyped a 28nm test chip: for a 1b/1b CNN model for CIFAR-10 and across 0.5-to-1.1V supply, the DIMC with double-approximate hardware (DIMC-D) achieves 2569F²/b, 932-2219TOPS/W, 475-20032GOPS, and 86.96% accuracy, while for a 4b/1b CNN model, the DIMC with the single-approximate hardware (DIMC-S) achieves 3814F²/b, 458-990TOPS/W (normalized to 1b/1b), 405-19215GOPS (normalized to 1b/1b), and 90.41% accuracy.

Figure 16.1.2 (left) shows the architecture of the proposed DIMC-D macro integrating 256×64 bitcells (DIMC-S has the same architecture except having 4b CPRS signals). We can store a 16k binary weight matrix in the macro, and by providing 256 bit-serial input activations from the left side of the macro, we can perform a binary vector-matrix dot-product in one cycle. Each of the 256-bitcell columns of the macro integrates 256 binary multiply cells, 16 approximate compressors, one 16-input adder tree, and one 11b shift accumulator. The 16 compressors count the number of 1's in the results of the 256 binary multiplications (MBL [0:255]) and generate 3b results (CPRS [0:2]). The adder tree sums up the outputs of the compressors. Finally, the shift accumulator accumulates the partial-sum of each cycle in a pipelined manner if input activations are bit-serial multi-bit values.

To improve the area efficiency of digital arithmetic hardware, we optimized the compressor and FA circuits. We designed three compressor circuits [6]. They are: exact (Fig. 16.1.2 center top), single-approximate (center middle), and double-approximate compressor (center bottom). The approximate compressors use interleaved AND and OR gates to replace FAs. While an AND gate can potentially cause -1 and an OR can cause +1 error, some of those errors can cancel each other. The double-approximate (single-approximate) compressor requires 55% (40%) fewer transistors than the exact counterpart, yet it exhibits the worst-case RMSE error of 6.76% (4.03%) over PVT variations (Fig. 16.1.2 right). The the worst-case RMSE of DIMC is smaller than that of AMS hardware (22.5%, Fig. 16.1.1 top), but the error still needs to be addressed to improve CNN accuracy.

Also, we have designed a custom 12T FA using pass-gate logic (Fig. 16.1.3 left) and a ripple-carry-adder (RCA) based on those FAs (Fig. 16.1.3 bottom right). The pass-gate logic has the well-known V_t drop problem. Therefore, we identified all nodes in a FA that do not have full-swing signals (marked in red in Fig. 16.1.3 top left). Then, we inserted inverters to ensure that the number of series-connected pass-gates is less than two. The inverters modify the RCA logic, and to keep the logic correct, we also made a second version of the 12T FA, which has *A_bar* and *B_bar* instead of *A* and *B* as inputs (schematic difference marked in red in Fig. 16.1.3 bottom left) and employed them accordingly in the RCA. The 12T FA consumes 1.764□m² (2250F²) (Fig. 16.1.3 top right). Through the area optimizations, each 256-bitcell column of DIMC-D having binary multipliers, compressors, adder tree, and shift-accumulator uses 4336 transistors, yielding device efficiency of 16.94T/b (Fig. 16.1.1 bottom left).

The optimized approximate arithmetic hardware negatively affects CNN accuracy. We benchmarked our approximate hardware using a VGG-like 1b/1b weight/activation CNN model (128C3-128C3-P2-256C3-256C3-P2-512C3-512C3-P2-FC1024-FC1024-FC10, 128C3: 128 features 3×3 convolution, P2: 2×2 pooling, FC1024: 1024 fully-connected) for CIFAR-10. Using the conventional training model, the version using double (single)-approximate hardware achieves a poor accuracy of 25.2% (50.9%), while the exact hardware achieves 89.6%. To compensate for the inaccuracy induced by the approximate hardware, we developed an approximation-aware training algorithm. In this algorithm, the forward path performs the vector-matrix multiplication using a bitwise operation considering the approximate hardware. Gradient calculations are performed using full accuracy for training. We then benchmarked the approximate hardware for the newly trained VGG-like 1b/1b CNN model and CIFAR-10. The double approximate version achieved higher accuracy of 86.9%, and the single approximate version achieved 89.0% — close to the exact hardware (Fig. 16.1.4 top left).

Interestingly, even with the approximation-aware training, the approximate hardware still results in lower accuracy for a multi-bit activation CNN model (Fig. 16.1.4 bottom right) because multi-bit activation tends to require more accurate hardware [3]. Specifically, multi-bit activations are often Gaussian distributed and thus MSBs are sparse and suffer from approximate errors. To improve the accuracy of a multi-bit activation CNN, we propose a new number format called multi-bit XNOR (MB-XNOR). Conventionally, in a 1b-weight neural network, each weight and activation represents +1 or -1 and XNOR realizes bitwise multiplication. If we use the 2's complement format for activations, however, the binary weight also needs to be in 2's complement and can represent only -1 or 0. We found that this results in large degradation to CNN accuracy. Therefore, we extended the format of the binary weight to represent an *N*-bit activation $b_{N-1}b_{N-2}\ldots b_0 = \sum_i b_i \times 2_i$, where b_i is +1 or -1. This format cannot represent 0, which disallows some of the activation functions such as ReLU. However, we can still use other popular activations such as hyperbolic tangent (tanh) (Fig. 16.1.4 top right) and leaky ReLU.

We confirmed that the proposed MB-XNOR format improves the accuracy of a multi-bit activation CNN model. We investigated the improvement both in SNR (signal-to-noise ratio) simulation and via CNN accuracy measurement. SNR is formulated as: $SNR = \sum y^2_{true} / \sum (y_{true}-y_{approx})^2$, where y_{true} is the ground truth of the dot product between a 256D Gaussian-distributed input vector quantized to 1-to-4b and a 256D binomial-distributed weight vector. y_{approx} is the same dot product but computed with approximate hardware. The DIMC-D macro with the 4b input activations in the MB-XNOR fomat yields a 0.15 higher SNR than 2's complement (Fig. 16.1.4 bottom left). The CNN accuracy measurement confirms the same improvement: DIMC-S using the MB-XNOR successfully increases the CNN accuracy by 5.4% (Fig. 16.1.4 bottom right). Despite that DIMC-D also benefits from the MB-XNOR format, the accuracy with multi-bit activations is still lower than that with binary activations, making DIMC-D suitable for only a 1b/1b weight/activation CNN model.

We prototype the DIMC test chip in 28nm (Fig. 16.1.7). The 16kb DIMC-D (DIMC-S) takes 0.033mm² (0.049mm²), implying an area efficiency of 2569F²/b (3814F²/b). We measured the macros at 0.5-1.1V at 25°C. DIMC-D achieves 932-2219TOPS/W and 475-20032GOPS; DIMC-S 458-990TOPS/W and 405-19215GOPS (normalized to 1b/1b for comparison) (Fig. 16.1.5 top left). We also measured the energy efficiency and throughput across five chips at the nominal voltage 0.9V (Fig. 16.1.5 top right), the energy efficiency across supply voltage at 25% and 50% input toggle rates (Fig. 16.1.5 bottom left). The power breakdown is shown in Fig. 16.1.5 bottom right. The SRAM mode takes 340ns (256 cycles at 752MHz) to update in total 16kb weights at 0.9V. Figure 16.1.6 shows the comparison to the recent work. The proposed DIMC macros achieve the high area efficiency, while maintaining the-state-of-the-art throughput, energy-efficiency and CNN accuracy.

Acknowledgement:
This work is supported by NSF (PFI-RP1919147) and SRC (TxACE 2810.034).

References:
[1] S. Srinivasa et al., "Trends and Opportunities for SRAM Based In-Memory and Near-Memory Computation," *ISQED*, pp. 547-552, 2021.
[2] H. Jia et al., "A Programmable Neural-Network Inference Accelerator Based on Scalable In-Memory Computing," *ISSCC*, pp. 236-237, 2021.
[3] Z. Jiang, S. Yin, J. Seo and M. Seok, "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," *IEEE JSSC*, vol. 55, no. 7, pp. 1888-1897, July 2020.
[4] K. Bowman et al., "Circuit techniques for dynamic variation tolerance," *ACM/IEEE Design Automation Conf.*, pp. 4-7, 2009.
[5] Y. -D. Chih et al., "An 89TOPS/W and 16.3TOPS/mm2 All-Digital SRAM-Based Full-Precision Compute-In Memory Macro in 22nm for Machine-Learning Edge Applications," *ISSCC*, pp. 252-253, 2021.
[6] K. Kim et al., "Approximate De-randomizer for Stochastic Circuits," *IEEE SoC Conf.*, pp. 123-124, 2015.
[7] H. Kim et al., "A 1-16b Precision Reconfigurable Digital In-Memory Computing Macro Featuring Column-MAC Architecture and Bit-Serial Computation," *ESSCIRC*, 2019.

2022 IEEE International Solid-State Circuits Conference (ISSCC) | 978-1-6654-2800-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISSCC42614.2022.9731659

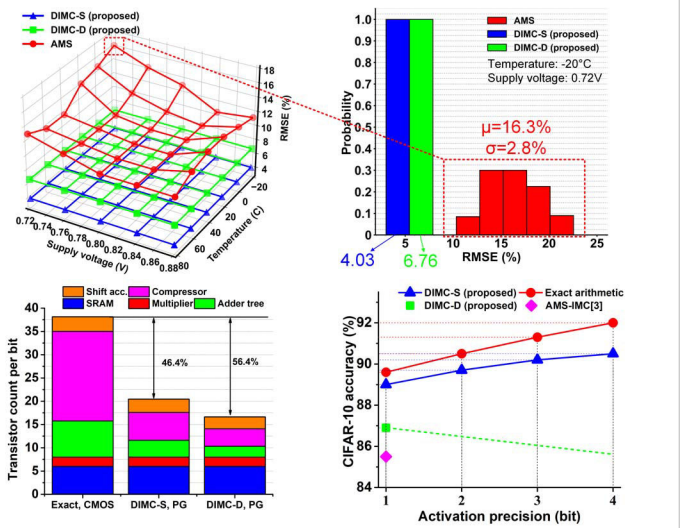

Figure 16.1.1: Voltage and temperature (top left) and process (top right) variations affect AMS computing hardware's accuracy. Approximate hardware improves the area efficiency of DIMC SRAM (bottom left). The custom training and number format improves CNN accuracy for CIFAR-10 (bottom right).

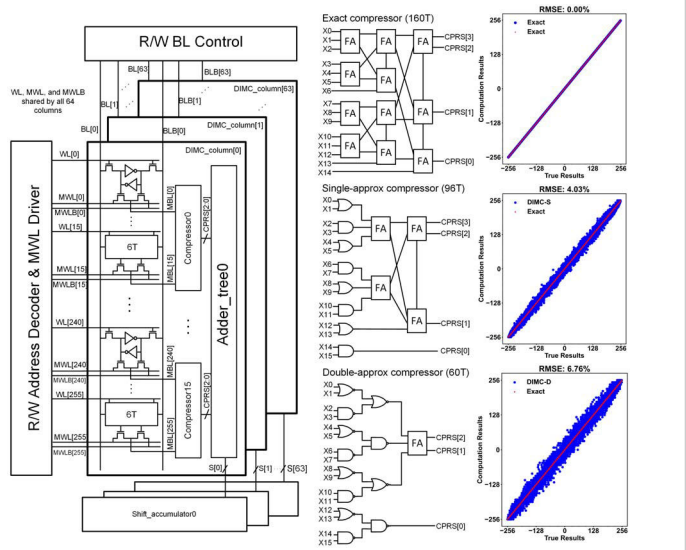

Figure 16.1.2: Proposed DIMC architecture (left). Three compressor schematics and the corresponding transistor count (middle). The RMSE of 256D binary dot product utilizing three types of compressors (right).

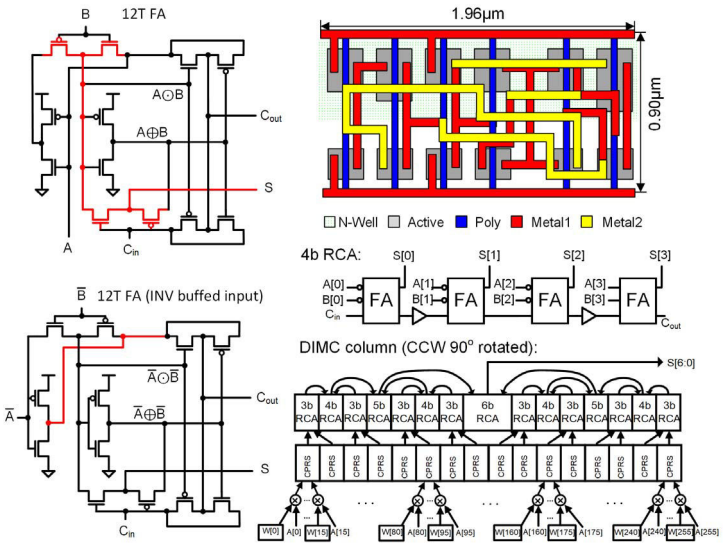

Figure 16.1.3: Two 12T-FA schematics with either regular inputs or inverter-buffered inputs (left). Layout of the 12T FA circuits (top right). Schematics of 4b RCA (middle right) and digital arithmetic hardware of one column (bottom right).

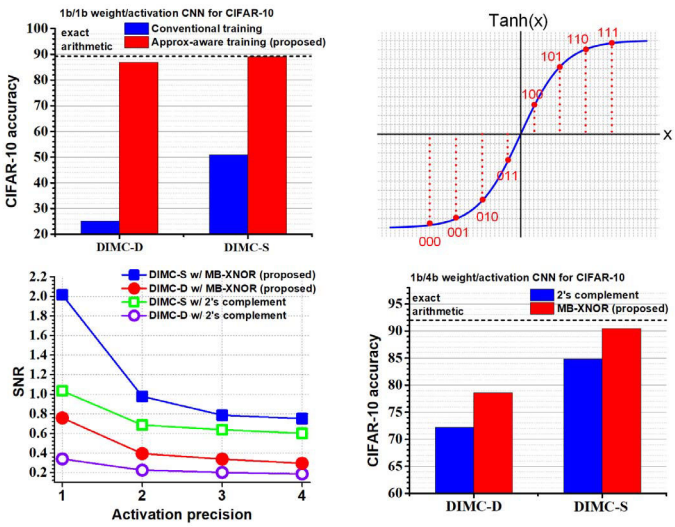

Figure 16.1.4: CIFAR-10 accuracy of conventional training and approximation-aware training (top left). Tanh activation quantized to 3b in the MB-XNOR format (top right). The MB-XNOR format offers better SNR (bottom left) and CIFAR-10 accuracy (bottom right) compared with 2's complement.

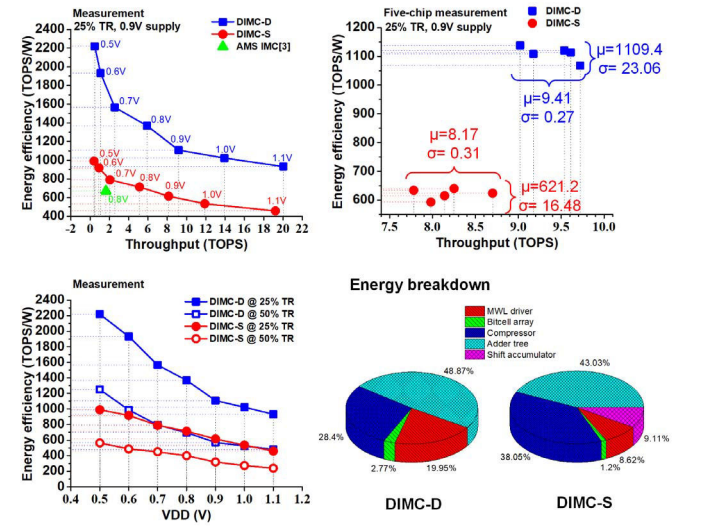

Figure 16.1.5: Measurement results. Energy-efficiency and throughput across different supply voltages (top left). Multi-chip measurements at 0.9V supply (top right). Energy-efficiency at 25% and 50% toggle rates (TR) (bottom left). Power breakdown of two proposed DIMC macros (bottom right).

	This work		ISSCC21[2]	JSSC20[3]	ISSCC21[5]	ESSCIRC19[7]
	DIMC-D	DIMC-S				
Technology[nm]	28	28	16	65	22	65
MAC operation	Digital	Digital	AMS	AMS	Digital	Digital
Array size	16Kb	16Kb	4.5Mb	16Kb	64Kb	16Kb
Macro size [mm^2]	0.033	0.049	11	0.081	0.202	0.227
Area efficiency [F^2/b]	2,569	3,814	9,179	1,170	6,368	3,279
Supply voltage [V]	0.45-1.10	0.45-1.10	0.8	0.8	0.72	0.6-0.8
Activation precision [bit]	1	1-4	1-8	1	1-8	1-16
Weight precision [bit]	1	1	1-8	1	4/8/12/16	4/8/12/16
Operating frequency [MHz]	280	250	20[1]	50	500	138
Input toggle rate	25%	25%	NA	NA	18%	NA
Energy efficiency [TOPS/W]	1,108 @ 0.9V	154 @ 0.9V (4b1b)	121 @ 0.8V (4b4b)	671 @ 0.8V	89 @ 0.72V (4b4b)	117 @ 0.6V (1b1b)
	2,219 @ 0.5V	248 @ 0.5V (4b1b)				
Throughput [GOPS][2]	9,175 @ 0.9V	2,035 @ 0.9V (4b1b)	41 @ 0.8V (4b4b)	1,638 @ 0.8V	825 @ 0.72 (4b4b)	567 @ 0.8V (1b1b)
	20,032 @ 1.1V	4,804 @ 1.1V (4b1b)				
CIFAR-10 accuracy	86.96%	90.41%	91.51%	85.50%	NA	NA

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

Figure 16.1.6: Comparison with recent IMC SRAMs using AMS or digital arithmetic hardware.

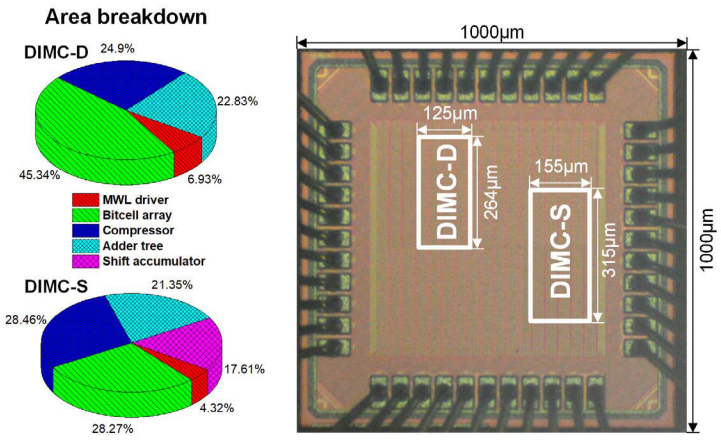

Figure 16.1.7: Die micrograph and area breakdown.

978-1-6654-2800-2/22/$31.00 ©2022 IEEE