PROTEUS: A 40nm Programmable General-Purpose Digital Compute-In-Memory Accelerator with eNVM and Hierarchical ISA for Versatile Edge AI

Luqi Zheng, *Graduate Student Member, IEEE,* Amir Massah Bavani, *Graduate Student Member, IEEE,* Mufeng Chen, Shuting Du, *Graduate Student Member, IEEE,* Te-Yu Hsin, Win-San Khwa, *Senior Member, IEEE,* Ping-Sheng Wu, Ashwin Sanjay Lele, Bo Zhang, Brian Crafton, Harry Chuang, Yu-Der Chih, *Member, IEEE,* Meng-Fan Chang, *Fellow, IEEE,* and Haitong Li, *Member, IEEE*

***Abstract*—We present PROTEUS, an 18 mm^2 programmable general-purpose digital compute-in-memory (GP-DCIM) accelerator integrating 4 Mb RRAM and 2.6 Mb tensor SRAM with a 32-bit hierarchical DCIM instruction set architecture (ISA). PROTEUS features fine-grained 1-D matrix tiling and a reconfigurable DCIM datapath/pipeline for near-100% memory utilization, supporting INT8/INT16/FP8/FP16 DCIM computations. PROTEUS unifies SRAM/RRAM dataflows and embeds non-volatile micro-programs in RRAM to enable rapid switching among pre-stored kernels without incurring off-chip instruction feeds or RRAM rewrites. Fabricated in 40 nm ULP CMOS with foundry RRAM, PROTEUS delivers 702 GOPS throughput, 6.4 TOPS/W energy efficiency, and 0.039 TOPS/mm^2 compute density. It is validated on ResNet-20, BERT-Tiny, MobileViT, GraphSAGE, and Vision Mamba, demonstrating versatility across CNN, Transformer, hybrid CNN-Transformer, Graph Neural Network (GNN), and State-Space Model (SSM) workloads.**

I. Introduction

COMPUTE-in-memory (CIM) has rapidly advanced as a promising hardware fabric for edge AI by executing dominant tensor primitives inside memories, thereby reducing energy-hungry data movement. Across silicon CMOS memories, SRAM-based CIM has evolved from early in-array dot-product and vector primitives targeting CNN and DSP workloads [1], [2], [3], [4], [5], to fully-digital and multi-tiled implementations that push efficiency frontiers in advanced nodes [6], [7], [8]. In parallel, recent SRAM-CIM designs further explore multiple axes of circuit- and system-level innovations, including analog/mixed-signal techniques for quantization-aware CIM [9] and extreme energy efficiency [10], multiply-less approximation reducing multiplication costs [11], plasticity for on-chip learning [12], floating-point (FP) and mixed-precision numerical support [13], [14], [15], [16], and Transformer-oriented kernels/dataflows with sparse/structured mappings [13], [17]. Driven by the needs for higher density while staying in silicon, eDRAM-based CIM macros [18], [19], [20], [21], [22], [23] have exploited the density advantage [22], multi-mode operations [21], [19], [20], and efficient FP 3D-MAC utilizing the gain-cell structure [23]. Beyond silicon charge-based memories, embedded non-volatile memory (eNVM) based CIM designs have been extensively explored. For instance, RRAM-CIM macros and systems, some combined with near-SRAM [24] or in-SRAM

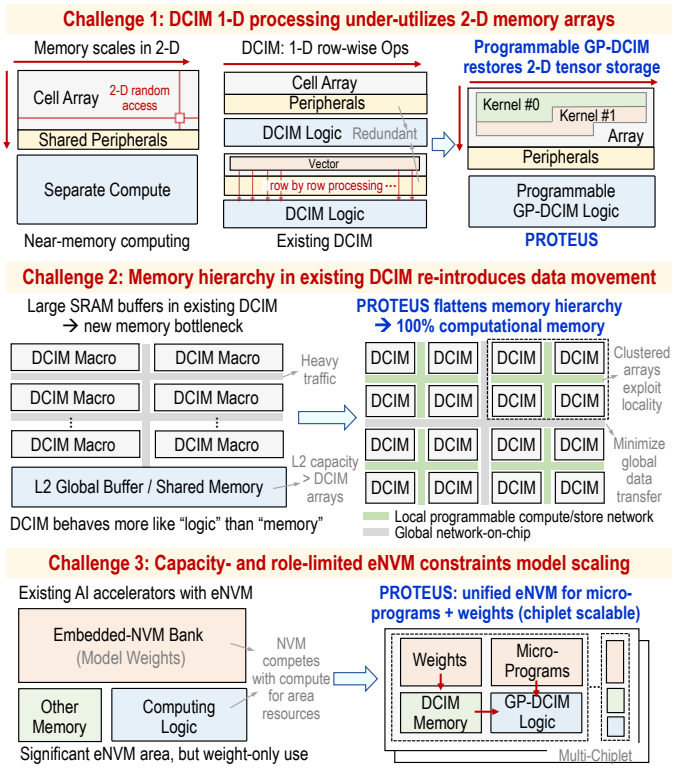


Fig. 1. Memory-centric view of architectural challenges in digital compute-in-memory (DCIM) and NVM-embedded accelerators, which informs the design goals of the GP-DCIM fabric and CIM-native ISA in PROTEUS.

This work was supported in part by the U.S. National Science Foundation FuSe2 program (Award No. 2425498). *(Luqi Zheng and Amir Massah Bavani contributed equally to this work.) (Corresponding author: Haitong Li.)*

Luqi Zheng, Amir Massah Bavani, Mufeng Chen, Shuting Du, Te-Yu Hsin, and Haitong Li are with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA (e-mail: haitongli@purdue.edu).

Win-San Khwa, Ping-Sheng Wu, and Meng-Fan Chang are with Corporate Research, Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu 30075, Taiwan.

Ashwin Sanjay Lele, Bo Zhang, and Brian Crafton are with Corporate Research, Taiwan Semiconductor Manufacturing Company (TSMC), San Jose, CA 95134, USA.

Harry Chuang is with Embedded Technology Division, Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu 30075, Taiwan.

Yu-Der Chih is with Design Technology Platform, Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu 30075, Taiwan.

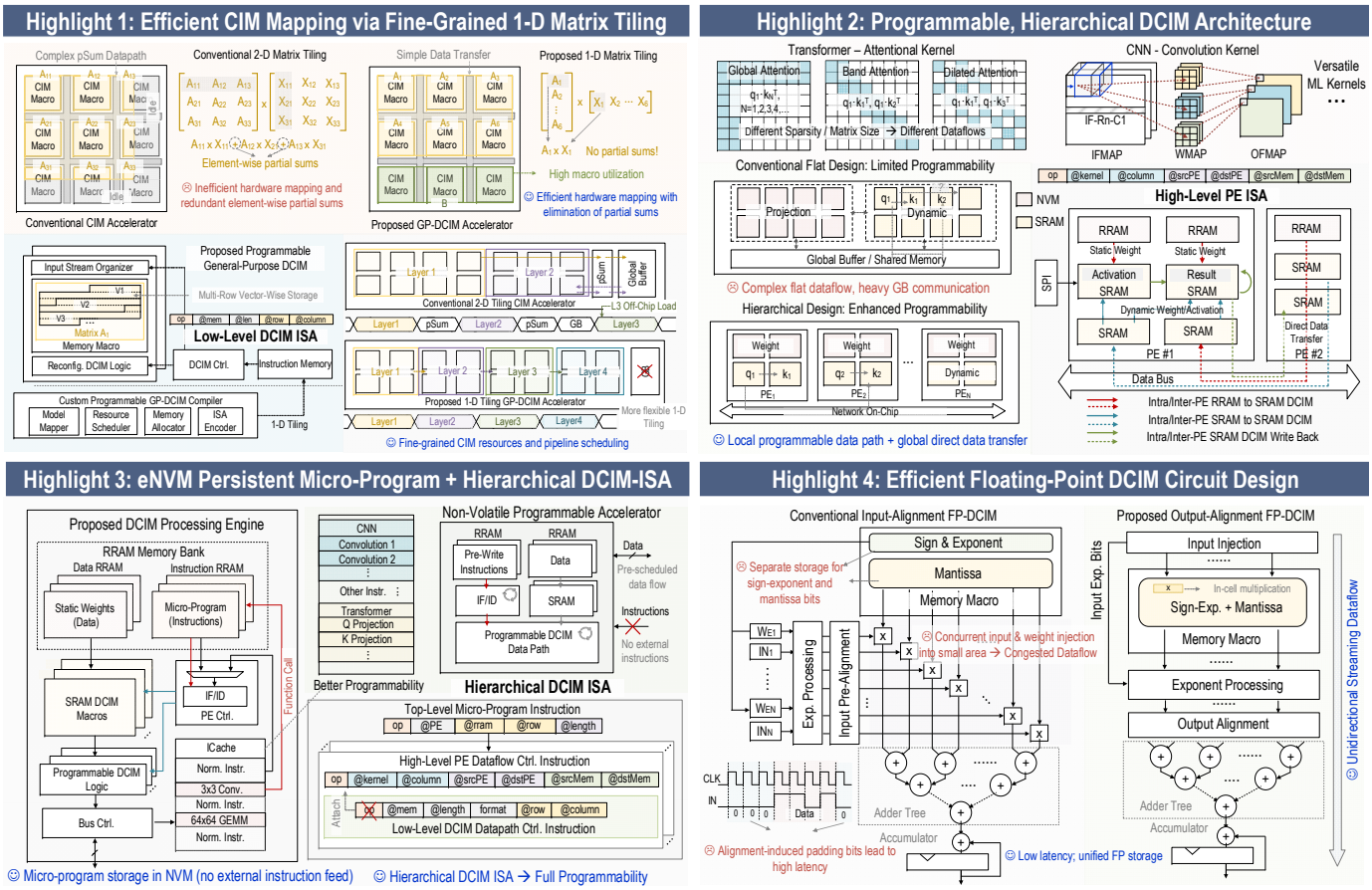


Fig. 2. Key highlights of PROTEUS: (1) fine-grained 1-D matrix tiling for efficient DCIM mapping, (2) programmable, hierarchical DCIM architecture, (3) on-chip NVM-resident micro-programs with a hierarchical DCIM-ISA, and (4) output-alignment floating-point (FP) DCIM for efficient FP computation.

compute macros [25], [26], are demonstrated for applications ranging from edge inference [27], [28], [29], [25] and fine-tuning [26], to associative one-shot learning [30] and event-driven tracking [24]. Meanwhile, advances in MRAM-CIM [31], [32], [33], [34] and PCM-CIM [35] broaden the landscape of back-end-of-line (BEOL) compatible CIM primitives. Despite these impressive macro-level and domain-specific system advances, many existing CIM designs are built upon coarse-grained execution abstractions constrained by rigid compute kernels and dataflows, with limited model mapping flexibility and numerical support. As a result, macro-level or even system-level efficiency gains often do not translate into application-level benefits, because real edge deployments (e.g., autonomous agents, AR/VR, mobile GenAI) could compose diverse AI model architectures with heterogeneous kernels and mixed precisions [36]. Recent near-memory/CIM processors therefore introduce ISA support by coupling CIM fabrics with embedded controllers (e.g., ARM/RISC-V/VLIW) and conventional or lightly extended ISAs [37], [38], [39], [40], [41], [42], [43], [44]. For example, [37] couples an ARM microcontroller to in-RRAM/near-SRAM compute macros tailored to a specific workload; Vecim [38] integrates CIM as a RISC-V vector extension effective for register-file–style vector kernels but not designed for general, fine-grained tensor placement and operations; MINOTAUR [39] and the VLIW edge accelerator in [41] rely on embedded CPU/VLIW control over near-memory matrix units (e.g., systolic or digital MAC arrays) under conventional 2-D matrix tiling and single data format; the CIM processor in [43], [44] supports both general programming and DNNs via a custom RISC-V stack, but largely alternates between CPU and CIM modes and retains limited CIM utilization. Across these state-of-the-art, processor-centric approaches, CIM resources are typically managed at coarse, macro-level granularity with conventional tiling, limited mixed-precision support, and no on-chip persistent micro-programs for kernel reuse. This leaves a gap for a fine-grained, CIM-native ISA that directly orchestrates data placement, movement, and flexible in-array compute in digital CIM (DCIM). This work addresses that gap between general-purpose programmability and in-memory compute efficiency and density by co-designing a programmable general-purpose DCIM fabric with a hierarchical CIM-native ISA.

To ground the co-design of a CIM-native ISA and efficient DCIM fabric, we revisit fundamental DCIM design challenges from a memory-centric perspective, as illustrated in Fig. 1. Viewed through this lens, existing DCIM and NVM-embedded accelerators for edge AI typically exhibit three limitations: (1) 1-D row-wise DCIM logic operations that under-utilize the inherently 2-D structure of memory arrays, (2) dependence on a multi-level memory hierarchy that re-introduces data

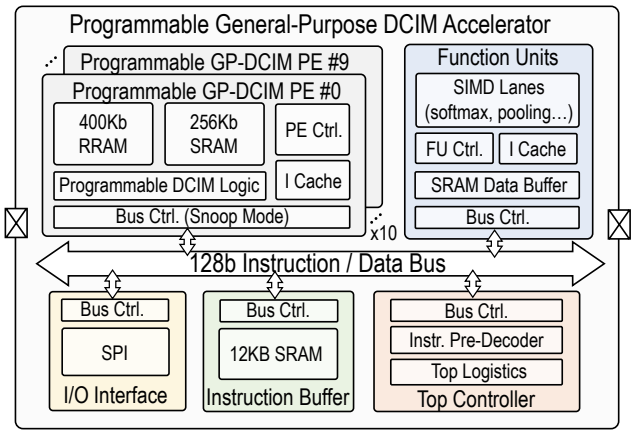


Fig. 3. PROTEUS chip-top architecture integrating ten programmable GP-DCIM PEs, functional units, an SPI-based I/O interface, an instruction buffer, and a top controller, all interconnected through a 128-bit instruction/data bus.

movement and undermines the compute-in-memory objective, and (3) capacity- and role-limited eNVM that under-utilizes non-volatility and limits the potential for multi-chip scaling of persistent models. These architectural challenges make it even harder to co-design a dedicated CIM-native ISA that simultaneously delivers programmability and efficiency, so current designs remain processor-centric, relying on embedded cores with conventional [37], [41] or modestly extended ISAs [38], [44].

Motivated by these interwoven memory- and ISA-level challenges, we develop PROTEUS, a programmable general-purpose DCIM (GP-DCIM) accelerator with heterogeneous RRAM-SRAM macros and a hierarchical CIM-native ISA. The key architectural and design features of PROTEUS are summarized in Fig. 2. First, conventional 2-D matrix tiling and mapping suffer from low array and PE utilization and costly partial-sum handling [13], [24]. In contrast, PROTEUS uses fine-grained 1-D matrix tiling enabled by the low-level DCIM-ISA to achieve near-100% memory utilization, eliminate partial sums, and provide flexible mapping and CIM resource assignment (Section II). Second, a hierarchical architecture together with the high-level DCIM-ISA flattens the memory hierarchy by removing reliance on large global buffers and a flat, complex datapath [7], [17], [25], and enhances programmability by coordinating local programmable datapaths inside PEs with a global programmable inter-PE data bus (Section II). Third, unlike non-volatile MCUs where instruction fetch from RRAM is a consequence of using RRAM as code memory [45], PROTEUS leverages RRAM non-volatility by embedding DCIM micro-programs alongside weights, enabling function-call-like reuse without repeated external instruction feeds; these micro-programs are invoked locally by the PE controller, reducing instruction cache accesses and off-chip communication. This extends eNVM in CIM from predominantly weight storage to local kernel control. These mechanisms collectively form a hierarchical DCIM-ISA spanning top-level micro-program calls, PE-level RRAM/SRAM dataflow scheduling, and low-level datapath control for fine-grained MAC operations using 1-D matrix tiling (Section III).

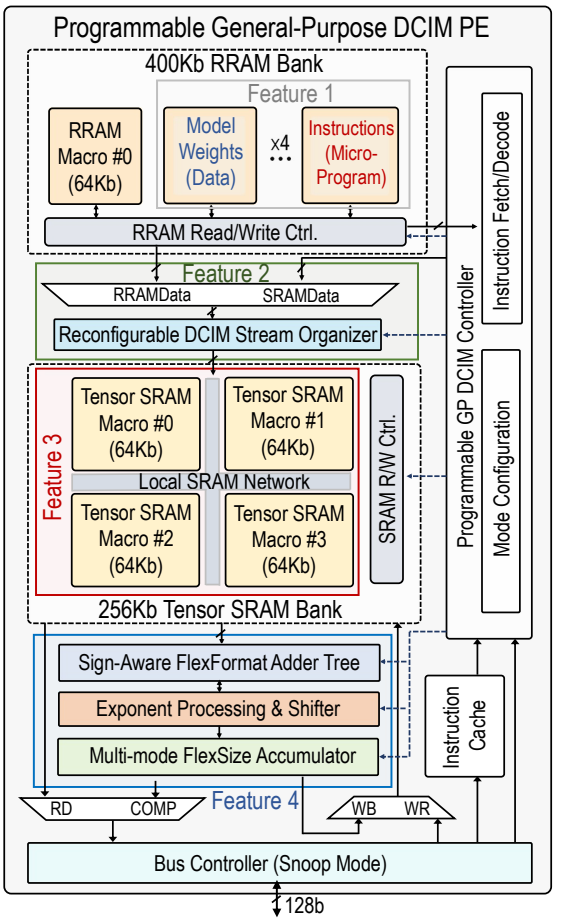


Fig. 4. Programmable GP-DCIM PE architecture with six 64-Kb RRAM macros, four 64-Kb tensor-SRAM macros, programmable local network, reconfigurable DCIM logic, instruction cache, bus controller, and top controller.

Finally, to address the limitations of prior FP-DCIM architectures [19], [28], including separated exponent–mantissa storage, congested dataflow from concurrent input and weight injection, and padding-induced alignment overhead, PROTEUS introduces an output-alignment FP-DCIM design that unifies FP data storage and eliminates padding bits (Section IV). This enables unidirectional, streamlined data injection and a unified dataflow across FP and INT computations, improving both efficiency and flexibility.

II. PROTEUS Architecture

A. Top Architecture

The top architecture of PROTEUS is illustrated in Fig. 3. The fully programmable general-purpose DCIM accelerator integrates ten GP-DCIM processing engines (PEs), dedicated function units (FUs), an SPI-based I/O interface, a 12 KB SRAM instruction buffer, and a global top controller. The top controller performs global scheduling across all modules and is responsible for instruction pre-decoding and distribution.

32-bit General-Purpose Tensor DCIM Instruction Set Architecture						
Memory Load / Store (RRAM LD, SRAM LD/ST)						
Op Code 5b	Src. Unit ID*1 4b	Src. RRAM ID 3b	Src. SRAM ID 2b	Dst. Unit ID 4b	Dst. SRAM ID 2b	
Intra-Engine Block Move (IBLKMOV)						
Op Code 5b	Unit ID 4b	Src. SRAM ID 2b	Src. Row 8b	Copy Len. 3b	Dst. SRAM ID 2b	Dst. Row 8b
Inter-Engine Block Move (EBLKMOV)						
Op Code*2 5b	Dst. Unit ID 4b	Src. SRAM ID 2b	Src. Row 8b	Copy Len. 3b	Dst. SRAM ID 2b	Dst. Row 8b
Digital Compute-In-Memory Tensor Multiply-Accumulate (TensorMAC)*3						
Op Code 5b	Src. PE ID 4b	Data Format 2b	Vector Len. 8b	Src. Mem (RRAM/SRAM) 4b	Src. Mem. Row 8b	
Kernel Size 6b	Dst. PE ID 4b	Src. Column 5b	Dst. Column 5b	Dst. SRAM ID 4b	Dst. SRAM Row 8b	
Function Unit Operation (FuncOp)						
Op Code 5b	FU ID 4b	Vector Len. 8b	Softmax Size 8b	Pooling Size. 3b	Data SRAM ID 4b	
SRAM Data Write Back (WBK)						
Op Code 5b	Src. PE ID 4b	Dst. PE ID 4b	Dst. SRAM ID 2b	Dst. Row 8b	Dst. Column 5b	AccFlag*4 4b
RRAM Micro-Program Load (MPLD)						
Op Code 5b	PE ID 4b	RRAM ID 3b	RRAM Row 8b	Micro-Program Len. 10b		

*1Unit refers to 10 PEs, Function Units, and SPI, total 12 on-chip components *2EBLKMOV op code includes 4 bit source Unit ID. *3TensorMAC op code merges two consecutive 32-bit instructions into one 64-bit computation instruction. *4AccFlag controls partial sum accumulation.

Fig. 5. 32-bit DCIM-ISA with seven instruction types (LD/ST, IBLKMOV, EBLKMOV, TensorMAC, FuncOp, WBK, and MPLD) enabling hierarchical programmability and flexible dataflows in PROTEUS.

A 128-bit snoop-mode instruction/data bus interconnects all subsystems, including the PEs, FUs, and memory interface, to enable simultaneous data broadcast, snoop-based sharing, and low-latency updates without external synchronization. Instruction sequences are first loaded into the instruction buffer, predecoded, and then dispatched by the top controller to the instruction caches within each PE and FU. The FUs complement the PEs by using their SIMD lanes to execute nonlinear and reduction operations such as softmax, activations, and pooling, and they can directly stream intermediate tensors to the global bus through the SRAM data buffer inside each FU.

B. Programmable Processing Engine

Within PROTEUS, each PE forms a self-contained DCIM computing node that integrates storage, compute logic, and data-routing capabilities, as shown in Fig. 4. A PE contains six 64-Kb RRAM macros and four 64-Kb tensor-SRAM macros, managed by a local PE controller and bus controller. The RRAM macros serve as dense, non-volatile memory banks that persistently store both model weights and DCIM micro-programs, while the tensor-SRAM macros function as computational memories, performing DCIM computations and buffering input, output, and intermediate data. During inference, no RRAM reprogramming is required. Instead, supporting additional kernels consumes proportionally more RRAM capacity for micro-programs, which represents an inherent trade-off of using non-volatility to enable rapid switching among pre-stored micro-programs without relying on off-chip instruction feeds. A programmable local compute/store network interconnects these memories, enabling high-speed, reconfigurable parallel data movement and cooperative computation among tensor-SRAM macros. This local network supports fine-grained 1-D matrix tiling, maximizing memory utilization and allowing flexible placement of tensor partitions across DCIM macros and PEs.

To coordinate multiple memory sources and destinations, each PE employs a reconfigurable DCIM data-stream organizer that dynamically schedules row-wise data movement between RRAM and SRAM macros. The stream organizer enables flexible injection of contiguous row segments into the programmable DCIM logic without requiring tensor reshaping. It supports two complementary DCIM operation modes: (1) a static mode, where data are streamed from RRAM into SRAM for stationary computation, and (2) a dynamic mode, where SRAM-to-SRAM transfers enable non-stationary computation such as attention, while supporting multiple data formats including INT8, INT16, FP8, and FP16. It is implemented as a lightweight control and data-path module that organizes input data, coordinates format-dependent injection, and ensures correct row-wise, bit-serial data delivery to the DCIM macros. By coordinating memory access and dataflow scheduling, it enables seamless switching between static and dynamic DCIM operations without additional buffering overhead. The programmable DCIM logic, composed of sign-aware flexible-format adders, exponent-processing and shifting modules, and a multi-mode accumulator, supports flexible tensor sizes and both row-wise and tensor-wise partial-sum accumulation. Computation results can be written back to any address within the local SRAM macros or directly to remote SRAM locations in other PEs via the interconnect bus. Each PE includes a 1-KB instruction cache for PE execution program and local micro-program invocation. By combining fine-grained dataflow control with a unified programmable DCIM fabric, each PE efficiently executes diverse workloads from convolution filters to attention kernels without external intervention.

The PE granularity needs to simultaneously consider three factors: (1) the persistent weight/micro-program capacity per PE, (2) the available tensor-SRAM resources for static RRAM-to-SRAM and dynamic SRAM-to-SRAM DCIM kernels, and (3) the pressure on the global interconnect when a layer is partitioned across multiple PEs. In PROTEUS, with 400-Kb RRAM bank and 256-Kb tensor-SRAM bank per PE, macro-level capacity trade-off is coarsely quantized, as our custom RRAM macro design offers about four times higher density than our custom tensor-SRAM. The nearest area-feasible alternatives would either reduce SRAM below what's efficient for SRAM-to-SRAM kernels and all buffering, or significantly reduce per-PE RRAM weight capacity and increase partitioning/communication overhead on the 128-bit bus. The hierarchical organization of PROTEUS incorporates local computing autonomy into each PE with global coordination over the 128-bit interconnect bus. This architecture minimizes long-distance data movement, enhances locality, and enables concurrent execution across all PEs and the FUs.

III. Hierarchical DCIM-ISA

The 32-bit DCIM-ISA defines a hierarchical, memory-centric programming interface for fine-grained control of both

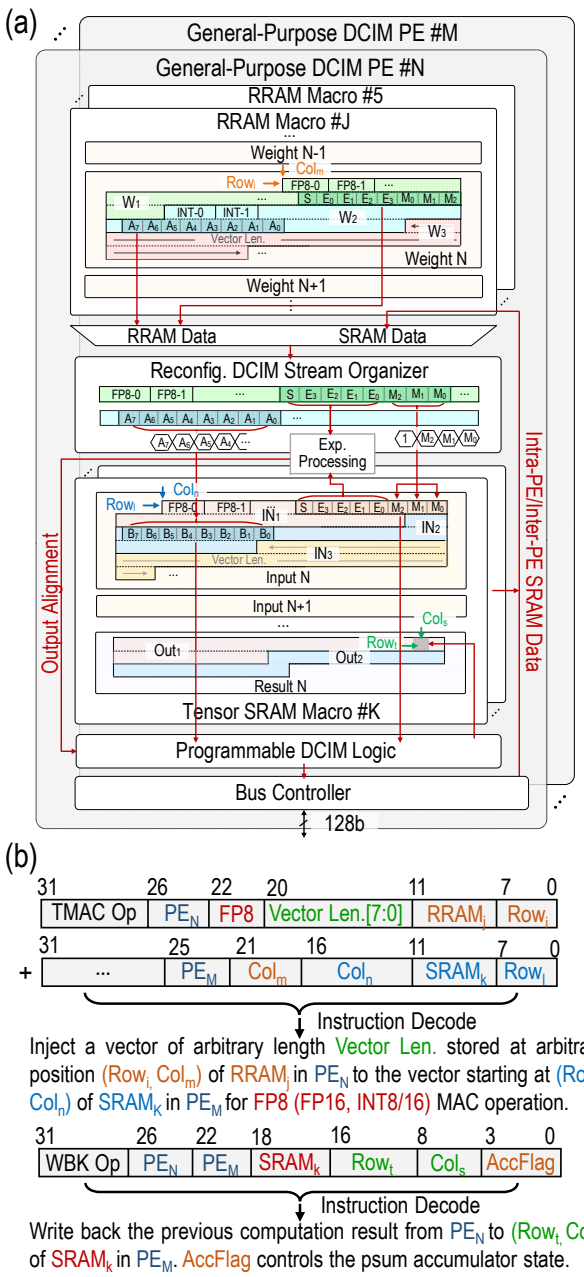


Fig. 6. (a) DCIM-ISA execution and dataflow with reconfigurable stream organizer and programmable logic coordinating RRAM/SRAM transfers, exponent alignment, and accumulation for multi-format DCIM operations both within and across PEs. (b) TensorMAC performs RRAM/SRAM-to-SRAM tensor MACs with arbitrary vector length and mixed-precision (FP8/FP16/INT8/INT16) support, while WBK writes back accumulated results to target SRAM under AccFlag control.

data movement and computation in PROTEUS. Unlike conventional accelerators with centralized control, the DCIM-ISA in PROTEUS embeds programmability directly into each PE and FU, enabling local execution and data reuse with minimal coordination overhead. Low-level instructions are grouped into micro-programs stored in RRAM and invoked through high-level calls, eliminating off-chip instruction streaming and allowing persistent, in-place kernel control.

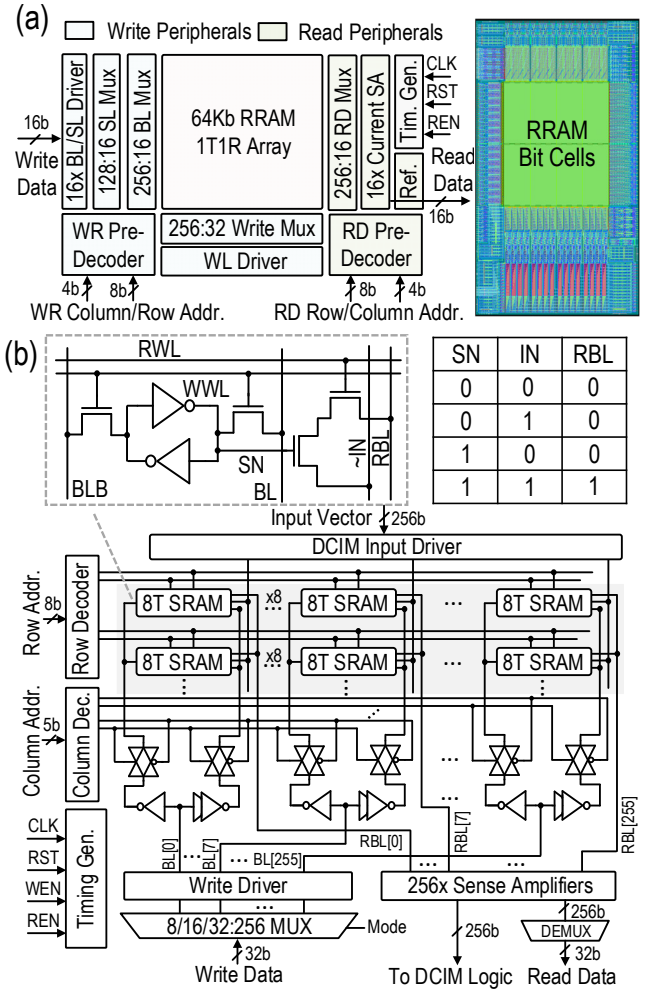


Fig. 7. (a) Custom RRAM macro schematic and layout based on foundry 1T1R cell arrays. (b) SRAM macro architecture using an 8T unit cell with in-cell multiplication and a unified structure that supports multi-mode memory access and DCIM streaming modes.

A. Instruction Set Architecture

There are seven major categories of instructions in the DCIM-ISA, as captured in Fig. 5: **(1)** Memory Load/Store (RRAM LD, SRAM LD/ST) moves complete data chunks between local memories and the interconnect, supporting both RRAM-to-SRAM and SRAM-to-SRAM transfers for initializing operands or writing results; **(2)** Intra-Engine Block Move (IBLKMOV) enables selective data relocation within a single PE, allowing sub-tensor copying or rearrangement across different SRAM rows to support 1-D tiling and dynamic weight reuse; **(3)** Inter-Engine Block Move (EBLKMOV) performs data exchange among PEs via the global 128-bit bus, supporting both broadcast and peer-to-peer transfers without host intervention; **(4)** Tensor Multiply-Accumulate (TensorMAC) executes tensor-level MAC operations on either RRAM-to-SRAM or SRAM-to-SRAM paths. Two consecutive 32-bit words are merged into a single 64-bit compute instruction, specifying source/destination PEs, vector length, data format (INT8/INT16/FP8/FP16), and kernel size. The AccFlag bits further control partial-sum accumulation across rows or vectors; **(5)** Function-Unit Operation (FuncOp) triggers nonlinear

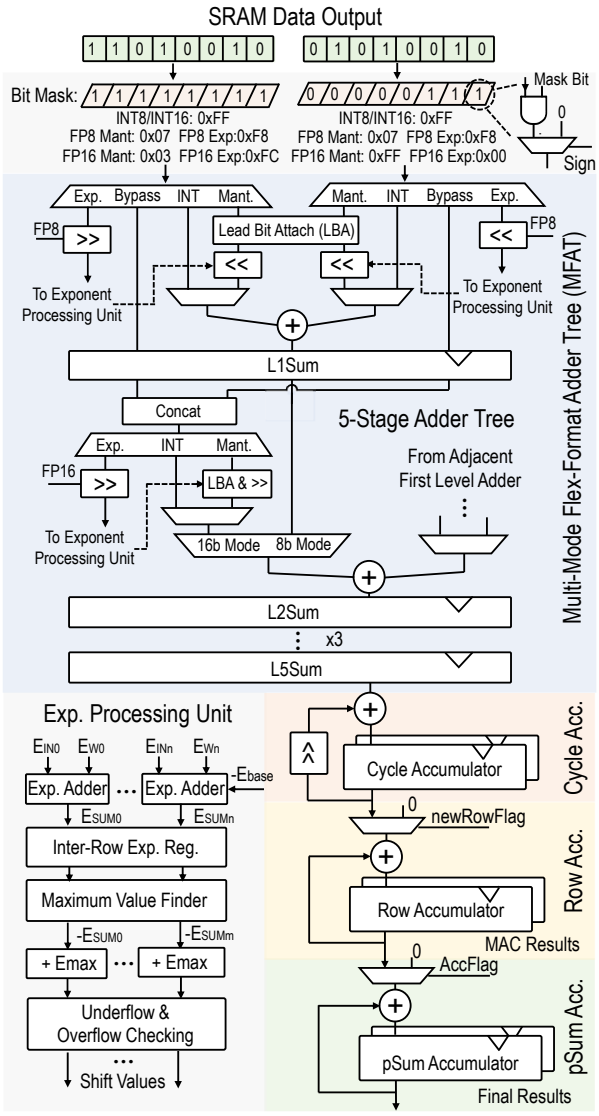


Fig. 8. Programmable DCIM logic architecture including a bit-mask unit, 5-stage Multi-Mode Flex-Format Adder Tree (MFAT), and exponent processing unit (EPU) for multi-format (INT8/INT16/FP8/FP16) operations. Hierarchical row/vector accumulators controlled by AccFlag and newRowFlag enable fine-grained programmable accumulation across tensor computations.

and reduction operations in the FU (e.g., softmax, pooling, normalization, activation, and element-wise arithmetic) through parameterized vector length, softmax/pooling size, and target SRAM address; **(6)** SRAM Write-Back (WBK) writes computation results from local DCIM logic or FU outputs back to target SRAM addresses within the same PE or to global SRAM locations in other PEs through the interconnect; **(7)** Micro-Program Load (MPLD) directly reads and executes micro-programs stored in RRAM, enabling programmable sequence definition and rapid switching among pre-stored kernels without explicit instruction fetching or reloading.

B. Execution and Dataflow

Fig. 6 illustrates the DCIM-ISA execution and dataflow. The reconfigurable stream organizer and programmable DCIM logic coordinate RRAM/SRAM transfers, exponent alignment,

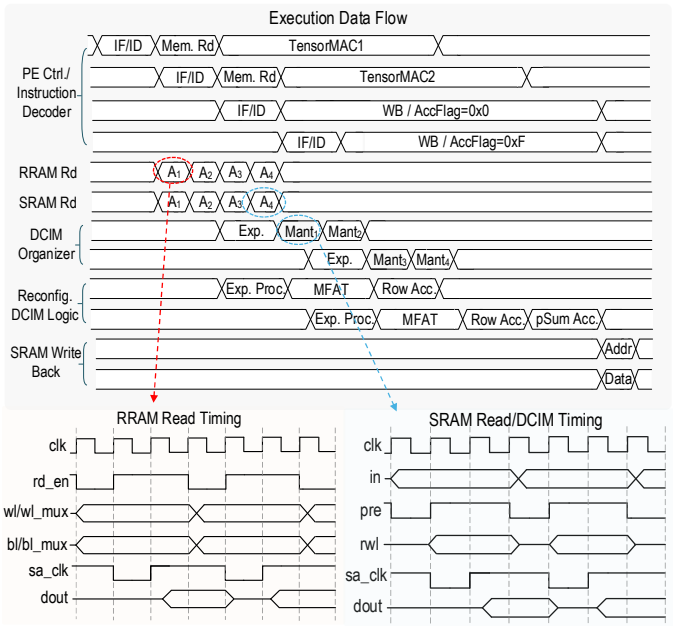


Fig. 9. GP-DCIM execution flow illustrating instruction decoding, memory load, DCIM data injection, exponent processing, MFAT computation with hierarchical accumulation, and SRAM write-back, with cycle-accurate timing of RRAM read and SRAM read/DCIM operations.

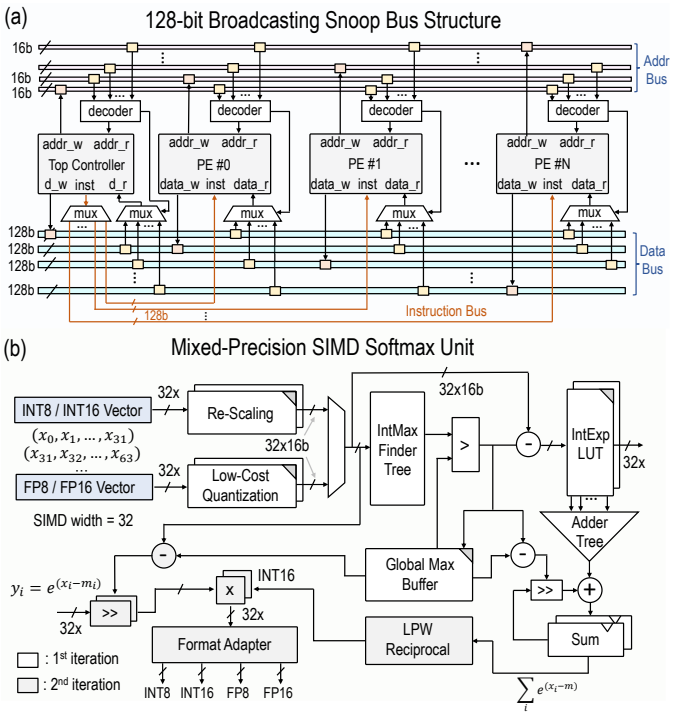


Fig. 10. (a) 128-bit broadcasting snoop bus composed of address, data, and instruction channels. The physically parallel multi-lane implementation enables arbitration-free concurrent transfers from multiple sources, providing effective bandwidth exceeding 128 bits per cycle. (b) Mixed-precision SIMD softmax unit supporting INT8/INT16/FP8/FP16 while maintaining high parallelism.

and accumulation for multi-format (FP8/FP16/INT8/INT16) DCIM operations. TensorMAC executes mixed-precision vector MACs, while WBK writes back accumulated results under AccFlag control, completing a unified in-place execution

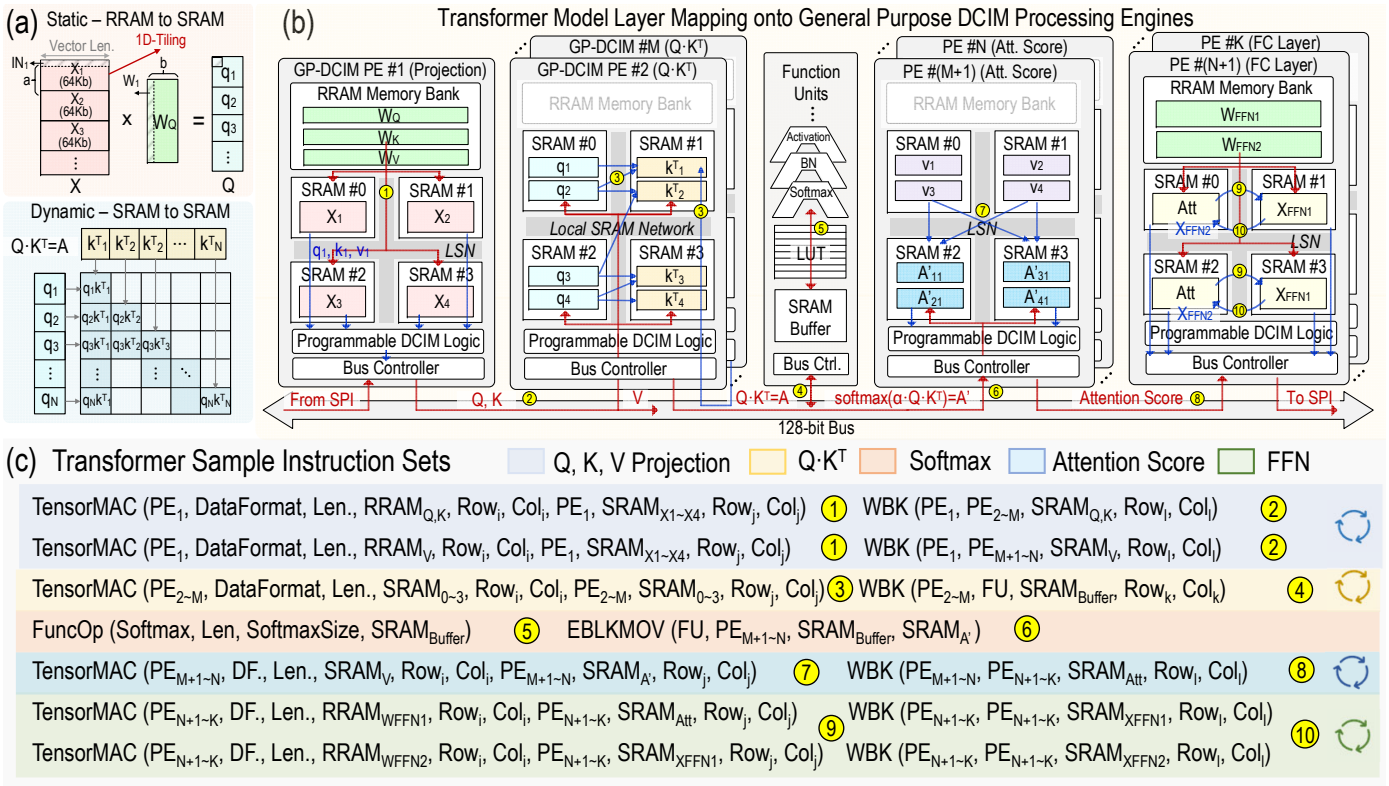


Fig. 11. Transformer model mapping on PROTEUS programmable GP-DCIM processing engines. (a) Task allocation of projection and attention operations onto the RRAM/SRAM dataflow with fine-grained 1-D matrix tiling and arbitrary attention sparsity. (b) On-chip dataflow within and across PEs and functional units for end-to-end Transformer execution. (c) Example instruction sequences implementing Q, K, V projection, $Q{\cdot}K^T$, attention score, softmax, and FFN.

pipeline across PEs. Each instruction can be dispatched by the top controller or invoked locally within a PE during micro-program execution, achieving hierarchical control between global scheduling and local autonomy. Through this CIM-native ISA, PROTEUS unifies programmability and in-memory compute efficiency, supporting diverse workloads such as convolution, matrix multiplication, attention, and graph aggregation within a single dataflow fabric.

IV. Circuit Implementation

PROTEUS's memory system integrates foundry 1T1R RRAM arrays [46] with our custom peripheral circuits, and 8T tensor-SRAM macros with in-cell multiplication and a unified multi-mode architecture, as shown in Fig. 7. Each RRAM macro serves as a dense, non-volatile storage bank, connected to peripheral write drivers, sense amplifiers, and data-path multiplexers for programmable access. Together with the tensor-SRAM banks, these arrays form a tightly coupled hybrid memory fabric that supports both persistent weight storage and dynamic computation. The tensor-SRAM design enables seamless switching between conventional memory access and DCIM streaming, allowing the same array to function as both data storage and a compute fabric, thereby maximizing density, on-chip data locality, and reuse efficiency.

As shown in Fig. 8, each programmable DCIM engine consists of a bit-mask unit, a five-stage Multi-Mode Flex-Format Adder Tree (MFAT), and an Exponent Processing Unit (EPU) supporting INT8, INT16, FP8, and FP16 operations. The MFAT performs hierarchical accumulation across bit, row, and vector stages, while the EPU handles the output exponent alignment and normalization. Together, they realize the precision-adaptive, flex-format, output-alignment-based DCIM architecture with a unidirectional streaming dataflow (Fig. 2, Highlight 4). In conventional input-alignment designs, weight pre-alignment and padding to accommodate alignment-induced bit growth may require early mantissa truncation or rounding prior to accumulation, potentially introducing precision degradation beyond the intrinsic finite-precision behavior of the target FP format. By performing exponent alignment only once within the exponent-processing unit before accumulation, the proposed output-alignment avoids such additional truncation stages while preserving standard floating-point numerical semantics. Fine-grained programmable accumulation is achieved through hierarchical row/vector accumulators governed by AccFlag and newRowFlag, enabling precision-aware aggregation across tensor computations.

Based on the DCIM-ISA and the circuit implementations discussed above, Fig. 9 illustrates a representative GP-DCIM execution sequence, using the FP mode as an example since it activates the most complete datapath. The operation starts with instruction decoding and memory loading, followed by DCIM-organizer data injection, exponent alignment, and MFAT computation. In FP mode, both exponent processing and mantissa accumulation are enabled within the programmable DCIM logic, after which row-wise and vector-wise partial-sum accumulations proceed in a pipelined manner. The final

results are written back through the WBK instruction. Two TensorMAC operations and their corresponding WBK write-backs are co-issued from the instruction cache with data-dependency checking to hide decoding latency and maintain full pipeline utilization. To complement the execution flow, cycle-accurate timing of RRAM read and SRAM read/DCIM operations is also shown in Fig. 9, highlighting their multi-cycle behavior.

Fig. 10 illustrates PROTEUS's custom bus and softmax unit design. The snoop bus is logically organized as a 128-bit interface; however, it is physically implemented using parallel multi-lane broadcasting links, allowing multiple inter-PE transfers to occur concurrently. This design significantly alleviates communication contention without requiring a bus arbitrator, making it well matched to the high-throughput DCIM PEs. Meanwhile, the mixed-precision SIMD softmax unit further improves system parallelism by processing a 32-element vector in a batch. Its two-iteration streaming scheme reduces the need for large intermediate buffers, allowing more area to be allocated to computation resources while maintaining high throughput.

V. Transformer and CNN Model Deployment

The programmability and workload versatility of PROTEUS are demonstrated through the mapping and execution of Transformer and CNN models across its ten GP-DCIM PEs. A detailed mapping scheme for Transformer models is provided in Fig. 11. The PEs are dynamically grouped to execute projection and attention submodules. Q, K, V projections and feed-forward (FFN) layers are performed through RRAM-to-SRAM dataflow using fine-grained 1-D matrix tiling, while the $Q \cdot K^T$ and attention score computation are carried out through SRAM-to-SRAM streaming across multiple PEs. The FU handles nonlinear and reduction operations such as softmax and normalization, directly streaming intermediate tensors to the global interconnect for subsequent stages. The sample instruction sets illustrate how the hierarchical DCIM-ISA orchestrates computation, enabling arbitrary attention sparsity and flexible vector lengths. The evaluated Transformer workloads in this work are encoder-style (BERT and MobileViT), where the K/V tensors are computed once per layer and reused within that layer.

Switching from Transformer to CNN workloads, as shown in Fig. 12, the new model execution begins with the MPLD instruction, which signals the PE controller to load and execute the convolution micro-program stored in RRAM. Convolution kernels are then injected from RRAM into SRAM-resident feature pixels at arbitrary locations and variable lengths, eliminating the redundancy of conventional unrolling. Although the illustrated mapping emphasizes direct RRAM-to-SRAM streaming enabled by 1-D matrix tiling, weights can also be preloaded into tensor-SRAM through RRAM LD instructions to enable weight-stationary execution when higher temporal reuse is desired. Fine-grained ISA-controlled data injection and the local stream organizer ensure precise kernel placement, maximizing reuse and locality during both convolution and pooling. The representative instruction sequence demonstrates

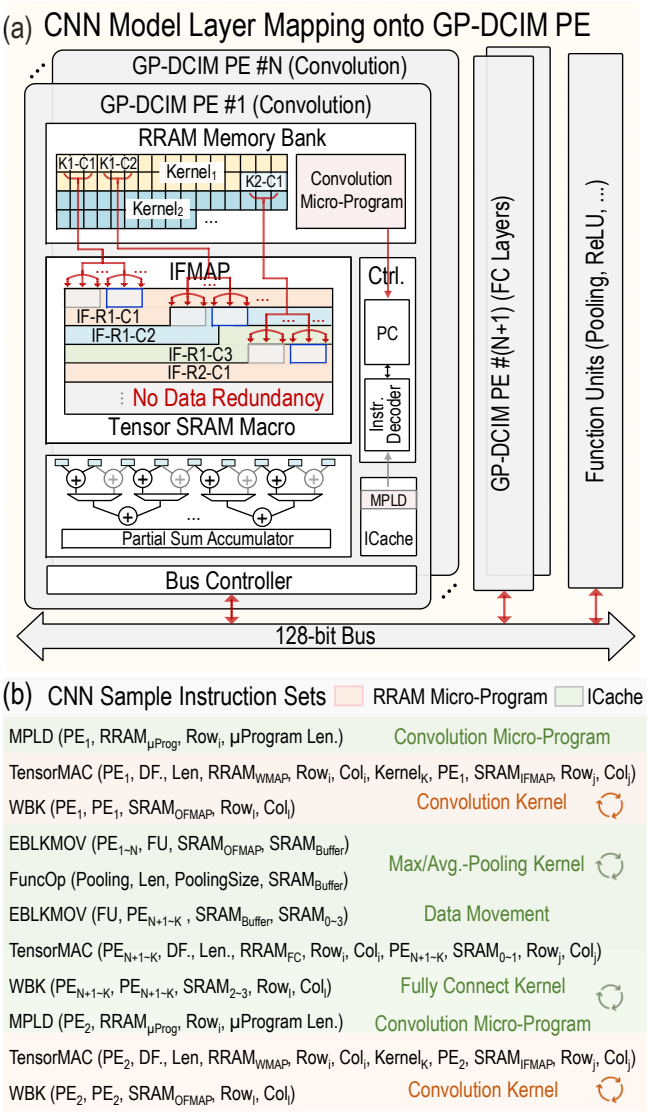


Fig. 12. CNN model mapping onto PROTEUS programmable GP-DCIM processing engines. (a) Convolution execution flow, where the MPLD instruction signals the controller to load the convolution micro-program from RRAM. Kernels are injected from RRAM into input feature pixels in SRAM without unrolling, enabled by fine-grained, ISA-controlled arbitrary-position and variable-length data injection. (b) Sample instruction sequences implementing MPLD, TensorMAC, WBK, EBLKMOV, and FuncOp for convolution, pooling, and fully connected layers.

programmable coordination of MPLD, TensorMAC, WBK, EBLKMOV, and FuncOp instructions for end-to-end CNN execution including convolution, pooling, and fully connected layers.

Fig. 13 illustrates PROTEUS's custom compiler framework. As shown in Fig. 13(a), the compiler converts a high-level AI model description (layer graph, tensor shapes, and operator types) into a full task directed acyclic graph (DAG), where nodes represent per-layer/per-tile kernels and edges capture both dataflow and execution dependencies. Based on the target hardware resources (number of chips/PEs and per-PE RRAM/SRAM capacity) and optimization objectives (maximize PE utilization and locality while minimizing latency), the compiler performs an early deployment estimation to

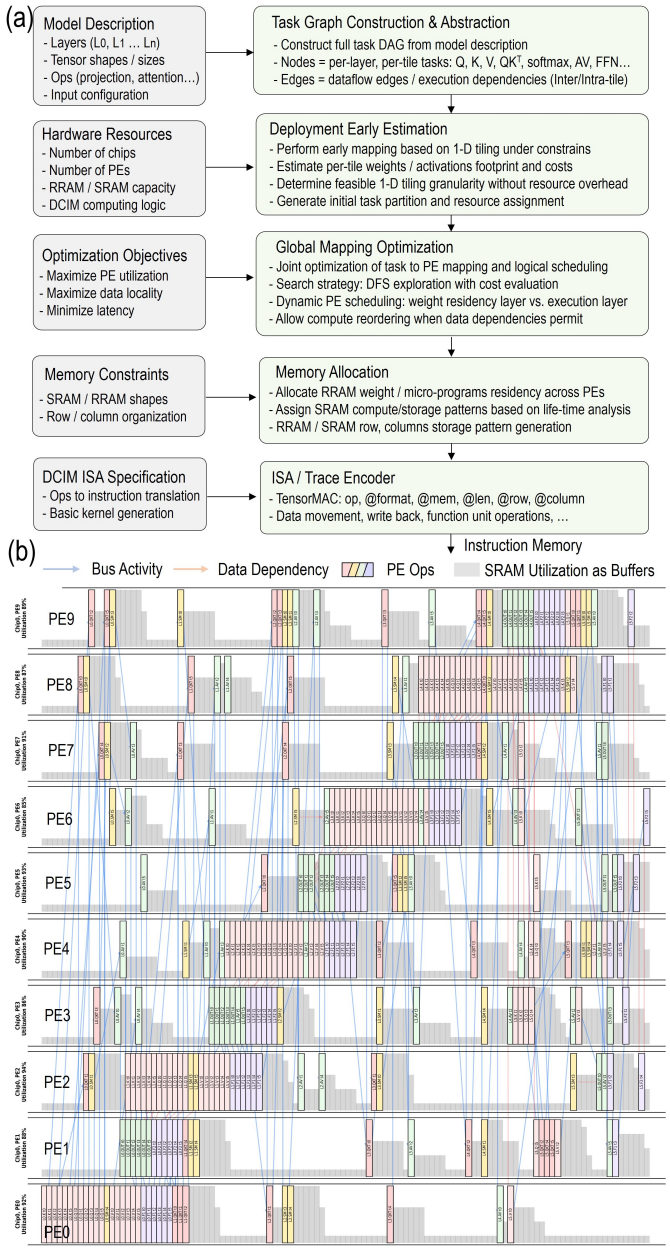


Fig. 13. PROTEUS's custom compiler framework. (a) End-to-end compilation flow that converts an AI model into executable DCIM instruction traces. (b) Example mapping of a MobileViT transformer layer onto a multi-PE execution timeline, showing dynamic scheduling, inter-PE data transfers, and high resource utilization.

determine feasible 1-D tiling granularity and to estimate per-tile weight/activation footprints. It then conducts global mapping optimization to jointly decide task-to-PE assignment and logical scheduling under dependency constraints, followed by memory allocation to generate RRAM/SRAM data placement and row/column placement patterns. Finally, an ISA/trace encoder emits executable hierarchical instruction traces (e.g., TensorMAC, EBLKMOV, WBK, and FuncOp) that directly control compute and data movement. Fig. 13(b) provides a compiled example of one MobileViT transformer layer mapped onto a multi-PE execution timeline. The visualization highlights dynamic task scheduling across PEs, bus data transfer activity (blue links), data dependencies (orange links), and SRAM utilization as buffers (gray background), demonstrating how the compiler orchestrates computation, buffering, and inter-PE communication to sustain high resource utilization.

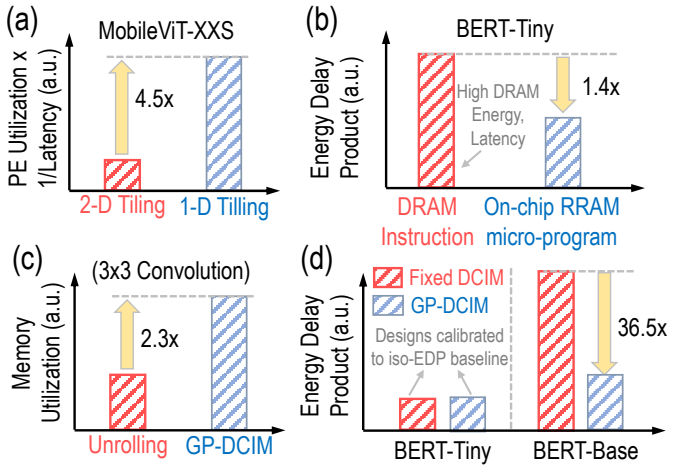


Fig. 14. Performance comparison between PROTEUS and conventional baselines. (a) Improved PE utilization and reduced latency enabled by fine-grained 1-D tiling compared with conventional 2-D tiling (b) Energy–delay product (EDP) reduction enabled by storing micro-program instructions in on-chip RRAM instead of DRAM. (c) Memory utilization improvement for convolution execution compared with conventional unrolling. (d) EDP comparison between GP-DCIM and fixed-datapath DCIM designs across BERT-Tiny and BERT-Base workloads.

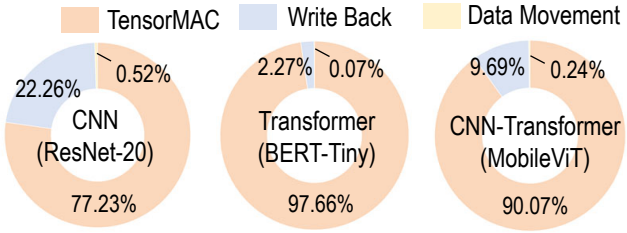


Fig. 15. Instruction distribution for ResNet-20, BERT-Tiny, and MobileViT execution on PROTEUS, showing TensorMAC dominance followed by WBK and data movement instructions.

With the flexibility of model deployment on PROTEUS, we evaluate the efficiency and scalability of the proposed GP-DCIM architecture against conventional baselines. As shown in Fig. 14(a), fine-grained 1-D tiling improves PE utilization and reduces latency by 4.5× compared with conventional 2-D tiling on MobileViT-XXS. Fig. 14(b) shows that storing micro-program instructions in on-chip RRAM instead of DRAM reduces instruction-access energy and latency, achieving a 1.4× EDP improvement on BERT-Tiny. In addition, Fig. 14(c) demonstrates a 2.3× improvement in memory utilization for convolution execution compared with conventional unrolling. We then compare PROTEUS with a conventional fixed-datapath DCIM baseline under an iso-energy-delay-product (iso-EDP) constraint using the BERT-Tiny workload. The baseline represents a flattened, model-specific DCIM architecture with a fixed execution datapath and predetermined dataflow, designed for a single target model without programmability. In contrast, PROTEUS employs a

programmable hierarchical GP-DCIM architecture supporting ISA-based control and workload-adaptive scheduling. The operating point of both designs is aligned at iso-EDP on BERT-Tiny to ensure a fair comparison of architectural efficiency. The same configuration is then applied to the larger BERT-Base model to evaluate scalability trends. As shown in Fig. 14(d), PROTEUS achieves a 36.5× EDP improvement over the fixed-datapath DCIM baseline, highlighting the context-switching advantage enabled by programmability.

Finally, instruction-level profiling reveals the execution breakdown across representative workloads—ResNet-20, BERT-Tiny, and MobileViT-XXS, as summarized in Fig. 15. TensorMAC operations dominate across all models, followed by WBK and data-movement instructions. The relatively larger WBK proportion in CNNs reflects the repeated accumulation of intermediate activations, whereas the reduced WBK ratio in Transformers indicates more efficient in-place accumulation within the GP-DCIM array, achieving higher utilization and throughput across heterogeneous AI tasks.

VI. Measurements and Results

A. Test Setup and Chip Summary

The die photo (Fig. 16) and a summary of chip specs (Table I) are provided. Within 10 programmable GP-DCIM PEs in an 18 mm^2 die in 40 nm, 2.6 Mb tensor SRAM and 4.0 Mb RRAM are integrated along with a 128-bit bus. The chip operates across a clock frequency range of 50–275 MHz, consumes 21–248 mW, and achieves a peak energy efficiency of 2.8–6.4 TOPS/W with 8-bit workloads. Fig. 17 shows the chip measurement setup, which employs an array of nine PROTEUS chips interconnected for distributed computation in a multi-chip system, coordinated by a Xilinx Zynq UltraScale+ MPSoC host board. Multi-chip communication with FPGA is supported through SPI and AXI interfaces. SPI is chosen for multi-chip prototyping and validation purpose, while the key features of PROTEUS are compatible with high-bandwidth chip-to-chip links and chiplet fabrics.

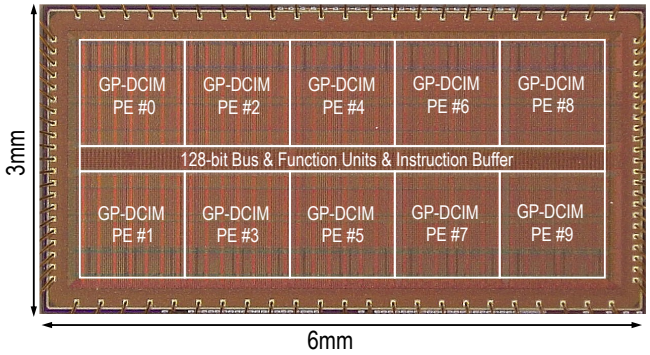


Fig. 16. PROTEUS die photo. The micrograph highlights ten programmable GP-DCIM engines, functional units, and the 128-bit interconnect bus.

Under the multi-chip configuration, model partitioning is performed at the layer or subgraph level according to per-chip RRAM and tensor-SRAM capacity. Each chip is statically assigned a disjoint portion of the neural network, while fine-grained 1-D tiling and PE grouping are applied locally within each chip as described in Section V. Inter-chip communication occurs only at subgraph boundaries, where intermediate feature maps are transferred through SPI and buffered locally before execution resumes. The FPGA host orchestrates synchronization and data movement across chips, while each PROTEUS chip executes its assigned subgraph autonomously using its hierarchical DCIM-ISA.

In PROTEUS, RRAM is written only during model deployment (including writing micro-programs) and is read-only during inference, taking into account the reliability considera-

TABLE I
Summary of PROTEUS Chip Specs

Chip Size	6 mm × 3 mm
Technology	40 nm ULP CMOS + Embedded RRAM
Package	QFN100
RRAM Capacity	4.0 Mb
Tensor SRAM Capacity	2.6 Mb
Data Bus Width	128 bit
Communication	High-Speed SPI
Data Format Support	INT8, INT16, FP8, FP16
Application	Versatile Edge AI (CNN, Transformer, CNN-Transformer, GNN, SSM)
Supply Voltage	0.75–1.20 V
RRAM Program Voltage	1.5–3.3 V (3.3 V for Forming)
RRAM Read Voltage	0.2 V
Clock Frequency	50–275 MHz
Power Consumption[*1]	21–248 mW
Peak Efficiency @ 8b[*2]	2.8–6.4 TOPS/W

[*1]Measured at 50% input toggle rate @ INT8. [*2]1 MAC = 2 OP.

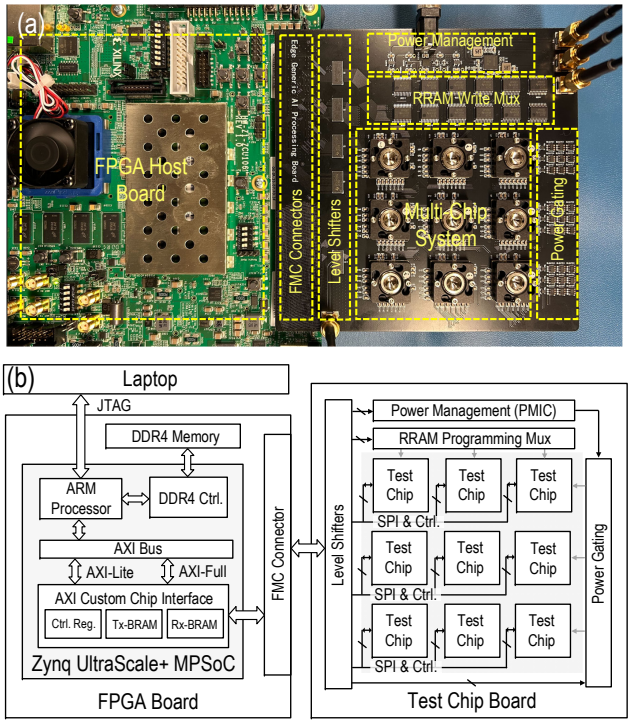


Fig. 17. PROTEUS multi-chip edge-AI processing system setup. The photo (a) shows the test-chip board and FPGA host board, while the schematic (b) illustrates multi-chip communication with FPGA through SPI and AXI interfaces for data transfer and control.

TABLE II
MEASURED INFERENCE RESULTS ACROSS MODELS AND PRECISIONS

Edge AI Model	ResNet-20	BERT-Tiny	MobileViT-XXS	GraphSAGE	Vision-Mamba
Model Family	CNN	Transformer	CNN-Transformer	GNN	SSM
Task	CIFAR-10	SST-2	CIFAR-100	CORA	CIFAR-10
Model Size (Mb)[*1]	2.16	35.09	7.86	3.20	50.47
INT8 Inference Accuracy (%)	85.36% (85.54%[*2])	80.34% (80.99%[*2])	68.78% (70.25%[*2])	30.82% (30.82%[*3])	79.29% (79.32%[*2])
INT16 Inference Accuracy (%)	82.21% (84.97%[*2])	80.73% (82.41%[*2])	72.80% (73.38%[*2])	30.82% (30.82%[*3])	80.73% (82.41%[*2])
FP8 Inference Accuracy (%)	91.35% (91.41%[*2])	81.64% (81.71%[*2])	78.74% (80.71%[*2])	32.15% (32.15%[*3])	88.86% (89.22%[*2])
FP16 Inference Accuracy (%)	88.93% (90.47%[*2])	81.11% (81.38%[*2])	94.09% (94.85%[*2])	86.47% (86.92%[*3])	86.79% (89.71%[*2])
Number of Instructions	18K	22K	16K	72K	36K
End-to-End Latency (ms)[*4]	41	311	220	1538	458
Workload Energy Efficiency[*5]	25.93 GOPS/W	16.32 GOPS/W	35.22 GOPS/W	13.35 GOPS/W	22.54 GOPS/W

[*1]Parameters are in 8-bit; multi-chip scenario used when model exceeds single-die RRAM capacity; [*2]Software baseline accuracy from evaluated precision and model; [*3]F1 score is reported instead of accuracy for GNN workload; [*4]End-to-end latency includes system-level SPI I/O and FPGA processing and control; [*5]8b precision. Workload efficiency includes all compute and non-compute periods, including SPI I/O phases.

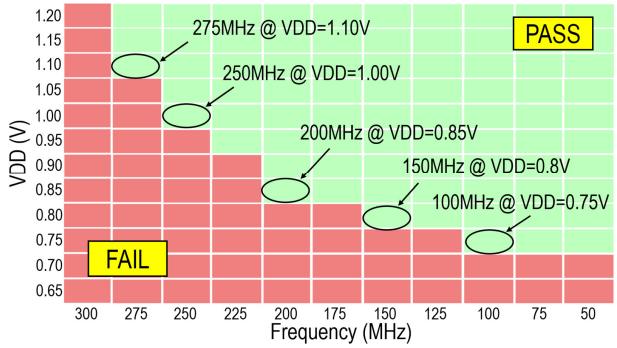


Fig. 18. Measured frequency versus voltage operating conditions in the shmoo plot. The frontier indicates the best performance/efficiency for PROTEUS.

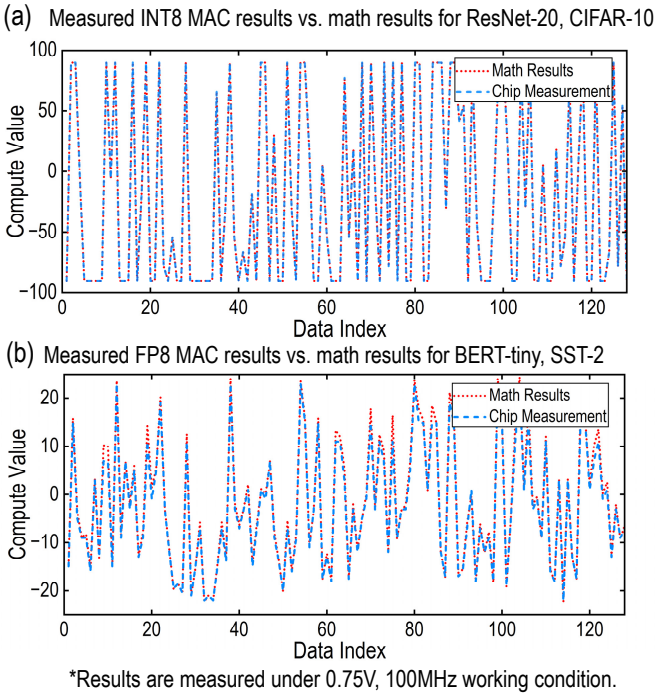


Fig. 19. Integer/floating-point MAC accuracy verification with chip measurements running (a) ResNet-20 with CIFAR-10 and (b) BERT with SST-2.

tions on retention, read disturb, and variability. Our operating conditions align with the reliability targets demonstrated in foundry RRAM characterizations to ensure high data retention, with read disturb immunity under 200 mV read condition [46], [47], [48]. The measured RRAM macro access latency is 10 ns with an energy of 0.2 pJ/bit, benefiting from low-voltage sensing in our custom RRAM macro design. To mitigate variability-induced errors, we use the standard write-verify programming to ensure well-separated binary bit (SLC) distributions in the RRAM micro-programs, which is more tolerant than multi-bit-per-cell (MLC) programming [27]. Combining SLC-RRAM for micro-programs and MLC-RRAM for model weights could further enhance system capability.

B. Silicon Results with Versatile Edge AI Models

First, we test the chip operating conditions. As captured in Fig. 18, the shmoo plot shows the operational range of voltage and frequency, forming a Pareto frontier that demonstrates reliable operations from 0.75 V at 50 MHz up to 1.2 V at 275 MHz. Next, we conduct two sets of measurements for accuracy verification of the multiply-accumulate (MAC) results. With different models (Transformer, CNN) and different data format (INT, FP) as examples, the measured MAC results during real on-chip inference runs (at 0.75 V, 100 MHz) are directly compared against the ground-truth mathematical results from the bit-accurate software reference. As shown in Fig. 19(a), on-chip INT8 computations are validated running ResNet-20 model on the CIFAR-10 dataset. Similarly, computational correctness of FP8 is verified with the BERT-Tiny model on the SST-2 dataset (sentiment classification task), as shown in Fig. 19(b). Since the fully-digital output-alignment FP-DCIM uses a deterministic EPU for output exponent alignment and normalization with overflow/underflow checking, no extra approximation is introduced beyond the inherent finite-precision behavior of the FP8 format.

TABLE III
COMPARISON WITH STATE-OF-THE-ART PROGRAMMABLE IN-MEMORY AND NEAR-MEMORY ACCELERATORS

Reference	ISSCC'23 [37]	ISSCC'24 [38]	VLSI'24 [39]	ISSCC'24 [41]	JSSC'25 [44]	**This Work**
Technology Node	40 nm	65 nm	40 nm	40 nm	65 nm	**40 nm**
Die Area (mm^2)	20.25	4.0	65.6	20.25	1.52	**18.0**
Memory Capacity (Mb)[*1]	20	0.13	96	46	0.06	**6.6**
Application	Hybrid CNN + SNN	HPC Kernels	Transformer, CNN	NN Perception + Localization	General Programming or DNN	**Versatile Edge AI (Transformer/CNN/GNN/SSM)**
Acceleration Unit	Analog CIM	Digital CIM	Systolic Array	Digital MAC	Digital CIM	**Programmable DCIM**
Precision Support	INT8	INT8, FP16	Posit8	INT8	INT8, INT32	**INT8, INT16, FP8, FP16**
Instr. Set Architecture	ARM	RISC-V Extension	RISC-V	128b VLIW	Custom RISC-V	**32b DCIM-Native ISA**
Workload Mapping	2-D Tiling	2-D Tiling	2-D Tiling	2-D Tiling	2-D Tiling	**1-D Matrix Tiling**
On-Chip Instr. Storage	No	No	No	No	No	**Micro-Programs in eNVM**
Programmable Execution Granularity	Microcontroller	RISC-V/DCIM Co-Processor	Embedded CPU	VLIW-Controlled Matrix Units	CPU-or-DCIM (Exclusive Modes)	**Fine-Grained General-Purpose DCIM**
Memory Utilization	Low	High[*2]	Low	Limited	Limited[*3]	**High**
Peak Throughput (GOPS) @ 8b	230.3[*4]	31.8	5.14	268.8	64.0	**702.0**
Energy Efficiency (TOPS/W) @ 8b	1.20[*4]	0.28	0.50	0.84	7.62	**6.40**
Compute Density (GOPS/mm^2) @ 8b	11.40[*4]	7.95	0.78	13.30	42.11	**39.00**

[*1]Memory capacity includes all on-chip memories (SRAM only or SRAM + eNVM); [*2]Benefiting from small-size CIM vector register file; [*3]High utilization when used as CPU memory or buffer, but low utilization as CIM with 2-D tiling. [*4]1 × 8b TOPS = 64 × 1b TOPS; data from [41].

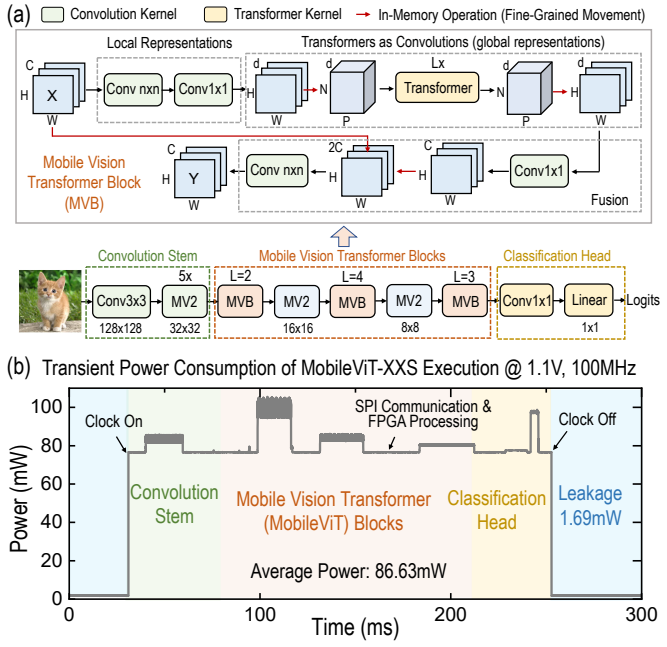


Fig. 20. (a) Mobile-ViT model architecture and its hardware mapping. (b) Measured transient power profile during MobileViT execution.

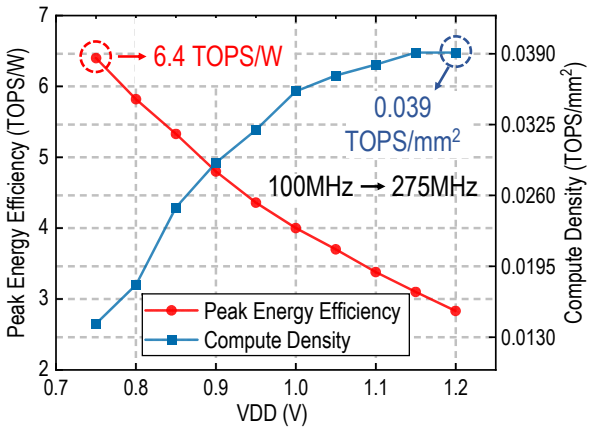


Fig. 21. Measured energy efficiency and compute density across Pareto front.

With the established functional correctness, we then show how to exploit the programmability and versatility of PROTEUS for edge AI workloads with inherent heterogeneity. We perform real-time workload characterization on PROTEUS using MobileViT-XXS as a testbed. MobileViT is a lightweight, general-purpose vision Transformer model for mobile devices that merges CNNs' efficiency and inductive biases with Transformer's capability of capturing global context [49]. The heterogeneous model requires multiple layer types, including convolution kernels, mobile vision Transformer blocks (MVB) with attention kernels, and classification heads (Fig. 20(a)). Running MobileViT-XXS testbed on PROTEUS, the measurement captures a transient power profile that visualizes the layer execution, kernel switching, and layer transition from clock-on to clock-off, as shown in Fig. 20(b). The measured power profile provides a breakdown across feature extraction, MVB, and classification, with an average power of 86.63 mW obtained, supporting low-power edge deployment.

Owing to the programmable GP-DCIM architecture, hierarchical DCIM-ISA, and persistent micro-programs in RRAM, PROTEUS supports versatile edge AI models beyond CNNs and Transformers. We conduct extensive measurements across diverse models covering different model families. At the workload level, measured results in terms of inference accuracy, latency, and energy efficiency of PROTEUS are reported in Table II. Here, the end-to-end latency is defined and measured at the system level from the host issuing an inference request to receiving the final output, and thus includes the SPI overhead as well as FPGA-assisted processing and control between layer executions.

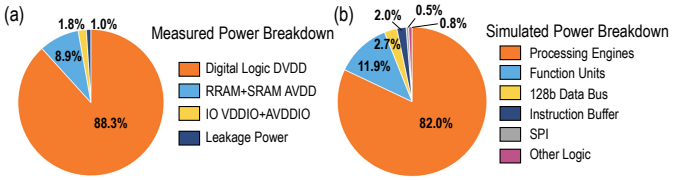


Fig. 22. Measured (a) and simulated (b) power breakdown of PROTEUS.

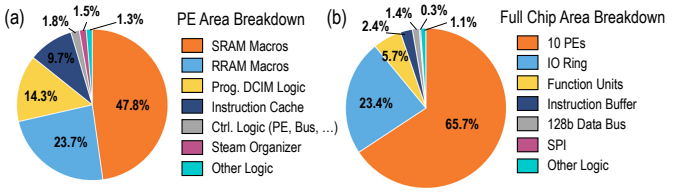


Fig. 23. Area breakdown of (a) a single PE, and (b) the full chip.

The workload energy efficiency is computed from the integrated energy at the PROTEUS chip supply rails and therefore includes compute, on-chip data movement, and I/O drivers during both compute and non-compute phases. Importantly, because both model weights and DCIM micro-programs reside in on-chip RRAM (partitioned across chips when the model exceeds a single die), PROTEUS does not fetch weights from external DRAM during continuous inference; therefore, the remaining off-chip traffic is mainly input/output activations rather than weight re-loading, unlike designs that focus on explicitly mitigating external-weight traffic [50], [51]. Since weight transfers typically dominate external memory access (EMA) in such systems, the remaining activation-related access becomes a secondary factor and is highly dependent on system-level assumptions. Hence, consistent with the EMA evaluation methodology in [52], we exclude the secondary activation-related EMA from system-level metrics. The workload-level energy efficiency is diluted as expected due to the SPI, multi-chip, and idle overheads, while the underlying GP-DCIM utilization is still kept high owing to the developed 1-D matrix tiling. The demonstration spans diverse models, including ResNet-20, BERT-Tiny [53], MobileViT-XXS [49], GraphSAGE [54], and Vision Mamba [55]. These representative models cover a wide spectrum of model families, from CNN, Transformer, hybrid CNN-Transformer, to Graph Neural Network (GNN) and State-Space Model (SSM).

C. Comparison with Prior Works

We first extract the measured energy efficiency and compute density of PROTEUS and provide a breakdown for power and area. The measured results shown in Fig. 21 highlight the trade-off between energy efficiency and compute density across Pareto front, revealing a peak energy efficiency of 6.4 TOPS/W and compute density of 0.039 TOPS/mm^2. Power and area breakdowns show that computation dominates total power while memory macros occupy the largest share of chip area. In the measured breakdown, digital logic accounts for 88.3% of total power, followed by RRAM+SRAM at 8.9%, as shown in Fig. 22. The area breakdown captured in Fig. 23 shows the memory-centric nature of the design. Within each GP-DCIM PE, SRAM macros (47.8%) and RRAM macros (23.7%) dominate, followed by reconfigurable DCIM logic (14.3%). At the chip level, 10 PEs occupy 65.7% of area.

We benchmark PROTEUS against state-of-the-art near-memory and CIM accelerators with ISAs [37], [38], [39], [41], [44], with key features and chip metrics summarized in Table III. PROTEUS combines a 32-bit hierarchical DCIM-ISA, RRAM-based unified storage for micro-programs and weights, and a programmable DCIM datapath/pipeline with efficient INT8/INT16/FP8/FP16 support (widest data-format coverage among compared designs), enabling high memory utilization and flexible mapping of versatile edge AI models. Consequently, PROTEUS achieves among the highest energy efficiency (6.4 TOPS/W) and compute density (0.039 TOPS/mm²), while offering 11× higher peak throughput than [44]. Compared to analog CIM [37], DCIM [38], and near-memory design [41], PROTEUS improves energy efficiency by 5.3×, 22.8×, and 7.6×, respectively. One limitation of PROTEUS is the modest eNVM capacity on die compared to [39], [41]. However, the combination of a GP-DCIM architecture, RRAM-based non-volatile libraries of DCIM micro-programs and weights on each die, and a high-speed multi-chip fabric demonstrates an architectural path where adding chiplets directly scales persistent model capacity and compute.

VII. Conclusion

PROTEUS integrates hierarchical programmability into a unified CIM fabric while preserving DCIM's precision, efficiency, and compute density. With its DCIM-native ISA, PROTEUS provides a fully programmable and adaptable datapath supporting INT8, INT16, FP8, and FP16. Exploiting the synergy of SRAM and RRAM, PROTEUS uses high-density RRAM for persistent storage of model weights and micro-programs, enabling on-demand inference and rapid backbone switching, while SRAM supports efficient dynamic computations. Fabricated in 40 nm ULP CMOS with foundry RRAM, PROTEUS achieves 702 GOPS peak throughput, 6.4 TOPS/W peak energy efficiency and 0.039 TOPS/mm² compute density. PROTEUS is validated on CNN, Transformer, hybrid CNN–Transformer, GNN, and SSM architectures with representative models such as ResNet-20, BERT-Tiny, MobileViT, GraphSAGE, and Vision Mamba, demonstrating its versatility for edge AI applications.

Acknowledgment

The authors would like to thank additional TSMC colleagues for the chip fabrication support, as well as other collaborators from TSMC Corporate Research Design Solution Department. We also thank Dr. Luke Upton from Stanford University (now with Texas Instruments) for valuable technical discussions on the RRAM circuitry design, and Evelyn Colon from the University of Florida for her contributions to the custom DCIM-ISA compiler development.

References

[1] 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 Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 217–230, 2018.

[2] H. Jia, H. Valavi, Y. Tang, J. Zhang, and N. Verma, "A programmable heterogeneous microprocessor based on bit-scalable in-memory computing," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 9, pp. 2609–2621, 2020.

[3] 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 Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, 2020.

[4] J. Wang, X. Wang, C. Eckert, A. Subramaniyan, R. Das, D. Blaauw, and D. Sylvester, "14.2 A compute SRAM with bit-serial integer/floating-point operations for programmable in-memory vector acceleration," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 224–226.

[5] S. Jain, L. Lin, and M. Alioto, "±CIM SRAM for signed in-memory broad-purpose computing from DSP to neural processing," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 10, pp. 2981–2992, 2021.

[6] M. E. Sinangil, B. Erbagci, R. Naous, K. Akarvardar, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, "A 7-nm compute-in-memory SRAM macro supporting multi-bit input, weight and output and achieving 351 TOPS/W and 372.4 GOPS," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 188–198, 2020.

[7] G. Desoli, N. Chawla, T. Boesch, M. Avodhyawasi, H. Rawat, H. Chawla, V. Abhijith, P. Zambotti, A. Sharma, C. Cappetta, M. Rossi, A. De Vita, and F. Girardi, "16.7 A 40-310TOPS/W SRAM-based all-digital up to 4b in-memory computing multi-tiled NN accelerator in FD-SOI 18nm for deep-learning edge applications," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2023, pp. 260–262.

[8] H. Fujiwara, H. Mori, W.-C. Zhao, K. Khare, C.-E. Lee, X. Peng, V. Joshi, C.-K. Chuang, S.-H. Hsu, T. Hashizume, T. Naganuma, C.-H. Tien, Y.-Y. Liu, Y.-C. Lai, C.-F. Lee, T.-L. Chou, K. Akarvardar, S. Adham, Y. Wang, Y.-D. Chih, Y.-H. Chen, H.-J. Liao, and T.-Y. J. Chang, "34.4 A 3nm, 32.5 TOPS/W, 55.0 TOPS/mm 2 and 3.78 Mb/mm 2 fully-digital compute-in-memory macro supporting INT12× INT12 with a parallel-MAC architecture and foundry 6T-SRAM bit cell," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2024, pp. 572–574.

[9] P. Chen, W. Zhao, M. Wu, L. Shen, T. Jia, R. Huang, L. Ye, and Y. Ma, "A 22-nm Delta–Sigma Computing-In-Memory SRAM Macro With Near-Zero-Mean Outputs and LSB-First ADCs for Edge AI Processing," *IEEE Journal of Solid-State Circuits*, 2025.

[10] K. Yoshioka, "A 818–4094 tops/w capacitor-reconfigured analog cim for unified acceleration of cnns and transformers," *IEEE Journal of Solid-State Circuits*, vol. 60, no. 5, pp. 1844–1855, 2025.

[11] H. Diao, Y. He, X. Li, C. Tang, W. Jia, J. Yue, H. Luo, J. Song, X. Li, H. Yang, H. Jia, Y. Liu, Y. Wang, and X. Tang, "A multiply-less approximate sram compute-in-memory macro for neural-network inference," *IEEE Journal of Solid-State Circuits*, 2024.

[12] L. Wang, W. Li, Z. Zhou, H. Gao, Z. Li, W. Ye, H. Hu, J. Liu, J. Yue, J. Yang, Q. Luo, C. Dou, Q. Liu, and M. Liu, "34.9 a flash-sram-adc-fused plastic computing-in-memory macro for learning in neural networks in a standard 14nm finfet process," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67. IEEE, 2024, pp. 582–584.

[13] R. Guo, Z. Yue, X. Si, H. Li, T. Hu, L. Tang, Y. Wang, H. Sun, L. Liu, M.-F. Chang, Q. Li, S. Wei, and S. Yin, "TT@ CIM: A tensor-train in-memory-computing processor using bit-level-sparsity optimization and variable precision quantization," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 3, pp. 852–866, 2022.

[14] A. Guo, C. Xi, F. Dong, X. Pu, D. Li, J. Zhang, X. Dong, H. Gao, Y. Zhang, B. Wang, J. Yang, and X. Si, "A 28-nm 64-kb 31.6-TFLOPS/W digital-domain floating-point-computing-unit and double-bit 6T-SRAM computing-in-memory macro for floating-point CNNs," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 9, pp. 3032–3044, 2024.

[15] A. Guo, X. Dong, F. Dong, D. Li, Y. Zhang, J. Zhang, J. Yang, and X. Si, "A 28 nm 128-kb Exponent-and Mantissa-Computation-In-Memory Dual-Macro for Floating-Point and INT CNNs," *IEEE Journal of Solid-State Circuits*, 2025.

[16] Y. Yuan, B. Zhang, Y. Yang, Y. Luo, Q. Chen, S. Lv, H. Wu, C. Ma, M. Li, J. Yue, X. Wang, G. Xing, P.-I. Mak, X. Li, and F. Zhang, "14.5 a 28nm 192.3 tflops/w accurate/approximate dual-mode-transpose digital 6t-sram cim macro for floating-point edge training and inference," in *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 68. IEEE, 2025, pp. 258–260.

[17] F. Tu, Z. Wu, Y. Wang, L. Liang, L. Liu, Y. Ding, L. Liu, S. Wei, Y. Xie, and S. Yin, "TranCIM: Full-digital bitline-transpose CIM-based sparse transformer accelerator with pipeline/parallel reconfigurable modes," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 6, pp. 1798–1809, 2022.

[18] S. Xie, C. Ni, A. Sayal, P. Jain, F. Hamzaoglu, and J. P. Kulkarni, "16.2 edram-cim: Compute-in-memory design with reconfigurable embedded-dynamic-memory array realizing adaptive data converters and charge-domain computing," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 248–250.

[19] W.-S. Khwa, P.-C. Wu, J.-J. Wu, J.-W. Su, H.-Y. Chen, Z.-E. Ke, T.-C. Chiu, J.-M. Hsu, C.-Y. Cheng, Y.-C. Chen, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, and M.-F. Chang, "34.2 A 16nm 96Kb integer/floating-point dual-mode-gain-cell-computing-in-memory macro achieving 73.3-163.3TOPS/W and 33.2-91.2TFLOPS/W for AI-edge devices," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2024, pp. 568–570.

[20] W.-S. Khwa, P.-C. Wu, J.-W. Su, C.-Y. Cheng, J.-M. Hsu, Y.-C. Chen, L.-J. Hsieh, J.-C. Bai, Y.-S. Kao, T.-H. Lou, A. S. Lele, J.-J. Wu, J.-C. Tien, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, and M.-F. Chang, "14.2 A 16nm 216kb, 188.4TOPS/W and 133.5TFLOPS/W microscaling multi-mode gain-cell CIM macro for edge-AI devices," in *2025 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2025, pp. 1–3.

[21] S. Kim, Z. Li, S. Um, W. Jo, S. Ha, J. Lee, S. Kim, D. Han, and H.-J. Yoo, "Dynaplasia: An edram in-memory computing-based reconfigurable spatial accelerator with triple-mode cell," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 1, pp. 102–115, 2023.

[22] Y. He, S. Fan, X. Li, L. Lei, W. Jia, C. Tang, Y. Li, Z. Huang, Z. Du, J. Yue, X. Li, H. Yang, H. Jia, and Y. Liu, "34.7 a 28nm 2.4 mb/mm 2 6.9-16.3 tops/mm 2 edram-lut-based digital-computing-in-memory macro with in-memory encoding and refreshing," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67. IEEE, 2024, pp. 578–580.

[23] L. Zheng, A. M. Bavani, S. Du, T.-Y. Hsin, M. Chen, W.-S. Khwa, A. Lele, H. Chuang, Y.-D. Chih, M.-F. Chang, and H. Li, "CENTAUR: A 38.5-TFLOPS/W 600MHz Floating-Point Digital Compute-In-Memory Engine with 40nm Fusion RRAM-eDRAM Macros Featuring 3D-MAC Operation," in *IEEE Asian Solid-State Circuits Conference (ASSCC)*, 2025.

[24] A. S. Lele, M. Chang, S. D. Spetalnick, B. Crafton, S. Konno, Z. Wan, A. Bhat, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A heterogeneous rram in-memory and sram near-memory soc for fused frame and event-based target identification and tracking," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 1, pp. 52–64, 2023.

[25] T.-H. Wen, J.-M. Hung, W.-H. Huang, C.-J. Jhang, Y.-C. Lo, H.-H. Hsu, Z.-E. Ke, Y.-C. Chen, Y.-H. Chin, C.-I. Su, W.-S. Khwa, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, M.-S. Ho, C.-C. Chou, Y.-D. Chih, T.-Y. J. Chang, and M.-F. Chang, "Fusion of memristor and digital compute-in-memory processing for energy-efficient edge computing," *Science*, vol. 384, no. 6693, pp. 325–332, 2024.

[26] C. Mu, Z. Huang, H. Jiang, J. Liao, Y. Zhou, L. Chen, Y. Zhang, H. Zhu, J. Yang, Q. Liu, and C. Chen, "A 28-nm RRAM/SRAM Collaborative CIM Accelerator Supporting RRAM-Endurance-Latency Awareness for Edge Fine-Tuning," *IEEE Journal of Solid-State Circuits*, 2025.

[27] J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40-nm 118.44-TOPS/W voltage-sensing compute-in-memory RRAM macro with write verification and multi-bit encoding," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 3, pp. 845–857, 2022.

[28] T.-H. Wen, H.-H. Hsu, W.-S. Khwa, W.-H. Huang, Z.-E. Ke, Y.-H. Chin, H.-J. Wen, Y.-C. Chang, W.-T. Hsu, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, S.-H. Teng, C.-C. Chou, Y.-D. Chih, T.-Y. J. Chang, and M.-F. Chang, "34.8 A 22nm 16Mb floating-point ReRAM compute-in-memory macro with 31.2TFLOPS/W for AI edge devices," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2024, pp. 580–582.

[29] H.-H. Hsu, T.-H. Wen, W.-H. Huang, W.-S. Khwa, Y.-C. Lo, C.-J. Jhang, Y.-H. Chin, Y.-C. Chen, C.-C. Lo, R.-S. Liu, K.-T. Tang, C.-C. Hsieh, Y.-D. Chih, T.-Y. J. Chang, and M.-F. Chang, "A nonvolatile AI-edge processor with SLC–MLC hybrid ReRAM compute-in-memory macro using current–voltage-hybrid readout scheme," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 1, pp. 116–127, 2023.

[30] H. Li, W.-C. Chen, A. Levy, C.-H. Wang, H. Wang, P.-H. Chen, W. Wan, H.-S. P. Wong, and P. Raina, "One-shot learning with memory-augmented neural networks using a 64-kbit, 118 GOPS/W RRAM-based non-volatile associative memory," in *2021 Symposium on VLSI Technology*. IEEE, 2021, pp. 1–2.

[31] D.-Q. You, W.-S. Khwa, B. Zhang, F.-Y. Chen, A. Lee, Y.-C. Hung, Y.-M. Li, Y.-H. Wang, C.-C. Lo, R.-S. Liu, K.-T. Tang, C.-C. Hsieh, Y.-D.

Chih, T.-Y. J. Chang, and M.-F. Chang, "14.1 A 22nm 104.5TOPS/W μ-NMC-Δ-IMC heterogeneous STT-MRAM CIM macro for noise-tolerant Bayesian neural networks," in *2025 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2025, pp. 1–3.

[32] P. Deaville, B. Zhang, and N. Verma, "A fully row/column-parallel in-memory computing macro in foundry mram with differential readout for noise rejection," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 7, pp. 2070–2080, 2024.

[33] H. Cai, Z. Bian, Y. Hou, Y. Zhou, J.-l. Cui, Y. Guo, X. Tian, B. Liu, X. Si, Z. Wang, J. Yang, and W. Shan, "33.4 A 28nm 2Mb STT-MRAM computing-in-memory macro with a refined bit-cell and 22.4-41.5 TOPS/W for AI inference," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2023, pp. 500–502.

[34] S. Kwon, S. Myung, J. An, H. Kim, M. Kim, H. Lee, W. Yi, S. Jung, D. Yoon, S. Han, S. Chung, K. Lee, J.-H. Park, K. Lee, S. J. Kim, and D. Ham, "37.3 monolithic in-memory computing microprocessor for end-to-end dnn inferencing in mram-embedded 28nm cmos technology with 1.1 mb weight storage," in *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 68. IEEE, 2025, pp. 1–3.

[35] R. Khaddam-Aljameh, M. Stanisavljevic, J. Fornt Mas, G. Karunaratne, M. Brändli, F. Liu, A. Singh, S. M. Müller, U. Egger, A. Petropoulos, T. Antonakopoulos, K. Brew, S. Choi, I. Ok, F. L. Lie, N. Saulnier, V. Chan, I. Ahsan, V. Narayanan, S. R. Nandakumar, M. Le Gallo, P. A. Francese, A. Sebastian, and E. Eleftheriou, "HERMES-core—A 1.59-TOPS/mm 2 PCM on 14-nm CMOS in-memory compute core using 300-ps/LSB linearized CCO-based ADCs," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1027–1038, 2022.

[36] M. Verhelst, L. Benini, and N. Verma, "How to Keep Pushing ML Accelerator Performance? Know Your Rooflines!" *IEEE Journal of Solid-State Circuits*, vol. 60, no. 6, pp. 1888–1905, Jun. 2025.

[37] M. Chang, A. S. Lele, S. D. Spetalnick, B. Crafton, S. Konno, Z. Wan, A. Bhat, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "29.5 A 73.53TOPS/W 14.74TOPS heterogeneous RRAM in-memory and SRAM near-memory SoC for hybrid frame and event-based target tracking," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2023, pp. 426–428.

[38] Y. Wang, M. Yang, C.-P. Lo, and J. P. Kulkarni, "30.6 Vecim: A 289.13GOPS/W RISC-V vector co-processor with compute-in-memory vector register file for efficient high-performance computing," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2024, pp. 492–494.

[39] K. Prabhu, R. M. Radway, J. Yu, K. Bartolone, M. Giordano, F. Pedding-haus, Y. Urman, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, P. Raina, and S. Mitra, "MINOTAUR: An edge Transformer inference and training accelerator with 12 MBytes on-chip resistive RAM and fine-grained spatiotemporal power gating," in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2024, pp. 1–2.

[40] Z. Fan, Q. Zhang, H. An, B. Xu, L. Xu, C.-W. Tseng, Y. Peng, A. Bejarano-Carbo, P. Abillama, A. Cao, B. Liu, C. Lee, Z. Wang, H.-S. Kim, D. Blaauw, and D. Sylvester, "AIMMI: Audio and Image Multi-Modal Intelligence via a Low-Power SoC With 2-MByte On-Chip MRAM for IoT Devices," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 10, pp. 3488–3501, 2024.

[41] S. D. Spetalnick, A. S. Lele, B. Crafton, M. Chang, S. Ryu, J. H. Yoon, Z. Hao, A. Ansari, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "30.1 A 40nm VLIW edge accelerator with 5MB of 0.256pJ/b RRAM and a localization solver for bristle robot surveillance," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2024, pp. 482–484.

[42] S. Yin, B. Zhang, M. Kim, J. Saikia, S. Kwon, S. Myung, H. Kim, S. J. Kim, M. Seok, and J.-s. Seo, "PIMCA: A 3.4-Mb programmable in-memory computing accelerator in 28nm for on-chip DNN inference," in *2021 Symposium on VLSI Circuits (VLSI Circuits)*. IEEE, 2021, pp. 1–2.

[43] Y. Ju, Y. Wei, X. Chen, and J. Gu, "A general-purpose compute-in-memory processor combining CPU and deep learning with elevated CPU efficiency and enhanced data locality," in *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2023, pp. 1–2.

[44] Y. Ju, Y. Wei, and J. Gu, "A 65 nm general-purpose compute-in-memory processor supporting both general programming and deep learning tasks," *IEEE Journal of Solid-State Circuits*, vol. 60, no. 4, pp. 1500–1511, 2025.

[45] T. F. Wu, B. Q. Le, R. Radway, A. Bartolo, W. Hwang, S. Jeong, H. Li, P. Tandon, E. Vianello, P. Vivet *et al.*, "14.3 a 43pj/cycle non-volatile microcontroller with 4.7 μs shutdown/wake-up integrating 2.3-bit/cell resistive ram and resilience techniques," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 226–228.

[46] C.-F. Lee, H.-J. Lin, C.-W. Lien, Y.-D. Chih, and J. Chang, "A 1.4 mb 40-nm embedded reram macro with 0.07 um 2 bit cell, 2.7 ma/100mhz low-power read and hybrid write verify for high endurance application," in *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, 2017, pp. 9–12.

[47] C.-Y. Wu, C.-F. Yang, C.-W. Lai, Y.-T. Wu, T.-C. Chien, M.-H. Yang, M.-T. Yang, Y.-N. Kao, C.-L. Cheng, C.-Y. Wang *et al.*, "Emerging memory rram embedded in 12ffc finfet technology for industrial applications," in *2023 International Electron Devices Meeting (IEDM)*. IEEE, 2023, pp. 1–4.

[48] Y.-C. Huang, S.-H. Liu, H.-S. Chen, H.-C. Feng, C.-F. Li, C.-Y. Yang, W.-K. Chang, C.-F. Yang, C.-Y. Wu, Y.-C. Lin *et al.*, "15.7 a 32mb rram in a 12nm finfet technology with a 0.0249 μm 2 bit-cell, a 3.2 gb/s read throughput, a 10kcycle write endurance and a 10-year retention at 105° c," in *2024 IEEE international solid-state circuits conference (ISSCC)*, vol. 67. IEEE, 2024, pp. 288–290.

[49] S. Mehta and M. Rastegari, "MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer," in *International Conference on Learning Representations*, 2022.

[50] S. Kim, S. Kim, W. Jo, S. Kim, S. Hong, and H.-J. Yoo, "20.5 C-Transformer: A 2.6–18.1μj/token homogeneous DNN-Transformer/Spiking-Transformer processor with big-little network and implicit weight generation for large language models," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67. IEEE, 2024, pp. 368–370.

[51] B. Zhang, S. Moon, and M. Seok, "A 1-TFLOPS/W, 28-nm deep neural network accelerator featuring online compression and decompression and BF16 digital in-memory-computing hardware," in *2024 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2024, pp. 1–2.

[52] S. Moon, M. Li, G. K. Chen, P. C. Knag, R. K. Krishnarmurthy, and M. Seok, "T-rex: Hardware–software co-optimized transformer accelerator with reduced external memory access and enhanced hardware utilization," *IEEE Journal of Solid-State Circuits*, 2025.

[53] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[54] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 1024–1034.

[55] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: efficient visual representation learning with bidirectional state space model," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 62 429–62 442.

Luqi Zheng received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2021. He is currently pursuing the Ph.D. degree with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. He has experience in AI hardware and chip design across both academia and industry. He currently holds an internship position with Kilby Labs, Texas Instruments, Dallas, TX.

His research interests include AI accelerators, system-on-chip (SoC) design, compute-in-memory architectures, emerging memory systems, and hardware–software co-design.

Amir Massah Bavani received the M.Sc. degree in electrical and computer engineering from the Technical University of Kaiserslautern, Kaiserslautern, Germany. He also received the M.Sc. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA. He has worked as a Digital Design Engineer for ASIC/FPGA systems in both industry and academia, with experience in RTL design, FPGA implementation, ASIC design flows, and SoC integration.

His research interests and working areas include compute-in-memory architectures, AI hardware accelerators, RISC-V-based SoCs, digital ASIC design, FPGA-based prototyping, and energy-efficient high-performance digital systems.

Mufeng Chen received the B.S. degree in optics and electronic information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2021. From 2021 to 2024, he was a Research Assistant with Zhejiang University, Hangzhou, China, where he worked on hardware/software co-design and system integration for emerging computing systems. He is currently pursuing the Ph.D. degree with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, USA.

His research interests include heterogeneous memory systems, memory-centric architectures, hardware/software co-design, and system-level integration for data-intensive computing.

Shuting Du received the B.S. degree in electronic science and engineering from Southeast University, Nanjing, China, in 2020, and the M.S. degree in electrical and computer engineering from Duke University, Durham, NC, USA, in 2023. She is currently pursuing the Ph.D. degree with Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, specializing in algorithm-hardware co-design and memory-centric AI hardware architectures.

Her research interests include in-memory computing, emerging memory technologies and 3D integration.

Te-Yu Hsin received the B.S. degree and the M.S. degree in computer engineering from Purdue University, West Lafayette, IN, USA.

He is currently an ASIC Design Engineer with OmniVision, Santa Clara, CA, USA, where he focuses on digital IC design. He previously held an internship position at Intel Corporation, where he contributed to qubit control logic and circuit design. His research and development interests include compute-in-memory (CIM) architectures and systems, LCoS (Liquid Crystal on Silicon) display drivers, and cryogenic/advanced control systems for emerging technologies.

Win-San Khwa (Senior Member, IEEE) received the B.S. degree from the University of California at Los Angeles, Los Angeles, CA, USA, in 2007, the M.S. degree from the University of Michigan, Ann Arbor, MI, USA, in 2010, and the Ph.D. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2017. In 2012, he joined Macronix International (MXIC), Hsinchu. He is currently a Manager with the Corporate Research Design Solution Department, Taiwan Semiconductor Manufacturing Company, Hsinchu, on emerging memory path finding and IP development. His research interests include circuit-device optimization designs of emerging memories for artificial intelligence applications.

Dr. Khwa served as the Emerging Technologies Subcommittee Chair for CICC 2025, the CICC TPC from 2022 to 2025, the DAC TPC from 2024 to 2026, and the ISSCC TPC since 2027.

Ping-Sheng Wu received his B.S. degree in Electrical Engineering and M.S. degree in Electronics Engineering from National Taiwan University, Taipei, Taiwan, in 2021 and 2024, respectively. He is currently a circuit design engineer with the Corporate Research Design Solution Department, Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu, Taiwan.

His research interests include energy-efficient hardware for edge-AI acceleration.

Ashwin Sanjay Lele (Member, IEEE) received the B.Tech. and M.Tech. degrees in electrical engineering from IIT Bombay, Mumbai, India, in 2019, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2023. He previously held internship positions at Intel, Bengaluru, India, and Qualcomm Inc., San Jose, CA, USA. He is currently a Principal Engineer with the Corporate Research Department, Taiwan Semiconductor Manufacturing Company (TSMC), San Jose.

His research interests broadly lie in low-power hardware design, and emerging memory systems and their applications.

Bo Zhang received the B.S. degree in microelectronics (major) and mathematics (minor) from Shanghai Jiao Tong University, Shanghai, China, in 2018, and the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 2020 and 2023, respectively.

Since 2024, he has been with Taiwan Semiconductor Manufacturing Company (TSMC), San Jose, CA, USA, as a Principal Engineer. His research interests include computer architecture for machine learning algorithms, compute-in-memory architectures, and ultralow-power VLSI circuit and system design.

Dr. Zhang was a recipient of the IBM Ph.D. Fellowship in 2022.

Brian Crafton (Member, IEEE) received the B.S. degree in computer engineering from Northeastern University, Boston, MA, USA, in 2017, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA, in 2023.

He is currently a Principal Engineer with TSMC, San Jose, CA, USA.

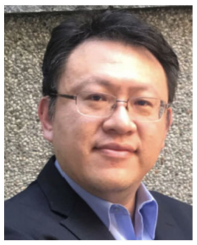
His research interests include integrated circuit design for machine learning applications.

Harry Chuang received the B.S. degree in electrical engineering from Oregon State University, Corvallis, OR, USA, and the M.S.E.E./Ph.D. and degrees in electrical engineering from Purdue University, West Lafayette, IN, USA. From 1990 to 1994, he was with IBM Analog and Mixed Signal Technology Development, Burlington, VT, USA. From 1994 to 2000, he was with Motorola APRDL PowerPC CPU Technology Development, Austin, TX, USA.

Since 2000, he has been with TSMC R&D Logic Technology Development for CPU/GPU in Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu, Taiwan. He is currently a R&D Director of the MRAM Technology Development. He has authored or co-authored over 20 papers. He holds 304 U.S. patents and more than 500 patents worldwide.

Yu-Der Chih received the B.S. degree in physics from National Taiwan University, Taipei, Taiwan, in 1988, and the M.S. degree in electronics engineering from National Tsing-Hua University, Hsinchu, Taiwan, in 1992.

From 1992 to 1997, he was a Design Engineer of Ethernet transceiver circuits for data communication with Macronix, Hsinchu, and a Circuit Design Engineer of SDRAM with Powerchip, Hsinchu. In 1997, he joined Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu, where he was involved in the development of embedded nonvolatile memory IP, including embedded flash, OTP, MTP, and emerging memory. He is currently a TSMC Academician and the Director of the Embedded Nonvolatile Memory Library Department with the Memory Solution Division.



Meng-Fan Chang (Fellow, IEEE) received the M.S. degree from The Pennsylvania State University, State College, PA, USA, and the Ph.D. degree from National Chiao Tung University, Hsinchu, Taiwan.

Prior to 2006, he worked in the industry for over ten years. This included the design of memory compilers (Mentor Graphics, Wilsonville, OR, USA, from 1996 to 1997) and the design of embedded static random access memory (SRAM) and Flash macros (Design Service Division, TSMC, Hsinchu, from 1997 to 2001). In 2001, he co-founded IPLib, Hsinchu, where he developed embedded SRAM and ROM compilers, flash macros, and flat-cell ROM products until 2006. He is currently a Distinguished Professor with National Tsing Hua University (NTHU) and the Director of the Corporate Research, TSMC. His research interests include circuit design for volatile and nonvolatile memory (NVM), ultralow-voltage systems, 3-D-memory circuit–device interactions, spintronic circuits, memristor logics for neuromorphic computing, and computing-in-memory (CIM) for artificial intelligence.

Dr. Chang was a recipient of several prestigious national-level awards in Taiwan, including the Outstanding Research Award from MOST-Taiwan, the Outstanding Electrical Engineering Professor Award, the Academia Sinica Junior Research Investigator Award, and the Ta-You Wu Memorial Award. He was a Distinguished Lecturer of the IEEE Solid-State Circuits Society (SSCS) and the Circuits and Systems Society (CASS), the Chair of the Nano-Giga technical Committee of CASS, an Administrative Committee (AdCom) Member of the IEEE Nanotechnology Council, the Chair of SSCS Taipei Chapter, and Chair of the IEEE Taipei Section. He has been serving as an Associate Editor for IEEE Journal of Solid-State Circuits (JSSC), IEEE Transactions on Very Large Scale Integration (TVLSI), IEEE Transactions on Circuits and Systems—I: Regular Papers (TCAS-I), and IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), and as the Guest Editor for IEEE JSSC, IEEE Transactions on Circuits and Systems—II: Express Briefs (TCAS-II), and IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS). He has also been serving on the Executive Committee for IEDM, as well as Subcommittee Chairs for ISSCC, IEDM, DAC, ISCAS, VLSI-DAT, and ASP-DAC. He was the Program Director of the Micro-Electronics Program at the Ministry of Science and Technology (MOST), the Associate Executive Director of the National Program of Intelligent Electronics (NPIE), and the NPIE Bridge Program initiated by the Taiwan Government. He received recognition as a top-10 contributor of papers to ISSCC over the past 70 years.

Haitong Li (Member, IEEE) is an Assistant Professor in the Elmore Family School of Electrical and Computer Engineering at Purdue University. He obtained his Ph.D. and M.S. in Electrical Engineering from Stanford University advised by Prof. H.-S. Philip Wong. His research interests are in the new memory device technologies, circuit designs, and 3D integration for energy-efficient AI and unconventional computing. His work was featured as part of the "New Compute Trajectories for Energy-Efficient Computing" in "The Decadal Plan for Semiconductors" by SRC. He has served as a committee member of the IEEE Electron Devices Society (EDS) VLSI Technology and Circuits Committee, as a TPC member of DAC, VLSI-TSA, DATE, ICCAD, and as a Track Chair of GLSVLSI.

His work has been recognized through an NSF CAREER Award in 2026, Purdue's Seed for Success Acorn Award in 2025, Roger A. Haken Best Paper Award at IEDM 2024, and IEEE EDS PhD Student Fellowship in 2019. He was an invited participant in the 2025 Japan-America Frontiers of Engineering Symposium co-organized by the U.S. National Academy of Engineering.