RETROSPECTIVE: Neurocube: a programmable digital neuromorphic architecture with high-density 3D memory

Duckhwan Kim[1], Jaeha Kung[2], Sek Chai[3], and Saibal Mukohpadhyay[4]

[1]Meta Platforms, Inc.
[2]School of Electrical Engineering, Korea University, Seoul, South Korea
[3]Latent AI, Inc.
[4]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

I. Motivation behind Neurocube

“Deep learning” (DL) led to tremendous advancement in Artificial Intelligence (AI), showcasing its ability to solve complex non-linear problems like image recognition. The field of DL was rapidly advancing in the years leading to our Neurocube research. Increasingly complex multi-layer Convolutional Neural Networks, Long-Short-Term-Memory and Recurrent Neural Networks were being investigated for various classes of computer vision problems. The DL models were being used in problems beyond computer vision. For example, there was tremendous anticipation on what will be the outcome of the match between AlphaGo, a DL model trained for the game of Go, and Sedol Lee, the world Go master. Despite the vast number of potential moves in Go, many anticipated Lee’s victory. But ultimately, in March 2016, AlphaGo won with a score of 4-1, relying on 1,200 CPUs and 176 GPUs.

The advancing DL was thought of as mainly an algorithmic challenge and Graphics Processing Units (GPU) was the defacto computing platform. Hardware acceleration for DL was relatively new. Our team including two Ph.D. students (Dr. Duckhwan Kim and Prof. Jae-ha Kung), one industrial advisor (Dr. Sek Chai from SRI), and two faculty (Profs. (late) Sudhakar Yalamanchili and Saibal Mukhopadhyay), were inspired by many of the early, now seminal papers on compute substrate, arithmetic engines, data-flow models, and network-on-chip for DL. However, we wondered how we can maintain compute efficiency as the model complexity grows as increases memory accesses. It was well known that latency and energy demand for memory access is far higher than that of computing. We predicted as the model complexity grows, memory access will become a major bottleneck for efficient DL hardware. In fact, the number of parameters (and operations) of today’s DL models is drastically higher than that of the models a decade ago. For example, “Billion” is now the unit of count for the parameters in large language models, which people can freely experiment. As we look back, our prediction of parameter sizes expanding beyond the capacity of on-chip SRAM or eDRAM was correct, but the pace of growth has far exceeded our initial expectations.

II. Concept of Neurocube

During this time period, several new memory architectures were being investigated including Hybrid Memory Cube (HMC), High Bandwidth Memory (HBM), and Wide IO 2. In particular, both HBM and HMC were forms of 3D stacked DRAM and had the potential of 3D integration of logic and memory. Motivated by these trends, we envisioned a special-purpose 3D memory stack, where the bottom logic layer contains a computing fabric to accelerate mathematical operations in deep learning models. We realized that such a design could satisfy the growing need for the memory capacity and bandwidth required for large models. The vision led to the conceptual architecture of a deep learning accelerator with 3D stacked DRAM. We don’t remember who among us came up with the name Neurocube (Neural network within a 3D Memory Cube), but we wonder whether the paper would have received the attention it did with a different name! As we look around the industry today, we see the critical role memory has played in ML acceleration. For example, HBM is heavily utilized in AI training chips, GPUs or dedicated ASICs and often determines their end-to-end performance. Moreover, Samsung recently introduced a process-in-memory architecture using an HBM stack, claiming a 70% energy savings compared to existing HBM in data-intensive applications like AI. The exact Neurocube architecture may not (yet!) exist, but the memory-centric view of the AI accelerators has been materialized.

After defining the concept architecture, the first hurdle was the lack of open-source and standardized simulators for evaluating DL accelerators in 3D. We (to be precise, the two graduate students) had to write a simulator from scratch. Coming from an Electrical Engineering background, interestingly, we started using MATLAB to develop the first version of the Neurocube simulator for a simple MLP model. Our initial excitement soon gave away to frustration as MATLAB was not suitable for implementing a performance simulator capable of addressing data movement inefficiencies. We switched to Python as it offered more flexibility and easy to code but we grossly underestimated the scale of the experiments needed for the ISCA submission. When presenting the paper at conferences, the most common question we received was why not use SystemC. In retrospect, we should have considered large-scale simulation challenges from day one. Eventually, for our doctoral thesis, we decided to switch the simulator to SystemC. During the course of this research, we realized the need for having an open-source, potentially standardized, simulation environment to quickly evaluate new special-purpose hardware architectures of ML. We are excited to see the tremendous progress made in the Computer Architecture community in this direction in recent years.

Another interesting question was selecting between HBM and HMC. Although we had various technological reasons to choose HMC, ultimately, the deciding factor was that there were more academic resources available for HMC about the feasibility of implementing compute functions on the logic die. During the presentation, we remember being asked by audiences from Samsung and SK Hynix about why we assumed HMC instead of HBM. Our answer often was "there is no academic resources for HBM". After the publication of that paper, Georgia Tech, under the leadership of Prof. Yalamanchili, established a larger Near Memory Computation consortium. Micron had generously provided educational purpose development boards with HMC and FPGA which helped us prototype various near-memory acceleration concepts. Although we never prototyped Neurocube on those boards (and in retrospect, we should have), this was a valuable opportunity that was not easily available within an academic setting during this period. HMC may no longer be under active development (to the best of our knowledge), but the academic resources and development boards with HMC had played a critical role in advancing the concept of near-memory computing.

A central concept in Neurocube was memory-centric neural computing (MCNC). The core idea was to have a local controller embedded within each memory vault to orchestrate the data flow between memory and compute. A predefined memory layout was generated for the tensor for each layer. A finite-state machine (FSM) embedded in each of the local memory controller generated addresses and controlled data flow from the memory, instead of processing engines (PEs) generating memory access requests. We observed that distributed control provided important concurrency and hence, performance advantages. However, even today, we see that the DL accelerators still follow a traditional approach, where a PE contains RISC-V or ARM cores with custom instruction processing (CIP) units issue memory access requests. The process is not dramatically different from any non-DL application. We believe there is room to optimize the memory access by leveraging DL application properties such as predetermined memory layout, address sequence, and so on. In fact, our recent work has shown that the concept of distributed memory control is applicable to many domains beyond DL. In retrospect, we realize that further research should have been conducted on distributed control aspects of the memory-centric computing notion in Neurocube. For example, a better understanding of the buffer overhead when a single tensor has to be partitioned into multiple DRAM dies or a further optimization the NoC routing process to leverage the predetermined data-layout and access sequence. Efficient memory control and data flow remain a challenge in today's DL accelerators. In this regard, one of our regrets is that, as Neurocube was one of the initial papers on DL accelerators with distributed memory controllers and 3D stacked DRAM, it would have been beneficial to release the simulators as open-source. Researchers could have applied different DRAM models and explored PE and NoC designs for specific DRAM interfaces in the DL accelerator. The open-source simulators would have also generated more ideas on the distributed memory controller for DL applications.

III. Summary

In summary, since the publication of this paper, the parameter count of DL models have increased exponentially making high-density memory and its bandwidth crucial factors in DL accelerators today. DL models used in today's ranking systems and large language models have already surpassed the memory capacity limits of the latest GPU. We, therefore, expect to see the emergence of 3D stacked processors designed with a process-in-memory approaches envisioned in the Neurocube paper. We look forward to the research that Neurocube has inspired, and to the new AI architecture that was able to leverage our work.

We consider fortunate to have had the honor of presenting the concept of Neurocube at ISCA. For the graduate students, the connections made at the conference have continued to shape their career after graduation. Dr. Kim joined NVIDIA followed by Meta where he continued to work on AI acceleration. Dr. Kung, after graduation, become a faculty and contributing to the field of ML accelerators. Dr. Chai's pursual of machine learning research eventually led to founding Latent AI, a startup focusing on model optimization, including compression. For Prof. Mukhopadhyay's lab at Georgia Tech, the Neurocube was the first-ever ISCA submission. The paper made a major impact on their follow-on research on domain-specific acceleration and AI models. Now, having the opportunity to reflect and write this retrospective, we are filled with deep gratitude. It is a truly rare privilege, and at the same time, it brings a sense of humility as we revisit the paper we wrote eight years back.

This article would not be completed without expressing our sincere gratitude to Prof. Sudhakar Yalamanchili. He was a visionary to foresee that near-memory processing will become critical for the continuing performance growth in computing systems. He not only continued to the field himself but also inspired many of us to explore this nascent field ten years back. We were initially planning to submit the paper to Design Automation Conference (DAC), but Prof. Yalamanchili convinced us ISCA will be the appropriate home of this work. As we write this article, we could not agree more. We sincerely wish he was here with us to celebrate this amazing journey of Neurocube from a concept drawn in the whiteboard to a retrospective article in ISCA.