Prospects of Computing In or Near Flash Memories

Hang-Ting Lue, Chun-Hsiung Hung, Keh-Chung Wang and Chih-Yuan Lu

Macronix International Co., Ltd
16 Li-Hsin Road, Hsinchu Science Park, Hsinchu, Taiwan.
E-mail: htlue@mxic.com.tw

***Abstract*—Memory-centric computing is emerging as a potential solution to address the memory bottleneck, especially in generative AI. In current AI hardware, most users focus primarily on the processor and DRAM as the main computing devices, while Flash memory remains peripheral to AI. In this paper, we explore the potential of Flash memory to play a more effective role in AI computing. Flash memory offers advantages such as high density and non-volatility, making it suitable for both edge and cloud AI computing, which require low-cost and low-power solutions. Flash memory will continue to leverage monolithic 3D stacking for higher density, improve read bandwidth to approach that of DRAM, and most importantly, add computing functions near or in Flash memory to support various AI computing tasks. We provide two specific examples. First, we introduce 3D NOR technology and the concept of high-bandwidth digital computing in memory (HB dCIM) to support the general matrix-vector (GEMV) accelerator for large language model (LLM) inference. Second, we discuss an in-memory search (IMS) accelerator using 3D NAND for approximate nearest-neighbor search applications.**

I. Introduction

Flash memories are classic and mainstream non-volatile memory devices. NAND Flash has successfully evolved into high-layer stacked 3D NAND and is currently the most dominant high-density non-volatile memory device on the market. NOR Flash, an older classic device, ceased scaling at the 2D 45nm node but still maintains a significant niche market due to its unique advantages. **Figure 1** provides a brief comparison of NAND and NOR Flash. In summary, NAND offers very high density, while NOR Flash features fast random access read speeds and excellent reliability without the need for a controller chip.

Figure 1 NAND and NOR Flash memory architectures and comparisons.

Figure 2 illustrates the role of Flash memory in computing. In current AI hardware, Flash memory devices are often secondary to DRAM for initial data loading. Computing primarily revolves around the processor (including SRAM) and DRAM. When DRAM capacity is insufficient, intensive data swapping between DRAM and Flash occurs, usually hindering system performance. We anticipate a future paradigm shift. Flash memory will continually improve read bandwidth and latency to complement DRAM. Additionally, computing in or near Flash memory will help reduce data traffic in AI data flow. The high density and non-volatile properties of Flash memory are expected to benefit AI systems, particularly for edge inference devices, and hold potential for big data cloud computing.

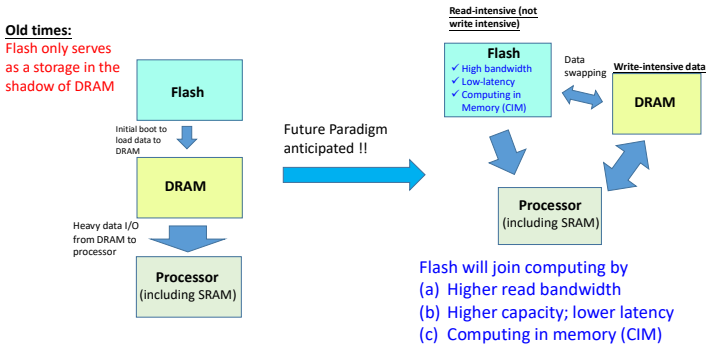


Figure 2 Future prospects of Flash memory to join AI computing.

II. Introduction of 3D NOR Technology

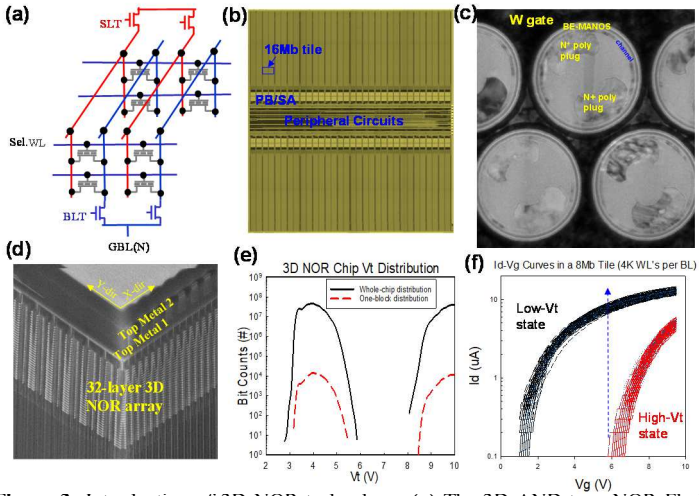


Figure 3 Introduction of 3D NOR technology. **(a)** The 3D AND-type NOR Flash architecture. **(b)** The 4Gb 3D NOR test chip. **(c)** The 3D NOR device plan-view. **(d)** The 3D bird's eye view of 32-layer 3D NOR. **(e)** The whole-chip and one-block initial (P/E=1) Vt distribution for low-Vt and high-Vt state. The initial RBER=0 at Gb-density, with large Vt separation between "0" and "1". **(f)** The measured Id-Vg curves in a 16Mb tile. The Ion/Ioff ratio is sufficient, with Ion around several uA at low V_{BL} bias (~0.3V), while Ioff from a 8K-WL BL background leakage is well below 0.1uA.

We are developing a 32-layer 3D NOR Flash technology, summarized in **Figure 3**. This technology features a 3D AND-type architecture [1], with +FN/-FN operations, differing from the channel hot electron mechanism used in 2D NOR. We have developed a 4Gb test chip, which offers eight times the memory capacity of our last-generation 512Mb 2D NOR Flash. The 3D NOR technology utilizes similar processing and equipment as 3D NAND to achieve high-layer stacking and supports continuous 3D stacking. Unlike the 3D NAND structure, our design requires enlarging the hole size to insert two additional vertical poly plugs, creating an AND-type structure that connects memory transistors in parallel in each local bitline. This allows 3D NOR to achieve a much higher sensing current of several uA (compared to approximately 20nA in

NAND) for random access reads, which take around 100ns. 3D NOR Flash creates an excessive large Vt window, enabling an initial RBER=0 for Gb-density (after few repair). A 1-bit ECC is embedded in the standard NOR Flash (SPI, Octa) design to provide robust reliability for high cycling and retention. Additionally, the BL background leakage current can be kept well below 0.1 uA in a 16Mb tile (with 8K parallel-connected WLs) after erase-state Vt distribution control. These properties are crucial for enabling HB dCIM, which will be discussed later.

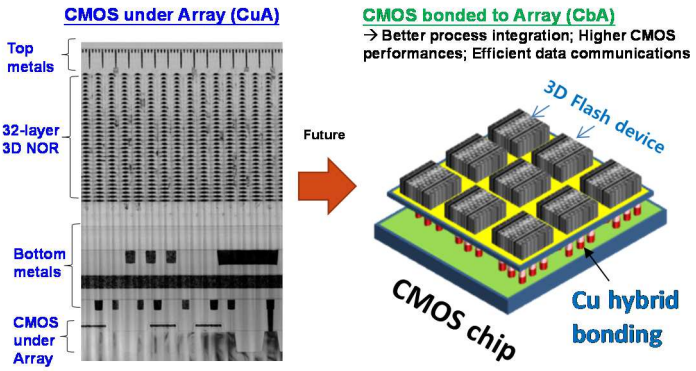


Figure 4 Current 3D NOR adopts CMOS under Array (CuA) process integration. In the future, we will go for Cu hybrid bonding to connect separated CMOS chip with 3D array (like 3D NAND).

Figure 4 illustrates the CMOS and array process integration for 3D NOR Flash. Similar to current standard 3D NAND, 3D NOR adopts CMOS under Array (CuA) integration. Peripheral WL driver circuits are hidden under each small tile (16Mb), though some peripheral circuits cannot be concealed beneath the array due to the smaller tile size of 3D NOR compared to the larger plane of 3D NAND. Multiple tiles share the same global bitline and are connected to peripheral page buffer and sense amplifier circuits.

Cu hybrid bonding has become a popular method for 3DIC integration. Both 3D NAND and 3D NOR will adopt Cu hybrid bonding in the future to meet the demand for fast I/O and integrate high-speed CMOS. This method not only simplifies process integration but also significantly improves data communication with CMOS chips, which is beneficial for high-performance system-on-chip (SoC) designs integrated with advanced memory.

III. Introduction of 3D NOR CIM

Figure 5 provides a brief summary of the training and inference characteristics related to memory requirements in AI. Training processes are highly write-intensive and are typically bottlenecked by GPU computing. In contrast, inference operations involve mostly stationary weights and are predominantly read-intensive, making Flash memory a viable candidate for computing tasks. For inference devices in generative AI, such as large language models (LLMs), it has been reported [2] that GPU utilization is often very low (<1%), with memory bandwidth being the primary bottleneck. The challenge lies in the need to read billions of weights once for every token without reuse. Benchmark parameters frequently include metrics like tokens per second, energy consumption, and perplexity scores across various precision levels. Edge AI presents numerous opportunities for customized ASIC and memory designs aimed at achieving low-power and cost-effective inference devices. Computing in memory (CIM) offers advantages in accelerating GEMV matrix-vector computations, reducing data traffic in chip I/O, and lowering energy consumption and system costs significantly.

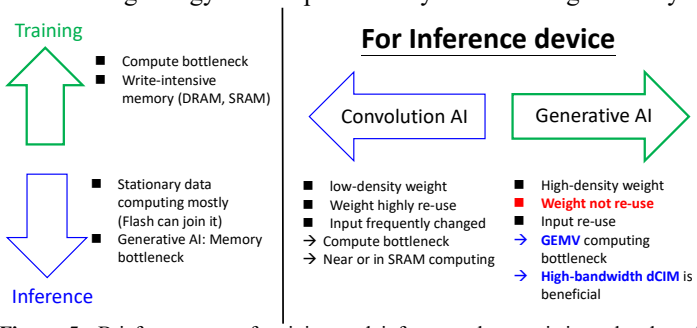


Figure 5 Brief summary of training and inference characteristics related to the memory requirements. Flash memory is suitable for inference device. For generative AI that has huge parameters without weight reuse, a HB dCIM is beneficial to resolve the memory bottleneck.

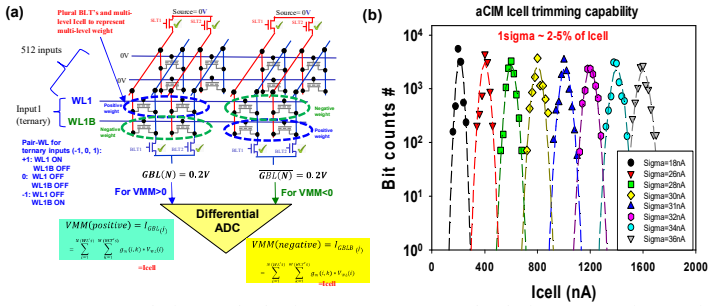


Figure 6 aCIM design method using 3D NOR array. The design concept is generic and usually adopted for various kinds of aCIM using non-volatile memory devices. **(a)** Sum cell currents with many WL inputs to represent the GEMV, using pair GBL's and differential ADC to detect the matrix-vector computing result. **(b)** The cell current (weight) has physical limitation with certain standard deviation (sigma) and offset. Sigma ranges from 2 to 5%, depending on the Icell mean value and algorithms.

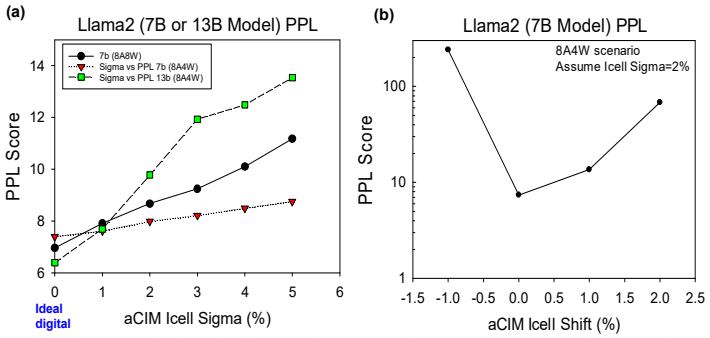


Figure 7 aCIM design challenges for LLM (Llama 2). **(a)** Error tolerance of Icell sigma; **(b)** Error tolerance of Icell shifts. Major aCIM parameters: 1024 WL inputs, 8-bit ADC (assuming optimized quantization for various summed current range). W4A8 stands for 4b weight and 8b input, and so on. PS: **PPL score is the lower the better**.

There are two paths for the GEMV accelerator: analog Computing in Memory (aCIM) and digital Computing in Memory (dCIM). We have explored the feasibility of 3D NOR aCIM for VGG and ResNET neural networks used in image classification [3] with the Cifar-10 test database. In **Figure 6**, aCIM involves parallel summation of cell currents from multiple word lines (WLs) into a global bitline (BL). Negative weights (or inputs) can be managed effectively using a differential analog-to-digital converter (ADC) scheme. Due to device limitations such as random telegraph noise (RTN), programming noise, and cell variations, the normalized standard deviation (sigma) of Icell typically ranges between 2% to 5%, depending on the Icell range and programming algorithms. While aCIM shows promise in simple neural networks with reasonable accuracy, we encountered significant challenges when applying it to generative AI applications.

We have studied the public-domain edge LLM: Llama-2 (7B and 13B) models. **Figure 7** illustrates the perplexity (PPL) score of aCIM and its error tolerance with non-ideal Icell sigma and Icell shifts. The

simulations were conducted using Llama-2 models (7B or 13B) with 4-bit weights and 8-bit inputs, incorporating non-ideal weights to simulate practical aCIM devices. Additional design parameters are detailed in the figure caption. The results highlight the LLM's stringent error tolerance for Icell sigma (<1% required), with the most critical scenario being a negative Icell shift (e.g., -1%), which can lead to significant PPL score deterioration. As of now, aCIM has not yet gained mainstream adoption in commercial design and is awaiting breakthroughs.

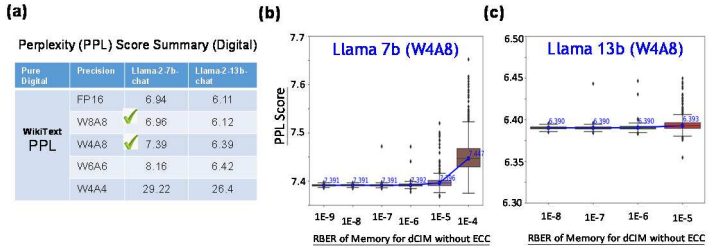


Figure 8 **(a)** The PPL score of digital computing at various precision, from FP16, W8A8, W4A8 and so on. Quantization toward 4bit weight is a favored trend to relieve the AI hardware burden for edge inference. **(b)** Llama-2 7B model (W4A8) for dCIM (no ECC) at various RBER. **(c)** Llama-2 13B model (W4A8) for dCIM (no ECC) at various RBER. WikiText test data set is applied to study the PPL score.

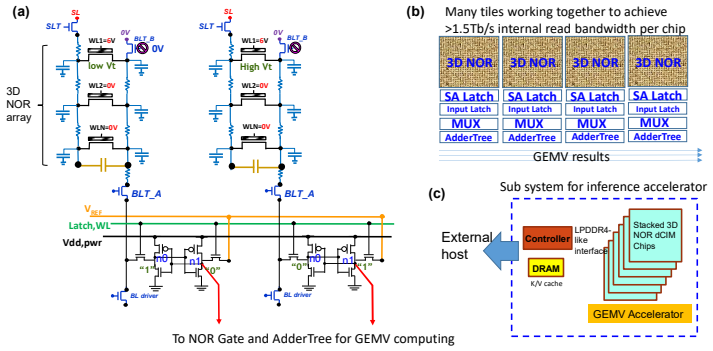


Figure 9 Brief introduction of 3D NOR HB dCIM. **(a)** 3D NOR array local bitline is connected to sense amplifier latch (SA Latch) "n0" node. Low-power voltage sensing method is applied to allow parralellly operated all-BL sensing together in a tile. When the developed BL voltage is above the reference voltage, the SA power refresh will flip the latch "n1" voltage, which will be connected to NOR gate plus AdderTree circuits to perform digital-mode GEMV computing. It's non-Von-Neuman dCIM without read-out the data, and no ECC is allowed. It's the fastest path for dCIM with lowest power consumption. **(b)** We can parallelly operate all tiles (ex: 256 tiles in a chip) to generate >1.5Tb/s internal read bandwidth at chip power around 1W only. The very near-memory digital computing can perform GEMV for a large Matrix (ex: 4K*4K in Llama 2/3) before chip I/O, thus largely save the data traffic between chips. **(c)** A subsystem for edge inference of LLM requires stacked die of 3D NOR dCIM chips, with simple LPDDR4 interface to communicate with a simple NPU controller chip, and a light-DRAM to carry out K/V cache computing in LLM.

We turn our attention to digital Computing in Memory (dCIM) using 3D NOR Flash. Drawing upon the concept of DRAM AiM (or PIM) [2], DRAM possesses much higher internal read bandwidth (>1Tb/s) compared to chip I/O bandwidth. DRAM AiM facilitates near-memory digital computing for GEMV operations prior to chip I/O, thereby substantially reducing data traffic. This approach contrasts with high-bandwidth memory (HBM), which necessitates thousands of chip I/O connections to GPUs, consuming considerable power and incurring high costs.

Figure 8(a) summarizes the perplexity (PPL) scores at various digital precisions. In edge AI, there is a trend towards using integer (INT8 or INT4) weights to reduce AI hardware requirements. For Llama-2, employing 4-bit weights and 8-bit inputs is a reasonable choice with minimal PPL score loss. In HB dCIM design, there is no ECC protection as data must be computed quickly for optimal power efficiency and high bandwidth. We must consider the impact of Raw-Bit-Error-Rate (RBER). **Figures 8(b) and 8(c)** illustrate the effects of RBER on dCIM for Llama-2 7B and 13B models, respectively. The results indicate that an RBER of <1E-7 is necessary to avoid outliers in PPL scores. This indicates the stringent need for a robust memory device with small RBER for dCIM.

Fortunately, our 3D NOR technology exhibits an excellent memory window with an initial RBER=0 for Gb-density, providing a wide margin for reliability. This is a crucial prerequisite for implementing dCIM without ECC. To achieve ultra-high bandwidth with low power consumption, we need to change from conventional current-sensing to a voltage-sensing design. **Figure 9(a)** illustrates the concept of 3D NOR dCIM, where the local bitline (BL) is developed by biasing the voltage of the local source line according to the "1" or "0" states. Achieving accurate data readouts in voltage sensing requires a high Ion/Ioff ratio in large memory tiles, a capability successfully demonstrated in the 3D NOR array.

The slight difference of "n0" (controlled by BL developing) vs. "n1" nodes (controlled by reference voltage) will flip the latch data to be either Vdd (1.1V) or GND (0V) after power refresh. The "n1" node can be connected to NOR gate plus AdderTree to perform integer Matrix-Vector computing, similar to SRAM dCIM [4]. **Figure 9(b)** shows that we will store the inputs in the local latch within each memory tile for GEMV computing where inputs are re-used while changing the 3D NOR WL's. The 3D NOR memory sensing time is in the range of ~100ns, which can be suitably pipeline designed to match the latency of AdderTree computing of 8b inputs by modest MUX ratio control. Local BUS carry the AdderTree results toward peripheral digital circuits to complete GEMV.

Figure 9(c) illustrates the scenario of subsystem for inference accelerator. The 3D NOR dCIM can compute the large matrix (ex: 4K*4K for Llama 2/3) inside chip, and the I/O data traffic is reduced by nearly 1/1000 times, thus a regular LPDDR4 chip interface can be enough to communicate with an external NPU controller. A light-density and modest-bandwidth DRAM is used to compute the K/V cache in transformer AI. Depending on the attention depth of LLM, the smaller depth requires lighter usage of DRAM. For LLM, we can roughly estimate the token/sec performance by:

Token/sec = Memory Bandwidth / (Memory Capacity) * Factor (peripheral digital overhead in chip communications)

A 6-die 4.5Gb 3D NOR dCIM subsystem can efficiently handle the GEMV part of the Llama-7B model with 4-bit weight quantization. The estimated token/sec exceeds 200, including digital overhead in the NPU and DRAM communications. The advantages of 3D NOR dCIM for LLM can be summarized as follows:

(1) High-Bandwidth GEMV but saving chip I/O: dCIM significantly reduces data traffic between chip I/O by up to a thousandfold, eliminating the need for expensive wide I/O HBM designs and CoWoS advanced package integration. Both NPU and DRAM requirements are minimized with a standard LPDDR4 interface, leading to substantial cost savings and reduced power consumption, crucial for avoiding heavy data movements.

(2) Power Efficiency and Instant-ON from sleep mode: Unlike DRAM, which requires periodic refresh cycles to maintain data, non-volatile Flash memory in dCIM can operate flexibly either in fully-active mode for maximum bandwidth or in partially-active mode with reduced bandwidth at lower duty cycles during inference. Standby power consumption is significantly lower. Moreover, Flash memory can quickly transition from sleep mode to "instant-ON," advantageous for edge inference devices that often remain dormant

until activated for LLM inference. In contrast, DRAM typically incurs longer boot-up times to load data from Flash memory initially.

IV. In-Memory Search (IMS) Accelerator

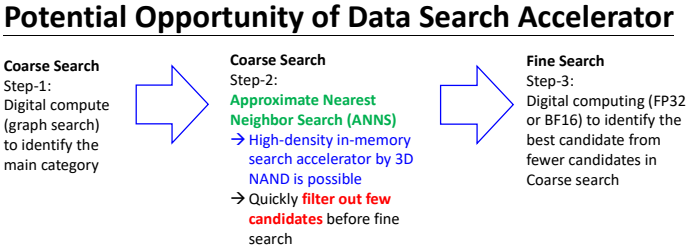
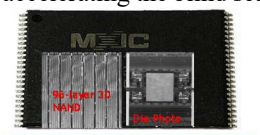


Figure 10 Potential opportunity for in-memory search (IMS) accelerator. 1st step is the coarse search with digital graph search method to find the labeled category. The second stage can be performed by IMS accelerator to quickly filter out few-percentage relevant candidates for the 3rd-step fine search with high-precision digital computing.

Data retrieval plays a crucial role in data center applications. Recently, Retrieval Augmented Generation (RAG) can augment LLM by incorporating knowledge database references to provide answers beyond their training. With the exponential growth of big data, data retrieval tasks are becoming increasingly complex and time-consuming. **Figure 10** outlines our proposed data search operation steps. Initially, the original database is transformed into a vector database for efficient searching. The search process begins with a coarse search in the first step, utilizing a graph-based algorithm that performs high-precision digital computing to identify major categories. With a highly accumulated big data within the same category, an In-Memory Search (IMS) accelerator can help in accelerating the blind search during the second step of coarse search.



Technology	MXIC 96-layer 3D NAND
Mem. density	43Gb (pSLC) for IMS
Chip I/O	1.8V; 200MHz Toggle NAND
In-Memory Search (IMS) functions	Search Bit: 128b *N (flexible) Match line: 16KB TCAM or Proximity search (Hamming distance) Max. 300Gb/s search bandwidth

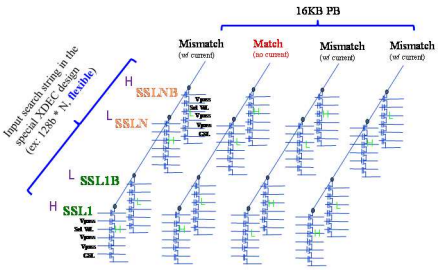


Figure 11 3D NAND in-memory search (IMS) [5] accelerator test chip, using SSL inputs for vector search. It's a 43Gb pSLC NAND for IMS device, with paired SSL inputs that can perform both exact TCAM or approximate Hamming distance computing. The parallelly operated 128 SSL inputs provides 128 times of normal pSLC read bandwidth, thus providing a search accelerator inside a 3D NAND storage.

For 3D NAND Flash IMS, there are either dCIM or aCIM design for IMS. So far we do not find suitable design of HB dCIM using 3D NAND, because all internal page buffer read bandwidth are already output to chip I/O, while the internal read bandwidth is much lower than DRAM and 3D NOR. On the other hand, we can find various semi-analog CIM techniques for IMS. A successful example is shown in **Figure 11** [5]. In our 96-layer 3D NAND test chip, we use the multiple SSL inputs as the vector inputs for search, and we can carry out the exact search TCAM or approximate TCAM (for Hamming distance). Due to the parallelly operated 128-pair SSL inputs, the effective search bandwidth is boosted by 128 times of normal page buffer read bandwidth, giving an effective search bandwidth of ~300Gb/s (per 16KB PB in a pSLC 3D NAND chip).

Figure 12 shows the Hamming distance sorter concept. We can use the 3D NAND IMS device to get the similar Hamming distance distribution of "matched" and "mismatched" dataset with software data. A tunable V_{WL} bias can be applied to get the quick sorting to filter out only a few percentage (<3%) data in the coarse search. The final fine search still goes back to conventional high-precision digital search and the accuracy is not loss.

Different from dCIM for LLM inference, the semi-analog CIM for IMS accelerator is not critical for accuracy, since it is just a coarse computing before the next step of digital computing. Error is not accumulated. In addition, we do not require ADC circuit design.

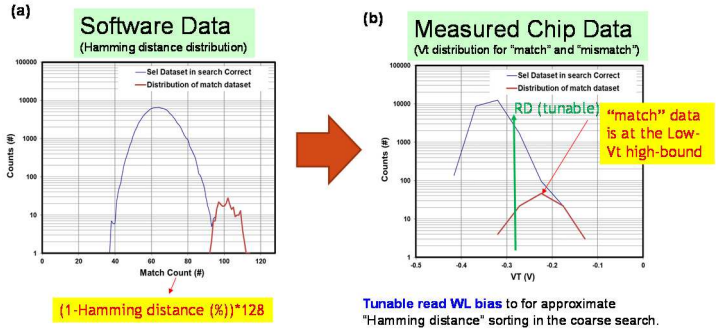


Figure 12 Hamming distance sorter concept. **(a)** For a face recognition example (VGGFace2), we truncate the vectors into 128 dimension with binary data. The matched and mismatched dataset has a clear Hamming distance discrimination, where matched data set has a smaller Hamming distance (peak around 25%), while mismatch average Hamming distance is around ~50%. **(b)** The hardware detected Hamming distance measured from our 3D NAND IMS device. The matched dataset is at the low-Vt high-bound. We can use the tunable VWL bias at read to sort the relevant data within a few percentage (<3%), and carry out fine search using high-precision digital computing. The accuracy is the same with ideal software computing results.

V. Conclusions

The growing demand for low-cost and low-power AI inference devices is expected to create numerous opportunities for custom-designed ASICs and memory solutions. We anticipate that Flash memory can be redesigned with new features to support edge inference and cloud computing applications. This paper explores two examples: the first being High-Bandwidth digital Computing in Memory (HB dCIM) using 3D NOR for accelerating Large Language Model (LLM) inference, and the second being a 3D NAND In-Memory Search (IMS) accelerator for approximate coarse search in data retrieval tasks. We are optimistic about the potential for Flash memory to effectively contribute to AI computing, complementing processors and DRAM to optimize system power and cost. Continued innovation is expected to further enhance Flash memory's capabilities for AI applications.

References: **[1]** H.T. Lue, et al, "3D AND: A 3D Stackable Flash Memory Architecture to realize High-Density and Fast-Read 3D NOR Flash and Storage-Class Memory ", IEDM Symposia, TFS1-1, 2023. **[2]** Euicheol Lim, et al, "Cost effective LLM accelerator using PIM technology", VLSI 2024, JFS6.2. **[3]** Ming-Liang Wei, Hang-Ting Lue, et al, "Analog Computing in Memory (CIM) Technique for General Matrix Multiplication (GEMM) to Support Deep Neural Network (DNN) and Cosine Similarity Search Computing using 3D AND-type NOR Flash Devices", IEDM, 2022, Session33-3, pp. 787-790. **[4]** Yu-Der Chih, et al, "A 89 TOPS/W and 16.3 TOPS/mm2 All Digital SRAM Based Full Precision Compute-In-Memory in 22nm for Machine-Learning Edge Applications", ISSCC 2021, Session 16.4. **[5]** Chih-Chang Hsieh, Hang-Ting Lue, et al, "Chip Demonstration of a High-Density (43Gb) and High-Search-Bandwidth (300Gb/s) 3D NAND Based In-Memory Search Accelerator for Ternary Content Addressable Memory (TCAM) and Proximity Search of Hamming Distance", VLSI 2023, Session T15.1.

AiMX: Accelerator-in Memory Based Accelerator for Cost-effective Large Language Model Inference (Invited)

Haerang Choi[1], Guhyun Kim[1], Woojae Shin[1], Jongsoon Won[1], Changhyun Kim[1], Hyunha Joo[1], Byeongju An[1], Gyeongcheol Shin[1], Jeongbin Kim[1], Dayeon Yun[1], Jaehan Park[1], Yosub Song[1], Byeongsu Yang[1], Hyeongdeok Lee[1], Seungyeong Park[1], Wonjun Lee[1], Seonghun Kim[1], Yonghoon Park[1], Yousub Jung[1], Ilkon Kim[1], Gi-Ho Park[2], and Euicheol Lim[1]

[1]SK hynix Inc., South Korea, email: haerang.choi@sk.com

[2]Sejong University, South Korea

***Abstract*—**We presented an Accelerator-in-Memory (AiM) device and AiM-based LLM inference acceleration system. LLM inference can be divided into prompt phase and response phase. Considering the characteristics of LLM inference, we proposed a disaggregated inference system where the prompt phase is executed on high-throughput GPUs or NPUs, and the response phase is executed on AiM. Using AiM for single GEMV operations can ideally achieve up to 16 times the performance. The measured performance of the prototype AiM-based Accelerator is 1.7 times higher than that of a comparable GPU, and the expected performance at the highest data rate is 11.7 times higher.

I. INTRODUCTION

With the popularization of Large Language Models (LLMs), the performance of LLM inference has become more important than training performance. Consequently, memory manufacturers are dedicating substantial resources to improve the performance of acceleration systems specialized for LLM inference [1, 2, 3]. The reason why the approach of memory manufacturers is effective is as follows.

LLM inference can be divided into the prompt phase, which processes the input sequence, and the response phase, which generates a response corresponding to the input [1]. As shown in Fig. 1, the prompt phase converts the input sequence into a matrix and processes it at once, while the response phase repeats the process of generating a new token by receiving one token as input. Therefore, the performance of LLM inference is determined by the repeatedly executed response phase, and the performance metric of LLM inference is token/s, which refers to the number of response phases executed per unit time.

The performance of the response phase is determined by the performance of the DRAM where the data is stored. This is because every time a token is generated, all the data of the model must be transferred from the DRAM to the processing core. We aim to address this issue where the computational performance of the response phase is limited by DRAM bandwidth (BW) and the problem of consuming energy by redundantly reading data every time a token is generated. In this paper, we introduce our developed Accelerator-in-Memory (AiM) device, and AiM-based accelerator (AiMX) and outline the issues that need to be investigated further.

II. MOTIVATION

A. Data Movement Overheads in LLM

The 'decoder only' architecture is most commonly used in LLMs, and such models are created by stacking decoder blocks repeatedly, as shown in Fig. 2 [4]. The figure illustrates the basic unit block that constitutes GPT, consisting of an Attention layer that extracts context information from the current input and previous response results (history), and a Feed Forward Network (FFN) layer that computes the extracted information with pre-learned weight matrices.

In the response phase, the input token is transformed into a single vector, which is then processed by the Attention layer's history matrix. The resulting vector is subsequently processed by the FFN's weight matrix. In General Matrix-Vector multiplication (GEMV), matrix data is not reused, so the execution time is mainly determined by the time required to read the matrix data from DRAM. It was found that when executing GPT-3 175B on V100 GPUs, the GEMV execution time accounted for approximately 90% of the response phase. This is because every time the response phase is executed, an enormous data movement of at least 350GB (datatype: FP16) repeatedly occurs for the GEMV operation, and the execution time is bounded by DRAM bandwidth.

B. GDDR6-AiM Characteristics

AiM can improve latency and energy overhead caused by off-chip data movement because it performs computations within DRAM without any off-chip data movement. We developed the GDDR6-AiM based on GDDR6 [5, 6]. As shown in Fig. 3, we implement a Processing-unit (PU) to each BANK and a Global buffer in the peripheral area to store the input vector. The main performance metrics are shown in Table 1. AiM performance is equivalent to the existing DRAM BW multiplied by the number of BANKs, and it is indicated as internal bandwidth. AiM is an optimal architecture for GEMV operations because all PUs share and compute the input in Global buffer.

C. *Key Metric of DRAM for LLM Inference*

In the design phase of an LLM inference system, it is crucial to determine the memory capacity considering the data size during inference execution and to select the memory type based on the performance requirements of the response phase. This is because the latency in reading model data determines the performance of the response phase in LLM inference. To facilitate the comparison of memory performance, we define the value obtained by dividing DRAM bandwidth (BW) by DRAM capacity (GB) as a key metric.

BW per GB allows us to easily compare which DRAM can meet the target performance for LLMs, as it corresponds to the meaning of token/s. Taking the reciprocal of this metric gives the ratio of DRAM capacity to DRAM bandwidth, representing the read latency when DRAM is fully utilized during the LLM inference operations. For example, a card A with a BW of 933GB/s and a capacity of 24GB has a BW per GB of approximately 39. Meanwhile, card B with a BW of 3.3TB/s and a capacity of 80GB has a metric of approximately 41. Therefore, if the model size is small, it can be easily determined that configuring the system with card A is more cost-effective.

III. AiM-based Accelerator

Noting that accelerators with high DRAM BW are useful for accelerating LLM inference, we developed a prototype AiMX card, as shown in Fig. 4. To provide scalability in performance and capacity of the card while connecting the Host and AiM via PCIe, we implemented an IP called AiM Control Hub in the FPGA chip, as illustrated in Fig. 5. Considering large models, card-to-card communication is supported via QSFP. Due to IO speed limitation of FPGA, AiM in our prototype card operates at 2.67Gbps, which is lower than the maximum data rate of GDDR6-AiM, 16Gbps. The detailed specifications of GDDR6-AiM are shown in Table 2.

Fig. 6 shows the software stack architecture to support AiMX operations. AiMX SDK enables AiMX working with existing frameworks with minimal modifications. Notably, to expand the ecosystem, we also provide a software emulator of the AiMX card.

Considering the different computational characteristics of the prompt phase and response phase, we proposed a disaggregated inference system where matrix operations in the prompt phase are executed on GPUs, and GEMV operations in the response phase are executed on AiMX cards. We built a prototype inference system for data center as shown in Fig. 7.

For a comparative analysis of the GPU-only system and our proposed disaggregated inference system solution, we executed the prompt phase on a GPU, then executed the response phase separately on both GPU and AiMX card. Fig.8 shows the response phase performance comparison of two 24GB GPU cards (BW 933GB/s) and two 16GB AiMX cards (BW 170GB/s) on the OPT-13B model. The measured performance of the AiMX prototype card is 1.7 times higher than that of the GPU, and if we consider the maximum speed of GDDR6-AiM, it is expected we can achieve 11.7 times higher performance in the real system.

IV. Discussions for On-device AI

We are currently developing the next version of AiM and AiMX, targeting on-device AI that executes LLMs standalone in mobile systems. In mobile systems, there is a trade-off relationship among the LLM inference performance, battery time, power budget, and thermal budget. Although AiM's high energy efficiency improves both LLM performance and battery time, executing the same computations in parallel in a shorter time can increase power consumption and require consideration of the thermal budget. Besides innovating individual technologies improving power and thermal performance, it is also important to find a system-level optimization that balances key metrics of on-device AI, considering AiM computation as a new resource. Thus, improving on-device AI performance is not achieved by enhancing a single device of technology alone, but by integrated optimizing technologies from various aspects of software and hardware components of the system together.

V. Conclusions

We have developed an AiM-based accelerator to address the limitation of response phase performance in LLM inference being constrained by DRAM BW. Considering the characteristics of computation in LLM inference, we proposed a disaggregated system having GPU and AiMX as a computing engine for difference phases of LLM inference. This system uses different type of accelerator cards for each phase: GPU as high throughput cards for the prompt phase and AiMX as high bandwidth cards for the response phase. Additionally, we designed the software stack for the disaggregated system as an extension of existing frameworks to enhance compatibility. As a result, prototype AiMX can deliver the measured performance with 1.7 times higher than that of a comparable GPU, and the expected performance at the highest data rate is 11.7 times higher.

References

[1] Y. Kwon *et al.*, "Memory-Centric Computing with SK Hynix's Domain-Specific Memory," *2023 IEEE Hot Chips 35 Symposium (HCS)*, Palo Alto, CA, USA, 2023, pp. 1-26.

[2] SK hynix Inc., "Accelerator-in-Memory (AiM): 4Gb/Channel GDDR6-based PIM from SK hynix." YouTube, Mar. 7, 2022, [Online]. Available: https://youtu.be/3LbvwrJFYoA?t=287, Accessed: 2024.

[3] B. Kim *et al.*, "The Breakthrough Memory Solutions for Improved Performance on LLM Inference," in *IEEE Micro*, vol. 44, no. 3, pp. 40-48, May-June 2024.

[4] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, Jun. 11, 2018, [Online]. Available: https://openai.com/index/language-unsupervised, Accessed: 2024.

[5] S. Lee *et al.*, "A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications," *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2022, pp. 1-3

[6] D. Kwon *et al.*, "A 1ynm 1.25V 8Gb 16Gb/s/Pin GDDR6-Based Accelerator-in-Memory Supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep Learning Application," in *IEEE Journal of Solid-State Circuits*, vol. 58, no. 1, pp. 291-302, Jan. 2023.

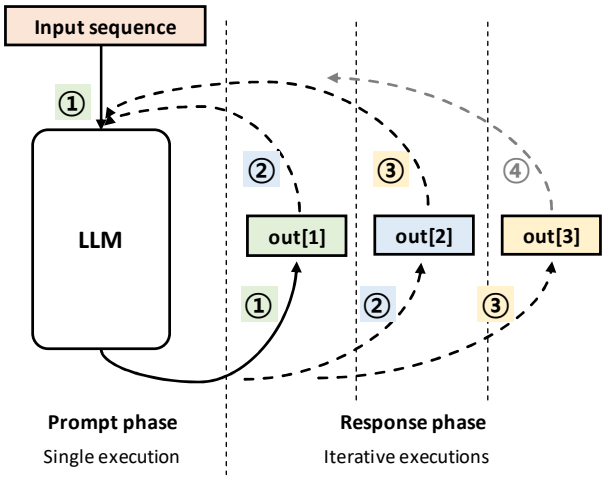


Fig. 1. Two phases of LLM inference: prompt and response phase.

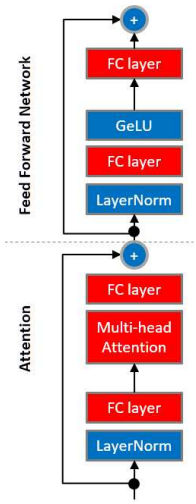


Fig. 2. GPT-2 decoder block architecture. The GPT model contains N decoder blocks [4].

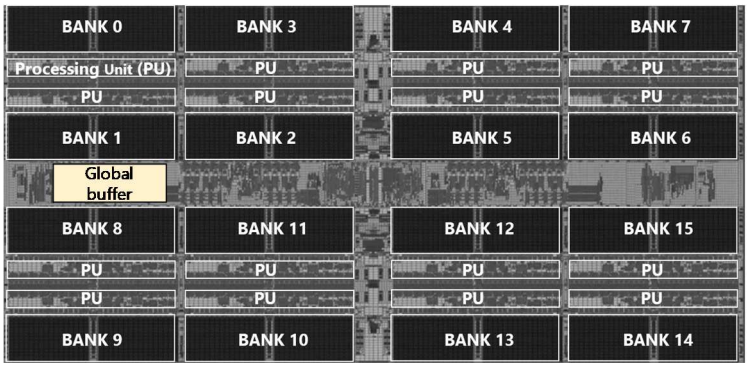


Fig. 3. GDDR6-AiM floor plan [5].

GDDR6-AiM* (per PKG)	
DRAM Type	GDDR6
Process Technology	1y
Memory Density	1GB
Organization	X32
IO Data rate	16 Gbs/pin (@1.25V)
(External) Bandwidth**	64 GB/s
Processing Unit (PU)	Total 32PU
Compute Throughput**	1 TFLOPS
Internal Bandwidth**	1 TB/s
Numeric Precision	BF16
Activation Function support***	Sigmoid, tanh, GELU, ReLU, Leaky ReLU, ...

Table 1. GDDR6-AiM specifications [5].

Fig. 4. AiMX prototype card.

Host Interface		PCIe Gen3 x8x8 (bifurcated)
Form Factor		FHFL (A100/A30 compatible)
Configuration		2 FPGA* x 16 AiM package
AiM	Capacity	16 GB
	Bandwidth	170 GB/s (@2.67Gbps**)
Scale out		chip2chip interconnect (QSFP28)
Thermal Cooling		Passive

Table 2. AiMX prototype card specifications [1].

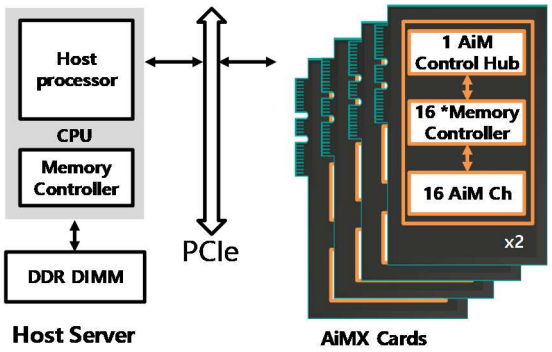


Fig. 5. Block diagram of LLM inference system using AiMX cards.

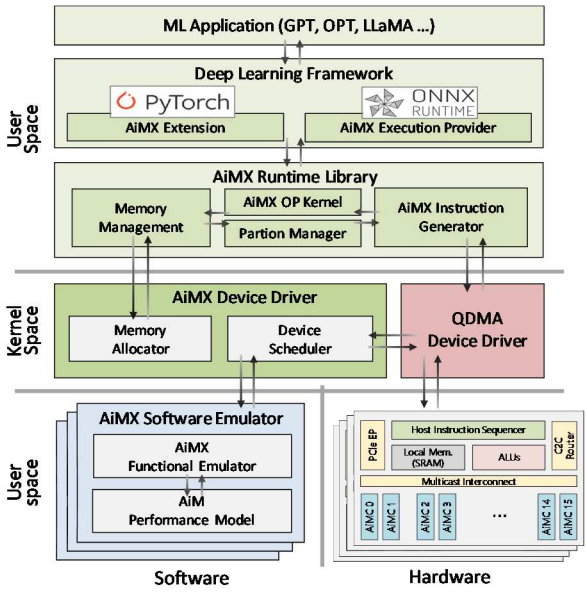


Fig. 6. AiM software stack architecture [1].

Fig. 7. Prototype inference system using GPU and AiMX cards.

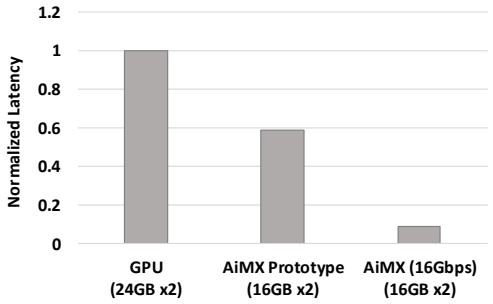


Fig. 8. Response latency comparison for OPT-13B inference.

Heterogeneous Embedded Neural Processing Units Utilizing PCM-based Analog In-Memory Computing

I. Boybat[1,*], T. Boesch[2,*], M. Allegra[3], M. Baldo[3], J.J. Bertolini-Agnoletto[3], G. W. Burr[4], A. Buschini[5], A. Cabrini[6], E. Calvetti[3], C. Cappetta[5], F. Conti[7], E. Ferro[1], E. Franchi Scarselli[7], A. Garofalo[8], F. Girardi[5], G. Islamoglu[8], V. P. Jonnalagadda[1], G. Karunaratne[1], C. Lammie[1], M. Le Gallo[1], C. Li[9], R. Massa[3], A. C. Ornstein[5], H. Pang[8], M. Pasotti[3], B. Rajendran[9], A. Redaelli[3], I. Sanli[1], W. A. Simon[1], A. Singh[1], S.-P. Singh[10], G. Urlini[3], A. Vasilopoulos[1], R. Zurla[11], G. Desoli[5,*], and A. Sebastian[1,*]

[1]IBM Research - Zurich, Switzerland [2]STMicroelectronics, Geneva, Switzerland
[3]STMicroelectronics, Agrate, Italy [4]IBM Research-Almaden, San Jose, CA, USA [5]STMicroelectronics, Cornaredo, Italy
[6]University of Pavia, Italy [7]University of Bologna, Italy [8]ETH Zurich, Switzerland
[9]King's College London, UK [10]STMicroelectronics, Noida, India [11]STMicroelectroics, Pavia, Italy
*email: ibo@zurich.ibm.com, thomas.boesch@st.com, giuseppe.desoli@st.com, ase@zurich.ibm.com

***Abstract*— We propose an embedded Neural Processing Unit (NPU) architecture for deep learning inference to address the stringent energy, area, and cost requirements of edge AI. This heterogeneous architecture integrates a variety of digital and analog accelerator nodes to cater to diverse operation types and precision requirements. To achieve high energy efficiency while maintaining substantial non-volatile on-chip weight capacity, we utilize Analog In-Memory Computing (AIMC) tiles based on Phase-Change Memory (PCM) for Matrix-Vector Multiplications (MVMs). Additionally, a digital data path and a programmable software cluster facilitate end-to-end inference across multiple precision levels. The NPU is projected to deliver competitive throughput for transformer Neural Networks (NNs), rivaling high-end System-on-Chips (SoCs) for mobile devices and edge accelerators fabricated at more advanced technology nodes.**

I. Introduction

The data intensive nature and highly parallel compute requirements of AI models lead to specialized Neural Processing Units (NPUs) being integrated on to System-on-Chip (SoC) devices for AI edge applications (Fig. 1). Typically, model weights are stored in external memory or on-chip Non-Volatile Memory (NVM). In-Memory Computing (IMC) can further reduce data movement by performing Matrix-Vector Multiplication (MVM) operations directly within customized memory arrays. Among the various forms of IMC, Digital IMC (DIMC) performs scalar multiplications of preloaded weights and activations close to the memory cells and performs the partial sum accumulation using digital adder trees [1]. On the other hand, Analog IMC (AIMC) exploits the inherent row parallelism of memory arrays by executing scalar multiplications and partial sum accumulations in the analog domain [2]. In comparison to DIMC, this approach offers the potential for higher density and lower power consumption. The most mature form of AIMC is based on SRAM technology [3], with multi-bit weights encoded across multiple devices. However, AIMC based on embedded NVM cells presents a promising prospect for NPUs, offering increased on-chip weight density. Among various NVM technologies for AIMC, Phase-Change Memory (PCM) cells are particularly attractive. The analog storage capability and Back-End-Of-the-Line implementation enables high on-chip weight capacity (Figs. 2,3), while offering a scalability path to more advanced nodes. As embedded memory, PCM has already been proven in industrial settings [4,5]. Moreover, recent multi-tile PCM-based AIMC chips showcase the energy efficiency of MVM operations with sufficient computational precision [6,7].

An NPU architecture for edge applications needs to offer a high degree of flexibility and versatility. It should support various model mappings based on model weight size, required latency/throughput, and the available power budget. Furthermore, seamlessly integrating both analog and digital compute nodes with a high bandwidth communication fabric is crucial for effectively managing end-to-end tasks.

II. Heterogeneous NPU

The NPU architecture comprises various nodes, in addition to the PCM-based AIMC tiles (Fig. 4). A Digital Processing Unit (DPU) serves as a programmable data path and can handle a large selection of operators including convolutions, pooling, arithmetic, and simple activation functions with 8-bit or 16-bit fixed-point precision. A software-programmable RISC-V cluster with shared scratchpad memory and a specialized tensor product data-path with fused multiply-accumulate units, complements the DPU in performing accuracy-critical operations in FP16 precision [8]. The DPU and the RISC-V co-processor units constitute the digital accelerator nodes of the NPU. The Local Storage Unit (LSU) provides data manipulation operations and local SRAM-based storage. These nodes are connected through a 64-bit wide 2D mesh interconnect [9] with bit-parallel communication. A digital

wrapper is present at each node to interface the interconnect and to perform control tasks (Fig. 5). Borderguard circuits handle data routing between the interconnect and node FIFOs which are controlled by borderguard controllers operating on a basic instruction set. A stall controller regulates data communication based on data availability. A RISC-V-based local controller is responsible for the overall control flow within each node. An NPU configuration with 20 nodes is shown in Fig. 4, with an estimated area of ~30 mm^2 on ST's 28nm FD-SOI technology [4], operating at 500MHz with an estimated average power dissipation below 1W. This architecture is configurable at design time, allowing to scale effectively with varying computational demands (Fig. 6).

III. PCM BASED ON GE-GST

A Ge-GST-based PCM with a wall device architecture [4] is adopted, where an n-type MOSFET selector ensures precise control over the current flowing through the device during programming, thereby enabling analog weight storage. Tight conductance distributions are achieved for various conductance targets using a feedback-driven program-and-verify scheme (Fig. 7). Fig. 8 shows the drift and read noise behavior across more than 280 devices, programmed to various conductance states. While the state-dependent drift and read noise characteristics can be detrimental for inference accuracies over time scales exceeding 1 hour, this can be mitigated by using optimized device structures that show lower drift and read noise as depicted in Fig. 9.

IV. FLEXIBLE ANALOG TILE ARCHITECTURE

The AIMC tile supports a 512 x 512 MVM. It is physically organized in 4x4 local arrays of unit cells containing a total of 1,024 word-lines and 4,096 bit-lines (Fig. 10). Each unit cell contains up to 16 PCM devices. Positive and negative inputs are applied through separate word lines. The input precision is flexible, supporting both signed values (1-8 bits) and unsigned values (1-7 bits). The bit lines are connected to current-controlled oscillator-based ADCs [10] with a flexible switch mechanism (Fig. 11). Higher precision for weight representation can be achieved by connecting multiple bit-lines to an ADC. Conversely, connecting fewer bit-lines results in a larger weight capacity. Fig. 12 shows that the energy efficiency of the AIMC tile is expected to exceed recent macros [3,6] with a component-wise energy breakdown provided in Fig. 13. A compact digital post-processing unit based on fixed-point arithmetic performs affine scaling functions on the ADC output to address circuit mismatches and PCM non-idealities such as drift [11]. In addition, the unit supports partial sum and residual addition, batch-normalization, ReLU activation, and scaling of the AIMC tile outputs down to 8-bit precision.

V. LANGUAGE PROCESSING ON NPU

The NPU architecture supports a wide range of Neural Network (NN) models, including CNNs, LSTMs, and transformers. We will focus on the latter and study MobileBERT, which has 24 encoder layers with 4 attention heads. It requires over 20 million weight parameters trained for a question-answering task on the SQUAD v1.1 dataset. One sample mapping of the model to an NPU with 20 nodes is shown in Fig. 14.

Fig. 15 compares MobileBERT model throughput performance across a spectrum of systems from high-end mobile and laptop devices to low-power and low-cost edge devices. The NPU with 3 digital accelerator nodes is comparable in throughput to the Google Pixel 6 with EdgeTPU™ and provides better throughput than the ARM Ethos-U65™. Moreover, even with an average power budget below 1W and cost-effective 28nm technology, the NPU throughput with 5 digital accelerators approaches half that of some advanced high-end SoCs for laptop and mobile devices on advanced technology nodes for this benchmark. Due to the limited publicly available information on the energy and area efficiency of these devices, we focus solely on the throughput comparison. However, significant energy benefits are anticipated for the NPU, as all model weights are stored on the AIMC tiles, eliminating the need for off-chip weight loading during the MobileBERT runtime.

VI. ACCURACY STUDIES

The IBM Analog Hardware Acceleration Kit [15] is used to perform hardware-aware re-training of a floating-point pre-trained MobileBERT model, which achieves an F1-score of 90.02 on SQUAD v1.1 dataset [16]. The re-training was done for 15 epochs using additive noise injection on weights with a standard deviation of 6.7% using an ADAM optimizer and a cross-entropy loss with an initial learning rate of $5x10^{-5}$. During training, weights were clipped 3 standard deviations from the mean and an input/output quantization (8 bits) was performed. During inference, statistical PCM models for programming noise, drift, and read noise data extracted from device measurements were used. A global drift compensation mechanism is applied during inference [15]. Fig. 16 shows the temporal dependence of the F1 score over a period of one month, showing only marginal drop from the floating-point baseline with the optimized PCM device structure. The heterogeneity of the NPU architecture can be leveraged for higher F1 scores by assigning selected NN layers to higher precision compute resources. This is demonstrated in Fig. 17 where the designated layers are computed with floating-point precision without any further re-training.

VII. CONCLUSION

We propose a heterogeneous embedded NPU architecture for edge AI inference applications, where mix of analog and digital accelerator nodes are interconnected through a 2D mesh. The Ge-GST PCM-based AIMC tiles offer dense analog weight storage while the digital accelerator nodes support both low precision fixed-point and floating-point precision, allowing for versatile operation. An NPU with a ~30mm^2 area footprint on 28nm ST's FD-SOI technology is expected to provide sufficient on-chip weight storage capacity for all the encoder weights of MobileBERT, with an average power dissipation below 1W. This NPU configuration is estimated to deliver competitive inference throughput for transformer NNs, approaching high-end SoCs for mobile devices.

ACKNOWLEDGMENT

We would like to thank all NeuroSoC partners (https://neurosoc.eu), in particular T. Antonakopoulos, L. Benini, F. Brutin, F. Buckley, N. Chawla, U. Egger, T. Jang, B. Kersting, A. Petropoulos, J. Quevremont for technical contributions. We would also like to thank R. Haas, A. Curioni, H. Tsai, and V. Narayanan for management support. This work was supported by European Union's Horizon Europe Research and Innovation Program (Grant 101070634), and Swiss State Secretariat for Education, Research and Innovation (SERI) (Grant 23.00205).

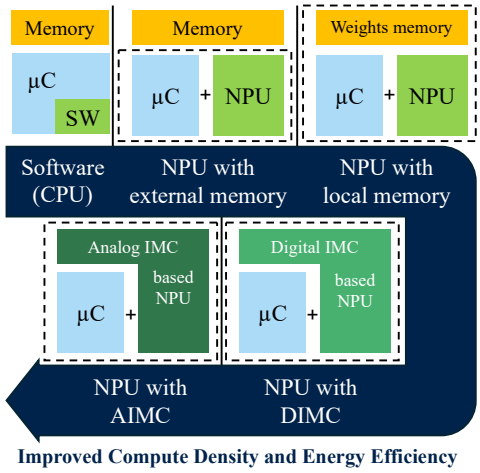


Fig. 1. Evolution of the Neural Processing Units (NPUs) towards digital and analog in-memory computing.

REFERENCES

[1] G. Desoli *et al.*, *IEEE ISSCC*, 2023, pp. 260-262.
[2] A. Sebastian *et al., Nat. Nanotechnol.* **15**, 529–544 (2020).
[3] H. Jia *et al.*, *IEEE Journal of Solid-State Circuits*, 2022, vol. 57, no. 1, pp. 198-211.
[4] F. Arnaud et al., *IEEE IEDM,*, 2018, pp. 18.4.1-18.4.4.
[5] F. Arnaud *et al.*, *IEEE IEDM*, 2020, pp. 24.2.1-24.2.4.
[6] M. Le Gallo *et al., Nat Electron* 2023, **6**, 680–693.
[7] S. Ambrogio *et al., Nature, 2023,* **620**, 768–775.
[8] Y. Tortorella *et al.*, *DATE*, 2022, pp. 1099-1102.
[9] S. Jain *et al.*, *IEEE Trans. VLSI*, 2023, vol. 31, no. 1, pp. 114-127.
[10] R. Khaddam-Aljameh *et al.*, *IEEE JSSC*, 2022, vol. 57, no. 4, pp. 1027-1038.
[11] E. Ferro *et al.*, *ISCAS*, 2024. 1-5.
[12] MLPerf Inference: Mobile Benchmark Suite Results v4.0, accessed on July 5, 2024, https://mlcommons.org/benchmarks/inference-mobile/
[13] MobileBERT-EdgeTPU Github Repository, accessed on July 5, 2024, https://www.kaggle.com/models/google/mobilebert-edgetpu/tfLite/xs
[14] TSMC logic technology, accessed on July 5, 2024, https://www.tsmc.com/english/dedicatedFoundry/technology/logic
[15] Le Gallo *et al.*, *APL Mach. Learn.,* 2023, 1, 041102
[16] Hugging Face Repository, https://huggingface.co/google/mobilebert-uncased
[17] Ethos Toolchain Repository, accessed on July 5, 2024, https://git.mlplatform.org/ml/ethos-u/ethos-u-vela.git/tree/ethosu/config_files/Arm/vela.ini

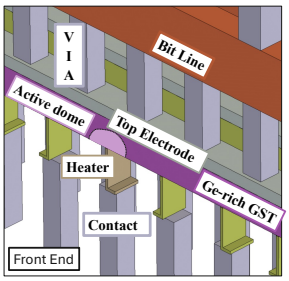


Fig. 2. PCM integration in ST 28nm FD-SOI technology.

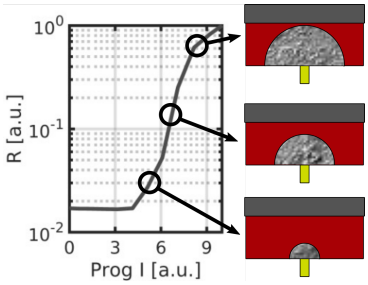


Fig. 3. The analog storage capability of PCM through creation of different amorphous dome dimensions.

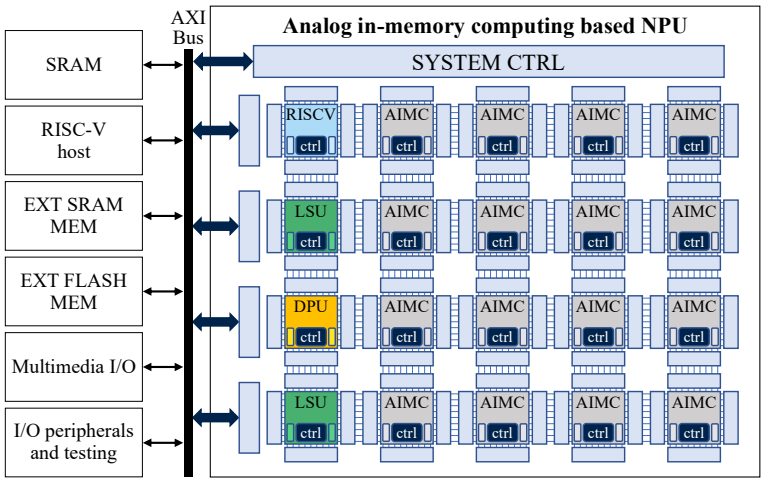


Fig. 4. The heterogeneous NPU architecture consisting of 20 nodes is shown. The nodes are interconnected with a 2D mesh-based communication fabric. The NPU is connected to the other system components through the AXI bus.

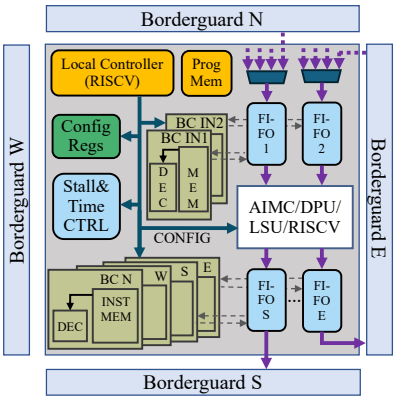


Fig. 5. Each NPU node consists of an accelerator (analog/digital) or a memory unit, integrated into a digital wrapper.

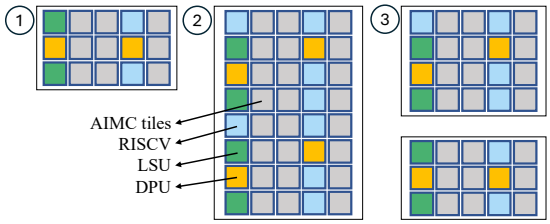


Fig. 6. The architecture is parametric and scalable at design-time, allowing to accommodate various target applications and use cases (①, ②, ③). Moreover, larger NPUs (②) or multiple NPUs (③) can be adopted in larger systems.

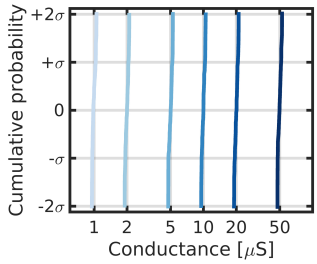


Fig. 7. Conductance distributions shown for six target states, obtained over 45 devices per state after 20 program-and-verify iterations with a precision of 5% around the programmed value.

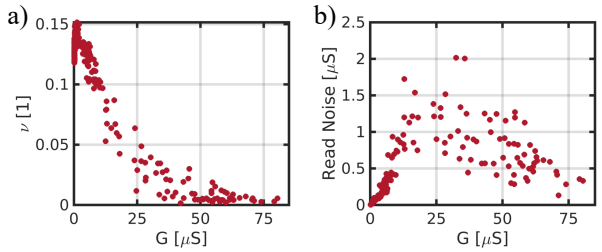


Fig. 8. The drift coefficient and read noise as a function of the programmed conductance across 288 devices. Drift follows the relation $G(t) = G(t_0)(t/t_0)^{-\nu}$, where G(t) and $G(t_0)$ denotes the conductances at time t and t_0, respectively and ν is the drift coefficient.

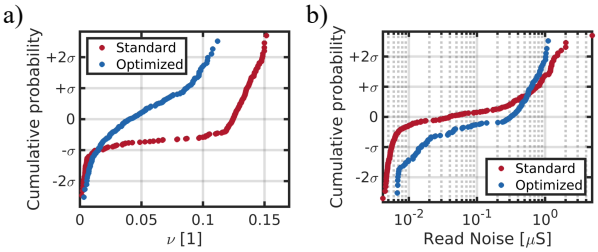


Fig. 9. Drift and read noise distributions for 288 standard and 170 optimized PCM devices.

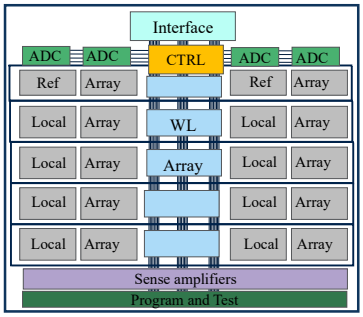
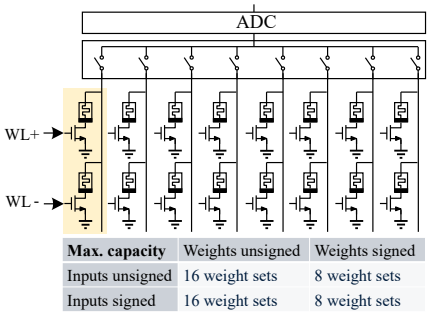


Fig. 10. PCM-based AIMC tile is organized in 16 local arrays. The maximum signed weight storage capacity per tile is 2.1M.



Max. capacity	Weights unsigned	Weights signed
Inputs unsigned	16 weight sets	8 weight sets
Inputs signed	16 weight sets	8 weight sets

Fig. 11. A switching mechanism supports ADC multiplexing across PCM devices, or alternatively, multiple PCM devices can be combined for higher compute precision.

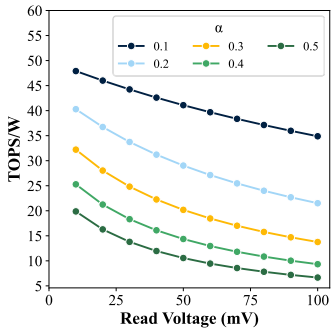


Fig. 12. Energy efficiency of the AIMC tile with 8-bit signed input and signed analog weights. α is the input and weight values relative to the maximum achievable input and weight values.

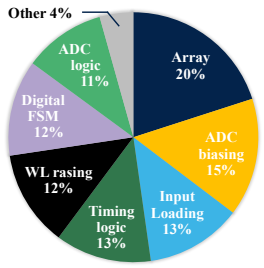


Fig. 13. Energy breakdown of the components, shown for read voltage of 30 mV and α = 0.2 for signed inputs and signed weights.

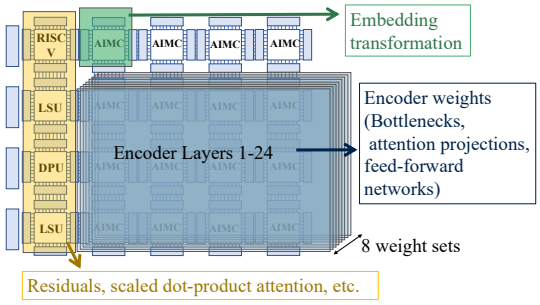


Fig. 14. The MobileBERT operator mapping strategy shown here utilizes AIMC tiles for MVMs, where each of the 8 weight sets represent three encoder layer weights. Off-chip memory is used to store the embeddings.

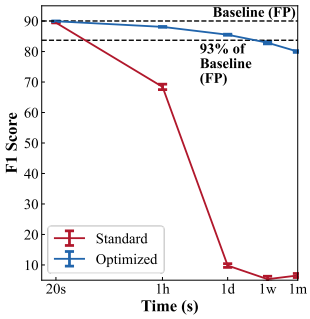


Fig. 16. Mean and standard deviation of MobileBERT F1 score over 10 inference runs. MobileBERT accuracy target is defined as 93% of the FP implementation by MLPerf [12].

Devices	Domain	Node [nm]	Freq.[GHz]	Inf/s	Inf/s [scaled]**
ARM Ethos-U65™ *	Edge	16	1	6	4
NPU, 2 digital accelerators	Edge	28	0.5	15.3	15.3
NPU, 3 digital accelerators	Edge	28	0.5	29.3	29.3
NPU, 5 digital accelerators	Edge	28	0.5	51.6	51.6
Google Pixel 6, EdgeTPU™ [13]	Phones (High-end SoC)	5	Up to 2.8	66.7	29.7
Qualcomm Snapdragon X Elite, Hexagon™ [12]	Laptop (High-end SoC)	4	Up to 3.8	298.9	121.2
Exynos 2400, Samsung NPU [12]	Phones (High-end SoC)	4	Up to 3.2	317.1	128.5
Qualcomm Snapdragon 8 Gen 3, Hexagon™ [12]	Phones (High-end SoC)	4	Up to 3	433.3	175.7

* Using MobileBERT-EdgeTPU-XS-quant model obtained from [13]
** Technology scaling done in line with ST process insights and [14]

Fig. 15. MobileBERT throughput performance (sequence length 384) comparing edge platforms (targeting low energy consumption) with high-end SoCs (comprising significantly advanced computation capability e.g., multi-core CPUs, GPUs, accelerators, high-bandwidth memory interfaces such as LPDDR5, and large system cache). Ethos-U65 figures obtained from the Ethos toolchain selecting the 512 MAC/cycle NPU variant paired with 512KB SRAM in the 'Ethos_U65_High_End' system configuration [17].

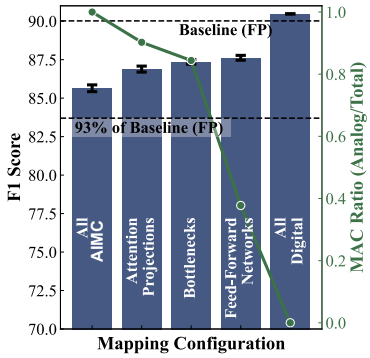


Fig. 17. Mapping studies (10 inference runs, 1 day inference time) reveal the impact of computing each specific MVM-layer-type at FP precision. Other layer-types remain on AIMC tiles.