Accelerator Design using 3D Stacked Capacitorless DRAM for Large Language Models

Janak Sharda, Po-Kai Hsu, Shimeng Yu
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA
shimeng.yu@ece.gatech.edu

2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS) | 979-8-3503-8363-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AICAS59952.2024.10595901

***Abstract*— Large language models (LLMs) have been immensely useful for natural language processing tasks. However, the current model sizes are increasing exponentially, along with generating large amounts of intermediate data. Here, we propose to use the capacitorless 3D stackable DRAM, which is an emerging memory enabling scaling of DRAM in the vertical direction like 3D NAND Flash. A 3D DRAM can store much larger LLMs compared to conventional DRAM at higher density. Further, to reduce the intermediate data size, we propose to use a layer-wise sparsity-quantization hybrid (LSQH) algorithm, which induces sparsity based on calculations performed using low-bit quantization to reduce both the energy consumption and the data storage requirements. Finally, a 3D heterogeneously integrated accelerator is designed by stacking a 3D DRAM with logic dies designed in the 3 nm technology node, which exploits the LSQH algorithm for the Llama2 model. The evaluation of the proposed system shows an energy efficiency of > 25 TOPS/W and an area efficiency of > 14.4 TOPS/mm2, with minimal drop in accuracy. Further model scaling is performed to obtain the energy efficiency and area efficiency for larger LLMs.**

Keywords—Large language model, 3D stackable DRAM, hardware accelerator, heterogenous integration

I. Introduction

Machine learning algorithms have advanced rapidly in recent years, proving to be essential in natural language tasks. Foundational models like large language models (LLMs) have been used for many text-generative tasks [1]. Currently, the size of the language model is scaling at an exponential rate, much faster than Moore's law [2], hence making it difficult for custom-designed accelerators to scale with model size. Current models like GPT-3, Llama2, etc. have very large memory size requirements of more than 50 GB [1][3], both for storing the model and for storing the intermediate data, as shown in Fig. 1. NAND Flash-based solid-state disks (SSDs) are usually used to meet such large memory requirements. However due to very slow read speed, in the current implementations of hardware accelerators, the LLM is required to be completely loaded to GPU's DRAM platform such as high-bandwidth-memory (HBM). Due to the limit of the available storage, prior works have demonstrated the use of multiple GPUs for running inference of LLMs. However, such large memory requires several GPUs to load a model and additionally incur the cost of data movement among different GPUs [4]. Monolithically 3D stackable DRAM [5][6] has been proposed as an alternative DRAM such that it is scalable in the vertical direction similar to 3D NAND Flash, as shown in Fig. 2. This vertical scaling enables high-capacity DRAM, which can be used for storage in LLM accelerators.

In this work, we propose to use 3D stackable DRAM for storing the entire model and the intermediate data being generated at the runtime. The proposed LLM accelerator takes advantage of the heterogeneous integration where a 3D capacitorless DRAM die is bonded with logic dies of digital systolic array architecture, to enable the superior bandwidth for data transfer. Furthermore, we use layer-wise mixed sparsity-quantization methods to reduce the amount of data storage required. The software and hardware performance metrics were evaluated for the open-source Llama2 model [3].

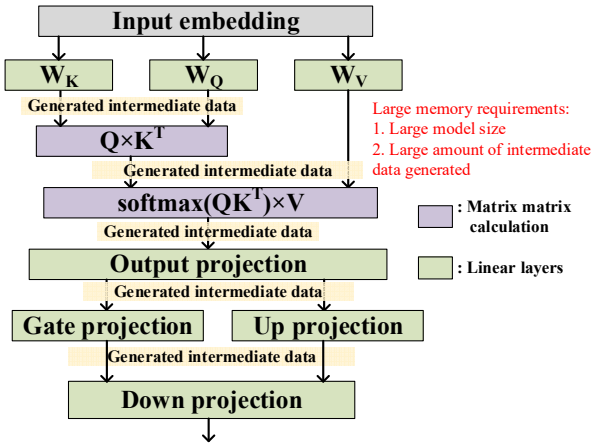

Fig. 1. Single decoder layer of the Llama2 [3] model. For simplicity, only multiply-and-accumulate (MAC) operations are shown. The total intermediate data size generated during runtime is on 50 GB-level.

II. Background

A. Large-language model

Large language models based on transformer decoder architecture have been fundamental for applications like ChatGPT [1] etc. for generative tasks. However, due to the large model size compared to CNNs, the inference speed is slower. Llama2 [3] is a publicly available LLM model with pre-trained weights and architecture open-sourced with a model size of tens of gigabytes (GBs). In this work, detailed

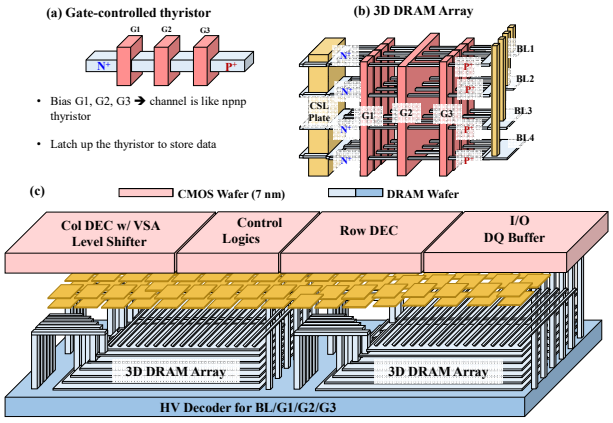

Fig. 2. (a) Gate-controlled thyristor-based device as capacitorless DRAM cell. (b) Subarray comprising of shared bitlines, gates G1/G2/G3 and common select lines. (c) The structure is further scaled up to obtain the complete 3D DRAM array as shown in the blue part at the bottom ((a) and (b) modified from [5]).

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on February 03,2025 at 09:49:11 UTC from IEEE Xplore. Restrictions apply.

evaluations are performed for the Llama2-13b model, which is the 13 billion parameter version of the Llama2 model. Llama2-13b model comprises of 40 decoder layers, each comprising 3 fully connected layers for query, key and value generation, and self-attention calculation followed by the 3 layers for output projection. Fig. 1 shows a single layer for the Llama2 model.

B. Challenges for accelerator design for LLMs

LLMs have large memory requirements, both for storing the model weights and the intermediate data. For example, the llama2-13b model has 40 decoder layers, producing nearly 50 GB of intermediate data. The model size is 26 GB, so the combined data size can be as large as 76 GB if all the layers are processed at once. Typically, to parallelize the model further, it is run on several GPUs at once to utilize the total amount of DRAM storage available on multiple GPUs. The number of resources required grows substantially with the model size, along with the additional cost of moving data between several devices. Such data movement usually occurs through PCIe links which reduces performance and increases energy consumption. The use of multiple GPUs also increases the total hardware cost due to higher total chip area as well as higher total power consumption. To improve the throughput of the system and run all the layers of the Llama2 at once, we propose to use 3D DRAM-based accelerator. To reduce the amount of memory required, in this work, we use a layer-wise sparsity-quantization hybrid (LSQH) algorithm. This reduces the memory requirements to store the intermediate data.

Table I: Device-level characteristics of 3D stackable DRAM device [5]

	3D DRAM device
Read/Write latency	< 50 ns
Endurance	> 10^{10} cycles
Retention	> 100 ms
Operating voltage	5 V

C. 3D stackable DRAM

3D stackable DRAM is an emerging memory technology targeting large memory capacity requirements while keeping the read and write latency to be similar to conventional DRAM. This technology is on the industry's roadmap for the sub-10 nm DRAM node. There are several device-level works for 3D stackable DRAM from commercial memory vendors like Macronix [5] and Samsung [6], etc. In this work, we perform evaluations for the capacitorless 3D stackable DRAM device reported in [5] using the available device-level dimensions and experimental results. Fig. 2.a shows the device structure of a single DRAM cell, which is biased to obtain a thyristor-like n-p-n-p junction that is gated by three biases. The thyristor can be latched to either program or erase state, depending upon the bias applied. The detailed operation and experimental results have been explained in [5]. The experimental results have reported competitive DRAM memory characteristics such as read/write latency < 50 ns, endurance > 10^{10} cycles (limited by measurement instrument), and retention > 100 ms at 85 °C. The device-level performance is summarized in Table I. This 3D stackable DRAM device can be monolithically fabricated in a 3D architecture as shown in the subarray in the Fig. 2.b. The figure also shows how different wordlines and bitlines are connected. In this case, the gate G1 and G3 are shared vertically, the gate G2 and common select line (CSL) are shared across the complete subarray, and the bitlines are shared horizontally.

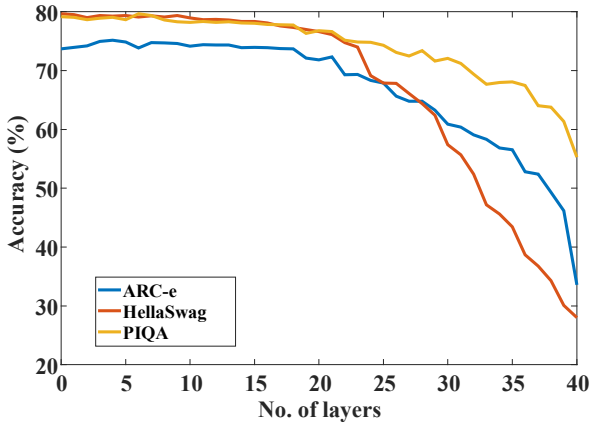

Fig. 3. The LSQH algorithm is applied to last 'n' number of layers. Plot shows the drop in accuracy vs 'n'. The initial point is the baseline accuracy. Three benchmark suites ARC-e, HellaSwag and PIQA are shown.

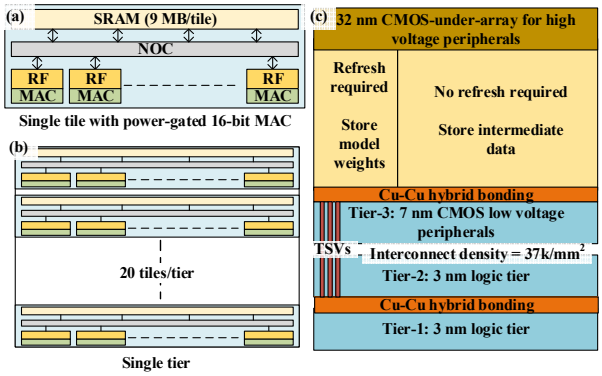

Fig. 4. (a) Complete accelerator design for single tile; (b) A single tier comprises 20 tiles to map 20 layers of Llama2; (c) Cross-section schematic of 3D stacked accelerator integrated with 3D DRAM.

III. Methodology

A. Layer-wise sparsity-quantization hybrid algorithm

Recent work [7] defines a methodology to use approximate computing with sparsity to efficiently use compute-in-memory, without losing accuracy. The intermediate matrix dimensions of QK^T, which is computed before calculating the self-attention, are very large. [7] uses compute-in-memory (CIM) to approximately compute the matrix-matrix multiplication and then compute only the significant values (output values higher than a chosen threshold) precisely, while setting the rest of the value to be 0. Such approximate computing works because the softmax function which follows the calculation QK^T amplifies larger values and reduces the smaller values. This reduces the significance of the smaller values and hence can be computed in an approximate way. However, CIM requires an expensive analog-to-digital converter (ADC), suffers from accuracy loss caused by process-voltage-temperature (PVT) variations, and is not scalable to leading edge logic tech nodes. Here, we use a similar idea in the digital domain by computing the matrix-matrix multiplication using low-bit precision of just 1-bit and then performing high-precision computing for the values higher than a certain threshold. The rest of the low values are set to 0 to introduce sparsity. This sparsity is then utilized with a simple sparsity-aware digital systolic array accelerator. However, applying this algorithm on every layer of Llama2-13b results in a large drop in final accuracy. Therefore, we use

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on February 03,2025 at 09:49:11 UTC from IEEE Xplore. Restrictions apply.

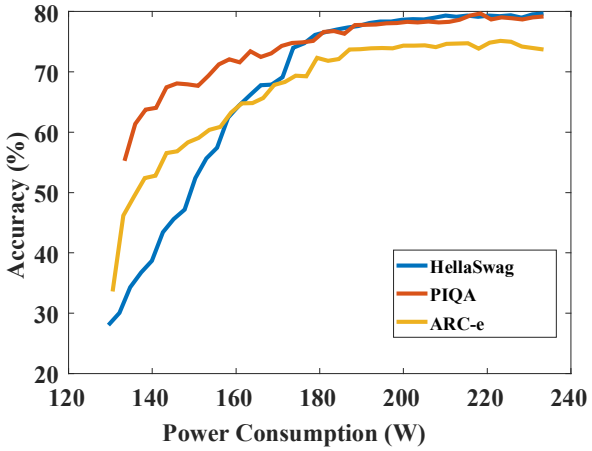

Fig. 5. Plot shows the tradeoff between accuracy vs. power consumption obtained by using LSQH algorithm.

the LSQH algorithm, where 1-bit quantization and sparsity are only applied to a few layers of the complete network. [8] describes that initial layers are more important to compute the final output and uses speculative decoding to perform early exit from the model. This is because the initial layers are able to predict the correct token. Here, we use a similar idea by performing accurate inference for initial layers and approximate inference of later layers. Hence, as shown in Fig. 3, the first 'n' number of layers are computed in an accurate way and for the remaining layers, the LSQH algorithm is applied. It is observed that the accuracy drop is negligible when the last 17 layers are dropped. Next, 'n' is varied and the final accuracy and PPA are computed, as explained in section IV. The evaluations are performed using the EleutherAI's package, lm-evaluation-harness [9], on 3 popular benchmarks: PIQA, HellaSwag, and ARC-e. These benchmarks have also been used in accuracy benchmarking of the actual Llama2 model in [3].

B. 3D stackable DRAM array design

The 3D DRAM array is designed with the peripherals to obtain a scalable architecture for high capacity. Fig. 2.b shows the array as in [5] and Fig. 2.c shows the complete array with the peripheral circuits. [5] describes the complete operation of the gate-controlled thyristor for all the modes, program, read and hold. Since this device has 3 gate controls and 1 bitline control, each operated at a high voltage, the peripheral circuits comprise 4 high voltage decoder which controls the gate and the bitline of the 3D DRAM array as shown in Fig. 2.c. The high voltage circuit is designed in the 3D NAND Flash-like CMOS-under-array (CUA) technology at 32 nm process node. Along with that, a few more necessary circuit blocks such as decoder level-shifter, control logic etc. are modeled in an advanced technology node of 7 nm and interconnected with the 3D DRAM array tier with Cu-Cu hybrid bonding. For evaluating the complete design, the power-performance-area (PPA) is calculated. For the array, the parasitic resistance and capacitance of bitlines and wordlines are modeled to obtain the energy consumption and the wire delay. For this work, we chose the minimum feature length for 3D DRAM to be 30 nm and the number of bitlines in each subarray to be 128 for the device dimensions described in [10]. A single device dimensions are $21F^2$, which after normalizing by number of layers gives $21F^2/128$. This gives us a bit cell area of 147 nm^2, which is nearly 6× better than commercially available DRAM. Based on the total memory requirements of the Llama2-13b model, the total storage is assumed to be 100 GB with a total area of 113 mm^2. For the peripherals, the PPA is calculated using NeuroSim [11]. Table II shows the computed PPA for 3D DRAM compared to conventional HBM [12] and 3D NAND Flash [13]. The bandwidth is reported assuming that data is being accessed through multiple channels at once. The number of channels can be much higher owing to advanced packaging techniques like Through Silicon Vias (TSVs), Cu-Cu hybrid bonding etc. [14]. Along with that, in most machine learning models, the data access pattern is fixed, hence reading multiple channels at once is possible. The read and write latency per bit of 3D DRAM is much faster as multiple bits can be read at the same time, owing to high-density interconnects between the CMOS tier and the 3D DRAM array tier. Since the device has a retention time of 100 ms, refresh is assumed with a total standby power consumption of 40 pW/bit. To refresh the complete 3D DRAM chip of storage 100 GB, the total refresh power consumption is 34.4 W.

Table II: 3D DRAM PPA compared to HBM and 3D NAND Flash

	HBM2 [12]	3D NAND Flash [13]	3D stackable DRAM
Read Energy (pJ/b)	3.9	1.3	2
Write Energy (pJ/b)	3.9	21	2
Read Bandwidth (GB/s)	1900	0.5	4000
Write Bandwidth (GB/s)	1900	0.2	4000
Bit cell size (nm^2)	1040	50	153

C. 3D heterogeneous integration for LLM accelerator

Next, a 3D heterogeneously integrated accelerator is designed by stacking 3D DRAM with logic dies, as shown in Fig. 4. The logic die consists of digital systolic array architecture where the processing element (PE) includes a power-gated sparsity-aware multiply-and-accumulated (MAC) unit [15] to handle unstructured sparsity. To obtain the optimal dataflow for the Llama2 model and further get the

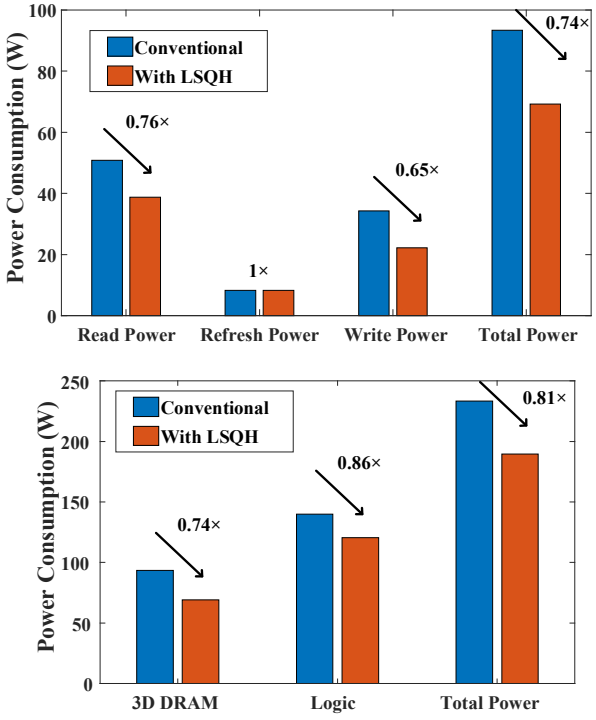

Fig. 6. Power consumption breakdown for various components, with (red) and without (blue) applying the LSQH algorithm. A significant drop in energy consumption is obtained in the DRAM write power to reduction in the size of total intermediate data.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on February 03,2025 at 09:49:11 UTC from IEEE Xplore. Restrictions apply.

cycle count for the number of MAC and the number of memory access for both the L1 and the L2 cache, the MAESTRO [16] and GAMMA [17] tools are used to design a single tile. A single tile maps one decoder layer (as shown in Fig. 1) of the Llama2 model to the hardware, as shown in Fig. 4.a. The tile is repeated 40 times to run all the 40 layers of Llama2 in parallel and a pipelined way to improve the throughput. To reduce the data communication energy and latency, a 3-tier 3D stacked accelerator is designed, as shown in Fig. 4.c. Different tiers of the accelerator are designed so that the total area is balanced. The top tier of 3D DRAM takes nearly 113 mm^2 area. The bottom 2-tier digital systolic array accelerator is designed in the 3 nm logic process, and the PPA of individual components is scaled using the latest NeuroSim V1.4 [11]. Each tier has a total area of 115 mm^2, by mapping 20 tiles on each tier. The obtained data access pattern is combined with the energy consumption of individual components to get the final PPA. The number of interconnects required between 3D DRAM and logic tiers is 37k. To minimize the overhead due to interconnects, both the logic tiers are interconnected using hybrid bonding with bond density of 185/mm^2 and the 3D DRAM is interconnected using TSV with a via density of 370/mm^2. The logic tiers are run at 1 GHz frequency.

Table III: PPA for Llama2-13b compared to GPU baselines.

	3D DRAM-based accelerator		Conventional GPUs [4]	
	Without LSQH	With LSQH	NVIDIA A100	NVIDIA H100
Throughput (tokens/s)	173k	163k	1280	1978
Power consumption (W)	233	193	~250-400	~500

IV. Results

In this work, we perform analysis for the Llama2-13b model for accuracy and PPA. The PPA estimates are obtained from system-level simulations performed using tools described in section III.C. Table III shows the obtained performance for the Llama2 model, compared to GPUs. The two-step process described in LSQH of first performing matrix-matrix multiplication for quantized inputs followed by full precision calculation for significant outputs results in an additional overhead of running the same layer twice as the hardware used for both cycles, which results in a higher latency for the LSQH algorithm. Table IV shows the breakdown of latency due to each operation described in Fig. 1. Here since the latency due to QK^T calculation is much smaller compared to the total latency of other layers, dedicated hardware is not designed, and the existing hardware is used to do the two-step process. The latency overhead incurred is only 6%. The total throughput of the system is 160 tokens/s. The total latency of a single layer is 25 ms, with a total end-to-end latency for text generation to be 1 s. Fig. 5 shows the trade-off between the accuracy and the energy consumption by varying the amount of sparsity. The intermediate dimensions for the QK^T calculation are 4096×4096×40 with 16-bit precision per layer. Since all the layers are being run at once for higher throughput, the total amount of intermediate data without sparsity is 50 GBs. Since the overall latency is smaller than the retention time of the device, refresh of intermediate data is not required and only the model parameters are refreshed to reduce the energy consumption. Fig. 6 shows the breakdown of the total energy consumption, without the introduction of sparsity. The energy consumption due to 3D DRAM read/write energy is very high, to read/write the intermediate data. After applying LSQH, the trade-off between accuracy drop and power consumption is observed. The write energy consumption reduces by 35% with the introduction of sparsity in Fig. 5.b in the last 17 layers and the overall energy savings are nearly 20%. The accuracy drop by introducing sparsity in the last 17 layers across 3 popular benchmarks, PIQA, HellaSwag, and ARC-e, compared to the pre-trained llama2-13b [3] is less than 1% as seen from the figure. Here, we assume naïve mapping of unstructured sparsity, which stores the address of the non-zero values along with the values itself. Storing the complete address requires 20 additional bits based on the size of the intermediate matrix. This requires the minimum sparsity to be at least 65% in a layer, for LSQH to be beneficial. Next, we perform scaling of the model size to a large model, assuming similar trends in sparsity. Fig. 7 shows the scaling of PPA with an increase in the model size by scaling the number of attention heads, context length, etc. To further evaluate the accelerator performance on larger models, we scale the model size, by increasing the number of layers in the architecture. Other parameters such as the number of attention heads etc. are assumed to be the same. For simplicity, we assume the sparsity ratio to be the same, i.e., last 43% of the layers are sparse. Fig. 7 shows the scaling of latency and energy consumption for the designed hardware with model size, with and without LSQH. The power and latency increase linearly with the number of layers. Our current design can only provide scaling up to 70 layers, with a model size of 45.5 GB. Further scaling of model size will be part of the future work.

Table IV: Latency of each compute layer

Compute	Latency (ms)
Q, K and V	4.65
QK^T	1.38
softmax(QK^T) × V	3.46
Output projection	1.55
Up projection	4.03
Down projection	4.03
Gate projection	4.60

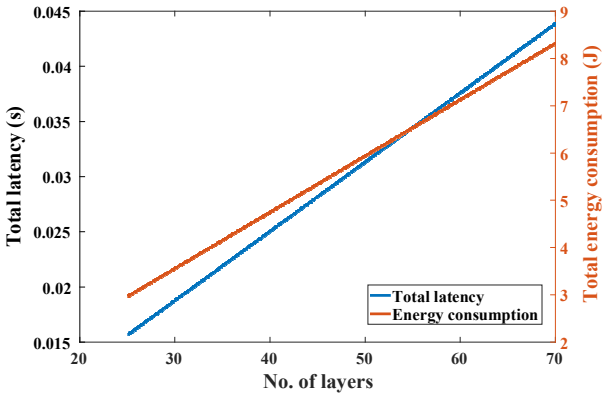

Fig. 7. PPA vs model size scaling

V. Conclusions

In this work, we design a 3D heterogeneously integrated LLM accelerator using 3D stackable DRAM and a digital systolic array. The 3D stackable DRAM offers the larger capacity required for LLMs. Evaluations are performed for the accelerator. Further, software-hardware co-design is performed to introduce sparsity without a significant drop in accuracy. Finally, an accelerator with an energy efficiency of > 25 TOPS/W and an area efficiency of > 14.4 TOPS/mm^2 is obtained. The model size is scaled with the number of layers to obtain the trend of power consumption with model size.

Acknowledgment

This work is supported by the CHIMES center and the PRISM center of the SRC/DARPA JUMP 2.0 program.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on February 03,2025 at 09:49:11 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] T. Brown *et al.,* "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901

[2] J. Kaplan *et al.,* "Scaling laws for neural language models." *arXiv preprint arXiv:2001.08361* (2020).

[3] H. Touvron *et al.,* "Llama 2: Open foundation and fine-tuned chat models." *arXiv preprint arXiv:2307.09288* (2023).

[4] NVIDIA NeMo framework, https://docs.nvidia.com/nemo-framework/user-guide/latest/performance/index.html

[5] W. -C. Chen et al., "A 3D Stackable DRAM: Capacitor-less Three-Wordline Gate-Controlled Thyristor (GCT) RAM with >40 μ A Current Sensing Window, >1010 Endurance, and 3-second Retention at Room Temperature," International Electron Devices Meeting (IEDM), 2022

[6] J. W. Han et al., "Ongoing Evolution of DRAM Scaling via Third Dimension -Vertically Stacked DRAM -," IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), 2023

[7] Y. Luo et al., " H3D-Transformer: A Heterogeneous 3D (H3D) Computing Platform for Transformer Model Acceleration on Edge Devices," ACM Transactions on Design Automation of Electronic Systems.

[8] Y. Leviathan et al.. "Fast inference from transformers via speculative decoding." International Conference on Machine Learning. PMLR, 2023.

[9] L. Gao et al., "A framework for few-shot language model evaluation," Zenodo, 2021. https://github.com/EleutherAI/lm-evaluation-harness

[10] W. -C. Chen et al., "A Simulation Study of Scaling Capability toward 10nm for the 3D Stackable Gate-Controlled Thyristor (GCT) DRAM Device," IEEE International Memory Workshop (IMW), 2023.

[11] X. Peng et al., "DNN+NeuroSim: An End-to-End Benchmarking Framework for Compute-in-Memory Accelerators with Versatile Device Technologies," IEEE International Electron Devices Meeting (IEDM), 2019. https://github.com/neurosim/DNN_NeuroSim_V1.4

[12] M. O'Connor et al., "Fine-grained DRAM: energy-efficient DRAM for extreme bandwidth systems," IEEE/ACM International Symposium on Microarchitecture (MICRO-50 '17), 2017.

[13] B. Kim et al., "28.2 A High-Performance 1Tb 3b/Cell 3D-NAND Flash with a 194MB/s Write Throughput on over 300 Layers" IEEE International Solid-State Circuits Conference (ISSCC), 2023.

[14] F. Sheikh, et al., "2.5D and 3D Heterogeneous Integration: Emerging applications," in IEEE Solid-State Circuits Magazine, 2021.

[15] H. Jiang et al., "Benefits and costs of power-gating technique," 2005 International Conference on Computer Design, 2005.

[16] H. Kwon et al., "MAESTRO: A Data-Centric Approach to Understand Reuse, Performance, and Hardware Cost of DNN Mappings," IEEE Micro, 2020.

[17] S.-C Kao et al., "GAMMA: automating the HW mapping of DNN models on accelerators via genetic algorithm," International Conference on Computer-Aided Design, 2020.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on February 03,2025 at 09:49:11 UTC from IEEE Xplore. Restrictions apply.