PADCIM: A 1966.9-TOPS/W 2.09%-RMSE Probabilistic Approximate Dynamic-Logic Based Digital Computing-In-Memory Macro in 40nm

Wente Yi*, Zhenyu Xue*, Tianshuo Bai, Lehao Tan, Han Zhang, Jingcheng Gu, Jintao Mo, Wenjia Wang, Biao Pan†, and Weisheng Zhao
*Equally Credited Authors. Beihang University, Beijing, China (†Contact Email: panbiao@buaa.edu.cn)

***Abstract*—This work presents a 1966.9-TOPS/W 2.09%-RMSE digital computing-in-memory (CIM) macro named PADCIM. It has three key features: 1) Probabilistic approximate adder tree (PAAT) coupled with delay clock chain, achieve high energy efficiency and low RMSE while resolving slope deviation. 2) Full dynamic-logic computing circuits (FDCC) combined with NP-Domino cascading, reduce chip power and area. 3) XOR-based multi-bit computation scheme (XMCS) integrated with bias-error compensation, enable accurate binary and multi-bit computations. Fabricated in 40nm technology, PADCIM demonstrates 1.12× higher energy efficiency and 1.92× lower RMSE than the SOTA ADCIM. It further achieves up to 1.70× higher energy efficiency over 28nm/12nm-nodes chips, while delivering 87.56% (1b/1b) and 91.19% (4b/1b) inference accuracy on CIFAR-10.**

***Index Terms*—Digital computing-in-memory, probabilistic approximate, full dynamic-logic, multi-bit computation**

I. Introduction

Computing-in-Memory (CIM), as a chip architecture that integrates computing capabilities into memory, directly solves the challenges of the *Memory Wall* caused by the compute-storage separation in the *Von Neumann* architecture, and offers high computational accuracy and energy efficiency. From the perspective of the CIM computing paradigm, it can be categorized into digital CIM (DCIM) and analog CIM (ACIM). DCIM provides higher computational accuracy, while ACIM achieves superior energy and area efficiency [1] [2].

To leverage the advantages of two computational paradigms, approximate DCIM (ADCIM) has been proposed [1]. When compared with precise DCIM, ADCIM achieves a significant reduction in adding unit area while maintaining arithmetic precision (Fig. 1(a), left). By employing a 1-bit approximate design, the ADCIM with approximate adder tree (AT) implementation achieves 35% area reduction and 2.7× energy efficiency improvement (Fig. 1(a), top right). Experimental results reveal that the root mean square error (RMSE) of ACIM exhibits significant sensitivity to process variations, voltage fluctuations, and temperature (PVT) changes [2]. In contrast, the computational error of ADCIM is substantially reduced and exhibits insensitivity to PVT variations (Fig. 1(a), bottom). Furthermore, existing research demonstrates that neural networks can tolerate a certain level of computational error. Therefore, integrating ADCIM into neural network accelerators represents a promising and viable approach.

However, previously reported ADCIM suffers from these limitations: 1) Existing approaches focus on approximate design at discrete device-level, which leads to non-negligible

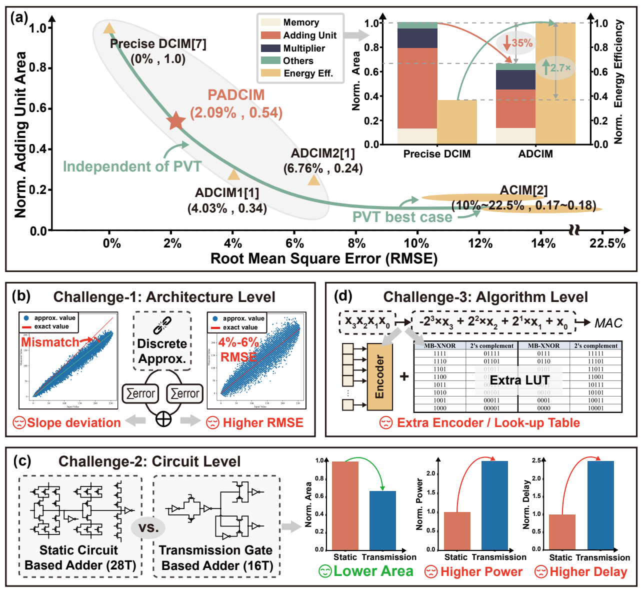


Fig. 1: (a) ADCIM integrates the advantages of both DCIM and ACIM, (b)(c)(d) three challenges of previously reported ADCIM design.

RMSE (4-6%) [1] and induces slope deviation during parallel computing due to error accumulation (Fig. 1(b)) [3]. 2) While transmission-gate adder tree (TG-AT) achieves transistor count reduction, it exhibit suboptimal power and delay performance, and requires additional level restoration circuit to ensure logical integrity (Fig. 1(c)) [4]. 3) The existing approach employs a multi-bit encoding scheme with look-up table (LUT) circuits to enable binary and multi-bit computing on unified hardware, but introduces additional encoders and LUTs, thereby increasing hardware resource overhead (Fig. 1(d)) [5].

In this work, we introduce a probabilistic approximate DCIM with dynamic-logic circuits (PADCIM), overcoming the challenges outlined above through three key features:

- **PAAT**, a probabilistic approximate AT replacing conventional designs with AND/OR logic. It splits precise computations into two equal-probability schemes combined with delay clock chain, achieving state-of-the-art (SOTA) RMSE value while overcoming slope deviation.
- **FDCC**, a full dynamic-logic computing circuits for multiply-accumulate (MAC) operations using 14T multipliers and NP-Domino AT, enabling cross-layer cascading while reducing power and transistor count.
- **XMCS**, an XOR-based multi-bit computation scheme.

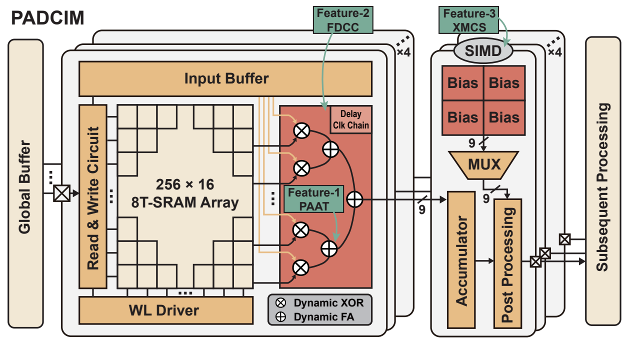


Fig. 2: Overall architecture and main features of PADCIM Macro.

It approximates multi-bit numbers in two's complement numbers while introducing statistically derived bias values to mitigate errors, enabling unified hardware support for both binary and multi-bit computations.

II. PADCIM Accelerator Architecture

Overall Architecture. Fig. 2 depicts the overall architecture of PADCIM. The system consists of an 8Kb input buffer, a 16Kb 8T SRAM-based weight storage array, dynamic XOR multipliers, a PAAT, a delay clock chain and a bias compensation SIMD. Input feature maps are temporarily stored in the input buffer, where MAC computations are executed in the FDCC with weights pre-stored in the 8T SRAM array. The SIMD receives computation results and supports both binary and multi-bit computations based on XMCS.

Main Features. Fig. 2 marks three features of PADCIM. PADCIM adopts a PAAT to implement its approximate computing paradigm. This design decomposes precise computations into two probabilistic approximate schemes with equal probabilities, constructs an AT using two distinct approximate half-adders (AHA1 and AHA2), and achieves significant reductions in power and area while maintaining low RMSE (**Feature 1**). For MAC operations, the FDCC leverages dynamic-logic to implement multipliers and adders, alongside NP-Domino logic for cross-layer cascading. This architecture optimizes area, power, and latency through device group optimization (**Feature 2**). PADCIM further employs an XMCS to support binary and multi-bit computations. By applying approximate encoding to multi-bit numbers in two's complement representation and integrating statistically derived bias values to mitigate encoding errors, the design eliminates additional encoder and LUT circuits, thereby enhancing system energy and area efficiency (**Feature 3**). The following subsections elaborate on these three features.

A. Probabilistic Approximate Adder Tree

In this section, we propose the architecture and computational dataflow of PAAT. As depicted in Fig. 3(a), the design incorporates two distinct approximate half-adders: AHA1 utilizes an OR gate configuration to suppress redundant carry, while AHA2 implements a parallel OR-AND gate structure to retain critical carry. Truth table analysis reveals that precise computation ($1 + 1 = 2'b10$) is approximated to either $1'b1$ (AHA1) or $2'b11$ (AHA2) exclusively when both inputs reach

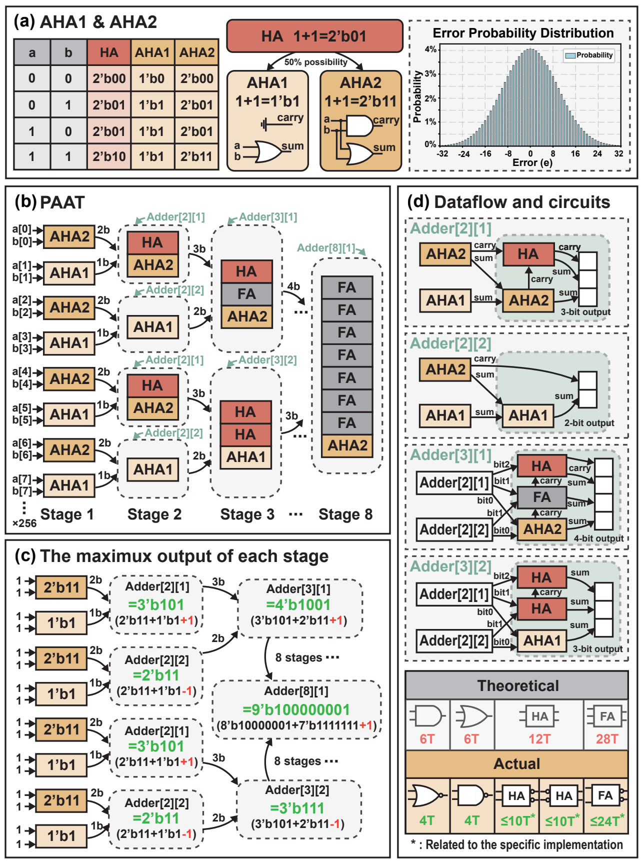

(a) AHA1 & AHA2
a | b | HA | AHA1 | AHA2
0 | 0 | 2'b00 | 1'b0 | 2'b00
0 | 1 | 2'b01 | 1'b1 | 2'b01
1 | 0 | 2'b01 | 1'b1 | 2'b01
1 | 1 | 2'b10 | 1'b1 | 2'b11
HA 1+1=2'b01
50% possibility
AHA1 1+1=1'b1
AHA2 1+1=2'b11
carry
sum
Error Probability Distribution
Probability
Error (e)
(b) PAAT
Adder[2][1]
Adder[2][2]
Adder[3][1]
Adder[3][2]
Adder[8][1]
Stage 1
Stage 2
Stage 3
Stage 8
×256
(c) The maximux output of each stage
Adder[2][1] =3'b101 (2'b11+1'b1+1)
Adder[2][2] =2'b11 (2'b11+1'b1-1)
Adder[3][1] =4'b1001 (3'b101+2'b11+1)
Adder[3][2] =3'b111 (3'b101+2'b11-1)
8 stages ···
Adder[8][1] =9'b100000001 (8'b10000001+7'b1111111+1)
(d) Dataflow and circuits
2-bit output
3-bit output
4-bit output
Theoretical
6T
6T
12T
28T
Actual
4T
4T
≤10T*
≤10T*
≤24T*
* : Related to the specific implementation

Fig. 3: (a) Truth tables, schematics and error probability distribution of two AHAs, (b)(c)(d) the architectural and computational dataflow of PAAT.

maximum values. To minimize the impact of approximate errors on results, PAAT integrates both AHA variants in an equiprobable alternating topology. Error estimation modeling of the implemented PAAT demonstrates that error probability distribution under 256-bit parallel stochastic inputs exhibits a characteristic bell-shaped curve centered at zero. This observation confirms that PAAT delivers relatively precise results in most cases with a low probability of significant errors.

Fig. 3(b) illustrates the hierarchical architecture of PAAT. Each stage features optimized bit-width management: AHA1 and $Adder[n][2]$ reduce computational bit-width requirements by 1-bit, while AHA2 and $Adder[n][1]$ maintain conventional bit-width outputs. Specifically, the initial layer (Stage 1) employs alternating AHA1/AHA2 configurations, generating 1-bit or 2-bit outputs respectively. In Stage 2 and subsequent layers, these two adder types produce n-bit or (n+1)-bit outputs according to the presence of carry, culminating in an 8-stage pipelined architecture supporting 256-bit parallel processing. The computational dataflow, detailed in Fig. 3(d), follows this progressive pattern through all pipeline stages.

Computational errors only occur during maximum value computation in each layer. Fig. 3(c) demonstrates the maximum output situation, where parallel all-ones inputs exhibit a ± 1 deviation from the true value. Statistical modeling with randomly generated data confirms SOTA RMSE performance. The PAAT implementation further incorporates logic inversion optimization, reconstructing conventional AND/OR gates and

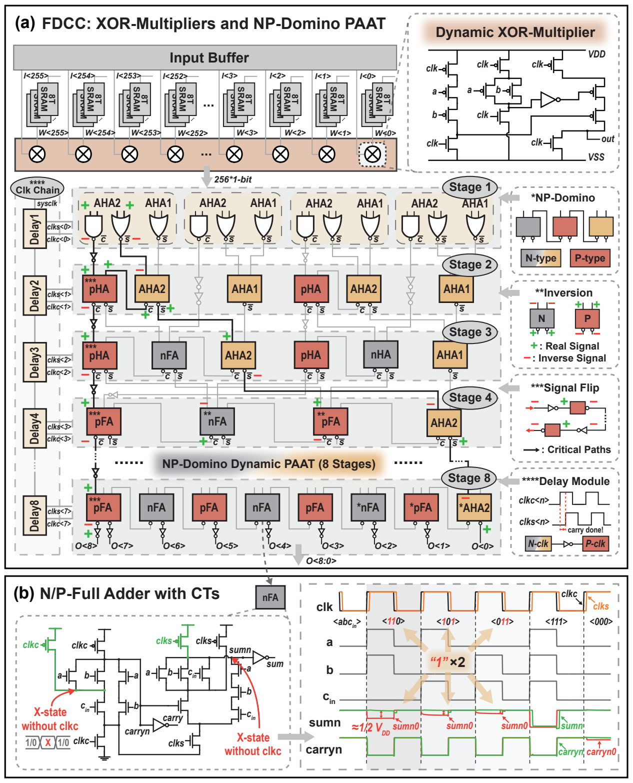


Fig. 4: (a) Schematic diagram of the cascaded XOR-multipliers and 8 stages NP-Domino PAAT in FDCC, (b) N/P-full adder with CTs.

AHA modules into NAND/NOR-based primitives as illustrated in Fig. 3(d). This transformation significantly reduces transistor count, thereby improving area efficiency.

B. Full Dynamic-Logic Computing Circuits

This section elaborates the design of FDCC, which achieves optimizations in circuit area, power, and delay by employing NP-Domino dynamic-logic and a delay clock chain. As illustrated in Fig. 4(a), the architecture cascades dynamic XOR multipliers with the NP-Domino PAAT. The 14T dynamic XOR multiplier executes 256 parallel binary computations per cycle and feeds the results to the PAAT. The PAAT employs an eight-stages adder array, where each stage utilizes complementary N-type and P-type logic to construct an interleaved topology, ensuring correct carry signal propagation between N-type and P-type adders. To address polarity inversion in N/P-type adder outputs, the eight-stages design integrates inverters on critical paths for even-stage signal restoration. Furthermore, a delay-based clock synchronization chain is proposed, which delays the sum circuit's clock until carry computation completes, ensuring correct signal transfer between stages with uniform delay per adder.

Conventional dynamic-logic circuits suffer from leakage current issues, requiring redundant keeper transistors to mitigate errors caused by voltage droop. In contrast, the AT in FDCC uses a timing constrained computation scheme that terminates computations before significant output voltage droop occurs, thereby eliminating the requirement for keeper transistors. Additionally, as shown in Fig. 4(b), the FDCC

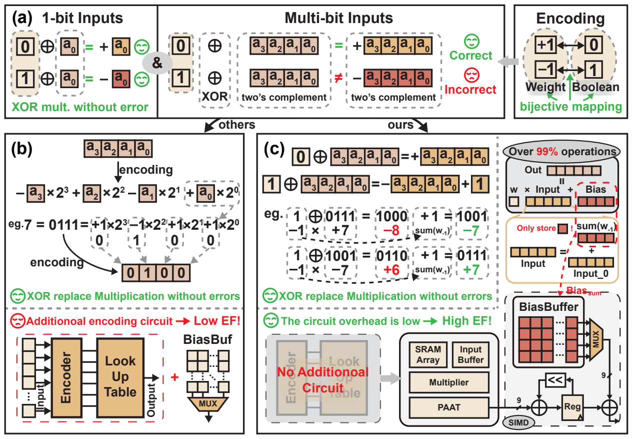


Fig. 5: (a) XOR multiplication introduces errors with multi-bit inputs, (b) others mitigate errors through encoding data, (c) XMCS introduces bias and SIMD implementation, eliminating encoder/decoder circuits and LUTs.

full adder adds compensation transistors (CTs) to solve charge sharing effects. When two of the three inputs in a full adder without CTs are logic "1", charge leakage causes unstable voltage. By adding CTs at high capacitance nodes, leakage current is reduced, ensuring correct computation for all input cases. Fig. 4(b) shows only the N-type full adder circuit; the P-type version simply inverts the MOSFET logic to achieve the same function.

C. XOR-based Multi-bit Computation Scheme

In this section, we propose XMCS that maps weights to the Boolean domain and computes with two's complement inputs to enable XOR-based multiplication without error. As shown in Fig. 5(a), quantized 1-bit weights $\{+1, -1\}$ mapped to the Boolean domain $\{0, 1\}$ demonstrate zero-error characteristics when performing XOR multiplication with 1-bit inputs. However, computational deviations emerge when applied to multi-bit inputs. A conventional solution implements powers of two for bit position weight scaling to recode multi-bit inputs prior to XOR multiplication, as illustrated in Fig. 5(b), but requires additional hardware resources. Our work observe that when the Boolean-mapped weight is 1 (actual weight value is -1), the computational result consistently undercounts the true value by 1 (Fig. 5(c), top left). This insight enables compensation through pre-counting the number of weights with actual weight value -1 as $sum(w_{-1})$, which is directly accumulated with XOR multiplication and approximate addition results in subsequent stages to restore arithmetic accuracy.

Taking the convolutional neural network ResNet-14 as an example, convolutional layers, fully connected layers, and BatchNorm operations account for over 99% of the network's computational workload. These operations can be uniformly expressed by $Out = Weight \times Input + Bias$. Furthermore, as illustrated in Fig. 5(c), we can merge the network's intrinsic $Bias$ with the statistical sum of weights $sum(w_{-1})$ to obtain $Bias_{sum}$, which is pre-stored in the SIMD's BiasBuffer and directly accumulated with results generated by PAAT. Through the architectural synergy between SIMD and XMCS, PADCIM achieves high-efficiency MAC computations while eliminating encoder/decoder circuits and LUT structures.

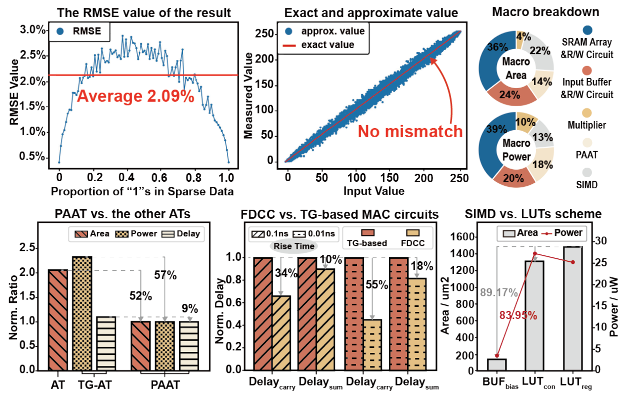
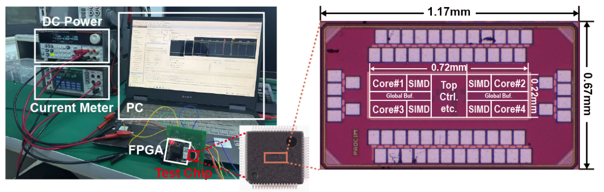
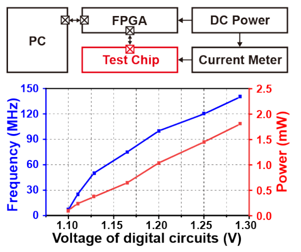


Fig. 6: Experimental data of the hardware implementation.





Technology	40nm
Chip Area	0.72mm×0.22mm
CIM Macro Area[*1]	0.25mm×0.088mm×4
CIM Capacity	24Kb
Voltage	1.10-1.29V
Frequency	7-140MHz
System Power	0.11-1.80mW
System Energy Efficiency[*2] (@I/W)	1966.90TOPS/W[*3](@1b/1b)
	278.83TOPS/W[*4](@4b/1b)

*1: CIM macro area includes both of the CIM Core and SIMD. *2: Input toggle rate @10%, weight sparsity @50%. *3: @140MHz, 1.10V. *4: @50MHz, 1.13V.

Fig. 7: Chip photograph, voltage-frequency scaling and specification.

III. Measurement Results

Fig. 6 presents experimental data of the hardware implementation. Under the 1-bit/1-bit computing mode with a parallelism of 256, the output results achieve an average RMSE of 2.09% by controlling the proportion of "1"s in inputs, with no slope deviation observed in 10^4 randomized statistical measurements (Fig. 6, top left). The area and power breakdown of the PADCIM macro components are illustrated (Fig. 6, top right). PAAT achieves 52% area reduction compared with precise AT, and 57%/9% power/delay reductions compared with TG-AT (Fig. 6, bottom left). FDCC reduces carry/sum delays by 34%/10% (rise time = 0.1ns) and 55%/18% (rise time = 0.01ns) compared with TG-based MAC circuits (Fig. 6, bottom middle). XMCS saves 89.17%/83.95% area/power compared with LUT-based schemes (Fig. 6, bottom right).

The fabricated 40nm chip and its test platform are shown in Fig. 7. The platform is composed of a PADCIM test chip, Xilinx Z-7035 FPGA carrier board, dc power, current meter and a computer. Test dataset are preloaded into FPGA's DDR memory and processed after power initialization. Results are retrieved through the FPGA interface and transmitted to the host computer for post processing. Experimental results demonstrate the PADCIM can work at 7-140MHz with 1.10-1.29V system voltage, delivering average energy efficiency of 1966.90TOPS/W (1b/1b) and 278.83TOPS/W (4b/1b).

Fig. 8 presents a comparison table. Compared with the SOTA ADCIM (28nm), PADCIM (40nm) achieves 1.12× energy efficiency improvement and 1.92× RMSE reduction, delivers 0.66% (1b/1b) and 0.78% (4b/1b) percentage point improvements in CIFAR-10 inference accuracy. When benchmarked against dynamic DCIM and other CIMs, PADCIM demonstrates 1.13× (28nm), 1.01× (12nm), and 1.70× (28nm) energy efficiency improvements respectively.

	JSSC'24 C. Lin[1]		This Work (PADCIM)	ISSCC'22 B. Yan[6]	VLSI'22 C. Lee[7]	ISSCC'23 Z. Yue[8]
	DIMCA2	DIMCA1				
Technology	28nm	28nm	40nm	28nm	12nm	28nm
CIM Type	Appro. Digital	Appro. Digital	Appro. Dynamic Digital	Dynamic Digital	Precise Digital	Analog+Digital
Cell Type	6T-SRAM	6T-SRAM	8T-SRAM	6T-SRAM	6T-SRAM	6T-SRAM
Array Size	16Kb	16Kb	24Kb	32Kb	8Kb	36Kb
Macro Area (mm^2)	0.049	0.049	0.088	0.030	0.032	0.052
Storage Density (Mb/mm^2)	0.327	0.327	0.273	1.067	0.248	0.687
Power Supply(V)	0.45-1.1	0.45-1.1	1.1-1.29	0.8	0.72	0.6-0.9
Frequency(MHz)	280	250	7-140	333	N/A	50-286
Input Precision	1b	1b-4b	1b-4b	1b-8b	4b-8b	1b/4b
Weight Precision	1b	1b	1b	1b/4b/8b	4b/8b	1b-8b
Input Toggle Rate	25%	25%	25%	25%	N/A	N/A
Calculation Error	6.76%[*1]	4.03%[*1]	2.09%[*1]	N/A	N/A	N/A
Energy Efficiency @(I/W)(TOPS/W)	2219(1b/1b)	248(4b/1b)	1966.9(1b/1b) 278.8(4b/1b)	27.4(8b/8b)	30.3(8b/8b)	1158(1b/1b)
CIFAR-10 Accuracy	86.90%[*2]	90.41%[*2]	87.56%[*3] 91.19%[*3]	N/A	N/A	N/A

*1: Average RMSE value. *2: No model information available. *3: @ResNet-14, 3rd residual block approximated.

Fig. 8: Performance comparison table.

IV. Conclusions

In this work, we propose a probabilistic approximate dynamic-logic based DCIM accelerator. Through co-design of circuit, architecture, and algorithm, we develop approximate logic for dynamic MAC units and introduce an NP-Domino interconnective scheme in hierarchical architectures, integrated with an XOR-based multi-bit computation algorithm. PADCIM achieves optimal energy efficiency and RMSE trade-offs, demonstrating 1.12× higher energy efficiency and 1.92× lower RMSE than SOTA ADCIM, alongside 0.66% (1b/1b) and 0.78% (4b/1b) accuracy improvement on CIFAR-10.

Acknowledgment

This work is supported partly by National Key Research and Development Program of China (Grant No. 2021YFB3601304, 2021YFB3601300), partly by Joint laboratory fund of Beihang University and SynSense.

References

[1] C. Lin *et al.*, "DIMCA: An Area-Efficient Digital In-Memory Computing Macro Featuring Approximate Arithmetic Hardware in 28 nm," in *IEEE JSSC*, vol. 59, no. 3, pp. 960-971, 2024.
[2] S. -E. Hsieh *et al.*, "A 70.85-86.27TOPS/W PVT-Insensitive 8b Word-Wise ACIM with Post-Processing Relaxation," in *IEEE ISSCC*, pp. 136-138, 2023.
[3] A. Guo *et al.*, "A 22nm 64kb Lightning-Like Hybrid Computing-in-Memory Macro with a Compressed Adder Tree and Analog-Storage Quantizers for Transformer and CNNs," in *IEEE ISSCC*, vol. 67, pp. 570-572, 2024.
[4] P. Deaville *et al.*, "A 22nm 128-kb MRAM Row/Column-Parallel In-Memory Computing Macro with Memory-Resistance Boosting and Multi-Column ADC Readout," in *Symp. on VLSI Circuits*, pp. 268-269, 2022.
[5] S. Bavikadi *et al.*, "ReApprox-PIM: Reconfigurable Approximate Look up-Table (LUT)-Based Processing-in-Memory (PIM) Machine Learning Accelerator," in *IEEE TCAD*, vol. 43, no. 8, pp. 2288-2300, 2024.
[6] B. Yan *et al.*, "A 1.041-Mb/mm2 27.38-TOPS/W Signed-INT8 Dynamic-Logic Based ADC-less SRAM Compute-In-Memory Macro in 28nm with Reconfigurable Bitwise Operation for AI and Embedded Applications," in *IEEE ISSCC*, vol. 65, pp. 188-190, 2022.
[7] C. -F. Lee *et al.*, "A 12nm 121-TOPS/W 41.6-TOPS/mm2 All Digital Full Precision SRAM-based Compute-in-Memory with Configurable Bit-width For AI Edge Applications," in *Symp. on VLSI Circuits*, pp. 24-25, 2022.
[8] Z. Yue *et al.*, "CV-CIM: A 28nm XOR-Derived Similarity-Aware Computation-in-Memory for Cost-Volume Construction," in *IEEE ISSCC*, pp. 138-140, 2023.