

HINT: A Hybrid SRAM–MRAM Compute-In-Memory with INput-aware Skipping SAR-ADC for Energy Efficient Ternary LLMs

Jaebeom Park, Seung-Eon Hwang, and Jongsun Park

School of Electrical Engineering, Korea University

Seoul, Republic of Korea

{parkjb3123, eonion56, jongsun}@korea.ac.kr

Abstract—Although large language models (LLMs) show remarkable performance in natural language processing tasks, their deployment on resource-constrained devices remains challenging due to a substantial memory footprint and high-energy consumption. To address these challenges, low-bit and ternary quantization reduce the model size, while hardware approaches such as compute-in-memory (CIM) alleviate the overhead of external memory accesses. However, billions of parameters of LLMs still cause significant data movement, and existing ternary CIM suffers from a low-density bitcell as well as accuracy degradation due to cut-off analog-to-digital converters (ADCs). In this paper, we propose HINT, a CIM architecture incorporating two energy efficient techniques. First, hybrid ternary bitcell leverages the reliability of SRAM and the high-density of MRAM, reducing area and energy overhead. Second, input-aware skipping SAR-ADC exploits input sparsity to skip unnecessary conversion cycles without sacrificing accuracy. On BitNet b1.58 (700M), compared to SRAM-based and eDRAM-based CIM baselines, HINT improves bitcell density by 1.85× and achieves up to 2.67× higher energy efficiency, respectively. By skipping up to 21% of conversion cycles, the proposed ADC improves energy efficiency up to 1.27× while maintaining model accuracy.

Keywords—*Compute-In-Memory (CIM), Large Language Models (LLMs), SAR-ADC, Ternary LLMs, Hybrid, Energy Efficient*

I. INTRODUCTION

Large language models (LLMs) have achieved remarkable performance across diverse natural language processing (NLP) tasks, which has been accompanied by a rapid increase in the number of model parameters and training data [1]–[3]. This growth in parameters considerably increases memory and computational costs as it requires more frequent data transfers and arithmetic operations between memory and processing units. For instance, the smallest LLaMA-7B released in 2023 outperforms the largest GPT-2-1.5B released in 2019, yet LLaMA-7B requires about 4.6× more memory than GPT-2-1.5B [4]. The increasing demand for resources highlights the necessity of compression techniques that can reduce data transfers as well as arithmetic operations. Among various compression approaches, quantization has gained significant attention due to its effectiveness in reducing memory access and computational complexity [5]–[7]. For instance, BitNet b1.58 employs ternary quantization $\{+1, 0, -1\}$ for each model parameter, significantly reducing storage and computation requirements [8]. Despite the storage reductions using ternary quantization, the parameter count of LLMs still remains in billions. Thus, the amount of data transfer continues to be a dominant bottleneck for the deployment of LLMs on mobile and edge devices.

To alleviate the data transfer burden, compute-in-memory (CIM) integrates computation logic within memory macros, enabling multiply-accumulate (MAC) operations. CIM efficiently reduces data movement between memory and the processing units, significantly improving energy efficiencies, particularly in deep learning applications. However, deploying prior CIM architectures for ternary LLMs acceleration on mobile and edge devices remains limited due to the following two critical reasons. First, static random-access memory (SRAM)-based bitcells for ternary computation require 16T to 19T [9], [10] as they employ two SRAM cells. For instance, even with ternary quantization, LLaMA-3B requires 2.22 GB of storage, meaning that billions of model parameters should be uploaded to SRAM. These large SRAM-based bitcells for ternary computation become a huge area and leakage burden for the overall CIM architecture system. To reduce the area burden, an embedded dynamic random-access memory (eDRAM)-based bitcell [11] for ternary computation has been proposed, but the substantial refresh energy of eDRAM further exacerbates the energy consumption. Hence, a compact and energy efficient bitcell for ternary LLMs computation is highly required. Second, another important module in CIM is analog-to-digital converters (ADCs), as it takes a dominant power and area in CIM. To reduce the ADC design cost, cut-off techniques [12]–[15] are commonly adopted in CIM design. However, as ternary LLMs produce MAC distributions with rare but influential outliers, simply applying the cut-off techniques leads to severe accuracy degradation. Therefore, it is essential to design a low-cost ADC capable of supporting full-range conversion.

In this paper, we propose HINT, a CIM architecture based on a hybrid ternary bitcell (HTC) and an input-aware skipping SAR-ADC (IAS-ADC). The HTC combines SRAM and magnetic random-access memory (MRAM) to increase storage density. It efficiently stores ternary weights, increasing bitcell density as well as improving energy efficiencies for inference. IAS-ADC exploits input sparsity to skip conversions, while supporting full-range conversion of ternary LLMs without compromising accuracy. IAS-ADC achieves substantial energy efficiency improvement during inference.

The rest of the paper is organized as follows: Section II provides the background and related work. Section III introduces our proposed techniques, including the array composed of a hybrid ternary bitcell, the description of the MAC operation, and the operation of the proposed input-aware skipping SAR-ADC. Section IV presents experimental results, including comparisons with related work and various analyses of the proposed CIM. Finally, Section V concludes the paper with a discussion of related work.

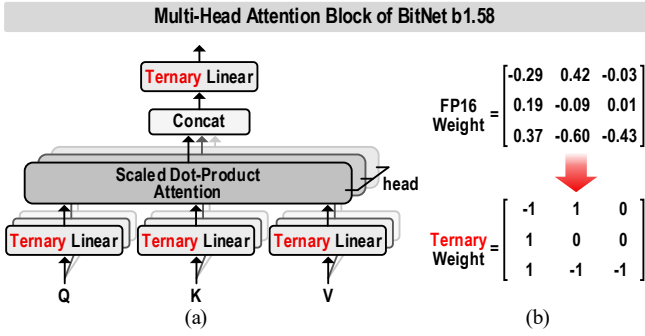


Fig. 1. (a) Multi-Head Attention block of BitNet b1.58 [8] with ternary linear layers and (b) FP16-to-ternary weight quantization example

II. BACKGROUND AND MOTIVATION

A. Ternary Linear Layers of BitNet b1.58

BitNet b1.58 [8] replaces all 16-bit floating point (FP16) linear layers in the multi-head attention block and feedforward network with ternary linear layers [16]. Fig. 1(a) shows the multi-head attention block, where linear projections for Query, Key, and Value, as well as the output projection, are done using ternary linear layers. In a feedforward network, two linear layers are implemented with ternary linear layers, improving efficiency throughout the transformer block. Conventional transformers use FP16-based weights for linear transformations, whereas BitNet b1.58 limits weight values to $\{+1, 0, -1\}$, removing the need for costly floating-point multiplications. Fig. 1(b) shows how FP16 weights are converted into ternary values. The ternary quantization approach lowers computational complexity, memory access, and latency with negligible accuracy loss. With the ternary representation of weights, the BitNet b1.58 shows promise for effective deployment in resource-constrained applications such as mobile and edge devices [8].

B. Ternary Bitcell and ADC Design

As compute-in-memory (CIM) has been proposed to accelerate various deep learning (DL) models, many previous research works have tried to design bitcells supporting ternary computations. One of the drawbacks encountered with the previous ternary CIM studies [9]–[11] is that the designs of SRAM-based bitcells for ternary computations are based on two 6T SRAMs, employing 16T–19T structures [9], [10]. In order to relieve the bitcell area burden of SRAM-based bitcells, eDRAM-based bitcells with a 4T2C structure [11] have been proposed. The 4T2C design improves bitcell density, but it requires periodic refresh operations due to leakage currents in the storage node. These refresh operations significantly degrade the energy efficiency of CIM.

Beyond bitcell design, the analog-to-digital converter (ADC) is a critical element in CIM architecture design. For low-overhead data conversion, many studies have adopted the cut-off technique [12]–[15]. The cut-off technique first analyzes multiply-accumulate (MAC) voltage distribution and restricts the conversion range to the region where most data values are concentrated, while the values outside the range are saturated to the maximum or minimum. Reducing the conversion range lowers ADC resolution, which in turn reduces both ADC area and energy overhead. In previous deep neural networks (DNNs) with ternary weights, outliers occur infrequently, and they have a negligible impact on model performance, allowing the MAC-based cut-off approach to maintain model accuracy sufficiently.

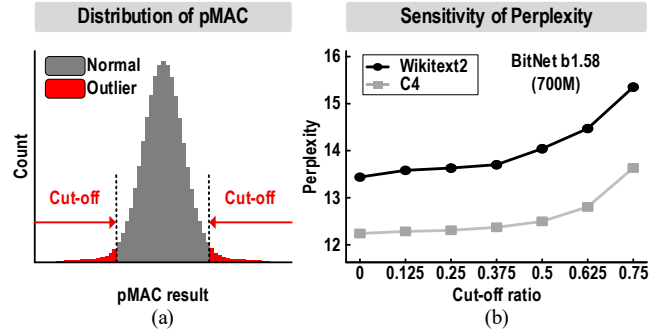


Fig. 2. (a) Distribution of partial multiply-accumulate result, where normal cases and outliers are separated by a cut-off and (b) perplexity variations on Wikitext2 and C4 with respect to different cut-off ratios

C. Sensitivity of Cut-off Ratio in Ternary LLMs

Fig. 2(a) shows the partial multiply-accumulate (pMAC) distribution of CIM based on the BitNet b1.58 (700M) model, where rare but influential outliers are observed. In LLMs, such outliers significantly affect model accuracy [17]–[20]. To analyze the sensitivity of ternary LLMs to the cut-off, we simulated perplexity on the Wikitext2 and C4 datasets, and the results are shown in Fig. 2(b). Here, perplexity means a metric for evaluating language model performance, and lower values indicate better predictive ability. As shown in the figure, the perplexity increases sharply in both datasets as the cut-off ratio grows. This demonstrates that cut-off techniques, although effective for previous DNNs with ternary weights, are no longer effective for LLMs. Therefore, a new energy efficient ADC design technique that is capable of full-range conversion is highly needed.

III. ARCHITECTURE OF HINT

In this section, we present the HINT architecture, which addresses the limitations of prior CIM designs for ternary LLMs. Subsection III-A provides an overview of the architecture, while Subsections III-B and III-C describe the design and operation of the proposed ternary bitcell and ADC, respectively.

A. Overall HINT Architecture

Fig. 3(a) presents a 32×129 HINT architecture that uses pMAC operations with binary-sliced inputs and ternary weights. These pMAC results are accumulated through the shift & add module to complete the MAC operation for up to 8-bit signed inputs. The current-domain computing CIM has a compute array, a minimum and maximum (min&max) array, a reference array, an input-aware skipping SAR-ADC (IAS-ADC), and peripheral circuits. The compute array is configured as 32×96 , supporting 32 inputs in parallel for stable ADC sensing and reflecting BitNet b1.58 700M’s queries and keys of dimension ($d_k = 96$). The proposed HINT employs two key techniques. First, an SRAM–MRAM hybrid ternary bitcell (HTC) is employed in the compute, min&max, and reference arrays. By leveraging HTC’s compact weight storage and low-leakage, the implementation enables efficient pMAC computation for ternary LLMs while eliminating the refresh overhead of eDRAM. Second, input-aware voltages (RBL_{\min} , RBL_{\max}) are generated using the min&max array, and the data range of the pMAC is dynamically determined. Our input-aware voltages enable the IAS-ADC to perform full-range conversion with improved energy efficiency, accurately capturing rare but influential ternary LLMs workload outliers.

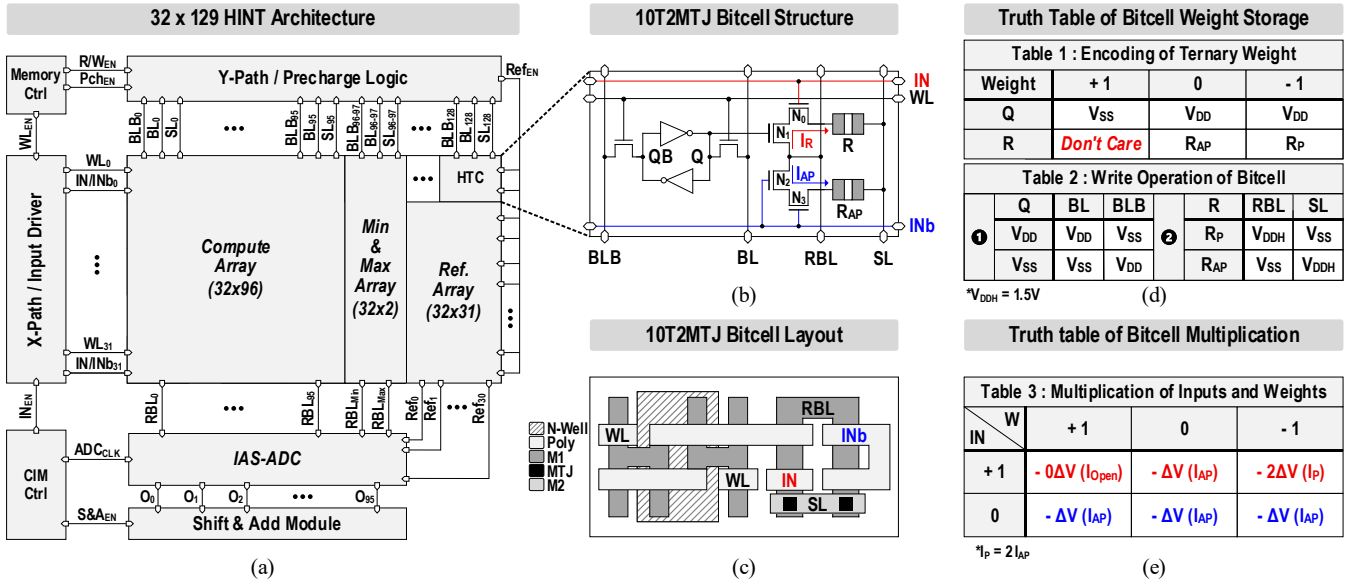


Fig. 3. (a) Overall architecture of the HINT (b) Schematic of 10T2MTJ bitcell (c) Layout of 10T2MTJ bitcell (d) Ternary weight encoding method and storage method of 10T2MTJ bitcell (e) Multiplication details of 10T2MTJ bitcell

B. Proposed Hybrid Ternary Bitcell

Fig. 3(b) shows the structure of a 10T2MTJ-based HTC. The HTC consists of 6T SRAM, two magnetic tunnel junction (MTJ), and four N-type transistors (N_0 – N_3). It also includes the bit line (BL) and bit line bar (BLB), along with the read bit line (RBL) and source line (SL). The SRAM stores data at node Q as a voltage level (V_{DD} or V_{SS}), while the upper MTJ stores data at node R as either a parallel (R_P , low-resistance) or anti-parallel (R_{AP} , high-resistance) state. Thus, the ternary weight is encoded by the voltage at Q and the resistance at R. The main idea of ternary bitcell is to use SRAM as a switch to control discharge path formation and to adjust the resistance of MTJ, thereby implementing ternary operations through three distinct IR drop levels. The lower MTJ is fixed in the R_{AP} state so that it provides the I_{AP} path when INb is active. For linearity, HTC is implemented using two transistors (N_2 , N_3) in the same configuration as the I_R path. By combining the reliability of SRAM with the high-density and non-volatility of MRAM, this hybrid structure provides a more compact ternary bitcell design with lower leakage and refresh-free operation compared to conventional ternary bitcells. Fig. 3(c) illustrates the HTC layout, where the MTJ is vertically stacked between the M1 metal layer of transistors (N_0 , N_3) and the M2 metal layer of the SL, enabling integration without area overhead. In addition, source–drain sharing among N-type transistors (N_{0-1} , N_{2-3}) further reduces the layout area.

Let us consider the ternary weight encoding and write operations. Fig. 3(d) shows the encoding method for ternary weights (Table 1) and how to store them in SRAM (Q) and MTJ (R) (Table 2). Writing a ternary weight into the HTC involves two steps: SRAM (Q) write and MTJ (R) write. As an example, the procedure for storing a ternary weight of -1 is as follows: First, as shown in ① of Table 2, BL is set to V_{DD} and BLB to V_{SS} while WL is activated, storing SRAM (Q) as V_{DD} . Next, ② states that RBL is at V_{DDH} and SL at V_{SS} while IN is active, switching the MTJ (R) to R_P state. Although MTJ writes are more costly than SRAM, the overhead is a one-time cost, since the model weights are written once per layer. Furthermore, in the +1 ternary weight case of Table 1, the MTJ (R) is in a don't care state, and thus the MTJ (R) write step is skipped.

The multiplication result of a binary input and a ternary weight is $\{+1, 0, -1\}$, and it requires the bitcell to generate corresponding analog outputs in CIM. The proposed HTC adopts a current discharge-based scheme, where the line resistance can be open, high, or low, generating changes in current magnitude and corresponding voltage drops across the RBL, which is precharged to V_{DD} . As a result, three distinct discharge voltage levels ($-0\Delta V$, $-\Delta V$, $-2\Delta V$) are produced according to the multiplication result of each bitcell. As shown in Fig. 3(e), I_R or I_{AP} path is determined depending on input value, and the voltage drop corresponding to the multiplication result is determined by the stored weight. When the input is +1, the I_R path is activated, as illustrated in Fig. 3(b). For the weight of +1, V_{SS} is stored in Q, causing the N_1 transistor to turn off and I_R path to become open. Therefore, in this case, no voltage drop ($-0\Delta V$) occurs. When the weight is 0, V_{DD} is stored in Q and forms I_R path. In this case, I_{AP} flows due to R_{AP} state, resulting in a voltage drop of $-\Delta V$. When the weight is -1, I_P flows due to the corresponding R_P state. By sizing transistors (N_0 – N_3) and configuring the tunnel magnetoresistance ratio (TMR), I_P is designed to be twice of I_{AP} . As a result, the voltage drop becomes $-2\Delta V$. When the input is 0, the multiplication result is always 0. In this case, I_{AP} flows through the lower MTJ fixed in the R_{AP} state, consistently generating a voltage drop of $-\Delta V$ regardless of the weight. The proposed bitcell supports ternary multiplication while ensuring consistent operation across all input–weight combinations.

Fig. 4 shows the timing diagram for pMAC operations in the HINT, illustrating one compute array column (RBL_N), minimum array (RBL_{Min}), and maximum array (RBL_{Max}). Each RBL connects to 32 HTCs, so with 32 inputs activated in parallel, pMAC operations are performed within the range of $|\pm 32|$. RBL_N stores model weights, while RBL_{Max} and RBL_{Min} are set to +1 and -1, respectively, to provide reference boundaries for the maximum and minimum pMAC values. HINT operates in two phases: precharge and pMAC. In the precharge phase, Pch_{EN} is asserted to charge all RBLs to V_{DD} through the precharge logic. In the pMAC phase, input pulses applied to IN and INb discharge each HTC according to the stored weight. The 32 discharge voltages combine on one RBL to produce the analog pMAC output. The output of the

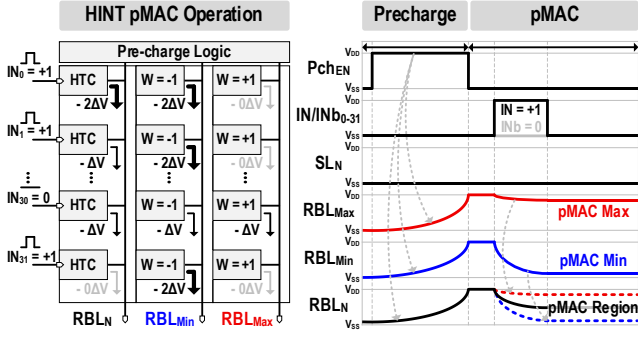


Fig. 4. Overview of HINT pMAC operation and timing diagram of the CIM mode

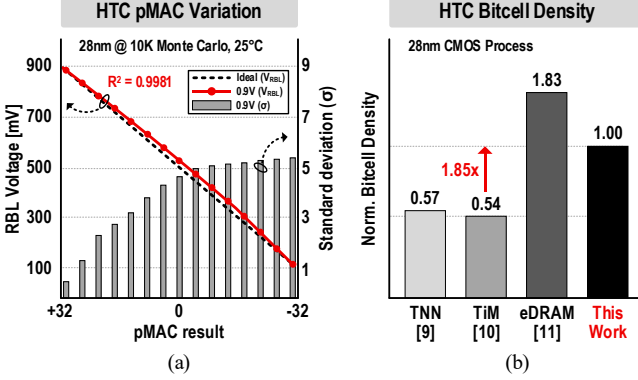


Fig. 5. (a) RBL voltage curve and the standard deviation with the pMAC result and (b) comparison of normalized bitcell density with previous ternary bitcells in 28nm process

pMAC varies depending on the input, while RBL_{Max} and RBL_{Min} always generate the voltages corresponding to the maximum and minimum values of the input. So, RBL_{Max} and RBL_{Min} ensure that all the compute array outputs (RBL_{0-95}) remain within the defined region. These boundaries are subsequently utilized to generate the skip enable signal for the IAS-ADC.

To validate the correctness and robustness of the proposed current discharge-based operations, a 3σ Monte Carlo simulations with 10K samples have been conducted to check the linearity and variation of the current discharge-based HINT, and the results are shown in Fig. 5(a). After comparing the ideal case (dotted line) with simulation results (solid line) at $V_{DD} = 0.9$ V, we can notice from the results that linearity and variability stay stable across voltage levels. Fig. 5(b) compares the normalized bitcell density of SRAM-CIM and eDRAM-CIM, which are reproduced using 28nm CMOS process. The proposed HTC has 1.75 \times and 1.85 \times higher bitcell density than the previous TNN [9] and TiM [10], compared to SRAM-CIM. While the bitcell density is lower than that of eDRAM-CIM [11], the HTC provides higher energy efficiency without refresh operations, which will be further discussed in Section IV.

C. Proposed Input-aware Skipping SAR-ADC

As discussed in Subsection II-C, rare but influential outliers in ternary LLMs require full-range conversion, which traditional range cut-off methods cannot effectively achieve. Conventional SAR-ADCs further exacerbate the problem since their energy cost increases with precision, and range cut-off approaches fail to handle outliers effectively [17]–[20]. To tackle this challenge, we propose input-aware skipping SAR-ADC (IAS-ADC), which enables energy efficient full-range conversion. Fig. 6 shows the IAS-ADC schematic, where the

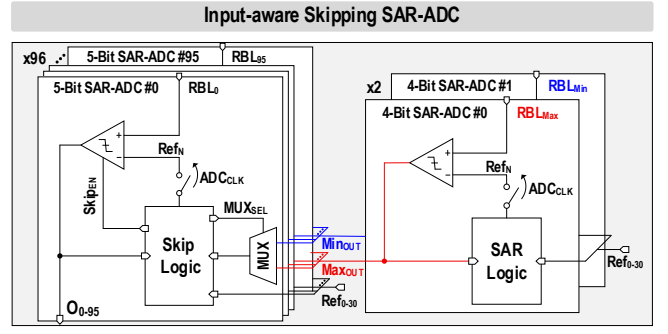


Fig. 6. Schematic of Input-aware Skipping SAR-ADC

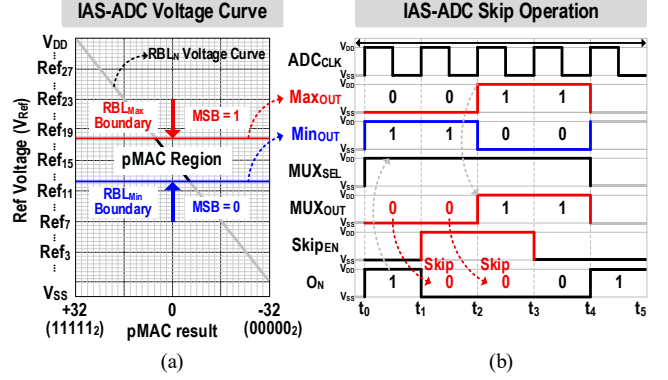


Fig. 7. (a) Voltage curve of IAS-ADC operation with minimum and maximum boundaries defining the pMAC region and (b) timing diagram of IAS-ADC operation with skip when O_N MSB is 1

design integrates 96 5-bit SAR-ADCs for RBL_{0-95} conversion and two 4-bit SAR-ADCs for RBL_{Max} and RBL_{Min} conversion. The outputs from the two 4-bit SAR-ADCs are multiplexed to the 96 5-bit SAR-ADCs, where the MSB of each 5-bit SAR-ADC serves as MUX_{SEL} control signal for skipping selection. To ensure consistent conversion, all SAR-ADCs share 31 reference voltages generated by the reference array. A centralized controller distributes ADC_{CLK} and control signals, and the skip logic in each 5-bit SAR-ADC enables 96 SAR-ADCs to perform skip operations simultaneously. As a result, RBL_{Max} and RBL_{Min} allow the IAS-ADC to adjust conversion ranges dynamically, maintaining accuracy even under outlier conditions.

Let us consider the operational principle of IAS-ADC schematic (Fig. 6), which is illustrated in Fig. 7. Fig. 7(a) shows that the pMAC results range from -32 to $+32$ and are quantized into 5-bit binary codes (00000_2 – 11111_2). The range -32 to 0 for pMAC has $MSB = 0$, while the range $+1$ to $+32$ has $MSB = 1$. In the conventional SAR-ADC [21], MSB is determined in the first cycle, and the remaining bits are resolved by binary search, halving the reference range each cycle. By contrast, IAS-ADC uses RBL_{Max} and RBL_{Min} from the min&max array to define a pMAC region, avoiding reference voltage comparisons. This approach reduces energy consumption without accuracy degradation.

Fig. 7(b) illustrates the overall process, where the outputs of RBL_{Max} and RBL_{Min} serve as boundaries for determining early decisions in the compute array outputs. At t_1 , a conventional SAR-ADC compares RBL_N with the reference voltage Ref_{23} when $MSB = 1$. In contrast, IAS-ADC has already confirmed at t_0 , by comparing RBL_{Max} and Ref_{23} , that the MAX_{OUT} bit is 0, ensuring the upper bound of the second bit is 0. In this case, the skip condition is satisfied, activating $Skip_{EN}$. As a result, O_N skips the second-bit comparison and

HINT Architecture Layout		HINT Architecture Summary	
Memory Ctrl	Y-Path / Precharge Logic	Technology	28nm CMOS
X-Path & Input Driver	Compute Array	Supply Voltage	0.9V (1.5V), 100MHz
ADC Ctrl	Min&Max IAS-ADC	Bitcell Type	10T2MTJ
		Array Size	32 x 129
		Total Area	0.0167mm ²
		Input / Weight	1 – 8-bit / 1.58-bit
		MAC Technique	Current-domain
		Energy Efficiency	839 TOPS/W

Fig. 8. The proposed HINT architecture layout and its summary

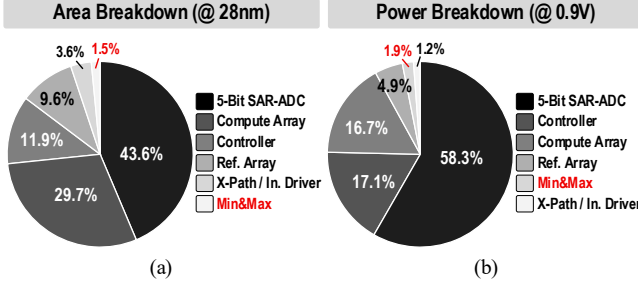


Fig. 9. (a) The area breakdown based on the proposed HINT architecture layout in 28nm process and (b) power breakdown at 0.9V

directly sets the bit to 0. If $MAX_{OUT} = 0$ at t_1 , the O_N conversion at t_2 is skipped and set to 0. If $MAX_{OUT} = 1$, the next O_N bit can be 0 or 1, so a comparison with the reference voltage is needed. If the comparison matches, the skip condition remains valid, and the decision to skip is then determined by the MAX_{OUT} value in the next cycle. Conversely, if it does not match, the skipping process is terminated in the 5-bit SAR-ADC of the relevant RBL column, and the remaining bits are determined by comparing with the reference voltage. When $MSB = 0$, the operation is symmetric. If $MIN_{OUT} = 1$, the next O_N bit is set to 1 automatically. When all inputs are 0, the pMAC is 0, and the MSB is 0. MIN_{OUT} is chosen by the multiplexer, and its output is 1 each cycle, so the lower bits of O_N are set to 1 without comparison. Then, all the cycles can be skipped except for the MSB, and this skipping operation is implemented in parallel across the 96 5-bit SAR-ADCs. As a result, the proposed IAS-ADC leverages the ternary property that fixes the MSB based on the sign of the pMAC and it is designed to determine the second bit early using a 4-bit SAR-ADC. This approach maintains the same throughput as a conventional 5-bit SAR-ADC. In addition, since the proposed min&max circuits are shared across 96 5-bit SAR-ADCs, the area and energy overhead are negligible, occupying only 1.5% of area. With input-dependent boundaries, the proposed IAS-ADC preserves accuracy in outlier cases and reduces reference comparisons, thereby enhancing energy efficiency and realizing low-power CIM.

IV. EXPERIMENTAL RESULTS

A. HINT Architecture Design and Implementation Results

Fig. 8 shows the layout and summary of the HINT in 28nm CMOS technology. By leveraging HTC and IAS-ADC, the proposed HINT achieves 839 TOPS/W for ternary LLM acceleration. The 10T2MTJ cell stacks MTJ between the M1 metal layer and the M2 metal layer without increasing area, as shown in Fig. 3(c). This HTC cell structure enables compact integration and contributes to both high-density and low-power operation. The resulting HINT architecture, sized 32×129 with a total area of 0.0167 mm², achieves high array density. Moreover, the HINT supports flexible precision with 1–8-bit inputs and ternary weight representation, designed to

TABLE I. MTJ DEVICE PARAMETERS

Parameter	Value
Free Layer Dimension [nm ³]	$30 \times 30 \times 0.6$
Damping Factor α	0.1
MTJ Resistance [Ω]	9K (P) / 22.5K (AP)
Switching Current [A]	52 μ (P) / 38.3 μ (AP)
Pulse-width [s]	3n
Tunnel Magnetoresistance Ratio (TMR) [%]	150
Variation in Resistance	5% @ 1- σ

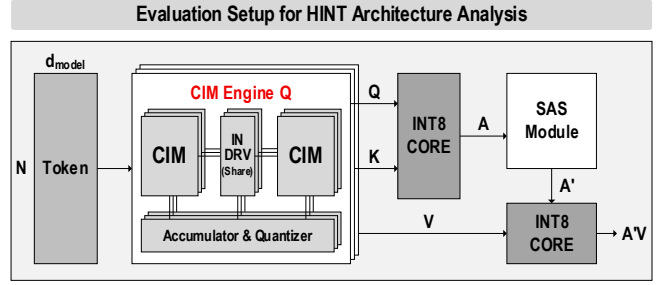


Fig. 10. Evaluation setup for Self-Attention block (d_k) computation of BitNet b1.58 (700M)

support ternary LLM workloads that benefit from reduced precision while maintaining accuracy.

Fig. 9 shows the area and power breakdown of the proposed HINT. Fig. 9(a) shows that the 5-bit SAR-ADC takes the largest 43.6% of the total area. The compute array with HTCs makes up 29.7% of the total area, while the min&max array and the 4-bit SAR-ADC for IAS-ADC add just 1.5% overhead. Fig. 9(b) illustrates the power breakdown of the proposed HINT at a supply voltage of 0.9 V. The analysis shows that the 5-bit analog readout uses 58.3% of the total power, while the compute array uses 16.7%. The min&max array and 4-bit SAR-ADC for IAS-ADC add only 1.9% overhead, allowing for a 1.27 \times increase in energy efficiency based on BitNet b1.58 (700M). Table I summarizes the MTJ device parameters used in the simulations [22]. The parameters have been used for Monte Carlo and device-level simulations. The tunnel magnetoresistance ratio (TMR), which represents the relative resistance difference between the parallel (low-resistance) and anti-parallel (high-resistance) states of an MTJ, is assumed to be 150%. This value considers the transistor resistance so that I_P becomes twice I_{AP} , as shown in Fig. 3(e). The previous baseline CIM architectures have also been reproduced in 28nm CMOS process technology for area and energy comparison.

B. Evaluation of HINT Architecture on BitNet b1.58

Fig. 10 illustrates the evaluation setup, which is designed to analyze the energy efficiency of HINT and IAS-ADC while performing computations of the self-attention block (d_k). In BitNet b1.58, all the operations except for linear layer are executed with INT8 precision. INT8 core is used for $A = QK^T$ and $A'V$ operations, while $A' = \text{softmax}(A/\sqrt{d_k})$ operation occurs in the scaling and softmax (SAS) module. The Q, K, and V blocks are each generated in a CIM engine consisting of 48 macros (d_{model}), where all the required weights are stored and reused. The attention computation is tiled into 128×128 blocks along the sequence dimension, where each tile is computed using an INT8 core. To ensure seamless operation between the CIM engine and the INT8 core, each INT8 core comprises 16×96 8-bit MAC units and a 72 KB double buffer sized to the CIM output (Q, K, and V).

Energy Evaluation in Self-Attention Block of BitNet b1.58 (700M)

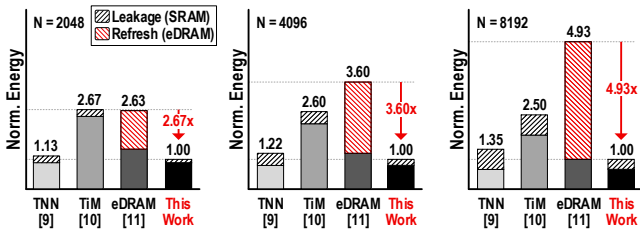


Fig. 11. Energy evaluation in the Self-Attention block for different token sizes

IAS-ADC Cycle-Skipping Evaluation for Dataset Inference of BitNet b1.58

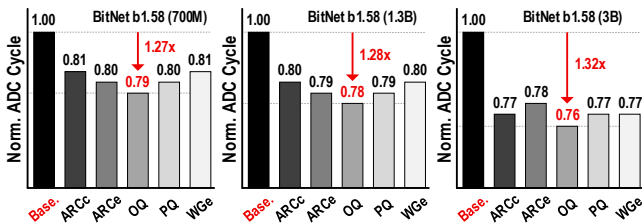


Fig. 12. IAS-ADC Cycle-Skipping evaluation in dataset inference for different model sizes

Fig. 11 shows the normalized energy consumption of the BitNet b1.58 (700M) self-attention block based on token size (N). Energy is analyzed by CIM operation energy, SRAM leakage, and eDRAM refresh. Unlike the conventional method that uses one ADC per line, TiM [10] employs two ADCs to generate a MAC result, leading to higher computation energy. The proposed HINT improves energy efficiency from 2.67 \times at N = 2048 to 4.93 \times at N = 8192 as the token size increases. The proposed HTC performs low-power computations and uses a single SRAM, which greatly reduces leakage energy compared to SRAM-based bitcells. Unlike eDRAM-based bitcells, it does not need refresh operations, making HINT more energy efficient as the token size increases. It operates with high energy efficiency due to the features of LLMs, which have large token sizes and long computation times.

Fig. 12 presents the analysis of SAR-ADC cycle requirements for dataset inference across different model sizes, evaluated on five zero-shot datasets per model size. The baseline represents the cycles required by a conventional 5-bit SAR-ADC, while IAS-ADC achieves a lower cycle count through input-aware skipping. For 700M model, consistent reductions are observed across all datasets, with OQ dataset showing up to a 21% decrease. This demonstrates that IAS-ADC requires fewer cycles than a conventional 4-bit SAR-ADC. Moreover, as model size increases, the proportion of skipped cycles grows, indicating that IAS-ADC scales favorably with the ever-expanding size of LLMs. In summary, IAS-ADC supports full-range conversion to handle outliers while applying skipping to most operations, thereby enabling inference with significantly higher energy efficiency.

C. Comparison With Previous Works

Table II summarizes the comparison results with existing ternary CIMs, including supply voltage, process node, bitcell type, input/weight precision, and array size. As shown in the table, the proposed HINT uses 0.9 V and 1.5 V as supply voltages. 1.5 V is used only for MRAM write operations, while all the MAC operations are performed at 0.9 V. The proposed 10T2MTJ bitcell uses fewer transistors than TiM [10], achieving a 1.85 \times improvement in density, while its hybrid structure mitigates leakage and refresh issues, leading

TABLE II. COMPARISON TO PRIOR TERNARY CIM WORKS

Reference	T-CAS II'23[9]	T-VLSI'20[10]	T-CAS I'21[11]	This Work
Technology	65nm	32nm	65nm	28nm
Memory Type	SRAM	SRAM	eDRAM	Hybrid
Bitcell Type	16T1C	19T	4T2C	10T2MTJ
¹⁾ Bitcell Area	1.042 μm^2	1.105 μm^2	0.324 μm^2	0.595 μm^2
Input	1.58-bit	1.58-bit	1 – 2.32-bit	1 – 8-bit
Weight	1.58-bit	1.58-bit	1.58 – 2.81-bit	1.58-bit
Supply Voltage	1.0V	1.0V	0.5V – 0.7V	0.9V (1.5V)
Array size	96 \times 32	256 \times 256	128 \times 128	32 \times 129
TOPS/W	823	132.7	552.5 (1:1b/W:1.58b)	839 (1:1b/W:1.58b)
ML Algorithm	CNN	CNN	CNN	LLM
Dataset	CIFAR-10	ImageNet	CIFAR-10	²⁾ Zero-shot
Accuracy	91.2%	56.5%	82.8%	³⁾ 42.48%

¹⁾The previous bitcells are reproduced in 28nm CMOS process technology.

²⁾The results are averaged over ARcC, ARcCe, OQ, PQ, and WGe.

³⁾The baseline accuracy of the software is 43.23%.

up to 2.67 \times energy efficiency improvement on BitNet b1.58 (700M). By further applying IAS-ADC, HINT supports full-range conversion and achieves up to 21% cycle reduction through cycle skipping. Overall, HINT achieves an energy efficiency of 839 TOPS/W with 42.48% accuracy on zero-shot dataset inference, with less than a 1% accuracy loss compared with the baseline.

V. CONCLUSION

In this paper, we proposed a CIM architecture, HINT, that integrates two energy efficient techniques. It incorporates an SRAM–MRAM-based hybrid ternary bitcell (HTC) to reduce area usage and suppress leakage. This hybrid structure enables compact integration while maintaining energy efficiency. In addition, HINT introduces an input-aware skipping SAR-ADC (IAS-ADC) that leverages input-dependent boundaries to support full-range conversion. By skipping unnecessary reference voltage comparisons, IAS-ADC preserves accuracy while improving energy efficiency. Experimental results on the BitNet b1.58 (700M) model show that HINT improves bitcell density and energy efficiency compared to SRAM-based and eDRAM-based designs. With IAS-ADC, it achieves additional efficiency gains without significant overhead or accuracy loss. Overall, HINT effectively addresses the challenges of low-density and ADC accuracy degradation in ternary CIM, enabling compact and energy efficient LLM acceleration.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00345481) and (RS-2024-00405495, Plug&Play (P&P) Chiplet Integration research center), in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (RS-2023-00229028, eMRAM based Highly Reliable and Lower Power Authentication Hardware Design), and in part by Samsung Electronics, Co., Ltd under Grant IO251216-14700-01. The EDA tool was supported by the IC Design Education Center (IDEC), Korea.

REFERENCES

- [1] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," arXiv:2001.08361, Jan. 2020.
- [2] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han, "Large Language Models on Graphs: A Comprehensive Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 8622–8642, Dec. 2024.
- [3] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A Survey on Mixture of Experts in Large Language Models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 7, pp. 3896–3915, Jul. 2025.
- [4] Y. Qin, Y. Wang, Z. Zhao, X. Yang, Y. Zhou, S. Wei, Y. Hu, and S. Yin, "MECLA: Memory-Compute-Efficient LLM Accelerator with Scaling Sub-matrix Partition," in *Proc. ACM/IEEE 51st Annual International Symposium on Computer Architecture*, pp. 1032–1047, Jun. 2024.
- [5] S. Shen, Z. Dong, J. Y. Ye, L. J. Ma, Z. W. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, "Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT," in *Proc. 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 8815–8821, Feb. 2020.
- [6] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "HAQ: Hardware-Aware Automated Quantization with Mixed Precision," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8612–8620, Jun. 2019.
- [7] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," in *Proc. International Conference on Learning Representations*, pp. 1–14, May. 2016.
- [8] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei, "The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits," arXiv:2402.17764, Feb. 2024.
- [9] H. Jeong, S. Kim, K. Park, J. Jung, and K. J. Lee, "A Ternary Neural Network Computing-In-Memory Processor With 16T1C Bitcell Architecture," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1739–1743, May. 2023.
- [10] S. Jain, S. K. Gupta, and A. Raghunathan, "TiM-DNN: Ternary In-Memory Accelerator for Deep Neural Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 7, pp. 1567–1577, Jul. 2020.
- [11] C. Yu, T. Yoo, H. Kim, T. T.-H. Kim, K. C. T. Chuan, and B. Kim, "A Logic-Compatible eDRAM Compute-In-Memory With Embedded ADCs for Processing Neural Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 2, pp. 667–679, Feb. 2021.
- [12] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1733–1743, Jun. 2020.
- [13] 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, Jul. 2020.
- [14] J. Song, X. Tang, X. Qiao, Y. Wang, R. Wang, and R. Huang, "A 28 nm 16 Kb Bit-Scalable Charge-Domain Transpose 6T SRAM In-Memory Computing Macro," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 5, pp. 1835–1845, May. 2023.
- [15] N. Kang, H. Kim, H. Oh, and J.-J. Kim, "TAIM: Ternary Activation In-Memory Computing Hardware with 6T SRAM Array," in *Proc. 59th ACM/IEEE Design Automation Conference*, pp. 1081–1086, Jul. 2022.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Proc. 31st International Conference on Neural Information Processing Systems*, vol. 30, pp. 6000–6010 Dec. 2017.
- [17] S. He, H. Zhu, H. Zhang, Y. Ma, Z. Chen, M. Li, D. Zhai, C. Chen, Q. Liu, X. Zeng, and M. Liu, "A 22-nm 109.3-to-249.5-TFLOPS/W Outlier-Aware Floating-Point SRAM Compute-in-Memory Macro for Large Language Models," *IEEE Journal of Solid-State Circuits*, pp. 1–14, May. 2025.
- [18] C. Guo, J. Tang, W. Hu, J. Leng, C. Zhang, F. Yang, Y. Liu, M. Guo, and Y. Zhu, "OliVe: Accelerating Large Language Models via Hardware-friendly Outlier-Victim Pair Quantization," in *Proc. ACM/IEEE 50th Annual International Symposium on Computer Architecture*, pp. 1–15, Jun. 2023.
- [19] Y. Bondarenko, M. Nagel, and T. Blankevoort, "Quantizable Transformers: Removing Outliers by Helping Attention Heads Do Nothing," in *Proc. 37th International Conference on Neural Information Processing Systems*, vol. 36, pp. 75067–75096, Dec. 2023.
- [20] J. Lee, W. Lee, and J. Sim, "Tender: Accelerating Large Language Models via Tensor Decomposition and Runtime Requantization," in *Proc. ACM/IEEE 51st Annual International Symposium on Computer Architecture*, pp. 1048–1062, Jun. 2024.
- [21] X. Tang, J. Liu, Y. Shen, S. Li, L. Shen, A. Sanyal, K. Ragab, and N. Sun, "Low-Power SAR ADC Design: Overview and Survey of State-of-the-Art Techniques," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 6, pp. 2249–2262, Jun. 2022.
- [22] I. Ahmed, Z. Zhao, M. G. Mankalale, S. S. Sapatnekar, J.-P. Wang, and C. H. Kim, "A Comparative Study Between Spin-Transfer-Torque and Spin-Hall-Effect Switching Mechanisms in PMTJ Using SPICE," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 3, pp. 74–82, Dec. 2017.