LLMMMM: Large Language Models Matrix-Matrix Multiplications Characterization on Open Silicon

Louis Ledoux*†, Marc Casas*†
*Barcelona Supercomputing Center, Barcelona, Spain
†Universitat Politècnica de Catalunya, Barcelona, Spain
E-mail: {louis.ledoux,marc.casas}@bsc.es

Keywords—Large Language Models (LLM), Transformers, Generative Pre-Trained (GPT), Matrix-Matrix Multiplications, Floating-Points, arithmetic, ASIC, Open-Source Silicon (OSS)

I. Extended Abstract

A. Introduction

GPT transformers are useful for various applications, offering significant advancements in natural language processing tasks. However, their operational costs are substantial has shown in prior work which highlights the financial implications of deploying these models [1].

Essentially, matrix-matrix multiplications (MMM), with their intensive data movement and manipulation of arithmetic weights, underscore the computational demands of these architectures. Naturally, these observations are also found in recent efforts within the research community, which have concentrated on devising specialized formats and algorithms aimed at mitigating these costs. These innovations include reducing bit-width exemplified by Machine Learning eXchange (MLX) formats (essentially small floats), specialized hardware such as TPUs' systolic arrays, model pruning of up to 40%, and more recently, ternary and binary LLMs (see BitNets [2]).

We introduce a generator of ASIC kernels agnostic to the PDK of MMM units for emerging and small floating-point formats, followed by the evaluation of such units. Concretely, our contributions include the automated generation of circuits for any floating-point format with automated pipelining, a systolic array architecture proposal—these two combined form the foundation of MMM units, a framework to automate the translation from high-level language (Python) to silicon for such matrices, the generation of 4 arithmetic formats $\times$ 2 accumulator configurations $\times$ 4 PDKs $= 32$ chips, and their performance and efficiency evaluation, all provided as open source.

B. Asynchronous and Parallel Compilation

The necessity for rapid generation of specialized circuits primarily arises due to the challenges posed to the established laws of computer science, including Dennard scaling, Moore's law, and the emergence of issues like power walls and the dark silicon era. An advanced and emerging solution that keeps pace is Open Source EDA, supported by its community [3]. In adopting this approach, we build our own open-source tool, accessible online[1]. Figure 1 illustrates this framework, which facilitates the creation of multiple independent design entries, transitioning from high-level Python descriptions to silicon GDS outputs.

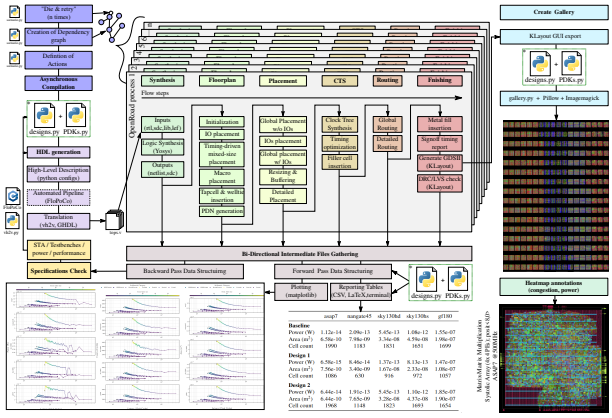

Fig. 1. Schematic Overview SUF: Centralized Management of Asynchronous OpenROAD Forks, Derived from Dependency Task Graphs. This illustration also encapsulates the extended capabilities ranging from Code Generation without manual RTL Writing to Advanced Plotting and Visualization Features.

C. Functional and Performance Specifications

We define and assess four computational formats distinguished by their compactness and mathematical attributes (dynamic range and precision). These include Nvidia's e4m3 and e5m2, and the tapered formats, posit4 (es=0), and posit8(es=2) [4], [5]. Another significant aspect of our work is the proposal of two variations of internal paths for each of these formats. Internally, we execute the dot product as a fused operation (without rounding) in a fixed accumulator with varying boundaries (bit weights for lsb/msb/ovf). These variations, named α and β, are configured as follows: $(\text{ovf} = 2, \text{msb} = 3, \text{lsb} = -2)$ for an aggregate of 8 bits, and $(\text{ovf} = 5, \text{msb} = 5, \text{lsb} = -5)$ for the 16-bit model. The

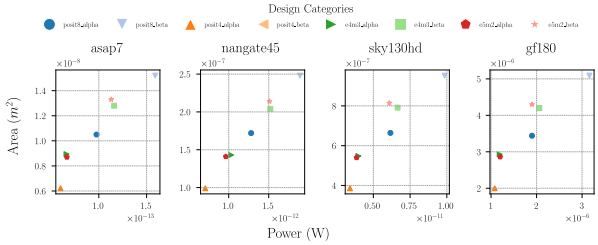

Fig. 2. Area vs. Power for 28 different MMM units combining diffrent computer format, accumulator sizes, and Process Development Kits.

[1]https://github.com/Bynaryman/SUF

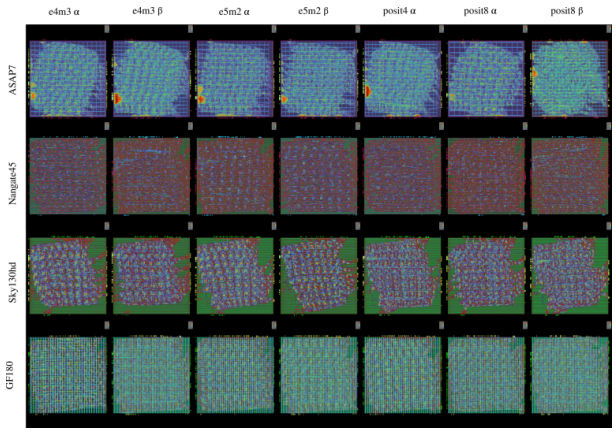

Fig. 3. Overview of the GDS layout of the 28 generated MMM units categorized by Arithmetics vs. PDKs. Each layout has a congestion heatmap which helps in the visualization of Processing Elements.

weights distribution of the embedding layers in the Llama-2-7b model dictates these boundaries. All Systolic Arrays of this work are set to $8 \cdot 8 = 64$ PEs, which ends the defintion of the *Functional* specifications set.

We augment this set with *Performance* specifications across four Process Development Kits (PDKs), specifically GF180, Sky130hd, nangate45, and ASAP7, all of which remain open source. The assessment of multiple PDKs facilitates the validation of design scalability, obviating the need for often imprecise manual scaling techniques.

D. Results

All configurations successfully "taped-out" within an hour, with the exception of posit4β, culminating in a total of 28 produced chips. Figure 3 illustrates the 28 systolic arrays, each a 2D mesh, arranged across arithmetic units versus PDK dimensions. Figure 2 details the performance metrics for the MMM units, indicating that beta costs exceed those of alpha, as expected. Posit arithmetic units incur higher costs relative to their counterparts of equivalent size and accumulation capabilities. Nonetheless, this observation warrants further investigation into accuracy (designated as future work). The configurations e5m2 and e4m3 exhibit minor differences and are positioned in closely situated clusters, reflecting their similar hardware characteristics.

Figure 4 zooms in one of the chip, specifivally the e4m3 arith with beta accumulator for the Sky130 nanometer high density PDK. The picture allows to see the PEs thanks to routed congestion heatmap.

E. Conclusions

Overall, by the mean of a custom open source framework, we are able to generate MMM units for several arithmetic specifciations and ttechnology nodes. We show the performance metrics of 28 distinct chips that have been generated within an hour, which is possible thanks to open source EDA tools.

As a future perpectives, we need to correlate the performance metrics measured with accuracy metrics in order to find the best entry in the vast accuracy/energy efficiency design space exploration. In light of these promising results, we ecnourage researchers to interact with our tool.

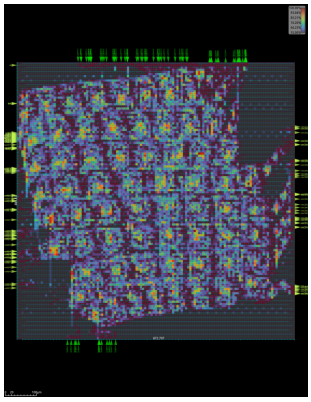

Fig. 4. Zoom-in view of the e4m3 beta chip with fine-tunes congestion heatmap allowing to clearly distinguish the $8 \times 8 = 64$ PEs.

F. Biography

Louis Ledoux, originating from a comprehensive computer science background in Rennes (Bretagne, France), has transitioned towards a hardware focus. His journey began with a Bachelor's degree, followed by a Master's in Computer Science, culminating in a one-year internship in 2017, where he explored FPGA virtualization in the cloud. Since 2018, Louis has been engaged in a PhD in computer arithmetic at the Universitat Politècnica de Catalunya and the Barcelona Supercomputing Center, in Barcelona, Spain. His main focus are hardware implementations to address numerical requirements sparsity in HPC workloads.

II. Acknowledgment

Marc Casas has been partially supported by the Grant RYC-2017-23269 funded by MCIN/AEI/10.13039/501100011033 and by ESF Investing in your future. This research was supported by grant PID2019-107255GB-C21 funded by MCIN/AEI/ 10.13039/501100011033. Els autors agraeixen el suport del Departament de Recerca i Universitats de la Generalitat de Catalunya al Grup de Recerca "Performance understanding, analysis, and simulation/emulation of novel architectures" (Codi: 2021 SGR 00865).

References

[1] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of bloom, a 176b parameter language model," 2022.

[2] 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," 2024.

[3] T. Ajayi, V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny, A. B. Kahng, M. Kim, J. Lee, U. Mallappa, M. Neseem *et al.*, "Toward an open-source digital flow: First learnings from the openroad project," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–4.

[4] P. Micikevicius, D. Stosic, N. Burgess, M. Cornea, P. Dubey, R. Grisenthwaite, S. Ha, A. Heinecke, P. Judd, J. Kamalu, N. Mellempudi, S. Oberman, M. Shoeybi, M. Siu, and H. Wu, "FP8 Formats for Deep Learning," arXiv, Tech. Rep. arXiv:2209.05433, Sep. 2022, arXiv:2209.05433 [cs]. [Online]. Available: http://arxiv.org/abs/2209.05433

[5] J. L. Gustafson and I. T. Yonemoto, "Beating Floating Point at its Own Game: Posit Arithmetic," *Supercomputing Frontiers and Innovations*, vol. 4, no. 2, pp. 71–86, Apr. 2017, number: 2. [Online]. Available: https://superfri.org/index.php/superfri/article/view/137