UNTETHER AI

At-Memory Computation

The Winning Architecture for Deep Learning Inference

The Untether AI at-memory architecture for neural network inference delivers the highest performance and efficiency of any competing architecture. The Untether AI inaugural chip, runAI200, deploys at-memory computation in a spatial architecture designed around minimizing the data movement required for each multiply-accumulate (MAC) operation, thus minimizing power and latency. This is done by interleaving processing elements (PEs) into an SRAM array, with each PE physically abutted with dedicated SRAM. RunAI200 demonstrates the superiority of the at-memory architecture by achieving industry leading TeraOperations per second (TOPs) and TOPs per watt on an older 16nm process technology. Tunable frequency and voltage enable the user to handle compute intensive applications in Sport Mode with 502 INT8 TOPs in a 75 watt power envelope, or in Eco Mode to achieve over 8 TOPs per watt in a 47 watt power envelope. The Untether AI tsunAImi PCIe card, which features four runAI200 chips, achieves industry-best efficiency and a single-card record of 2 PetaOperations per second of performance in a 300 watt power envelope.

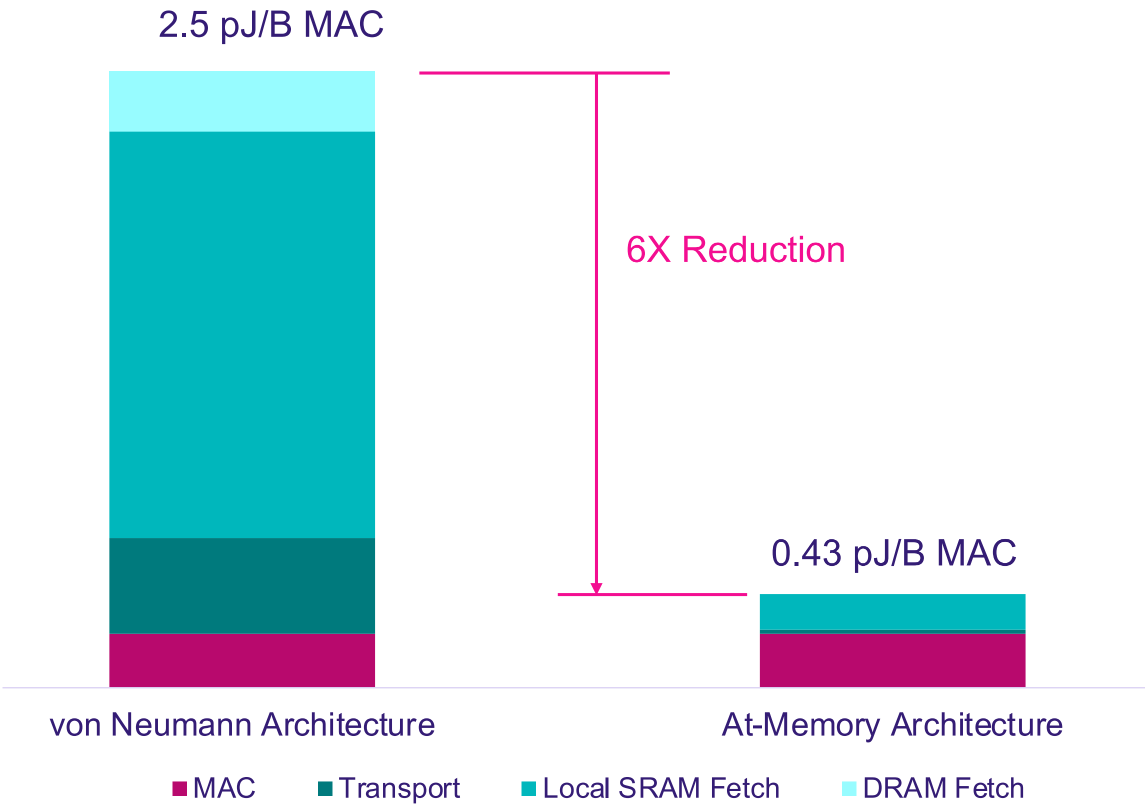
1. Introduction

We are amid the third wave of artificial intelligence investment, dubbed the Deep Learning Era. Like each of the previous waves, this one was started by the repurposing of existing hardware. By implementing what used to be a prohibitively expensive training algorithm on GPUs, researchers enabled deep neural networks to make the jump from the theoretical to the practical domain. Since then, the field of deep learning has grown exponentially. Deep learning practitioners are promising everything from self-driving cars to life-changing medical advancements, and although the results thus far are impressive, a clear trend has emerged. The success of deep neural networks is fueled by an insatiable hunger for compute power[1] . Every year, state-of-the-art benchmarks are broken by increasingly complex neural nets. The size of neural nets is growing parallel to the demand for applications powered by them. While GPUs and CPUs may be enough for training, the demand for inference at scale challenges these architectures in terms of power efficiency and latency. There are two main reasons for this.

First, neural net inference workloads are shaped differently from traditional ones. The old way involves a processor fetching a single, relatively small piece of data and doing a large portion of the desired and sometimes complex computation on it. Neural network inference flips this on its head; the computation itself may be relatively simpler, usually boiling down to some form of a dot product, but every step requires a massive amount of data being moved and processed in parallel.

Second, while transistors have shrunk in area by many orders of magnitude, the wire lengths have only shrunk linearly, and the overall size of high-end processors remains roughly the same. That means the energy used inside a chip has transitioned from being dominated by the transistors doing the computation to the wires that get the data to them.

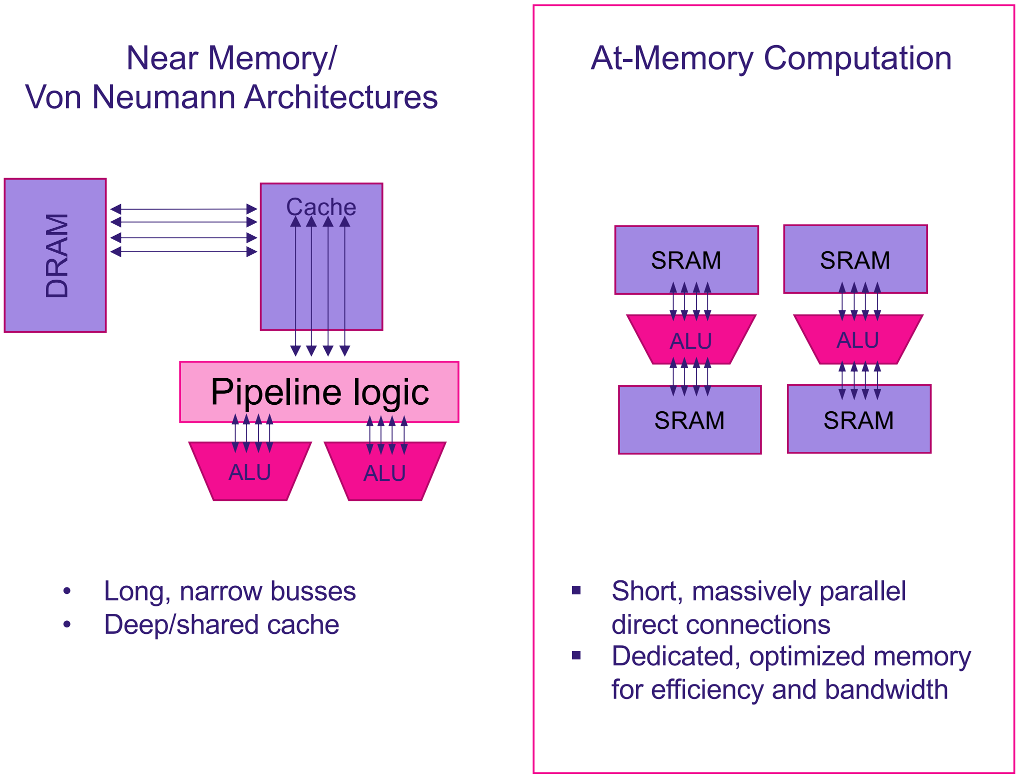
Given these challenges, the traditional von Neumann architecture of CPUs and GPUs is not very well suited for the compute requirements of neural net inference. Moving the large volumes of coefficients (weights) and data between memory and the processing element wastes a great deal of energy. Over 90% of the energy in these processors is wasted on data movement.



This is the foundational insight for the Untether AI at-memory architecture - minimizing data movement is the most important aspect for building a highly efficient neural net inference processor. Minimizing data movement creates an efficient overall system that can do more total work with lower overall latency.

2. The At-Memory Architecture

The namesake characteristic of the At-Memory Architecture is the physical connection between the processing elements “at the memory” that feeds them. Each processing element in our architecture is connected via short, wide busses to dedicated SRAM cells. Compare this to the long, narrow busses of traditional von-Neumann architectures found in CPUs and GPUs, where a processing element is fed data from a cache or external DRAM. By doing away with cache and external dram, we significantly reduce latency and power.



Near Memory/
Von Neumann Architectures

- Long, narrow busses
- Deep/shared cache

At-Memory Computation

- Short, massively parallel direct connections
- Dedicated, optimized memory for efficiency and bandwidth

Untether AI’s at-memory architecture was designed with a careful balance between compute intensity and memory capacity. As with everything in our architecture, distance and area are the dominant considerations. Area may be dedicated to memory or to computation, but as any given area becomes larger, the distance to move data between blocks on the chip becomes higher and therefore costs more energy.

We struck the right balance in runAI200 by interleaving over 260,000 processing elements into a 204MB SRAM array. Traditionally, a large block of SRAM is tied to a very fast central processing unit, and memory throughput speed is critical to feed that processing unit. Because each of the over 260,000 processing elements are directly connected to a small block of dedicated SRAM, the throughput of each individual block can be much lower while as a system still providing extremely high aggregate memory bandwidth. In addition, running standard SRAM cells at a lower speed allows them to be run at a lower voltage and cuts memory access energy costs.

Once coefficients and activations are loaded into SRAM, the distance the data needs to be moved is extremely short because the SRAM block and the processing element are physically abutted. In a traditional architecture, after the expensive memory read, the data must still flow through a high-speed memory bus with the associated energy cost of sending it that distance. The energy cost of high-speed SRAM and long fast buses means that even in the most advanced processes you see traditional architectures doing no better than 2.5 TOPs per watt, before factoring in the cost of DRAM access, compared to our architecture which delivers 8 TOPs per watt.

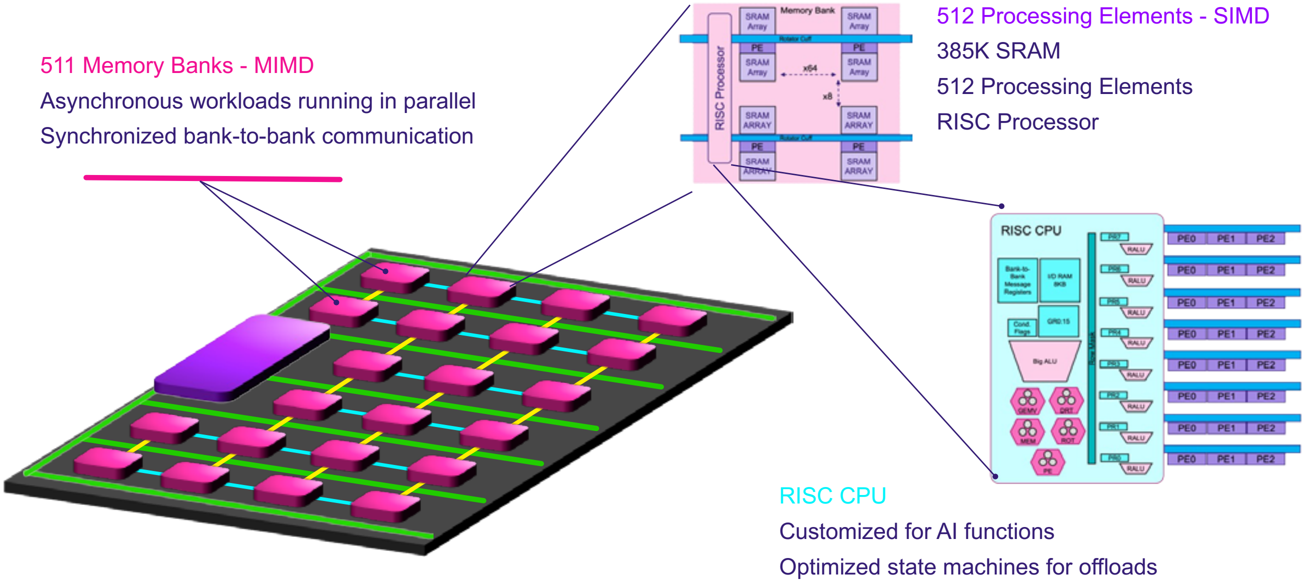
The energy cost for retrieving coefficient data gets even higher if you need to go off chip to DRAM. Most companies do not include this energy in their advertised calculations, but it is a large energy cost and DRAM access is essential to making use of their architecture. By maximizing the amount of SRAM available, and keeping the coefficients on chip, there is no external DRAM load and store, meaning that the advertised energy of our solution is the total energy it consumes.

3. Architecture Implementation

Overview

The at-memory architecture is implemented in the runAI200 device in the form of 511 memory banks. Each memory bank houses a custom RISC-V controller, 512 processing elements (PEs), and 385KB of SRAM. The banks are connected horizontally by a proprietary high speed, pipelined interconnect called the Rotator Cuff, and vertically by a mechanism called Direct Row Transfers. The chip connects to its host processor with a PCIe Gen4 x16 interface, which uses a pipelined bus (PBUS) to feed into each of the memory banks.

Untether AI's at-memory compute architecture is a "best of both worlds" approach in that it mixes multiple instruction, multiple data (MIMD) and single instruction, multiple data (SIMD) processing. MIMD allows for spatial optimization with 511 memory banks operating asynchronously, while sequential optimization is achieved through SIMD processing, with 512 process elements per memory bank executing on a single instruction.



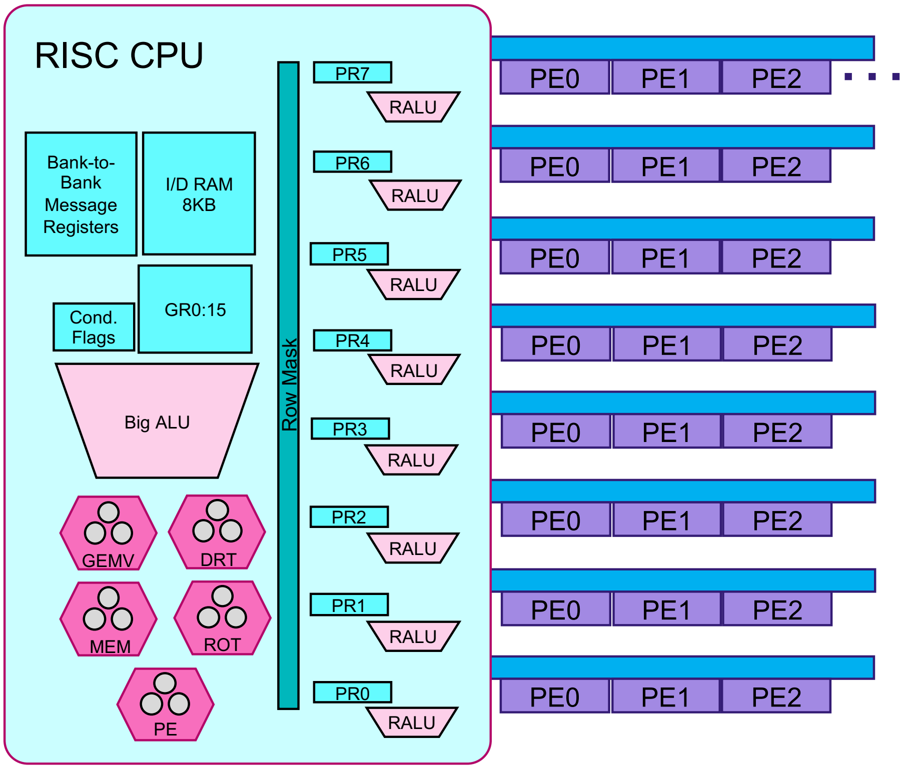
511 Memory Banks - MIMD
Asynchronous workloads running in parallel
Synchronized bank-to-bank communication

512 Processing Elements - SIMD
385K SRAM
512 Processing Elements
RISC Processor

RISC CPU
Customized for AI functions
Optimized state machines for offloads

RISC-V processor

Each memory bank has a custom 32-bit RISC-V processor built to accelerate neural networks. Its primary function is configuring a 2D SIMD array of PEs along with the PE memory array. The array is organized into 8 rows, with 64 PEs in each row. The processor also allows for bank-to-bank coordination, manages DMA and atomic operations with the host, and controls the special purpose state machines that drive the PE rows and data movement.



Message Registers enable nearest neighbor communication.
A “Big ALU” and 32-bit General Registers (GR) enable program counter and pre-fetch.
Special purpose state machines offload the processor to accelerate compute (GEMV, PE) and data movement (DRT, MEM, ROT).
The Row Mask disables individual rows, and with the PE mask, enables fine-grained control of the SIMD operations. Each row has a 32-bit Row ALU (RALU) and dedicated register, enabling aggregate operations like softmax.

The Processing Element

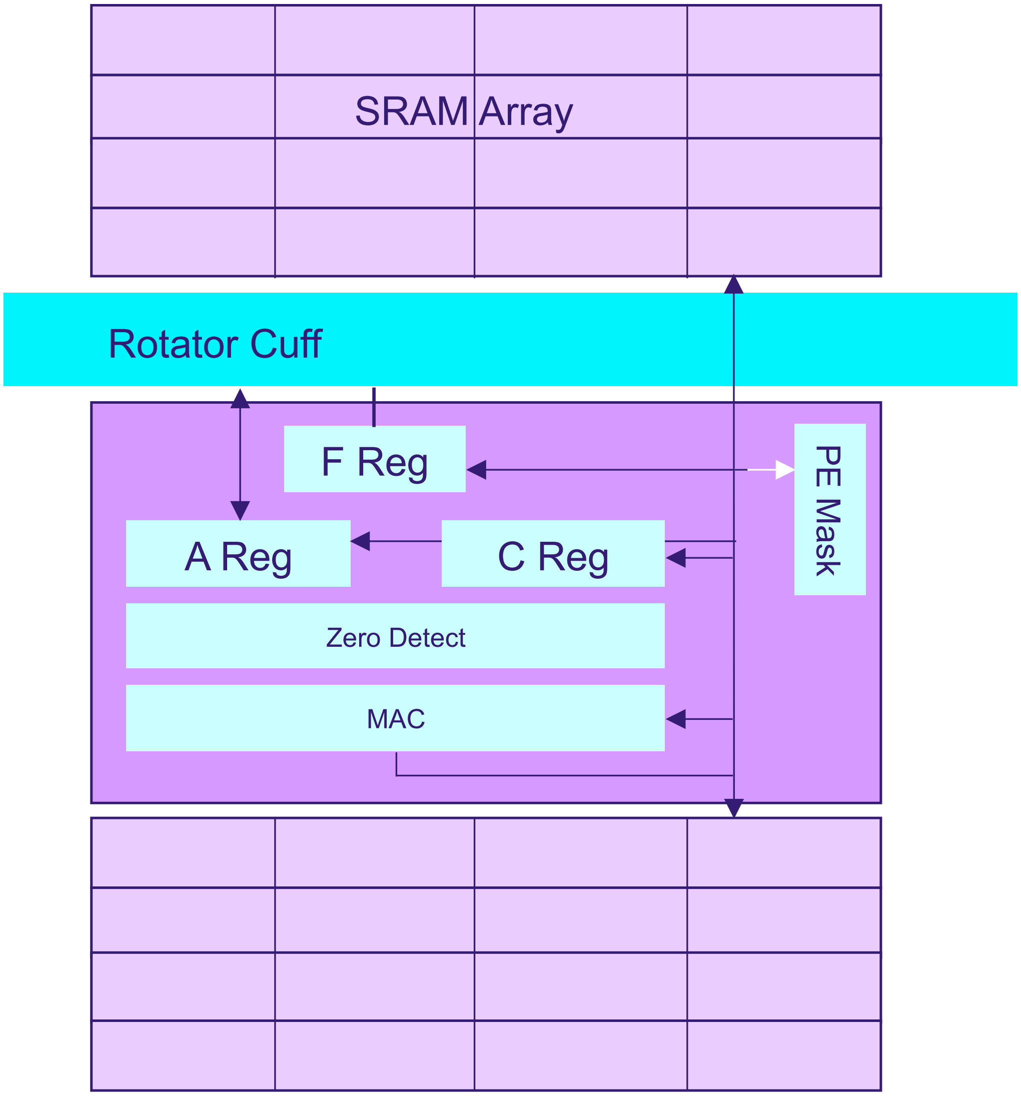
The dominant workload in neural net applications is the dot product. For convolutional neural networks, this is typically a matrix-matrix dot product General Matrix-Matrix Multiply, or GEMM. Many AI accelerators focus on performing this calculation as efficiently as possible. However, in many networks, like natural language processing (NLP) networks such as attention, transformer, and BERT, matrix-vector dot products (GEMV) are prevalent. Our at-memory computation works at the GEMV level, giving us the flexibility to most efficiently run both GEMV and GEMM operations.

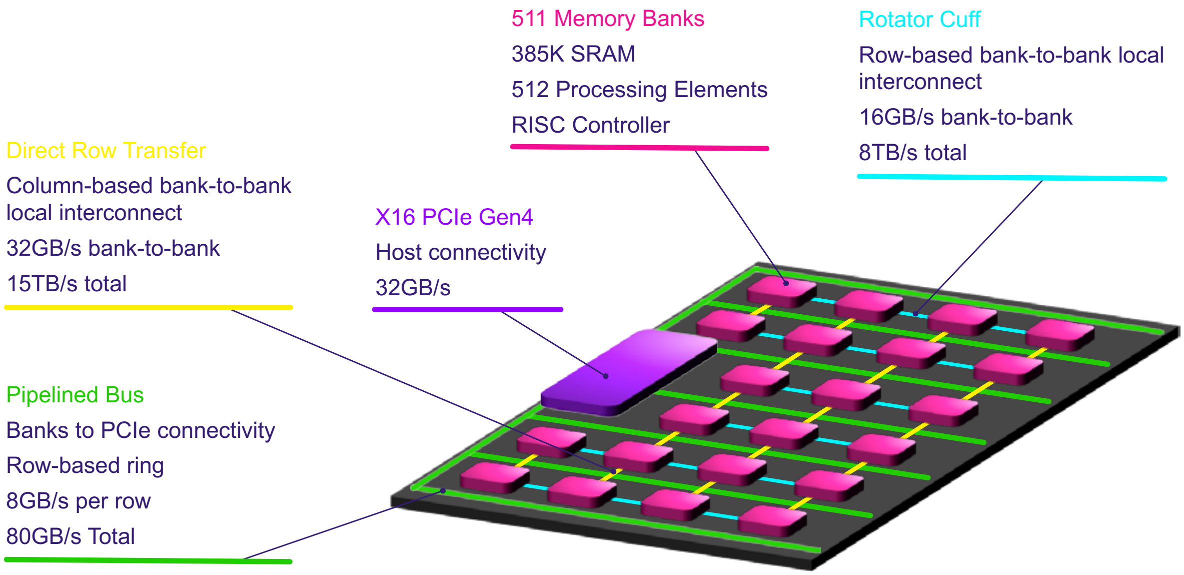
Our GEMV architecture gives us the flexibility to build up to GEMM by doing multiple GEMV operations, either done in series in one physical location or in parallel by taking advantage of multiple rows or banks of rows. This flexibility allows us to trade between computation, SRAM and area on a per-layer basis. We further improve efficiency by disabling the ALU at the individual PE level with zero detection. If either an activation or a coefficient is zero, the PE only pays for the data fetch, but not the cost of the MAC operation.

Our PE is designed to be very efficient at the GEMV operation but still capable of doing a wide variety of other like discrete Fourier transforms (DFTs) and other common signal processing operations. Our low-level instruction set allows very fine grained control over the behavior of the PE, and a small set of primitive building blocks allow complex operations to be executed without the complexity, area and energy cost of moving the data to a dedicated block or conventional processor core.

Our GEMV engine is supported by our patented Rotator Cuff, which allows the GEMV engine to run at 100% utilization for the full duration of any matrix-vector dot product operation. It achieves that by moving activations in one of a variety of circular patterns, achieved by using a “squashed loop”, hence the name “Rotator Cuff”.

Larger operations can be performed by extending the row of PEs up to a total length of 512 by configuring the rotator in “snake mode”, which connects alternating ends of the rows together to treat the entire set of PEs like a long squashed and folded loop.


SRAM Array
Rotator Cuff
F Reg
PE Mask
A Reg
C Reg
Zero Detect
MAC

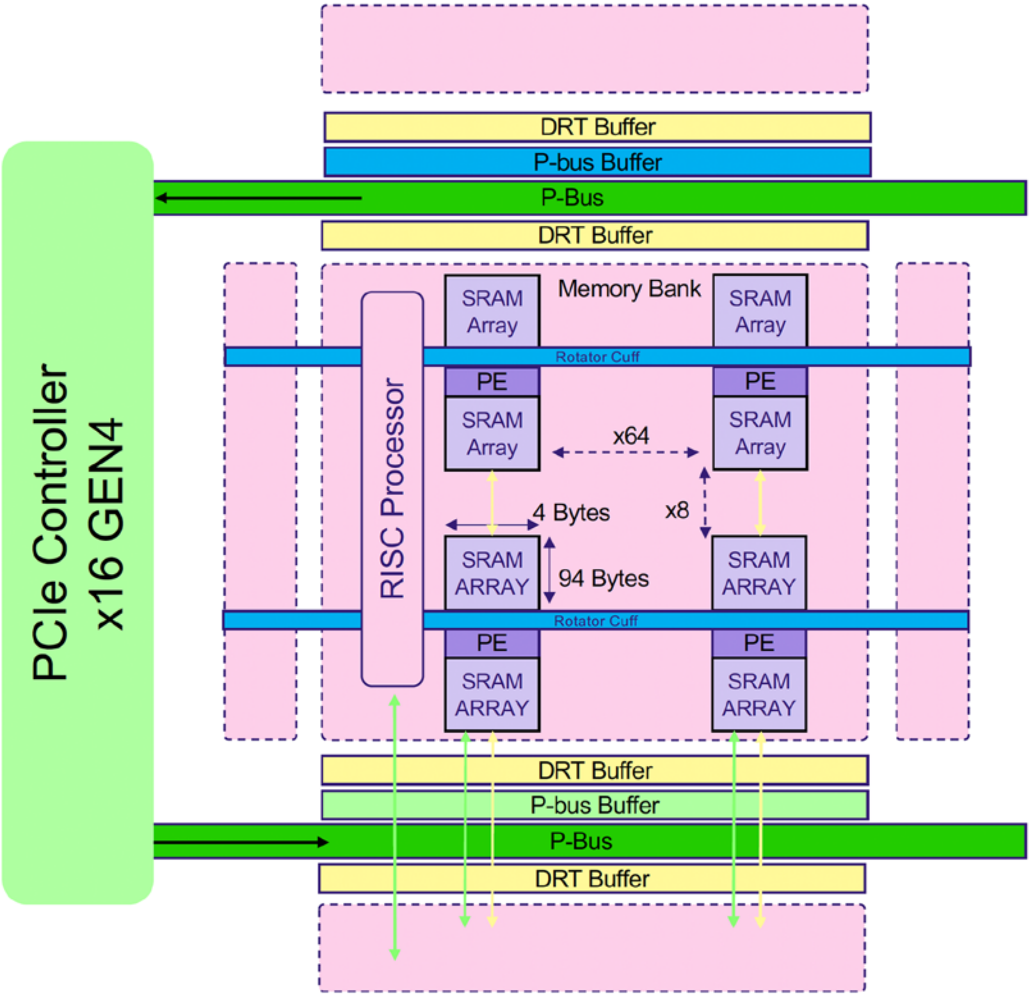


Moving Data Around

Our Rotator Cuff allows flexibility in data movement. In addition to being used during the GEMV operation, it can be used to rearrange data, feed data to our per-row ALUs in the RISC controller, or to move data between adjacent banks to the East and West. PEs can individually be configured at runtime to move data either left or right, meaning that both circular patterns and linear patterns that move all data in the same direction are available. Rotation between banks is achieved by coordinating with the neighboring bank and moving from the source Rotator Cuff, through a bank-to-bank FIFO, into the destination Rotator Cuff. If all rows are moving data, this mechanism can stream data from bank to bank at 16 GB/s.

To move data in the column direction, a mechanism called Direct Row Transfer (DRT) is used. Each row of 64 PEs can move 64B per cycle up or down by one row, giving a maximum rate of 32 GB/s per bank, or 15 TB/s for the whole chip. Bank to bank communication is enabled through a pair of depth-4 FIFOs at the top and bottom of the row, and row masking can be used by the controller to limit movement to just a subset of rows.

Data can be moved between the system host and the chip through direct memory access (DMA), which is available to every bank on the chip via our PBUS. The PBUS is a very energy efficient mechanism for moving data long distances within the chip. It does that by moving the data only a single step for each clock cycle, minimizing the driver size and overall energy needed to transmit the data between the banks and the PCIe Gen4 x16 interface at the edge of the chip. The PBUS is efficient while allowing us to support transfers at the full theoretical bandwidth of PCIe.



4. Taking Advantage of At-Memory Compute

The at-memory architecture does come with trade-offs. Perhaps the most fundamental one is that the amount of SRAM available to store a neural net is fixed and some well-known neural nets are large, with some, like GPT-3 being extraordinarily large. Regardless of the size of the neural net, however, it can be run using our at-memory accelerator. There are three fundamental strategies for doing this: partitioning the graph onto multiple chips, swapping coefficients on-chip, and offloading certain parts of the net to the host CPU. These strategies can also be combined by being applied at any level of the graph from the complete subgraph level down to aspects of specific layers of the graph.

The first strategy is to partition a net and run it on multiple chips. This is the default strategy and works great if the net can be made to fit onto the available number of chips. Each tsunAImi ships with 4 chips totaling over 800MB of SRAM space, so for most networks this solution will be the answer. For customers that have multiple boards, this strategy can be extended to encompass all available boards, allowing full at-memory acceleration of multi-GB sized networks.

If the net is extremely large, the strategy shifts to batching, with larger batch sizes amortizing the energy and time cost of swapping the contents of the chip at the expense of latency. Because we rely on the host computer for DRAM access, the range of batch sizes and maximum network size is effectively unlimited.

A key enabler of this last strategy is our built-in support for atomic PCIe operations, which allow the host and accelerator to coordinate with strong guarantees, a feature we've found surprisingly lacking in competing accelerators.

By applying these strategies, we can leverage the efficiency and speed of at-memory computation for a much wider range of applications than would be possible by limiting the architecture to only working with on-chip SRAM only, while still getting maximum benefit for the majority of workloads where the network can fit.

5. Conclusion

Untether AI's at-memory compute architecture is optimized for large-scale inference workloads and delivers the ultra-low latency that a typical near-memory or von Neumann architecture can't. By using integer-only arithmetic units, we can increase the throughput while reducing the cost. Flexibility is maintained to provide broad support for a wide variety of neural networks for AI inference applications that employ NLP, vision-oriented neural networks, and recommender systems in diverse industry segments, including industrial vision, finance, smart retail, and autonomous vehicles, among others.

Our AI Compute Engine is expressed in two hardware offerings. For inference acceleration, Untether AI's runAI200 devices operate using integer data types and a batch mode of 1, employing our unique at-memory architecture to deliver 502 TOPs and efficiency as high as 8 TOPs per watt. These devices power our tsunAImi accelerator card, which provides 2 peta operations per second of compute power per single card.

But hardware alone is not enough to successfully deploy AI workloads. Untether AI's hardware offerings are complemented by our imAIgine software development kit that's compatible with familiar machine-learning frameworks, including TensorFlow and PyTorch with Jupyter Notebook integration. It consists of a compiler for automated, optimized graph lowering; a toolkit, which supports extensive allocation and simulation feedback; and easily integrated communication and health-monitoring software in the form of a runtime.

Untether AI's at-memory compute-based hardware coupled with its software development kit provides high performance low power AI inference across a wide range of networks, making it flexible for today's neural-network architectures while anticipating the diverse unpredictability of AI workloads in the future.

[1] Source: https://arxiv.org/pdf/2007.05558.pdf

PLEASE READ THE ENTIRETY OF THIS “DISCLAIMER” SECTION CAREFULLY.

THE INFORMATION IN THIS WHITE PAPER IS PROVIDED “AS IS” WITHOUT ANY REPRESENTATIONS, WARRANTIES OR CONDITIONS OF ANY KIND OR NATURE, EXPRESS OR IMPLIED. UNTETHER AI CORPORATION (“UNTETHER AI”) MAKES NO WARRANTIES, EXPRESS OR IMPLIED, GUARANTEES OR CONDITIONS WITH RESPECT TO OR REGARDING THE THIS WHITE PAPER OR ANY PRODUCT REFERENCED HEREIN, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. UNDER NO CIRCUMSTANCES WILL UNTETHER AI, EVEN IF INFORMED ABOUT THE POSSIBILITY OF THE FOLLOWING, BE LIABLE FOR: SPECIAL, INCIDENTAL, EXEMPLARY, INDIRECT, OR ECONOMIC CONSEQUENTIAL DAMAGES; LOST PROFITS, BUSINESS VALUE, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS; OR LOSS OF OR DAMAGE TO DATA. THE DISCLAIMERS AND EXCLUSIONS IN THIS WHITE PAPER ALSO APPLY TO UNTETHER AI’S AFFILIATES, LICENSORS, CONTRACTORS, AND SUPPLIERS. UNTETHER AI HAS NO RESPONSIBILITY FOR CLAIMS BASED ON THIS WHITE PAPER, ANY VIOLATION OF LAW, INTELLECTUAL PROPERTY RIGHTS OR THIRD-PARTY RIGHTS RELATED TO THIS WHITE PAPER. UNTETHER AI MAKES NO REPRESENTATION ON THE AVAILABILITY OF ANY OF UNTETHER AI’S PRODUCTS AND ANY ASSOCIATED SERVICES. NOTHING HEREIN CONSTITUTES LEGAL, SECURITIES, FINANCIAL, BUSINESS OR TAX ADVICE AND YOU SHOULD CONSULT YOUR OWN LEGAL, FINANCIAL, SECURITIES, TAX OR OTHER PROFESSIONAL ADVISOR(S) BEFORE ENGAGING IN ANY ACTIVITY IN CONNECTION HEREWITH.

This White Paper is intended for general informational purposes only. The information herein may not be exhaustive and does not imply any element of a contractual relationship. There is no assurance as to the accuracy or completeness of such information and no representation, warranty or undertaking is, or is purported to be, provided as to the accuracy or completeness of such information. You acknowledge that circumstances may change and that the White Paper may become outdated as a result; and Untether is not under any obligation to update or correct this White Paper in connection therewith. The performance data and examples cited in the White Paper are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions. Where the White Paper includes information that has been obtained from third party sources, Untether AI has not independently verified, and does not guarantee, the accuracy or completeness of such information.

No regulatory authority has examined or approved, whether formally or informally, of any of the information set out in the White Paper. No such action or assurance has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of the White Paper does not imply that the applicable laws, regulatory requirements or rules have been complied with.

Untether AI’s products are warranted according to the terms and conditions of sale for such products. Notwithstanding any damages that you might incur for any reason whatsoever, Untether AI’s aggregate and cumulative liability towards you for the products described herein shall be limited in accordance with such terms and conditions of sale for such product.

Untether AI reserves the right to make corrections, modifications, enhancements, improvements, and other changes to this White Paper, at any time, without any notice. Reproduction of information in this White Paper is permissible only if reproduction is approved by Untether AI in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

By attending any presentation on this White Paper or by accepting any hard or soft copy of the White Paper, you agree to be bound by the foregoing limitations.

Copyright

Design by Thinkrs Creative Studio