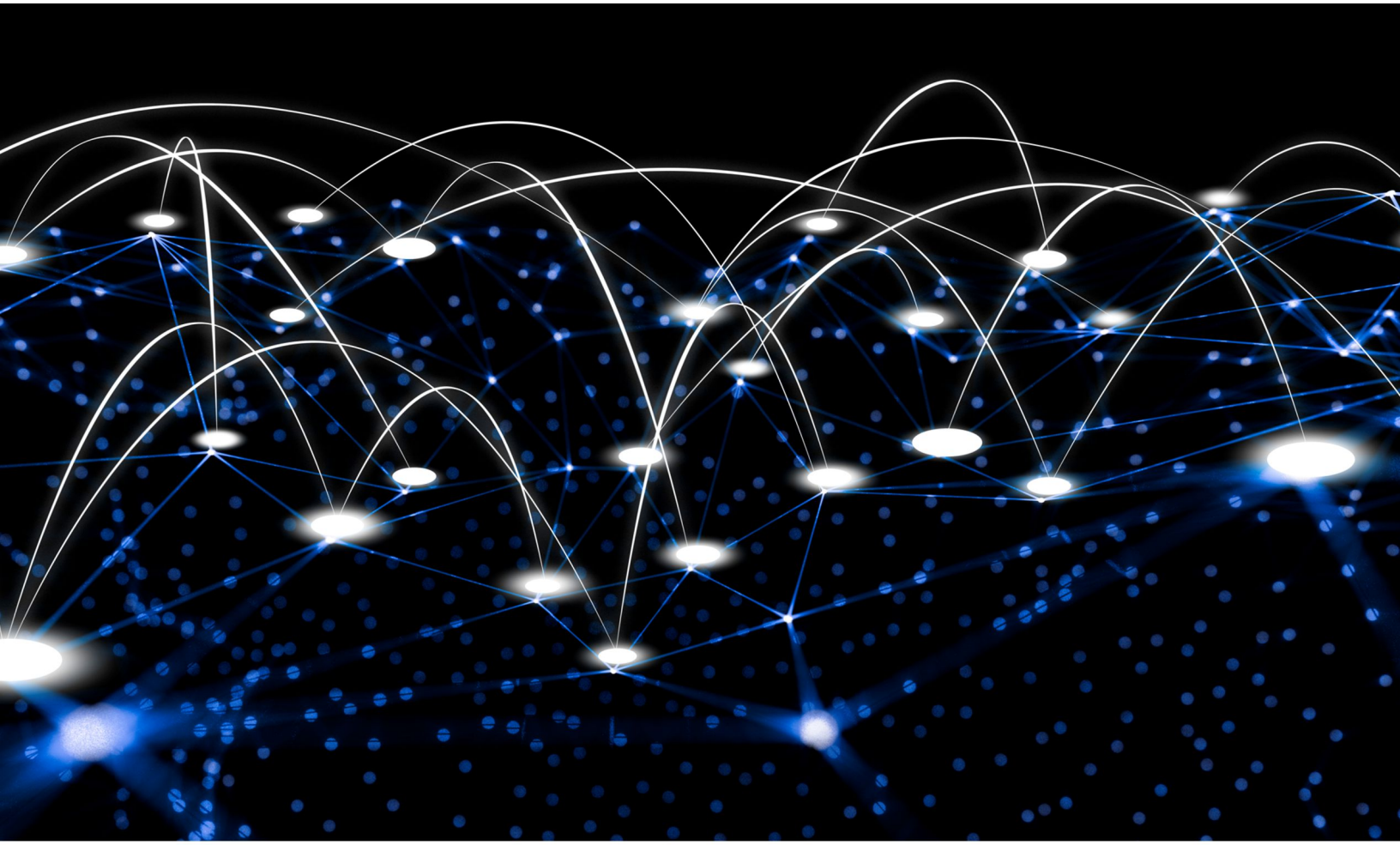
In-Place Associative Computing:

A New Concept in Processor Design



www.gsitechnology.com

Rev. 1.00, 11/2018

Abstract

Machine learning is coming of age, and its subfield, deep learning, is set to reshape entire industries, from healthcare to online retail. Consequently, demand is growing for more powerful processors that are able to handle the ever-increasing amounts of data and complex calculations associated with deep learning applications, such as image recognition and natural language processing.

The Associative Processing Unit (APU) is GSI Technology's patented processing technology and a new breed of processor that computes in-place—directly in the memory array—thus significantly reducing the overheads associated with data movement. This gives the APU a performance edge as a hardware accelerator of similarity search applications.

This white paper introduces GSI's Associative Programming Unit (APU). The paper begins with an overview of existing processor design and shortcomings. It proceeds to a high-level description of the APU and its main memory block, followed by the advantages of this new design, and an overview of the architecture and software. The paper concludes with some performance benchmarks.

What's Wrong with Existing Processors?

For the past 50 years, the Von Neumann architecture has dominated computer design. Von Neumann-based computers feature separate processing and memory units, and data is processed serially—the processor reads data from a specific memory address, processes it, and writes it back to memory, one instruction at a time.

Von Neumann-based CPU performance improved for years both in terms of processing speed and memory density. However, since around 2006, improvements in clock speed—an important indicator of computer performance—has started to flatten out. All that power and memory is no longer delivering the desired improvements in performance.

Furthermore, the Von Neumann model, with its separate processor and memory units, has a significant limitation—improvements in data transfer rates between the memory and processing

Rev. 1.00, 11/2018

unit have not matched the improvements seen in processor speeds and memory densities. This is a significant limitation because these data transfers rates are typically the most bandwidth-costly part of computation.

The introduction of local memory (cache) and processors with multiple cores working in parallel have pushed performance boundaries even further. However, even with these designs, local memory must regularly bring data from the global memory, thus exacting a bandwidth cost.

Designs featuring multiple processors require inter-processor coordination, and furthermore, each individual processor can only handle a relatively smaller load.

A case in point is the GPGPU, a processor strongly associated with machine and deep learning applications. The design features thousands of small processors, and thus each processor can only handle a relatively small amount of data. Bringing this data together is complex and bandwidth costly. The GPGPU also features different computation blocks, each designed for a specific operation: FP64, INT, FP32, and TENSOR CORE to name a few. Therefore, overall processing power for specific operations is more limited.

In general, the GPGPU architecture is geared towards applications that use matrix multiplication; less so for other deep learning applications, such as similarity search.

Google's TPU also features separate memory and processing units. The TPU's main computing power is provided by the scalar, vector, and matrix units (also known as MXU). Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU's inputs and outputs are 32-bit floating point values, the MXU performs multiples at reduced bfloat16 precision.

Could there be a faster and more efficient processing technology designed specifically for deep learning applications?

Introducing the Associative Processing Unit

To meet the demand for faster deep learning processors, GSI Technology is developing a patented processing technology called the Associative Processing Unit (APU). The APU features massive parallel data processing, compute, and search in-place, directly in the memory array.

The APU's architecture enables program instructions to be broken down into basic operations that can be performed in parallel. The microcode orchestrates these parallel operations, utilizing Single Instruction, Multiple Data (SIMD), allowing multiple processing elements to perform the same operation on multiple data points, simultaneously.

GSI's in-place associative computing technology eliminates bandwidth-costly data transfers between the memory and processor. Data is accessed by content (or value) and processed in-place in the memory array. The result is an orders of magnitude performance-over-power ratio improvement compared to conventional methods that use CPU and GPGPU (General Purpose GPU) along with DRAM.

The APU performs simple logic operations—with the aid of additional patented logic—directly on the bitline (an array of memory cells) of a standard SRAM memory array. Millions of bitlines become millions of processors.

By combining logical operations, the APU's microcode instructions can convey higher level arithmetic functions (e.g., ADD, MUL, DIV, and element-wise logical operations). The microcode can control all bitlines simultaneously, but also perform operations on a subset of the data, through conditional execution of the microcode, per bitline. The APU also has patented technology for performing neighborhood operations—such as convolutions—on a set of bitlines.

The APU Edge

The APU's in-place design delivers flexible processing power that can be fully utilized for a wide range of functions including add, subtract, multiply, divide, transpose, top-k, conv2d, maxpool, and SoftMax. The APU also allows fast data storage, retrieval, and search.

Rev. 1.00, 11/2018

This gives the APU a performance edge as a hardware accelerator of similarity search applications.

The following table summarizes the main differences between the APU and the current generation of CPU/GPGPU processors:

CPU / GPGPU	In-Place Computing (APU)
▪ Send an address to memory	▪ Search by content
▪ Fetch the data from memory and send it to the processor	▪ Mark in place
▪ Compute serially per core (thousands of cores at most)	▪ Compute in place on millions of processors (the memory itself becomes millions of processors)
▪ Write the data back to memory, further wasting I/O resources	▪ There is no need to write data back. The result is already in the memory
▪ Send data to each location that needs it	▪ If needed, distribute or broadcast immediately

Overview of APU Architecture

The APU solution is available in two forms:

1. **An APU Card**: A PCIe Gen3 x 16 add-in card, with a width of two PCIe slots. The APU card's DRAM memory capacity is up to 32 GB. The APU card will come with built-in FPGA and SW Firmware code, and SW Libraries for GNL, for similarity search.

2. **As Separate components:** Including APU ASIC, APU card schematics with PCB layout recommendation, FPGA code (Verilog and Executable code), firmware software, APU microcode, software libraries, and host driver.

APU Software Stack

The APU programming model consists of five distinct layers. Top down, these include machine and deep learning applications developed by GSI, as well as third-party services, such as data centers and cloud computing service providers; support for application development using third-party deep learning frameworks, such as TensorFlow; the GSI Numeric Library, which enables users to leverage the APU device for multi-dimensional array (a.k.a. Tensor) processing; the GSI Vector Math Library, a device-side library that provides the functional abstraction for using the APU for vector processing; and finally the APU Programming Language (APL) and the APL preprocessor for development of low-level logic processing of user data.

Performance Comparison for Similarity Search

At its core, similarity search performs as k-nearest neighbors (kNN), a rudimentary method which is computationally basic and consists of highly parallel distance calculations to return a global top-k sort. The fact that current architectures don't support kNN due to high memory bandwidth demands, gives the APU an edge as a hardware accelerator.

We replicated the *Billion-scale similarity search with GPU* study (associated with Facebook's FAISS). The study takes 10^9 records with feature vectors of 32 and 80 dimensions, the objective being to compute k-NN (k = 1 to 1024). The APU performed the task **10 times faster** than the GPU used in the original study.

Main Applications

As a hardware accelerator for similarity search, the APU can power applications such as:

- Content-based search for images and video
- Recommendation systems
- Data deduplication
- Natural language processing
- Computer vision, databases
- Computational biology
- Computer graphics

More Information

For more detail about the APU, please refer to the ***An Introduction to the APU Architecture*** whitepaper.

Rev. 1.00, 11/2018