Architecture Apocalypse
Dream Architecture for Deep Learning Inference and Compute -VERSAL AI Core

Alok G (*Author*)
Product and Platforms Marketing, Xilinx
Xilinx Inc, San Jose, California
Alok.Gupta@Xilinx.com

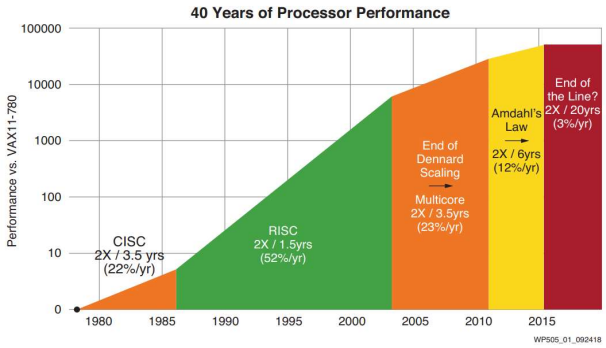
***Abstract*—With Deep Learning algorithmic advances outpacing hardware advances, How do you ensure that algorithms of tomorrow are a good fit for existing AI chips under development? , Most of these AI chips are being designed for the AI algorithms of today, Given the rate and the magnitude of algorithm evolution, many of these AI chip designs may become obsolete even before their commercial releases. Algorithms of tomorrow demands overhaul of architecture, memory/data resources and capabilities. Dream architecture for Inference has to redefine some fundamental chip techniques that rewrite the rules in computing and delivers breakthrough AI acceleration and flexible compute capability beyond that of server-class CPUs and versatile than GPUs/ASICs to support breadth of applications and dynamic workloads. This Paper will discuss how these industry challenges can be addressed at various levels of hardware and software design using Xilinx VERSAL AI Core, the industry's first ACAP (Adaptable compute acceleration platform) device which leapfrogs the performance of CPU/GPU's and FPGA's.**

Keywords—DeepLearning,Inference,AI,Accelerated Computing

I. INTRODUCTION

Modern technical challenges have forced the industry to explore options beyond the stereotypical "one size fits all" CPU scalar processing solution. Very large vector processing (DSP, GPU) solves some problems, but it runs into customary scaling challenges due to inflexible, inefficient memory bandwidth usage. Traditional FPGA solutions provide programmable memory hierarchy, but the traditional hardware development flow has been a barrier to broad, high-volume adoption in application spaces like the Data Center and Automotive markets. Industry needs a solution that can combine all three elements with a single software stack that offers a variety of different abstractions—from framework to C to RTL-level coding.

Deep learning algorithms are coinciding with a breakdown in Moore's Law, the decades-old rule of thumb of progress in computer chips, forcing radical new computer designs. It's not as if someone flipped a switch, and Moore's Law suddenly vanished. Its validity has been in decline for a while now, but evidence of that is just now coming to the fore. That's because of several activities that prolonged the performance curve. It is not just Moore's Law that is coming to an end with respect to processor performance but also Dennard Scaling and Amdahl's Law. Processor performance over the last 40 years and the decline of these laws is displayed in the graph below



Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e. 2018

Computing technology is moving into a direction in which functionality has priority over physical specifications like clock speed or device feature sizes. When Dennard's scaling rules became unsustainable around 2004, there was a sudden shift away from the GHz war between manufacturers. A similar phenomenon has occurred recently with advertised on-chip feature sizes, witness Semi-conductor giant (Intel's) statement that there will be more focus on performance improvements rather than the underlying technology node. Such improvements will not necessarily be quantitative, e.g., more Floating-Point Operations per Second (FLOPS), but rather will emphasize energy efficiency (FLOPS per Watt), or quality in terms of how well applications will actually be served. Until around 2004 Moore's Law ran in conjunction with Dennard's scaling rules, which provided a predictable scaling of design parameters. Dennard's scaling rules had provided a free lunch for engineers for decades, as the decreased transistor sizes delivered by Moore's Law automatically led to better performance in terms of speed and power consumption. An early sign that Moore's law was in its final stages came with the end of Dennard's

scaling rules, when it became more difficult to decrease voltage at the same high pace due to increased leakage currents in devices. Thus came to an end the race between manufacturers for increasing clock speeds, only to be replaced by a race to manufacture chips by the most advanced technology node.

Well, Enter Artificial Intelligence, and how it fixes the situation. Computer pioneer, Alan Turing in 1950 proposed:

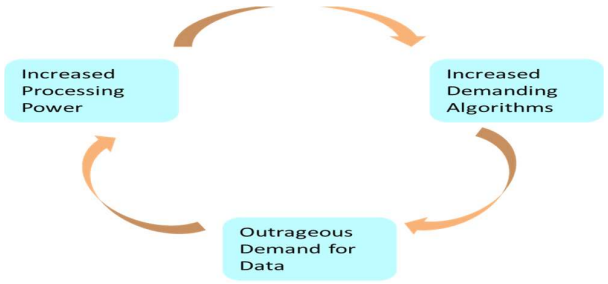
"*Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education, one would obtain the adult brain.".*

This idea grew on to become Deep Learning. Fast forward to 2018: we have, and are still gathering, massive amounts of data. We have and are still developing more and more advanced algorithms (Generative Adversarial Networks and Capsule Networks stand as strong examples.) But do we have the hardware to crunch all those calculations within a reasonable time? And if we do, can it be done without having all those GPUs? , As a result, the semiconductor industry is exploring alternate domain-specific architectures, such as vector-based processing (DSPs, GPUs) and fully parallel programmable hardware (FPGAs). The question becomes, Which architecture is best for which task?

II. FUTURE OF COMPUTING IMPACTS AI REVOLUTION

A. *The reinforcing cycle*

Data is exploding at a massive pace across the globe and needs sophisticated algorithms, software, and powerful compute that can handle, manage, and provide real-time artificial intelligence (AI) inferencing. The tremendous growth in data and the need for better insights, has led to the rapid adoption of AI. A common approach to implement AI algorithms for enterprises is through using machine learning and it's subset, deep learning. They both use huge quantities of data to train AI models and then deploy those models across various use cases, including image classification and recognition, object detection, and recommender systems among others. Right there, Artificial Intelligence is imposing a constraint: keep the power constant or decrease it, but increase performance… doesn't that sound a bit familiar to some scaling rule we have just seen? Exactly, by forcing the tech industry to come up with new processors which can perform more calculations per unit time, while maintaining power consumption and price, Artificial Intelligence is imposing Dennard Scaling again, and hence forcing Moore's Law back to life!

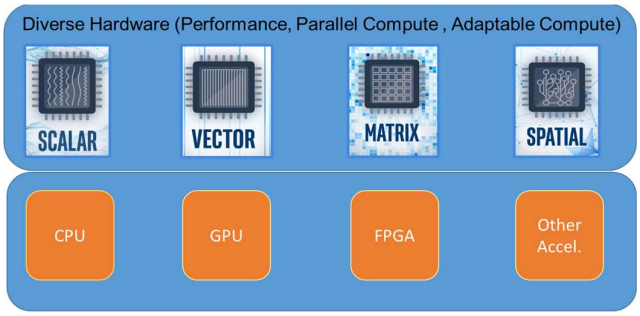


B. *Which Architecture is best for which task*

• Scalar processing elements (e.g., CPUs) are very efficient at complex algorithms with diverse decision trees and a broad set of libraries but are limited in performance scaling.

• Vector processing elements (e.g., DSPs, GPUs) are more efficient at a narrower set of parallelizable compute functions but they experience latency and efficiency penalties because of inflexible memory hierarchy.

• Programmable logic (e.g., FPGAs) can be precisely customized to a particular compute function, which makes them best at latency-critical real-time applications (e.g., automotive driver assist) and irregular data structures (e.g., genomic sequencing)—but algorithmic changes have traditionally taken hours to compile versus minutes.



Algorithms of tomorrow demands architecture overhaul of the memory/data resources and capabilities. Industry demands a new heterogeneous compute architecture, which can deliver the best of all worlds. A World-class vector and scalar processing elements which can be tightly coupled to advanced programmable logic (PL), with a very high-bandwidth network on chip. With possibility of memory-map access to all processing element types. This tightly coupled hybrid architecture should allow more dramatic customization and performance increase than any one implementation alone. Such a dramatic increase in performance necessitates a similarly dramatic improvement in tools focusing on ease of use. Which demands a fully integrated, memory-mapped platform for programming through a unified toolchain.

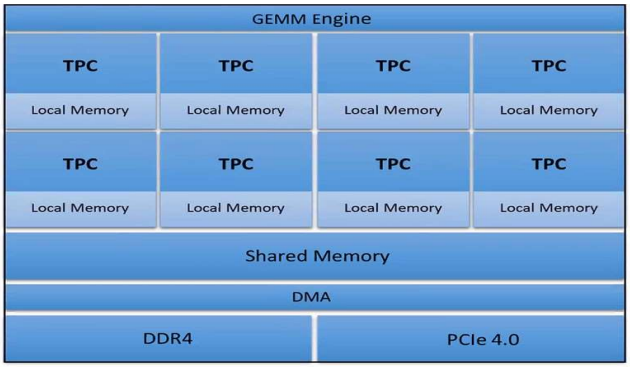
C. *Solving for the bottleneck*

The reason why devices heat up, and the main problem with our current computer hardware designs, is the so called "von Neumann bottleneck", classic computer architectures separate the processing from the data storage, which means that data need to be transferred back and forth from one place to the other overtime a calculation takes place. Parallelism solves part of this problem by breaking down calculations and distributing processing, but you still need to move data at the end, to reconcile everything into a desired output. So , what if there was a way to get rid of the hardware bottleneck altogether? What if processing and data resided in the same place and nothing had to be moved around and produce heat, and consume so much energy? After all, that is how our brains works we do not have separate areas for processing and data storage as computers do everything is happening at our neurons.

III. ARCHITECTURE OF EXISTING ASIC'S

Let's take a look at the High-level architectures of some of the Inference accelerators currently being developed in the Industry and see why they are capable of tackling AI inference.

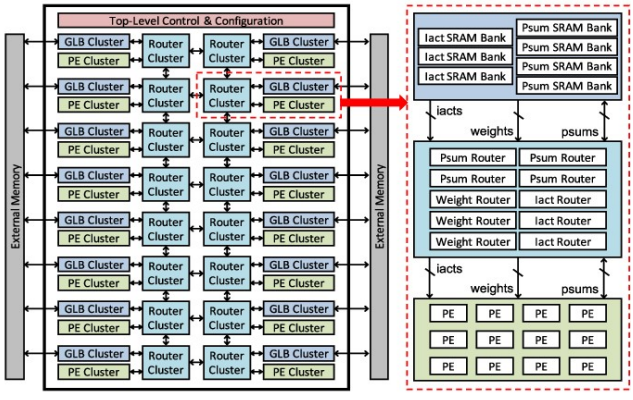
Habana GOYA™ Deep learning Inference Platform



Source: Habana Goya™ Inference Platform Aug 2019

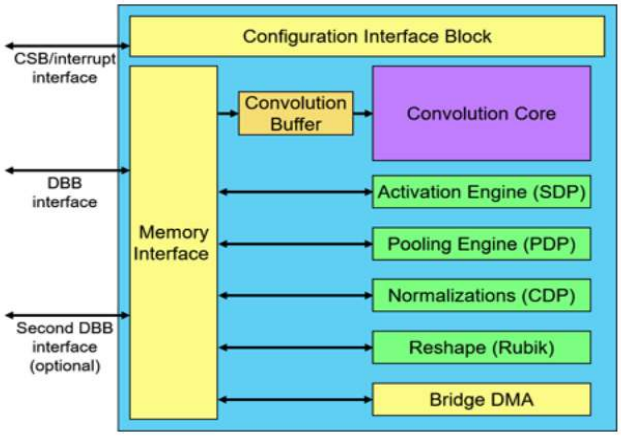
The two main components of the chip are the TPC (Tensor Processor Core) and the GEMM (general matrix multiply) engine. The TPCs are the company ground up VLIW SIMD CPU/DSP design. Those cores are based on a custom VLIW ISA which features specialized AI SIMD vector instructions. An interesting aspect of the chip is that the TPCs do not have local caches. Instead, they have a local chunk of scratchpad memory along with a large shared memory which is shared by both the GEMM engine and the TPCs.

Eyeriss — The Eyeriss team from MIT has been working on deep learning inference accelerators. Eyeriss is deep convolutional neural network (CNN) accelerator chip featuring a spatial array of multiple processing elements (PE) fed by a reconfigurable multicast on-chip network that handles many shapes and minimizes data movement by exploiting data reuse. Eyeriss uses a hierarchical mesh network (HM-NoC) & takes advantage of the all-to-all network. The all-to-all network is limited within the scope of a cluster at the lower level.



Source: Eyerissv2 - Published in IEEE Journal on Emerging Circuits and Systems 2018

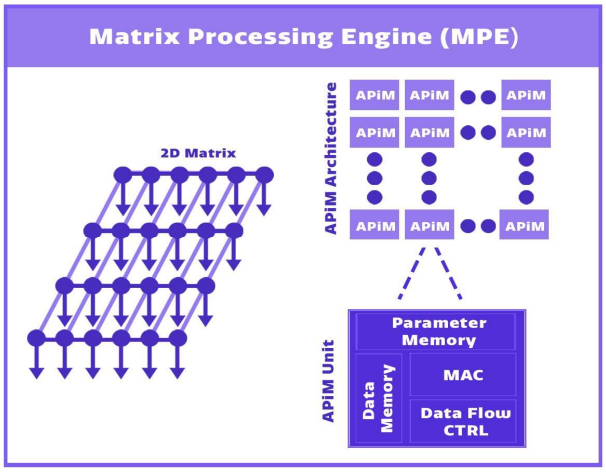
Nvidia Deep Learning Accelerator (NVDLA)



Source: NVDLA Open Source Project

The NVDLA instruction set differs from other DLAs by supporting four convolution modes: direct, Winograd, image direct, and batch. The direct mode is the most basic convolution operation, enabling parallelization up to the MAC-array width. Winograd transform to the input data boosts convolutional-neural-network (CNN) performance and power efficiency by reducing the number of MAC operations. To reduce bandwidth for fully connected layers, the NVDLA has a batching feature that allows multiple sets of activations to run at the same time. By allowing multiple activation sets to share the same weight data, So performance and memory bandwidth are affected. Though NVDLA will accelerate many of today's DL networks but what about tomorrows?

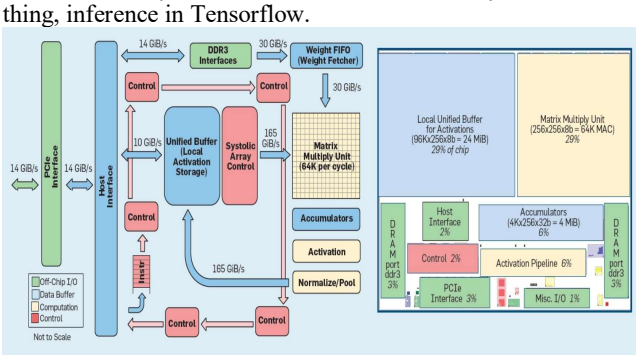
Gyrfalcon argues that by surrounding each identical computing unit with memory, an approach it calls "AI Processing in Memory," or APiM, use of external memory can be greatly reduced, thereby lowering the power budget of AI chips drastically.



Source: Gyrfalcon AI accelerators Lightspeeur® 2803S

Google TPU comprised of systolic data flow engine. Modern CPUs are strengthened by a massive cache, branch prediction and high clock rate on each of its cores. Which all contribute to a lower latency of the CPU.A GPU does the same thing but has thousands of ALU's to perform its calculations. A calculation can be parallelized over all ALU's. This is called a SIMD, A GPU does however not use the fancy features which lower the latency. In short, a GPU drastically increases its throughput by parallelizing its computation in exchange for an increase in its latency.

A TPU, on the other hand, operates very differently. Its ALU's are directly connected to each other without using the memory. They can directly give pass information which will drastically decrease latency. TPU has almost no flexibility. It does one thing, inference in Tensorflow.



Source: Google TPU Custom Chip 2016

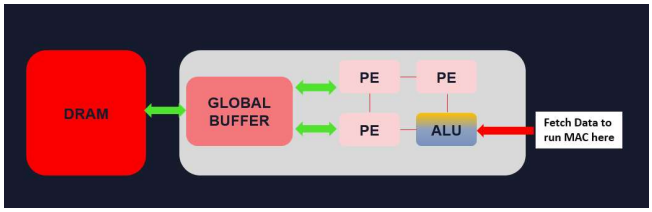
Now building an ASIC isn't for the faint of heart. First, an ASIC has limited functionality. For example, the Google TPU only supports TensorFlow, leaving the users of the other major AI frameworks, championed by Microsoft, Amazon, Facebook, *etc*., to run on GPUs. ASICs are also expensive to develop a complex design can cost in excess of $100M and take years to design and debug. Inference needs are specific and often extraordinarily specialized. After studying some of the above architectures its evident that most of these chips are massively parallel that keeps data close to the processing points and maximizes data reuse as much as possible deep learning inference accelerator.

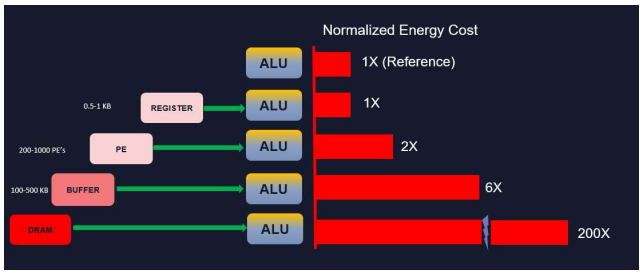
IV. Design Aspects Of DNN Processor

A. *Key Components of DNN Processor*

Matrix multiplication Unit referred by different names like TPC (Tensor processing core), PE (Processing unit),etc. GEMM is the core computation involved in DNN's.

SRAM is the local memory used to store the weights or intermediate outputs/activations.

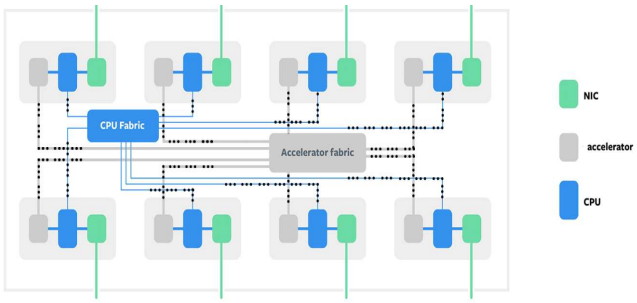




To reduce energy consumption the memory should be located as close as possible to the processing unit and should be accessed as little as possible.

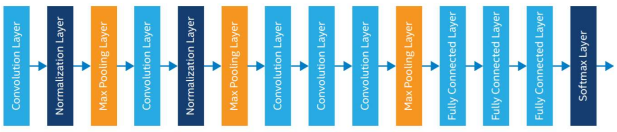
Interconnect/Fabric This is the logic which connects all the different processing units and memory so that output from one layer or block can be transferred to the next block. Also referred to as Network on Chip (NoC).
Interfaces (DDR, PCIE) These blocks are needed to connect to external memory (DRAM) and an external processor.
Controller — This can be a RISC-V or ARM processor or custom logic which is used to control and communicate with all the other blocks and the external processor.

B. *Architecture and Instruction Set*



Source: Facebook AI Inference with Kings Canyon

If we look at all the architectures, we will see memory is always placed as close as possible to the compute. The reason is that moving data consumes more energy than compute. Let's look at the computation and memory involved in AlexNet architecture.



AlexNet consists of 5 Constitutional layers and 3 fully connected layers. The total number of parameters/weights for AlexNet is around 62 million. Let's say after weight quantization each weight is stored as an 8-bit value so if we want to keep all the weights in on-chip memory it would require at least 62 MB of SRAM or 62*8 Mega-bits = 496 Million SRAM cells. So while deciding HW architecture we have to keep in mind which DNN architectures we can support without keeping weights off-chip (which increases power consumption).

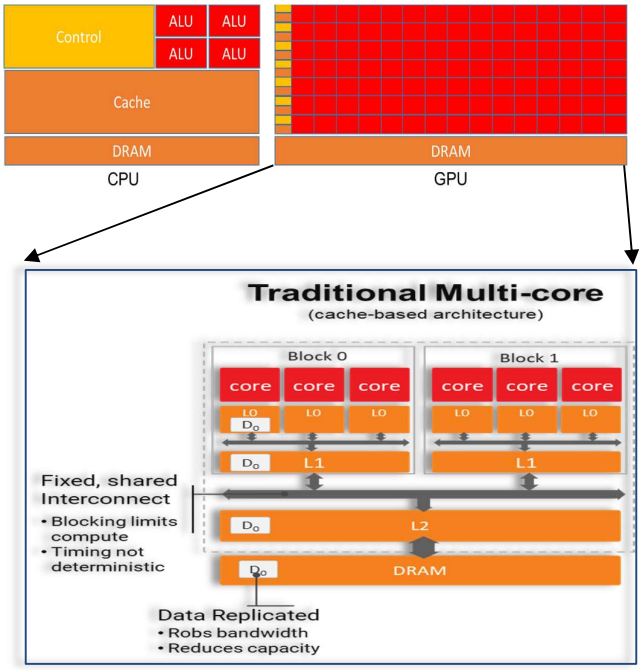
Typical Instructions Set on Accelerators are domain-specific Instruction Set Architecture (ISA) for NN, which is a load-store

architecture that integrates scalar, vector, matrix, logical, data transfer, and control instructions. instruction set is simple: matrix multiply, linear algebra, convolutions.

V. XILINX REINVENTS MULTI-CORE COMPUTE

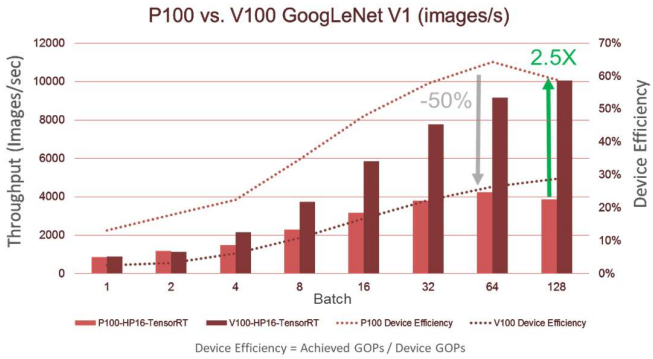
Dream architecture for Inference has to redefine some fundamental chip techniques that rewrite the rules in computing and delivers breakthrough AI acceleration and flexible compute capabilities. Getting 2X Performance from one process node to another is easier than getting it in the architecture, that's where Xilinx took a leap forward to answer the question for dream architecture, Xilinx has spawned a revolutionary compute architecture designed to strike a balance between throughput, latency, and power consumption by introducing AI Engine in some of their Versal ACAP devices. The engine is capable of massive data crunching with low latency and high accuracy, as required by these workloads.

Traditional Multi-Core Architectures



GPUs are embedded with hundreds to thousands of arithmetic processing units on one die and have tremendous computing power. However, their realistic performance is often limited by the huge performance gap between the processors and the GPU memory system. For example, NVIDIA's GTX980 has a raw computational power of 4,612 GFlop/s, but its theoretical memory bandwidth is only 224 GB/s. The realistic memory throughput is even lower. The memory bottleneck remains a significant challenge for these parallel computing chips. The GPU memory hierarchy is rather complex, and includes the GPU-unique shared, texture and constant memory. The theoretical bandwidths of both global memory and shared memory are difficult to saturate, and hardware resources are imbalanced with a low utilization rate.

A. Memory Hierarchy a Potential Bottleneck for GPGPUs

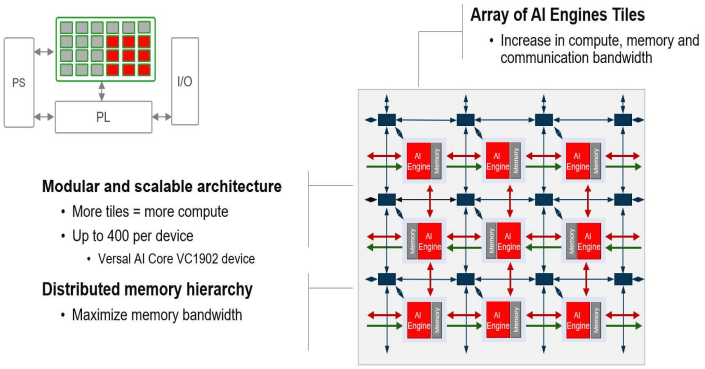


Nvidia V100 Claimed 6X Compute Performance (Tensor Cores) with respect to P100 and 25% More Memory B/W (more HBM perf) versus Pascal GP100. But the actual DNN benchmarks shows Only 2.5X V100 performance increase and ~50% Less device efficiency

Memory hierarchy limitations - Data flow in a GPU is defined by software and is governed by the GPU's rigid and complex memory hierarchy, if the compute and efficiency potential of the GPU is to be realized, a workload's data flow must map precisely to the GPU memory hierarchy. In reality, very few workloads have sufficient data locality patterns to enable efficient mapping to GPUs. For such workloads, the realizable compute and efficiency are substantially reduced, and the latency of the solution is increased when implemented on a GPU.

Xilinx – Versal AI Engine Array.

AI Engine is a new building block for Xilinx Versal 7nm devices. Xilinx AI Engines are an array of innovative very long instruction word (VLIW) and single instruction, multiple data (SIMD) processing engines and memories, all interconnected with 100s of terabits per second of interconnect and memory bandwidth. The AI Engine array Interface provide the required functionality to integrate with the rest of the Versal device through the PL and the Network-on Chip (NoC). AI Engine does not have cache memories in order to achieve predictable performance. So, there is no coherency issue in the tile-to-tile communication.

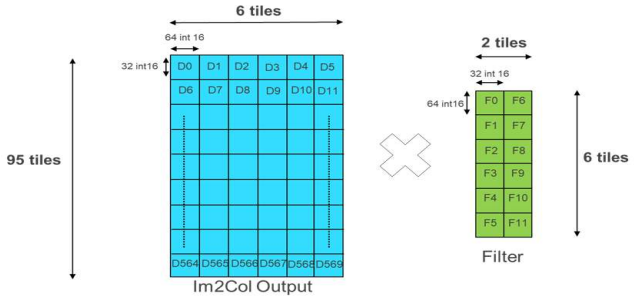


Unique combinations of PL & AI Engine gives best of both worlds: Highly efficient, future proof, compute platform across a wide range of end applications and AI Engine data path flexibility.

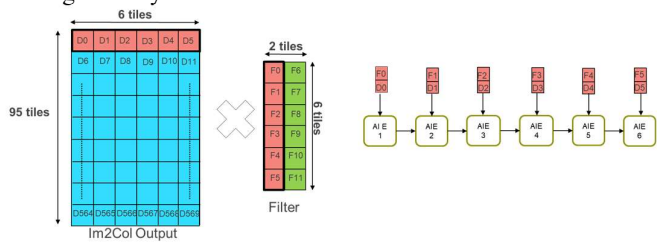
Let's take a look at the AlexNet v2 1st layer mapping on AI Engine in Versal device (GEMM based), below tale shows the Layer 1 Input, Filter, Output matrices sizes required and compute ops.

Layer	Input Image	Input Depth	Filter	Output Image	Output Depth	Matrix D	Matrix F	Compute (ops)
1	227 x 227	3	11 x 11	55 x 55	64	3025 x 363	363 x 64	70.3 M

The Alexnet v2, 1st layer can be architected as shown in figure below and process the matrices 32x64 and 64x32 int16 tiles at a time per AI Engine core.



The below figure shows the 1st layer mapping on multi-core inner product approach and how multiplications spread in six AI Engine arrays.



Trace analysis for cascade of 6 AI Engines arrays, running in parallel.

Partitioning the GEMM problem in space and time helps distribute sub-matrix multiplications across multiple cores in AI Engine array and complete the big matrix multiply operations in multiple iterations using the Versal compute resources.

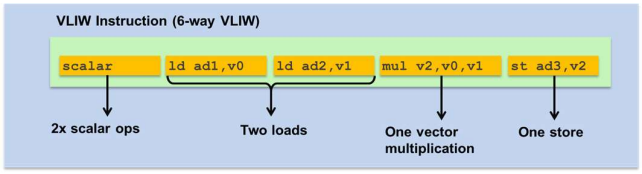
The key objectives for the AI Engine Architecture are:

- Provide a highly-optimized DSP signal processing architecture optimized for functions in the following markets: like Machine Learning (ML) for Data Center acceleration and Automotive Driver Assistance (ADAS) , Embedded Vision , Wireless Radio/DFE, 5G/Baseband, and opportunistically wireless Backhaul.
- Provide efficient performance improvements, power reduction and software programmability. Xilinx AI Engine architecture targets customer ASIC replacements
- Provide a higher-level programming experience than a traditional RTL design flow with Languages such as C, C++ , Provide Predictable and guaranteed throughput and latency (e.g., no timing closure). From Flexibility perspective time needed to reconfigure a particular block (region or particular IP) from one configuration to another in the order of minutes.

B. VLIW Processor Advantages

VLIW Processor Achieves multiple forms of Parallelism through instruction-level and data-level parallelism.

a) Instruction-level Parallelism (ILP) Multiple operations issued in one cycle. b) Data-level Parallelism (DLP) Vector datapath (SIMD approach).



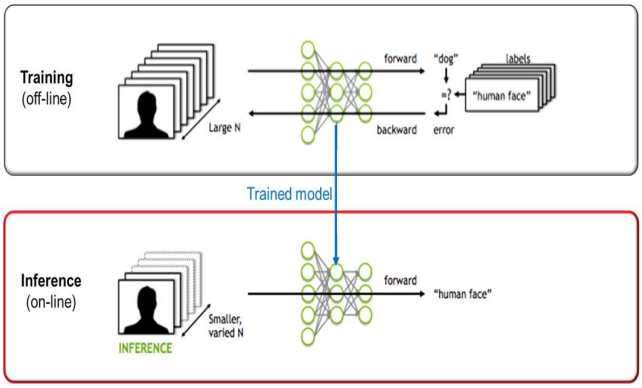
For each clock cycle the Xilinx AI Engine VLIW processor can perform two scalar instructions, two vector reads, a single vector write, and a single vector instruction executed 6-way VLIW. Other advantages are Simpler hardware (Less power hungry), More scalable (Allow more instructions per VLIW bundle).

C. Scope of Deep Learning Inference

School's in session. That's how to think about deep neural networks going through the "training" phase. Neural networks get an education for the same reason most people do to learn to do a job.

More specifically, the trained neural network is put to work out in the digital world using what it has learned to recognize images, spoken words, a blood disease, or suggest the shoes someone is likely to buy next, you name it in the streamlined form of an application.

This speedier and more efficient version of a neural network infers things about new data it's presented with, based on its training. In the AI lexicon this is known as "inference."



Inference can't happen without training. Makes sense. That's how we gain and use our own knowledge for the most part. And just as we don't haul around all our teachers, a few overloaded bookshelves and a red-brick schoolhouse to read a Shakespeare sonnet, inference doesn't require all the infrastructure of its training regimen to do its job well.

Xilinx Versal devices are inherently designed for low latency, high throughput, power efficiency, and flexibility. ACAP's make real-time inference possible by providing completely customizable hardware acceleration while retaining the flexibility to evolve with rapidly changing machine learning (ML) and deep learning (DL) models and providing performance better than ASIC's. Versal AI Engine core architecture has many features which naturally fit with machine learning and deep learning applications like:

Highly parallel architecture: Facilitates efficient low-batch stream processing and reduces latency.

Configurable distributed floating-point blocks: Accelerates computation by tuning compute performance. Whatever you choose from lower precision integers to high precision floating point numerics, you can continue to adjust along the performance/power curve.

Tightly-couple high-bandwidth memory: Aggregate bandwidth of 1Tb/s+ bandwidth, random access, reduces latency, minimizes external memory access.

Programmable data path: Reduces unnecessary data movement, improving latency and efficiency.

Adaptable and future proof: ACAPs provide customizable adaptable hardware acceleration that can be programmed and tuned again and again to achieve maximum performance.

VI. Deep Learning Inference Challenges

Key challenges for AI Inference are:

- Rate of AI Innovation
- Performance at low latency
- Low power Consumption

As artificial intelligence starts to pervade modern life, the demand for enhanced compute efficiency has begun to drive innovation in the semiconductor space, but it is difficult for any single implementation to handle with maximum efficiency. This is one area where the tight coupling between vector processing and programmable hardware is invaluable. There are two trends in DNN research that are driving the adoption of FPGAs over GPUs: low precision data types and sparsity.

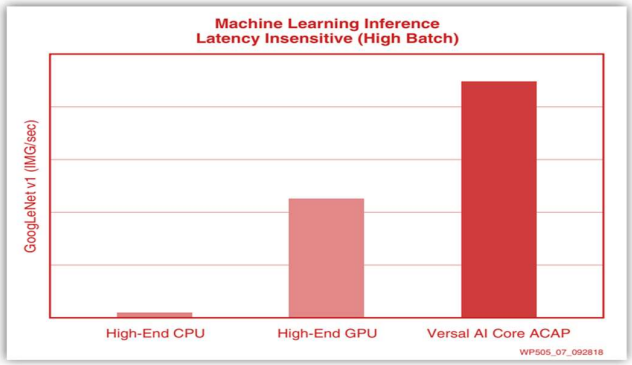
There has been a lot of attention on the precision of the compute unit (FP32 versus FP16 versus INT16 versus INT8, etc.), but inattention to the divergence in memory hierarchy requirements between network types has caused many of the most recent ML chips to drop sharply in efficiency for different networks. For example, the current state-of-the-art machine learning inference engines require four HBM memories (7.2Tb/s of external memory bandwidth) to reach their peak performance—but their cache-based memory hierarchy only operates at around 25–30% efficiency and creates significant latency uncertainty for real-time applications. The solution is to augment the vector processing performed by intelligent engines with a programmable memory hierarchy, precisely optimized for each network type and enabled by the massive parallelism of FPGA logic. Operands of 8-bit, 16-bit, 32-bit, and single-precision floating point (SPFP) are supported with different operands-per-clock cycle as shown in table below showcasing multi-precision support in Xilinx Versal device.

Operand 1	Operand 2	Output	Number of GMACs @ 1 GHz
8 real	8 real	16b real	128
16 real	8 real	48 real	64
16 real	16 complex	48 complex	16
16 complex	16 complex	48 complex	8
16 real	32 real	48/80 real	16
16 real	32 complex	48/80 complex	8

Versal ACAP Multi-Precision

Industry trends in accelerated computing suggests that the workloads in machine learning are rapidly changing whether its Database analytics, Video Transcode, Machine Learning. Key point to observe is because it can take so long to develop an ASIC like the TPU, a chip's design may miss the window of recent innovations in a fast-moving market like AI. This is why many datacenters, including Baidu, Amazon and Microsoft, prefer to accelerate key workloads with GPUs as well as FPGAs from Xilinx, which deliver high performance and power efficiency while retaining the flexibility to evolve the hardware on the fly as needed. And a talented design team can build a new FPGA in months, not years.

Versal platform implementation of GoogLeNet enables extraordinarily high performance for latency insensitive applications, 43X more throughput than today's top-of-the-line Skylake Platinum CPU, and about 3X today's top-of-the-line GPU all at much lower power. See Figure below



GoogLeNet Performance (< 7ms Latency) = 43X Higher Than a High-End CPU.

1. Measured on Xeon Platinum 8124 Skylake, c5.18xlarge AWS instance, Intel Caffe: 2. V100 numbers taken from Nvidia Technical Overview, "Deep Learning Platform, Giant Leaps in Performance and Efficiency for AI Services."

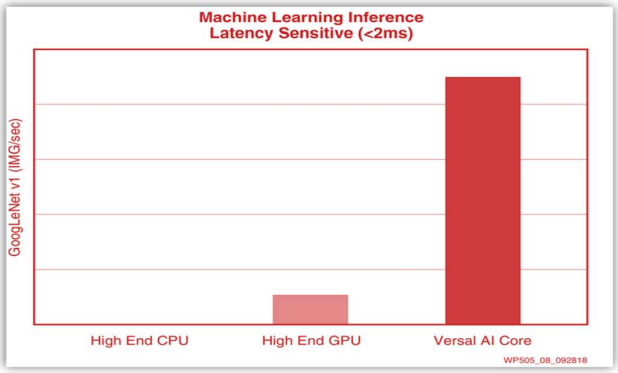
As the number of real-time applications continues to increase, it is important for Data Center customers to choose a technology that can scale to keep up with their future needs.

Two trends are emerging:

• Deterministic latency is becoming increasingly important to improve software design efficiency

• Neural network latency requirements continue to tighten as increasingly complex interactions are modeled (human interaction, financial trading), and safety-critical applications rise in importance (e.g., automotive, industrial).

These two requirements necessitate the removal of batching, which causes the performance of fixed, cache-based memory hierarchy of CPU and GPU-based solutions to degrade significantly. Even a high-end CPU caps out at 5ms latency, and below 7ms, even high-end GPUs degrade significantly in performance. Only the Versal ACAP achieves sub-2ms latency with acceptable performance. See Figure



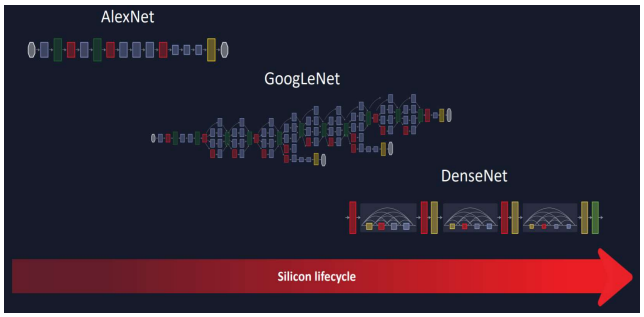
Real-Time GoogLeNet Performance (< 2ms Latency) = 8X Higher Than High-End GPU (Nvidia)

1. Measured on Xeon Platinum 8124 Skylake, c5.18xlarge AWS instance, Intel Caffe: 2. V100 numbers taken from Nvidia Technical Overview, "Deep Learning Platform, Giant Leaps in Performance and Efficiency for AI Services."

As a result, the unique programmable memory hierarchy of ACAP-based solutions offers both the highest performance for machine learning inference as well as unmatched scalability as future applications demand lower and more deterministic latency.

A. AI is outpacing Moore's Law

Stanford University finds that AI is outpacing Moore's Law. Every three months, the speed of artificial intelligence computation doubles, according to Stanford University's 2019 AI Index report.



Following the structures of DL models in above diagram, Trend shows these networks are very diverse, DL models evolving to more complex in nature and the diversity is massive, changing every day , need a AI hardware solution which can scale with these solutions as they evolve in future with increased network complexities. In consideration of the fact that the range of applications for which CNNs offer state-of-the-art performance and accuracy is growing, it is likely that the divergence of the memory demands across these applications will also continue. As this gap widens, designing hardware accelerators that offer both performance and efficiency across a variety of CNN benchmarks will become more difficult.

Hence the Fixed Silicon architectures are not the answer, Answer is Adaptable hardware where you can not only program your hardware minute by minute for broad range of different workloads, but which can also be updated with latest and greatest algorithms as industry trends and requirements change which provides a future proof system.

The new Versal ACAP architecture also yields a dramatic improvement in ease of use. It provides a fully integrated, memory-mapped platform for programming through a unified toolchain. The Xilinx toolchain supports multiple entry methods for every type of developer. For example, certain applications (such as AI machine learning inference) can be coded at the framework level (e.g., Caffe, TensorFlow); others can be coded in C using pre-optimized libraries (e.g., filters for 5G radio). Traditional hardware developers can still port their existing RTL to ACAP via the traditional RTL entry flow. And to program ACAP Versal devices Xilinx has introduced Vitis AI development environment which is a specialized development environment for accelerating AI inference on Xilinx embedded platforms, Accelerator cards, or on the FPGA-instances in the cloud. Vitis AI development environment supports the

industry's leading deep learning frameworks like Tensorflow and Caffe, and offers comprehensive APIs to prune, quantize, optimize, and compile your trained networks to achieve the highest AI inference performance for your deployed application.

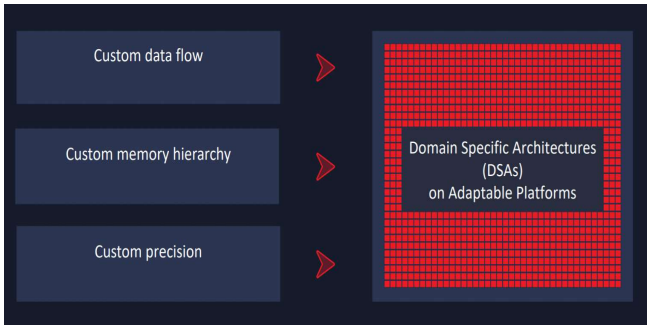
The four major advantages of Versal ACAP include:

1. **Software Programmability -** The ability to quickly develop optimized applications through software-abstracted toolchains.

2. **Acceleration -** Metrics for a wide range of applications from artificial intelligence, smart network interface cards, high density storage, 5G wireless, self-driving cars, advanced modular radar, and terabit optical networks.

3. **Dynamically Adaptable Reconfiguration -** The ability to reconfigure the hardware to accelerate new loads within milliseconds. When state of the art changes, platform can be re-implement all of this without a Silicon Wrap.

4. **Infinite memory depth/width granularities –** Combining blocks of URAMs, BRAMs that can be configured in a variety of data width/depth combinations to offer the most flexible way of building memories of different sizes at the lowest cost per bit of memory, A very unique advantage in Versal ACAP.



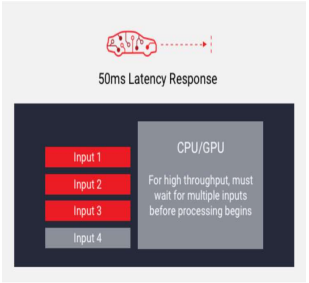
B. Low Latency is Critical for Inference

The basic principle of batch-size selection is to select a small batch-size for latency-sensitive services and a large batch-size for throughput-sensitive services.

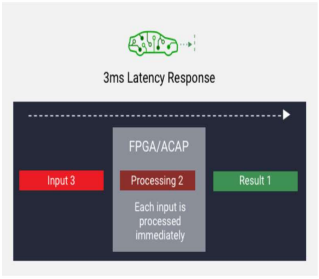
Performance with different batch-size on an Intel® Xeon® processor E5-2650v4

Figure above shows the effect of choosing a different batch-size on inference service throughput and latency. Test results show that when the batch-size is small, increasing the batch-size appropriately (for example, batch-size from 1 to 2) has less impact on the latency, but can quickly improve the performance of the throughput; when the batch-size is large, increasing its value (for example, from 8 to 32) does not improve throughput, but greatly affects the service latency performance. Therefore, in practice, it is necessary to optimize the selection of batch-size according to the number of CPU cores and service performance requirements of the deployed service node.

One of the ways to achieve the highest performance in GPU computing is to hide the long latency and other computational overheads by high data-level parallelism to achieve a high throughput, for example by the high batch size values, which combines many (potentially tens) of input images to achieve optimal throughput. However, high batch size also comes with a latency penalty. So, for more real-time oriented usages, lower batch sizes (as low as a single input) are used.



High Throughput OR Low Latency



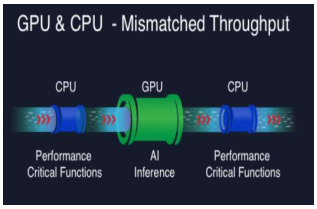
High Throughput AND Low Latency

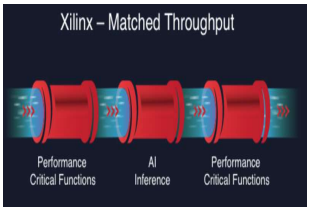
Xilinx ACAP Achieves throughput using low-batch size. Processes each input as soon as it's ready, resulting in low latency.

C. Accelerate Your Whole Application

AI tasks like inference are just a piece of the overall end-product puzzle. AI is the new, buzz-worthy, not-so-secret ingredient that is being hot-glued to a bushel basket full of embedded applications to make them somehow "better".

From Xilinx's perspective, it's not enough to accelerate just the AI portion of the job. Real performance improvement comes from accelerating as many tasks as possible.





Xilinx provides an optimized hardware acceleration of both AI inference and other performance-critical functions by tightly coupling custom accelerators into a dynamic architecture silicon device. This delivers end-to-end application performance that is significantly greater than a fixed-architecture AI accelerator like a GPU because with a GPU, the

other performance-critical functions of the application must still run in software, without the performance or efficiency of custom hardware acceleration.

CONCLUSION AND SUMMARY

Only Xilinx Versal Adaptable Devices Can:

- Match the speed of AI innovation
- Give the best performance at low latency
- Give the best power results
- Accelerate the whole application

AI Engines represent a new class of high-performance compute. Integrated within a Versal-class device, the AI Engine can be optimally combined with PL and PS to implement high-complexity systems in a single Xilinx ACAP. AI Engines deliver three to eight times better silicon area compute density when compared with traditional programmable logic DSP and ML implementations, while reducing power consumption by nominally 50%. A C/C++ programming paradigm raises the level of abstraction and promises to significantly increase the developer's productivity and unmatchable hardware capability. A true hetrogenous platform in every sense with combnation of Adaptable Engines, Integrated DDR Memory Controllers , Intelligent Engines, Programmable Network on Chip, Scalar Engines.

Xilinx ACAP Versal device has all key ingredients in adequate quantity to call itself a versatile multi-function accelerator device that allow maximum programming flexibility, and easier reconfiguration. This white paper reviews the needs driving the change from the traditional CPU-based compute model, explores the other options in detail, and how Xilinx Versal ACAP, the industry's first heterogeneous compute platform answers Industry Inference challenges.

Summary here relies on two main key aspects

- *Programmability and Performance*
- *AI Everywhere Means Inference Everywhere*

Programmability and Performance

Efficient deep learning is about solving data delivery problems, and the Xilinx Versal ACAP programmability, flexibility, and performance capabilities are all built to move data rapidly. This dream architecture is able to deliver greater performance with less power, giving the it an important edge in overall cost of operation and ownership. Better results need not coincide with larger power bills, hotter server rooms, or strained capabilities on technological infrastructure.

AI Everywhere Means Inference Everywhere

As AI continues to proliferate in real-world deployments, inference performance will become more critical to delivering enterprise insights and results. The Xilinx Versal ACAP is built to accelerate those transformative applications, with an eye on ease-of-use and power consumption. Algorithms of tomorrow demands ground-up architecture revamp of memory/data resources and capabilities. Xilinx has reformulated some fundamental chip techniques that rewrite the rules in computing and gave a disruptive dream architecture for Inference in Deep Learning Field to the Industry and Research Enthusiast.

Please visit Xilinx Versal product page for the latest updates https://www.xilinx.com/products/silicon-devices/acap/versal.html.

AUTHOR

Mr. Alok Gupta, is a Senior member of Staff with Xilinx Inc, San Jose for Artificial Intelligence & DSP , He has over 15+ years of Industry experience in System & Design Architectures of GPU/FPGA/CPU and have previously worked as Senior Graphics Architect to accelerate the innovation in Nvidia Graphics and Intel Corp. Alok holds a Post Graduate Diploma in Embedded Systems & VLSI design from Center for Development of Advanced computing, India and an Engineering degree in Electronics and Communication. His Engineering and Architecture contribution work related to Graphics and Compute at leading edge companies such as Intel, Nvidia, Xilinx, Synopsys, has given him multidisciplinary experiences in the GPU, Parallel computation, FPGA, Deep learning and Computer graphics. He has published multiple white papers in past and presented them in International conferences worldwide.

ACKNOWLEDGMENT

This paper content and research was supported by Xilinx Inc. The author of this paper would like to thank our colleague from Xilinx Ireland Office, Mr. Ambrose Finnerty who provided insight and expertise that greatly assisted the paper content.

REFERENCES

[1] https://www.rle.mit.edu/eems/wp-content/uploads/2019/04/2019_jetcas_eyerissv2.pdf
[2] http://www.eecg.toronto.edu/~mostafam/files/Memory%20Requirements%20for%20Convolutional%20Neural%20Network%20Hardware%20Accelerators.pdf.
[3] H. Esmaeilzadeh, E. Blem, R. St. Amant, et al., Dark Silicon and the End of Multicore Scaling. In International Symposium on Computer Architecture (ISCA 2011). Retrieved from gatech.edu, 2018.
[4] A. Putnam, Large-Scale Reconfigurable Computing in a Microsoft Datacenter. In IEEE Hot Chips 26 Symposium (2014). Retrieved from microsoft.com, 2018.
[5] https://www.xilinx.com/applications/megatrends/machine-learning.html.
[6] http://eyeriss.mit.edu/#papers.
[7] https://habana.ai/wp-content/uploads/pdf/habana_labs_goya_whitepaper.pdf.
[8] https://cloud.google.com/tpu/docs/tpus.
[9] https://ieeexplore.ieee.org/document/8686088
[10] https://ieeexplore.ieee.org/document/8457638
[11] http://isfpga.org/fpga2019/slides/FPGA_2019_keynote_Vissers_final.pdf
[12] https://www.xilinx.com/content/dam/xilinx/imgs/press/media-kits/Xilinx_AI_Acceleration_Publish.pdf
[13] https://engineering.fb.com/data-center-engineering/accelerating-infrastructure/
[14] https://cloud.google.com/tpu/docs/tpus